Allegato N. 3 al punto dell'ordine del giorno N. 3

Allegato n. 1

DIPARTIMENTO

DI INGEGNERIA

TABELLA DI RICLASSIFICAZIONE DELLE DISPONIBILITA' RISULTANTI AL 31/12/2017

PREDISPOSIZIONE BILANCIO UNICO DI ATENEO - ESERCIZIO CONTABILE 2017

DIPARTIMENTO
DI INGEGNERIA

Allegato n. 1

TABELLA DI RICLASSIFICAZIONE DELLE DISPONIBILITA' RISULTANTI AL 31/12/2017

		PJ	UA		
Voce COAN	Denominazione	Somme da riapplicare	Somme da riapplicare	Economia	Totale
CA.01.10.01.01.01	Costi di impianto, di ampliamento e di sviluppo				0,00
CA.01.10.01.02.01	Diritti di brevetto				0,00
CA.01.10.01.02.02	Altri diritti di utilizzazione delle opere di ingegno				0,00
CA.01.10.01.03.01	Concessioni marchi e diritti similari				0,00
CA.01.10.01.03.02	Licenze d'uso				0,00
CA.01.10.01.04.01	Immobilizzazioni immateriali in corso e acconti				0,00
CA.01.10.01.05.01	Software				0,00
CA.01.10.01.05.02	Altre immobilizzazioni immateriali				0,00
CA.01.10.01.05.03	Interventi ed opere su beni di terzi				0,00
CA.01.10.02.01.01	Terreni				0,00
CA.01.10.02.01.02	Interventi edilizi su terreni				0,00
CA.01.10.02.01.03	Fabbricati				0,00
CA.01.10.02.01.04	Interventi edilizi su Fabbricati				0,00
CA.01.10.02.01.05	Manutenzione straordinaria su fabbricati				0,00
CA.01.10.02.02.01	Impianti generici				0,00
CA.01.10.02.02.02	Manutenzione straordinaria impianti generici				0,00
CA.01.10.02.02.03	Impianti per la ricerca scientifica				0,00
CA.01.10.02.02.04	Manutenzione straordinaria impianti per la ricerca scientifica				0,00
CA.01.10.02.02.05	Attrezzature				0,00
CA.01.10.02.03.01	Attrezzatura per la ricerca scientifica	51.358,60			51.358,60
CA.01.10.02.04.01	Beni di valore culturale, storico, archeologico ed artistico				0,00
CA.01.10.02.04.02	Interventi di restauro su beni di valore culturale, storico, archeologico ed artistico				0,00
CA.01.10.02.04.03	Materiale bibliografico				0,00
CA.01.10.02.04.04	Opere artistiche				0,00
CA.01.10.02.04.05	collezioni scientifiche				0,00

DIPARTIMENTO
DI INGEGNERIA

Allegato n. 1

TABELLA DI RICLASSIFICAZIONE DELLE DISPONIBILITA' RISULTANTI AL 31/12/2017

		PJ	UA		
Voce COAN	Denominazione	Somme da riapplicare	Somme da riapplicare	Economia	Totale
CA.01.10.02.04.06	Altro materiale bibliografico				0,00
CA.01.10.02.05.01	Mobili e Arredi				0,00
CA.01.10.02.06.01	Costi e acconti per acquisizione di terreni				0,00
CA.01.10.02.06.02	Costi e acconti per Interventi edilizi su terreni				0,00
CA.01.10.02.06.03	Costi e acconti per interventi edilizi su fabbricati				0,00
CA.01.10.02.06.04	Costi e acconti per manutenzione straordinaria su fabbricati				0,00
CA.01.10.02.06.05	Costi e acconti per acquisizione di fabbricati				0,00
CA.01.10.02.06.06	Costi e acconti per acquisizione di impianti generici				0,00
CA.01.10.02.06.07	Costi e acconti per acquisizione di impianti per la ricerca scientifica				0,00
CA.01.10.02.06.08	Costi e acconti per altre immobilizzazioni materiali				0,00
CA.01.10.02.07.01	Apparecchiature di natura informatica				0,00
CA.01.10.02.07.02	Autovetture di rappresentanza e di servizio				0,00
CA.01.10.02.07.03	Autocarri, mezzi agricoli e altri mezzi di trasporto				0,00
CA.01.10.02.07.04	Altri beni mobili				0,00
CA.01.10.03.01.01	Partecipazioni in imprese ed enti controllati				0,00
CA.01.10.03.01.02	Partecipazioni in altre imprese ed enti				0,00
CA.01.10.03.01.03	Altri titoli				0,00
CA.01.10.03.01.04	Partecipazione in imprese ed enti collegati				0,00
CA.01.11.01.01.01	F.do di riserva vincolato ad investimenti				0,00
CA.01.12.01.01.01	Trasferimenti interni budget investimenti				0,00
CA.08.80.01.01.01	Costi di investimento progetti - quota di competenza per finanziamenti competitivi da miur - progetti di ricerca di rilevante interesse nazionale	77.937,86			77.937,86
CA.08.80.01.01.02	Costi di investimento progetti - quota di competenza per finanziamenti competitivi da miur - fondo per gli investimenti della ricerca di base (firb)				0,00
CA.08.80.01.01.03	" Costi di investimento progetti - quota di competenza per altri finanziamenti competitivi da miur"	18.926,40			18.926,40
CA.08.80.01.02.01	Costi di investimento progetti - quota di competenza per finanziamenti competitivi da altri ministeri per ricerca scientifica				0,00
CA.08.80.01.02.02	Costi di investimento progetti - quota di competenza per finanziamenti competitivi da stato (organi diversi da ministeri) per ricerca scientifica				0,00

DIPARTIMENTO
DI INGEGNERIA

Allegato n. 1

TABELLA DI RICLASSIFICAZIONE DELLE DISPONIBILITA' RISULTANTI AL 31/12/2017

		PJ	UA		
Voce COAN	Denominazione	Somme da riapplicare	Somme da riapplicare	Economia	Totale
CA.08.80.01.02.03	Costi di investimento progetti - quota di competenza per finanziamenti competitivi per ricerca da regioni e province autonome				0,00
CA.08.80.01.02.04	Costi di investimento progetti - quota di competenza per finanziamenti competitivi per ricerca da province				0,00
CA.08.80.01.02.05	Costi di investimento progetti - quota di competenza per finanziamenti competitivi per ricerca da città metropolitane				0,00
CA.08.80.01.02.06	Costi di investimento progetti - quota di competenza per finanziamenti competitivi per ricerca da comuni				0,00
CA.08.80.01.02.07	Costi di investimento progetti - quota di competenza per finanziamenti competitivi per ricerca da camere di commercio				0,00
CA.08.80.01.02.08	Costi di investimento progetti - quota di competenza per finanziamenti competitivi per ricerca da altre università				0,00
CA.08.80.01.02.09	Costi di investimento progetti - quota di competenza per finanziamenti competitivi per ricerca da altre amministrazioni pubbliche				0,00
CA.08.80.01.03.01	Costi di investimento progetti - quota di competenza per finanziamenti competitivi da cnr				0,00
CA.08.80.01.03.02	Costi di investimento progetti - quota di competenza per finanziamenti competitivi per ricerca da enti di ricerca diversi dal cnr				0,00
CA.08.80.01.04.01	Costi di investimento progetti - quota di competenza per finanziamenti competitivi per ricerca da parte dell'unione europea	50.000,00			50.000,00
CA.08.80.01.04.02	Costi di investimento progetti - quota di competenza per finanziamenti competitivi per ricerca da parte di organismi internazionali	16.040,82			16.040,82
CA.08.80.01.05.01	Costi di investimento progetti - attivita' in conto terzi e cessione di risultati di ricerca	201.033,73			201.033,73
CA.08.80.01.06.01	Costi di investimento progetti - finanziamenti non competitivi per la ricerca	22.121,84			22.121,84
CA.08.80.01.07.01	Costi di investimento progetti - Centri Autonomi di Gestione con Autonomia Negoziale				0,00
CA.10.10.01.01.01	Costruzione, ristrutturazione e restauro fabbricati				0,00
CA.10.10.01.01.02	Costruzione impianti				0,00
CA.10.10.01.01.03	Ricostruzione e trasformazione fabbricati				0,00
CA.10.10.01.01.04	Ricostruzione e trasformazione impianti				0,00
CA.10.10.01.01.05	Manutenzione straordinaria immobili				0,00
CA.10.10.01.01.06	Manutenzione straordinaria impianti				0,00
CA.10.10.01.01.07	Spese in applicazione D.L. 626/94				0,00
CA.10.10.01.01.08	Manutenzione straordinaria immobili - Messa a norma e sicurezza - Spese in applicazione D.Lgs. 81/2008				0,00
CA.10.10.01.01.09	Informatizzazione Servizi - Budget investimenti	172,17			172,17
CA.10.10.01.01.10	Gestione e sviluppo Rete di Ateneo - Budget investimenti				0,00
CA.10.10.01.01.11	Mobilità e scambi culturali docenti - Budget investimenti				0,00

DIPARTIMENTO
DI INGEGNERIA

Allegato n. 1

TABELLA DI RICLASSIFICAZIONE DELLE DISPONIBILITA' RISULTANTI AL 31/12/2017

		PJ	UA		
Voce COAN	Denominazione	Somme da riapplicare	Somme da riapplicare	Economia	Totale
CA.10.10.01.01.12	Rapporti Internazionali, scambi culturali - Budget investimenti				0,00
CA.10.10.01.01.13	Comunicazione di Ateneo - Budget investimenti				0,00
CA.10.10.01.01.14	Acquisto, manutenzione, noleggio, esercizio veicoli - Budget investimenti				0,00
CA.10.10.01.01.15	Spese inerenti l'orientamento universitario - Budget investimenti				0,00
CA.10.10.01.01.16	Progetti III Missione - Budget investimenti				0,00
CA.10.10.01.01.17	Spese funzionamento Servizio Prevenzione e Protezione - Budget investimenti				0,00
CA.10.10.01.01.18	Funzionamento Strutture Didattiche finanziate da Esterni - Budget investimenti				0,00
CA.10.10.01.01.19	Ricerca di base - Budget investimenti	3.068,37			3.068,37
CA.10.10.01.01.20	Funzionamento strutture didattiche - Budget investimenti	51.438,82			51.438,82
CA.10.10.01.01.21	Costi operativi su economie progetti - Budget investimenti				0,00
CA.10.10.01.01.22	Costi operativi altri progetti Amministrazione centrale - Budget investimenti				0,00
CA.04.06.01.01.01	Rimanenze iniziali materiale di consumo				0,00
CA.04.06.01.02.01	Rimanenze iniziali prodotti in corso di lavorazione				0,00
CA.04.06.01.03.01	Rimanenze iniziali prodotti finiti				0,00
CA.04.06.01.04.01	Rimanenze iniziali lavori in corso su ordinazione				0,00
CA.04.06.01.05.01	Rimanenze iniziali merci				0,00
CA.04.08.01.01.01	Costo per competenze fisse del personale docente a tempo indeterminato				0,00
CA.04.08.01.01.02	Costo per competenze fisse del personale docente a tempo determinato				0,00
CA.04.08.01.01.03	Costo per supplenze e affidamenti a personale docente a tempo indeterminato	2.025,22			2.025,22
CA.04.08.01.01.04	Costo per supplenze e affidamenti a personale docente a tempo determinato				0,00
CA.04.08.01.01.05	Indennita' di missione, rimborsi spese viaggi e iscrizione a convegni del personale docente e ricercatori				0,00
CA.04.08.01.01.06	Costo per competenze fisse del personale ricercatore a tempo indeterminato				0,00
CA.04.08.01.01.07	Costo per supplenze e affidamenti a personale ricercatore a tempo indeterminato				0,00
CA.04.08.01.01.08	Costo per competenze fisse del personale ricercatore a tempo determinato				0,00
CA.04.08.01.01.09	Costo per supplenze e affidamenti a personale ricercatore a tempo determinato				0,00

DIPARTIMENTO
DI INGEGNERIA

Allegato n. 1

TABELLA DI RICLASSIFICAZIONE DELLE DISPONIBILITA' RISULTANTI AL 31/12/2017

		PJ	UA		
Voce COAN	Denominazione	Somme da riapplicare	Somme da riapplicare	Economia	Totale
CA.04.08.01.01.10	Costo delle competenze accessorie del personale docente e ricercatore				0,00
CA.04.08.01.01.11	Indennità di rischio del personale docente e dei ricercatori				0,00
CA.04.08.01.01.12	Indennità di rischio radiologico del personale docente e dei ricercatori- non convenzionato				0,00
CA.04.08.01.01.13	Punti organico per personale docente e ricercatore				0,00
CA.04.08.01.01.14	Fondo di Ateneo per la premialità				0,00
CA.04.08.01.02.01	Assegni di ricerca	12.317,88	23.591,88		35.909,76
CA.04.08.01.02.02	Indennita' di missione, rimborsi spese viaggi per gli assegni di ricerca				0,00
CA.04.08.01.03.01	Costo del personale docente a contratto				0,00
CA.04.08.01.04.01	Costo per i collaboratori ed esperti linguistici a tempo indeterminato				0,00
CA.04.08.01.04.02	Competenze fisse a collaboratori ed esperti linguistici di madre lingua a tempo determinato (fd)				0,00
CA.04.08.01.04.03	Costo per supplenze e affidamenti a collaboratori ed esperti linguistici a tempo indeterminato				0,00
CA.04.08.01.04.04	Costo per supplenze e affidamenti a collaboratori ed esperti linguistici a tempo determinato				0,00
CA.04.08.01.04.05	Indennita' di missione, rimborsi spese viaggi per collaboratori ed esperti linguistici a tempo indeterminato				0,00
CA.04.08.01.04.06	Indennita' di missione, rimborsi spese viaggi per collaboratori ed esperti linguistici a tempo determinato				0,00
CA.04.08.01.04.07	Costi di formazione esperti linguistici				0,00
CA.04.08.01.05.01	Costo per competenze fisse per altro personale dedicato alla ricerca ed alla didattica				0,00
CA.04.08.01.05.02	Competenze accessorie per altro personale dedicato alla ricerca ed alla didattica				0,00
CA.04.08.01.06.01	Compensi a personale docente convenzionato ssn (per attività assistenziale)				0,00
CA.04.08.01.06.02	Compensi a personale ricercatore a tempo indeterminato convenzionato ssn (per attivtà assistenziale)				0,00
CA.04.08.01.06.03	Compensi a personale ricercatore a tempo determinato convenzionato ssn (per attività assistenziale)				0,00
CA.04.08.01.07.01	Costo delle competenze per personale docente e ricercatore su attivita' conto terzi				0,00
CA.04.08.02.01.01	Costo dei dirigenti a tempo indeterminato				0,00
CA.04.08.02.02.01	Costo del direttore generale e dei dirigenti a tempo determinato				0,00
CA.04.08.02.03.01	Costo del personale tecnico-amministrativo a tempo indeterminato				0,00
CA.04.08.02.04.01	Costo del personale tecnico-amministrativo a tempo determinato				0,00

DIPARTIMENTO
DI INGEGNERIA

Allegato n. 1

TABELLA DI RICLASSIFICAZIONE DELLE DISPONIBILITA' RISULTANTI AL 31/12/2017

		PJ	UA		
Voce COAN	Denominazione	Somme da riapplicare	Somme da riapplicare	Economia	Totale
CA.04.08.02.05.01	Competenze accessorie del personale dirigente				0,00
CA.04.08.02.05.02	Competenze accessorie al personale EP				0,00
CA.04.08.02.05.03	Competenze accessorie al personale tecnico-amministrativo				0,00
CA.04.08.02.05.04	Indennità centralinisti non vedenti				0,00
CA.04.08.02.05.05	Indennità di rischio radiologico del personale tecnico-amministrativo a tempo indeterminato - non convenzionato				0,00
CA.04.08.02.06.01	Indennita' di missione, rimborsi spese viaggi del personale dirigente e tecnico-amministrativo				0,00
CA.04.08.02.06.02	Buoni pasto per il personale tecnico-amministrativo				0,00
CA.04.08.02.06.03	Formazione del personale dirigente e tecnico-amministrativo				0,00
CA.04.08.02.06.04	Punti organico per personale dirigente, tecnico-amministrativo e cel				0,00
CA.04.08.02.07.01	Compesi attività conto terzi personale tecnico amministrativo				0,00
CA.04.08.02.08.01	Compensi a personale tecnico-amministrativo a tempo indeterminato convenzionato ssn (per attività assistenziale)				0,00
CA.04.08.02.08.02	Compensi a personale tecnico-amministrativo a tempo determinato convenzionato ssn (per attività assistenziale)				0,00
CA.04.08.02.09.01	Compenso a personale tecnico amministrativo ai sensi del Codice dei contratti				0,00
CA.04.09.01.01.01	Costi per borse di studio per scuole di specializzazione mediche a norma ue				0,00
CA.04.09.01.01.02	Costi per borse di studio per scuole di specializzazione				0,00
CA.04.09.01.01.03	Costi per borse di studio per dottorato di ricerca				0,00
CA.04.09.01.01.04	Borse di studio per post dottorato				0,00
CA.04.09.01.01.05	Costi per altre borse				0,00
CA.04.09.01.01.06	Indennita' di missione, rimborsi spese viaggi per borse di studio per scuole di specializzazione mediche a norma ue				0,00
CA.04.09.01.01.07	Indennita' di missione, rimborsi spese viaggi per borse di studio per scuole di specializzazione				0,00
CA.04.09.01.01.08	Indennita' di missione, rimborsi spese viaggi per borse di studio per post dottorato				0,00
CA.04.09.01.01.09	Indennita' di missione, rimborsi spese viaggii per altre borse				0,00
CA.04.09.01.01.10	Indennita' di missione, rimborsi spese viaggi per dottorato di ricerca				0,00
CA.04.09.01.02.01	Programmi di mobilita' e scambi culturali studenti				0,00
CA.04.09.01.02.02	Iniziative ed attivita' culturali gestite dagli studenti				0,00

DIPARTIMENTO
DI INGEGNERIA

Allegato n. 1

TABELLA DI RICLASSIFICAZIONE DELLE DISPONIBILITA' RISULTANTI AL 31/12/2017

		PJ	UA		
Voce COAN	Denominazione	Somme da riapplicare	Somme da riapplicare	Economia	Totale
CA.04.09.01.02.03	Interventi a favore degli studenti diversamente abili				0,00
CA.04.09.01.02.04	Assegni per l'incentivazione dell'attivita' di tutorato				0,00
CA.04.09.01.02.05	Altri interventi a favore degli studenti				0,00
CA.04.09.01.02.06	Altri premi				0,00
CA.04.09.01.03.01	Convegni e seminari				0,00
CA.04.09.01.03.02	Ospitalita' visiting professor, esperti e relatori convegni				0,00
CA.04.09.01.03.03	Compensi e soggiorno a visiting professor, esperti e relatori convegni				0,00
CA.04.09.02.01.01	Borse di collaborazione studenti, attivita' a tempo parziale art. 11 D.Lgs 29/03/2012 n° 68				0,00
CA.04.09.03.01.01	Costi per la ricerca e l'attività editoriale				0,00
CA.04.09.04.01.01	Trasferimenti a patner di progetti coordinati				0,00
CA.04.09.05.01.01	Materiale di consumo per laboratorio	221,80			221,80
CA.04.09.06.01.01	Rimanenze iniziali materiale di consumo per laboratorio				0,00
CA.04.09.06.02.01	Rimanenze finali materiale di consumo per laboratori				0,00
CA.04.09.07.01.01	Riviste e giornali				0,00
CA.04.09.07.01.02	Libri e altro materiale bibliografico non costituenti immobilizzazioni				0,00
CA.04.09.08.01.01	Utenze e canoni per energia elettrica				0,00
CA.04.09.08.02.01	Utenze e canoni per telefonia fissa				0,00
CA.04.09.08.02.02	Utenze e canoni per telefonia mobile				0,00
CA.04.09.08.02.03	Utenze e canoni per reti di trasmissione				0,00
CA.04.09.08.03.01	Utenze e canoni per acqua				0,00
CA.04.09.08.03.02	Utenze e canoni per gas				0,00
CA.04.09.08.03.03	Riscaldamento e condizionamento				0,00
CA.04.09.08.03.04	Altre utenze e canoni				0,00
CA.04.09.08.04.01	Pulizia				0,00
CA.04.09.08.04.02	Smaltimento rifiuti nocivi			2.907,95	2.907,95

Allegato n. 1

TABELLA DI RICLASSIFICAZIONE DELLE DISPONIBILITA' RISULTANTI AL 31/12/2017

		PJ	UA		
Voce COAN	Denominazione	Somme da riapplicare	Somme da riapplicare	Economia	Totale
CA.04.09.08.04.03	Traslochi e facchinaggio				0,00
CA.04.09.08.04.04	Vigilanza e altri servizi ausiliari				0,00
CA.04.09.08.05.01	Manutenzione ordinaria e riparazioni di immobili				0,00
CA.04.09.08.05.02	Manutenzione ordinaria e riparazioni di impianti				0,00
CA.04.09.08.05.03	Manutenzione ordinaria e riparazioni di apparecchiature				0,00
CA.04.09.08.05.04	Manutenzione ordinaria e riparazioni di autovetture di rappresentanza e di servizio				0,00
CA.04.09.08.05.05	Manutenzione ordinaria e riparazioni di autocarri, mezzi agricoli e altri mezzi di trasporto				0,00
CA.04.09.08.05.06	Manutenzione ordinaria e riparazioni mobili e arredi				0,00
CA.04.09.08.05.07	Altre spese di manutenzione ordinaria e riparazioni				0,00
CA.04.09.08.06.01	Rappresentanza				0,00
CA.04.09.08.06.02	Organizzazione manifestazioni e convegni				0,00
CA.04.09.08.06.03	Spese postali				0,00
CA.04.09.08.06.04	Assicurazioni				0,00
CA.04.09.08.06.05	Spese per le pubblicazioni dell'ateneo				0,00
CA.04.09.08.06.06	Spesa corrente per brevetti				0,00
CA.04.09.08.06.07	Altre spese per servizi				0,00
CA.04.09.08.06.08	Costi annuali per pubblicita'				0,00
CA.04.09.08.06.09	Spese per pubblicita' degli atti				0,00
CA.04.09.08.06.10	Spese per lavorazioni agricole effettuate da terzi				0,00
CA.04.09.08.06.11	Spese per commissioni ed intermediazioni bancarie				0,00
CA.04.09.08.07.01	Consulenze tecnico-scientifiche				0,00
CA.04.09.08.07.02	Consulenze tecnico-amministrative				0,00
CA.04.09.08.07.03	Spese per liti (patrocinio legale)				0,00
CA.04.09.08.07.04	Spese notarili				0,00
CA.04.09.08.08.01	Prestazioni di lavoro autonomo				0,00

DIPARTIMENTO
DI INGEGNERIA

Allegato n. 1

TABELLA DI RICLASSIFICAZIONE DELLE DISPONIBILITA' RISULTANTI AL 31/12/2017

		PJ	UA		
Voce COAN	Denominazione	Somme da riapplicare	Somme da riapplicare	Economia	Totale
CA.04.09.08.09.01	Prestazioni di servizi tecnico/amministrativi da enti terzi				0,00
CA.04.09.08.09.02	Altre prestazioni e servizi da terzi				0,00
CA.04.09.08.10.01	Collaborazioni coordinate e continuative				0,00
CA.04.09.08.11.01	Costi per fornitura di lavoro interinale				0,00
CA.04.09.09.01.01	Carburanti, combustibili e lubrificanti per autovetture				0,00
CA.04.09.09.01.02	Carburanti, combustibili e lubrificanti per autocarri, mezzi agricoli e altri mezzi di trasporto				0,00
CA.04.09.09.01.03	Cancelleria e altro materiale di consumo				0,00
CA.04.09.09.01.04	Libretti e diplomi				0,00
CA.04.09.09.01.05	Vestiario				0,00
CA.04.09.09.01.06	Materiale per ricorrenze elettorali				0,00
CA.04.09.09.01.07	Altri materiali				0,00
CA.04.09.09.01.08	Combustibili per riscaldamento				0,00
CA.04.09.09.02.01	Acquisto beni strumentali				0,00
CA.04.09.09.02.02	Acquisto software per pc				0,00
CA.04.09.09.03.01	Sconti e abbuoni passivi				0,00
CA.04.09.09.03.02	Sconti abbuoni e premi su acquisti				0,00
CA.04.09.10.01.01	Rimanenze iniziali di materiali				0,00
CA.04.09.11.01.01	Leasing operativo ed altre forme di locazione di beni mobili				0,00
CA.04.09.11.01.02	Leasing operativo ed altre forme di locazione di autovetture di rappresentanza e di servizio				0,00
CA.04.09.11.01.03	Leasing operativo ed altre forme di locazione di autocarri, mezzi agricoli e altri mezzi di trasporto				0,00
CA.04.09.11.01.04	Noleggio fax e fotocopiatrici				0,00
CA.04.09.11.01.05	Altri noleggi				0,00
CA.04.09.11.02.01	Fitti passivi per locazione di edifici				0,00
CA.04.09.11.02.02	Altri fitti passivi				0,00
CA.04.09.11.03.01	Licenze software				0,00

DIPARTIMENTO
DI INGEGNERIA

Allegato n. 1

TABELLA DI RICLASSIFICAZIONE DELLE DISPONIBILITA' RISULTANTI AL 31/12/2017

		PJ	UA		
Voce COAN	Denominazione	Somme da riapplicare	Somme da riapplicare	Economia	Totale
CA.04.09.12.01.01	Missioni e rimborsi spese trasferta organi istituzionali				0,00
CA.04.09.12.01.02	Gettoni/indennita' ai membri degli organi istituzionali di governo e controllo				0,00
CA.04.09.12.01.03	Indennita' di carica				0,00
CA.04.09.12.01.04	Gettoni/indennita' ai membri degli organi istituzionali che non siano di amministrazione e controllo				0,00
CA.04.09.12.01.05	Gettoni/indennita' ai membri del collegio dei revisori				0,00
CA.04.09.12.01.06	Gettoni/indennita' ai membri del nucleo di valutazione				0,00
CA.04.09.12.01.07	Garante di Ateneo				0,00
CA.04.09.12.02.01	Quote associative				0,00
CA.04.09.12.02.02	Compensi per commissioni di concorso del personale interno ed esterno				0,00
CA.04.09.12.02.03	Altri costi per attivita' istituzionali				0,00
CA.04.09.12.02.04	Costi per attivita' sportive l. 394/77				0,00
CA.04.09.12.02.05	Cus - attivita' sportiva e gestione impianti sportivi				0,00
CA.04.09.12.02.06	Arrotondamenti negativi				0,00
CA.04.09.12.02.07	Visite medico-fiscali				0,00
CA.04.09.12.02.08	Accertamenti sanitari				0,00
CA.04.09.12.02.09	Equo indennizzo				0,00
CA.04.09.12.02.10	Provvidenze a favore del personale				0,00
CA.04.09.12.02.11	Circolo San Martino - attività sociali del personale				0,00
CA.04.09.12.02.12	Prestazioni INAIL - gestione per conto				0,00
CA.04.09.12.02.13	Spese condominiali				0,00
CA.04.10.01.01.01	QUOTE DI AMMORTAMENTO COSTI DI IMPIANTO, DI AMPLIAMENTO E DI SVILUPPO				0,00
CA.04.10.01.01.02	QUOTE DI AMMORTAMENTO DIRITTI DI BREVETTO E DIRITTI DI UTILIZZAZIONE DELLE OPERE DI INGEGNO				0,00
CA.04.10.01.01.03	QUOTE DI AMMORTAMENTO CONCESSIONI, LICENZE, MARCHI E DIRITTI SIMILI				0,00
CA.04.10.01.01.04	QUOTE DI AMMORTAMENTO ALTRE IMMOBILIZZAZIONI IMMATERIALI				0,00
CA.04.10.02.01.01	QUOTE DI AMMORTAMENTO TERRENI E FABBRICATI				0,00

DIPARTIMENTO
DI INGEGNERIA

Allegato n. 1

TABELLA DI RICLASSIFICAZIONE DELLE DISPONIBILITA' RISULTANTI AL 31/12/2017

		PJ	UA		
Voce COAN	Denominazione	Somme da riapplicare	Somme da riappilcare	Economia	Totale
CA.04.10.02.01.02	QUOTE DI AMMORTAMENTO IMPIANTI E ATTREZZATURE				0,00
CA.04.10.02.01.03	QUOTE DI AMMORTAMENTO ATTREZZATURE SCIENTIFICHE				0,00
CA.04.10.02.01.04	QUOTE DI AMMORTAMENTO PATRIMONIO LIBRARIO, OPERE D'ARTE, D'ANTIQUARIATO E MUSEALI				0,00
CA.04.10.02.01.05	QUOTE DI AMMORTAMENTO MOBILI E ARREDI				0,00
CA.04.10.02.01.06	QUOTE DI AMMORTAMENTO ALTRE IMMOBILIZZAZIONI MATERIALI				0,00
CA.04.10.03.01.01	SVALUTAZIONE COSTI DI IMPIANTO, DI AMPLIAMENTO E DI SVILUPPO				0,00
CA.04.10.03.01.02	SVALUTAZIONE DIRITTI DI BREVETTO E DIRITTI DI UTILIZZAZIONE DELLE OPERE DI INGEGNO				0,00
CA.04.10.03.01.03	SVALUTAZIONE CONCESSIONI, LICENZE, MARCHI E DIRITTI SIMILI				0,00
CA.04.10.03.01.04	SVALUTAZIONE ALTRE IMMOBILIZZAZIONI IMMATERIALI				0,00
CA.04.10.03.01.05	SVALUTAZIONE TERRENI E FABBRICATI				0,00
CA.04.10.03.01.06	SVALUTAZIONE IMPIANTI E ATTREZZATURE				0,00
CA.04.10.03.01.07	SVALUTAZIONE ATTREZZATURE SCIENTIFICHE				0,00
CA.04.10.03.01.08	SVALUTAZIONE PATRIMONIO LIBRARIO, OPERE D'ARTE, D'ANTIQUARIATO E MUSEALI				0,00
CA.04.10.03.01.09	SVALUTAZIONE MOBILI E ARREDI				0,00
CA.04.10.03.01.10	SVALUTAZIONE ALTRE IMMOBILIZZAZIONI MATERIALI				0,00
CA.04.10.04.01.01	Perdita su crediti compresi nell'attivo circolante e nelle disponibilità liquide				0,00
CA.04.11.01.01.01	Quote di accantonamento al fondo svalutazione crediti				0,00
CA.04.11.01.01.02	QUOTE DI ACCANTONAMENTO AI FONDI ARRETRATI DA CORRISPONDERE AL PERSONALE NEGLI ESERCIZI FUTURI				0,00
CA.04.11.01.03.01	Quote di accantonamento ai fondi per cause e controversie in corso				0,00
CA.04.11.01.04.01	Quote di accantonamento ai fondi per altri rischi e oneri				0,00
CA.04.11.01.05.01	Quota di esercizio per altri accantonamenti				0,00
CA.04.11.01.06.01	Accantonamento per fondi di quiescenza				0,00
CA.04.11.01.06.02	Accantonamento per fondi tfr personale				0,00
CA.04.12.01.01.01	TRASFERIMENTI INTERNI CORRENTI				0,00
CA.04.12.01.01.02	TRASFERIMENTI INTERNI PER INVESTIMENTI				0,00

DIPARTIMENTO
DI INGEGNERIA

Allegato n. 1

TABELLA DI RICLASSIFICAZIONE DELLE DISPONIBILITA' RISULTANTI AL 31/12/2017

Voce COAN	Denominazione	PJ	UA		
		Somme da riapplicare	Somme da riapplicare	Economia	Totale
CA.04.12.01.01.03	TRASFERIMENTI INTERNI SU ATTIVITA' CONTO TERZI				0,00
CA.04.12.01.01.04	TRASFERIMENTI INTERNI PER RESTITUZIONI E RIMBORSI				0,00
CA.04.12.01.01.05	ALTRI TRASFERIMENTI INTERNI				0,00
CA.04.12.01.01.06	TRASFERIMENTI VARI				0,00
CA.04.12.01.02.01	Versamenti allo Stato per riduzioni di spesa				0,00
CA.04.12.01.03.01	Tassa di rimozione rifiuti solidi urbani				0,00
CA.04.12.01.03.02	Imposte sul registro				0,00
CA.04.12.01.03.03	Valori bollati				0,00
CA.04.12.01.03.04	Altri tributi				0,00
CA.04.12.01.03.05	Iva indetraibile				0,00
CA.04.12.01.03.06	Imposte sul patrimonio				0,00
CA.04.12.01.03.07	Tributi su lasciti e donazioni				0,00
CA.04.13.02.01.01	Oneri finanziari				0,00
CA.04.13.02.02.01	Interessi passivi				0,00
CA.04.13.03.01.01	Perdite su cambi				0,00
CA.04.14.02.01.01	Svalutazione titoli e partecipazioni				0,00
CA.04.15.02.01.01	Oneri straodinari per il personale				0,00
CA.04.15.02.02.01	Rimborsi tasse e contributi agli studenti				0,00
CA.04.15.02.03.01	Oneri straordinari per recuperi e rimborsi				0,00
CA.04.15.02.04.01	Altri oneri straordinari				0,00
CA.04.15.02.05.01	Imposte relative ad esercizi precedenti			44.192,57	44.192,57
CA.04.16.01.01.01	Imposte sul reddito				0,00
CA.04.17.01.01.01	Voce per variazione tecnica				0,00
CA.07.70.01.01.01	Costi operativi progetti - quota di competenza per finanziamenti competitivi da miur - progetti di ricerca di rilevante interesse nazionale	182.210,69			182.210,69
CA.07.70.01.01.02	Costi operativi progetti - quota di competenza per finanziamenti competitivi da miur - fondo per gli investimenti della ricerca di base (firb)				0,00

DIPARTIMENTO
DI INGEGNERIA

Allegato n. 1

TABELLA DI RICLASSIFICAZIONE DELLE DISPONIBILITA' RISULTANTI AL 31/12/2017

		PJ	UA		
Voce COAN	Denominazione	Somme da riapplicare	Somme da riapplicare	Economia	Totale
CA.07.70.01.01.03	Costi operativi progetti - quota di competenza per altri finanziamenti competitivi da miur	1.781.846,61			1.781.846,61
CA.07.70.01.02.01	Costi operativi progetti - quota di competenza per finanziamenti competitivi da altri ministeri per ricerca scientifica				0,00
CA.07.70.01.02.02	Costi operativi progetti - quota di competenza per finanziamenti competitivi da stato (organi diversi da ministeri) per ricerca scientifica				0,00
CA.07.70.01.02.03	Costi operativi progetti - quota di competenza per finanziamenti competitivi per ricerca da regioni e province autonome				0,00
CA.07.70.01.02.04	Costi operativi progetti - quota di competenza per finanziamenti competitivi per ricerca da province				0,00
CA.07.70.01.02.05	Costi operativi progetti - quota di competenza per finanziamenti competitivi per ricerca da città metropolitane				0,00
CA.07.70.01.02.06	Costi operativi progetti - quota di competenza per finanziamenti competitivi per ricerca da comuni				0,00
CA.07.70.01.02.07	Costi operativi progetti - quota di competenza per finanziamenti competitivi per ricerca da camere di commercio				0,00
CA.07.70.01.02.08	Costi operativi progetti - quota di competenza per finanziamenti competitivi per ricerca da altre università				0,00
CA.07.70.01.02.09	Costi operativi progetti - quota di competenza per finanziamenti competitivi per ricerca da altre amministrazioni pubbliche				0,00
CA.07.70.01.03.01	Costi operativi progetti - quota di competenza per finanziamenti competitivi da cnr				0,00
CA.07.70.01.03.02	Costi operativi progetti - quota di competenza per finanziamenti competitivi per ricerca da enti di ricerca diversi dal cnr				0,00
CA.07.70.01.04.01	Costi operativi progetti - quota di competenza per finanziamenti competitivi per ricerca da parte dell'unione europea	1.388.366,47			1.388.366,47
CA.07.70.01.04.02	Costi operativi progetti - quota di competenza per finanziamenti competitivi per ricerca da parte di organismi internazionali	1.197.515,31			1.197.515,31
CA.07.70.01.05.01	Costi operativi progetti - attivita' c/terzi e cessione di risultati di ricerca	4.273.327,73			4.273.327,73
CA.07.70.01.06.01	Costi operativi progetti - finanziamenti non competitivi per la ricerca	1.893.916,53			1.893.916,53
CA.07.70.01.07.01	Costi operativi progetti - Centri Autonomi di Gestione con Autonomia Negoziale				0,00
CA.07.70.01.08.01	Costi operativi progetti per attività di formazione				0,00
CA.09.90.01.01.01	Mobilità e scambi culturali docenti - Budget economico				0,00
CA.09.90.01.01.02	Rapporti Internazionali, scambi culturali - Budget economico				0,00
CA.09.90.01.01.03	Comunicazione di Ateneo - Budget economico				0,00
CA.09.90.01.01.04	Acquisto, manutenzione, noleggio, esercizio veicoli - Budget economico				0,00
CA.09.90.01.01.05	Spese inerenti l'orientamento universitario - Budget economico				0,00
CA.09.90.01.01.06	Progetti III Missione - Budget economico				0,00
CA.09.90.01.01.07	Spese funzionamento Servizio Prevenzione e Protezione - Budget economico			27.783,07	27.783,07

DIPARTIMENTO
DI INGEGNERIA

Allegato n. 1

TABELLA DI RICLASSIFICAZIONE DELLE DISPONIBILITA' RISULTANTI AL 31/12/2017

		PJ	UA		
Voce COAN	**Denominazione**	**Somme da riapplicare**	**Somme da riapplicare**	**Economia**	**Totale**
CA.09.90.01.01.08	Funzionamento Strutture Didattiche finanziate da Esterni - Budget economico				0,00
CA.09.90.01.01.09	Ricerca di base - Budget economico	145.347,00			145.347,00
CA.09.90.01.01.10	Funzionamento strutture didattiche - Budget economico	215.620,73			215.620,73
CA.09.90.01.01.11	Costi operativi su economie progetti - Budget economico				0,00
CA.09.90.01.01.12	Costi operativi altri progetti Amministrazione centrale - Budget economico				0,00
CA.09.90.01.01.13	Informatizzazione Servizi - Budget economico				0,00
CA.09.90.01.01.14	Gestione e sviluppo Rete di Ateneo - Budget economico				0,00
	TOTALE DISPONIBILITA' AL 31/12/2017	**11.584.814,58**	**23.591,88**	**74.883,59**	11.683.290,05

IL SEGRETARIO AMM.VO
(Sig. Giovanni Magara)

UNIVERSITA' DEGLI STUDI DI PERUGIA
DIPARTIMENTO DI INGEGNERIA

IL DIRETTORE
(Prof. Giuseppe Saccomandi)

Allegato N. ....4..... al punto dell'ordine del giorno N. ....3.....

Allegato n. 2

DIPARTIMENTO

DI INGEGNERIA

TABELLA DI RICLASSIFICAZIONE DELLE RETTIFICHE SU ANTICIPATE DI RIPORTO (VOCI COAN E UA) AL 31/12/2017

PREDISPOSIZIONE BILANCIO UNICO DI ATENEO - ESERCIZIO CONTABILE 2017

DIPARTIMENTO/CENTRO
DI INGEGNERIA

Allegato n. 2

TABELLA DI RICLASSIFICAZIONE DELLE RETTIFICHE SU ANTICIPATE DI RIPORTO (VOCI COAN E UA) AL 31/12/2017

		UA	
Voce COAN	Denominazione	Rettifiche Anticipate di Riporto di RICAVO	Totale
CA.03.01.01.01.01	Tasse e contributi per l'iscrizione		0,00
CA.03.01.01.02.01	Altri ricavi da studenti		0,00
CA.03.01.02.01.01	Ricerche e trasferimento tecnologico in conto/terzi		0,00
CA.03.01.03.01.01	Finanziamenti competitivi da miur - progetti di ricerca di rilevante interesse nazionale		0,00
CA.03.01.03.01.02	Finanziamenti competitivi da miur - fondo per gli investimenti della ricerca di base (firb)		0,00
CA.03.01.03.01.03	Altri finanziamenti competitivi da miur		0,00
CA.03.01.03.02.01	Finanziamenti per ricerca derivanti da bandi di istituzioni pubbliche nazionali diverse dal miur		0,00
CA.03.01.03.03.01	Finanziamenti competitivi erogati da enti di ricerca		0,00
CA.03.01.03.04.01	Finanziamenti competitivi erogati da organismi internazionali		0,00
CA.03.02.01.01.01	Fondo finanziamento ordinario delle universita'		0,00
CA.03.02.01.01.02	Fondo per borse di dottorato di ricerca		0,00
CA.03.02.01.01.03	Fondo sostegno giovani		0,00
CA.03.02.01.01.04	Fondo per attivita' sportiva		0,00
CA.03.02.01.01.05	Fondo per la programmazione delle universita'		0,00
CA.03.02.01.01.06	Fondo per edilizia universitaria		0,00
CA.03.02.01.01.08	Altri fondi per il finanziamento delle universita'		0,00
CA.03.02.01.01.09	Fondo assegni di ricerca		0,00
CA.03.02.01.02.01	Trasferimenti correnti da organi dello stato diversi dal miur - contributi diversi		0,00
CA.03.02.01.02.02	Trasferimenti per investimenti da Stato - Finanziamenti di altri Ministeri per Ricerca Scientifica		0,00
CA.03.02.02.01.01	Contributi per investimenti da regioni e province autonome		0,00
CA.03.02.02.01.02	Contributi correnti da regioni e province autonome		0,00
CA.03.02.03.01.01	Contributi per investimenti da altre amministrazioni locali		0,00

DIPARTIMENTO/CENTRO
DI INGEGNERIA

Allegato n. 2

TABELLA DI RICLASSIFICAZIONE DELLE RETTIFICHE SU ANTICIPATE DI RIPORTO (VOCI COAN E UA) AL 31/12/2017

		UA	
Voce COAN	Denominazione	Rettifiche Anticipate di Riporto di RICAVO	Totale
CA.03.02.03.01.02	Contributi correnti da altre amministrazioni locali		0,00
CA.03.02.04.01.01	Contributi per investimenti da parte dell'unione europea		0,00
CA.03.02.04.01.02	Contributi correnti da parte dell'unione europea		0,00
CA.03.02.04.02.01	Contributi per investimenti da parte di organismi internazionali		0,00
CA.03.02.04.02.02	Contributi correnti da parte di organismi internazionali		0,00
CA.03.02.05.01.01	Contributi per investimenti da altre universita'		0,00
CA.03.02.05.01.02	Contributi correnti da altre universita'		0,00
CA.03.02.06.01.01	Contributi per investimenti altri soggetti (pubblici)		0,00
CA.03.02.06.01.02	Contributi correnti da altri soggetti (pubblici)		0,00
CA.03.02.07.01.01	Contributi per investimenti da altri (privati)		0,00
CA.03.02.07.01.02	Contributi correnti da altri (privati)		0,00
CA.03.03.01.01.01	Proventi per attivita' assistenziale		0,00
CA.03.04.01.01.01	Proventi per gestione diretta interventi per il diritto allo studio		0,00
CA.03.05.01.01.01	Contratti/convenzioni/accordi programma: con il miur		0,00
CA.03.05.01.01.02	Contratti/convenzioni/accordi programma: con altri ministeri		0,00
CA.03.05.01.01.03	Contratti/convenzioni/accordi programma: con unione europea		0,00
CA.03.05.01.01.04	Contratti/convenzioni/accordi programma: con organismi pubblici esteri o internazionali		0,00
CA.03.05.01.01.05	Contratti/convenzioni/accordi programma: con regioni e province autonome		0,00
CA.03.05.01.01.06	Contratti/convenzioni/accordi programma: con province		0,00
CA.03.05.01.01.07	Contratti/convenzioni/accordi programma: con comuni		0,00
CA.03.05.01.01.08	Contratti/convenzioni/accordi programma: con enti di ricerca		0,00
CA.03.05.01.01.09	Contratti/convenzioni/accordi programma: con altre amministrazioni pubbliche		0,00

DIPARTIMENTO/CENTRO
DI INGEGNERIA

Allegato n. 2

TABELLA DI RICLASSIFICAZIONE DELLE RETTIFICHE SU ANTICIPATE DI RIPORTO (VOCI COAN E UA) AL 31/12/2017

		UA	
Voce COAN	Denominazione	Rettifiche Anticipate di Riporto di RICAVO	Totale
CA.03.05.01.01.10	Contratti/convenzioni/accordi programma: con altri soggetti		0,00
CA.03.05.01.01.11	Contratti/convenzioni/accordi programma: con Aziende Ospedaliere		0,00
CA.03.05.01.02.01	Altre vendite di beni e servizi in attività istituzionale		0,00
CA.03.05.01.02.02	Altre vendite di beni e servizi in attività Commerciale		0,00
CA.03.05.01.03.01	Fitti attivi		0,00
CA.03.05.01.04.01	Sconti e abbuoni attivi		0,00
CA.03.05.01.05.01	Lasciti, oblazioni e donazioni		0,00
CA.03.05.01.06.01	Entrate eventuali non classificabili in altre voci		0,00
CA.03.05.01.06.02	Ricavi da concessione diritti reali di godimento		0,00
CA.03.05.01.07.01	Recuperi e rimborsi		0,00
CA.03.05.01.07.02	Recuperi e rimborsi per personale comandato		0,00
CA.03.05.01.08.01	Altre poste correttive e compensative di spese		0,00
CA.03.05.01.08.02	Resi su acquisti		0,00
CA.03.05.01.08.03	Voce di riequilibrio		0,00
CA.03.05.01.09.01	TRASFERIMENTI INTERNI CORRENTI		0,00
CA.03.05.01.10.01	Trasferimenti interni per investimenti		0,00
CA.03.05.01.11.01	Trasferimenti interni su attività conto terzi		0,00
CA.03.05.01.12.01	Trasferimenti interni vari		0,00
CA.03.05.01.13.01	Altri trasferimenti interni		0,00
CA.03.05.02.01.01	UTILIZZO DI RISERVE DI PATRIMONIO NETTO DERIVANTI DALLA CONTABILITÀ FINANZIARIA		0,00
CA.03.05.03.01.01	Ricavi per sterilizzazione ammortamenti beni acquisiti in regime di contabilità finanziaria		0,00
CA.03.06.01.01.01	Rimanenze finali materiale di consumo		0,00

DIPARTIMENTO/CENTRO
DI INGEGNERIA

Allegato n. 2

TABELLA DI RICLASSIFICAZIONE DELLE RETTIFICHE SU ANTICIPATE DI RIPORTO (VOCI COAN E UA) AL 31/12/2017

		UA	
Voce COAN	Denominazione	Rettifiche Anticipate di Riporto di RICAVO	Totale
CA.03.06.01.02.01	Rimanenze finali prodotti in corso di lavorazione		0,00
CA.03.06.01.03.01	Rimanenze finali prodotti finiti		0,00
CA.03.06.01.04.01	Rimanenze finali lavori in corso su ordinazione		0,00
CA.03.06.01.05.01	Rimanenze finali merci		0,00
CA.03.07.01.01.01	Storno di costi per incremento delle immobilizzazioni per lavori interni		0,00
CA.03.07.01.02.01	Rettifiche e rivalutazioni di immobilizzazioni immateriali		0,00
CA.03.07.01.03.01	Rettifiche e rivalutazioni di immobilizzazioni materiali		0,00
CA.03.09.10.01.01	Rimanenze finali di materiali		0,00
CA.03.13.01.01.01	Proventi finanziari da titoli e partecipazioni		0,00
CA.03.13.02.01.01	Interessi attivi		0,00
CA.03.13.03.01.01	Utili su cambi		0,00
CA.03.14.01.01.01	Rettifiche e rivalutazioni di attività finanziarie		0,00
CA.03.15.01.01.01	Proventi straordinari		0,00
	TOTALE DISPONIBILITA' AL 31/12/2017	0,00	0,00

DIPARTIMENTO
DI INGEGNERIA

Allegato n. 2

TABELLA DI RICLASSIFICAZIONE DELLE RETTIFICHE SU ANTICIPATE DI RIPORTO (VOCI COAN E UA) AL 31/12/2017

		UA		
Voce COAN	Denominazione	Somme da riapplicare	Economia	Totale
CA.01.10.01.01.01	Costi di impianto, di ampliamento e di sviluppo			0,00
CA.01.10.01.02.01	Diritti di brevetto			0,00
CA.01.10.01.02.02	Altri diritti di utilizzazione delle opere di ingegno			0,00
CA.01.10.01.03.01	Concessioni marchi e diritti similari			0,00
CA.01.10.01.03.02	Licenze d'uso			0,00
CA.01.10.01.04.01	Immobilizzazioni immateriali in corso e acconti			0,00
CA.01.10.01.05.01	Software			0,00
CA.01.10.01.05.02	Altre immobilizzazioni immateriali			0,00
CA.01.10.01.05.03	Interventi ed opere su beni di terzi			0,00
CA.01.10.02.01.01	Terreni			0,00
CA.01.10.02.01.02	Interventi edilizi su terreni			0,00
CA.01.10.02.01.03	Fabbricati			0,00
CA.01.10.02.01.04	Interventi edilizi su Fabbricati			0,00
CA.01.10.02.01.05	Manutenzione straordinaria su fabbricati			0,00
CA.01.10.02.02.01	Impianti generici			0,00
CA.01.10.02.02.02	Manutenzione straordinaria impianti generici			0,00
CA.01.10.02.02.03	Impianti per la ricerca scientifica			0,00
CA.01.10.02.02.04	Manutenzione straordinaria impianti per la ricerca scientifica			0,00
CA.01.10.02.02.05	Attrezzature			0,00
CA.01.10.02.03.01	Attrezzatura per la ricerca scientifica			0,00
CA.01.10.02.04.01	Beni di valore culturale, storico, archeologico ed artistico			0,00
CA.01.10.02.04.02	Interventi di restauro su beni di valore culturale, storico, archeologico ed artistico			0,00
CA.01.10.02.04.03	Materiale bibliografico			0,00

DIPARTIMENTO
DI INGEGNERIA

Allegato n. 2

TABELLA DI RICLASSIFICAZIONE DELLE RETTIFICHE SU ANTICIPATE DI RIPORTO (VOCI COAN E UA) AL 31/12/2017

		UA		
Voce COAN	Denominazione	Somme da riapplicare	Economia	Totale
CA.01.10.02.04.04	Opere artistiche			0,00
CA.01.10.02.04.05	collezioni scientifiche			0,00
CA.01.10.02.04.06	Altro materiale bibliografico			0,00
CA.01.10.02.05.01	Mobili e Arredi			0,00
CA.01.10.02.06.01	Costi e acconti per acquisizione di terreni			0,00
CA.01.10.02.06.02	Costi e acconti per interventi edilizi su terreni			0,00
CA.01.10.02.06.03	Costi e acconti per interventi edilizi su fabbricati			0,00
CA.01.10.02.06.04	Costi e acconti per manutenzione straordinaria su fabbricati			0,00
CA.01.10.02.06.05	Costi e acconti per acquisizione di fabbricati			0,00
CA.01.10.02.06.06	Costi e acconti per acquisizione di impianti generici			0,00
CA.01.10.02.06.07	Costi e acconti per acquisizione di impianti per la ricerca scientifica			0,00
CA.01.10.02.06.08	Costi e acconti per altre immobilizzazioni materiali			0,00
CA.01.10.02.07.01	Apparecchiature di natura informatica			0,00
CA.01.10.02.07.02	Autovetture di rappresentanza e di servizio			0,00
CA.01.10.02.07.03	Autocarri, mezzi agricoli e altri mezzi di trasporto			0,00
CA.01.10.02.07.04	Altri beni mobili			0,00
CA.01.10.03.01.01	Partecipazioni in imprese ed enti controllati			0,00
CA.01.10.03.01.02	Partecipazioni in altre imprese ed enti			0,00
CA.01.10.03.01.03	Altri titoli			0,00
CA.01.10.03.01.04	Partecipazione in imprese ed enti collegati			0,00
CA.01.11.01.01.01	F.do di riserva vincolato ad investimenti			0,00
CA.01.12.01.01.01	Trasferimenti interni budget investimenti			0,00
CA.08.80.01.01.01	Costi di investimento progetti - quota di competenza per finanziamenti competitivi da miur - progetti di ricerca di rilevante interesse nazionale			0,00

DIPARTIMENTO
DI INGEGNERIA

Allegato n. 2

TABELLA DI RICLASSIFICAZIONE DELLE RETTIFICHE SU ANTICIPATE DI RIPORTO (VOCI COAN E UA) AL 31/12/2017

		UA		
Voce COAN	Denominazione	Somme da riapplicare	Economia	Totale
CA.08.80.01.01.02	Costi di investimento progetti - quota di competenza per finanziamenti competitivi da miur - fondo per gli investimenti della ricerca di base (firb)			0,00
CA.08.80.01.01.03	" Costi di investimento progetti - quota di competenza per altri finanziamenti competitivi da miur"			0,00
CA.08.80.01.02.01	Costi di investimento progetti - quota di competenza per finanziamenti competitivi da altri ministeri per ricerca scientifica			0,00
CA.08.80.01.02.02	Costi di investimento progetti - quota di competenza per finanziamenti competitivi da stato (organi diversi da ministeri) per ricerca scientifica			0,00
CA.08.80.01.02.03	Costi di investimento progetti - quota di competenza per finanziamenti competitivi per ricerca da regioni e province autonome			0,00
CA.08.80.01.02.04	Costi di investimento progetti - quota di competenza per finanziamenti competitivi per ricerca da province			0,00
CA.08.80.01.02.05	Costi di investimento progetti - quota di competenza per finanziamenti competitivi per ricerca da città metropolitane			0,00
CA.08.80.01.02.06	Costi di investimento progetti - quota di competenza per finanziamenti competitivi per ricerca da comuni			0,00
CA.08.80.01.02.07	Costi di investimento progetti - quota di competenza per finanziamenti competitivi per ricerca da camere di commercio			0,00
CA.08.80.01.02.08	Costi di investimento progetti - quota di competenza per finanziamenti competitivi per ricerca da altre università			0,00
CA.08.80.01.02.09	Costi di investimento progetti - quota di competenza per finanziamenti competitivi per ricerca da altre amministrazioni pubbliche			0,00
CA.08.80.01.03.01	Costi di investimento progetti - quota di competenza per finanziamenti competitivi da cnr			0,00
CA.08.80.01.03.02	Costi di investimento progetti - quota di competenza per finanziamenti competitivi per ricerca da enti di ricerca diversi dal cnr			0,00
CA.08.80.01.04.01	Costi di investimento progetti - quota di competenza per finanziamenti competitivi per ricerca da parte dell'unione europea			0,00
CA.08.80.01.04.02	Costi di investimento progetti - quota di competenza per finanziamenti competitivi per ricerca da parte di organismi internazionali			0,00
CA.08.80.01.05.01	Costi di investimento progetti - attivita' in conto terzi e cessione di risultati di ricerca			0,00
CA.08.80.01.06.01	Costi di investimento progetti - finanziamenti non competitivi per la ricerca			0,00
CA.08.80.01.07.01	Costi di investimento progetti - Centri Autonomi di Gestione con Autonomia Negoziale			0,00
CA.10.10.01.01.01	Costruzione, ristrutturazione e restauro fabbricati			0,00
CA.10.10.01.01.02	Costruzione impianti			0,00
CA.10.10.01.01.03	Ricostruzione e trasformazione fabbricati			0,00
CA.10.10.01.01.04	Ricostruzione e trasformazione impianti			0,00
CA.10.10.01.01.05	Manutenzione straordinaria immobili			0,00

DIPARTIMENTO
DI INGEGNERIA

Allegato n. 2

TABELLA DI RICLASSIFICAZIONE DELLE RETTIFICHE SU ANTICIPATE DI RIPORTO (VOCI COAN E UA) AL 31/12/2017

		UA		
Voce COAN	Denominazione	Somme da riapplicare	Economia	Totale
CA.10.10.01.01.06	Manutenzione straordinaria impianti			0,00
CA.10.10.01.01.07	Spese in applicazione D.L. 626/94			0,00
CA.10.10.01.01.08	Manutenzione straordinaria immobili - Messa a norma e sicurezza - Spese in applicazione D.Lgs. 81/2008			0,00
CA.10.10.01.01.09	Informatizzazione Servizi - Budget investimenti			0,00
CA.10.10.01.01.10	Gestione e sviluppo Rete di Ateneo - Budget investimenti			0,00
CA.10.10.01.01.11	Mobilità e scambi culturali docenti - Budget investimenti			0,00
CA.10.10.01.01.12	Rapporti Internazionali, scambi culturali - Budget investimenti			0,00
CA.10.10.01.01.13	Comunicazione di Ateneo - Budget investimenti			0,00
CA.10.10.01.01.14	Acquisto, manutenzione, noleggio, esercizio veicoli - Budget investimenti			0,00
CA.10.10.01.01.15	Spese inerenti l'orientamento universitario - Budget investimenti			0,00
CA.10.10.01.01.16	Progetti III Missione - Budget investimenti			0,00
CA.10.10.01.01.17	Spese funzionamento Servizio Prevenzione e Protezione - Budget investimenti			0,00
CA.10.10.01.01.18	Funzionamento Strutture Didattiche finanziate da Esterni - Budget investimenti			0,00
CA.10.10.01.01.19	Ricerca di base - Budget investimenti			0,00
CA.10.10.01.01.20	Funzionamento strutture didattiche - Budget investimenti			0,00
CA.10.10.01.01.21	Costi operativi su economie progetti - Budget investimenti			0,00
CA.10.10.01.01.22	Costi operativi altri progetti Amministrazione centrale - Budget investimenti			0,00
CA.04.06.01.01.01	Rimanenze iniziali materiale di consumo			0,00
CA.04.06.01.02.01	Rimanenze iniziali prodotti in corso di lavorazione			0,00
CA.04.06.01.03.01	Rimanenze iniziali prodotti finiti			0,00
CA.04.06.01.04.01	Rimanenze iniziali lavori in corso su ordinazione			0,00
CA.04.06.01.05.01	Rimanenze iniziali merci			0,00
CA.04.08.01.01.01	Costo per competenze fisse del personale docente a tempo indeterminato			0,00

DIPARTIMENTO
DI INGEGNERIA

Allegato n. 2

TABELLA DI RICLASSIFICAZIONE DELLE RETTIFICHE SU ANTICIPATE DI RIPORTO (VOCI COAN E UA) AL 31/12/2017

		UA		
Voce COAN	Denominazione	Somme da riapplicare	Economia	Totale
CA.04.08.01.01.02	Costo per competenze fisse del personale docente a tempo determinato			0,00
CA.04.08.01.01.03	Costo per supplenze e affidamenti a personale docente a tempo indeterminato			0,00
CA.04.08.01.01.04	Costo per supplenze e affidamenti a personale docente a tempo determinato			0,00
CA.04.08.01.01.05	Indennita' di missione, rimborsi spese viaggi e iscrizione a convegni del personale docente e ricercatori			0,00
CA.04.08.01.01.06	Costo per competenze fisse del personale ricercatore a tempo indeterminato			0,00
CA.04.08.01.01.07	Costo per supplenze e affidamenti a personale ricercatore a tempo indeterminato			0,00
CA.04.08.01.01.08	Costo per competenze fisse del personale ricercatore a tempo determinato			0,00
CA.04.08.01.01.09	Costo per supplenze e affidamenti a personale ricercatore a tempo determinato			0,00
CA.04.08.01.01.10	Costo delle competenze accessorie del personale docente e ricercatore			0,00
CA.04.08.01.01.11	Indennità di rischio del personale docente e dei ricercatori			0,00
CA.04.08.01.01.12	Indennità di rischio radiologico del personale docente e dei ricercatori- non convenzionato			0,00
CA.04.08.01.01.13	Punti organico per personale docente e ricercatore			0,00
CA.04.08.01.01.14	Fondo di Ateneo per la premialità			0,00
CA.04.08.01.02.01	Assegni di ricerca			0,00
CA.04.08.01.02.02	Indennita' di missione, rimborsi spese viaggi per gli assegni di ricerca			0,00
CA.04.08.01.03.01	Costo del personale docente a contratto			0,00
CA.04.08.01.04.01	Costo per i collaboratori ed esperti linguistici a tempo indeterminato			0,00
CA.04.08.01.04.02	Competenze fisse a collaboratori ed esperti linguistici di madre lingua a tempo determinato (td)			0,00
CA.04.08.01.04.03	Costo per supplenze e affidamenti a collaboratori ed esperti linguistici a tempo indeterminato			0,00
CA.04.08.01.04.04	Costo per supplenze e affidamenti a collaboratori ed esperti linguistici a tempo determinato			0,00
CA.04.08.01.04.05	Indennita' di missione, rimborsi spese viaggi per collaboratori ed esperti linguistici a tempo indeterminato			0,00
CA.04.08.01.04.06	Indennita' di missione, rimborsi spese viaggi per collaboratori ed esperti linguistici a tempo determinato			0,00
CA.04.08.01.04.07	Costi di formazione esperti linguistici			0,00

Allegato n. 2

TABELLA DI RICLASSIFICAZIONE DELLE RETTIFICHE SU ANTICIPATE DI RIPORTO (VOCI COAN E UA) AL 31/12/2017

		UA		
Voce COAN	Denominazione	Somme da riapplicare	Economia	Totale
CA.04.08.01.05.01	Costo per competenze fisse per altro personale dedicato alla ricerca ed alla didattica			0,00
CA.04.08.01.05.02	Competenze accessorie per altro personale dedicato alla ricerca ed alla didattica			0,00
CA.04.08.01.06.01	Compensi a personale docente convenzionato ssn (per attività assistenziale)			0,00
CA.04.08.01.06.02	Compensi a personale ricercatore a tempo indeterminato convenzionato ssn (per attività assistenziale)			0,00
CA.04.08.01.06.03	Compensi a personale ricercatore a tempo determinato convenzionato ssn (per attività assistenziale)			0,00
CA.04.08.01.07.01	Costo delle competenze per personale docente e ricercatore su attivita' conto terzi			0,00
CA.04.08.02.01.01	Costo dei dirigenti a tempo indeterminato			0,00
CA.04.08.02.02.01	Costo del direttore generale e dei dirigenti a tempo determinato			0,00
CA.04.08.02.03.01	Costo del personale tecnico-amministrativo a tempo indeterminato			0,00
CA.04.08.02.04.01	Costo del personale tecnico-amministrativo a tempo determinato			0,00
CA.04.08.02.05.01	Competenze accessorie del personale dirigente			0,00
CA.04.08.02.05.02	Competenze accessorie al personale EP			0,00
CA.04.08.02.05.03	Competenze accessorie al personale tecnico-amministrativo			0,00
CA.04.08.02.05.04	Indennità centralinisti non vedenti			0,00
CA.04.08.02.05.05	Indennità di rischio radiologico del personale tecnico-amministrativo a tempo indeterminato - non convenzionato			0,00
CA.04.08.02.06.01	Indennita' di missione, rimborsi spese viaggi del personale dirigente e tecnico-amministrativo			0,00
CA.04.08.02.06.02	Buoni pasto per il personale tecnico-amministrativo			0,00
CA.04.08.02.06.03	Formazione del personale dirigente e tecnico-amministrativo			0,00
CA.04.08.02.06.04	Punti organico per personale dirigente, tecnico-amministrativo e cel			0,00
CA.04.08.02.07.01	Compesi attività conto terzi personale tecnico amministrativo			0,00
CA.04.08.02.08.01	Compensi a personale tecnico-amministrativo a tempo indeterminato convenzionato ssn (per attività assistenziale)			0,00
CA.04.08.02.08.02	Compensi a personale tecnico-amministrativo a tempo determinato convenzionato ssn (per attività assistenziale)			0,00
CA.04.08.02.09.01	Compenso a personale tecnico amministrativo ai sensi del Codice dei contratti			0,00

DIPARTIMENTO
DI INGEGNERIA

Allegato n. 2

TABELLA DI RICLASSIFICAZIONE DELLE RETTIFICHE SU ANTICIPATE DI RIPORTO (VOCI COAN E UA) AL 31/12/2017

		UA		
Voce COAN	Denominazione	Somme da riapplicare	Economia	Totale
CA.04.09.01.01.01	Costi per borse di studio per scuole di specializzazione mediche a norma ue			0,00
CA.04.09.01.01.02	Costi per borse di studio per scuole di specializzazione			0,00
CA.04.09.01.01.03	Costi per borse di studio per dottorato di ricerca			0,00
CA.04.09.01.01.04	Borse di studio per post dottorato			0,00
CA.04.09.01.01.05	Costi per altre borse			0,00
CA.04.09.01.01.06	Indennita' di missione, rimborsi spese viaggi per borse di studio per scuole di specializzazione mediche a norma ue			0,00
CA.04.09.01.01.07	Indennita' di missione, rimborsi spese viaggi per borse di studio per scuole di specializzazione			0,00
CA.04.09.01.01.08	Indennita' di missione, rimborsi spese viaggi per borse di studio per post dottorato			0,00
CA.04.09.01.01.09	Indennita' di missione, rimborsi spese viaggii per altre borse			0,00
CA.04.09.01.01.10	Indennita' di missione, rimborsi spese viaggi per dottorato di ricerca			0,00
CA.04.09.01.02.01	Programmi di mobilita' e scambi culturali studenti			0,00
CA.04.09.01.02.02	Iniziative ed attivita' culturali gestite dagli studenti			0,00
CA.04.09.01.02.03	Interventi a favore degli studenti diversamente abili			0,00
CA.04.09.01.02.04	Assegni per l'incentivazione dell'attivita' di tutorato			0,00
CA.04.09.01.02.05	Altri interventi a favore degli studenti			0,00
CA.04.09.01.02.06	Altri premi			0,00
CA.04.09.01.03.01	Convegni e seminari			0,00
CA.04.09.01.03.02	Ospitalita' visiting professor, esperti e relatori convegni			0,00
CA.04.09.01.03.03	Compensi e soggiorno a visiting professor, esperti e relatori convegni			0,00
CA.04.09.02.01.01	Borse di collaborazione studenti, attivita' a tempo parziale art. 11 D.Lgs 29/03/2012 n° 68			0,00
CA.04.09.03.01.01	Costi per la ricerca e l'attività editoriale			0,00
CA.04.09.04.01.01	Trasferimenti a patner di progetti coordinati			0,00
CA.04.09.05.01.01	Materiale di consumo per laboratorio			0,00

Allegato n. 2

TABELLA DI RICLASSIFICAZIONE DELLE RETTIFICHE SU ANTICIPATE DI RIPORTO (VOCI COAN E UA) AL 31/12/2017

		UA		
Voce COAN	Denominazione	Somme da riapplicare	Economia	Totale
CA.04.09.06.01.01	Rimanenze iniziali materiale di consumo per laboratorio			0,00
CA.04.09.06.02.01	Rimanenze finali materiale di consumo per laboratori			0,00
CA.04.09.07.01.01	Riviste e giornali			0,00
CA.04.09.07.01.02	Libri e altro materiale bibliografico non costituenti immobilizzazioni			0,00
CA.04.09.08.01.01	Utenze e canoni per energia elettrica			0,00
CA.04.09.08.02.01	Utenze e canoni per telefonia fissa			0,00
CA.04.09.08.02.02	Utenze e canoni per telefonia mobile			0,00
CA.04.09.08.02.03	Utenze e canoni per reti di trasmissione			0,00
CA.04.09.08.03.01	Utenze e canoni per acqua			0,00
CA.04.09.08.03.02	Utenze e canoni per gas			0,00
CA.04.09.08.03.03	Riscaldamento e condizionamento			0,00
CA.04.09.08.03.04	Altre utenze e canoni			0,00
CA.04.09.08.04.01	Pulizia			0,00
CA.04.09.08.04.02	Smaltimento rifiuti nocivi			0,00
CA.04.09.08.04.03	Traslochi e facchinaggio			0,00
CA.04.09.08.04.04	Vigilanza e altri servizi ausiliari			0,00
CA.04.09.08.05.01	Manutenzione ordinaria e riparazioni di immobili			0,00
CA.04.09.08.05.02	Manutenzione ordinaria e riparazioni di impianti			0,00
CA.04.09.08.05.03	Manutenzione ordinaria e riparazioni di apparecchiature			0,00
CA.04.09.08.05.04	Manutenzione ordinaria e riparazioni di autovetture di rappresentanza e di servizio			0,00
CA.04.09.08.05.05	Manutenzione ordinaria e riparazioni di autocarri, mezzi agricoli e altri mezzi di trasporto			0,00
CA.04.09.08.05.06	Manutenzione ordinaria e riparazioni mobili e arredi			0,00
CA.04.09.08.05.07	Altre spese di manutenzione ordinaria e riparazioni			0,00

Allegato n. 2

TABELLA DI RICLASSIFICAZIONE DELLE RETTIFICHE SU ANTICIPATE DI RIPORTO (VOCI COAN E UA) AL 31/12/2017

		UA		
Voce COAN	Denominazione	Somme da riapplicare	Economia	Totale
CA.04.09.08.06.01	Rappresentanza			0,00
CA.04.09.08.06.02	Organizzazione manifestazioni e convegni			0,00
CA.04.09.08.06.03	Spese postali			0,00
CA.04.09.08.06.04	Assicurazioni			0,00
CA.04.09.08.06.05	Spese per le pubblicazioni dell'ateneo			0,00
CA.04.09.08.06.06	Spesa corrente per brevetti			0,00
CA.04.09.08.06.07	Altre spese per servizi			0,00
CA.04.09.08.06.08	Costi annuali per pubblicita'			0,00
CA.04.09.08.06.09	Spese per pubblicita' degli atti			0,00
CA.04.09.08.06.10	Spese per lavorazioni agricole effettuate da terzi			0,00
CA.04.09.08.06.11	Spese per commissioni ed intermediazioni bancarie			0,00
CA.04.09.08.07.01	Consulenze tecnico-scientifiche			0,00
CA.04.09.08.07.02	Consulenze tecnico-amministrative			0,00
CA.04.09.08.07.03	Spese per liti (patrocinio legale)			0,00
CA.04.09.08.07.04	Spese notarili			0,00
CA.04.09.08.08.01	Prestazioni di lavoro autonomo			0,00
CA.04.09.08.09.01	Prestazioni di servizi tecnico/amministrativi da enti terzi			0,00
CA.04.09.08.09.02	Altre prestazioni e servizi da terzi			0,00
CA.04.09.08.10.01	Collaborazioni coordinate e continuative			0,00
CA.04.09.08.11.01	Costi per fornitura di lavoro interinale			0,00
CA.04.09.09.01.01	Carburanti, combustibili e lubrificanti per autovetture			0,00
CA.04.09.09.01.02	Carburanti, combustibili e lubrificanti per autocarri, mezzi agricoli e altri mezzi di trasporto			0,00
CA.04.09.09.01.03	Cancelleria e altro materiale di consumo		123,70	123,70

Allegato n. 2

TABELLA DI RICLASSIFICAZIONE DELLE RETTIFICHE SU ANTICIPATE DI RIPORTO (VOCI COAN E UA) AL 31/12/2017

		UA		
Voce COAN	Denominazione	Somme da riapplicare	Economia	Totale
CA.04.09.09.01.04	Libretti e diplomi			0,00
CA.04.09.09.01.05	Vestiario			0,00
CA.04.09.09.01.06	Materiale per ricorrenze elettorali			0,00
CA.04.09.09.01.07	Altri materiali			0,00
CA.04.09.09.01.08	Combustibili per riscaldamento			0,00
CA.04.09.09.02.01	Acquisto beni strumentali			0,00
CA.04.09.09.02.02	Acquisto software per pc			0,00
CA.04.09.09.03.01	Sconti e abbuoni passivi			0,00
CA.04.09.09.03.02	Sconti abbuoni e premi su acquisti			0,00
CA.04.09.10.01.01	Rimanenze iniziali di materiali			0,00
CA.04.09.11.01.01	Leasing operativo ed altre forme di locazione di beni mobili			0,00
CA.04.09.11.01.02	Leasing operativo ed altre forme di locazione di autovetture di rappresentanza e di servizio			0,00
CA.04.09.11.01.03	Leasing operativo ed altre forme di locazione di autocarri, mezzi agricoli e altri mezzi di trasporto			0,00
CA.04.09.11.01.04	Noleggio fax e fotocopiatrici			0,00
CA.04.09.11.01.05	Altri noleggi			0,00
CA.04.09.11.02.01	Fitti passivi per locazione di edifici			0,00
CA.04.09.11.02.02	Altri fitti passivi			0,00
CA.04.09.11.03.01	Licenze software			0,00
CA.04.09.12.01.01	Missioni e rimborsi spese trasferta organi istituzionali			0,00
CA.04.09.12.01.02	Gettoni/indennita' ai membri degli organi istituzionali di governo e controllo			0,00
CA.04.09.12.01.03	Indennita' di carica			0,00
CA.04.09.12.01.04	Gettoni/indennita' ai membri degli organi istituzionali che non siano di amministrazione e controllo			0,00
CA.04.09.12.01.05	Gettoni/indennita' ai membri del collegio dei revisori			0,00

DIPARTIMENTO
DI INGEGNERIA

Allegato n. 2

TABELLA DI RICLASSIFICAZIONE DELLE RETTIFICHE SU ANTICIPATE DI RIPORTO (VOCI COAN E UA) AL 31/12/2017

		UA		
Voce COAN	Denominazione	Somme da riapplicare	Economia	Totale
CA.04.09.12.01.06	Gettoni/indennita' ai membri del nucleo di valutazione			0,00
CA.04.09.12.01.07	Garante di Ateneo			0,00
CA.04.09.12.02.01	Quote associative			0,00
CA.04.09.12.02.02	Compensi per commissioni di concorso del personale interno ed esterno			0,00
CA.04.09.12.02.03	Altri costi per attivita' istituzionali			0,00
CA.04.09.12.02.04	Costi per attivita' sportive l. 394/77			0,00
CA.04.09.12.02.05	Cus - attivita' sportiva e gestione impianti sportivi			0,00
CA.04.09.12.02.06	Arrotondamenti negativi			0,00
CA.04.09.12.02.07	Visite medico-fiscali			0,00
CA.04.09.12.02.08	Accertamenti sanitari			0,00
CA.04.09.12.02.09	Equo indennizzo			0,00
CA.04.09.12.02.10	Provvidenze a favore del personale			0,00
CA.04.09.12.02.11	Circolo San Martino - attività sociali del personale			0,00
CA.04.09.12.02.12	Prestazioni INAIL - gestione per conto			0,00
CA.04.09.12.02.13	Spese condominiali			0,00
CA.04.10.01.01.01	QUOTE DI AMMORTAMENTO COSTI DI IMPIANTO, DI AMPLIAMENTO E DI SVILUPPO			0,00
CA.04.10.01.01.02	QUOTE DI AMMORTAMENTO DIRITTI DI BREVETTO E DIRITTI DI UTILIZZAZIONE DELLE OPERE DI INGEGNO			0,00
CA.04.10.01.01.03	QUOTE DI AMMORTAMENTO CONCESSIONI, LICENZE, MARCHI E DIRITTI SIMILI			0,00
CA.04.10.01.01.04	QUOTE DI AMMORTAMENTO ALTRE IMMOBILIZZAZIONI IMMATERIALI			0,00
CA.04.10.02.01.01	QUOTE DI AMMORTAMENTO TERRENI E FABBRICATI			0,00
CA.04.10.02.01.02	QUOTE DI AMMORTAMENTO IMPIANTI E ATTREZZATURE			0,00
CA.04.10.02.01.03	QUOTE DI AMMORTAMENTO ATTREZZATURE SCIENTIFICHE			0,00
CA.04.10.02.01.04	QUOTE DI AMMORTAMENTO PATRIMONIO LIBRARIO, OPERE D'ARTE, D'ANTIQUARIATO E MUSEALI			0,00

Allegato n. 2

TABELLA DI RICLASSIFICAZIONE DELLE RETTIFICHE SU ANTICIPATE DI RIPORTO (VOCI COAN E UA) AL 31/12/2017

		UA		
Voce COAN	Denominazione	Somme da riapplicare	Economia	Totale
CA.04.10.02.01.05	QUOTE DI AMMORTAMENTO MOBILI E ARREDI			0,00
CA.04.10.02.01.06	QUOTE DI AMMORTAMENTO ALTRE IMMOBILIZZAZIONI MATERIALI			0,00
CA.04.10.03.01.01	SVALUTAZIONE COSTI DI IMPIANTO, DI AMPLIAMENTO E DI SVILUPPO			0,00
CA.04.10.03.01.02	SVALUTAZIONE DIRITTI DI BREVETTO E DIRITTI DI UTILIZZAZIONE DELLE OPERE DI INGEGNO			0,00
CA.04.10.03.01.03	SVALUTAZIONE CONCESSIONI, LICENZE, MARCHI E DIRITTI SIMILI			0,00
CA.04.10.03.01.04	SVALUTAZIONE ALTRE IMMOBILIZZAZIONI IMMATERIALI			0,00
CA.04.10.03.01.05	SVALUTAZIONE TERRENI E FABBRICATI			0,00
CA.04.10.03.01.06	SVALUTAZIONE IMPIANTI E ATTREZZATURE			0,00
CA.04.10.03.01.07	SVALUTAZIONE ATTREZZATURE SCIENTIFICHE			0,00
CA.04.10.03.01.08	SVALUTAZIONE PATRIMONIO LIBRARIO, OPERE D'ARTE, D'ANTIQUARIATO E MUSEALI			0,00
CA.04.10.03.01.09	SVALUTAZIONE MOBILI E ARREDI			0,00
CA.04.10.03.01.10	SVALUTAZIONE ALTRE IMMOBILIZZAZIONI MATERIALI			0,00
CA.04.10.04.01.01	Perdita su crediti compresi nell'attivo circolante e nelle disponibilità liquide			0,00
CA.04.11.01.01.01	Quote di accantonamento al fondo svalutazione crediti			0,00
CA.04.11.01.01.02	QUOTE DI ACCANTONAMENTO AI FONDI ARRETRATI DA CORRISPONDERE AL PERSONALE NEGLI ESERCIZI FUTURI			0,00
CA.04.11.01.03.01	Quote di accantonamento ai fondi per cause e controversie in corso			0,00
CA.04.11.01.04.01	Quote di accantonamento ai fondi per altri rischi e oneri			0,00
CA.04.11.01.05.01	Quota di esercizio per altri accantonamenti			0,00
CA.04.11.01.06.01	Accantonamento per fondi di quiescenza			0,00
CA.04.11.01.06.02	Accantonamento per fondi tfr personale			0,00
CA.04.12.01.01.01	TRASFERIMENTI INTERNI CORRENTI			0,00
CA.04.12.01.01.02	TRASFERIMENTI INTERNI PER INVESTIMENTI			0,00
CA.04.12.01.01.03	TRASFERIMENTI INTERNI SU ATTIVITA' CONTO TERZI			0,00

DIPARTIMENTO
DI INGEGNERIA

Allegato n. 2

TABELLA DI RICLASSIFICAZIONE DELLE RETTIFICHE SU ANTICIPATE DI RIPORTO (VOCI COAN E UA) AL 31/12/2017

		UA		
Voce COAN	Denominazione	Somme da riapplicare	Economia	Totale
CA.04.12.01.01.04	TRASFERIMENTI INTERNI PER RESTITUZIONI E RIMBORSI			0,00
CA.04.12.01.01.05	ALTRI TRASFERIMENTI INTERNI			0,00
CA.04.12.01.01.06	TRASFERIMENTI VARI			0,00
CA.04.12.01.02.01	Versamenti allo Stato per riduzioni di spesa			0,00
CA.04.12.01.03.01	Tassa di rimozione rifiuti solidi urbani			0,00
CA.04.12.01.03.02	Imposte sul registro			0,00
CA.04.12.01.03.03	Valori bollati			0,00
CA.04.12.01.03.04	Altri tributi			0,00
CA.04.12.01.03.05	Iva indetrabile			0,00
CA.04.12.01.03.06	Imposte sul patrimonio			0,00
CA.04.12.01.03.07	Tributi su lasciti e donazioni			0,00
CA.04.13.02.01.01	Oneri finanziari			0,00
CA.04.13.02.02.01	Interessi passivi			0,00
CA.04.13.03.01.01	Perdite su cambi			0,00
CA.04.14.02.01.01	Svalutazione titoli e partecipazioni			0,00
CA.04.15.02.01.01	Oneri straodinari per il personale			0,00
CA.04.15.02.02.01	Rimborsi tasse e contributi agli studenti			0,00
CA.04.15.02.03.01	Oneri straordinari per recuperi e rimborsi			0,00
CA.04.15.02.04.01	Altri oneri straordinari			0,00
CA.04.15.02.05.01	Imposte relative ad esercizi precedenti			0,00
CA.04.16.01.01.01	Imposte sul reddito			0,00
CA.04.17.01.01.01	Voce per variazione tecnica			0,00
CA.07.70.01.01.01	Costi operativi progetti - quota di competenza per finanziamenti competitivi da miur - progetti di ricerca di rilevante interesse nazionale			0,00

DIPARTIMENTO
DI INGEGNERIA

Allegato n. 2

TABELLA DI RICLASSIFICAZIONE DELLE RETTIFICHE SU ANTICIPATE DI RIPORTO (VOCI COAN E UA) AL 31/12/2017

		UA		
Voce COAN	Denominazione	Somme da riapplicare	Economia	Totale
CA.07.70.01.01.02	Costi operativi progetti - quota di competenza per finanziamenti competitivi da miur - fondo per gli investimenti della ricerca di base (firb)			0,00
CA.07.70.01.01.03	Costi operativi progetti - quota di competenza per altri finanziamenti competitivi da miur			0,00
CA.07.70.01.02.01	Costi operativi progetti - quota di competenza per finanziamenti competitivi da altri ministeri per ricerca scientifica			0,00
CA.07.70.01.02.02	Costi operativi progetti - quota di competenza per finanziamenti competitivi da stato (organi diversi da ministeri) per ricerca scientifica			0,00
CA.07.70.01.02.03	Costi operativi progetti - quota di competenza per finanziamenti competitivi per ricerca da regioni e province autonome			0,00
CA.07.70.01.02.04	Costi operativi progetti - quota di competenza per finanziamenti competitivi per ricerca da province			0,00
CA.07.70.01.02.05	Costi operativi progetti - quota di competenza per finanziamenti competitivi per ricerca da città metropolitane			0,00
CA.07.70.01.02.06	Costi operativi progetti - quota di competenza per finanziamenti competitivi per ricerca da comuni			0,00
CA.07.70.01.02.07	Costi operativi progetti - quota di competenza per finanziamenti competitivi per ricerca da camere di commercio			0,00
CA.07.70.01.02.08	Costi operativi progetti - quota di competenza per finanziamenti competitivi per ricerca da altre università			0,00
CA.07.70.01.02.09	Costi operativi progetti - quota di competenza per finanziamenti competitivi per ricerca da altre amministrazioni pubbliche			0,00
CA.07.70.01.03.01	Costi operativi progetti - quota di competenza per finanziamenti competitivi da cnr			0,00
CA.07.70.01.03.02	Costi operativi progetti - quota di competenza per finanziamenti competitivi per ricerca da enti di ricerca diversi dal cnr			0,00
CA.07.70.01.04.01	Costi operativi progetti - quota di competenza per finanziamenti competitivi per ricerca da parte dell'unione europea			0,00
CA.07.70.01.04.02	Costi operativi progetti - quota di competenza per finanziamenti competitivi per ricerca da parte di organismi internazionali			0,00
CA.07.70.01.05.01	Costi operativi progetti - attivita' c/terzi e cessione di risultati di ricerca			0,00
CA.07.70.01.06.01	Costi operativi progetti - finanziamenti non competitivi per la ricerca			0,00
CA.07.70.01.07.01	Costi operativi progetti - Centri Autonomi di Gestione con Autonomia Negoziale			0,00
CA.07.70.01.08.01	Costi operativi progetti per attività di formazione			0,00
CA.09.90.01.01.01	Mobilità e scambi culturali docenti - Budget economico			0,00
CA.09.90.01.01.02	Rapporti Internazionali, scambi culturali - Budget economico			0,00
CA.09.90.01.01.03	Comunicazione di Ateneo - Budget economico			0,00
CA.09.90.01.01.04	Acquisto, manutenzione, noleggio, esercizio veicoli - Budget economico			0,00

DIPARTIMENTO
DI INGEGNERIA

Allegato n. 2

TABELLA DI RICLASSIFICAZIONE DELLE RETTIFICHE SU ANTICIPATE DI RIPORTO (VOCI COAN E UA) AL 31/12/2017

		UA		
Voce COAN	**Denominazione**	**Somme da riapplicare**	**Economia**	**Totale**
CA.09.90.01.01.05	Spese inerenti l'orientamento universitario - Budget economico			0,00
CA.09.90.01.01.06	Progetti III Missione - Budget economico			0,00
CA.09.90.01.01.07	Spese funzionamento Servizio Prevenzione e Protezione - Budget economico			0,00
CA.09.90.01.01.08	Funzionamento Strutture Didattiche finanziate da Esterni - Budget economico			0,00
CA.09.90.01.01.09	Ricerca di base - Budget economico			0,00
CA.09.90.01.01.10	Funzionamento strutture didattiche - Budget economico			0,00
CA.09.90.01.01.11	Costi operativi su economie progetti - Budget economico			0,00
CA.09.90.01.01.12	Costi operativi altri progetti Amministrazione centrale - Budget economico			0,00
CA.09.90.01.01.13	Informatizzazione Servizi - Budget economico			0,00
CA.09.90.01.01.14	Gestione e sviluppo Rete di Ateneo - Budget economico			0,00
	TOTALE DISPONIBILITA' AL 31/12/2017	0,00	123,70	123,70

IL SEGRETARIO AMM.VO
(Sig. Giovanni Magara)

UNIVERSITÀ DEGLI STUDI DI PERUGIA
DIPARTIMENTO DI INGEGNERIA

IL DIRETTORE
(Prof. Giuseppe Saccomandi)

RETTIFICA ANTICIPATA DI RIPORTO VOCE COAN E UA - ANNO 2017

(importi da riclassificare nell'allegato n. 2)

Esercizio	Numero scrittura	Data Scrittura	Descrizione Causale Variazione	Descrizione Det	Ammontare	UA	Voce COAN	Denominazione Voce coan
2017	10932	26/04/2017	Rettifica Indiretta di Scrittura Anticipata di Riporto		24,40	UA.PG.DING	CA.04.09.09.01.03.01	Cancelleria e altro materiale di consumo
2017	10932	26/04/2017	Rettifica Indiretta di Scrittura Anticipata di Riporto		48,80	UA.PG.DING	CA.04.09.09.01.03.01	Cancelleria e altro materiale di consumo
2017	10932	26/04/2017	Rettifica Indiretta di Scrittura Anticipata di Riporto		24,40	UA.PG.DING	CA.04.09.09.01.03.01	Cancelleria e altro materiale di consumo
2017	10932	26/04/2017	Rettifica Indiretta di Scrittura Anticipata di Riporto		24,40	UA.PG.DING	CA.04.09.09.01.03.01	Cancelleria e altro materiale di consumo
2017	8847	06/03/2017	Rettifica Diretta di Scrittura Anticipata di Riporto	Koinor - sgabello con ruote DF-T7 nero in plastica	1,70	UA.PG.DING	CA.04.09.09.01.03.01	Cancelleria e altro materiale di consumo
					123,70	UA.PG.DING Totale		

IL SEGRETARIO AMM.VO
(Sig. Giovanni Magdra)

Allegato N. 5 al punto dell'ordine del giorno N. 3

Allegato n. 3

DIPARTIMENTO

DI INGEGNERIA

PROPOSTA DI RIAPPLICAZIONE ALL'ESERCIZIO 2018 DELLE DISPONIBILITA' LIBERE RISULTANTI AL 31/12/2017

DIPARTIMENTO
DI INGEGNERIA

Allegato n. 3

PROPOSTA DI RIAPPLICAZIONE ALL'ESERCIZIO 2018 DELLE DISPONIBILITA' LIBERE RISULTANTI AL 31/12/2017

Voce COAN	Denominazione	Totale
CA.01.10.01.01.01	Costi di impianto, di ampliamento e di sviluppo	
CA.01.10.01.02.01	Diritti di brevetto	
CA.01.10.01.02.02	Altri diritti di utilizzazione delle opere di ingegno	
CA.01.10.01.03.01	Concessioni marchi e diritti similari	
CA.01.10.01.03.02	Licenze d'uso	
CA.01.10.01.04.01	Immobilizzazioni immateriali in corso e acconti	
CA.01.10.01.05.01	Software	
CA.01.10.01.05.02	Altre immobilizzazioni immateriali	
CA.01.10.01.05.03	Interventi ed opere su beni di terzi	
CA.01.10.02.01.01	Terreni	
CA.01.10.02.01.02	Interventi edilizi su terreni	
CA.01.10.02.01.03	Fabbricati	
CA.01.10.02.01.04	Interventi edilizi su Fabbricati	
CA.01.10.02.01.05	Manutenzione straordinaria su fabbricati	
CA.01.10.02.02.01	Impianti generici	
CA.01.10.02.02.02	Manutenzione straordinaria impianti generici	
CA.01.10.02.02.03	Impianti per la ricerca scientifica	
CA.01.10.02.02.04	Manutenzione straordinaria impianti per la ricerca scientifica	
CA.01.10.02.02.05	Attrezzature	
CA.01.10.02.03.01	Attrezzatura per la ricerca scientifica	
CA.01.10.02.04.01	Beni di valore culturale, storico, archeologico ed artistico	
CA.01.10.02.04.02	Interventi di restauro su beni di valore culturale, storico, archeologico ed artistico	
CA.01.10.02.04.03	Materiale bibliografico	
CA.01.10.02.04.04	Opere artistiche	
CA.01.10.02.04.05	Collezioni scientifiche	
CA.01.10.02.04.06	Altro materiale bibliografico	
CA.01.10.02.05.01	Mobili e Arredi	
CA.01.10.02.06.01	Costi e acconti per acquisizione di terreni	
CA.01.10.02.06.02	Costi e acconti per interventi edilizi su terreni	
CA.01.10.02.06.03	Costi e acconti per interventi edilizi su fabbricati	
CA.01.10.02.06.04	Costi e acconti per manutenzione straordinaria su fabbricati	
CA.01.10.02.06.05	Costi e acconti per acquisizione di fabbricati	
CA.01.10.02.06.06	Costi e acconti per acquisizione di impianti generici	
CA.01.10.02.06.07	Costi e acconti per acquisizione di impianti per la ricerca scientifica	
CA.01.10.02.06.08	Costi e acconti per altre immobilizzazioni materiali	
CA.01.10.02.07.01	Apparecchiature di natura informatica	

Allegato n. 3

PROPOSTA DI RIAPPLICAZIONE ALL'ESERCIZIO 2018 DELLE DISPONIBILITA' LIBERE RISULTANTI AL 31/12/2017

Voce COAN	Denominazione	Totale
CA.01.10.02.07.02	Autovetture di rappresentanza e di servizio	
CA.01.10.02.07.03	Autocarri, mezzi agricoli e altri mezzi di trasporto	
CA.01.10.02.07.04	Altri beni mobili	
CA.01.10.03.01.01	Partecipazioni in imprese ed enti controllati	
CA.01.10.03.01.02	Partecipazioni in altre imprese ed enti	
CA.01.10.03.01.03	Altri titoli	
CA.01.10.03.01.04	Partecipazione in imprese ed enti collegati	
CA.01.11.01.01.01	F.do di riserva vincolato ad investimenti	
CA.01.12.01.01.01	Trasferimenti interni budget investimenti	
CA.04.06.01.01.01	Rimanenze iniziali materiale di consumo	
CA.04.06.01.02.01	Rimanenze iniziali prodotti in corso di lavorazione	
CA.04.06.01.03.01	Rimanenze iniziali prodotti finiti	
CA.04.06.01.04.01	Rimanenze iniziali lavori in corso su ordinazione	
CA.04.06.01.05.01	Rimanenze iniziali merci	
CA.04.08.01.01.01	Costo per competenze fisse del personale docente a tempo indeterminato	
CA.04.08.01.01.02	Costo per competenze fisse del personale docente a tempo determinato	
CA.04.08.01.01.03	Costo per supplenze e affidamenti a personale docente a tempo indeterminato	
CA.04.08.01.01.04	Costo per supplenze e affidamenti a personale docente a tempo determinato	
CA.04.08.01.01.05	Indennita' di missione, rimborsi spese viaggi e iscrizione a convegni del personale docente e ricercatori	
CA.04.08.01.01.06	Costo per competenze fisse del personale ricercatore a tempo indeterminato	
CA.04.08.01.01.07	Costo per supplenze e affidamenti a personale ricercatore a tempo indeterminato	
CA.04.08.01.01.08	Costo per competenze fisse del personale ricercatore a tempo determinato	
CA.04.08.01.01.09	Costo per supplenze e affidamenti a personale ricercatore a tempo determinato	
CA.04.08.01.01.10	Costo delle competenze accessorie del personale docente e ricercatore	
CA.04.08.01.01.11	Indennità di rischio del personale docente e dei ricercatori	
CA.04.08.01.01.12	Indennità di rischio radiologico del personale docente e dei ricercatori- non convenzionato	
CA.04.08.01.01.13	Punti organico per personale docente e ricercatore	
CA.04.08.01.01.14	Fondo di Ateneo per la premialità	
CA.04.08.01.02.01	Assegni di ricerca	
CA.04.08.01.02.02	Indennita' di missione, rimborsi spese viaggi per gli assegni di ricerca	
CA.04.08.01.03.01	Costo del personale docente a contratto	
CA.04.08.01.04.01	Costo per i collaboratori ed esperti linguistici a tempo indeterminato	
CA.04.08.01.04.02	Competenze fisse a collaboratori ed esperti linguistici di madre lingua a tempo determinato (td)	
CA.04.08.01.04.03	Costo per supplenze e affidamenti a collaboratori ed esperti linguistici a tempo indeterminato	
CA.04.08.01.04.04	Costo per supplenze e affidamenti a collaboratori ed esperti linguistici a tempo determinato	
CA.04.08.01.04.05	Indennita' di missione, rimborsi spese viaggi per collaboratori ed esperti linguistici a tempo indeterminato	

Allegato n. 3

PROPOSTA DI RIAPPLICAZIONE ALL'ESERCIZIO 2018 DELLE DISPONIBILITA' LIBERE RISULTANTI AL 31/12/2017

Voce COAN	Denominazione	Totale
CA.04.08.01.04.06	Indennita' di missione, rimborsi spese viaggi per collaboratori ed esperti linguistici a tempo determinato	
CA.04.08.01.04.07	Costi di formazione esperti linguistici	
CA.04.08.01.05.01	Costo per competenze fisse per altro personale dedicato alla ricerca ed alla didattica	
CA.04.08.01.05.02	Competenze accessorie per altro personale dedicato alla ricerca ed alla didattica	
CA.04.08.01.06.01	Compensi a personale docente convenzionato ssn (per attività assistenziale)	
CA.04.08.01.06.02	Compensi a personale ricercatore a tempo indeterminato convenzionato ssn (per attività assistenziale)	
CA.04.08.01.06.03	Compensi a personale ricercatore a tempo determinato convenzionato ssn (per attività assistenziale)	
CA.04.08.01.07.01	Costo delle competenze per personale docente e ricercatore su attivita' conto terzi	
CA.04.08.02.01.01	Costo dei dirigenti a tempo indeterminato	
CA.04.08.02.02.01	Costo del direttore generale e dei dirigenti a tempo determinato	
CA.04.08.02.03.01	Costo del personale tecnico-amministrativo a tempo indeterminato	
CA.04.08.02.04.01	Costo del personale tecnico-amministrativo a tempo determinato	
CA.04.08.02.05.01	Competenze accessorie del Direttore Generale e del personale dirigente	
CA.04.08.02.05.02	Competenze accessorie al personale EP	
CA.04.08.02.05.03	Competenze accessorie al personale tecnico-amministrativo	
CA.04.08.02.05.04	Indennità centralinisti non vedenti	
CA.04.08.02.05.05	Indennità di rischio radiologico del personale tecnico-amministrativo a tempo indeterminato - non convenzionato	
CA.04.08.02.06.01	Indennita' di missione, rimborsi spese viaggi del personale dirigente e tecnico-amministrativo	
CA.04.08.02.06.02	Buoni pasto per il personale tecnico-amministrativo	
CA.04.08.02.06.03	Formazione del personale dirigente e tecnico-amministrativo	
CA.04.08.02.06.04	Punti organico per personale dirigente, tecnico-amministrativo e cel	
CA.04.08.02.07.01	Compesi attività conto terzi personale tecnico amministrativo	
CA.04.08.02.08.01	Compensi a personale tecnico-amministrativo a tempo indeterminato convenzionato ssn (per attività assistenziale)	
CA.04.08.02.08.02	Compensi a personale tecnico-amministrativo a tempo determinato convenzionato ssn (per attività assistenziale)	
CA.04.08.02.09.01	Compenso a personale tecnico amministrativo ai sensi del Codice dei contratti	
CA.04.09.01.01.01	Costi per borse di studio per scuole di specializzazione mediche a norma ue	
CA.04.09.01.01.02	Costi per borse di studio per scuole di specializzazione	
CA.04.09.01.01.03	Costi per borse di studio per dottorato di ricerca	
CA.04.09.01.01.04	Borse di studio per post dottorato	
CA.04.09.01.01.05	Costi per altre borse	
CA.04.09.01.01.06	Indennita' di missione, rimborsi spese viaggi per borse di studio per scuole di specializzazione mediche a norma ue	
CA.04.09.01.01.07	Indennita' di missione, rimborsi spese viaggi per borse di studio per scuole di specializzazione	
CA.04.09.01.01.08	Indennita' di missione, rimborsi spese viaggi per borse di studio per post dottorato	
CA.04.09.01.01.09	Indennita' di missione, rimborsi spese viaggii per altre borse	
CA.04.09.01.01.10	Indennita' di missione, rimborsi spese viaggi per dottorato di ricerca	
CA.04.09.01.01.11	Borse di collaborazione studenti, attivita' a tempo parziale art. 11 D.Lgs 29/03/2012 n° 68	

Allegato n. 3

PROPOSTA DI RIAPPLICAZIONE ALL'ESERCIZIO 2018 DELLE DISPONIBILITA' LIBERE RISULTANTI AL 31/12/2017

Voce COAN	Denominazione	Totale
CA.04.09.01.02.01	Programmi di mobilita' e scambi culturali studenti	
CA.04.09.01.02.02	Iniziative ed attivita' culturali gestite dagli studenti	
CA.04.09.01.02.03	Interventi a favore degli studenti diversamente abili	
CA.04.09.01.02.04	Assegni per l'incentivazione dell'attivita' di tutorato	
CA.04.09.01.02.05	Altri interventi a favore degli studenti	
CA.04.09.01.02.06	Altri premi	
CA.04.09.01.03.01	Convegni e seminari	
CA.04.09.01.03.02	Ospitalita' visiting professor, esperti e relatori convegni	
CA.04.09.01.03.03	Compensi e soggiorno a visiting professor, esperti e relatori convegni	
CA.04.09.02.01.01	Borse di collaborazione studenti, attivita' a tempo parziale art. 11 D.Lgs 29/03/2012 n° 68 (Da non Utilizzare)	
CA.04.09.03.01.01	Costi per la ricerca e l'attività editoriale	
CA.04.09.04.01.01	Trasferimenti a partner di progetti coordinati	
CA.04.09.05.01.01	Materiale di consumo per laboratori	
CA.04.09.06.01.01	Rimanenze iniziali materiale di consumo per laboratori	
CA.04.09.06.02.01	Rimanenze finali materiale di consumo per laboratori	
CA.04.09.07.01.01	Riviste e giornali	
CA.04.09.07.01.02	Libri e altro materiale bibliografico non costituenti immobilizzazioni	
CA.04.09.08.01.01	Utenze e canoni per energia elettrica	
CA.04.09.08.02.01	Utenze e canoni per telefonia fissa	
CA.04.09.08.02.02	Utenze e canoni per telefonia mobile	
CA.04.09.08.02.03	Utenze e canoni per reti di trasmissione	
CA.04.09.08.03.01	Utenze e canoni per acqua	
CA.04.09.08.03.02	Utenze e canoni per gas	
CA.04.09.08.03.03	Riscaldamento e condizionamento	
CA.04.09.08.03.04	Altre utenze e canoni	
CA.04.09.08.04.01	Pulizia	
CA.04.09.08.04.02	Smaltimento rifiuti nocivi	2.907,95
CA.04.09.08.04.03	Traslochi e facchinaggio	
CA.04.09.08.04.04	Vigilanza e altri servizi ausiliari	
CA.04.09.08.05.01	Manutenzione ordinaria e riparazioni di immobili	
CA.04.09.08.05.02	Manutenzione ordinaria e riparazioni di impianti	
CA.04.09.08.05.03	Manutenzione ordinaria e riparazioni di apparecchiature	
CA.04.09.08.05.04	Manutenzione ordinaria e riparazioni di autovetture di rappresentanza e di servizio	
CA.04.09.08.05.05	Manutenzione ordinaria e riparazioni di autocarri, mezzi agricoli e altri mezzi di trasporto	
CA.04.09.08.05.06	Manutenzione ordinaria e riparazioni mobili e arredi	
CA.04.09.08.05.07	Altre spese di manutenzione ordinaria e riparazioni	

Allegato n. 3

PROPOSTA DI RIAPPLICAZIONE ALL'ESERCIZIO 2018 DELLE DISPONIBILITA' LIBERE RISULTANTI AL 31/12/2017

Voce COAN	Denominazione	Totale
CA.04.09.08.06.01	Rappresentanza	
CA.04.09.08.06.02	Organizzazione manifestazioni e convegni	
CA.04.09.08.06.03	Spese postali	
CA.04.09.08.06.04	Assicurazioni	
CA.04.09.08.06.05	Spese per le pubblicazioni dell'ateneo	
CA.04.09.08.06.06	Spesa corrente per brevetti	
CA.04.09.08.06.07	Altre spese per servizi	
CA.04.09.08.06.08	Costi annuali per pubblicita'	
CA.04.09.08.06.09	Spese per pubblicita' degli atti	
CA.04.09.08.06.10	Spese per lavorazioni agricole effettuate da terzi	
CA.04.09.08.06.11	Spese per commissioni ed intermediazioni bancarie	
CA.04.09.08.07.01	Consulenze tecnico-scientifiche	
CA.04.09.08.07.02	Consulenze tecnico-amministrative	
CA.04.09.08.07.03	Spese per liti (patrocinio legale)	
CA.04.09.08.07.04	Spese notarili	
CA.04.09.08.08.01	Prestazioni di lavoro autonomo	
CA.04.09.08.09.01	Prestazioni di servizi tecnico/amministrativi da enti terzi	
CA.04.09.08.09.02	Altre prestazioni e servizi da terzi	
CA.04.09.08.10.01	Collaborazioni coordinate e continuative	
CA.04.09.08.11.01	Costi per fornitura di lavoro interinale	
CA.04.09.09.01.01	Carburanti, combustibili e lubrificanti per autovetture	
CA.04.09.09.01.02	Carburanti, combustibili e lubrificanti per autocarri, mezzi agricoli e altri mezzi di trasporto	
CA.04.09.09.01.03	Cancelleria e altro materiale di consumo	
CA.04.09.09.01.04	Libretti e diplomi	
CA.04.09.09.01.05	Vestiario	
CA.04.09.09.01.06	Materiale per ricorrenze elettorali	
CA.04.09.09.01.07	Altri materiali	
CA.04.09.09.01.08	Combustibili per riscaldamento	
CA.04.09.09.02.01	Acquisto beni strumentali	
CA.04.09.09.02.02	Acquisto software per pc	
CA.04.09.09.03.01	Sconti e abbuoni passivi	
CA.04.09.09.03.02	Sconti abbuoni e premi su acquisti	
CA.04.09.10.01.01	Rimanenze iniziali di materiali	
CA.04.09.11.01.01	Leasing operativo ed altre forme di locazione di beni mobili	
CA.04.09.11.01.02	Leasing operativo ed altre forme di locazione di autovetture di rappresentanza e di servizio	
CA.04.09.11.01.03	Leasing operativo ed altre forme di locazione di autocarri, mezzi agricoli e altri mezzi di trasporto	

Allegato n. 3

PROPOSTA DI RIAPPLICAZIONE ALL'ESERCIZIO 2018 DELLE DISPONIBILITA' LIBERE RISULTANTI AL 31/12/2017

Voce COAN	Denominazione	Totale
CA.04.09.11.01.04	Noleggio fax e fotocopiatrici	
CA.04.09.11.01.05	Altri noleggi	
CA.04.09.11.02.01	Fitti passivi per locazione di edifici	
CA.04.09.11.02.02	Altri fitti passivi	
CA.04.09.11.03.01	Licenze software	
CA.04.09.12.01.01	Missioni e rimborsi spese trasferta organi istituzionali	
CA.04.09.12.01.02	Gettoni/indennita' ai membri degli organi istituzionali di governo e controllo	
CA.04.09.12.01.03	Indennita' di carica	
CA.04.09.12.01.04	Gettoni/indennita' ai membri degli organi istituzionali che non siano di amministrazione e controllo	
CA.04.09.12.01.05	Gettoni/indennita' ai membri del collegio dei revisori	
CA.04.09.12.01.06	Gettoni/indennita' ai membri del nucleo di valutazione	
CA.04.09.12.01.07	Garante di Ateneo	
CA.04.09.12.02.01	Quote associative	
CA.04.09.12.02.02	Compensi per commissioni di concorso del personale interno ed esterno	
CA.04.09.12.02.03	Altri costi per attivita' istituzionali	
CA.04.09.12.02.04	Costi per attività sportive L. 394/77	
CA.04.09.12.02.05	Cus - attivita' sportiva e gestione impianti sportivi	
CA.04.09.12.02.06	Arrotondamenti negativi	
CA.04.09.12.02.07	Visite medico-fiscali	
CA.04.09.12.02.08	Accertamenti sanitari	
CA.04.09.12.02.09	Equo indennizzo	
CA.04.09.12.02.10	Provvidenze a favore del personale	
CA.04.09.12.02.11	Circolo San Martino - attività sociali del personale	
CA.04.09.12.02.12	Prestazioni INAIL - gestione per conto	
CA.04.09.12.02.13	Spese condominiali	
CA.04.10.01.01.01	QUOTE DI AMMORTAMENTO COSTI DI IMPIANTO, DI AMPLIAMENTO E DI SVILUPPO	
CA.04.10.01.01.02	QUOTE DI AMMORTAMENTO DIRITTI DI BREVETTO E DIRITTI DI UTILIZZAZIONE DELLE OPERE DI INGEGNO	
CA.04.10.01.01.03	QUOTE DI AMMORTAMENTO CONCESSIONI, LICENZE, MARCHI E DIRITTI SIMILI	
CA.04.10.01.01.04	QUOTE DI AMMORTAMENTO ALTRE IMMOBILIZZAZIONI IMMATERIALI	
CA.04.10.02.01.01	QUOTE DI AMMORTAMENTO TERRENI E FABBRICATI	
CA.04.10.02.01.02	QUOTE DI AMMORTAMENTO IMPIANTI E ATTREZZATURE	
CA.04.10.02.01.03	QUOTE DI AMMORTAMENTO ATTREZZATURE SCIENTIFICHE	
CA.04.10.02.01.04	QUOTE DI AMMORTAMENTO PATRIMONIO LIBRARIO, OPERE D'ARTE, D'ANTIQUARIATO E MUSEALI	
CA.04.10.02.01.05	QUOTE DI AMMORTAMENTO MOBILI E ARREDI	
CA.04.10.02.01.06	QUOTE DI AMMORTAMENTO ALTRE IMMOBILIZZAZIONI MATERIALI	
CA.04.10.03.01.01	SVALUTAZIONE COSTI DI IMPIANTO, DI AMPLIAMENTO E DI SVILUPPO	

Allegato n. 3

PROPOSTA DI RIAPPLICAZIONE ALL'ESERCIZIO 2018 DELLE DISPONIBILITA' LIBERE RISULTANTI AL 31/12/2017

Voce COAN	Denominazione	Totale
CA.04.10.03.01.02	SVALUTAZIONE DIRITTI DI BREVETTO E DIRITTI DI UTILIZZAZIONE DELLE OPERE DI INGEGNO	
CA.04.10.03.01.03	SVALUTAZIONE CONCESSIONI, LICENZE, MARCHI E DIRITTI SIMILI	
CA.04.10.03.01.04	SVALUTAZIONE ALTRE IMMOBILIZZAZIONI IMMATERIALI	
CA.04.10.03.01.05	SVALUTAZIONE TERRENI E FABBRICATI	
CA.04.10.03.01.06	SVALUTAZIONE IMPIANTI E ATTREZZATURE	
CA.04.10.03.01.07	SVALUTAZIONE ATTREZZATURE SCIENTIFICHE	
CA.04.10.03.01.08	SVALUTAZIONE PATRIMONIO LIBRARIO, OPERE D'ARTE, D'ANTIQUARIATO E MUSEALI	
CA.04.10.03.01.09	SVALUTAZIONE MOBILI E ARREDI	
CA.04.10.03.01.10	SVALUTAZIONE ALTRE IMMOBILIZZAZIONI MATERIALI	
CA.04.10.04.01.01	Perdita su crediti compresi nell'attivo circolante e nelle disponibilità liquide	
CA.04.11.01.01.01	Quote di accantonamento al fondo svalutazione crediti	
CA.04.11.01.01.02	QUOTE DI ACCANTONAMENTO AI FONDI ARRETRATI DA CORRISPONDERE AL PERSONALE NEGLI ESERCIZI FUTURI	
CA.04.11.01.03.01	Quote di accantonamento al fondo per cause e controversie in corso	
CA.04.11.01.04.01	Quote di accantonamento ai fondi per altri rischi e oneri	
CA.04.11.01.05.01	Quota di esercizio per altri accantonamenti	
CA.04.11.01.06.01	Accantonamento per fondi di quiescenza	
CA.04.11.01.06.02	Accantonamento per fondi tfr personale	
CA.04.12.01.01.01	TRASFERIMENTI INTERNI CORRENTI	
CA.04.12.01.01.02	TRASFERIMENTI INTERNI PER INVESTIMENTI	
CA.04.12.01.01.03	TRASFERIMENTI INTERNI SU ATTIVITA' CONTO TERZI	
CA.04.12.01.01.04	TRASFERIMENTI INTERNI PER RESTITUZIONI E RIMBORSI	
CA.04.12.01.01.05	ALTRI TRASFERIMENTI INTERNI	
CA.04.12.01.01.06	TRASFERIMENTI VARI	
CA.04.12.01.02.01	Versamenti allo Stato per riduzioni di spesa	
CA.04.12.01.03.01	Tassa di rimozione rifiuti solidi urbani	
CA.04.12.01.03.02	Imposte sul registro	
CA.04.12.01.03.03	Valori bollati	
CA.04.12.01.03.04	Altri tributi	
CA.04.12.01.03.05	Iva indetrabile	
CA.04.12.01.03.06	Imposte sul patrimonio	
CA.04.12.01.03.07	Tributi su lasciti e donazioni	
CA.04.13.02.01.01	Oneri finanziari	
CA.04.13.02.02.01	Interessi passivi	
CA.04.13.03.01.01	Perdite su cambi	
CA.04.14.02.01.01	Svalutazione titoli e partecipazioni	
CA.04.15.02.01.01	Oneri straordinari per il personale	

Allegato n. 3

PROPOSTA DI RIAPPLICAZIONE ALL'ESERCIZIO 2018 DELLE DISPONIBILITA' LIBERE RISULTANTI AL 31/12/2017

Voce COAN	Denominazione	Totale
CA.04.15.02.02.01	Rimborsi tasse e contributi agli studenti	
CA.04.15.02.03.01	Oneri straordinari per recuperi e rimborsi	
CA.04.15.02.04.01	Altri oneri straordinari	44.192,57
CA.04.15.02.05.01	Imposte relative ad esercizi precedenti	
CA.04.16.01.01.01	Imposte sul reddito	
CA.04.17.01.01.01	Voce per variazione tecnica	
CA.07.70.01.01.01	Costi operativi progetti - quota di competenza per finanziamenti competitivi da miur - progetti di ricerca di rilevante interesse nazionale	
CA.07.70.01.01.02	Costi operativi progetti - quota di competenza per finanziamenti competitivi da miur - fondo per gli investimenti della ricerca di base (firb)	
CA.07.70.01.01.03	Costi operativi progetti - quota di competenza per altri finanziamenti competitivi da miur	
CA.07.70.01.02.01	Costi operativi progetti - quota di competenza per finanziamenti competitivi da altri ministeri per ricerca scientifica	
CA.07.70.01.02.02	Costi operativi progetti - quota di competenza per finanziamenti competitivi da stato (organi diversi da ministeri) per ricerca scientifica	
CA.07.70.01.02.03	Costi operativi progetti - quota di competenza per finanziamenti competitivi per ricerca da regioni e province autonome	
CA.07.70.01.02.04	Costi operativi progetti - quota di competenza per finanziamenti competitivi per ricerca da province	
CA.07.70.01.02.05	Costi operativi progetti - quota di competenza per finanziamenti competitivi per ricerca da città metropolitane	
CA.07.70.01.02.06	Costi operativi progetti - quota di competenza per finanziamenti competitivi per ricerca da comuni	
CA.07.70.01.02.07	Costi operativi progetti - quota di competenza per finanziamenti competitivi per ricerca da camere di commercio	
CA.07.70.01.02.08	Costi operativi progetti - quota di competenza per finanziamenti competitivi per ricerca da altre università	
CA.07.70.01.02.09	Costi operativi progetti - quota di competenza per finanziamenti competitivi per ricerca da altre amministrazioni pubbliche	
CA.07.70.01.03.01	Costi operativi progetti - quota di competenza per finanziamenti competitivi da cnr	
CA.07.70.01.03.02	Costi operativi progetti - quota di competenza per finanziamenti competitivi per ricerca da enti di ricerca diversi dal cnr	
CA.07.70.01.04.01	Costi operativi progetti - quota di competenza per finanziamenti competitivi per ricerca da parte dell'unione europea	
CA.07.70.01.04.02	Costi operativi progetti - quota di competenza per finanziamenti competitivi per ricerca da parte di organismi internazionali	
CA.07.70.01.05.01	Costi operativi progetti - attivita' c/terzi e cessione di risultati di ricerca	
CA.07.70.01.06.01	Costi operativi progetti - finanziamenti non competitivi per la ricerca	123,70
CA.07.70.01.07.01	Costi operativi progetti - Centri Autonomi di Gestione con Autonomia Negoziale	
CA.07.70.01.08.01	Costi operativi progetti per attività di formazione	
CA.08.80.01.01.01	Costi di investimento progetti - quota di competenza per finanziamenti competitivi da miur - progetti di ricerca di rilevante interesse nazionale	
CA.08.80.01.01.02	Costi di investimento progetti - quota di competenza per finanziamenti competitivi da miur - fondo per gli investimenti della ricerca di base (firb)	
CA.08.80.01.01.03	Costi di investimento progetti - quota di competenza per altri finanziamenti competitivi da miur	
CA.08.80.01.02.01	Costi di investimento progetti - quota di competenza per finanziamenti competitivi da altri ministeri per ricerca scientifica	
CA.08.80.01.02.02	Costi di investimento progetti - quota di competenza per finanziamenti competitivi da stato (organi diversi da ministeri) per ricerca scientifica	
CA.08.80.01.02.03	Costi di investimento progetti - quota di competenza per finanziamenti competitivi per ricerca da regioni e province autonome	
CA.08.80.01.02.04	Costi di investimento progetti - quota di competenza per finanziamenti competitivi per ricerca da province	
CA.08.80.01.02.05	Costi di investimento progetti - quota di competenza per finanziamenti competitivi per ricerca da città metropolitane	
CA.08.80.01.02.06	Costi di investimento progetti - quota di competenza per finanziamenti competitivi per ricerca da comuni	
CA.08.80.01.02.07	Costi di investimento progetti - quota di competenza per finanziamenti competitivi per ricerca da camere di commercio	

Allegato n. 3

PROPOSTA DI RIAPPLICAZIONE ALL'ESERCIZIO 2018 DELLE DISPONIBILITA' LIBERE RISULTANTI AL 31/12/2017

Voce COAN	Denominazione	Totale
CA.08.80.01.02.08	Costi di investimento progetti - quota di competenza per finanziamenti competitivi per ricerca da altre università	
CA.08.80.01.02.09	Costi di investimento progetti - quota di competenza per finanziamenti competitivi per ricerca da altre amministrazioni pubbliche	
CA.08.80.01.03.01	Costi di investimento progetti - quota di competenza per finanziamenti competitivi da cnr	
CA.08.80.01.03.02	Costi di investimento progetti - quota di competenza per finanziamenti competitivi per ricerca da enti di ricerca diversi dal cnr	
CA.08.80.01.04.01	Costi di investimento progetti - quota di competenza per finanziamenti competitivi per ricerca da parte dell'unione europea	
CA.08.80.01.04.02	Costi di investimento progetti - quota di competenza per finanziamenti competitivi per ricerca da parte di organismi internazionali	
CA.08.80.01.05.01	Costi di investimento progetti - attivita' in conto terzi e cessione di risultati di ricerca	
CA.08.80.01.06.01	Costi di investimento progetti - finanziamenti non competitivi per la ricerca	
CA.08.80.01.07.01	Costi di investimento progetti - Centri Autonomi di Gestione con Autonomia Negoziale	
CA.09.90.01.01.01	Mobilità e scambi culturali docenti - Budget economico	
CA.09.90.01.01.02	Rapporti Internazionali, scambi culturali - Budget economico	
CA.09.90.01.01.03	Comunicazione di Ateneo - Budget economico	
CA.09.90.01.01.04	Acquisto, manutenzione, noleggio, esercizio veicoli - Budget economico	
CA.09.90.01.01.05	Spese inerenti l'orientamento universitario - Budget economico	
CA.09.90.01.01.06	Progetti III Missione - Budget economico	
CA.09.90.01.01.07	Spese funzionamento Servizio Prevenzione e Protezione - Budget economico	27.783,07
CA.09.90.01.01.08	Funzionamento Strutture Didattiche finanziate da Esterni - Budget economico	
CA.09.90.01.01.09	Ricerca di base - Budget economico	
CA.09.90.01.01.10	Funzionamento strutture didattiche - Budget economico	
CA.09.90.01.01.11	Costi operativi su economie progetti - Budget economico	
CA.09.90.01.01.12	Costi operativi altri progetti Amministrazione centrale - Budget economico	
CA.09.90.01.01.13	Informatizzazione Servizi - Budget economico	
CA.09.90.01.01.14	Gestione e sviluppo Rete di Ateneo - Budget economico	
CA.10.10.01.01.01	Costruzione, ristrutturazione e restauro fabbricati	
CA.10.10.01.01.02	Costruzione impianti	
CA.10.10.01.01.03	Ricostruzione e trasformazione fabbricati	
CA.10.10.01.01.04	Ricostruzione e trasformazione impianti	
CA.10.10.01.01.05	Manutenzione straordinaria immobili	
CA.10.10.01.01.06	Manutenzione straordinaria impianti	
CA.10.10.01.01.07	Spese in applicazione D.L. 626/94	
CA.10.10.01.01.08	Manutenzione straordinaria immobili - Messa a norma e sicurezza - Spese in applicazione D.Lgs. 81/2008	
CA.10.10.01.01.09	Informatizzazione Servizi - Budget investimenti	
CA.10.10.01.01.10	Gestione e sviluppo Rete di Ateneo - Budget investimenti	
CA.10.10.01.01.11	Mobilità e scambi culturali docenti - Budget investimenti	
CA.10.10.01.01.12	Rapporti Internazionali, scambi culturali - Budget investimenti	
CA.10.10.01.01.13	Comunicazione di Ateneo - Budget investimenti	

DIPARTIMENTO
DI INGEGNERIA

Allegato n. 3

PROPOSTA DI RIAPPLICAZIONE ALL'ESERCIZIO 2018 DELLE DISPONIBILITA' LIBERE RISULTANTI AL 31/12/2017

Voce COAN	Denominazione	Totale
CA.10.10.01.01.14	Acquisto, manutenzione, noleggio, esercizio veicoli - Budget investimenti	
CA.10.10.01.01.15	Spese inerenti l'orientamento universitario - Budget investimenti	
CA.10.10.01.01.16	Progetti III Missione - Budget investimenti	
CA.10.10.01.01.17	Spese funzionamento Servizio Prevenzione e Protezione - Budget investimenti	
CA.10.10.01.01.18	Funzionamento Strutture Didattiche finanziate da Esterni - Budget investimenti	
CA.10.10.01.01.19	Ricerca di base - Budget investimenti	
CA.10.10.01.01.20	Funzionamento strutture didattiche - Budget investimenti	
CA.10.10.01.01.21	Costi operativi su economie progetti - Budget investimenti	
CA.10.10.01.01.22	Costi operativi altri progetti Amministrazione centrale - Budget investimenti	
	TOTALE GENERALE AL 31/12/2017	75.007,29

IL SEGRETARIO AMM.VO
(Sig. Giovanni Magara)

UNIVERSITÀ DEGLI STUDI DI PUGLIA DIPARTIMENTO DI INGEGNERIA

IL DIRETTORE
(Prof. Giuseppe Saccomandi)

Development of technologies for knowledge transfer and adaptation among intelligent systems for home automation applications

Sviluppo di tecnologie per il trasferimento e l'adattamento di conoscenza tra sistemi intelligenti per applicazioni domotiche

Relazione Assegno di Ricerca

December, 2016 - December, 2017

Gabriele Costante

Dipartimento di Ingegneria
Università degli studi di Perugia

Contents

Chapter 1

Introduction

This document contains the report of my research activities in the last year (2016/2017).

The first part of the report describes a novel contribution on human-computer interaction systems. The proposed strategy addresses the problem of providing a natural language description of a scene perceived through a vision sensors. This is referred by the research community as Natural Language Video Description (NLVD) and it has a huge impact on systems for home automation. This is the case blind people assistance, where an NLVD approach could provide them a description about what is happening in the scene. Another important application is the anomaly detection inside buildings, to detect dangerous scenarios, such as fire or security breaches. Finally, a user-friendly human-computer interface, such as a natural language interface, could definitely help users to better perceive and interact with home automation systems. In particular, in this work, we investigate the ability to generate natural language descriptions for the scene it observes. We achieve this capability via a Deep Recurrent Neural Network (D-RNN) architecture completely based on the Gated Recurrent Unit (GRU) paradigm. The system is able to generate complete sentences describing the scene, dealing with the hierarchical nature of the temporal information contained in image sequences. The proposed approach has fewer parameters than previous State-of-the-Art architectures, thus it is faster to train and smaller in memory occupancy. These benefits do not affect the prediction performance. In fact, we show that our method outperforms or is comparable to previous approaches in terms of quantitative metrics and qualitative evaluation when tested on benchmark publicly available datasets and on a new dataset we introduce in this work.

The second part of the report describes a novel approach for robot localization based on vision sensors. Applications in the context of home automation system are usually characterized by GPS-denied environments (*i.e.* buildings or offices). Hence, vision-based localization is crucial to enable mobile platforms to operate in these scenarios. This work proposes a novel deep network architecture to solve the camera Ego-Motion estimation problem. A motion estimation network generally learns features similar to Optical Flow (OF) fields starting from sequences of images. This OF can be described by a lower dimensional latent space. Previous research has shown how to find linear approximations of this space. We propose to use an Auto-Encoder network to find a non-linear representation of the OF manifold. In addition, we propose to learn the latent space jointly with the estimation task, so that the learned OF features become a more robust description of the OF input. We call this novel architecture Latent Space Visual Odometry (LS-VO). The experiments show that LS-VO achieves a considerable increase in performances with respect to baselines, while the number of parameters of the estimation network only slightly increases.

Chapter 2

Full-GRU Natural Language Video Description

2.1 Introduction

The ability to provide a description of the scene in a form that every user can easily understand is keystone for the success of effective and user-friendly service robotics products. In fact, a natural language description offers an interpretable manifestation of the robot's inner representation of the scene and is also a good basis for natural language question answering about what is happening in the environment. Hence, this functionality would provide a friendly interface also for non-expert people who would then be able to easily interact with their home robot in the near future.

In the sight of this, this work addresses the problem of describing a scene in natural language, which is usually referred to as Natural Language Video Description (NLVD). Here we formalize this problem as a Machine Translation (MT) one, from "visual language" to English. Basically, the information in form of a varying length video sequence is encoded in a fixed-length vector and then decoded in form of varying length English sentence (Fig. 2.1).

The video translation is performed via D-RNNs, *i.e.* recurrent models that are able to deal with both long and short term dependencies in data sequences. Most of the previous approaches rely on the Long Short-Term Memory (LSTM) [1] architecture. However, recent works have devised novel recurrent architectures, such as Gated Recurrent Unit (GRU) [2], Neural Turing Machines (NTM) [3] and Memory Networks [4], that have shown promising results in different applications. Hence, which memory management strategy is the most suitable one for the problem of NLVD is still an open question that is worth being investigated. For this purpose, in this work we compare a NLVD system completely based on the GRU paradigm and State-of-the-Art approaches that exploit LSTMs.

In addition, the applicability of such algorithms to mobile robots poses additional constraints in terms of both time and memory complexity. In fact, in these applications particular attention must be paid to the robot's limited memory capacity and to the quick reactivity to the user's requests.

In this work, a full-GRU NLVD system is proposed, that is able to deal with the hierarchical nature of the temporal information typical of natural and generic video sequences and obtains comparable to superior performance with respect to more complex State-of-the-Art systems. The proposed system features a GRU cell modified in order to automatically change its temporal connection if a boundary, *i.e.* a significant modification in the scene, is detected. To the best of our knowledge, this is the first full-GRU encoder-decoder architecture applied to the problem of NLVD. In particular, having a simpler structure (fewer parameters) than other gated recurrent layers (*e.g.* the LSTM), the GRU block is faster-training and memory saving. This makes it appealing for robotics applications. In addition, a new small dataset for NLVD in typical service robotics scenarios is used, which offers a fair test bench for the specific application we target. The relevance of this dataset, is twofold. First, this is the first dataset specifically collected in typical applicative contexts of a service robot. Second, it gives more insights on the actual performance of the NLVD models we are testing. Indeed, State-of-the-Art NLVD systems are commonly trained and tested on videos from the same datasets, which may make their evaluation biased.

To summarize, the main contributions of this work are:

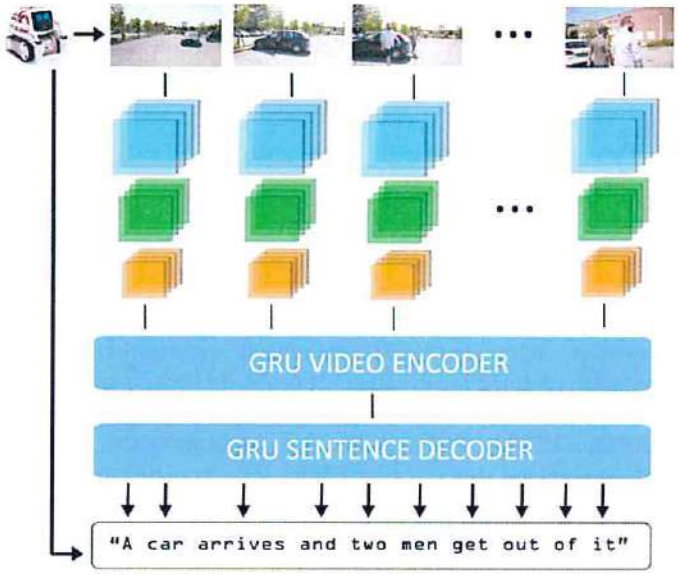


Figure 2.1: Overview of the proposed NLVD system. The robot observes a generic and complex scene and represents it taking into account both the visual and temporal information, represented via ConvNet features and an encoding vector, respectively. Then, it outputs a natural language sentence describing the observed scene. The proposed encoder-decoder scheme is entirely based on GRU recurrent units.

- We devise a novel architecture for NLVD that is able to capture hierarchical temporal information in general video sequences.
- We show that our full-GRU method obtains superior performance compared to State-of-the-Art methods that harness LSTM, while having fewer parameters.
- We present a dataset that features a wide range of contexts that are typical for service robotics applications.

The remainder of this chapter is organized as follows. In Section 2.3 the proposed approach is described. Section 2.4 provides a detailed description of the experimental results and conclusion are drawn in Section 2.5.

2.2 Related Work

In recent years, many researchers from both Computer Vision and Natural Language Processing communities are studying the problem of describing generic videos using natural language phrases (see *e.g.* [5, 6]).

Some popular approaches [6, 7] are based on filling-in predefined template sentences with the subject-verb-object concepts detected in the video. In particular, an object detector (*e.g.* a CNN as in [7]) is used to recognize the main actors in the video and a Probabilistic Graphical Model (PGM) (*e.g.* an Hidden Markov Model as in [6]) is used to predict the relation between them. These approaches have major limitations. First, the type and the number of the objects and the relations that can be described are limited to those that the detector and the PGM can estimate. Second, the output descriptions lack in diversity and naturalness.

Other works [8] propose to tackle the NLVD task in a multi-modal retrieval fashion. In particular, given a corpus of paired videos and text, the system describes a new video using the sentence associated to the most similar video in the corpus [8]. Also this approach has some weaknesses. In particular, the

system is constrained to use the same sentences in the corpus, which may be not semantically relevant for the new scene to describe.

Among the proposed strategies, treating the NLVD problem as a Machine Translation (MT) one gained popularity [9] and D-RNN demonstrated to be a very promising instrument [10, 11, 12]. This is particularly true when recurrent models are combined with State-of-the-Art Convolutional Neural Networks (ConvNet), even pre-trained.

Despite of the success of recent State-of-the-Art approaches, NLVD is still a particularly challenging problem, firstly due to the "object" of the description itself, *i.e.* the video sequence, that is typically open-domain and complex in real scenarios. In particular, the content of the videos can be highly diverse and the temporal dependencies between the depicted events can be at different granularity. Some architectures exist that produce accurate descriptions of videos, but in general these are either very short or very specific, or both, *i.e.* they depict simple activities of a particular domain with few "actors" in the scene [11, 7]. Those kinds of video sequences are far simpler than the typical complexity that a robot faces in real application contexts. The systems presented in [10] and [12] deal with generic and complex videos. Both of them represent the video sequence by mean-pooling the ConvNet features extracted from each frame, then decode the sentence with a LSTM-based decoder. A major drawback of those strategies is that they do not take into account the temporal structure of the video sequences due to mean-pooling.

Indeed, when considering more complex and generic video sequences it is crucial to deal with temporal dependencies at different granularity. This is done in [13, 14] and also in this work, where a hierarchical representation of the temporal information is explicitly learned. In [13] the authors draw from ConvNets the idea of convolutional operations and build a multi-level LSTM-based encoding able to capture longer time dependencies between the content of the frames. Then, a LSTM decoder produces the description exploiting an attention mechanism (that is basically a learned weighting strategy). The work of [14] is the most similar to our work. It presents a LSTM-based decoder that contains a boundary-aware LSTM cell. This cell and a second layer LSTM block build an encoding of the video sequence which is then decoded via a GRU.

All of the above approaches, either consists of full-stack LSTM architectures or limit the use of the GRU to the decoding phase. In this work, we present an encoder-decoder architecture that is completely based on GRU blocks, which have fewer parameters than LSTM, thus resulting arguably more suitable for robotics applications. This is motivated also by the study reported in [15], that compares the GRU and the LSTM cells on various tasks. Using input, state and output vectors of the same dimensionality, the GRU outperforms or is comparable to the LSTM in terms of convergence time, parameters update and generalization.

2.3 Encoder-Decoder full-GRU Architecture

In this section our proposed model is presented. The video frames are described via the *ResNet50* and the *C3D* ConvNets (see 2.3.1). The obtained feature vectors are then fed, one at each time-step, in the first layer of the encoder. This is our proposed BA-GRU recurrent block, that encodes the video frames until a boundary is detected. Afterwards, the first-layer encoding is fed to the second layer of the encoder, which consists of a classical GRU block (see 2.3.2). The output of the encoding phase is a vector representing the entire video sequence. Finally, the GRU decoder produces the description emitting the most probable word at each time-step, conditioned to the video vector representation and the previous emitted words (see 2.3.3). The captioning process ends when a <EOS> tag (*i.e.* the full-stop) is emitted. A pictorial representation of the system is shown in Fig. 2.2.

2.3.1 Video Frames and Caption Words Preprocessing

The video frames are preprocessed as follows. The output of the last fully connected layer of the *ResNet50* ConvNet [16] is computed every five video frames, to capture the appearance of the scene. To the same video frames is associated also the output of the *C3D* ConvNet [17] to capture the movement in the scene, based on partially overlapped sliding windows of frames. The output of the two ConvNets are concatenated (forming a 2048+4096-dimensional vector) and mapped in a learned 512-dimensional linear embedding. The entire video is then represented by a sequence of features vectors $(\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n)$, where the x_i vectors are the feature vectors extracted from the frames of the video.

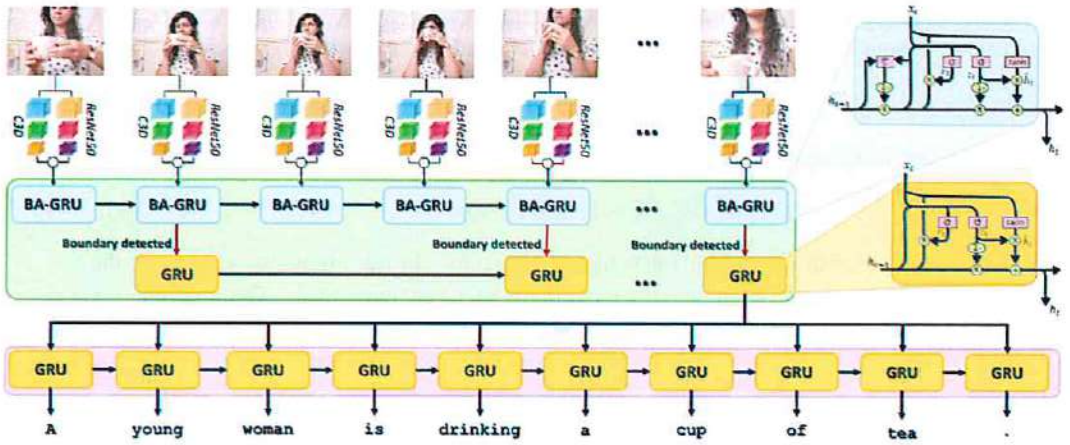


Figure 2.2: Architecture of the proposed system. Recurrent layers are depicted as unfolded graphs for explanatory purpose.

The captions are preprocessed as follows. First, the words are converted to lower-case and the punctuation characters are removed. Then, begin-of-sentence (<BOS>) and end-of-sentence (<EOS>) tags are added before and behind the sentence, respectively. Finally, the sentences are tokenized. From the tokenized sentences, we build a vocabulary (D). To prevent the formation of a large vocabulary containing many rare words, we retain only those tokens that appear at least five times in the caption corpus. To each token is associated an index in the vocabulary, based on its frequency in the vocabulary. A caption is then represented by a list of one-hot vectors $(\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_L)$, each of them corresponding to the representation of its words in the vocabulary. Similarly to what is done for the frames features, the captions are mapped in a learned 512-dimensional linear embedding.

2.3.2 Video Encoder

In this work, we build upon the boundary-aware LSTM (BA-LSTM) cell presented in [14] and devise a boundary-aware GRU (BA-GRU) cell. This cell is the first layer of a two-layers encoder. The second layer of the encoder is a simple GRU cell [2].

The BA-GRU is a modification of the classical GRU cell (see Fig. 2.2, top right). The GRU is a recurrent neural networks with gating strategies to model wider temporal dependencies in the input sequence. The GRU is characterized by an update gate $\mathbf{z}_t$ and a reset gate $\mathbf{r}_t$. At each timestep, a candidate activation $\tilde{\mathbf{h}}_t$ is computed based on the current input $\mathbf{x}_t$, the previous inner state $\mathbf{h}_{t-1}$ and the values of the gates. In particular, the $\mathbf{z}_t$ gate controls how much the inner state $\mathbf{h}_t$ has to be updated, the $\mathbf{r}_t$ gate controls how much the previous inner state $\mathbf{h}_{t-1}$ influences the candidate inner state value $\tilde{\mathbf{h}}_t$. More formally, the GRU is defined by the following equations:

$$\mathbf{h}_t = (1 - \mathbf{z}_t)\mathbf{h}_{t-1} + \mathbf{z}_t\tilde{\mathbf{h}}_t \tag{2.1}$$

$$\tilde{\mathbf{h}}_t = tanh(W_{hx}\mathbf{x}_t + W_{hh}(\mathbf{r}_t \odot \mathbf{h}_{t-1}) + \mathbf{b}_h) \tag{2.2}$$

$$\mathbf{r}_t = \sigma(W_{rx}\mathbf{x}_t + W_{rh}\mathbf{h}_{t-1} + \mathbf{b}_r) \tag{2.3}$$

$$\mathbf{z}_t = \sigma(W_{zx}\mathbf{x}_t + W_{zh}\mathbf{h}_{t-1} + \mathbf{b}_z) \tag{2.4}$$

where the W_{**}s and $\mathbf{b}_*$s are learnable weight matrices and bias vectors, σ is the sigmoid function, $tanh$ is the hyperbolic tangent function and $\odot$ is the element-wise product.

In this work, we modify the GRU by adding a boundary aware gate s_t, that modifies the inner connectivity of the unit based on the input and the inner state. In particular, when a substantial change in input

sequence occurs, a boundary is estimated by a learnable function. Consequently, the inner state $\mathbf{h}_{t-1}$ is emitted as output (we denote it as $\mathbf{h}_k^{e1} \doteq \mathbf{h}_{t-1}$) and then re-initialized to zero according to:

$$\mathbf{h}_{t-1} \leftarrow \mathbf{h}_{t-1}(1 - s_t) \tag{2.5}$$

The boundary-aware gate is defined as follows:

$$s_t = \tau(\mathbf{w}_s^T(W_{sx}\mathbf{x}_t + W_{sh}\mathbf{h}_{t-1} + \mathbf{b}_s)) \tag{2.6}$$

where W_{**}s and $\mathbf{b}_s$ are learnable weights matrices and bias vectors. In this study, we set to 128 the number of their rows. The row vector $\mathbf{w}_s^T$ makes the input to the $\tau(\cdot)$ function a scalar. The $\tau(\cdot)$ function is given by:

$$\tau(\cdot) = \begin{cases} 1 & \text{if } \sigma(\cdot) < 0.5 \\ 0 & \text{otherwise} \end{cases} \tag{2.7}$$

The given output $\mathbf{h}_k^{e1}$ summarizes the video substream before the boundary, which is then composed by homogeneous frames. For an input video, the BA-GRU block outputs as many vectors $\mathbf{h}_k^{e1}$, as the number of detected boundaries $(\mathbf{h}_1^{e1}, \mathbf{h}_2^{e1}, ..., \mathbf{h}_m^{e1})$, with $m \leq n$. Those vectors are given in input to the second layer of the encoder, which is a standard GRU block. This layer encodes the $\mathbf{h}_k^{e1}$ vectors in a unique vector $\mathbf{v}$, that represents the entire video. The $\mathbf{v}$ vector, that is the final output of the two-layer encoder, is fed to the decoder.

The Boundary-Aware Gate Training Details

The output s_t of the boundary-aware gate can be either 0 or 1, depending on the value of a sigmoid function applied to the input of the gate. Thus, following the approach of [14], in the training phase we model it as a stochastic binary neuron and learned its weights, while in test phase we use it with the learned weights as the deterministic neuron defined in Eq.2.7. In particular, we re-write the activation function $\tau(\cdot)$ as:

$$\tau(\cdot) = \mathbf{1}_{\sigma(\cdot)>z}, \qquad z \sim \mathcal{U}(0,1) \tag{2.8}$$

where $\mathbf{1}_{\cdot}$ is the indicator function and $\mathcal{U}(0,1)$ denotes the uniform distribution between 0 and 1.

Note that $\tau(\cdot)$ in Eq.2.7 is basically the composition of a step function and a sigmoid function. Thus, its derivative is equal to 0 everywhere except in 0, *i.e.* it is not continuous and smooth and it is also mostly flat. Hence, we cannot apply the standard back-propagation to compute the gradient in this gate. To overcome this issue, we follow the same approach of [14], that estimated the gradient by approximating the step function $\tau(\cdot)$ as the identity function [18]. The derivative of $\tau(\cdot)$ then becomes:

$$\frac{\partial \tau}{\partial(\cdot)}(\cdot) = \frac{\partial \sigma}{\partial(\cdot)}(\cdot) = \sigma(\cdot)(1 - \sigma(\cdot)) \tag{2.9}$$

In the test phase, we use the deterministic form of $\tau(\cdot)$ (Eq.2.7), the parameters of which have been learned in the training phase using Eq.2.8 (in the forward pass) and Eq.2.9 (in the backward pass).

2.3.3 Caption Decoder

The decoder takes as input the video representation $\mathbf{v}$ and the ground truth sentence $(\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_L)$. At each timestep, it outputs a word $\mathbf{y}_i$ that is the most probable next word of the description, given the previous output words and the video representation.

To handle both the time-varying input $(\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_L)$ and the constant input $\mathbf{v}$, we modify Eq.2.2-2.4 from the original GRU formulation as:

$$\tilde{\mathbf{h}}_t = tanh(W_{hy}W_w\mathbf{y}_t + W_{hv}\mathbf{v} + W_{hh}(\mathbf{r}_t \odot \mathbf{h}_{t-1}) + \mathbf{b}_h) \tag{2.10}$$

$$\mathbf{r}_t = \sigma(W_{ry}W_w\mathbf{y}_t + W_{rv}\mathbf{v} + W_{rh}\mathbf{h}_{t-1} + \mathbf{b}_r) \tag{2.11}$$

$$\mathbf{z}_t = \sigma(W_{zy}W_w\mathbf{y}_t + W_{zv}\mathbf{v} + W_{zh}\mathbf{h}_{t-1} + \mathbf{b}_z) \tag{2.12}$$

where the W_{**}s and $\mathbf{b}_*$s are learnable weight matrices and bias vectors respectively, σ is the sigmoid function and $\odot$ is the element-wise product. The matrix W_w maps the input one-hot vectors representing the words $\mathbf{y}_t$ in the vocabulary space in a lower dimensional space (512-dimensional embedding). The output of the decoder (which we denote $\mathbf{h}_t^d \doteq \mathbf{h}_t$) is then mapped back in the original higher dimensional space as $\mathbf{y}_t = W_p \mathbf{h}_t^d$.

The probability of the next word in the description is modelled via the softmax function, *i.e.*

$$Pr(\mathbf{y}_t|\mathbf{y}_0, \mathbf{y}_1, ..., \mathbf{y}_{t-1}, \mathbf{v}) \sim \frac{e^{\mathbf{y}_t^T W_p \mathbf{h}_t^d}}{\sum_{\mathbf{y} \in D} e^{\mathbf{y}^T W_p \mathbf{h}_t^d}} \tag{2.13}$$

Finally, the objective function to optimize is the log-likelihood of the correct words over the sentence *i.e.*

$$\max_{\mathbf{W}} \sum_{t=1}^{L} log Pr(\mathbf{y}_t|\mathbf{y}_0, \mathbf{y}_1, ..., \mathbf{y}_{t-1}, \mathbf{v}) \tag{2.14}$$

where $\mathbf{W}$ denotes all the parameters of the model.

2.4 Experiments and Results

In this section, we present the experimental setup and the obtained results of our method.

2.4.1 Datasets Details

We employ two publicly available large datasets that are commonly used to study the NLVD problem. In addition, we test on a smaller dataset that we collected to be representative of daily activities that are typical of service robotics scenarios.

Max Plank Institute for Informatics Movie Description Dataset (MPII-MD) This dataset [19] contains over 68 000 clips of average $4s$ each, from a corpus of 94 HD movie of different genres. Those clips are associated with sentences taken from the movie script and the transcribed Descriptive Video Service (DVS[1]) track. As a common practice, we use the training/validation/test split provided by the authors of the dataset, resulting in 56 816 training clips, 4930 validation clips and 6584 test clips. This split is the same typically used for NLVD systems [5, 9, 13, 14, 20, 21]. The vocabulary is obtained from the training corpus and consists of 7198 words.

The Microsoft Research Video Description Corpus (MSVD) This dataset [22] contains home-made 10-20s long videos from YouTube. The topics of the videos include sports, animals and music. We retain the 1970 clips that have English captions associated. The captions are on average 43 for each video and have been collected by the Amazon Mechanical Turk service. As the common practice [9, 10, 12, 13, 14, 21], we use the first 1200 videos for training, the next 100 video for validation and the last 670 video for testing. Note that each video-caption pair is considered as a unique sample, so the actual number of samples in each split is average 43 times the number of videos. Again, we construct the vocabulary from the training set and obtain a vocabulary of 4215 words.

Intelligent Systems, Automation and Robotics Laboratory Video Description Dataset (ISARLab-VD) For this work, we collect a relatively small dataset. Despite that, our dataset is still generic in terms of depicted actions, environment and involved actors. Note that, none of the above datasets have been conceived for service robotics applications. This was a major motivation for us to produce the dataset. It contains 100 videos which length varies from $5s$ to $30s$. Each video is paired with 5 manually obtained independent captions, for a total of 500 samples. The dataset features both high resolution and low resolution videos. In particular, the latter are obtained using the built-in camera of the COZMO toy

[1] Descriptive Video Service is an audio track associated to a movie to allow the visually impaired people to enjoy also the visual content of the movie.

robot by Anki[2] during the experimental phase of this study. In this work, we use the entire ISARLab-VD dataset for test only.

2.4.2 Evaluation Metrics Overview

In this work, we adopt classical natural language processing metrics for the evaluation of our method, which is a common practice in the NLVD research. These metrics are briefly described here for clarity and we refer to [23, 24, 25, 26] for further details. First note that a n-gram is a sequence of n consecutive words. When comparing a candidate sequence X and a reference sequence Y, the n-gram recall is the proportion of n-grams in Y that appear also in X, while the n-gram precision is the proportion of n-grams in X that appear also in Y.

The first metric we use is BLEU [23], in its *4*-gram variant. It is a precision-oriented metric designed for MT evaluation. Basically, it combines the n-gram precision for each n-gram up to length 4 and penalizes the difference in length between the candidate and the reference sentences. BLEU correlates well with human judgement on the quality of the translation if evaluated on the entire test corpus, but its correlation at sentence level is poor.

We also adopt another MT evaluation metric, namely METEOR [24]. It combines unigram precision and recall based on matching unigrams in the candidate and reference sentences. Unigrams can be matched in their exact form, stemmed form, and meaning. METEOR correlates well with human judgement also at sentence level.

The third metric we use is ROUGE [25] in its variant ROUGE_L, that considers the Longest Common Subsequence (LCS) of the candidate and the reference sentence. ROUGE is a recall-oriented metric designed for summarization evaluation following the idea that a good candidate summary overlaps a reference summary. Note that all ROUGE variants correlate well with human judgement.

Finally, we adopt a recently developed metric for assessing image description quality capturing human consensus on it, namely CIDEr [26]. It is based on the average cosine similarity between n-grams of different order (up to *4*-grams) and rewards length similarity between candidate and reference sentences. Cosine similarity allows taking into account both precision and recall. This metric correlates well with human judgement by design, thus is particularly suitable for the task of NLVD.

2.4.3 Baseline Methods Overview

We quantitatively compare our system to some of the State-of-the-Art techniques presented in Section 2.2, namely SA-GoogleNet+3D-ConvNet [21], S2VT [9], LSTM-YT [10], LSTM-E [12], HRNE [13] and BA-LSTM [14]. In addition, we compare to Venugopalan et al. [20] and to Rohrbach et al. [5]. SA-GoogleNet+3D-CNN applies an attention mechanism to select the most relevant video frames based on GoogLeNet [27] and 3D-CNN [28] extracted features, and an LSTM to generate the description sentence. S2VT uses a stacked LSTM encoder-decoder on the basis of ConvNet features extracted from each frame via VGG-16 [29]. LSTM-YT mean-pools each frame's AlexNet [30] ConvNet features and decodes this representation via a LSTM. LSTM-E learns an embedding based on the frame-level extracted mean-pooled VGG-19 [29] and C3D [17] ConvNet features and the video description, then generates a sentence via a LSTM. HRNE represents each video frame via GoogLeNet features and applies a hierarchical multi-layer LSTM encoder and a LSTM with soft-attention decoder. BA-LSTM is the most similar to our approach, but it uses LSTM blocks in the encoding phase. Venugopalan et al. [20] improves S2VT using a neural language model and distributional semantics learned from a large text corpus. Rohrbach et al. [5] uses CRFs to obtain tuples of verbs, objects and places on the basis of ConvNet features extracted from the video via pre-trained ConvNets, then translated the tuple into a sentence via a LSTM.

Differently from SA-GoogleNet+3D-CNN and HRNE, we do not apply any attention mechanism to deal with different-granularity time dependencies in the videos. As opposed to LSTM-YT and LSTM-E, we explicitly model the temporal dimension of the video sequence via the recurrent encoder. Finally, another major difference between our approach and the baselines is that we use a full-GRU architecture.

Note that, since BA-LSTM is the closest to our method, we used the same settings as the authors of [14] to better compare the two architectures. In particular, we set to 1024 the size of the inner state vectors and use the same size for input vectors, embeddings, weight matrices and bias vectors. Embedding

[2] https://www.anki.com/en-us/cozmo

Model	B_4	M	R_L	C
SA-GoogleNet+3D-CNN [21]	-	5.7	-	-
S2VT-RGB [9]	0.5	6.3	15.3	9.0
Venugopalan et al. [20]	-	6.8	-	-
Rohrbach et al. [5]	**0.8**	**7.0**	16.0	10.0
BA-LSTM [14]	**0.8**	**7.0**	**16.7**	10.8
BA-GRU (ours)	**0.8**	6.8	16.5	**11.7**

Table 2.1: Experiment results on the MPII-MD dataset in terms of the quantitative evaluation metrics BLEU in its *4*-gram variant (B_4), METEOR (M), ROUGE in its LCS variant (R_L) and CIDEr (C). Bold indicates the best performance.

Model	B_4	M	R_L	C
SA-GoogleNet+3D-CNN [21]	41.9	29.6	-	-
LSTM-YT [10]	33.3	29.1	-	-
S2VT [9]	-	29.8	-	-
LSTM-E [12]	45.3	31.0	-	-
HRNE [13]	**46.7**	**33.9**	-	-
BA-LSTM [14]	*41.5*	*31.3*	68.6	*55.5*
BA-GRU (ours)	42.5	32.0	**68.8**	**59.0**

Table 2.2: Experiment results on the MSVD dataset in terms of the quantitative evaluation metrics BLEU in its *4*-gram variant (B_4), METEOR (M), ROUGE in its LCS variant (R_L) and CIDEr (C). Bold indicates the best performance. Values in italic are obtained by re-running the code released by the authors of [14], which differ from those declared in their paper.

matrices and weight matrices applied to inputs are initialized via the Glorot normal initializer, those applied to inner states are initialized via the orthogonal initializer and the bias vectors are initialized to zero. We perform the training until the validation loss stops improving (or up to 100 epochs), with mini-batch size of 128. As optimizer, we apply Adadelta with learning rate $l_r = 1.0$, decay constant $\rho = 0.95$ and parameter $\epsilon = 10^{-8}$. The input and the output of the BA-GRU and the GRU in the encoding phase are regularized via Dropout with retain probability $p = 0.5$.

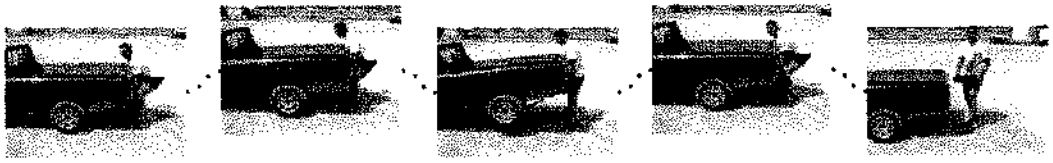
2.4.4 Results on the Standard Datasets

The performance is evaluated on the MPII-MD and MSVD datasets and expressed in terms of the widely used metrics presented in 2.4.2. For consistency sake with the baselines, we use the original COCO evaluation script[3].

The results are summarized in Tab. 2.1 for the MPII-MD dataset and in Tab. 2.2 for the MSVD dataset. It can be observed that our method is competitive with all the other approaches in terms of all the metrics. More importantly, it outperforms all the baselines in terms of the CIDEr metric, that has been reported in [26] best capturing human consensus on captions.

A direct human evaluation of the quality of the produced caption is usually more representative of the actual performance of NLVD algorithms, since video description is somewhat a tricky task to evaluate via a numerical metric. In this respect, we qualitatively evaluate our system and the most similar among the baselines *i.e.* BA-LSTM on the MPII-MD and MSVD datasets. Some examples of this comparison are reported in Fig. 2.3. Note that, for the MSVD dataset the reported ground truth description is the most representative of the multiple caption associated to the clips. From these examples, we can argue the cause of such low evaluation metrics values, especially for the MPII-MD dataset (Tab. 2.1). Indeed, in this dataset the ground truth was obtained from the script and the transcribed DVS track, aligned to the original movie. Hence, in many cases, the ground truth description is not limited to the visual content of the scene, but also takes into account the contextual events of the plot. NLVD systems instead can only rely on the visual information in the isolated clip. Conversely, the ground truth captions in the MSVD

[3]https://github.com/tylin/coco-caption



GT: A man is lifting a truck.
BA-LSTM: A man is riding a car.
BA-GRU: A man is lifting a car.

(a)

GT: News crew helicopters hover in the air above the scene.
BA-LSTM: The crowd is in the river.
BA-GRU: Someone looks at the crowd and turns to the ground.

(b)

Figure 2.3: Example results on a video from the MSVD test subset 2.3(a) and on a video from a movie in the MPII-MD test subset 2.3(b).

dataset are more precise and higher in number when compared to those of the MPII-MD dataset (∼40 versus 1-2). Thus, we can conclude that the performance of NLVD algorithms trained and tested on the MSVD dataset are both quantitatively and qualitatively better.

In addition, to gain some insights on the statistical significance of the presented quantitative results, we perform a K-fold cross-validation (with K= 10) of our approach and the BA-LSTM baseline on the MSVD dataset. We choose this dataset because it is smaller than the MPII-MD dataset, thus the model assessment experiment can be run in less time. The resulting values for the evaluation metrics, expressed in terms of mean and standard deviation, are reported in Tab. 2.3. It is observed that our method is still comparable to the BA-LSTM baseline.

We also evaluate the training and testing time of the ten different variants of both BA-GRU and BA-LSTM. In particular, for BA-GRU the test time is on average 190.89 ± 5.28 ms, while for BA-LSTM is on average 197.78 ± 3.70 ms. In terms of training time, for BA-GRU it is on average $\sim 8h21' \pm \sim 5h34'$, while for BA-LSTM it is on average $\sim 13h40' \pm \sim 3h22'$. Despite both the BA-GRU and the BA-LSTM require much time to complete the training phase, saving 5 hours could make the difference during the deployment of the architecture in a real robotic application. This is the case, for instance, of parameter tuning procedures, where many different models are trained and evaluated to select the best network configuration and, thus, using the BA-GRU makes a huge difference with respect to the overall time.

The GRU block has fewer parameters than the LSTM block. In particular, our method BA-GRU requires approximately $114MB$ of memory to store network weights, while the BA-LSTM needs $128MB$. Another benefit of using fewer parameters is that it reduces the risk of overfitting and, potentially, it allows the model to better generalize on completely new datasets.

2.4.5 Results on the ISARLab-VD Datasets

We further evaluate and compare BA-GRU with BA-LSTM on our collected dataset. Note that, in this case the algorithms are not trained on any subset of the ISARLab-VD dataset. With this experiment we want to test the generalization capabilities of the two architectures. We report the results of both

Model	B_4	M	R_L	C
BA-LSTM	41.5±1.0	31.4±0.3	68.5±0.5	56.1±2.0
BA-GRU	41.1±1.1	31.2±0.7	68.3±0.5	53.5±3.8

Table 2.3: Experiment results of the K-fold cross-validation on the MSVD dataset in terms of the quantitative evaluation metrics BLEU in its 4-gram variant (B_4), METEOR (M), ROUGE in its LCS variant (R_L) and CIDEr (C). The results are expressed in terms of mean and standard deviation.

Model	B_4	M	R_L	C
BA-LSTM on MSVD	14.0	19.5	51.6	23.3
BA-GRU on MSVD	**14.7**	**20.0**	**52.8**	**27.7**
BA-LSTM on MPII-MD	00.0	08.4	18.2	06.9
BA-GRU on MPII-MD	00.0	12.1	20.2	10.6

Table 2.4: Experiment results on the ISARLab-VD dataset in terms of the quantitative evaluation metrics BLEU in its 4-gram variant (B_4), METEOR (M), ROUGE in its LCS variant (R_L) and CIDEr (C). Bold indicates the best performance.

the BA-GRU and BA-LSTM architectures trained on either the MPII-MD and MSVD datasets, both in quantitative and qualitative terms.

In particular, in Tab. 2.4 we report the results in terms of the previously defined evaluation metrics. For the statistical significance of those results, we refer to Tab. 2.5. There we also report the results of the ten variants of the BA-GRU and BA-LSTM models obtained via K-fold cross-validation on the MSVD dataset.

Some exemplar cases of the qualitative evaluation on this dataset are reported in Fig. 2.4 with examples on high resolution and low resolution videos. The reported ground truth description is the most representative of the multiple caption associated to the clips. We refer to the complete results corpus available at `http://isar.unipg.it/index.php?option=com_content&view=article&id=46&catid=26&Itemid=188` for further examples. It can be observed that the quality of the videos does not influence the semantic and syntactic correctness of the description produced by the two methods. On the other hand, we observe that the captions for the videos of the ISARLab-VD dataset are simpler and less precise than those produced for the test subset videos of the public dataset used for the training. This suggests that these NLVD systems do not generalize well with respect to scenarios that significantly differ from those observed in training phase. Despite that, we can observe that the use of the BA-GRU gives a slight performance improvement. This suggests that the BA-GRU could be better suited to achieve architecture more robust to domain changes. The exploration of this aspect is beyond the scope of this work, but this insights could be definitely useful for future investigations.

2.5 Conclusions and Future Developments

This work focuses on the NLVD task and presents a full-GRU encoder-decoder architecture to address it. We show that the proposed approach is faster to train and less memory consuming that other State-of-the-Art algorithms. Our method is also competitive or superior in terms of performance on the public datasets which were partially used also for training. The experimental results on the devised dataset we use only

Model	B_4	M	R_L	C
BA-LSTM	14.2±0.8	19.0±0.3	50.8±0.7	25.2±4.0
BA-GRU	15.0±1.0	19.4±0.5	51.2±0.8	24.7±2.6

Table 2.5: Experiment results of the ten variants of the BA-GRU and BA-LSTM models obtained via K-fold cross-validation on the MSVD dataset in terms of the quantitative evaluation metrics BLEU in its 4-gram variant (B_4), METEOR (M), ROUGE in its LCS variant (R_L) and CIDEr (C). The results are expressed in terms of mean and standard deviation.

GT: A man is walking in a office corridor.
BA-LSTM (MSVD): A man is jumping.
BA-GRU (MSVD): A man is running on a wall.
BA-LSTM (MPII-MD): Two man walks up.
BA-GRU (MPII-MD): Man opens the door and walks out of the office.

(a)

GT: Someone is driving a car.
BA-LSTM (MSVD): A car is driving down the road.
BA-GRU (MSVD): A man is driving a car.
BA-LSTM (MPII-MD): Two car pulls up.
BA-GRU (MPII-MD): Car pulls u the street and runs out of the car.

(b)

GT: A man is playing a guitar.
BA-LSTM (MSVD): A man is playing a guitar.
BA-GRU (MSVD): A man is playing a guitar.
BA-LSTM (MPII-MD): Two man is a gun.
BA-GRU (MPII-MD): Sound of the man is in the middle of the window.

(c)

Figure 2.4: Example results on videos from the ISARLab-VD dataset. In particular, 2.4(a) and 2.4(b) refer to videos that have been collected with two different hig resolution cameras, while 2.4(c) refers to a low resolution video collected during the experiments with the Anki's COZMO robot.

for test demonstrate that our proposed BA-GRU architecture can generalize better than the BA-LSTM baseline. This is a pivotal aspect for applications to service robotics, where the training phase has to be absent or at least very short.

In a lifelong application the robot will likely collect a continuous video stream. This means that the videos it will describe will be longer than those in the datasets we used both for training and test. This drawback can be overcome by cutting the continuous video sequence in shorter chunks and describing each chunk using our proposed method as it is. However, being able to deal with much longer videos is surely of great interest and the development of effective solutions to this problem will be the subject of future work.

Finally, note that the training strategy for our method is based on maximum likelihood estimation. This encourages the syntactic and semantic correctness of the produced descriptions and allows the robot to accomplish the NLVD task and be helpful to the end user. However, diversity is not taken into account. Providing the capability to generate diverse descriptions for the same video sequence could be an interesting extension of our proposed approach. This could be accomplished by exploiting different learning strategies, as for example adversarial learning.

The code and the dataset used for this study are publicly available at `http://isar.unipg.it/index.php?option=com_content&view=article&id=46&catid=2&Itemid=188`.

Chapter 3

LS-VO: Learning Optical Subspace for Robust VO Estimation

3.1 Introduction

Learning based Visual Odometry (L-VO) in the last few years has seen an increasing attention of the robotics community because of its desirable properties of robustness to image noise and camera calibration independence [31], mostly thanks to Convolutional Neural Networks (CNNs) representational power, which can complement current geometric solutions [32]. While current results are very promising, making these solutions easily applicable to different environments still presents challenges. One of them is that most of the approaches so far explored have not shown strong domain independence and suffer from high dataset bias, i.e. the performances considerably degrade when tested on sequences with motion dynamics and scene depth significantly different from the training data [33]. In the context of L-VO this bias is expressed in different Optical Flow (OF) field distribution in training and test data, due to differences in scene depth and general motion of the camera sensor.

One possible explanation for the poor performances of learned methods on unseen contexts is that most current learning architectures try to extract both visual features and motion estimate as a single training problem, coupling the appearance and scene depth with the actual camera motion information contained in the OF input. Some works have addressed the problem with an unsupervised, or semi-supervised approach, trying to learn directly the motion representation and scene depth from some kind of frame-to-frame photometric error [34] [35] [36]. While very promising, these approaches are mainly devised for scene depth estimation and still fall short in terms of general performances on Ego-Motion estimation.

At the same time, previous research has shown how OF fields have a bilinear dependence on motion and inverse scene depth [37]. We suggest that this is the main reason for the low generalization properties shown by learned algorithms so far. Past research has shown that the high dimensional OF field, when scene depth can be considered locally constant, can be projected on a much lower dimensional linear space [38] [39]. However, when these conditions do not hold, the OF field subspace exists but is highly non-linear.

In this work we propose to exploit this knowledge, estimating the latent OF representation using an Auto-Encoder (AE) Neural Network architecture as a non-linear subspace approximator. AE networks are able to extract latent variable representation of high dimensional inputs. Since our aim is to make the Ego-Motion estimation more robust to OF fields that show high variability in their distribution, we do not simply use this subspace to directly produce motion prediction. Instead, we propose a novel architecture that jointly trains the subspace estimation and Ego-Motion estimation so that the two network tasks are mutually reinforcing and at the same time able to better generalize OF field representation. The conceptual architecture is shown in Figure 3.1. To demonstrate the increased performances and reduced dataset bias with respect to high dynamical variation of the OF field, we test the proposed approach on a challenging scenario. We sub-sample the datasets, producing sequences that simulate high speed variations, then we train and test on sequences that are both different in appearance and sub-sampling rate.

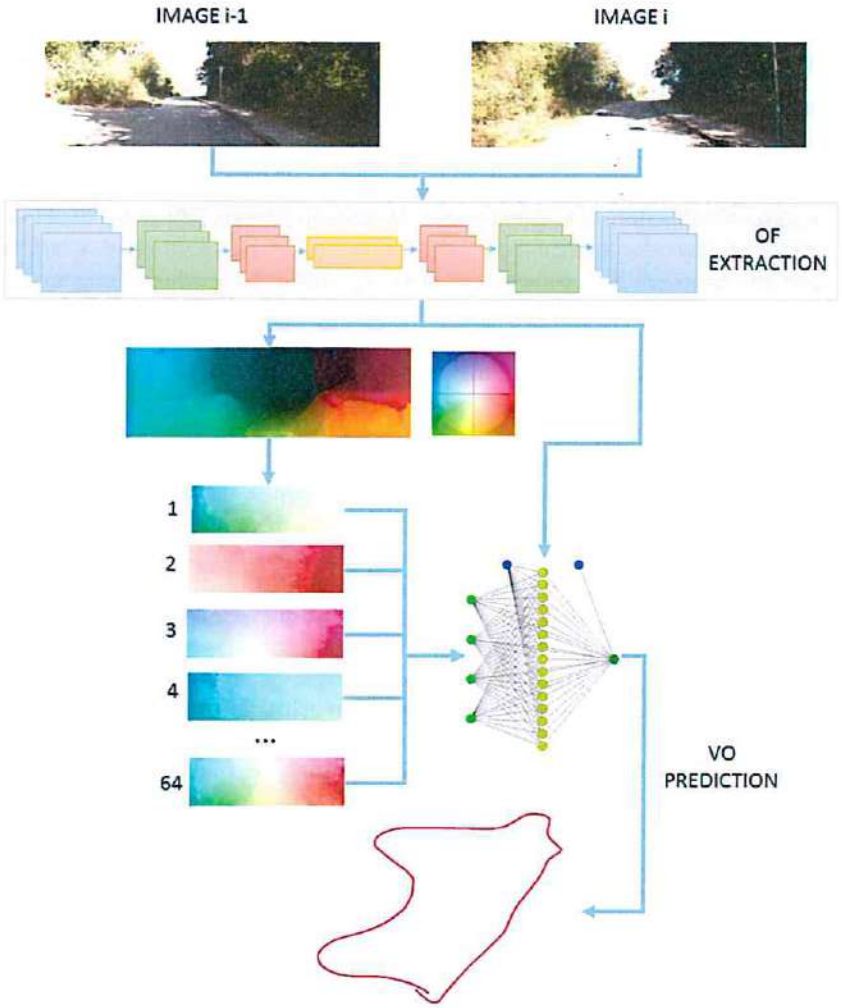


Figure 3.1: Overview of the method: We propose a network architecture that jointly learn a latent space representation of the Optical Flow field and estimates motion. The joint learning makes the estimation more robust to input domain changes. The latent representation is an input to the estimation network together with the lower level features.

3.2 Related Works

3.2.1 Ego-Motion estimation

Geometric Visual Odometry

G-VO has a long history of solutions. While the first approaches were based on sparse feature tracking, mainly for computational reasons, nowadays direct or semi-direct approaches are preferred. These approaches use the photometric error as an optimization objective. Research on this topic is very active. Engel at al. developed one of the most succesful direct approaches, LSD SLAM, both for monocular and stereoscopic cameras [40], [41]. Forster et al. developed the Semi-Direct VO (SVO) [42] and its more recent update [43], which is a direct method but tracks only a subset of features on the image and runs at very high frame rate compared to full direct methods. Even if direct methods have gained most of the attention in the last few years, the ORB-SLAM algorithm by Mur-Artal et al. [44] reverted to sparse feature tracking and reached impressive robustness and accuracy comparable with direct approaches.

Learned Visual Odometry

Learned approaches go back to the early explorations by Roberts et al. [38, 45], Guizilini et al. [46, 47], and Ciarfuglia et al. [48]. As for the geometric case, the initial proposal focused on sparse OF features that, faithful to the *there's no free lunch theorem*, explored the performances of different learning algorithms such as SVMs, Gaussian Processes and others. While these early approaches already showed some of the strengths of L-VO, it was only more recently, when Costante et al. [31] introduced the use of CNNs for feature extraction from dense optical flow, that the learned methods started to attract more interest. Since then a couple of methods have been proposed. Muller and Savakis [49] added the FlowNet architecture to the estimation network, producing one of the first end-to-end approaches. Clark et al. [50] proposed an end-to-end approach that merged camera inputs with IMU readings using an LSTM network. Through this sensor fusion, the resulting algorithm is able to give good results but requires sensors other than a single monocular camera. The use of LSTM is further explored by Wang et al. in [51], this time without any sensor fusion. The resulting architecture gives again good performances on KITTI sequences but does not show any experiments on environments with different appearance from the training sequences. On a different track is the work of Pillai et al. [52], that, like [47], looked at the problem as a generative probabilistic problem. Pillai proposes an architecture based on an MDN network and a Variational Auto-Encoder (VAE) to estimate the motion density given the OF inputs as a GMM. While Frame to Frame (F2F) performances are on a par with other approaches, they also introduce a loss term on the whole trajectory that mimics the bundle optimization that is often used in G-VO. The results of the complete system are thus very good. However, they use as input sparse KLT optical flow, since the joint density estimation for dense OF would become computationally intractable, meaning that they could be more prone to OF noise than dense methods.

Most of the described approaches claim independence from camera parameters. While this is true, we note that this is more an intrinsic feature of the learning approach than the merit of a particular architecture. The learned model implicitly learns also the camera parameters, but then it fails on images collected with other camera optics. This parameter generalization issue remains an open problem for L-VO.

3.2.2 Semi-supervised Approaches

Since dataset bias and domain independence are critical challenges for L-VO, it is not surprising that a number of unsupervised and semi-supervised methods have been recently proposed. However, all the architectures have been proposed as a way of solving the more general problem of joint scene depth and motion estimation, and motion estimation is considered more as a way of improving depth estimation. Konda and Memisevich [53] used a stereo pair to learn VO but the architecture was conceived only for stereo cameras. Ummenhofer and Zhou [34] propose the DeMoN architecture, a solution for F2F Structure from Motion (SfM) that trains a network end-to-end on image pairs, levering motion parallax. Zhou et al. [35] proposed an end-to-end unsupervised system based on a loss that minimizes image warping error from one frame to the next. A similar approach is used by Vijayanarasimhan et al. [36] with their SfM-Net. All these approaches are devised mainly for depth estimation and the authors give little or no attention to the performances on VO tasks. Nonetheless, the semi-supervised approach is one of the more relevant future directions for achieving domain independence for L-VO, and we expect that this approach will be integrated in the current research on this topic.

3.2.3 Optical Flow Latent Space Estimation

The semi-supervised approaches described in Section 3.2.2 make evident an intrinsic aspect of monocular camera motion estimation, that is, even when the scene is static, the OF field depends both on camera motion and scene depth. This relationship between inverse depth and motion is bilinear and well known [54] and is at the root of scale ambiguity in monocular VO. However, locally and under certain hypothesis of depth regularity, it is possible to express the OF field in terms of a linear subspace of OF basis vectors. Roberts et al. [45] used Probabilistic-PCA to learn a lower dimensional dense OF subspace without supervision, then used it to compute dense OF templates starting from sparse optical flow. They then used it to compute Ego-Motion. Herdtweck and Cristóbal extended the result and used Expert Systems to estimate motion [55]. More recently, a similar approach to OF field computation was proposed by Wulff

and Black [39] that complemented the PCA with MRF, while Ochs et al. [56] did the same by including prior knowledge with an MAP approach. These methods suggest that OF field, which is an intrinsically high dimensional space generated from a non-linear process, lies on an ideal lower dimensional manifold that sometimes can be linearly locally approximated. However, modern deep networks are able to find latent representation of high dimensional image inputs, and in this work we use this intuition to explore this OF latent space estimation.

3.3 Contribution

Inspired by the early work of Roberts on OF subspaces [37], and by recent advances in deep latent space learning [57], we propose a network architecture that jointly estimates a low dimensional representation of dense OF field using an Auto-Encoder (AE) and at the same time computes the camera Ego-Motion estimate with a standard Convolutional network, as in [31]. The two networks share the feature representation in the decoder part of the AE, and this constrains the training process to learn features that are compatible with a general latent subspace. We show through experiments that this joint training increases the Ego-Motion estimation performances and generalization properties. In particular, we show that learning the latent space and concatenating it to the feature vector makes the resulting estimation considerably more robust to domain change, both in appearance and in OF field dynamical range and distribution.

We train our network both in an end-to-end version, using deep OF estimation, and with standard OF field input, in order to explore the relative advantages and weaknesses. We show that while the end-to-end approach is more general, precomputed OF still has some performance advantages.

In summary our contributions are:

- A novel end-to-end architecture to jointly learn the OF latent space and camera Ego-Motion estimation is proposed. We call this architecture Latent Space-VO (LS-VO).
- The strength of the proposed architecture is demonstrated experimentally, both for appearance changes, blur, and large camera speed changes.
- Effects of geometrically computed OF fields are compared to end-to-end architectures in all cases.
- The adaptability of the proposed approach to other end-to-end architectures is demonstrated, without increasing the chances of overfitting them, due to parameters increase.

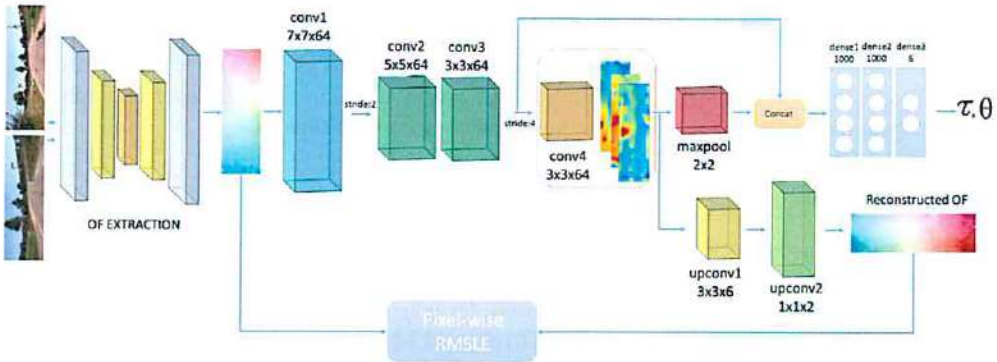


Figure 3.2: LS-VO network architecture. The shared part is composed of Flownet OF extraction, then three convolutional layers that start the feature extraction. The last layer of the Encoder, conv4, is not shared with the Estimator network. From conv4 the latent variables are produced. The Decoder network takes these variables and reconstructs the input, while the Estimator concatenates them to conv3 output. Then three fully connected layers produce the motion estimates.

3.4 Learning Optical Flow Subspaces

Given an optical flow vector $\mathbf{u} = (\mathbf{u}_x^\intercal, \mathbf{u}_y^\intercal)^\intercal$ from a given OF field $\mathbf{x}$, [37] [39] approximate it with a linear relationship:

$$\mathbf{u} \approx \mathbf{W}\mathbf{z} = \sum_{i=1}^{l} z_i \mathbf{w}_i \tag{3.1}$$

where the columns of $\mathbf{W}$ are the basis vectors that form the OF linear subspace and $\mathbf{z}$ is a vector of latent variables. This approximation is valid only if there are some regularities of scene depth and is applicable only to local patches in the image. The real subspace is non-linear in nature and, in this work, we express it as a generic function $\mathbf{u} = \mathcal{D}(\mathbf{z})$ that we learn from data by using the architecture described in the following.

3.4.1 Latent Space Estimation with Auto-Encoder Networks

Let $\mathbf{y} \in \mathbb{R}^6$ be the camera motion vector and $\mathbf{x} \in \mathbb{R}^{2\times w\times h}$ the input OF field, computed with some dense method, where $\mathbf{x}_{(i,j)} = \mathbf{u}_{(i,j)}$ is a 2-dimensional vector of the field at image coordinates (i, j). Both can be viewed as random variables with their own distributions. In particular, we make the hypothesis that the input images lie on a lower dimensional manifold, as in [58], and thus also the OF field lies on a lower dimensional space $\mathbb{O} \subset \mathbb{R}^{2\times w\times h}$ with a distance function $S(\mathbf{x}^{(a)}, \mathbf{x}^{(b)})$, where $\mathbf{x}^{(a)}, \mathbf{x}^{(b)} \in \mathbb{O}$. The true manifold is very difficult to compute, so we look for an estimate $\hat{\mathbb{O}} \approx \mathbb{O}$ using the model extracted by an encoding neural network.

Let $\mathbf{z} \in \mathbb{Y} \subset \mathbb{R}^l, l \ll w \times h$ be a vector of latent random variables that encodes the variabilities of OF field that lies on this approximate space. The decoder part of the AE can be seen as a function

$$\mathcal{D}(\mathbf{z}; \theta_d) = D(\mathbf{z}; \{\mathbf{W}_k, \mathbf{b}_k\}, k = 1 \cdots K) \tag{3.2}$$

where $\theta_d = (\{\mathbf{W}_k, \mathbf{b}_k\}, k = 1 \cdots K)$ is the set of learnable parameters of the network (with K upconv layers), that is able to generate a dense optical flow from a vector of latent variables $\mathbf{z}$. Note that the AE works similarly to a non-linear version of PCA [57]. We define the set $\hat{\mathbb{O}} = \{D(\mathbf{z}; \theta_d) \mid \mathbf{z} \in \mathbb{Y}\}$ as our approximation of the OF field manifold and use the logaritmic Euclidean distance (as described in Section 3.4.2 as a loss function) as an approximation of $S(D(\mathbf{z}^{(a)}), D(\mathbf{z}^{(b)}))$. Using this framework the problem of estimating the latent space is carried out by the AE network, where the Encoder part can be defined as the function $\mathbf{z} = E(\mathbf{x}; \theta_e)$.

While in [52] the AE is used to estimate motion, and $\mathbf{z}$ are the camera translation and rotations, here we follow a different strategy. We compute the latent space for a two-fold purpose: we use the latent variables as an input feature to the motion estimation network and we learn this latent space together with the estimator, thus forcing the estimator to learn features compatible with the encoder representation. Together these two aspects make the representation more robust to domain changes.

3.4.2 Network Architecture

The LS-VO network architecture in its end-to-end form is shown in Figure 3.2. It is composed of two main branches, one is the AE network and the other is the convolutional network that computes the regression of motion vector $\mathbf{y}$. The OF extraction section is Flownet [59], for which we use the pre-trained weights. We run tests fine-tuning this part of the network on KITTI [60] and Malaga [61] datasets, but the result was a degraded performance due to overfitting.

The next layers are convolutions that extract features from the computed OF field. After the first convolutional layers (conv1, conv2 and conv3), the network splits into the AE network and the estimation network. The two branches share part of the feature extraction convolutions, so the entire network is constrained in learning a general representation that is good for estimation and latent variable extraction. The Encoder is completed by another convolutional layer, that brings the input $\mathbf{x}$ to the desired representation $\mathbf{z}$, and its output is fed both in the Decoder and concatenated to the feature extracted before. The resulting feature vector, composed of latent variables and convolutional features is fed into a fully connected network that performs motion estimation. The details are summarized in Table 3.1.

	Layer name	Kernel size	Stride	output size
Input	-	-	-	$(94, 300, 2)$
	LS-VO			
Shared Features Layer	conv1	7×7	2×2	$(47, 150, 64)$
	conv2	5×5	1×1	$(47, 150, 64)$
	conv3 $\star$	3×3	4×4	$(12, 38, 64)$
Auto-Encoder	conv4	3×3	1×1	$(12, 38, 64)$
	upconv1	3×3	1×1	$(48, 152, 6)$
	crop	-	-	$(47, 150, 6)$
	upconv2	1×1	1×1	$(94, 300, 2)$
Estimator	maxpool $\dagger$	2×2	2×2	$(6, 19, 64)$
	concat $\star$ and $\dagger$	-	-	(36480)
	dense1	-	-	(1000)
	dense2	-	-	(1000)
	dense3	-	-	(6)
	ST-VO			
Feature Extraction	st-conv1	3×3	2×2	$(46, 149, 64)$
	st-maxpool1 $\bullet$	4×4	4×4	$(11, 37, 64)$
	st-conv2	4×4	4×4	$(9, 35, 20)$
	st-maxpool2 $\diamond$	2×2	2×2	$(4, 17, 20)$
Estimation	concat $\bullet$ and $\diamond$	-	-	(27408)
	st-dense1	-	-	(1000)
	st-dense2	-	-	(6)

Table 3.1: LS-VO and ST-VO network architectures

The AE is trained with a pixel-wise squared Root Mean Squared Log Error (RMSLE) loss:

$$\mathcal{L}_{AE} = \sum_i || \log(\hat{\mathbf{u}}^{(i)} + 1) - \log(\mathbf{u}^{(i)} + 1)||_2^2 \tag{3.3}$$

where $\hat{\mathbf{u}}^{(i)}$ is the predicted OF vector for the i-th pixel, and $\mathbf{u}^{(i)}$ is the corresponding input to the network, and the logarithm is intended as an element-wise operation. This loss penalizes the ratio difference, and not the absolute value difference of the estimated OF compared to the real one, so that the flow vectors of distant points are taken into account and not smoothed off.

We use the loss introduced by Kendall et al. in [62]:

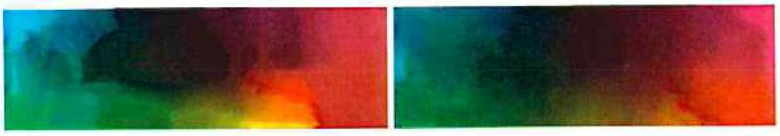
$$\mathcal{L}_{EM} = \sum_i ||\hat{\tau} - \tau||_2^2 + \beta||\hat{\theta} - \theta||_2^2 \tag{3.4}$$

where the τ is camera translation vector in meters, θ is the rotation vector in Euler notation in radians, and β is a scale factor that balances the angular and translational errors. β has been cross-validated on the trajectory reconstruction error ($\beta = 20$ for our experiments), so that the frame to frame error propagation to the whole trajectory is taken into account. The use of a Euclidean loss with Euler angle representation works well in the case of autonomous cars, since the yaw angle is the only one with significant changes. For more general cases, is better to use a quaternion distance metric [63].

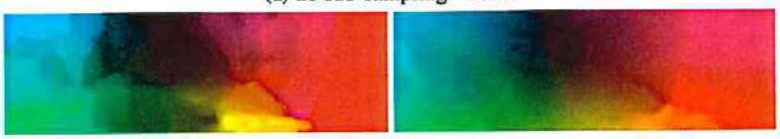
In Section 3.5, we compare this architecture both with SotA geometrical and learned methods. The baseline for the learned approaches is a Single Task (ST) network, similar to the 1b network presented in [31], and described in Table 3.1.

3.4.3 OF field distribution

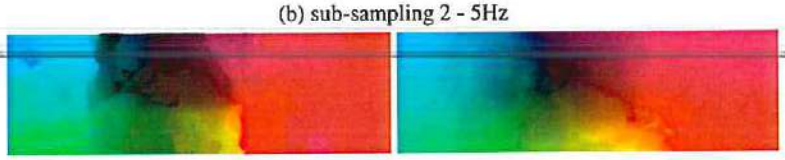
As mentioned in Section 3.4.1, the OF field has a probability distribution that lies on a manifold with lower dimensionality than the number of pixels of the image. We can argue that the actual density depends on the motion of the camera as much as the scene depth of the images collected. In this work, we test generalization properties of the network for both aspects:



(a) no sub-sampling - 10Hz



(b) sub-sampling 2 - 5Hz



(c) sub-sampling 3 - 3.33Hz

Figure 3.3: Examples of the OF field intensity due to different sub-sampling rates of the original sequences. In the left are the OF field extracted with Brox algorithm (BF) [64], while on the right the ones extracted with Flownet [59]. While the BF fields look more crisp, they require parameter tuning, while the Flownet version is non-parametric at test time.

(a) Standard and blurred image

(b) Standard and blurred OF field

Figure 3.4: Examples of OF fields obtained applying gaussian blur to image sequences. (a) The image and its blurred variant is shown, with blur radius 10. (b) The corresponding OF fields. Note the huge change in OF distribution.

i For the appearance we use the standard approach to test on completely different sequences than the ones used in training.

ii For the motion dynamics, we sub-sample the sequences, thus multiplying the OF dynamics by the same factor.

iii To further test OF distribution robustness, we also test the architecture on downsampled blurred images, as in [31].

Examples of the resulting OF field are shown in Figure 3.3, while an example of a blurred OF field is shown in Figure 3.4. In both images there are evident differences both in hue and saturation, meaning that both modulus and phase of the OF vectors change.

3.5 Experimental Results

3.5.1 Data and Experiments set-up

We perform experiments on two different datasets, the KITTI Visual Odometry benchmark [60] and the Malaga 2013 dataset [61]. Both datasets are taken from cars that travel in city suburbs and countryside, however the illumination conditions and camera setups are different. For the KITTI dataset we used the

	VISO2-M [65]		ORBSLAM2-M [44]		ST-VO (Flow)		ST-VO (BF)		LS-VO (Flow)		LS-VO (BF)	
	Trasl.	Rot.	Trasl.	Rot.	Trasl.	Rot.	Trasl.	Rot.	Trasl.	Rot.	Trasl.	Rot.
KITTI $d1$	18.13%	0.0193	62.71%	0.0058	12.73%	0.0507	8.06%	0.0205	10.71%	0.0290	6.98%	0.0199
KITTI $d2$	19.08%	0.0090	fail	fail	12.30%	0.0383	9.43%	0.0360	10.85%	0.0320	7.71%	0.0205
KITTI $d2$ + blur	52.54%	0.0688	fail	fail	16.35%	0.0502	16.39%	0.0627	14.47%	0.0375	8.13%	0.02710

Table 3.2: Performances summary of all methods on the Kitti experiments. The geometrical methods perform better on the angular rate estimation (in deg/m) on both datasets at standard rate, but usually fail on others (loss of tracking). Learned methods are consistent in their behaviour in all cases: even if the general error increases, they never fail to give an output even in the worst conditions tested, and the trajectories are always meaningful.

sequences 00 to 07 for training and the 08, 09 and 10 for test, as is common practice. The images are all around 1240×350, and we resize them to 300×94. The frame rate is 10Hz. For the Malaga dataset we use the sequences 02, 03 and 09 as test set, and the 01, 04, 06, 07, 08, 10 and 11 as training set. In this case the images are 1024×768 that we resize to 224×170. The frame rate is 20Hz. For the Malaga dataset there is no high precision GPS ground truth, so we use the ORBSLAM2 stereo VO [44] as a Ground truth, since its performances, comprising bundle adjustment and loop closing, are much higher than any monocular method.

The networks are implemented in Keras/Tensorflow and trained using an Nvidia Titan Xp. Training of the ST-VO variant takes $6h$, while LS-VO $27h$. The ST-VO memory occupancy is on average 460MB, while LS-VO requires 600MB. At test time, computing Flownet and BF features takes on average 12.5ms and 1 ms per sample, while the prediction requires, on average, 2 – 3ms for both ST-VO and LS-VO. The total time, when considering Flownet features, amounts to 14.5ms for ST-VO and 15.5ms for LS-VO. Hence, we can observe that the increased complexity does not affect much computational performance at test time.

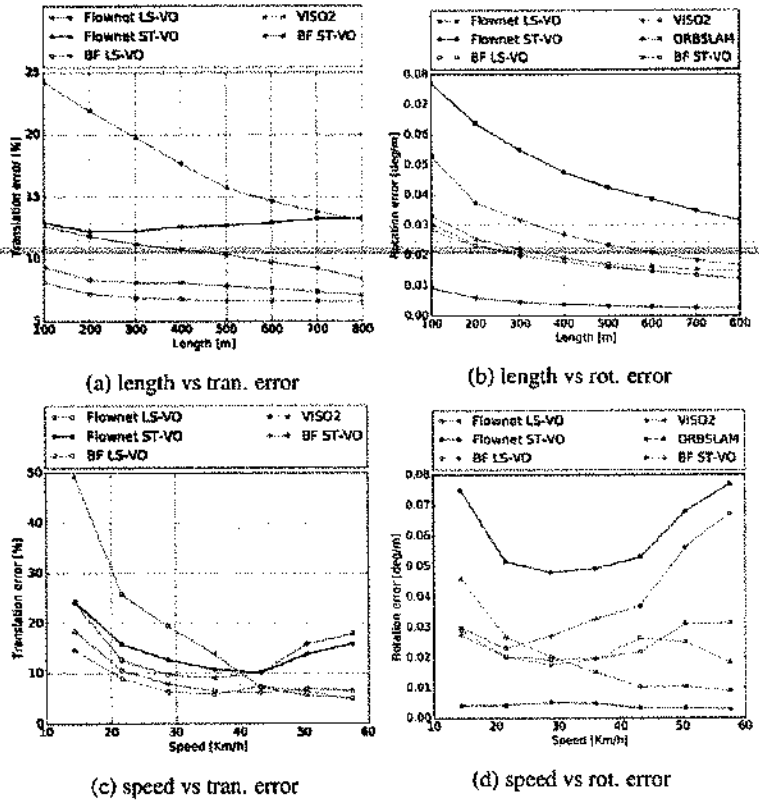
For all the experiments described in the following Section, we tested the LS-VO architecture and the ST-VO baseline. Furthermore, on all KITTI experiments we tested with both Flownet and BF features. While the contribution of this work relates mainly on showing the increased robustness of the proposed method with respect to learned architectures, we also sampled the performances of SotA geometrical methods, namely VISO2-M [65] and ORBSLAM2-M [44] in order to have a general baseline.

3.5.2 Experiments

As mentioned in Section 3.4.3, on both datasets we perform three kinds of experiments, of increasing difficulty. We observe that the original sequences show some variability in speed, since the car travels in both datasets at speeds of up to 60Km/h, but the distribution of OF field is still limited. This implies that the possible combinations of linear and rotational speeds are limited. We extend the variability of OF field distribution performing some data augmentation. Firstly, we sub-sample the sequences by 2 and 3 times, to generate virtual sequences that have OF vectors with very different intensity. In Figure 3.3, an example of the different dynamics is shown. In both KITTI and Malaga datasets we indicate the standard sequences by the $d1$ subscript, and the sequences sub-sampled by 2 and 3 times by $d2$ and $d3$, respectively. In addition to this, we generate blurred versions of the $d2$ test sequences, with gaussian blur, as in [31]. Then we perform three kinds of experiment and compare the results. The first is a standard training and test on $d1$ sequences. This kind of test explores the generalization properties on appearance changes alone. In the second kind of experiment we train all the networks on the sequences $d1$ and $d3$ and test on $d2$. This helps us to understand how the networks perform when both appearance and OF dynamics change. The third experiment is training on $d1$ and $d3$ sequences, and testing on the on the blurred versions of the $d2$ test set (Figure 3.4).

The proposed architecture is end-to-end, since it computes the OF field through a Flownet network. However, as a baseline, we decided to test the performances of all the architecture on a standard geometrical OF input, computed as in [64], and indicated as BF in the following.

In addition, we train the BF version on the RGB representation of OF, since from our experiments performs slightly better than the floating point one.



(a) length vs tran. error

(b) length vs rot. error

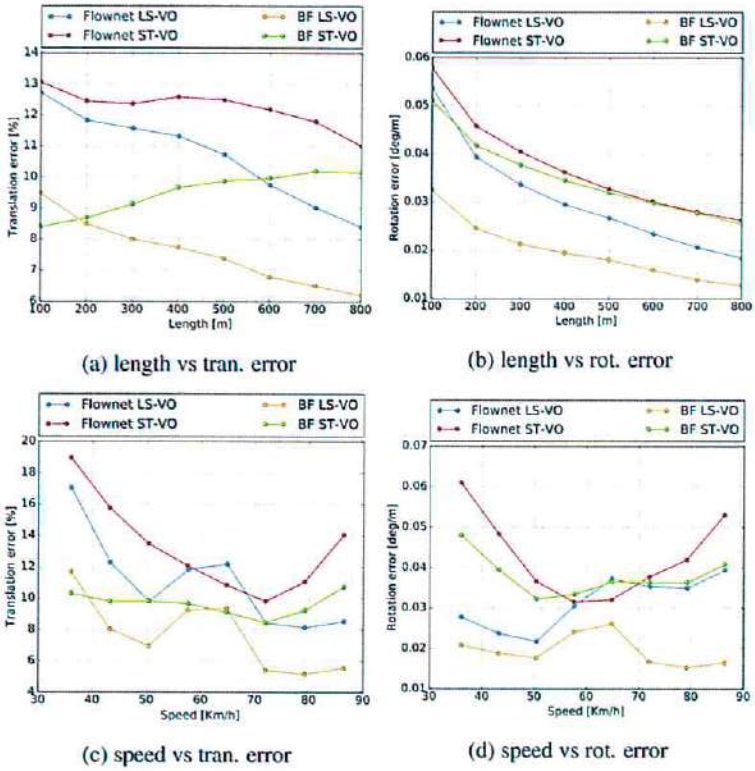
(c) speed vs tran. error

(d) speed vs rot. error

Figure 3.5: Comparison between all methods on KITTI dataset, with no sequence sub-sampling. It is evident that the LS-VO network outperforms the ST equivalent, and in the case of the BF OF inputs it is almost always better by a large margin. Geometrical methods outperform learned ones on angular rate. ORBSLAM2-M is not shown in 3.5(a) and 3.5(b) for axis clarity, since the error is greater than other methods.

	VISO2-M [65]		ORBSLAM2-M [44]		ST-VO (Flow)		LS-VO (Flow)	
	Trasl.	Rot.	Trasl.	Rot.	Trasl.	Rot.	Trasl.	Rot.
Malaga *d*1	43.90%	0.0321	86.60%	**0.0156**	23.20%	0.1241	**15.56%**	0.0690
Malaga *d*2	47.37%	0.0530	fail	fail	23.35%	0.1088	**21.44%**	**0.0472**
Malaga *d*2 + blur	fail	fail	fail	fail	25.14%	0.1262	**24.06%**	**0.0657**

Table 3.3: Performances summary of all methods on the Malaga experiments. The same considerations of Table 3.2 apply. In this set of experiments we analysed only the end-to-end architecture, for the sake of simplicity.



(a) length vs tran. error (b) length vs rot. error

(c) speed vs tran. error (d) speed vs rot. error

Figure 3.6: Comparison between the four network architectures on KITTI *d2* dataset. Again, the LS-VO architecture outperforms the other, except for speed around 60Km/h.

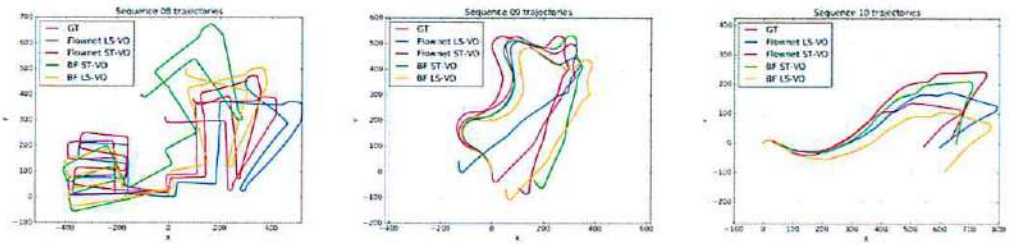
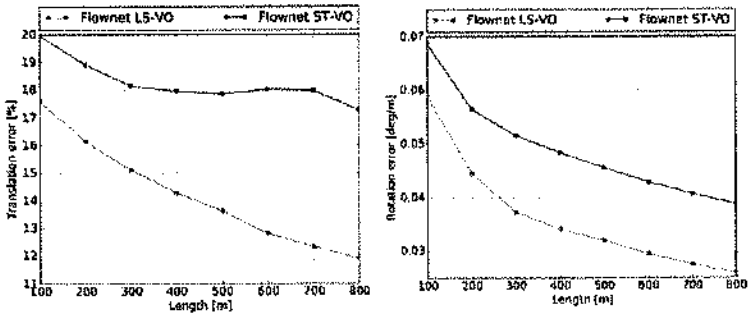
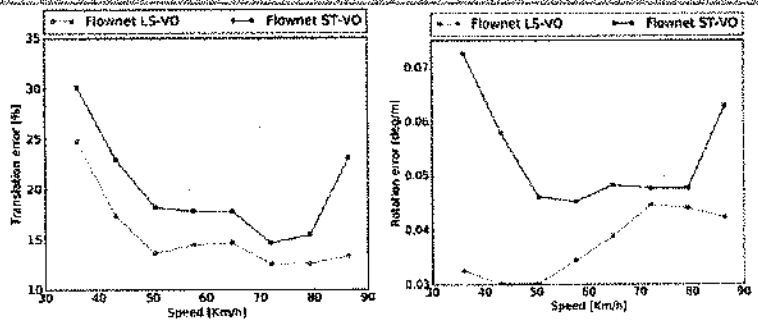


Figure 3.7: KITTI *d2* trajectories: Trajectories computed on the sub-sampled sequences for all architectures (*d2* - 5Hz).

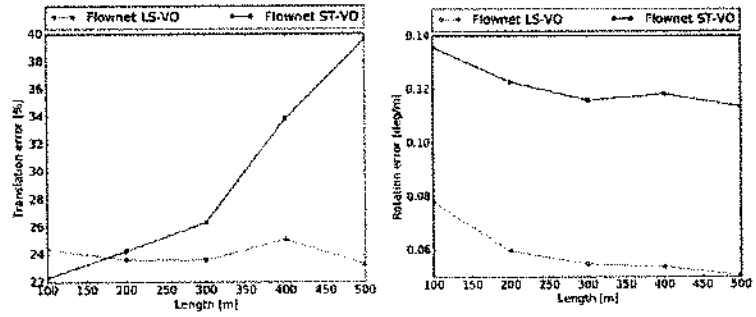


(a) length vs tran. error (b) length vs rot. error

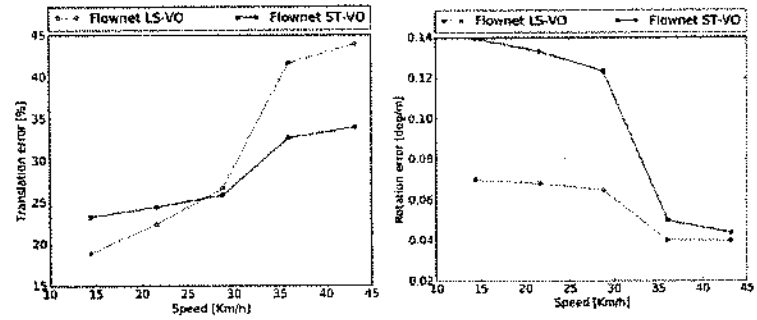


(c) speed vs tran. error (d) speed vs rot. error

Figure 3.8: Performances of the four architectures on blurred KITTI *d2* sequences. The difference in performances between the ST and LS-VO networks is huge. VISO2-M has been omitted, for axis scale reasons.



(a) length vs tran. error (b) length vs rot. error



(c) speed vs tran. error (d) speed vs rot. error

Figure 3.9: Performances of the end-to-end architecture on blurred Malaga *d2* sequences. The lack of samples at high speed make the LS-VO network slightly overfit those samples, as shown in 3.9(c), but in all other respects the behaviour is similar to Figure 3.8.

3.5.3 Discussion

The experiments described in Section 3.5.3 on both datasets have been evaluated with KITTI devkit [60], and the output plots have been reported in Figures 3.5, 3.6, 3.7, 3.8 and 3.9. In all Figures except 3.7, the upper sub-plots, (a) and (b), represent the translational and rotational errors averaged on sub-sequences of length 100m up to 800m. The lower plots represent the same errors, but averaged on vehicle speed (Km/h). The horizontal axis limits for the lower plots, in Figures relative to $d2$ downsampled experiments are different, since the sub-sampling is seen by the evaluation software as an increase in vehicle speed. In Table 3.2 and 3.3 the total average translational and rotational errors for all the experiments are reported.

Figure 3.5 summarises the performances of all methods on KITTI without frame downsampling. From Figures 3.5(a) and 3.5(b) we observe that the BF-fed architectures outperform the Flownet-fed networks by a good margin. This is expected, since BF OF fields have been tuned on the dataset to be usable, while Flownet has not been fine-tuned on KITTI sequences. In addition, the LS-VO networks perform almost always better than, or on a par with, the corresponding ST networks. When we consider Figures 3.5(c) and 3.5(d), we observe that the increase in performance from ST to LS-VO appears to be slight, except in the rotational errors for the Flownet architecture. However, the difference between the length errors and the speed errors is coherent if we consider that the errors are averaged. Therefore, the speed values that are less represented in the dataset are probably the ones that are more difficult to estimate, but at the same time their effect on the general trajectory estimation is consequently less important.

The geometrical methods do not work on frame pairs only, but perform local bundle adjustment and eventually scale estimation. Even if the comparison is not completely fair with respect to learned methods, it is informative nonetheless. In particular we observe (see Figure 3.5) that the geometrical methods are able to achieve top performances on angular estimation, because they work on full-resolution images and because there is no scale error on angular rate. On the contrary, on average, they perform sensibly worse than learned methods for translational errors. This is also expected, since geometrical methods lack in scale estimation, while learned methods are able to infer scale from appearance. Similar results are obtained for the Malaga dataset. The complete set of experiments is available online [66].

When we consider the second type of experiment, we expect that the general performances of all the architectures and methods should decrease, since the task is more challenging. At the same time, we are interested in probing the robustness and generalization properties of the LS-VO architectures over the ST ones. Figure 3.6 shows the KITTI results. From 3.6(a) and 3.6(b) we notice that, while all the average errors for each length increase with respect to the previous experiments, they increase much more for the two ST variants. If we consider the errors depicted in Figures 3.6(c) and 3.6(d), we observe that the LS-VO networks perform better than the ST ones, except on speed around 60Km/h, where they are on par. This is understandable, since the networks have been trained on $d1$ and $d3$, that correspond to very low and very high speeds, so the OF in between them are the less represented in the training set. However, the most important consideration here is that the LS-VO architectures show more robustness to domain shifts. The plots of the performances on Malaga can be found online [66], and the same considerations of the previous one apply.

The last experiment is on the downsampled and blurred image. On these datasets both VISO2-M and ORBSLAM2-M fail to give any trajectory, due to the lack of keypoints, while Learned methods always give reasonable results. The results are shown in Figure 3.8 and 3.9 for the KITTI and the Malaga dataset, respectively. In both KITTI and Malaga experiments we observe a huge improvement in performances of LS-VO over ST-VO. Due to the difference in sample variety in Malaga with respect to KITTI, we observe overfitting of the more complex network (LS-VO) over the less represented linear speeds (above 30Kmh).

This experiments demonstrate that the LS-VO architecture is particularly apt to help end-to-end networks in extracting a robust OF representation. This is an important result, since this architecture can be easily included in other end-to-end approaches, increasing the estimation performances by a good margin, but without significantly increasing the number of parameters for the estimation task, making it more robust to overfitting, as mentioned in Section 3.4.2.

3.6 Conclusions

This work presented LS-VO, a novel network architecture for estimating monocular camera Ego-Motion. The architecture is composed by two branches that jointly learn a latent space representation of the input

OF field, and the camera motion estimate. The joint training allows for the learning of OF features that take into account the underlying structure of a lower dimensional OF manifold. The proposed architecture has been tested on the KITTI and Malaga datasets, with challenging alterations, in order to test the robustness to domain variability in both appearance and OF dynamic range. Compared to the data-driven architectures, LS-VO network outperformed the single branch network on most benchmarks, and in the others performed at the same level. Compared to geometrical methods, the learned methods show outstanding robustness to non-ideal conditions and reasonable performances, given that they work only on a frame to frame estimation and on smaller input images. The new architecture is lean and easy to train and shows good generalization performances. The results provided here are promising and encourage further exploration of OF field latent space learning for the purpose of estimating camera Ego-Motion. All the code, datasets and trained models are made available online [66].

Chapter 4

Publications

1. **Gabriele Costante** and Thomas A. Ciarfuglia *LS-VO: Learning Dense Optical Subspace for Robust Visual Odometry Estimation* 2017 IEEE Robotics and Automation Letters (RA-L), under review.

2. Silvia Cascianelli, **Gabriele Costante**, Thomas A. Ciarfuglia, Paolo Valigi and Mario L. Fravolini, *Full-GRU Natural Language Video Description for Service Robotics Applications* to appear in 2018 IEEE Robotics and Automation Letters (RA-L).

3. Michele Mancini, **Gabriele Costante**, Paolo Valigi and Thomas A. Ciarfuglia *J-MOD^2: Joint Monocular Obstacle Detection and Depth Estimation* 2017 IEEE Robotics and Automation Letters (RA-L), under review.

4. Michele Mancini, **Gabriele Costante**, Paolo Valigi, Thomas A. Ciarfuglia, Jeffrey Delmerico and Davide Scaramuzza *Toward Domain Independence for Learning-Based Monocular Depth Estimation* 2017 IEEE Robotics and Automation Letters (RA-L), vol. 2, no. 3, pp. 1778-1785.

Bibliography

[1] Sepp Hochreiter and Jürgen Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[2] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[3] Alex Graves, Greg Wayne, and Ivo Danihelka, "Neural turing machines," *arXiv preprint arXiv:1410.5401*, 2014.

[4] Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al., "End-to-end memory networks," in *Advances in neural information processing systems*, 2015, pp. 2440–2448.

[5] Anna Rohrbach, Marcus Rohrbach, and Bernt Schiele, "The long-short story of movie description," in *German Conference on Pattern Recognition*. Springer, 2015, pp. 209–221.

[6] Andrei Barbu, Alexander Bridge, Zachary Burchill, Dan Coroian, Sven Dickinson, Sanja Fidler, Aaron Michaux, Sam Mussman, Siddharth Narayanaswamy, Dhaval Salvi, et al., "Video in sentences out," *arXiv preprint arXiv:1204.2742*, 2012.

[7] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell, "Long-term recurrent convolutional networks for visual recognition and description," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 2625–2634.

[8] Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler, "Aligning books and movies: Towards story-like visual explanations by watching movies and reading books," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 19–27.

[9] Subhashini Venugopalan, Marcus Rohrbach, Jeffrey Donahue, Raymond Mooney, Trevor Darrell, and Kate Saenko, "Sequence to sequence-video to text," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 4534–4542.

[10] Subhashini Venugopalan, Huijuan Xu, Jeff Donahue, Marcus Rohrbach, Raymond Mooney, and Kate Saenko, "Translating videos to natural language using deep recurrent neural networks," *arXiv preprint arXiv:1412.4729*, 2014.

[11] Marcus Rohrbach, Wei Qiu, Ivan Titov, Stefan Thater, Manfred Pinkal, and Bernt Schiele, "Translating video content to natural language descriptions," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 433–440.

[12] Yingwei Pan, Tao Mei, Ting Yao, Houqiang Li, and Yong Rui, "Jointly modeling embedding and translation to bridge video and language," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4594–4602.

[13] Pingbo Pan, Zhongwen Xu, Yi Yang, Fei Wu, and Yueting Zhuang, "Hierarchical recurrent neural encoder for video representation with application to captioning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1029–1038.

[14] Lorenzo Baraldi, Costantino Grana, and Rita Cucchiara, "Hierarchical boundary-aware neural encoder for video captioning," *arXiv preprint arXiv:1611.09312*, 2016.

[15] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[17] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 4489–4497.

[18] Yoshua Bengio, Nicholas Léonard, and Aaron Courville, "Estimating or propagating gradients through stochastic neurons for conditional computation," *arXiv preprint arXiv:1308.3432*, 2013.

[19] Anna Rohrbach, Marcus Rohrbach, Niket Tandon, and Bernt Schiele, "A dataset for movie description," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3202–3212.

[20] Subhashini Venugopalan, Lisa Anne Hendricks, Raymond Mooney, and Kate Saenko, "Improving lstm-based video description with linguistic knowledge mined from text," *arXiv preprint arXiv:1604.01729*, 2016.

[21] Li Yao, Atousa Torabi, Kyunghyun Cho, Nicolas Ballas, Christopher Pal, Hugo Larochelle, and Aaron Courville, "Describing videos by exploiting temporal structure," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 4507–4515.

[22] David L Chen and William B Dolan, "Collecting highly parallel data for paraphrase evaluation," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*. Association for Computational Linguistics, 2011, pp. 190–200.

[23] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th annual meeting on association for computational linguistics*. Association for Computational Linguistics, 2002, pp. 311–318.

[24] Satanjeev Banerjee and Alon Lavie, "Meteor: An automatic metric for mt evaluation with improved correlation with human judgments," in *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, 2005, vol. 29, pp. 65–72.

[25] Chin-Yew Lin, "Rouge: A package for automatic evaluation of summaries," in *Text summarization branches out: Proceedings of the ACL-04 workshop*. Barcelona, Spain, 2004, vol. 8.

[26] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh, "Cider: Consensus-based image description evaluation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 4566–4575.

[27] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[28] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu, "3d convolutional neural networks for human action recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 1, pp. 221–231, 2013.

[29] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[30] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[31] Gabriele Costante, Michele Mancini, Paolo Valigi, and Thomas A Ciarfuglia, "Exploring Representation Learning with CNNs for Frame-to-Frame Ego-Motion Estimation," *Robotics and Automation Letters, IEEE*, vol. 1, no. 1, pp. 18–25, 2016.

[32] Ruben Gomez-Ojeda, Zichao Zhang, Javier Gonzalez-Jimenez, and Davide Scaramuzza, "Learning-based image enhancement for visual odometry in challenging hdr environments," *arXiv preprint arXiv:1707.01274*, 2017.

[33] Tatiana Tommasi, Novi Patricia, Barbara Caputo, and Tinne Tuytelaars, "A deeper look at dataset bias," in *Pattern Recognition: 37th German Conference, GCPR 2015, Aachen, Germany, October 7-10, 2015, Proceedings*. 2015, pp. 504–516, Springer International Publishing.

[34] Benjamin Ummenhofer, Huizhong Zhou, Jonas Uhrig, Nikolaus Mayer, Eddy Ilg, Alexey Dosovitskiy, and Thomas Brox, "DeMoN: Depth and Motion Network for Learning Monocular Stereo," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[35] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G. Lowe, "Unsupervised learning of depth and ego-motion from video," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[36] Sudheendra Vijayanarasimhan, Susanna Ricco, Cordelia Schmid, Rahul Sukthankar, and Katerina Fragkiadaki, "SfM-Net: Learning of structure and motion from video," *arXiv preprint arXiv:1704.07804*, 2017.

[37] Richard Joseph William Roberts, *Optical flow templates for mobile robot environment understanding*, Ph.D. thesis, Georgia Institute of Technology, 2014.

[38] Richard Roberts, Hai Nguyen, Niyant Krishnamurthi, and Tucker R. Balch, "Memory-based learning for visual odometry," in *2008 IEEE International Conference on Robotics and Automation (ICRA)*, 2008, pp. 47–52.

[39] Jonas Wulff and Michael J. Black, "Efficient sparse-to-dense optical flow estimation using a learned basis and layers," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[40] Jakob Engel, Thomas Schöps, and Daniel Cremers, "LSD-SLAM: Large-scale direct monocular SLAM," in *European Conference on Computer Vision (ECCV)*. Springer, 2014, pp. 834–849.

[41] Jakob Engel, Jörg Stückler, and Daniel Cremers, "Large-scale direct SLAM with stereo cameras," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 1935–1942.

[42] Christian Forster, Matia Pizzoli, and Davide Scaramuzza, "SVO: Fast semi-direct monocular visual odometry," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 15–22.

[43] Christian Forster, Zichao Zhang, Michael Gassner, Manuel Werlberger, and Davide Scaramuzza, "SVO: Semidirect visual odometry for monocular and multicamera systems," *IEEE Transactions on Robotics*, vol. 33, no. 2, pp. 249–265, 2017.

[44] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos, "ORB-SLAM: a versatile and accurate monocular slam system," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[45] Richard Roberts, Christian Potthast, and Frank Dellaert, "Learning general optical flow subspaces for egomotion estimation and detection of motion anomalies," in *2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009, pp. 57–64.

[46] Vitor Guizilini and Fabio Ramos, "Visual odometry learning for unmanned aerial vehicles," in *2011 IEEE International Conference on Robotics and Automation (ICRA)*, 2011, pp. 6213–6220.

[47] Vitor Guizilini and Fabio Ramos, "Semi-parametric models for visual odometry," in *2012 IEEE International Conference on Robotics and Automation (ICRA)*, 2012, pp. 3482–3489.

[48] Thomas A. Ciarfuglia, Gabriele Costante, Paolo Valigi, and Elisa Ricci, "Evaluation of non-geometric methods for visual odometry," *Robotics and Autonomous Systems*, vol. 62, no. 12, pp. 1717 – 1730, 2014.

[49] Peter Muller and Andreas Savakis, "Flowdometry: An optical flow and deep learning based approach to visual odometry," in *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, March 2017, pp. 624–631.

[50] Ronald Clark, Sen Wang, Hongkai Wen, Andrew Markham, and Niki Trigoni, "VINet: Visual-inertial odometry as a sequence-to-sequence learning problem," in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.*, 2017, pp. 3995–4001.

[51] Sen Wang, Ronald Clark, Hongkai Wen, and Niki Trigoni, "DeepVO: Towards end-to-end visual odometry with deep recurrent convolutional neural networks," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, May 2017, pp. 2043–2050.

[52] Sudeep Pillai and John J Leonard, "Towards visual ego-motion learning in robots," *arXiv preprint arXiv:1705.10279*, 2017.

[53] Kishore Reddy Konda and Roland Memisevic, "Learning visual odometry with a convolutional network.," in *International Conference on Computer Vision Theory and Applications (VISAPP)*, 2015, pp. 486–490.

[54] David J Heeger and Allan D Jepson, "Subspace methods for recovering rigid motion I: Algorithm and implementation," *International Journal of Computer Vision*, vol. 7, no. 2, pp. 95–117, 1992.

[55] Christian Herdtweck and Cristóbal Curio, "Experts of probabilistic flow subspaces for robust monocular odometry in urban areas," in *2012 IEEE Intelligent Vehicles Symposium, 2012 IEEE*, June 2012, pp. 661–667.

[56] Matthias Ochs, Henry Bradler, and Rudolf Mester, "Learning rank reduced interpolation with Principal Component Analysis," in *Intelligent Vehicles Symposium (IV), 2017 IEEE*, June 2017, pp. 1126–1133.

[57] Ian Goodfellow, Yoshua Bengio, and Aaron Courville, *Deep Learning*, MIT Press, 2016, `http://www.deeplearningbook.org`.

[58] Jun-Yan Zhu, Philipp Krähenbühl, Eli Shechtman, and Alexei A Efros, "Generative visual manipulation on the natural image manifold," in *European Conference on Computer Vision (ECCV)*. Springer, 2016, pp. 597–613.

[59] Philipp Fischer, Alexey Dosovitskiy, Eddy Ilg, Philip Häusser, Caner Hazırbaş, Vladimir Golkov, Patrick van der Smagt, Daniel Cremers, and Thomas Brox, "FlowNet: Learning optical flow with convolutional networks," in *2015 IEEE International Conference on Computer Vision (ICCV)*, Dec 2015, pp. 2758–2766.

[60] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun, "Vision meets robotics: The KITTI dataset," *The International Journal of Robotics Research*, p. 0278364913491297, 2013.

[61] Jos-Luis Blanco, Francisco-Angel Moreno, and Javier Gonzalez-Jimenez, "The mlaga urban dataset: High-rate stereo and lidars in a realistic urban scenario," *International Journal of Robotics Research*, vol. 33, no. 2, pp. 207–214, 2014.

[62] Alex Kendall, Matthew Grimes, and Roberto Cipolla, "Posenet: A convolutional network for real-time 6-dof camera relocalization," in *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[63] J. J. Kuffner, "Effective sampling and distance metrics for 3d rigid body path planning," in *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*, April 2004, vol. 4, pp. 3993–3998 Vol.4.

[64] Thomas Brox, Andrés Bruhn, Nils Papenberg, and Joachim Weickert, "High accuracy optical flow estimation based on a theory for warping," in *European Conference on Computer Vision (ECCV)*. 2004, pp. 25–36, Springer.

[65] A. Geiger, J. Ziegler, and C. Stiller, "StereoScan: Dense 3d reconstruction in real-time," in *2011 IEEE Intelligent Vehicles Symposium (IV)*, June 2011, pp. 963–968.

[66] "Isarlab @ unipg website," .

Relazione Assegno di Ricerca:
periodo dicembre 2016 - dicembre 2017

Modelli ad apprendimento computazionale per lo sviluppo di sistemi intelligenti eterogenei e per la domotica avanzata

Thomas Alessandro Ciarfuglia

Introduzione

Lo sviluppo di sistemi intelligenti dal punto di vista sia accademico che industriale ha preso con decisione la direzione dello sviluppo di algoritmi di percezione e collaborazione avanzata basati sui dati (data driven). In particolare molta aspettativa, non solo da parte dell'industria, ma anche della società più allargata, è stata messa su metodi di apprendimento automatico supervisionato. La grande disponibilità di dati in svariati settori, o comunque la propensione a raccoglierli vista l'enorme potenzialità di sfruttamento con algoritmi di apprendimento automatico, hanno modellato anche il panorama di ricerca in robotica degli ultimi due anni. In particolare l'enfasi sul Deep Learning è cresciuta fino al punto di generare specifiche sessioni a conferenza, o addirittura conferenze autonome, con lo scopo di esplorare il potenziale che queste tecniche hanno nella soluzione dei problemi aperti di automazione e robotica.

Nel corso del 2015-16, quando ancora il Deep Learning era confinato alla computer Vision e al Natural Language Processing, abbiamo presentato un lavoro pionieristico di Robotica mobile utilizzando queste tecniche. Questo lavoro è stato accolto con entusiasmo, è stato candidato a miglior articolo di visione di ICRA 2016 e, soprattutto, ha iniziato un filone di ricerca nuovo, che nel corso del 2016-17 è stato seguito da diversi altri gruppi di ricerca internazionali.

Nel corso di questo anno si è perseguita ulteriormente questa linea di ricerca, espandendola sia in profondità che in ampiezza, per lavorare con problemi più olistici rispetto alla sola stima del moto di un veicolo autonomo. Pertanto gli articoli prodotti durante l'anno riguardano l'interazione dei robot con l'ambiente in maniera più ampia:

1. Monocular Depth Estimation: Notoriamente è possibile stimare la struttura 3D di una scena utilizzando coppie di immagini separate da una certa distanza, utilizzando la geometria epipolare. Tuttavia la qualità di questa stima dipende dalla distanza che separa le due immagini (baseline). In alcuni casi però il peso di una stereocamera è troppo oneroso per un robot (ad esempio nel casi di Micro Aerial Vehicles – MAV). Per questo motivo abbiamo proposto metodi di stima della profondità della scena a partire da immagini monoculari al fine di

sviluppare algoritmi di pianificazione della traiettoria e obstacle avoidance. Questa linea di ricerca ha prodotto due lavori, entrambi pubblicati su rivista internazionale. In particolare uno ha visto la collaborazione con un gruppo di ricerca Svizzero leader nell'area di ricerca sulla navigazione autonoma di MAV.

2. Un'altra linea di ricerca ha riguardato l'interazione dei robot con le persone, in particolare la capacità di un robot di comprendere una scena vista con la telecamera e tradurla in frasi in linguaggio naturale di senso compiuto. Questo ha prodotto un lavoro recente a rivista. È auspicabile che questa linea di ricerca cresca ulteriormente nei prossimi mesi.
3. Un terzo settore, limitrofo, è emerso dall'applicazione delle tecniche di depth estimation monoculari a immagini radar satellitari. Sebbene queste siano diverse da quelle ottiche, esistono alcune analogie che ci hanno convinto che fosse possibile utilizzare le stesse tecniche utilizzate in robotica per la stima dei Digital Elevation Models a partire da una immagine radar. Questo ha prodotto un lavoro esplorativo che verrà presentato tra poche settimane in una conferenza internazionale.

Oltre al lavoro di ricerca in senso stretto, quest'anno è stato dedicato agli altrettanto importanti impegni di didattica, orientamento e found raising. Questi tre ambiti risultano molto onerosi in termini di tempo, ma nel medio e lungo periodo, sono quelli che di più garantiscono la crescita dell'istituzione universitaria e del territorio che essa serve.

Il piano del prossimo anno vede un ulteriore sviluppo in termini di visibilità internazionale in merito alla ricerca nel settore della Computer Vision applicata alla robotica e all'intelligenza artificiale, possibilmente attivando nuove collaborazioni con gruppi di oltre confine.

Di seguito sono allegati i lavori prodotti, pubblicati o sottomessi durante l'anno in esame, che costituiscono il corpo di questa relazione.

Elenco dei lavori

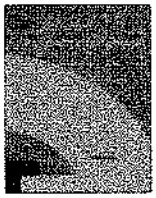
1. Cascianelli, S., Costante, G., Bellocchio, E., Valigi, P., Fravolini, M. L., & Ciarfuglia, T. A. "Robust visual semi-semantic loop closure detection by a covisibility graph and CNN features". *Robotics and Autonomous Systems*, *92*, 53-65. 2017
2. Mancini, M., Costante, G., Valigi, P., Ciarfuglia, T. A., Delmerico, J., & Scaramuzza, D. "Toward Domain Independence for Learning-Based Monocular Depth Estimation". *IEEE Robotics and Automation Letters*, *2*(3), 1778-1785. 2017
3. S. Cascianelli, G. Costante, T. A. Ciarfuglia, P. Valigi and M. L. Fravolini, "Full-GRU Natural Language Video Description for Service Robotics Applications," in *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 841-848, April 2018.
4. Mancini, M., Costante, G., Valigi, P., & Ciarfuglia, T. A. " J-MOD^2: Joint Monocular Obstacle Detection and Depth Estimation" in *IEEE Robotics and Automation Letters*, to appear (2018).
5. Costante, G., & Ciarfuglia, T. A. "LS-VO: Learning Dense Optical Subspace for Robust Visual Odometry Estimation" in *IEEE Robotics and Automation Letters*, to appear (2018).
6. Costante, G., Ciarfuglia, T.A., Biondi, F. "Towards Monocular Digital Elevation Model (DEM) Estimation by Convolutional Neural Networks - Application on Synthetic Aperture Radar Images" in *12th European Conference on Synthetic Aperture Radar*, upcoming (2018).

Robotics and Autonomous Systems 92 (2017) 53–65

Contents lists available at ScienceDirect

Robotics and Autonomous Systems

journal homepage: www.elsevier.com/locate/robot



Robust visual semi-semantic loop closure detection by a covisibility graph and CNN features☆

Silvia Cascianelli *, Gabriele Costante, Enrico Bellocchio, Paolo Valigi, Mario L. Fravolini, Thomas A. Ciarfuglia

Department of Engineering, University of Perugia, via Duranti 93, 06125, Perugia, Italy

HIGHLIGHTS

- A training-free appearance and viewpoint robust Place Recognition system is proposed.
- The method uses CNN features and preserves scene structure via a covisibility graph.
- A novel approach for synthesizing virtual views of the environment is proposed.
- Virtual views are particularly useful to face critical situations of viewpoint change.

ARTICLE INFO

Article history:
Received 26 August 2016
Available online 6 March 2017

Keywords:
Place recognition
Loop closing
CNN features
Graph
Semantic

ABSTRACT

Visual Self-localization in unknown environments is a crucial capability for an autonomous robot. Real life scenarios often present critical challenges for autonomous vision-based localization, such as robustness to viewpoint and appearance changes. To address these issues, this paper proposes a novel strategy that models the visual scene by preserving its geometric and semantic structure and, at the same time, improves appearance invariance through a robust visual representation. Our method relies on high level visual landmarks consisting of appearance invariant descriptors that are extracted by a pre-trained Convolutional Neural Network (CNN) on the basis of image patches. In addition, during the exploration, the landmarks are organized by building an incremental covisibility graph that, at query time, is exploited to retrieve candidate matching locations improving the robustness in terms of viewpoint invariance. In this respect, through the covisibility graph, the algorithm finds, more effectively, location similarities by exploiting the structure of the scene that, in turn, allows the construction of *virtual locations* i.e., artificially augmented views from a real location that are useful to enhance the loop closure ability of the robot. The proposed approach has been deeply analysed and tested in different challenging scenarios taken from public datasets. The approach has also been compared with a state-of-the-art visual navigation algorithm.

1. Introduction

In the last decade, vision-based navigation systems have achieved impressive results [1,2], considerably extending the application area of many robotic platforms. However, it is well known that, during long term operations, the localization performance may drop due to the drift of the estimation procedures, which can lead to a critical failure of most state-of-the-art systems. As a consequence, place recognition capabilities are crucial functions for loop closure detection and to increase the robustness of the overall estimation process.

Most of the existing place recognition strategies have been developed considering image sequences characterized by small viewpoint and lighting variations [3–5] and, within these scenarios, the results obtained are very promising. However, these simplified conditions do not hold in real life autonomous exploration contexts, where the visual scene is typically affected by a number of challenging problems. For instance, seasonal or weather changes, natural or artificial daily illumination variations may severely affect the global appearance of the scene; further, dynamic elements, *e.g.*, pedestrians, vehicles or new static objects may cause appearance changes, since they can occlude or alter portions of the scene. In addition, traversing the same environment

☆ This work has been partly supported by funds under the project SEAL [SCN-398]. We also gratefully acknowledge the support of NVIDIA Corporation with the donation of the Tesla K40 GPU used for this research.

* Corresponding author.
E-mail addresses: silvia.cascianelli@studenti.unipg.it (S. Cascianelli), gabriele.costante@unipg.it (G. Costante), enrico.bellocchio@unipg.it (E. Bellocchio), paolo.valigi@unipg.it (P. Valigi), mario.fravolini@unipg.it (M.L. Fravolini), thomas.ciarfuglia@unipg.it (T.A. Ciarfuglia).

http://dx.doi.org/10.1016/j.robot.2017.03.004
0921-8890/

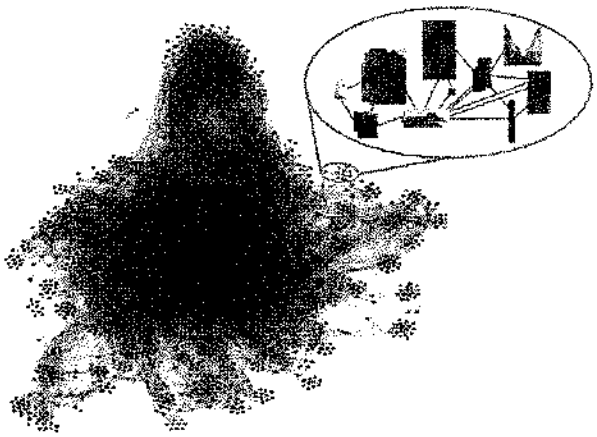


Fig. 1. Graph of Covisible CNN-Extracted features for semi-semantic visual Place Recognition: exemplar created graph.

with different orientations can change the scene viewpoint, which may alter significantly the relative position of objects in the scene.

Place recognition algorithms that exploit low level visual features [3,4] are typically very sensitive to strong image variations and, therefore, they do not provide good place recognition performance. Recent works [6–8] have shown that high level visual features, i.e., semantic cues, provide a more robust representation of the scene since they also encode information about object categories and their mutual relations. In fact, semantic features provide a better characterization of the scene, which may facilitate the place recognition process by an autonomous robot. However, the detection of different objects may not be enough to unequivocally identify a specific place (e.g., cars and buildings could be not discriminative in an urban environment). In these scenarios, the capability to discriminate between different spatial configurations and different views of the objects is crucial.

Motivated by the previous considerations, we have worked out a vision-based place recognition system that relies on a graph of semantic visual objects (see Fig. 1, where it is shown the graph produced by our algorithm using 623 images taken from the IDOL dum_sunny3 + dum_cloudy1 dataset [9]) that is built incrementally during navigation. In order to improve the robustness with respect to appearance changes, the graph was built in such a way that the nodes collect similar image patches that are represented by high level descriptors extracted by the inner convolutional layer of a public CNN trained specifically for object recognition purposes [10].

Furthermore, to handle viewpoint changes and to ease the place recognition task, the edges of the graph are used to encode covisibility information, that is edges are created to connect the objects that have been observed together from the same point of view. The result is a *covisibility graph* [11,12] that takes into account mutual object arrangements. In addition, the graph structure is exploited to build *virtual locations* [13] in a new strategy that relies only on graph algebraic properties. Virtual locations represent synthetic views of the scene that are not present in the image database. As a consequence, the algorithm has the potential ability to recognize places even in the presence of strong viewpoint changes.

To summarize, the main contributions of this work are:

- The employment of semi-semantic features extracted by a pre-trained CNN on the basis of image patches, which are robust to appearance changes, in a covisibility graph-based model of the environment, which enhances the viewpoint robustness of the place recognition algorithm.
- The development of a procedure for the construction of artificial virtual locations via a novel parameter-free approach that exploits the covisibility graph properties to face critical loop closure detection situations.
- The extension of the work in [14], with a different strategy for virtual location construction and with a deeper experimental analysis on the performance of each part of the proposed algorithm, which was evaluated on an extended number of datasets with respect to the work [14].

To the best of our knowledge, apart from [14], there are no previous applications that use high level features extracted by a CNN as nodes of a graph to build an incremental model of the environment during the exploration. Another important specific novelty of this study is the development of a parameter-free procedure for inferring artificial views on the basis of the developed graph model.

The remainder of this paper is organized as follows. In Section 2, related work is discussed, while in Section 3 the graph construction procedure is described. Section 4 describes the pipeline of the algorithm and Section 5 provides a detailed description of the experimental results. Conclusion and future development are discussed in Section 6.

2. Related work

Place recognition and loop closure detection are strictly related problems that are particularly important for autonomous robotic navigation in unknown environments. The main challenges for autonomous visual navigation in real life scenarios are viewpoint and appearance changes. A short categorization of the main research directions is provided below.

2.1. Appearance invariant approaches

The appearance change issue is typically faced via change removal methods, as in [15], via change prediction, as in Neubert et al. in [16], or by computing visual descriptors that exhibit invariance properties to appearance, as in [17], where the authors trained a multi-layer perceptron model to learn an appearance invariant set of descriptors. Among appearance invariant descriptors, features obtained from the inner layers of CNNs (that were pre-trained for object recognition tasks) have shown their effectiveness, as shown for instance in [18]. In particular, the authors in [15] and [19] were able to reduce significantly the effects of daily shadow and sunlight by transforming images in an illumination invariant colour space. The authors in [16] exploited the repeatability of the seasonal appearance changes, and built a super-pixel dictionary specific for each season and opportunely translated images captured in different seasons before matching. Authors in [17] studied the local changes of appearance of image patches subject to variation in lighting conditions and trained a multi-layer perceptron model and a convolutional multi-layer perceptron model for learning an appearance invariant feature descriptor. In [18,20] the authors extensively studied the appearance and viewpoint invariance properties of the outputs produced by different layers of pre-trained convolutional neural networks, specifically designed for object recognition and scene categorization. They demonstrated that the inner convolutional layer outputs provide robust appearance invariant features, while higher fully connected layers provide viewpoint robust features.

2.2. Viewpoint invariant approaches

Viewpoint changes are usually more critical than appearance changes. Some successful Simultaneous Navigation And Mapping (SLAM) systems exploit, as loop closure detection modules, Place

Recognition methods that are based on local invariant features. Some examples are FAB-MAP [3], which is based on SURF [21] features and ORB-SLAM [22], which is based on ORB [23] features. However, for visual Place Recognition algorithms viewpoint change is still a critical issue. Viewpoint invariance is generally addressed in an application dependent fashion, either by applying image rectification methods in case of mild viewpoint changes [24], or by considering the specific type of changes in the viewpoint that will be encountered while performing a specific task *e.g.*, [25–27]. In particular, the authors in [24] estimate and normalize affine parameters of local transformations in the images, but their approach is applicable only to objects with regular structure, as *e.g.*, buildings. Some heuristics or solutions designed for specific environments are applied to perform visual Place Recognition in case of specific severe viewpoint changes, such as in case of lane traversal in [25], panoramic vision in [26] or air–ground viewpoint change in [27].

2.3. Appearance and viewpoint invariant approaches

Scenarios characterized both by viewpoint and appearance changes are particularly challenging for the loop closure detection task. Promising solutions usually rely on CNNs specifically designed for place recognition [28] or on features extracted from a CNN designed for object recognition [6], or viewpoint synthesis [29], or exploiting robust sequence matching techniques [25].

2.4. Graph-based approaches

Modelling the environment as a graph requires the definition of what "a node is" and of a criterion that defines the node connection mechanism. In order to preserve geometric information, in [7,8] a geometric graph based on the distance between centres of 3D point clouds or 2D patches around a landmark was proposed. A recent work by Pepperell et al. [30] focused on maze urban environments and used roads as directed edges connecting intersections to facilitate sequence matching in place recognition. Another general criterion for building graphs of the environment, while dealing with bidimensional images, is based on the covisibility of the landmarks, *i.e.*, an edge is created between landmarks if they are present in the same image. This approach was proposed in [13] and is also adopted in this work, with the important difference that, instead of using hand-crafted descriptors, we use features extracted by a convolutional layer of a pre-trained CNN that receives as input unprocessed image patches. Using a graph to model the environment allows the integration of additional information from other sources, such as robots or other intelligent systems. Hence, it provides a framework that can be easily integrated with network information, and with other environment specific visual object galleries following a transfer learning paradigm [31].

3. Incremental covisibility graph construction

In this study we assume that the autonomous robot does not have at its disposal any prior information on the environment, that is, the visual exploration starts from scratch. As a new image is captured, patches containing objects are extracted and then processed by a CNN. The outputs of an inner layer of the CNN, along with the dimensions of the patches, are used to build a graph-based representation of the environment and to enrich the collection of landmarks encountered as the exploration progresses. In Fig. 2 a block diagram of the operations performed in this knowledge acquisition phase is shown; below, the building blocks of this scheme are described in detail.

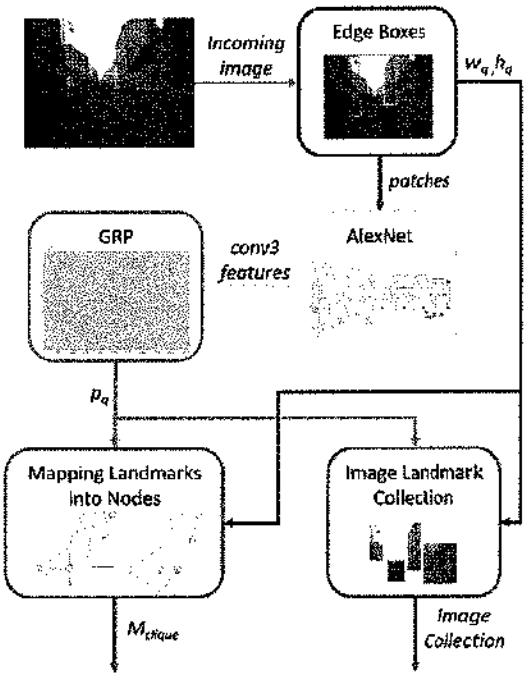


Fig. 2. Schematic representation of the visual information processing blocks used during the exploration. When a new image is acquired, the Edge Boxes algorithm (dark red block) extracts a pre-defined number of image patches. These are fed to AlexNet (yellow block), from which the output of conv3 layer is retained. The dimensionality of this output vector is reduced via Gaussian Random Projection (cyan block). Information about each patch enriches the incremental database of images (magenta block) and extends the covisibility graph by mapping landmarks in existing or new nodes (green block). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

3.1. Semi-semantic landmarks extraction

The model of the environment is here obtained using high level visual landmarks extracted from the scene acquired by the robot during navigation. For each new image the landmarks are derived from the processing of image patches that are likely to contain a generic object. In this work the number of extracted patches per frame is constant and fixed at 50. To obtain these patches we apply the algorithm by Zitnick et al. proposed in [32], named Edge Boxes, which efficiently detects a bounding box around a patch (of variable size and dimensions) that contains a high number of internal contours compared to the number of contours exiting from the box. This fact indicates the presence of an intelligible object in the enclosed patch. The visual content of these patches, however, is not associated with an 'object label' *i.e.*, the Edge Boxes algorithm does not provide any object categorization for the object within the patches. For this reason our method can be considered a "semi-semantic" approach.

The 2D patches extracted by the Edge Boxes algorithm are directly processed by a pre-trained CNN and the output produced by an inner layer of the CNN is used as descriptor vector of the patch.

The strategy of using, as descriptors, the outputs provided by inner layers of a pre-trained CNN was proposed by some authors as in [18,33,34] thanks to the high representational power of deep nets.

In this study, we use the pre-trained AlexNet CNN [10], that is a well-known CNN used for Object Recognition, and select the output of the conv3 layer as descriptor vector. This choice is mainly motivated by the study reported in [18], where the output descriptors provided by the different layers of some CNNs for Object

Recognition and Place Recognition were compared in order to find the best descriptor vector for the Place Recognition task. In particular, the authors of [18] demonstrated that in case of viewpoint changes, AlexNet has a slight performance improvement compared to CNNs trained on location-based images if considering the whole images. The same authors in [6] demonstrated that using region-based features rather than whole-image features provides a benefit in terms of viewpoint robustness. Since our region-based features are extracted on the basis of image patches containing objects, we decide to use AlexNet as feature extractor.

AlexNet works on fixed size images, while Edge Boxes produces patches with arbitrary dimensions, therefore we resize them in order to fit the AlexNet input dimensions. In order not to lose the original size information, the height and width of the patch are considered as additional descriptors, together with the conv3 output vector.

The conv3 layer output is a vector of $13 \times 13 \times 384 = 64\,896$ elements that provides a redundant representation of the input image which is useful to better discriminate between classes of objects. Considering that in the robotic exploration it is important to limit the real time computational load we decide to reduce the dimensionality of conv3 output by applying the Gaussian Random Projection method [35] obtaining a reduced vector of length 2048. This reduction does not significantly deteriorate matching performance, since Gaussian Random Projection provides a good approximation of radial metrics that are typically used to measure the similarity between vectors (as the Euclidean distance or the cosine similarity). The choice of the size for the reduced dimension of the conv3 output has been made considering both the results of the study in [6] and additional parametric studies that were carried out on the Gardens Point day-left and day-right dataset [18].

The Edge Boxes patches, described by the reduced AlexNet conv3 output p_q and their width w_q and height h_q (*i.e.*, by triples (p_q, w_q, h_q)), constitute the semi-semantic landmarks that are used as basic components of the graph-based representation of the environment.

3.2. Graph nodes and edges

The characterization of a graph requires the definition of its nodes and edges. Inspired by the work of Stumm et al. [13], we build a covisibility graph that models the environment as a structured collection of visual landmarks, acquired sequentially during the environment exploration.

In particular, the nodes of our graph are built on the basis of the semi-semantic landmarks (described in Section 3.1) using the procedure described in details in Section 3.3.

Covisibility information is modelled by connecting the nodes belonging to the same image by an unweighted edge, *i.e.*, nodes observed from the same point of view are connected. Landmarks in the same image are therefore fully connected, forming a complete subgraph for that image.

This node connection policy encodes proximity relations among patches (and their enclosed objects), but it is not strictly related to any metric distance information, that is objects that are metrically distant may be connected in the covisibility graph and metrically close objects (because of visual occlusions) can be not connected. Hence, our method does not rely on the metric position of the patches but uses only visual information.

3.3. Mapping landmarks into nodes

In the previous section we described how we build the covisibility subgraph of a new acquired image during the exploration.

Now, in order to incrementally build the graph of the whole environment, we need to specify how to connect each new subgraph to the current graph. This is carried out by mapping the landmarks extracted from a new image in nodes of the graph. For the first image (*i.e.*, at the beginning of the exploration), a node is created for each of the extracted landmarks. For the following images, new nodes are added only for new landmarks, while the landmarks having small distance from existing nodes are considered as "already seen landmarks" and are therefore mapped in the best matching existing node. An illustration of the graph building process is shown in Fig. 3 while in Fig. 4 we report an example of nodes that are generated by our algorithm on the Gardens Point day-left and day-right dataset [18] and the visual patches contained in these nodes.

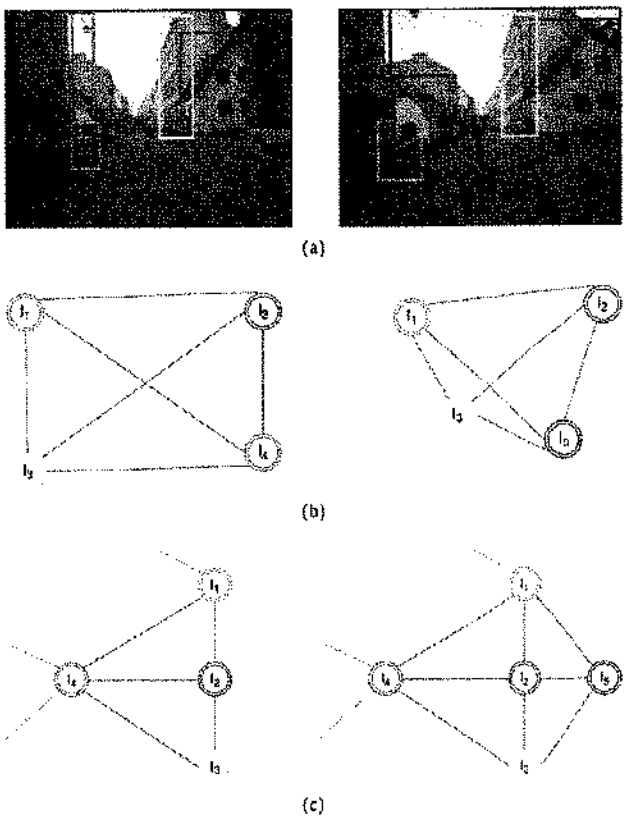


Fig. 3. Incremental covisibility graph construction during the environment exploration. Examples of Edge Boxes landmarks extracted from images at time $k-1$ (left) and at time k (right) respectively are shown in 3a. Relative landmark covisibility subgraphs of images visible at time $k-1$ (left) and at time k (right) respectively are shown in 3b: landmarks acquired in the same image are connected in a dense graph. Landmark covisibility whole graph at time $k-1$ (left) and at time k (right) respectively are shown in 3c: similar landmarks are mapped in the same node, while different landmarks produces new nodes.

In this study the similarity between landmarks is measured using the scalar cosine distance d_{ij} between the feature vector $p_{q,i}$ of the ith landmark in the current image and the one it is most similar to, $p_{c,j}$ taken among all the landmarks in the previous images.

To speed up the search for the most similar landmark, we exploit the KD-Tree algorithm proposed in [36]. This algorithm works only with distance metrics that are component-wise additive and monotonically increasing with components addition, as in the case of the Euclidean distance. Cosine similarity is more suitable than Euclidean distance for high dimensional data, but does not exhibit the characteristics requested by the KD-Tree algorithm. This technical problem is overcome by first calculating the Euclidean distance between l_2-normalized feature vectors and then applying

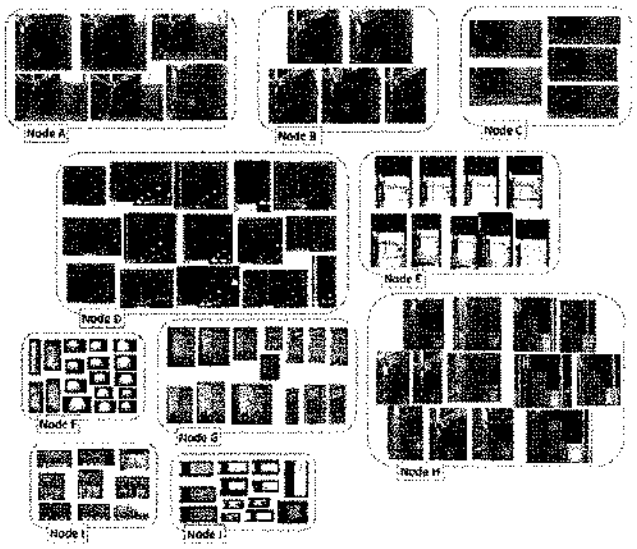


Fig. 4. Landmarks belonging to some sample nodes: different nodes can contain scaled versions of the same landmark (*e.g.*, nodes A, B and C), the same node can contain a small number of different outlier patches (*e.g.*, nodes F and J) and in the same node there can also be clusters of patches, smoothly similar each other.

the following transformation:

$$d_{ij} = 1 - \frac{d_{Euclidean,ij}}{2} \tag{1}$$

where $d_{Euclidean,ij}$ is the Euclidean distance and d_{ij} is the scalar cosine distance between the landmarks $p_{q,i}$ in the current image and $p_{c,j}$ in the previous images.

For each most similar pair of landmarks we also calculate the "dissimilarity" measure of the geometric shape of their bounding boxes s_{ij}. The definition of s_{ij} is taken from [6]:

$$s_{ij} = exp\left\{\frac{1}{2}\left(\frac{|w_{q,i} - w_{c,j}|}{max\left\{w_{q,i}, w_{c,j}\right\}} + \frac{|h_{q,i} - h_{c,j}|}{max\left\{h_{q,i}, h_{c,j}\right\}}\right)\right\}. \tag{2}$$

Values of s_{ij} that are close to 1 indicate that bounding boxes are similar, while larger values indicates differences in their area and shape.

The overall similarity between landmarks in the current image and the most similar landmarks in the previous images is then computed as:

$$P_{ij} = 1 - d_{ij} \cdot s_{ij}. \tag{3}$$

Values of P_{ij} that are close to 1 indicate that the two considered landmarks have both very similar shape and conv3 feature descriptor, while small values indicate a difference that can be due to both shape and conv3 features; negative values indicate a relevant difference in the shape of the patches. Using the shape dissimilarity coefficient s_{ij} as a multiplicative factor enhances the cosine distance d_{ij} between the conv3 features. This allows the information on the shape of patches, that is lost (as explained in Section 3.1) because of the resizing of the patches that is requested to use the AlexNet CNN, to be taken into account.

Finally, landmarks are considered to be "the same landmark" (and therefore mapped in the same node of the graph) when the overall similarity P_{ij} is larger than a user defined threshold. The higher this threshold is, the more similar are the landmarks contained in the same node. However, the algorithm becomes slower because of the fast growth of the whole covisibility graph, while the overall recognition performances are not significantly improved.

It is important to note that a new image produces at most as many new nodes as the maximum number of patches extracted by the Edge Boxes algorithm (50 in this study) since very similar (overlapping) patches are mapped in a unique node.

The analysis of Fig. 4 highlights some important characteristics of the nodes that are built with the above procedure. Specifically, different nodes can include scaled versions of the same landmarks (*e.g.*, nodes A, B and C); the same node can include some outlier patches (*e.g.*, nodes F and J) because of the resizing needed to feed AlexNet; in the same node there can be clusters of patches, similar to each other, since we associate new landmarks to a node computing the similarity with the whole set of landmarks associated to that node and not simply with a "centroid" landmark for that node (*e.g.*, node G).

3.4. Graph representation

In practice, the computed covisibility graph is encoded and managed using a sparse clique matrix, M_{clique}, whose rows represent nodes and whose columns represent image indices, so that a 1 in $M_{clique}[p, f]$ means that the node p is present in the image f.

The graph growth due to the allocation of a new node is implemented by the following matrix update:

$$M_{clique}|_{k-1} = \begin{pmatrix} \cdots & 1 \\ \vdots & 1 \\ \vdots & 1 \\ \cdots & 1 \end{pmatrix} \rightarrow M_{clique}|_{k} = \begin{pmatrix} \cdots & 1 & 1 \\ \vdots & 1 & 1 \\ \vdots & 1 & 1 \\ \vdots & 1 & 0 \\ \cdots & 0 & 1 \end{pmatrix} \tag{4}$$

where in (4) a new column is added for the current image, which has 1 s in the existing rows corresponding to already observed landmarks. In addition, when a landmark is assumed to be new, then a new row is allocated, having a 1 in the column associated to the last image, where the landmark was observed (allocated) the first time.

The representation via a sparse matrix also provides an efficient indexing for the image dataset. In fact, considering the definition of the M_{clique} matrix, we know that the rows that are associated with a specific landmark contain ones in positions corresponding to the indices of images where that landmark has been observed, and, conversely, for each image we can know which landmarks belong to that image. This information can be obtained in constant time.

It is instructive to look at the 2D geometry of the clique matrix. For this purpose we generate the clique matrix from the City Centre benchmark dataset [3], that is characterized by a trajectory that is traversed twice. In this representation, zeros are white dots, while ones are black dots. The corresponding clique matrix (shown in Fig. 5) presents a repeating nodes pattern in the image indices corresponding to images collected during the two traversals of the same path. This indicates a loop, since the algorithm recognizes many landmarks allocated during the first traversal, along with a few new nodes that are specific of the second traversal.

It is also observed that, due to the presence of already acquired landmarks, the M_{clique} matrix has a growth rate slower than 50 new nodes per image: for example, in the City Centre Dataset, which contains 1237 images, our algorithm creates 8326 nodes instead of $50 \times 1237 = 416\,300$ nodes. It is expected that the continuous exploration of the same environment will tend to decrease the allocation rate of new nodes over time. This aspect is very important for robotic applications because, for a space constrained environment, we expect a sort of saturation effect to slow down the graph growing process, thus limiting memory consumption of our system.

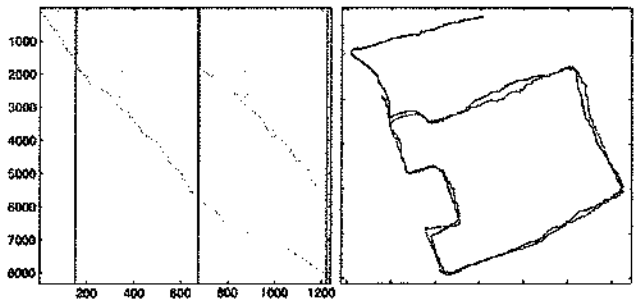


Fig. 5. Trajectory and Clique matrix M_{clique} relative to the City Centre Dataset. This dataset presents a circular trajectory traversed two times starting form image 152 to image 674 and from image 675 to image 1220 and its clique matrix presents a repeating nodes pattern in the corresponding image indices, along with few new nodes that are specific of the second traversal. The allocation rate of new nodes is inferior in the second traversal with respect to the first traversal because the robot sees many landmarks belonging to already allocated nodes and only a small number of new nodes is allocated.

4. Place recognition algorithm

In this section the proposed place recognition algorithm whose block diagram is shown in Fig. 6 is described. The purpose of this algorithm is to find possible matchings between the current image (that in this phase is called "query image") and a subset of the most promising images in the set of images (called "image collection") that has been acquired previously (also named as "candidate images"). In particular, the place recognition algorithm is based on the visual modelling of the environment described in Section 3. The matching score between images is computed taking into account two aspects: the mean similarity of landmarks in the query and candidate images and the similarity between images subgraphs. In addition, in order to facilitate the detection of possible loop closures in critical points along the path, a mechanism that produces artificial "enlarged views" (also named "virtual locations") on the basis of the candidate images is proposed.

4.1. Candidates retrieval

In this section the first block of the system which exploits the covisibility graph is described. Considering a query image, we select, from the whole image collection only a subset of images to be further analysed for the detection of possible loop closures. In particular we retrieve the images that share at least a minimum number of nodes (this number is a free design parameter) with the query image. The sparseness of the clique matrix allows us to efficiently identify (in constant time) the candidate images that fulfil this retrieval criterion.

In this work, the retrieval criterion is "unselective" and all the images that share at least one node with the query one are retrieved. It should be noted that a more selective criterion could be used, improving the speed and precision of the entire algorithm. In fact, a more selective criterion automatically excludes from the analysis many true negative matching images, so that the retrieved images are only those sharing a large number of landmarks with the query image, thus the loop closure detection system would prove to be more precise. However, a selective criterion also has the potential drawback of inducing a possible recall drop (*i.e.*, the fraction of relevant images that are effectively considered) due to the exclusion of many true positive matchings along with the true negative matchings. This side effect is more relevant with the increase in the minimum number of shared nodes requested by the algorithm. This trend is clearly confirmed in Table 1, which shows the percentage of true positive and true negative matching

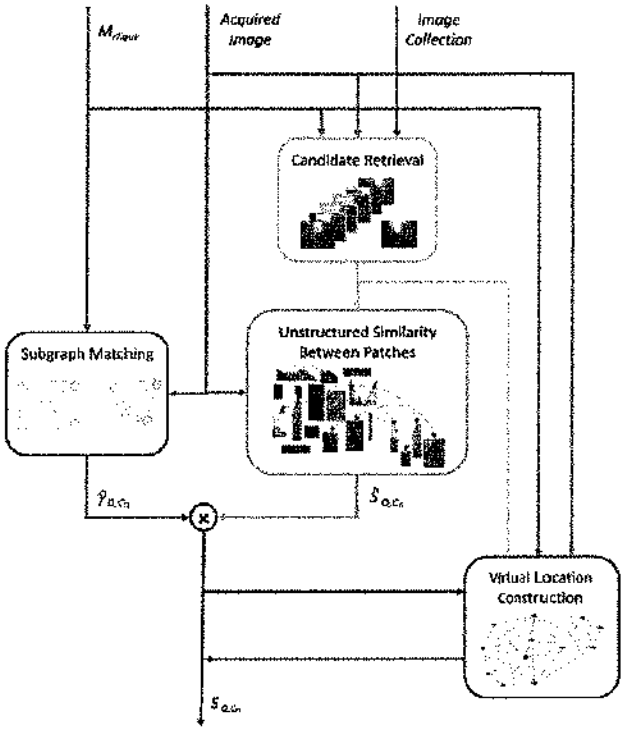


Fig. 6. Schematic representation of the proposed Place Recognition system. The covisibility graph is exploited to retrieve the most relevant candidate images (orange block). For each one of the retrieved images, it is calculated the landmarks similarity score (light green block) and the subgraph matching score (red block). Those values are multiplied and used as baseline score in the process of virtual location construction (blue block). Using this latter block the final similarity score for each candidate image is assessed. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

images that have been excluded from analysis due to the retrieval criterion in the four datasets that are used for the experiments (described in Section 5.1). Note that the New College and City Centre Dataset contain images from different environments (such as gardens, archways, squares, alleys and inner urban areas), *i.e.*, high "intra dataset" diversity. Thus, even a loose retrieval criterion is favourable in terms of a priori excluded true negatives. Conversely, the Malaga parking 6L dataset contains images that are more similar to each other. Thus, the positive effect, in terms of a priori excluded true negatives, of not strict retrieval criteria is less evident. Finally, unlike the other datasets, the IDOL dum_sunny3+dum_cloudy1 dataset was collected in an indoor environment and exhibits high sensitivity to the retrieval criterion. In particular, it is observed that the negative effects of strict criteria in terms of a priori excluded true positives are visible also for less strict criteria that, conversely, do not severely affect outdoor datasets.

Finally, the choice of a reasonable minimum number of shared nodes is application dependent: for example, for a localization and mapping task, precision is critical and a strict retrieval criterion (*e.g.*, minimum number of shared nodes equal to 10) is advisable.

4.2. Unstructured similarity between images

In this section we analyse the block that computes the similarity between landmarks to establish whether a candidate image from the image collection matches with the current query image. This block is based on the algorithm proposed by Sunderhauf et al. [6] and does not consider the covisibility information.

Table 1
Percentage of true positive (TP) and true negative (TN) matching images a priori excluded from matching due to the retrieval criterion (Minimum Number of Shared Nodes) in the four tested public datasets. A strict criterion causes the exclusion of many True Negatives, thus augmenting the precision, but it also causes the exclusion of many True Positives, thus reducing appreciably the recall.

Minimum number of shared nodes	New college		City centre		IDOL dum_sunny3+dum_cloudy1		Malaga parking 6L	
	Excluded TP	Excluded TN	Excluded TP	Excluded TN	Excluded TP	Excluded TN	Excluded TP	Excluded TN
1	0.10%	44.50%	0.48%	8.53%	0.00%	0.18%	0.00%	0.14%
5	12.23%	72.50%	13.10%	92.36%	74.92%	93.14%	4.51%	22.55%
10	44.33%	94.37%	43.29%	98.71%	91.67%	99.10%	50.93%	95.53%
20	99.29%	99.96%	96.13%	99.97%	98.14%	99.85%	84.38%	99.95%

The similarity measure between the query and candidate images is derived as a function of the landmarks' feature vectors and of the shape parameters of their bounding box. The algorithm computes a similarity score, P_{ij} (via Eq. (3)), between each landmark in the query image and the most similar landmark in the candidate image under investigation. The matching score is then assigned to a candidate image as the mean value of individual scores of its landmarks:

$$\hat{S}_{Q,C_n} = \frac{1}{N_P} \sum_{ij} P_{ij}. \tag{5}$$

Note that, since the considered landmarks in this phase are those of the query and a candidate image, the similarity score between a pair of landmarks can be smaller than the threshold that has been fixed in Section 3.3 to map them in the same node of the graph. This is reasonable because in this phase the similarity between images is computed on the basis of landmarks appearance, without exploiting the covisibility graph information.

4.3. Subgraph matching

The purpose of the Subgraph Matching block is to exploit the information embedded in the covisibility graph in order to refine the previously computed matching score $\hat{S}_{Q,C_n}$, which is based only on similarity between landmarks (Section 4.2). In particular, we exploit the graph Adjacency matrix to take into account the neighbouring information of the nodes in each image subgraph. The Adjacency matrix is obtained on the basis of the graph clique matrix M_{clique}.

As the exploration proceeds, the covisibility graph grows, thus, except in the initial phase, our system deals with a large clique matrix. In order to manage efficiently the large dimensionality, we implement an ad-hoc procedure (see the pseudo code in: Algorithm 1) that exploits the definition of the Adjacency matrix for its calculation, thus limiting significantly the computational cost needed to obtain it (*i.e.*, $O(N^2)$, that is further reduced to $O(N)$ thanks to the sparsity of the clique matrix).

Algorithm 1 Obtain Adjacency matrix

```
Input : M_clique
Output : A
A ← 0_{N×N}
for x ← 1 to N do
    ▷ isolate M_clique columns having 1 in row index x
    x_columns ← M_clique|x=1,:|
    ▷ set to 0 x_columns element in row index x
    x_columns(x) ← 0
    ▷ collect indices of node x's neighbours
    x_neighbours ← indexOf(x_columns = 1)
    A[x, x_neighbours] ← 1
end for
```

Note that during the graph construction the nodes maintain their order (that is the order in which they have been allocated during the exploration as explained in Section 3.4), thus the row and column indices of the Adjacency matrix are the same for the query and candidate images subgraphs. This implies that the subgraphs are aligned [37], with the great advantage that they can be directly compared by means of their Adjacency matrices. The similarity between the candidate and the query Adjacency matrices is measured by means of the normalized cross correlation as follows:

$$\gamma_{Q,C_n} = \frac{\sum_{ij} A^Q_{ij} \cdot A^{C_n}_{ij}}{\sqrt{\sum_{ij} (A^Q_{ij})^2 \cdot \sum_{ij} (A^{C_n}_{ij})^2}} \tag{6}$$

where in (6) A^Q_{ij} and $A^{C_n}_{ij}$ are the Adjacency matrix entries relative to landmarks p_i and p_j in the subgraphs of query location Q and candidate location C_n respectively.

Then we maintain only normalized cross-correlation values that are lower than a defined fraction α (set at 0.1 in this study) of the normalized cross-correlation between the query image and the previous one C_{k-1}, which is reasonably the most correlated with the current query image, as:

$$\hat{\gamma}_{Q,C_n} = \begin{cases} \gamma_{Q,C_n} & \text{if } \gamma_{Q,C_n} < \alpha \cdot \gamma_{Q,C_{k-1}} \\ 1 & \text{if } \gamma_{Q,C_n} \geq \alpha \cdot \gamma_{Q,C_{k-1}} \end{cases}. \tag{7}$$

Note that α can assume any value between 0 and 1. The choice of setting $\alpha = 0.1$ is guided by the consideration that a small value implies a small cross-correlation between the Adjacency matrices of the query and candidate images. In fact, the obtained $\hat{\gamma}_{Q,C_n}$ value is used to weight the similarity score $\hat{S}_{Q,C_n}$ (5) of each candidate location, thus filtering out matching scores of candidate location whose landmark arrangement is too different from that of the query location.

The resulting matching score between images is thus computed as follows:

$$S_{Q,C_n} = \hat{\gamma}_{Q,C_n} \cdot \hat{S}_{Q,C_n} \tag{8}$$

4.4. Virtual locations

Each new acquired query image is compared to a subset of images from the Image Collection which have been retrieved as described in Section 4.1. In this block each candidate image is "virtually" expanded using the visual information of neighbouring images.

This can be very useful in situations where viewpoint changes are critical. When a place is revisited it is reasonable to assume that the viewpoint is different, this especially in proximity of 90° corners or in stretches traversed with lateral displacement. In such a situation some detected landmarks can have a very different relative position, others can be occluded and some new ones can enter the current view. Thus, the place recognition algorithm can benefit from the generation of virtual locations in order to compensate viewpoint changes.

A possible strategy to build virtual locations is to temporarily add nodes (landmarks) to the current candidate image under

investigation. Previous works, such as [12,13] and [14], obtained virtual locations by "merging" subgraphs of candidate images that share a user-defined number of nodes. In this work, we remove this parameter and propose a strategy based on the spectral properties of the covisibility graph. In particular, the nodes to be added to the current candidate image are selected following an agglomerative clustering approach [38]. The agglomerative clustering algorithms start from a seed subgraph and iteratively include nodes among its neighbours (*i.e.*, nodes that are connected at least to one node that already belongs to the seed). In this respect, the subgraph of the current candidate image is used as seed and its neighbourhood contains nodes belonging to other candidate images. Our node selection criterion is based on the graph connectivity metrics, which is computed exploiting the algebraic graph theory as explained below.

Considering the Adjacency matrix A of the graph, its Degree matrix D can be immediately derived. This is a diagonal matrix with as many rows and columns as the number of nodes and, for an undirected graph (such as our covisibility graph) D contains the number of each node's neighbours in the corresponding diagonal positions, that is:

$$D_{ij} = \sum_{j=1}^{N} A_{ij} \tag{9}$$

where N is the total number of nodes in the graph.

On the basis of the Adjacency matrix and of the Degree matrix it is possible to compute the Laplacian matrix L of the graph as:

$$L = D - A. \tag{10}$$

By construction L is singular, symmetric and positive semidefinite in case of a undirected graph. Eigen-decomposition of the Laplacian matrix induces a clustering of the nodes of the graph (in particular it makes it possible to identify a specific number of groups of nodes depending on the eigenvector we select for clustering purpose).

The N ordered eigenvalues of L are defined as $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_N$. The sum of each row and column of L is zero, thus, by construction, the eigenvalue λ_1 is equal to zero and its associated eigenvector is $\mathbf{u}_1 = \mathbf{1}$, in fact $L\mathbf{u}_1 = \mathbf{0}$.

In this study we exploit the second eigenvector, $\mathbf{u}_2$, associated to eigenvalue λ_2 , since it provides a measure of the graph connectivity as explained in [39,40]. For instance, Fig. 7 shows the components of the eigenvector $\mathbf{u}_2$ mapped on the nodes of a sample covisibility graph computed on the first 20 images of the City Centre Dataset. It may be observed that the components of $\mathbf{u}_2$ vary smoothly from the smallest ones (in blue) to the largest ones (in red), thus inducing a natural ranking of the nodes of the graph.

Based on the previous considerations, each candidate image can be expanded by adding nodes, one by one, as a function of similarity measure provided by the $\mathbf{u}_2$ component value. This strategy reflects the fact that the node to be added is the most connected to those actually contained in the candidate image subgraph.

After the addition of a node to the candidate seed subgraph, the matching score of the expanded candidate location is recalculated. The expansion process is stopped if the similarity measure between query and candidate images decreases. The process is also stopped if a predefined maximum number of nodes is added to the seed location (we set this limit to 50% of the number of Edge Boxes extracted, which is equals to 25 in this work). The role of this additional stopping criterion is twofold. First, it limits the time complexity of the virtual location construction procedure and second, it prevents false positive matches. In fact, if the expansion were uncontrolled, a candidate image would likely obtain a high matching score because of the addition of many nodes not belonging to its original subgraph, thus the matching score might prove misleading. The pseudo-code of the virtual location construction process is reported in the Algorithm 2 table.

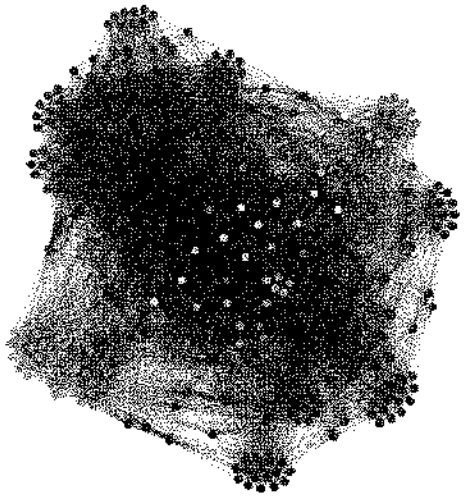


Fig. 7. Eigenvector $\mathbf{u}_2$ components associated to each node a subgraph of the City Centre Dataset, made of the first 20 images: note the inducted partition in two subsets.

Algorithm 2 Obtain Virtual Location

Input : $\mathbf{u}_2$, M_{clique}, S_{Q,C_n}

Output : S^*_{Q,C_n}, $M^{C_n^+}_{clique}$ ▷ $C_n^+ \doteq$ expanded candidate

$S^*_{Q,C_n} \leftarrow S_{Q,C_n}$

$M^{C_n^+}_{clique} \leftarrow M_{clique}[:, C_n]$

$M^{\hat{C}_n^+}_{clique} \leftarrow M_{clique}[:, C_n]$

▷ $\hat{C}_n^+ \doteq$ temporary expanded candidate

$added \leftarrow 0$

while $added < \frac{N_P}{2}$ **do** ▷ $N_P = 50$ in this study

▷ collect indices of nodes in seed subgraph

$seed \leftarrow indexOf(M^{C_n^+}_{clique} = 1)$

▷ collect indices of nodes not in seed subgraph

$\mathcal{N}(seed) \leftarrow indexOf(M^{C_n^+}_{clique} = 0)$

▷ $\mathcal{N}(seed) \doteq$ seed neighbourhood

▷ find the index of the best node to add to seed subgraph

$best\_neighbour \leftarrow \underset{j \in \mathcal{N}(seed)}{\operatorname{argmin}} \left\{ \sum_{i \in seed} (\mathbf{u}_2[i] - \mathbf{u}_2[j])^2 \right\}$

▷ add *best_neighbour* to the current seed subgraph

$M^{\hat{C}_n^+}_{clique}[best\_neighbour] \leftarrow 1$

calculate $S_{Q,\hat{C}_n^+}$

if $S^*_{Q,C_n} \geq S_{Q,\hat{C}_n^+}$ **then**

break

else

$M^{C_n^+}_{clique} \leftarrow M^{\hat{C}_n^+}_{clique}$

$S^*_{Q,C_n} \leftarrow S_{Q,\hat{C}_n^+}$

$added + 1$

end if

end while

To have an idea of the positions where the virtual locations are actually generated along the paths, in Fig. 8 we report the 2D GPS

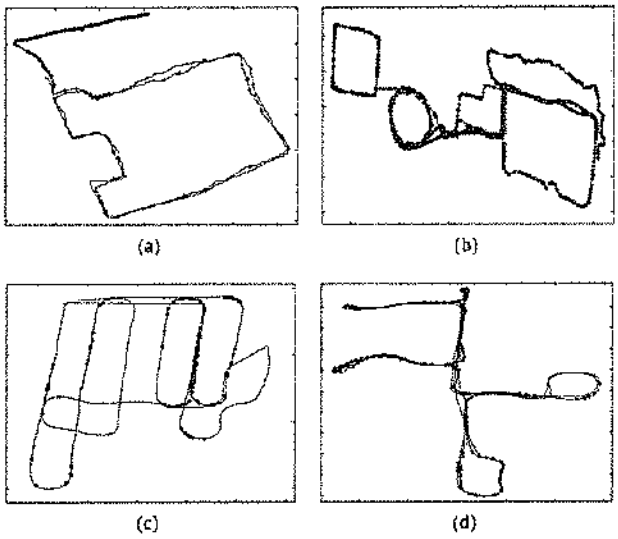


Fig. 8. GPS positions of candidate images (red dots) that are used as seed for the construction of a virtual location on the four tested datasets, namely the City Centre dataset 8a, the New College dataset 8b, the Malaga Parking 6L dataset 8c and the IDOL dum_sunny3 + dum_cloudy1 dataset 8d. Virtual locations are created near curves, 90° angles and stretches traversed in opposite directions or in cases of a severe lateral displacement. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

trajectories for the four test datasets where the red dots represent the GPS coordinates of the candidate images that were used as seeds for virtual locations. It can be observed that virtual locations are created near curves, 90° angles and stretches traversed in opposite directions or in cases of significant lateral displacement. Those points are particularly critical in terms of viewpoint changes, since even small variations in the trajectory (and thus, in the viewpoint) may cause a very different arrangement of the visible landmarks in the acquired scene, thus making the loop closure detection particularly challenging.

4.4.1. Computational complexity

The computational complexity of the procedure for computing a virtual location is quadratic in the number of nodes of the graph, *i.e.*, $O(N^2)$, in the worst case. In the average case the complexity is linear in the number of nodes *i.e.*, $O(N)$. In fact, the actual number of allocated nodes is much less than the product between the number of stored images and the fixed number of patches extracted in each query image *i.e.*, N_P (see Section 3.4). In addition, the number of candidate images is much less than the number of the database images thanks to the selection carried out by the retrieval criterion (see Section 4.1).

The construction of a virtual location is performed for each one of the retrieved candidate images, which is equal to the number of database images in the worst case. The most time consuming part of the virtual location construction algorithm is mainly due to the eigenvector decomposition procedure used to compute the $\mathbf{u}_2$ vector. This procedure is cubic in the number of nodes in the graph, *i.e.*, $O(N^3)$ but it is performed only once for all retrieved candidate images.

A possible strategy to limit the computational load is to use odometry information to "activate" the construction of virtual locations only in particular situations, such as during turns, where they proved to be particularly useful.

5. Experiments and results

In this section we describe the experimental setup and the public datasets selected for testing. In previous works [6,14], the superiority of semi-semantic feature based methods over low-level feature based methods has been clearly shown. For this reason, in this study the analysis is carried out with the purpose of highlighting the importance and the role of the different blocks of the overall algorithm based on semi-semantic features, and to perform a deep experimental evaluation of the performance in different operative scenarios.

5.1. Tuning and validation datasets

The parameters of the proposed algorithm were tuned on the Gardens Point day-left and day-right dataset used, for example, in [6]. To achieve a fair comparison, this dataset was not used for testing. This dataset presents both indoor and outdoor sections, repeating patterns along the path, dynamic objects such as pedestrians, many corners and curves along the trajectory, illumination condition variations such as shadows and sunlight and a typical scenario of viewpoint variation such as lateral displacement.

The main purpose of the tuning phase is the setting of the threshold value defining the minimum similarity score between landmarks in order to map them in a unique node (see Section 3.3). This threshold is set to 0.3 in our implementation. In light of the considerations made in Section 3.3, the selected value for the threshold value was deemed to provide a reasonable trade-off between speed and accuracy.

The performance evaluation was carried out using the following four public datasets.

City centre dataset.

This dataset [3] consists of left and right view images collected "roughly" with a spatial frequency of 1.5 m by a Segway robot along a 2 km path in a urban environment. Right and left images are acquired at the same time, thus we concatenated each pair and considered the new "panoramic" images in our experiments. This dataset is characterized by the presence of dynamic objects such as pedestrians and vehicles, mild illumination variation mainly due to shadows and sunlight and mild viewpoint variation due to lateral displacement while traversing the same path.

New college dataset.

This dataset [3] consists of left and right images collected with a spatial frequency of 1.5 m by a Segway robot along a 1.9 km path in a university campus. Since independent right and left images are acquired also in this case, we concatenated each pair and considered the new "panoramic" images in our experiments. The trajectory is articulated and presents many loops and straight segments traversed also in opposite directions. Also this dataset contains many dynamic elements, such as pedestrians, and repeated elements, since it was acquired in an area characterized by similar walls, archways and bushes.

Malaga Parking 6l dataset.

This dataset [41] was acquired in a university parking area using an electric car equipped with two Firewire colour cameras. For our experiments we considered the rectified images of the left camera. The sequence of images was subsampled at sampling rate 3, thus retaining a third of the entire number of images in the sequence. The explored area covers about 17 920 m^2 and images used here are taken every 0.4 s. The environment of this dataset presents moving vehicles and pedestrians and significant sunlight variations. The trajectory presents many loops, stretches traversed in opposite directions and many intersections, thus viewpoint changes are particularly severe in this dataset.

Table 2
Radius and minimum difference between indices used for ground truth construction for each test dataset.

Dataset	Radius [m]	Min. indices difference
City Centre	10	40
New College	10	40
Malaga Parking 6L	2	135
IDOL dum_sunny3 + dum_cloudy1	1.5	300

IDOL dum_sunny3 + dum_cloudy1 dataset.

This dataset [9] was acquired in a research laboratory consisting of five rooms, in different seasons, hours of the day and weather conditions, by a PowerBot robot equipped with a monocular camera whose height above the floor is 36 cm. In order to have significant illumination variation, we concatenated two sequences one taken on a sunny summer day and the other n a cloudy winter day. The two sequences have been concatenated after subsampling them at sampling rate 3, thus retaining a third of the entire number of images in each sequence. The same trajectory is traversed twice, with mild differences that however produce critical viewpoint changes in an indoor environment.

5.1.1. Ground truth

Although some of the above public datasets provide image matching information, it was decided to recompute the image matching matrix in order to use a consistent criterion for all the considered datasets.

The ground truth was computed on the basis of the GPS coordinates of the images. Namely, we considered two images to be matching if they were acquired within a small distance radius. Further, to avoid "trivial matchings" between consecutive images, a minimum difference between the index value of the matching images was also defined. In fact, it is obvious that the most similar images to the current query image are the ones acquired immediately before, but this similarity should be disregarded in the procedure for loop closure detection.

The IDOL dum_sunny3 + dum_cloudy1 dataset is the only indoor dataset that was used in our experiments. Due to the significant viewpoint variation caused by even small trajectory variations, for this indoor dataset we decided to match images of the first traversal with those of the second traversal. Thus, we imposed a minimum difference between matching images equal to 300, so that only images belonging to different traversals are considered.

Table 2 reports the parameters that were used for the computation of the ground truth for each dataset.

5.2. Plan for the experiments

In order to evaluate the performance of the different blocks of the proposed algorithm and to compare the overall performance with those of a state-of-the-art method we considered the following scenarios:

- A state-of-the-art technique that is based on the high level features extracted by Edge Boxes and AlexNet conv3, that are also used in our work, but does not use any graph based representation of the environment (named 'HOCE' – Heap Of CNN Extracted features – in this work). This is essentially the approach proposed by Sunderhauf et al. in [6]. This algorithm was here re-implemented and used in an incremental fashion to be consistent with our approach.
- Our complete approach (named 'GOCCE' – Graph Of Covisible CNN Extracted features), that exploits the covisibility graph as described in Section 4.

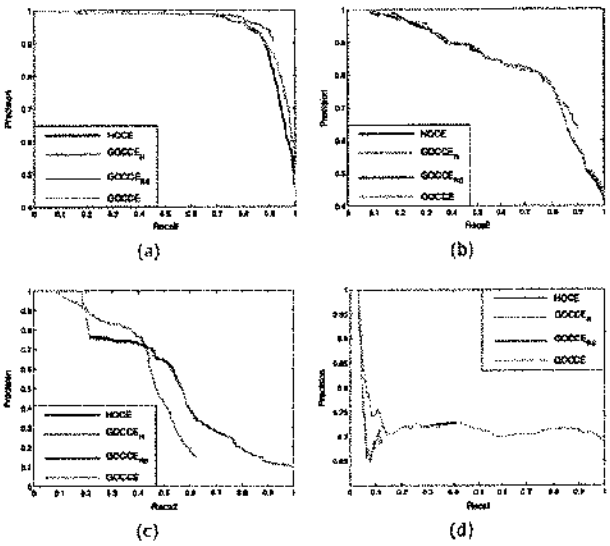


Fig. 9. Precision–recall curves comparing the different techniques with our novel approach on the four test datasets, namely the City Centre dataset 9a, the New College dataset 9b, the Malaga Parking 6L dataset 9c and the IDOL dum_sunny3 + dum_cloudy1 dataset 9d.

- A simplified version of the approach (named 'GOCCE$_R$') that uses only the covisibility graph for the retrieval of matching candidates, selected in case they share at least 10 nodes with the current query image (in the following this criterion will also be referred to as 'strict retrieval').
- Another simplified version of our approach (named 'GOCCE$_{RS}$') that uses the covisibility graph for matching candidates retrieval, selected if they share at least a node with the current query image, and for refining the matching score of candidate locations via subgraph comparison.

As for the settings, we used for each scenario the same values for the maximum number of patches extracted in each image and for the minimum similarity score between landmarks in order to be included in the same node.

5.3. Performance analysis

In a localization and mapping application, the loop closure detection module is essential since it allows an autonomous agent to self-relocalize and to adjust the map of the environment. This section reports the results of a detailed study that is mainly focused toward the evaluation of the loop closure detection performance of the proposed method. Considering a generic loop closing problem, it is generally more important to avoid wrong matchings along the trajectory, rather than not to miss a matching, *i.e.*, precision is usually a more critical requirement then recall.

To have a synthetic comparison of the performance provided by the considered variants of our method, in Fig. 9 the precision–recall curves obtained on the four test datasets are reported, while Table 3 shows the precision and recall values obtained at maximum recall and precision respectively. It is observed that in the case of strict retrieval (*i.e.*, for GOCCE$_R$) the precision is higher in every dataset (note in particular the performance for the City Centre dataset in Fig. 9a and for the Malaga Parking 6L dataset in Fig. 9c). This is mainly due to the fact that the strict retrieval criterion excludes a priori many true negative matchings, thus the precision is higher (see Section 4.1). The main drawback of this approach is that 100% recall is never reached. This is because true positive

Table 3
Precision and recall values at maximum recall and precision respectively comparing the different techniques on the four considered datasets.

	City Centre		New College		Malaga Parking 6L		IDOL dum_sunny3+dum_cloudy1	
	Recall at 100% precision	Precision at max recall	Recall at 100% precision	Precision at max recall	Recall at 100% precision	Precision at max recall	Recall at 100% precision	Precision at max recall
HOCE	15.64%	45.18%	10.09%	41.40%	08.17%	00.56%	03.44%	68.73%
$GOCCE_R$	15.64%	90.89% at 91.74%	10.30%	63.44% at 90.31%	18.27%	15.12% at 62.50%	03.13%	12.09% at 68.52%
$GOCCE_{RS}$	16.18%	45.85%	07.30%	43.21%	08.17%	00.56%	03.44%	68.73%
GOCCE	16.00%	45.68%	10.30%	43.53%	08.17%	00.56%	03.44%	68.73%

matchings with lower matching scores (that are considered by the other analysed methods) are a priori excluded by $GOCCE_R$. An important difference between the graph-free approach (*i.e.*, HOCE) and the graph-based approaches with unselective retrieval criterion (*i.e.*, $GOCCE_{RS}$ and GOCCE) was also observed. In fact, especially on the City Centre Dataset the precision obtained by HOCE at high recall is almost 10% inferior to the precision achieved by $GOCCE_{RS}$ and GOCCE. This fact confirms clearly the beneficial role of the subgraph matching score (Eq. (7)) as additional information to refine the overall matching score between images. Performance obtained by HOCE in the remaining datasets was comparable (just slightly inferior) to that of $GOCCE_{RS}$ and GOCCE.

To evaluate the performance of the loop closing module it is also important to evaluate the metric error produced by wrong matches. Indeed, in order to build a consistent map of the environment, a wrong loop closure detection can be considered somewhat useful if the metric error is small. In fact, it is reasonable that images having a similar visual content are acquired at close distance each other, thus the localization error produced by their matching can be considered acceptable for a coarse localization. In other words, errors of a few metres can still allow a reliable localization producing a consistent map of the environment. The results of the metric study are reported in Fig. 10, which shows the average metric error, *i.e.*, the average Euclidean distance between coordinates of false positive matching images, as a function of the threshold value applied to the matching score for assessing a loop closure. Analysing Fig. 10 it can be observed that a low threshold leads mainly to spatially distant false positive matches, while large threshold values do not produce false positive matches (this implies a high precision). Some differences among the methods were highlighted by this metrical study: the higher precision of the $GOCCE_R$ approach is confirmed also in metric terms, while the method of Sunderhauf et al. in [6] (HOCE) produces less precise results compared to the methods exploiting the covisibility graph, especially in metric terms.

Finally, to evaluate the role of the virtual locations, we carried out an additional study considering only candidate images that served as seed for the construction of a virtual location (shown in Fig. 8). In other words, we considered only those matches between a query image and candidate images that included nodes from other images, *i.e.*, were used as seeds for a virtual location. Precision–recall curves obtained considering only this subset of images are reported in Fig. 11. It may be observed that, especially in the Malaga Parking 6L dataset (Fig. 11c) our complete approach, GOCCE, obtains good performances thanks to the virtual locations construction function (note the difference with respect to the $GOCCE_{RS}$ approach that does not calculate virtual locations). The Malaga Parking 6L dataset exhibits an articulated trajectory, with many curves, intersections and stretches traversed in opposite directions. In these critical scenarios a localization and mapping system can benefit from the virtual location construction in terms of loop closure detection performance.

In light of this, we use the Malaga Parking 6L dataset for an additional study to evaluate the benefits of the virtual locations in terms

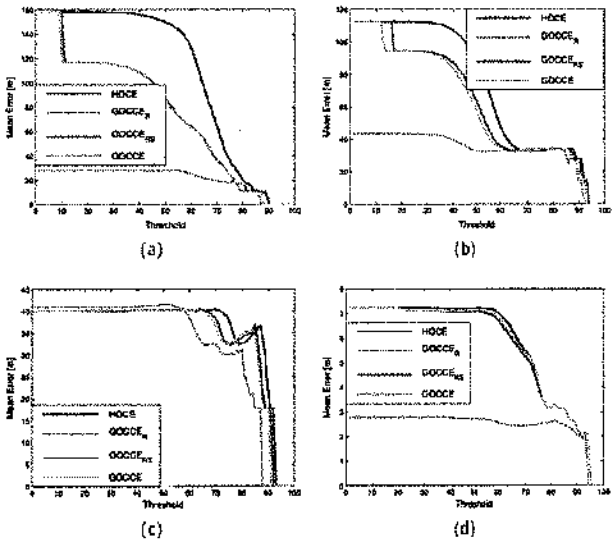


Fig. 10. Average metric error curves, relative to false positive matching errors comparing the different techniques on the four considered datasets, namely the City Centre dataset 10a, the New College dataset 10b, the Malaga Parking 6L dataset 10c and the IDOL dum_sunny3 + dum_cloudy1 dataset 10d.

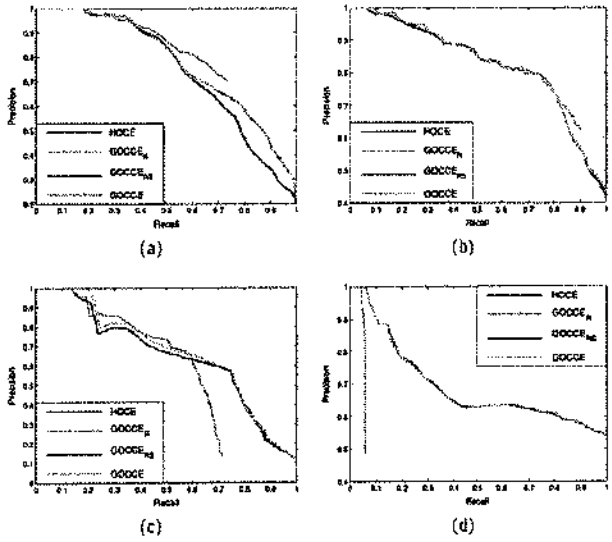


Fig. 11. Precision–recall curves comparing the different techniques with respect to our novel approach considering only candidate images that were used as seed for the construction of a virtual location on the four tested datasets, namely the City Centre dataset 11a, the New College dataset 11b, the Malaga Parking 6L dataset 11c and the IDOL dum_sunny3 + dum_cloudy1 dataset 11d.

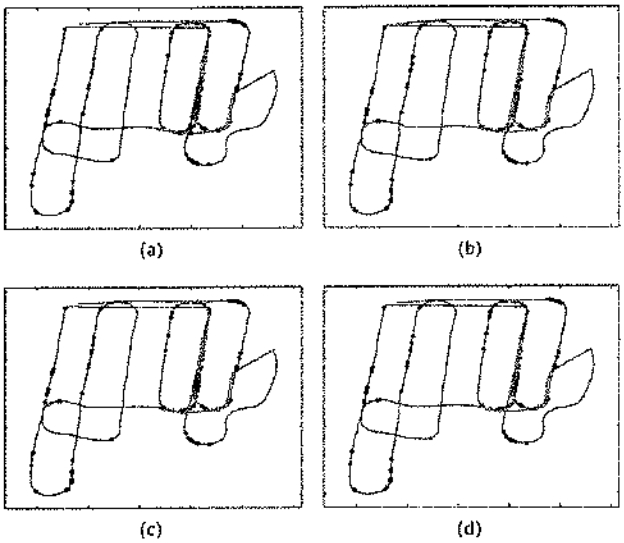


Fig. 12. GPS position of candidate images that produced a loop closure detection error on the Malaga Parking 6L dataset for the four considered methods: HOCE 12a, $GOCCE_R$ 12b, $GOCCE_{RS}$ 12c and GOCCE 12d. Black dots represent GPS positions of correctly matched images, green dots are GPS positions of false negative matching images and red dots are GPS positions of false positive matching images. Note that GOCCE has the smallest number of false positive matches. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

of loop closure detection performance. In particular, we consider the GPS position where virtual locations have been constructed, these are shown with dots in Fig. 12. For each one of the four methods under investigation, the threshold on the matching score is set at a value that guarantees at least 85% of precision and 20% of recall. With these settings, the output of the loop closure detection methods is evaluated. In Fig. 12, GPS positions of virtual locations are marked with coloured dots along the paths. In particular, black dots are used in the case of a correct output (i.e., true positives or true negatives), green dots in the case of a false negative output and red dots in the case of a false positive output. It can be observed that GOCCE produces the smallest number of false positive matches: 4 FP against 6 FP for $GOCCE_R$, 7 FP for $GOCCE_{RS}$ and 27 FP for HOCE. These results highlight the fact that virtual locations are useful in scenarios that are particularly challenging in terms of viewpoint changes, such as curves and oppositely traversed stretches. Exploiting virtual locations in these cases makes the loop closure detection system more precise.

6. Conclusion

In this work, we proposed an appearance and viewpoint invariant place recognition system. The method relies only on machine vision images and does not need any specific training when operating in new unexplored environments.

These characteristics are achieved by modelling inter object geometric relations in the environment by means of a covisibility graph, whose nodes are high level, semi-semantic landmarks. These landmarks are image patches containing generic objects and are described by means of features extracted by an inner convolutional layer of a pre-trained CNN, that are particularly robust to appearance changes.

We proposed novel specific algorithms that leverage the covisibility graph representation for a fast and robust retrieval of the most likely matching candidate images. The covisibility graph is also exploited for refining images matching score based on the co-presence of landmark contained in the images. We also proposed a novel strategy for synthesizing virtual locations via a parameter-free approach that is based on a local graph clustering method which exploits covisibility graph connectivity information.

Experimental validation carried out on four public datasets has shown that, with regard to precision and recall, our approach provides performance that is comparable (or superior) with respect to a state-of-the-art place recognition technique that does not rely on any graph representation of the environment.

In addition, the construction of virtual locations is useful in specific but critical situations such as turning near 90° corners or traversing a stretch in opposite directions. In these scenarios, virtual locations construction provides an improvement in terms of precision of the loop closure detection system.

Considering metric error (i.e., the metric distance between mismatched images' coordinates), our graph-based technique outperformed a state-of-the-art graph-free approach that was considered as benchmark.

A possible extension of this work would be the implementation of a strategy that compares sequences of images, rather than single images. This directly translates in the comparison of bigger subgraphs.

References

[1] J. Engel, T. Schöps, D. Cremers, LSD-SLAM: Large-scale direct monocular SLAM, in: European Conference on Computer Vision, ECCV, 2014, pp. 834–849.
[2] C. Forster, M. Pizzoli, D. Scaramuzza, SVO: Fast semi-direct monocular visual odometry, in: IEEE International Conference on Robotics and Automation, ICRA, 2014.
[3] M. Cummins, P. Newman, Fab-map: Probabilistic localization and mapping in the space of appearance, Int. J. Robot. Res. 27 (6) (2008) 647–665.
[4] M.J. Milford, G.F. Wyeth, Seqslam: Visual route-based navigation for sunny summer days and stormy winter nights, in: Robotics and Automation, ICRA, 2012 IEEE International Conference on, IEEE, 2012, pp. 1643–1649.
[5] T.A. Ciarfuglia, G. Costante, P. Valigi, E. Ricci, A discriminative approach for appearance based loop closing, in: IEEE International Conference on Intelligent Robots and Systems, IROS, 2012.
[6] N. Sunderhauf, S. Shirazi, A. Jacobson, F. Dayoub, E. Pepperell, B. Upcroft, M. Milford, Place recognition with convnet landmarks: Viewpoint-robust, condition-robust, training-free, Proceedings of Robotics: Science and Systems XII.
[7] R. Finman, L. Paull, J.J. Leonard, Toward object-based place recognition in dense rgb-d maps, in: ICRA Workshop Visual Place Recognition in Changing Environments, Seattle, WA, 2015.
[8] J.Oh, J. Jeon, B. Lee, Place recognition for visual loop-closures using similarities of object graphs, Electron. Lett. 51 (1) (2014) 44–46.
[9] J. Luo, A. Pronobis, B. Caputo, P. Jensfelt, Incremental learning for place recognition in dynamic environments, in: 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, 2007, pp. 721–728.
[10] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in Neural Information Processing Systems, NIPS, 2012.
[11] C. Mei, G. Sibley, P. Newman, Closing loops without places, in: Intelligent Robots and Systems, IROS, 2010 IEEE/RSJ International Conference on, IEEE, 2010, pp. 3738–3744.
[12] E.S. Stumm, C. Mei, S. Lacroix, Building location models for visual place recognition, Int. J. Robot. Res. 35 (4) (2016) 334–356.
[13] E. Stumm, C. Mei, S. Lacroix, M. Chli, Location graphs for visual place recognition, in: Robotics and Automation, ICRA, 2015 IEEE International Conference on, IEEE, 2015, pp. 5475–5480.
[14] S. Cascianelli, G. Costante, E. Bellocchio, P. Valigi, M.L. Fravolini, T.A. Ciarfuglia, A robust semi-semantic approach for visual localization in urban environment, in: Smart Cities Conference, ISC2, 2016 IEEE International, IEEE, 2016, pp. 1–6.
[15] C. McManus, W. Churchill, W. Maddern, A.D. Stewart, P. Newman, Shady dealings: Robust long-term visual localisation using illumination invariance, in: Robotics and Automation, ICRA, 2014 IEEE International Conference on, IEEE, 2014, pp. 901–906.

[16] P. Neubert, N. Sünderhauf, P. Protzel, Superpixel-based appearance change prediction for long-term navigation across seasons, Robot. Auton. Syst. 69 (2015) 15–27.
[17] N. Carlevaris-Bianco, R.M. Eustice, Learning visual feature descriptors for dynamic lighting conditions, in: Intelligent Robots and Systems, IROS, 2014, 2014 IEEE/RSJ International Conference on, IEEE, 2014, pp. 2769–2776.
[18] N. Sunderhauf, S. Shirazi, F. Dayoub, B. Upcroft, M. Milford, On the performance of convnet features for place recognition, in: Intelligent Robots and Systems, IROS, 2015 IEEE/RSJ International Conference on, IEEE, 2015, pp. 4297–4304.
[19] W. Maddern, A. Stewart, C. McManus, B. Upcroft, W. Churchill, P. Newman, Illumination invariant imaging: Applications in robust vision-based localisation, mapping and classification for autonomous vehicles, in: Proceedings of the Visual Place Recognition in Changing Environments Workshop, IEEE International Conference on Robotics and Automation, ICRA, Hong Kong, China, 2014.
[20] Z. Chen, O. Lam, A. Jacobson, M. Milford, Convolutional neural network-based place recognition, arXiv preprint arXiv:1411.1509.
[21] H. Bay, A. Ess, T. Tuytelaars, L. Van Gool, Speeded-up robust features (surf), Comput. Vis. Image Underst. 110 (3) (2008) 346–359.
[22] R. Mur-Artal, J. Montiel, J.D. Tardós, Orb-slam: a versatile and accurate monocular slam system, IEEE Trans. Robot. 31 (5) (2015) 1147–1163.
[23] E. Rublee, V. Rabaud, K. Konolige, G. Bradski, Orb: An efficient alternative to sift or surf, in: 2011 International conference on computer vision, IEEE, 2011, pp. 2564–2571.
[24] H. Yang, S. Cai, J. Wang, L. Quan, Low-rank sift: an affine invariant feature for place recognition, in: Image Processing, ICIP, 2014 IEEE International Conference on, IEEE, 2014, pp. 5731–5735.
[25] M. Milford, C. Shen, S. Lowry, N. Suenderhauf, S. Shirazi, G. Lin, F. Liu, E. Pepperell, C. Lerma, B. Upcroft, Sequence searching with deep-learnt depth for condition-and viewpoint-invariant route-based place recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2015, pp. 18–25.
[26] R. Arroyo, P.F. Alcantarilla, L.M. Bergasa, J.J. Yebes, S. Gámez, Bidirectional loop closure detection on panoramas for visual navigation, in: Intelligent Vehicles Symposium Proceedings, 2014 IEEE, IEEE, 2014, pp. 1378–1383.
[27] A.L. Majdik, D. Verda, Y. Albers-Schoenberg, D. Scaramuzza, Air-ground matching: Appearance-based gps-denied urban localization of micro aerial vehicles, J. Field Robot. 32 (7) (2015) 1015–1039.
[28] R. Arandjelović, P. Gronat, A. Torii, T. Pajdla, J. Sivic, Netvlad: Cnn architecture for weakly supervised place recognition, arXiv preprint arXiv:1511.07247.
[29] D. Mishkin, M. Perdoch, J. Matas, Place recognition with wxbs retrieval, in: CVPR 2015 Workshop on Visual Place Recognition in Changing Environments, 2015.
[30] E. Pepperell, P. Corke, M. Milford, Routed roads: Probabilistic vision-based place recognition for changing conditions, split streets and varied viewpoints, Int. J. Robot. Res. (2016).
[31] G. Costante, T.A. Ciarfuglia, P. Valigi, E. Ricci, A transfer learning approach for multi-cue semantic place recognition, in: Intelligent Robots and Systems, IROS, 2013 IEEE/RSJ International Conference on, IEEE, 2013, pp. 2122–2129.
[32] C.L. Zitnick, P. Dollár, Edge boxes: Locating object proposals from edges, in: Computer Vision–ECCV 2014, Springer, 2014, pp. 391–405.
[33] A. Sharif Razavian, H. Azizpour, J. Sullivan, S. Carlsson, Cnn features off-the-shelf: an astounding baseline for recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2014, pp. 806–813.
[34] J.L. Long, N. Zhang, T. Darrell, Do convnets learn correspondence? in: Advances in Neural Information Processing Systems, 2014, pp. 1601–1609.
[35] E.J. Candes, T. Tao, Near-optimal signal recovery from random projections: Universal encoding strategies? IEEE Trans. Inform. Theory 52 (12) (2006) 5406–5425.
[36] M. Muja, D.G. Lowe, Fast approximate nearest neighbors with automatic algorithm configuration, in: International Conference on Computer Vision Theory and Application VISSAPP'09), INSTICC Press, 2009, pp. 331–340.
[37] S. Feizi, G. Quon, M. Medard, M. Kellis, A. Jadbabaie, Spectral alignment of networks.
[38] S.E. Schaeffer, Graph clustering, Comput. Sci. Rev. 1 (1) (2007) 27–64.
[39] C. Godsil, G.F. Royle, Algebraic Graph Theory, vol. 207, Springer Science & Business Media, 2013.
[40] M. Newman, Networks: An Introduction, Oxford university press, 2010.
[41] J.-L. Blanco, F.-A. Moreno, J. González, A collection of outdoor robotic datasets with centimeter-accuracy ground truth, Auton. Robots 27 (4) (2009) 327–351, http://dx.doi.org/10.1007/s10514-009-9138-7, http://www.mrpt.org/Paper:Malaga_Dataset_2009.

Silvia Cascianelli received the B.Sc. degree in Electronic and Information Engineering in 2013, from University of Perugia, with a thesis on System Fault Detection and Accommodation for UAV's anemometers. Since then she collaborates with the Intelligent Systems, Automation and Robotics Laboratory (ISARLab). In 2015 she received the M.Sc. *magna cum laude* degree in Information and Automation Engineering with a thesis on Nuclear Image based Computer Aided Diagnosis systems for Alzheimer's Disease from the University of Perugia. She then joined the ISARLab in 2015 as a Ph.D. student. Her research interests are mainly machine learning and computer vision applied to robotics.

Gabriele Costante received the B.Sc. *magna cum laude* degree in Electronic and Information Engineering and the M.Sc. *magna cum laude* degree in Information and Automation Engineering from the University of Perugia respectively in 2010 and 2012. He then joined the Service and Industrial Robotics and Automation Laboratory (SIRALab) in 2012 and in 2016 he received the Ph.D. degree in Robotics from the University of Perugia. His research interests are mainly robotics, computer vision and machine learning.

Enrico Bellocchio received his B.Sc. degree in Electronic and Information Engineering in 2012 from University of Perugia. In 2014 received M.Sc. degree in Information and Automation Engineering From University of Perugia. His Master thesis work was about the implementation of an Human–Robot Interface using a Transfer Learning algorithm on a UAV platform. He joined the Intelligent Systems Automation and Robotics Laboratory (ISARLab) in 2014 as a researcher engineer. His current research activity is about machine learning, computer vision and robotics.

Paolo Valigi received the Laurea degree in 1986 from University of Rome La Sapienza and the Ph.D. degree from University of Rome Tor Vergata in 1991. From 1990 to 1994 he worked with the Fondazione Ugo Bordoni. From 1998 to 2004 he was Associate Professor at the University of Perugia, Department of Electronics and Informatics Engineering, where since 2004 he has been full Professor of System Theory and Optimization and Control. His research interests are in the field of robotics and systems biology. He has authored or co-authored more than 130 journal and conference 810 papers and book chapters.

Mario Luca Fravolini received his Ph.D. degree in Electronic Engineering from the University of Perugia in 2000. Currently he is Associate Professor at the Department of Engineering, University of Perugia. In 1999 he was with the Control Group at the School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, USA. He has been a visiting Research Assistant Professor at the Department of Mechanical and Aerospace Engineering West Virginia University, USA for several years. His research interests include: Fault Diagnosis, Intelligent and Adaptive Control, Predictive Control, Optical Feedback, Biomedical Imaging, Modelling and Control Biomedical Systems and Active Control of Structures.

Thomas Alessandro Ciarfuglia received the M.Sc. *magna cum laude* degree in Electronics Engineering from the University of Perugia in 2004. He worked as HW/FW/SW designer Engineer for various companies from 2004 to 2006. He then obtained an M.Sc. in Mechatronics and a Ph.D. degree in Robotics from the University of Perugia in 2008 and 2011 respectively. He joined the Service and Industrial Robotics and Automation Laboratory (SIRALab) in 2008 and he is currently working as a PostDoc there. His research interests are machine learning and computer vision applied to robotics.

Toward Domain Independence for Learning-Based Monocular Depth Estimation

Michele Mancini, Gabriele Costante, Paolo Valigi, Thomas A. Ciarfuglia, Jeffrey Delmerico, and Davide Scaramuzza

***Abstract*—Modern autonomous mobile robots require a strong understanding of their surroundings in order to safely operate in cluttered and dynamic environments. Monocular depth estimation offers a geometry-independent paradigm to detect free, navigable space with minimum space, and power consumption. These represent highly desirable features, especially for microaerial vehicles. In order to guarantee robust operation in real-world scenarios, the estimator is required to generalize well in diverse environments. Most of the existent depth estimators do not consider generalization, and only benchmark their performance on publicly available datasets after specific fine tuning. Generalization can be achieved by training on several heterogeneous datasets, but their collection and labeling is costly. In this letter, we propose a deep neural network for scene depth estimation that is trained on synthetic datasets, which allow inexpensive generation of ground truth data. We show how this approach is able to generalize well across different scenarios. In addition, we show how the addition of long short-term memory layers in the network helps to alleviate, in sequential image streams, some of the intrinsic limitations of monocular vision, such as global scale estimation, with low computational overhead. We demonstrate that the network is able to generalize well with respect to different real-world environments without any fine tuning, achieving comparable performance to state-of-the-art methods on the KITTI dataset.**

***Index Terms*—Collision avoidance, range sensing, visual-based navigation.**

I. Introduction

As AUTONOMOUS vehicles become more common in many applications outside the research laboratory, the requirements for safe and optimal operation of such systems become more challenging. In particular, the ability to detect and avoid still or mobile obstacles is crucial for field operations of the vast majority of ground and low altitude flight vehicles. Depth information can be used to estimate proximity to obstacles and, in general, to obtain an understanding of the surrounding 3D space. This perception of the 3D environment can then be used in reactive [1] or planned [2] control strategies to navigate safely. LIDAR and sonar sensors can provide sparse 3D information, but their installation may be costly, in terms of weight, space and power, all of which are constraints for mobile robots, and especially Micro Aerial Vehicles (MAVs). Vision-based systems, both mono and stereo, can provide dense depth maps and are more suitable for deploying on small vehicles. A primary shortcoming, though, is that the detection range and accuracy of stereo cameras are limited by the camera set-up and baseline [3], [4]. Exploiting geometric constraints on camera motion and planarity, obstacle detection and navigable ground space estimation can be extended far beyond the normal range (see [5] and [6]). However, these constraints hold mostly for ground, automotive applications, but do not generalize to MAVs.

Differently from stereo systems, monocular systems do not make specific assumptions on camera motion or set-up. Several monocular depth estimation methods have been proposed in recent years, mostly exploiting machine learning paradigms ([7]–[11]). The advantages of such systems are that they are able to learn scale without the use of external metric information, such as Inertial Measurement Unit (IMU) measurements, and are not subject to any geometrical constraint. On the downside, these systems rely on the quality and variety of the training set and ground truth provided, and often are not able to adapt to unseen environments.

The challenge of domain independence is one of the main obstacles to extensive use of learned monocular systems in place of stereo geometrical ones. The question of how does these systems perform in uncontrolled, previously unseen scenarios can be addressed by learning features that are more invariant to environment changes and also by using different network architectures that are able to learn more general models from the training samples they have. Unfortunately, there are only a few labeled depth datasets with the required ground truth density, and the cost and time required to create new ones is high.

In our previous work [12] we showed that training a Convolutional Neural Network (CNN) with a inexpensive generated, densely-labeled, synthetic urban dataset, achieved promising results on the KITTI dataset benchmark using RGB and optical flow inputs.

In this work, by using a deeper architecture and an extended synthetic dataset able to generalize from synthetic data to real

Manuscript received September 10, 2016; accepted January 9, 2017. Date of publication January 23, 2017; date of current version June 5, 2017. This letter was recommended for publication by Associate Editor J. L. Blanco-Claraco and Editor C. Stachniss upon evaluation of the reviewers' comments. This work was supported in part by the DARPA FLA Program and in part by the M.I.U.R. (Ministero dell'Istruzione dell'Università e della Ricerca) under Grant SCN_398/SEAL (Program Smart Cities).

M. Mancini, G. Costante, P. Valigi, and T. A. Ciarfuglia are with the Department of Engineering, University of Perugia, 06123 Perugia, Italy (e-mail: michele.mancini@unipg.it; gabriele.costante@unipg.it; thomas.ciarfuglia@unipg.it; paolo.valigi@unipg.it).

J. Delmerico and D. Scaramuzza are with the Robotics and Perception Group, University of Zurich, 8006 Zürich, Switzerland (e-mail: jeffdelmerico@ifi.uzh.ch; sdavide@ifi.uzh.ch).

Color versions of one or more of the figures in this letter are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/LRA.2017.2657002

unseen sequences, we take an important step towards domain independence for vision-based depth estimation applications (see Fig. 1). With robotic-based operations in mind, we reduce the computational complexity of the network by removing the network dependence on optical flow, even if it often acts as a environment-invariant feature. To balance this loss of information, we exploit the input stream sequentiality by using Long Short Term Memory (LSTM) layers, a specific form of Recurrent Neural Networks (RNNs).

Our experiments show that this solution significantly outperforms previous results. We validate our model on the KITTI dataset, where we obtain comparable performance to state-of-the-art, specially tuned methods.

We also perform validation on two challenging and different new datasets consisting of sequences captured in a dense forest and in a country road, in order to evaluate possible MAV operation environments. We show how the model is capable of reliable estimation even on video streams with vibration and motion blur, making our model suitable for tasks as obstacle avoidance and motion planning for mobile robots.

II. Related Work

Traditional vision-based depth estimation is based on stereo vision [13]. Its main limitations lie on the lack of robustness on long range measurements and pixel matching errors. This aspect is even more critical in MAV applications where maneuvers are on 6DOF and the lack of platform space makes it difficult to mount a stereo rig with a proper baseline. Finally, weight and power consumption minimization is highly desirable. For these reasons, monocular vision is becoming more and more important when it comes to MAV applications.

Monocular depth estimation based on geometric methods is grounded on the triangulation of consecutive frames. Despite the impressive results achieved by state-of-the-art approaches [14]–[16], the performance of their reconstruction routines drops during high-speed motion, as dense alignment becomes extremely challenging. In addition, it is not possible to recover the absolute scale of the object distances. Driven by the previous considerations, in this work, we address both the aforementioned aspects by exploiting the learning paradigm to learn models that compute the scene depth and the associated absolute scale from a single image (*i.e.* without processing multiple frames).

Learning-based methods for depth estimation have demonstrated good performance in specific scenarios, but these results are limited to these environments, and have not been shown to generalize well. Saxena *et al.* [17] first proposed a Markov Random Field to predict depth from a monocular, horizontally-aligned image, which then later evolved into the Make3D project [10]. This method tends to suffer in uncontrolled settings, especially when the horizontal alignment condition does not hold. Eigen *et al.*([7], [18], exploit for the first time in their work the emergence of Deep Learning solutions for this kind of problems, training a multi scale convolutional neural network (CNN) to estimate depth. Following this, several other CNN-based approaches have been proposed. Liu *et al.* [8] combine a CNN with a Conditional Random Field to improve smoothness. Roy *et al.* [9] recently proposed a novel depth estimation method based on Neural Regression Forest. However, the aforementioned methods [7]–[9], [17], [18] are specific for the scenario where they have been trained and, thus, they are not domain independent.

For our intended embedded application, computational efficiency is very important, and, in this respect, most of the existing methods for monocular depth estimation are not appropriate. In [8] and [9], although they reported slightly improved performances on several benchmarks with respect to Eigen *et al.*'s work, they cannot guarantee real-time performance on embedded hardware. They report a single image inference time of $\sim$1 s both on a GTX780 and a Tesla k80, far more powerful hardware than the ones generally embedded on MAVs. Conversely, Eigen *et al.* method is able to estimate a coarser resolution ($1/4$ of the input image) of the scene depth map with a inference time of about 10 ms. Our system's inference time is less than 30 ms on a comparable hardware (Tesla k40) and less than 0.4 s on an embedded hardware (Jetson TK1), making real-time application feasible. Based on these various factors, we chose the Eigen *et al.* [7] method to serve as a reference to the state of the art during our experiments.

Although we are interested in performing well against the state of the art in accuracy, our primary goal is to develop a robust estimator that is capable of generalizing well to previously unseen environments, in order to be useful in robotic applications. For this reason, we did not perform any finetuning on evaluation benchmarks, focusing on how architectural choices and synthetic datasets generation influence generalization. Our previous work propose a baseline solution to the problem, suggesting a Fully Convolutional Network (FCN) fed with both the current frame and the optical flow between current and previous frame [12]. Despite optical flow acts as a good environment-invariant feature, it is not sufficient to achieve generalization across different scenarios. Furthermore, the computation of the optical flow considerably increase the overall inference time. In this work, only the current frame is fed into the network: by using a deeper architecture and the LSTM paradigm together with a wise mix of different synthetic datasets we report a significant performance gain in a simpler and more efficient fashion.

A relatively unexplored area of research is the training of networks given data scarcity. Recently, Garg *et al.* [11] proposed an unsupervised approach for monocular depth estimation with CNNs. In their work they propose a data augmentation technique to deal with the cost of acquiring real images with depth ground truth. However, the augmented dataset has to be generated from already acquired images, and thus this technique is unable to generate unseen environments. For this reason the authors train and test only on the KITTI dataset. Our work is similar to theirs in the aspect of finding ways to effectively augment training data, but is aimed to generalize performances across different environments. We achieve this exploiting synthetic data, for which exact labels are easily generated. Synthetic training sets are able to represent any kind of scenario, illumination conditions, motion trajectory and camera optics, without any limitation imposed by real world data collection equipments. This allows us to reach good performance on dif-

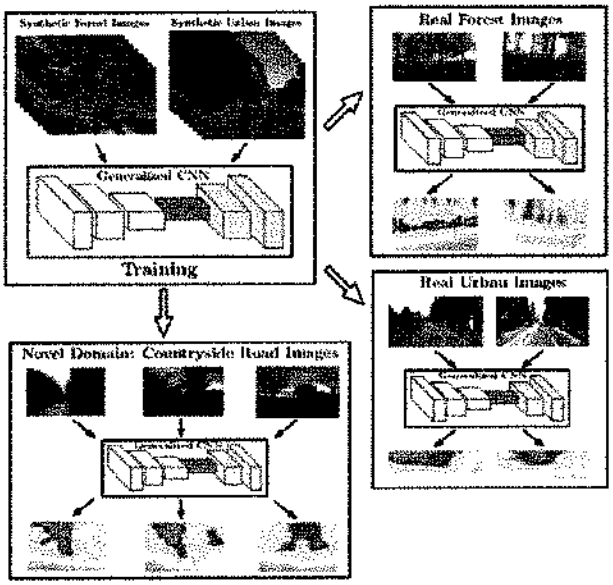


Fig. 1. Overview of the proposed domain independent approach for monocular depth estimation based on CNN. We first train our model on labeled synthetic data. We then deploy it for evaluation on real world scenarios. Our experiments show how the model is able to generalize well across different scenarios without requiring any domain specific fine-tuning procedures.

ferent domains, using different training and test images, and not requiring fine-tuning. However, at the time of the writing of this work, the authors of [11] did not yet make their trained model publicly available for an effective comparison.

III. Network Overview

A. Fully Convolutional Network

We propose as a baseline method a fully convolutional architecture, structured in a encoder-decoder fashion, as depicted in Fig. 2. This a very popular architectural choice for several pixel-wise prediction tasks, as optical flow estimation [19] or semantic segmentation [20]. In our proposed network, the encoder section corresponds to the popular VGG network [21], pruned of its fully connected layers.

We initialize the encoder weights with the VGG pre-trained model for image classification. Models trained on huge image classification datasets, as [22], proved to act as a great generic-purpose feature extractor [23]: low-level features are extracted by convolutional layers closer to the input layer of the net, while layers closer to the output of the net extract high-level, more task-dependent descriptors. During training, out of the 16 convolutional layers of the VGG net, the weights of the first 8 layers are kept fixed; remaining layers are fine-tuned. The decoder section of the network is composed by 2 deconvolutional layers and a final convolutional layer which outputs the predicted depth at original input resolution. These layers are trained from scratch, using random weight initialization.

B. Adding LSTM Layers into the Picture

Any monocular, single image depth estimation method suffers from the infeasibility of correctly estimating the global scale of the scene. Learning-based methods try to infer global scale from the learned proportions between depicted objects in the training dataset. This paradigm inevitably fails when previously unseen environments are evaluated or when the camera focal length is modified.

We can try to correct these failures by exploiting the sequential nature of the image stream captured by a vision module mounted on a deployed robot. Recurrent neural networks (RNN) are typically used in tasks where long-term temporal dependencies between inputs matter when it comes to performing estimation: text/speech analysis, action recognition in a video stream, person re-identification [24]–[26]. Their output is a function of both the current input fed into the network and the past output, so that memory is carried forward through time as the sequence progresses:

$$\mathbf{y}_t = f(\mathbf{W}\mathbf{x}_t + \mathbf{U}\mathbf{y}_{t-1}) \tag{1}$$

where W represents the weight matrix (as in common feedforward networks) and U is called *transition matrix*.

LSTMs are a special kind of recurrent neural network introduced by Hochreiter & Schmidhuber in 1997 to overcome some of the RNN main issues, as vanishing gradients during training, which made them very challenging to use in practical applications [27]. Memory in LSTMs is maintained as a gated cell where information can be read, written or deleted. During training, the cell learns autonomously how to treat incoming and stored information. We insert two LSTM layers between the encoder and decoder section of the previously introduced FCN network (see Fig. 3), in a similar fashion of [24]. Our motivation is to refine features extracted by the encoder according to the information stored in the LSTM cells, so that the decoder section can return a more coherent depth estimation. The proposed LSTM network is depicted in Image 3. Dropout is applied before, after and in the middle of the two LSTM layers to improve regularization during training.

C. Training the Networks

We developed two synthetic datasets for learning depth estimation: the *Urban Virtual Dataset (UVD)* and the *Forest Virtual Dataset (FVD)*, producing a total of more than 80 k images (see Fig. 4). We create the scenarios with Unreal Engine, and extract noise-free ground truth depth maps using its tools. To reduce network's output space dimensionality and ease training, we clip the depth maximum range to 40m, although it is theoretically possible to measure depth up to an unlimited range. Different illumination conditions, motion blur, fog, image noise and camera focal lengths can be easily simulated or modified, offering us a great sandbox to inexpensively generate highly informative datasets and high precision ground truths. The camera moves at speeds up to about 15 m/s with six degrees of freedom inside the built scenarios, collecting frames and corresponding depth maps at a resolution of 256 × 160 pixels and a frame rate of 10 Hz. Using these datasets, we trained the following networks:

1) *UVD_FCN:* Fully convolutional network trained on the Urban Virtual Dataset.

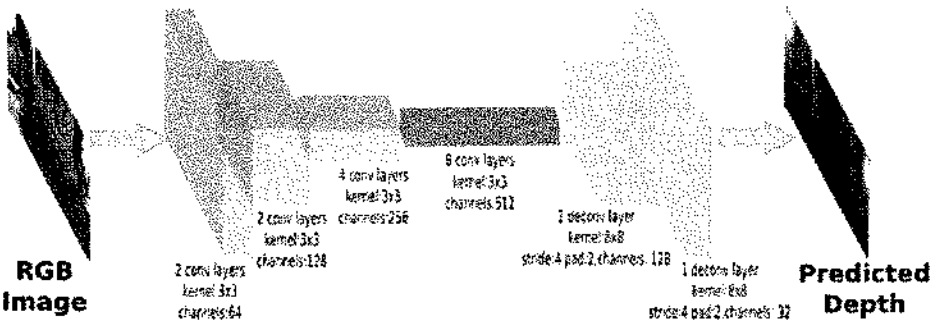


Fig. 2. FCN high-level architecture. Each block represent a set of layers with the depicted specifications. For the encoder section, pooling is applied between each block. Blue boxes: Unchanged VGG encoder layers. Red boxes: Finetuned VGG encoder layers. Green Boxes: Deconvolutional decoder layers.

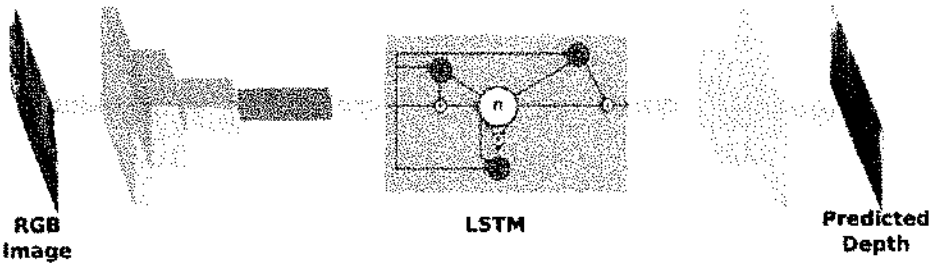


Fig. 3. In our LSTM network, we plug in two LSTM layers with 180 neurons between the encoder and the decoder section of the network.

Fig. 4. Some images from UVD and FVD dataset used for training the models.

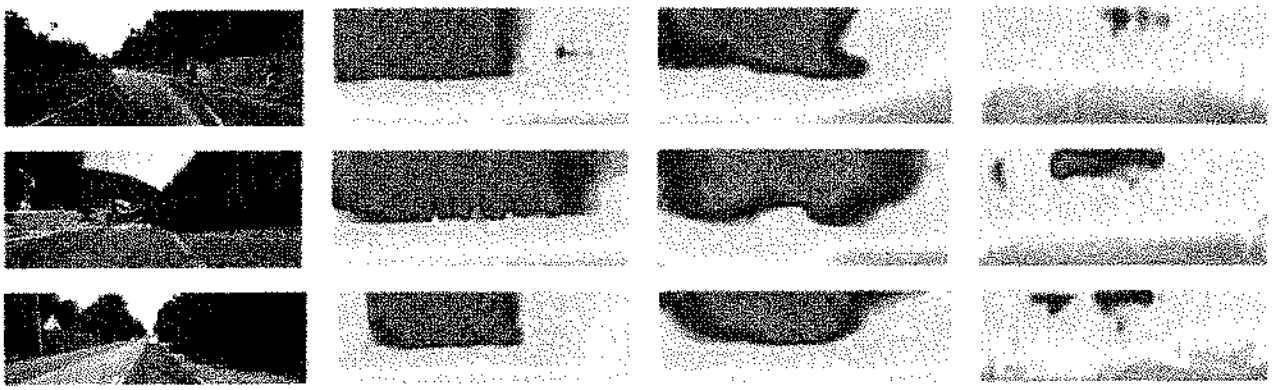


Fig. 5. Qualitative results on the KITTI dataset. On the first column RGB input images are depicted. The second and the third columns show the dense ground truths and MIX_FCN predictions, respectively. The fourth column shows MIX_EIGEN network prediction. Maximum depth range has been trimmed to 40 meters.

2) *FVD_FCN:* Fully convolutional network trained on the Forest Virtual Dataset.
3) *FVD_LSTM:* LSTM network trained on the Forest Virtual Dataset.
4) *MIX_FCN:* Fully convolutional network trained on both Urban and Forest Virtual Datasets.
5) *MIX_LSTM:* LSTM network trained on both Urban and Forest Virtual Datasets.

Networks have been implemented using the Caffe framework and trained on Log RMSE (2) using an Adam solver with a learning rate of $l = 10^{-4}$ until convergence. FCN networks required about 24 hrs for training, while LSTM networks took

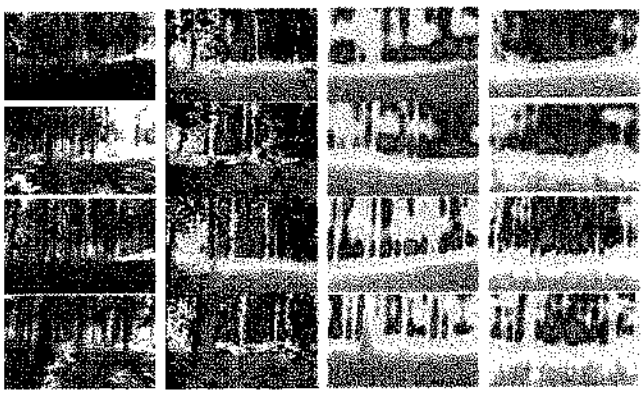


Fig. 6. Qualitative results on the Zurich Forest dataset. On the first column RGB input images are depicted. The second and the third columns show the dense ground truths and MIX_LSTM net predictions, respectively. The fourth column shows MIX_EIGEN network prediction. Maximum depth range has been trimmed to 40 meters. Black pixels in the ground truth represent missing depth measurements.

about 48 hrs on a Tesla K40 GPU.

$$\sqrt{\frac{1}{T}\sum_{Y\in T} ||\log y_i - \log y_i^*||^2}. \quad (2)$$

IV. Experiments

We test generalization capability of our proposed networks on the KITTI dataset [28], and on two datasets we gathered in a dense forest in the surroundings of Zurich, Switzerland and in the countryside near Perugia, Italy, respectively.[1]

We measure our performances with the following metrics:

1) *Threshold error:* % of y_i s.t. $\max(\frac{y_i}{y_i^*}, \frac{y_i^*}{y_i}) = \delta < thr$
2) *Absolute relative difference:* $\frac{1}{T}\sum_{Y\in T}\frac{|y-y^*|}{y^*}$
3) *Log RMSE:* $\sqrt{\frac{1}{T}\sum_{Y\in T} ||\log y_i - \log y_i^*||^2}$
4) *Linear RMSE:* $\sqrt{\frac{1}{T}\sum_{Y\in T} ||y_i - y_i^*||^2}$
5) *Scale-invariant Log MSE (as introduced by [7]):* $\frac{1}{n}\sum_i d_i^2 - \frac{1}{n^2}(\sum_i d_i)^2$, with $d_i = \log y_i - \log y_i^*$

We test on the same benchmark also our previous method proposed in [12], later referred as OPT_FLOW_FCN.

Furthermore, to properly compare our approach with respect to [7], we also implement their coarse+fine network following the details provided by the authors. We train it on both UVD and FVD datasets (*i.e.*, the same training set we use for our networks) with a Scale Inv. Log MSE loss. We first train their coarse model alone for 50 epochs, with a learning rate of 10^{-4}. Afterwards, we keep the weight of the coarse model fixed and train the fine network for about 40 epochs. Their method returns a 4× downsampled depth image, thus, during the evaluation, we upsample the obtained prediction with a nearest neighbor filter to match the original input resolution. In the following, we refer to this baseline as MIX_EIGEN.

Before discussing the results on the real datasets, we run a set of experiments to measure the performance loss when the test domain differs from the training one. In particular, in Table I, we compare the performance of the UVD models evaluated with respect to the urban domain (the same used for training) and the forest one. Similarly, in Table II, we show the results of the FVD networks. Clearly, performance drop when the network is tested on a domain different from the training one (see column 2 of Table I and column 1 of Table II). However, we can observe that extending the training set with images from multiple domains and with the LSTM structure helps the network to considerably increase the generalization capabilities of the CNNs, and as a consequence, the performance.

[1]Link to code, datasets and models: [29].

TABLE I
Results on UVD Dataset

	UVD_FCN	FVD_FCN	MIX_FCN	MIX_LSTM
thr. $\delta < 1.25$	**0.705**	0.211	0.462	0.590
thr. $\delta < 1.25^2$	**0.899**	0.365	0.778	0.872
thr. $\delta < 1.25^3$	**0.968**	0.493	0.938	0.950
RMSE	**4.527**	15.697	6.581	5.966
Log RMSE	**0.264**	1.076	0.356	0.327
Scale Inv. MSE	**0.065**	0.907	0.072	0.087
Abs.Rel.Diff.	0.211	0.825	0.269	**0.188**

For threshold errors, higher values are better. For RMSE, Log RMSE, Scale Inv. MSE and Abs.Rel.Diff., lower values are better

TABLE II
Results on FVD Dataset

	UVD_FCN	FVD_FCN	MIX_FCN	MIX_LSTM
thr. $\delta < 1.25$	0.326	**0.574**	0.469	0.511
thr. $\delta < 1.25^2$	0.571	**0.853**	0.777	0.766
thr. $\delta < 1.25^3$	0.733	**0.939**	0.911	0.897
RMSE	8.802	**4.132**	5.134	5.460
Log RMSE	0.656	**0.340**	0.402	0.413
Scale Inv. MSE	0.357	**0.091**	0.106	0.132
Abs.Rel.Diff.	0.564	**0.248**	0.300	0.316

A. KITTI Dataset

We evaluate our networks on a test set of 697 images used for evaluation in existing depth estimation methods [7], [8]. We do not perform any kind of fine-tuning or retraining on the target dataset. As reference, we compare with the method proposed by Eigen *et al.* [7]. The publicly available depth predictions they provide were specifically trained on the KITTI dataset, so comparison is not fully fair; our objective is to evaluate how close our performance can get relying solely on synthetic data.

We resize the input images from their original resolution of 1224 × 386 pixel to a resolution of 256 × 78 pixels for computational efficiency and feed them into our networks. From the provided sparse ground truth, captured by Velodyne lidar with a maximum range of about 80 meters, we generate a dense depth map utilizing the colorization routine proposed in [30]. As the lidar cannot provide depth information for the upper section of the image space, we perform evaluation only on the bottom section of the image space. We finally compute the performance metrics with respect of the windowed dense ground truth. We discard all the predictions whose corresponding ground truth

TABLE III
RESULTS ON KITTI DATASET

	OPTFLOW_FCN	UVD_FCN	FVD_FCN	MIX_FCN	MIX_LSTM	MIX_EIGEN	KITTI_EIGEN [7]	
thr. $\delta < 1.25$	0.421	0.414	0.160	**0.512**	0.338	0.183	0.498	Higher
thr. $\delta < 1.25^2$	0.679	0.695	0.351	0.786	0.644	0.456	**0.850**	is
thr. $\delta < 1.25^3$	0.813	0.849	0.531	0.911	0.848	0.665	**0.957**	better
RMSE	6.863	8.108	9.519	**5.654**	6.662	7.929	5.699	Lower
Log RMSE	0.504	0.470	0.877	0.366	0.472	0.589	**0.316**	is
Scale Inv. MSE	0.205	0.181	0.315	0.107	0.185	0.131	**0.051**	better
Abs.Rel.Diff.	–	0.393	0.494	**0.312**	0.430	0.390	0.322	

In this benchmark, our best model (MIX_FCN) outperforms the Eigen's one when the latter is trained on our same synthetic dataset (MIX_EIGEN). Furthermore, it gets results close to the ones achieved with the model specifically trained on the KITTI dataset (KITTI_EIGEN).

measurement is beyond 40 meters, to be compliant with our network's maximum detection range.

As for Eigen's method, we compare both their publicly available depth predictions from their coarse+fine model trained on the KITTI dataset (referred as KITTI_EIGEN) and the MIX_EIGEN model we trained with respect to the synthetic images on the KITTI test set with the same dense ground truth we generated, employing the same benchmark used for our networks, to ensure evaluation fairness.

On Table III and on Fig. 5 we report results for our FCN and LSTM networks, the baseline method [12] and Eigen *et al.*'s work.

The KITTI benchmark naturally favors networks trained on urban scenario datasets, as UVD_FCN. On the other hand, a forest scenario dataset as FVD does not suit well for this benchmark, as FVD_FCN performance clearly depicts. Anyway, mixing together FVD and UVD to form a heterogeneous training set allows MIX_FCN to improve significantly its prediction quality over UVD_FCN. With respect to KITTI_EIGEN, our best network obtains quite comparable performance on all metrics, recording slightly worse performance on threshold errors, Log RMSE and Scale Inv. MSE metrics but even some improvement on Linear RMSE and Absolute Relative Difference metrics. This is a very important result, especially considering how Eigen's work has been specifically trained on the target dataset. Heterogeneous synthetic training sets help the networks to learn a nicely generalizable model, without needing to resort on fine-tuning or collection of costly labeled real world datasets. Furthermore, our MIX_FCN network achieves better performance with respect to all the metrics than the MIX_EIGEN one. This suggests that our model has better generalization capabilities than the one presented in [7].

It is not surprising that the MIX_LSTM network does not achieve the best performance with respect to this dataset: the image frames of the test set are not always sequential and, thus, the LSTM model could not fully exploit its recurrent structure.

B. Zurich Forest Dataset

We gathered a new dataset in order to test the generalization of our networks on a real-world forest environment. The three sequences in the dataset consist of camera images captured while moving through a forested area at a walking pace of around 1 m/s. Each sequence lasted approximately 60 seconds and covered approximately 50 m of distance. These sequences include a variety of forest densities, tree sizes, and realistic lighting conditions.

The original images in this dataset were captured with a pair of time-synchronized MatrixVision mvBlueFOX-MLC200w monochrome cameras with 752 × 480 resolution in stereo configuration with a baseline of 20 cm. Both cameras were recorded at 50 Hz, resulting in sequences with approximately 3000 stereo pairs each. Stereo matching was performed on these image pairs using OpenCV's Semi-global Block Matching algorithm to generate ground truth depth for validation of the monocular depth produced by our networks [31].

We tested our architectures on the three sequences, for a total of 9846 images. We resize the images on a resolution of 256 × 160 pixels before feeding them into our networks. We report results for our baseline method OPTFLOW_FCN and all the networks trained on FVD and MIX dataset. We report results on Table IV and on Fig. 6.

In this experiment, the LSTM architecture outperforms in almost all metrics the FCN architecture on both training datasets. In particular, we observe significant improvements on global scale-dependent metrics like threshold errors, LogRMSE and the Absolute Relative Difference. This confirms our intuition: LSTM layers helps to improve global scale estimation by using past information to refine current estimations. This comes at a very low computational additional cost, as depicted on Table V. As for the experiments on the KITTI dataset, both the FCN and LSTM architectures perform better than the MIX_EIGEN model.

C. Perugia Countryside Dataset

To further evaluate the generalization capabilities of our approach, we collected a second dataset in the countryside area that surrounds the city of Perugia in Italy. Since the MIX_FCN and MIX_LSTM models are trained in forest and urban contexts, this new dataset has been specifically gathered to test whether our networks are able to generalize with respect to domains different from the training set ones or not. Images were collected using a stereo camera rig mounted on a car driven at around 14 m/s (see Fig. 7). The sequences cover many kilometers of distance and contain different scenarios, elements (*e.g.*, small town buildings, sparse tree landscapes, moving cars and others) and light conditions.

TABLE IV
RESULTS ON ZURICH FOREST DATASET

	OPTFLOW_FCN [12]	UVD_FCN	FVD_FCN	MIX_FCN	FVD_LSTM	MIX_LSTM	MIX_EIGEN	
thr. $\delta < 1.25$	0.096	0.115	0.106	0.149	0.126	**0.336**	0.111	Higher
thr. $\delta < 1.25^2$	0.202	0.238	0.231	0.316	0.269	**0.561**	0.246	is
thr. $\delta < 1.25^3$	0.295	0.409	0.380	0.520	0.439	**0.707**	0.436	better
RMSE	10.642	10.950	9.986	9.292	**9.126**	9.746	10.673	Lower
Log RMSE	1.133	0.924	1.007	0.856	0.908	**0.768**	0.960	is
Scale Inv. MSE	0.646	0.395	0.527	0.402	0.523	0.439	**0.357**	better
Abs.Rel.Diff.	2.127	1.604	1.797	1.378	1.427	**1.272**	1.777	

Both MIX_FCN and MIX_LSTM outperform MIX_EIGEN in most of the metrics.

TABLE V
FPS (FRAME PER SECOND) FOR FCN AND LSTM NETWORKS ON 256 × 160 PIXEL INPUTS

	FPS (K40)	FPS (TK1)
FCN nets	58.8	2.7
LSTM nets	35.3	2.4

Tested hardware: Tesla K40 and Jetson TK1 (for MAV onboard deploying).

Fig. 7. Car setup used for collecting the Perugia Countryside Dataset. On the right, some sample images of the recorded sequences are shown.

TABLE VI
RESULTS ON PERUGIA COUNTRYSIDE DATASET

	MIX_FCN	MIX_LSTM	MIX_EIGEN
thr. $\delta < 1.25$	0.204	**0.209**	0.197
thr. $\delta < 1.25^2$	0.396	**0.405**	0.389
thr. $\delta < 1.25^3$	0.567	**0.576**	0.564
RMSE	13.003	**12.766**	12.925
Log RMSE	**0.802**	0.811	0.820
Scale Inv. MSE	0.583	**0.542**	0.640
Abs.Rel.Diff.	0.678	**0.631**	0.720

The dataset was gathered with a pair of time-synchronized MatrixVision mvBlueFOX3 RGB cameras with 1280 × 960 resolution. In order to be able to compute the ground truth at higher ranges, we set up a stereo rig with a baseline of 60 cm. Both cameras recorded at 10 Hz, resulting in sequences with approximately 1600 stereo pairs each. Stereo matching was performed using the same strategy described in Section IV-B.

We compare our MIX_FCN and MIX_LSTM architectures (which showed good generalization capabilities in the previous experiments) and the baselines with respect to three sequences (5072 images). As the LSTM network and the Eigen's approach require input images with 256 × 160, we crop and resize them accordingly.

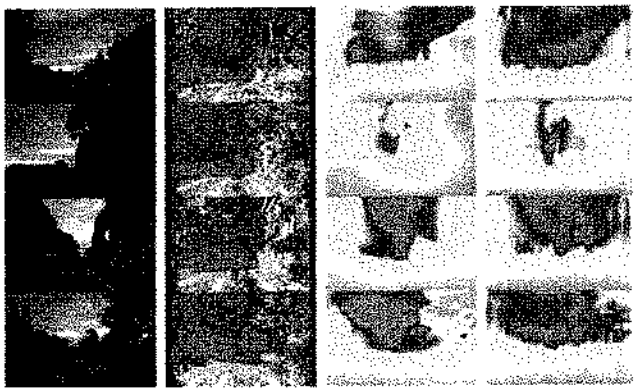


Fig. 8. Qualitative results on the Perugia Countryside dataset. On the first column RGB input images are depicted. The second and the third columns show the dense ground truths and the MIX_LSTM predictions, respectively. The fourth column shows MIX_EIGEN network prediction. Maximum depth range has been trimmed to 40 meters. Black pixels in the ground truth represent missing depth measurements.

The results (see Table VI) confirm that the recurrent structure provides better performance with respect to both the standard FCN network and the Eigen's approach. Depth estimates (shown in Fig. 8) are coherent with the actual scene depths. Thus, this suggests that our models (trained with images from different contexts, *e.g.*, dense forest and urban) are able to generalize with respect to different domains, considerably extending the applications contexts of depth estimation techniques.

We can also observe that the errors are higher with respect to the KITTI and Zurich forest dataset. However, this could be explained by the difference of camera intrinsics between the test and the train setup. Our networks are still able to provide reliable estimate when processing images with different focal lengths up to a scale factor. Despite the absolute metric errors are higher, the relative estimation are consistent (see Fig. 8).

V. CONCLUSION AND FUTURE WORK

We propose a novel, Deep Learning based monocular depth estimation method, aimed at micro aerial vehicles tasks, such as autonomous obstacle avoidance and motion planning. We demonstrate how, using solely synthetic datasets, we can train

a generalizable model that is capable of robust performance in real world scenarios. We obtained results that are comparable with the state of the art on the KITTI dataset without any fine-tuning. We also tested our algorithm in two other challenging scenario we gathered in a dense forest and a countryside, additionally showing how LSTM layers effectively help to improve estimation quality on typical MAV operating scenarios with a low added computational overhead. Future works will explore the possibility of integrating information coming from different sensors and/or modules (e.g. IMU, semantic segmentation) to gain a better understanding of the surroundings and implement an effective reactive control for obstacle avoidance over it.

Acknowledgment

The authors would like to acknowledge the support of NVIDIA Corporation with the donation of the Tesla K40 GPU used for this research.

References

[1] H. Oleynikova, D. Honegger, and M. Pollefeys, "Reactive avoidance using embedded stereo vision for MAV flight," in *Proc. 2015 IEEE Int. Conf. Robot. Autom.*, 2015, pp. 50–56.

[2] C. Richter, A. Bry, and N. Roy, "Polynomial trajectory planning for aggressive quadrotor flight in dense indoor environments," in *Robotics Research*. New York, NY, USA: Springer, 2016, pp. 649–666.

[3] P. Pinggera, D. Pfeiffer, U. Franke, and R. Mester, "Know your limits: Accuracy of long range stereoscopic object measurements in practice," in *Computer Vision*. New York, NY, USA: Springer, 2014, pp. 96–111.

[4] E. R. Davies, *Machine Vision: Theory, Algorithms, Practicalities*. Amsterdam, The Netherlands: Elsevier, 2004.

[5] P. Pinggera, U. Franke, and R. Mester, "High-performance long range obstacle detection using stereo vision," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2015, pp. 1308–1313.

[6] A. Harakeh, D. Asmar, and E. Shammas, "Ground segmentation and occupancy grid generation using probability fields," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2015, pp. 695–702.

[7] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2366–2374.

[8] F. Liu, C. Shen, G. Lin, and I. Reid, "Learning depth from single monocular images using deep convolutional neural fields," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 10, pp. 2024–2039, Oct. 2016.

[9] A. Roy and S. Todorovic, "Monocular depth estimation using neural regression forest," 2016.

[10] A. Saxena, M. Sun, and A. Y. Ng, "Make3D: Learning 3D scene structure from a single still image," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 5, pp. 824–840, May 2009.

[11] R. Garg and I. Reid, "Unsupervised CNN for single view depth estimation: Geometry to the rescue," arXiv preprint arXiv:1603.04992, 2016.

[12] M. Mancini, G. Costante, P. Valigi, and T. A. Ciarfuglia, "Fast robust monocular depth estimation for obstacle detection with fully convolutional networks," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2016, pp. 4296–4303.

[13] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *Int. J. Comput. Vis.*, vol. 47, no. 1–3, pp. 7–42, 2002.

[14] M. Pizzoli, C. Forster, and D. Scaramuzza, "Remode: Probabilistic, monocular dense reconstruction in real time," in *Proc. 2014 IEEE Int. Conf. Robot. Autom.*, 2014, pp. 2609–2616.

[15] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-scale direct monocular slam," in *European Conference on Computer Vision*. New York, NY, USA: Springer, 2014, pp. 834–849.

[16] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, "Dtam: Dense tracking and mapping in real-time," in *Proc. 2011 Int. Conf. Comput. Vis.*, 2011, pp. 2320–2327.

[17] A. Saxena, S. H. Chung, and A. Y. Ng, "Learning depth from single monocular images," in *Proc. Adv. Neural Inf. Process. Syst.*, 2005, pp. 1161–1168.

[18] D. Eigen and R. Fergus, "Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 2650–2658.

[19] P. Fischer *et al.*, "Flownet: Learning optical flow with convolutional networks," arXiv preprint arXiv:1504.06852, 2015.

[20] V. Badrinarayanan, A. Handa, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling," *arXiv preprint arXiv:1505.07293*, 2015.

[21] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[22] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2009, pp. 248–255.

[23] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the devil in the details: Delving deep into convolutional nets," *arXiv preprint arXiv:1405.3531*, 2014.

[24] T. N. Sainath, O. Vinyals, A. Senior, and H. Sak, "Convolutional, long short-term memory, fully connected deep neural networks," in *Proc. 2015 IEEE Int. Conf. Acoust., Speech Signal Process.*, 2015, pp. 4580–4584.

[25] S. Sharma, R. Kiros, and R. Salakhutdinov, "Action recognition using visual attention," *arXiv preprint arXiv:1511.04119*, 2015.

[26] N. McLaughlin, J. Martinez del Rincon, and P. Miller, "Recurrent convolutional network for video-based person re-identification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2016, pp. 1325–1334.

[27] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[28] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *Int. J. Robot. Res.*, vol. 32, pp. 1231–1237, 2013.

[29] [Online]. Available: http://sira.diei.unipg.it/supplementary/ral2016/extra.html

[30] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from RGBD images," in *Computer Vision*. New York, NY, USA: Springer, 2012, pp. 746–760.

[31] H. Hirschmuller, "Stereo processing by semiglobal matching and mutual information," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 2, pp. 328–341, Feb. 2008.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LRA.2018.2793345, IEEE Robotics and Automation Letters

Full-GRU Natural Language Video Description for Service Robotics Applications

Silvia Cascianelli[1], Gabriele Costante[1], Thomas A. Ciarfuglia[1], Paolo Valigi[1] and Mario L. Fravolini[1]

***Abstract*—Enabling effective Human-Robot Interaction (HRI) is crucial for any service robotics application. In this context, a fundamental aspect is the development of a user-friendly human-robot interface, such as a natural language interface. In this work, we investigate the robot side of the interface, in particular the ability to generate natural language descriptions for the scene it observes. We achieve this capability via a Deep Recurrent Neural Network (D-RNN) architecture completely based on the Gated Recurrent Unit (GRU) paradigm. The robot is able to generate complete sentences describing the scene, dealing with the hierarchical nature of the temporal information contained in image sequences. The proposed approach has fewer parameters than previous State-of-the-Art architectures, thus it is faster to train and smaller in memory occupancy. These benefits do not affect the prediction performance. In fact, we show that our method outperforms or is comparable to previous approaches in terms of quantitative metrics and qualitative evaluation when tested on benchmark publicly available datasets and on a new dataset we introduce in this paper.**

***Index Terms*—Cognitive Human-Robot Interaction; Visual Learning**

I. Introduction

THE ability to provide a description of the scene in a form that every user can easily understand is keystone for the success of effective and user-friendly service robotics products. In fact, a natural language description offers an interpretable manifestation of the robot's inner representation of the scene and is also a good basis for natural language question answering about what is happening in the environment. Hence, this functionality would provide a friendly interface also for non-expert people who would then be able to easily interact with their home robot in the near future.

In the sight of this, this work addresses the problem of describing a scene in natural language, which is usually referred to as Natural Language Video Description (NLVD). Here we formalize this problem as a Machine Translation (MT) one, from "visual language" to English. Basically, the information in form of a varying length video sequence is encoded in a fixed-length vector and then decoded in form of varying length English sentence (Fig. 1).

Manuscript received: September, 10, 2017; Revised December, 09, 2017; Accepted December, 29, 2017.

This paper was recommended for publication by Editor Dongheui Lee upon evaluation of the Associate Editor and Reviewers' comments. *We gratefully acknowledge the support of NVIDIA Corporation with the donation of the *Titan Xp* GPU used for this research.

[1]All the Authors are with Department of Engineering, University of Perugia, Perugia (Italy) silvia.cascianelli@studenti.unipg.it, {gabriele.costante, thomas.ciarfuglia, paolo.valigi, mario.fravolini}@unipg.it

Digital Object Identifier (DOI): see top of this page.

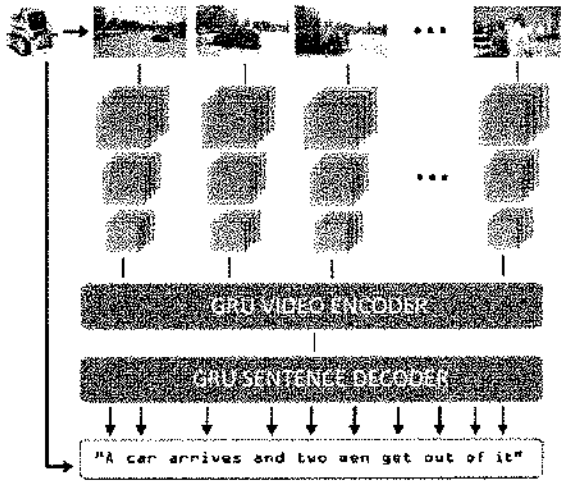


Fig. 1: Overview of the proposed NLVD system. The robot observes a generic and complex scene and represents it taking into account both the visual and temporal information, represented via ConvNet features and an encoding vector, respectively. Then, it outputs a natural language sentence describing the observed scene. The proposed encoder-decoder scheme is entirely based on GRU recurrent units.

The video translation is performed via D-RNNs, *i.e.*, recurrent models that are able to deal with both long and short term dependencies in data sequences. Most of the previous approaches rely on the Long Short-Term Memory (LSTM) [1] architecture. Recently, Generalized Recurrent Unit (GRU) [2] were proposed as a simplification of LSTM units. Their performance is similar to LSTMs', but with fewer parameters to train. This makes the recurrent networks based on GRU faster to train and less prone to overfitting. Saving training time in any deep learning application is critical when tuning hyper parameters for field application, such as robotics. For this reason, in this paper we explore the performances of a NLVD system completely based on the GRU paradigm, comparing it to State-of-the-Art approaches that exploit LSTMs.

In this work, a full-GRU NLVD system is proposed, that is able to deal with the hierarchical nature of the temporal information typical of natural and generic video sequences and obtains comparable performance with respect to more complex State-of-the-Art (SotA) systems. The proposed system features a GRU cell modified in order to automatically change its tem-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LRA.2018.2793345, IEEE Robotics and Automation Letters

poral connection if a boundary, *i.e.*, a significant modification in the scene, is detected. To the best of our knowledge, this is the first full-GRU encoder-decoder architecture applied to the problem of NLVD. In addition, a new small dataset for NLVD in typical service robotics scenarios is used, which offers a fair test bench for the specific application we target. The relevance of this dataset, is twofold. First, this is the first dataset specifically collected in typical applicative contexts of a service robot. Second, it helps to get insights on the actual performance of SotA NLVD models we are testing. Indeed, these systems are commonly trained and tested on videos from the same datasets, which may make their evaluation biased. The experiments on our dataset makes this more evident.

To summarize, the main contributions of this work are:

- We propose an improved architecture for NLVD that is based on GRU units, to save training time without impairing the performances.
- We perform an experimental evaluation of our method with other SotA approaches. The experiments show that this method obtains comparable performances with SotA methods that harness LSTM.
- We present a dataset that features a wide range of contexts that are typical for service robotics applications.

The remainder of this paper is organized as follows. In Section III the proposed approach is described. Section IV provides a detailed description of the experimental results and conclusion are drawn in Section V.

II. Related Work

In recent years, many researchers from both Computer Vision and Natural Language Processing communities are studying the problem of describing generic videos using natural language phrases (see *e.g.*, [3], [4]).

Some popular approaches [4], [5] are based on filling-in predefined template sentences with the subject-verb-object concepts detected in the video. In particular, an object detector (*e.g.*, a CNN as in [5]) is used to recognize the main actors in the video and a Probabilistic Graphical Model (PGM) (*e.g.*, an Hidden Markov Model as in [4]) is used to predict the relation between them. These approaches have major limitations. First, the type and the number of the objects and the relations that can be described are limited to those that the detector and the PGM can estimate. Second, the output descriptions lack in diversity and naturalness.

Other works [6] propose to tackle the NLVD task in a multi-modal retrieval fashion. In particular, given a corpus of paired videos and text, the system describes a new video using the sentence associated to the most similar video in the corpus [6]. Also this approach has some weaknesses. In particular, the system is constrained to use the same sentences in the corpus, which may be not semantically relevant for the new scene to describe.

Among the proposed strategies, treating the NLVD problem as a Machine Translation (MT) one gained popularity [7] and D-RNN demonstrated to be a very promising instrument [8], [9], [10]. This is particularly true when recurrent models are combined with State-of-the-Art Convolutional Neural Networks (ConvNet), even pre-trained.

Despite of the success of recent State-of-the-Art approaches, NLVD is still a particularly challenging problem, firstly due to the "object" of the description itself, *i.e.*, the video sequence, that is typically open-domain and complex in real scenarios. In particular, the content of the videos can be highly diverse and the temporal dependencies between the depicted events can be at different granularity. Some architectures exist that produce accurate descriptions of videos, but in general these are either very short or very specific, or both, *i.e.*, they depict simple activities of a particular domain with few "actors" in the scene [9], [5]. Those kinds of video sequences are far simpler than the typical complexity that a robot faces in real application contexts. The systems presented in [8] and [10] deal with generic and complex videos. Both of them represent the video sequence by mean-pooling the ConvNet features extracted from each frame, then decode the sentence with a LSTM-based decoder. A major drawback of those strategies is that they do not take into account the temporal structure of the video sequences due to mean-pooling.

Indeed, when considering more complex and generic video sequences it is crucial to deal with temporal dependencies at different granularity. This is done in [11], [12] and also in this work, where a hierarchical representation of the temporal information is explicitly learned. In [11] the authors draw from ConvNets the idea of convolutional operations and build a multi-level LSTM-based encoding able to capture longer time dependencies between the content of the frames. Then, a LSTM decoder produces the description exploiting an attention mechanism (that is basically a learned weighting strategy). The work of [12] is the most similar to our work. It presents a LSTM-based decoder that contains a boundary-aware LSTM cell. This cell and a second layer LSTM block build an encoding of the video sequence which is then decoded via a GRU.

All of the above approaches, either consists of full-stack LSTM architectures or limit the use of the GRU to the decoding phase. In this paper, we present an encoder-decoder architecture that is completely based on GRU blocks, which have fewer parameters than LSTM, thus resulting arguably more suitable for robotics applications. This is motivated also by the study reported in [13], that compares the GRU and the LSTM cells on various tasks. Using input, state and output vectors of the same dimensionality, the GRU outperforms or is comparable to the LSTM in terms of convergence time, parameters update and generalization.

III. Encoder-Decoder full-GRU Architecture

In this section our proposed model is presented. The video frames are described via the *ResNet50* and the *C3D* ConvNets (see III-A). The obtained feature vectors are then fed, one at each time-step, in the first layer of the encoder. This is our proposed BA-GRU recurrent block, that encodes the video frames until a boundary is detected. Afterwards, the first-layer encoding is fed to the second layer of the encoder, which consists of a classical GRU block (see III-B). The output of the encoding phase is a vector representing the entire video sequence. Finally, the GRU decoder produces the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LRA.2018.2793345, IEEE Robotics and Automation Letters

description emitting the most probable word at each time-step, conditioned to the video vector representation and the previous emitted words (see III-C). The captioning process ends when a <EOS> tag (*i.e.*, the full-stop) is emitted. A pictorial representation of the system is shown in Fig. 2.

A. Video Frames and Caption Words Preprocessing

The video frames are preprocessed as follows. The output of the last fully connected layer of the *ResNet50* ConvNet [14] is computed every five video frames, to capture the appearance of the scene. To the same video frames is associated also the output of the *C3D* ConvNet [15] to capture the movement in the scene, based on partially overlapped sliding windows of frames. The output of the two ConvNets are concatenated (forming a 2048+4096-dimensional vector) and mapped in a learned 512-dimensional linear embedding. The entire video is then represented by a sequence of features vectors $(\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n)$, where the x_i vectors are the feature vectors extracted from the frames of the video.

The captions are preprocessed as follows. First, the words are converted to lower-case and the punctuation characters are removed. Then, begin-of-sentence (<BOS>) and end-of-sentence (<EOS>) tags are added before and behind the sentence, respectively. Finally, the sentences are tokenized. From the tokenized sentences, we build a vocabulary (D). To prevent the formation of a large vocabulary containing many rare words, we retain only those tokens that appear at least five times in the caption corpus. To each token is associated an index in the vocabulary, based on its frequency in the vocabulary. A caption is then represented by a list of one-hot vectors $(\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_L)$, each of them corresponding to the representation of its words in the vocabulary. Similarly to what is done for the frames features, the captions are mapped in a learned 512-dimensional linear embedding.

B. Video Encoder

In this work, we build upon the boundary-aware LSTM (BA-LSTM) cell presented in [12] and devise a boundary-aware GRU (BA-GRU) cell. This cell is the first layer of a two-layers encoder. The second layer of the encoder is a simple GRU cell [2].

The BA-GRU is a modification of the classical GRU cell (see Fig. 2, top right). The GRU is a recurrent neural networks with gating strategies to model wider temporal dependencies in the input sequence. The GRU is characterized by an update gate $\mathbf{z}_t$ and a reset gate $\mathbf{r}_t$. At each timestep, a candidate activation $\tilde{\mathbf{h}}_t$ is computed based on the current input $\mathbf{x}_t$, the previous inner state $\mathbf{h}_{t-1}$ and the values of the gates. In particular, the $\mathbf{z}_t$ gate controls how much the inner state $\mathbf{h}_t$ has to be updated, the $\mathbf{r}_t$ gate controls how much the previous inner state $\mathbf{h}_{t-1}$ influences the candidate inner state value $\tilde{\mathbf{h}}_t$. More formally, the GRU is defined by the following equations:

$$\mathbf{h}_t = (1 - \mathbf{z}_t)\mathbf{h}_{t-1} + \mathbf{z}_t\tilde{\mathbf{h}}_t \tag{1}$$

$$\tilde{\mathbf{h}}_t = tanh(W_{hx}\mathbf{x}_t + W_{hh}(\mathbf{r}_t \odot \mathbf{h}_{t-1}) + \mathbf{b}_h) \tag{2}$$

$$\mathbf{r}_t = \sigma(W_{rx}\mathbf{x}_t + W_{rh}\mathbf{h}_{t-1} + \mathbf{b}_r) \tag{3}$$

$$\mathbf{z}_t = \sigma(W_{zx}\mathbf{x}_t + W_{zh}\mathbf{h}_{t-1} + \mathbf{b}_z) \tag{4}$$

where the W_{**}s and $\mathbf{b}_*$s are learnable weight matrices and bias vectors, σ is the sigmoid function, $tanh$ is the hyperbolic tangent function and $\odot$ is the element-wise product.

In this work, we modify the GRU by adding a boundary aware gate s_t, that modifies the inner connectivity of the unit based on the input and the inner state. In particular, when a substantial change in input sequence occurs, a boundary is estimated by a learnable function. Consequently, the inner state $\mathbf{h}_{t-1}$ is emitted as output (we denote it as $\mathbf{h}_k^{e1} = \mathbf{h}_{t-1}$) and then re-initialized to zero according to:

$$\mathbf{h}_{t-1} \leftarrow \mathbf{h}_{t-1}(1 - s_t) \tag{5}$$

The boundary-aware gate is defined as follows:

$$s_t = \tau(\mathbf{w}_s^T(W_{sx}\mathbf{x}_t + W_{sh}\mathbf{h}_{t-1} + \mathbf{b}_s)) \tag{6}$$

where W_{**}s and $\mathbf{b}_s$ are learnable weights matrices and bias vectors. In this study, we set to 128 the number of their rows. The row vector $\mathbf{w}_s^T$ makes the input to the $\tau(\cdot)$ function a scalar. The $\tau(\cdot)$ function is given by:

$$\tau(\cdot) = \begin{cases} 1 & \text{if } \sigma(\cdot) < 0.5 \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

The given output $\mathbf{h}_k^{e1}$ summarizes the video substream before the boundary, which is then composed by homogeneous frames. For an input video, the BA-GRU block outputs as many vectors $\mathbf{h}_k^{e1}$, as the number of detected boundaries $(\mathbf{h}_1^{e1}, \mathbf{h}_2^{e1}, ..., \mathbf{h}_m^{e1})$, with $m \leq n$. Those vectors are given in input to the second layer of the encoder, which is a standard GRU block. This layer encodes the $\mathbf{h}_k^{e1}$ vectors in a unique vector $\mathbf{v}$, that represents the entire video. The $\mathbf{v}$ vector, that is the final output of the two-layer encoder, is fed to the decoder.

1) The Boundary-Aware Gate Training Details: The output s_t of the boundary-aware gate can be either 0 or 1, depending on the value of a sigmoid function applied to the input of the gate. Thus, following the approach of [12], in the training phase we model it as a stochastic binary neuron and learned its weights, while in test phase we use it with the learned weights as the deterministic neuron defined in Eq.7. In particular, we re-write the activation function $\tau(\cdot)$ as:

$$\tau(\cdot) = \mathbf{1}_{\sigma(\cdot) > z}, \qquad z \sim \mathcal{U}(0, 1) \tag{8}$$

where $\mathbf{1}_{\cdot}$ is the indicator function and $\mathcal{U}(0, 1)$ denotes the uniform distribution between 0 and 1.

Note that $\tau(\cdot)$ in Eq.7 is basically the composition of a step function and a sigmoid function. Thus, its derivative is equal to 0 everywhere except in 0, *i.e.*, it is not continuous and smooth and it is also mostly flat. Hence, we cannot apply the standard back-propagation to compute the gradient in this gate. To overcome this issue, we follow the same approach of [12], that estimated the gradient by approximating the step function $\tau(\cdot)$ as the identity function [16]. The derivative of $\tau(\cdot)$ then becomes:

$$\frac{\partial \tau}{\partial(\cdot)}(\cdot) = \frac{\partial \sigma}{\partial(\cdot)}(\cdot) = \sigma(\cdot)(1 - \sigma(\cdot)) \tag{9}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LRA.2018.2793345, IEEE Robotics and Automation Letters

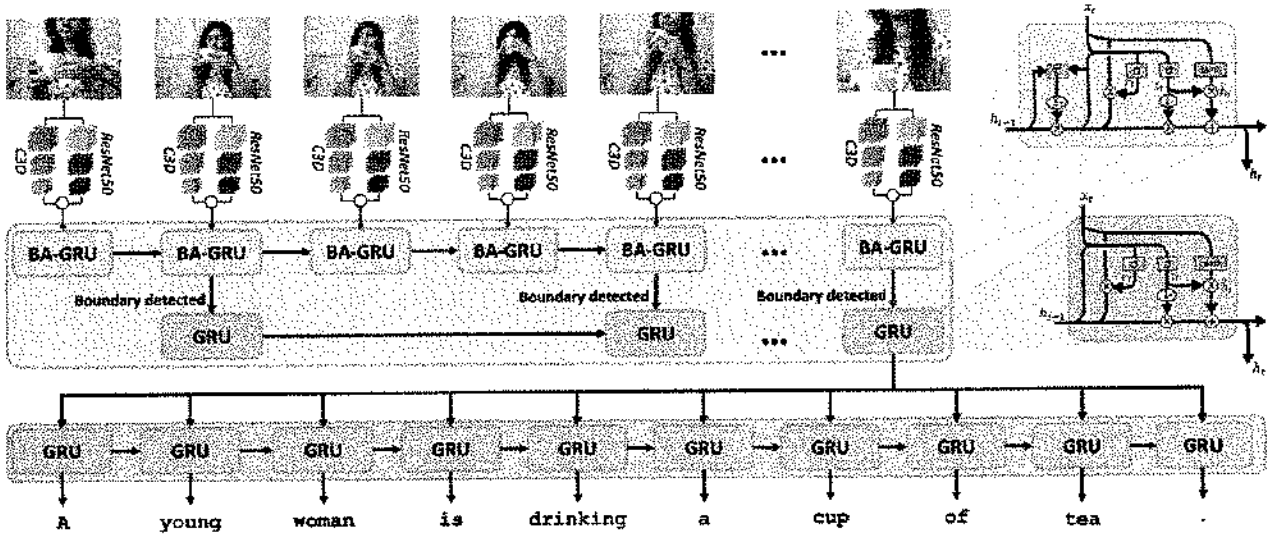


Fig. 2: Architecture of the proposed system. Recurrent layers are depicted as unfolded graphs for explanatory purpose.

In the test phase, we use the deterministic form of $\tau(\cdot)$ (Eq.7), the parameters of which have been learned in the training phase using Eq.8 (in the forward pass) and Eq.9 (in the backward pass).

C. Caption Decoder

The decoder takes as input the video representation $\mathbf{v}$ and the ground truth sentence $(\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_L)$. At each timestep, it outputs a word $\mathbf{y}_t$ that is the most probable next word of the description, given the previous output words and the video representation.

To handle both the time-varying input $(\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_L)$ and the constant input $\mathbf{v}$, we modify Eq.2-4 from the original GRU formulation as:

$$\tilde{\mathbf{h}}_t = tanh(W_{hy}W_w\mathbf{y}_t + W_{hv}\mathbf{v} + W_{hh}(\mathbf{r}_t \odot \mathbf{h}_{t-1}) + \mathbf{b}_h) \quad (10)$$

$$\mathbf{r}_t = \sigma(W_{ry}W_w\mathbf{y}_t + W_{rv}\mathbf{v} + W_{rh}\mathbf{h}_{t-1} + \mathbf{b}_r) \quad (11)$$

$$\mathbf{z}_t = \sigma(W_{zy}W_w\mathbf{y}_t + W_{zv}\mathbf{v} + W_{zh}\mathbf{h}_{t-1} + \mathbf{b}_z) \quad (12)$$

where the W_{**}s and $\mathbf{b}_*$s are learnable weight matrices and bias vectors respectively, σ is the sigmoid function and $\odot$ is the element-wise product. The matrix W_w maps the input one-hot vectors representing the words $\mathbf{y}_t$ in the vocabulary space in a lower dimensional space (512-dimensional embedding). The output of the decoder (which we denote $\mathbf{h}_t^d \doteq \mathbf{h}_t$) is then mapped back in the original higher dimensional space as $\mathbf{y}_t = W_p\mathbf{h}_t^d$.

The probability of the next word in the description is modelled via the softmax function, *i.e.*,

$$Pr(\mathbf{y}_t|\mathbf{y}_0, \mathbf{y}_1, ..., \mathbf{y}_{t-1}, \mathbf{v}) \sim \frac{e^{\mathbf{y}_t^T W_p \mathbf{h}_t^d}}{\sum_{\mathbf{y} \in D} e^{\mathbf{y}^T W_p \mathbf{h}_t^d}} \quad (13)$$

Finally, the objective function to optimize is the log-likelihood of the correct words over the sentence *i.e.*,

$$\max_{\mathbf{W}} \sum_{l=1}^{L} logPr(\mathbf{y}_l|\mathbf{y}_0, \mathbf{y}_1, ..., \mathbf{y}_{l-1}, \mathbf{v}) \quad (14)$$

where $\mathbf{W}$ denotes all the parameters of the model.

IV. Experiments and Results

In this section, we present the experimental setup and the obtained results of our method.

A. Datasets Details

We employ two publicly available large datasets that are commonly used to study the NLVD problem. In addition, we test on a smaller dataset that we collected to be representative of daily activities that are typical of service robotics scenarios.

a) Max Plank Institute for Informatics Movie Description Dataset (MPII-MD): This dataset [17] contains over 68 000 clips of average 4*s* each, from a corpus of 94 HD movie of different genres. Those clips are associated with sentences taken from the movie script and the transcribed Descriptive Video Service (DVS[1]) track. As a common practice, we use the training/validation/test split provided by the authors of the dataset, resulting in 56 816 training clips, 4930 validation clips and 6584 test clips. This split is the same typically used for NLVD systems [3], [7], [11], [12], [18], [19]. The vocabulary is obtained from the training corpus and consists of 7198 words.

b) The Microsoft Research Video Description Corpus (MSVD): This dataset [20] contains home-made 10-20*s* long videos from YouTube. The topics of the videos include sports, animals and music. We retain the 1970 clips that have English captions associated. The captions are on average 43 for each video and have been collected by the Amazon Mechanical Turk service. As the common practice [7], [8], [10], [11], [12], [19], we use the first 1200 videos for training, the next 100 video for validation and the last 670 video for testing. Note that each video-caption pair is considered as a unique sample, so

[1]Descriptive Video Service is an audio track associated to a movie to allow the visually impaired people to enjoy also the visual content of the movie.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LRA.2018.2793345, IEEE Robotics and Automation Letters

the actual number of samples in each split is average 43 times the number of videos. Again, we construct the vocabulary from the training set and obtain a vocabulary of 4215 words.

c) Intelligent Systems, Automation and Robotics Laboratory Video Description Dataset (ISARLab-VD): For this work, we collect a relatively small dataset. Despite that, our dataset is still generic in terms of depicted actions, environment and involved actors. Note that, none of the above datasets have been conceived for service robotics applications. This was a major motivation for us to produce the dataset. It contains 100 videos which length varies from 5*s* to 30*s*. Each video is paired with 5 manually obtained independent captions, for a total of 500 samples. The dataset features both high resolution and low resolution videos. In particular, the latter are obtained using the built-in camera of the COZMO toy robot by Anki[2] during the experimental phase of this study. In this work, we use the entire ISARLab-VD dataset for test only.

B. Evaluation Metrics Overview

In this work, we adopt classical natural language processing metrics for the evaluation of our method, which is a common practice in the NLVD research. These metrics are briefly described here for clarity and we refer to [21], [22], [23], [24] for further details. First note that a n-gram is a sequence of n consecutive words. When comparing a candidate sequence X and a reference sequence Y, the n-gram recall is the proportion of n-grams in Y that appear also in X, while the n-gram precision is the proportion of n-grams in X that appear also in Y.

The first metric we use is BLEU [21], in its *4*-gram variant. It is a precision-oriented metric designed for MT evaluation. Basically, it combines the n-gram precision for each n-gram up to length 4 and penalizes the difference in length between the candidate and the reference sentences. BLEU correlates well with human judgement on the quality of the translation if evaluated on the entire test corpus, but its correlation at sentence level is poor.

We also adopt another MT evaluation metric, namely METEOR [22]. It combines unigram precision and recall based on matching unigrams in the candidate and reference sentences. Unigrams can be matched in their exact form, stemmed form, and meaning. METEOR correlates well with human judgement also at sentence level.

The third metric we use is ROUGE [23] in its variant ROUGE$_L$, that considers the Longest Common Subsequence (LCS) of the candidate and the reference sentence. ROUGE is a recall-oriented metric designed for summarization evaluation following the idea that a good candidate summary overlaps a reference summary. Note that all ROUGE variants correlate well with human judgement.

Finally, we adopt a recently developed metric for assessing image description quality capturing human consensus on it, namely CIDEr [24]. It is based on the average cosine similarity between n-grams of different order (up to *4*-grams) and rewards length similarity between candidate and reference sentences. Cosine similarity allows taking into account both precision and recall. This metric correlates well with human judgement by design, thus is particularly suitable for the task of NLVD.

C. Baseline Methods Overview

We quantitatively compare our system to some of the State-of-the-Art techniques presented in Section II, namely SA-GoogleNet+3D-ConvNet [19], S2VT [7], LSTM-YT [8], LSTM-E [10], HRNE [11] and BA-LSTM [12]. In addition, we compare to Venugopalan et al. [18] and to Rohrbach et al. [3]. SA-GoogleNet+3D-CNN applies an attention mechanism to select the most relevant video frames based on GoogLeNet [25] and 3D-CNN [26] extracted features, and an LSTM to generate the description sentence. S2VT uses a stacked LSTM encoder-decoder on the basis of ConvNet features extracted from each frame via VGG-16 [27]. LSTM-YT mean-pools each frame's AlexNet [28] ConvNet features and decodes this representation via a LSTM. LSTM-E learns an embedding based on the frame-level extracted mean-pooled VGG-19 [27] and C3D [15] ConvNet features and the video description, then generates a sentence via a LSTM. HRNE represents each video frame via GoogLeNet features and applies a hierarchical multi-layer LSTM encoder and a LSTM with soft-attention decoder. BA-LSTM is the most similar to our approach, but it uses LSTM blocks in the encoding phase. Venugopalan et al. [18] improves S2VT using a neural language model and distributional semantics learned from a large text corpus. Rohrbach et al. [3] uses CRFs to obtain tuples of verbs, objects and places on the basis of ConvNet features extracted from the video via pre-trained ConvNets, then translated the tuple into a sentence via a LSTM.

Differently from SA-GoogleNet+3D-CNN and HRNE, we do not apply any attention mechanism to deal with different-granularity time dependencies in the videos. As opposed to LSTM-YT and LSTM-E, we explicitly model the temporal dimension of the video sequence via the recurrent encoder. Finally, another major difference between our approach and the baselines is that we use a full-GRU architecture.

Note that, since BA-LSTM is the closest to our method, we used the same settings as the authors of [12] to better compare the two architectures. In particular, we set to 1024 the size of the inner state vectors and use the same size for input vectors, embeddings, weight matrices and bias vectors. Embedding matrices and weight matrices applied to inputs are initialized via the Glorot normal initializer, those applied to inner states are initialized via the orthogonal initializer and the bias vectors are initialized to zero. We perform the training until the validation loss stops improving (or up to 100 epochs), with mini-batch size of 128. As optimizer, we apply Adadelta with learning rate $l_r = 1.0$, decay constant $\rho = 0.95$ and parameter $\epsilon = 10^{-8}$. The input and the output of the BA-GRU and the GRU in the encoding phase are regularized via Dropout with retain probability $p = 0.5$.

D. Results on the Standard Datasets

The performance is evaluated on the MPII-MD and MSVD datasets and expressed in terms of the widely used metrics

[2]https://www.anki.com/en-us/cozmo

Model	B_4	M	R_L	C
SA-GoogleNet+3D-CNN [19]	-	5.7	-	-
S2VT-RGB [7]	0.5	6.3	15.3	9.0
Venugopalan et al. [18]	-	6.8	-	-
Rohrbach et al. [3]	**0.8**	**7.0**	16.0	10.0
BA-LSTM [12]	**0.8**	**7.0**	**16.7**	10.8
BA-GRU (ours)	**0.8**	6.8	16.5	**11.7**

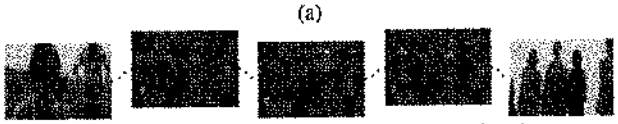
TABLE I: Experiment results on the MPII-MD dataset in terms of the quantitative evaluation metrics BLEU in its *4*-gram variant (B_4), METEOR (M), ROUGE in its LCS variant (R_L) and CIDEr (C). Bold indicates the best performance.

Model	B_4	M	R_L	C
SA-GoogleNet+3D-CNN [19]	41.9	29.6	-	-
LSTM-YT [8]	33.3	29.1	-	-
S2VT [7]	-	29.8	-	-
LSTM-E [10]	45.3	31.0	-	-
HRNE [11]	**46.7**	**33.9**	-	-
BA-LSTM [12]	*41.5*	*31.3*	68.6	*55.5*
BA-GRU (ours)	42.5	32.0	**68.8**	**59.0**

TABLE II: Experiment results on the MSVD dataset in terms of the quantitative evaluation metrics BLEU in its *4*-gram variant (B_4), METEOR (M), ROUGE in its LCS variant (R_L) and CIDEr (C). Bold indicates the best performance. Values in italic are obtained by re-running the code released by the authors of [12], which differ from those declared in their paper.

GT: A man is lifting a truck.
BA-LSTM: A man is riding a car.
BA-GRU: A man is lifting a car.

(a)



GT: News crew helicopters hover in the air above the scene.
BA-LSTM: The crowd is in the river.
BA-GRU: Someone looks at the crowd and turns to the ground.

(b)

Fig. 3: Example results on a video from the MSVD test subset 3a and on a video from a movie in the MPII-MD test subset 3b.

Model	B_4	M	R_L	C
BA-LSTM	41.5±1.0	31.4±0.3	68.5±0.5	56.1±2.0
BA-GRU	41.1±1.1	31.2±0.7	68.3±0.5	53.5±3.8

TABLE III: Experiment results of the *K*-fold cross-validation on the MSVD dataset in terms of the quantitative evaluation metrics BLEU in its *4*-gram variant (B_4), METEOR (M), ROUGE in its LCS variant (R_L) and CIDEr (C). The results are expressed in terms of mean and standard deviation.

presented in IV-B. For consistency sake with the baselines, we use the original COCO evaluation script[3].

The results are summarized in Tab. I for the MPII-MD dataset and in Tab. II for the MSVD dataset. It can be observed that our method is competitive with all the other approaches in terms of all the metrics. More importantly, it outperforms all the baselines in terms of the CIDEr metric, that has been reported in [24] best capturing human consensus on captions.

The lower performances of the MPII-MD dataset compared to MSVD are due to the fact that in the former the ground truth is taken from the DVS subtitle system, so it is not a real description of the scenes. Furthermore, the ground truth captions in the MSVD dataset are more precise and higher in number when compared to those of the MPII-MD dataset (∼40 versus 1-2). Some examples are given in Fig. 3.

In addition, to gain some insights on the statistical significance of the presented quantitative results, we perform a *K*-fold cross-validation (with K= 10) of our approach and the BA-LSTM baseline on the MSVD dataset. We choose this dataset because it is smaller than the MPII-MD dataset, thus the model assessment experiment can be run in less time. The resulting values for the evaluation metrics, expressed in terms of mean and standard deviation, are reported in Tab. III. It is observed that our method is still comparable to the BA-LSTM baseline.

[3]https://github.com/tylin/coco-caption

We also evaluate the training and testing time of the ten different variants of both BA-GRU and BA-LSTM. In particular, for BA-GRU the test time is on average 190.89 ± 5.28 ms, while for BA-LSTM is on average 197.78 ± 3.70 ms. In terms of training time, for BA-GRU it is on average $\sim 8h21' \pm \sim 5h34'$, while for BA-LSTM it is on average $\sim 13h40' \pm \sim 3h22'$. Despite both the BA-GRU and the BA-LSTM require much time to complete the training phase, saving 5 hours for each training helps in faster iteration when tuning hyperparameters for network deployment. For example, in the case of our 10-fold cross validation we saved on average $50h$ with respect to the BA-LSTM model, and this could make the difference during the deployment of the architecture in a real robotic application.

The GRU block has fewer parameters than the LSTM block. In particular, our method BA-GRU requires approximately 114MB of memory to store network weights, while the BA-LSTM needs 128MB. Another benefit of using fewer parameters is that it reduces the risk of overfitting and, potentially, it allows the model to better generalize on completely new datasets.

E. Results on the ISARLab-VD Datasets

We further evaluate and compare BA-GRU with BA-LSTM on our collected dataset. Note that, in this case the algorithms are not trained on any subset of the ISARLab-VD dataset. With this experiment we want to test the generalization capabilities

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LRA.2018.2793345, IEEE Robotics and Automation Letters

Model	B_4	M	R_L	C
BA-LSTM on MSVD	14.0	19.5	51.6	23.3
BA-GRU on MSVD	**14.7**	**20.0**	**52.8**	**27.7**
BA-LSTM on MPII-MD	00.0	08.4	18.2	06.9
BA-GRU on MPII-MD	00.0	12.1	20.2	10.6

TABLE IV: Experiment results on the ISARLab-VD dataset in terms of the quantitative evaluation metrics BLEU in its *4*-gram variant (B_4), METEOR (M), ROUGE in its LCS variant (R_L) and CIDEr (C). Bold indicates the best performance.

Model	B_4	M	R_L	C
BA-LSTM	14.2±0.8	19.0±0.3	50.8±0.7	25.2±4.0
BA-GRU	15.0±1.0	19.4±0.5	51.2±0.8	24.7±2.6

TABLE V: Experiment results of the ten variants of the BA-GRU and BA-LSTM models obtained via *K*-fold cross-validation on the MSVD dataset in terms of the quantitative evaluation metrics BLEU in its *4*-gram variant (B_4), METEOR (M), ROUGE in its LCS variant (R_L) and CIDEr (C). The results are expressed in terms of mean and standard deviation.

of the two architectures. We report the results of both the BA-GRU and BA-LSTM architectures trained on either the MPII-MD and MSVD datasets, both in quantitative and qualitative terms.

In particular, in Tab. IV we report the results in terms of the previously defined evaluation metrics. For the statistical significance of those results, we refer to Tab. V. There we also report the results of the ten variants of the BA-GRU and BA-LSTM models obtained via *K*-fold cross-validation on the MSVD dataset.

Some examples are given in Fig. 4 showing high resolution and low resolution videos. The reported ground truth description is the most representative of the multiple caption associated to the clips. We refer to the complete results corpus available online[4] for further examples. It can be observed that the quality of the videos does not influence the semantic and syntactic correctness of the description produced by the two methods. On the other hand, we observe that the captions for the videos of the ISARLab-VD dataset are simpler and less precise than those produced for the test subset videos of the public dataset used for the training. This suggests that these NLVD systems do not generalize well with respect to scenarios that significantly differ from those observed in training phase. Despite that, we can observe that the use of the BA-GRU gives a slight performance improvement. This suggests that the BA-GRU could be better suited to achieve architectures more robust to domain changes. The exploration of this aspect is beyond the scope of this paper, but this insights could be definitely useful for future investigations.

V. Conclusions and Future Developments

This paper focuses on the NLVD task and presents a full-GRU encoder-decoder architecture to address it. We show that the proposed approach is faster to train and less memory consuming that other State-of-the-Art algorithms. Our method is also competitive in terms of performance on the public datasets which were partially used also for training. The experimental results on the devised dataset show that all methods have serious overfitting, making the generalization capabilities of new algorithm one of the most important questions to solve in future work.

Other future work is the ability to better cope with videos of variable lengths. This issue could be tackled by cutting the continuous video sequence in shorter chunks and describing each chunk using our proposed method as it is. However, being able to deal with much longer videos is surely of great interest and the development of effective solutions to this problem will be the subject of future work.

The code and the dataset used for this study are publicly available online.

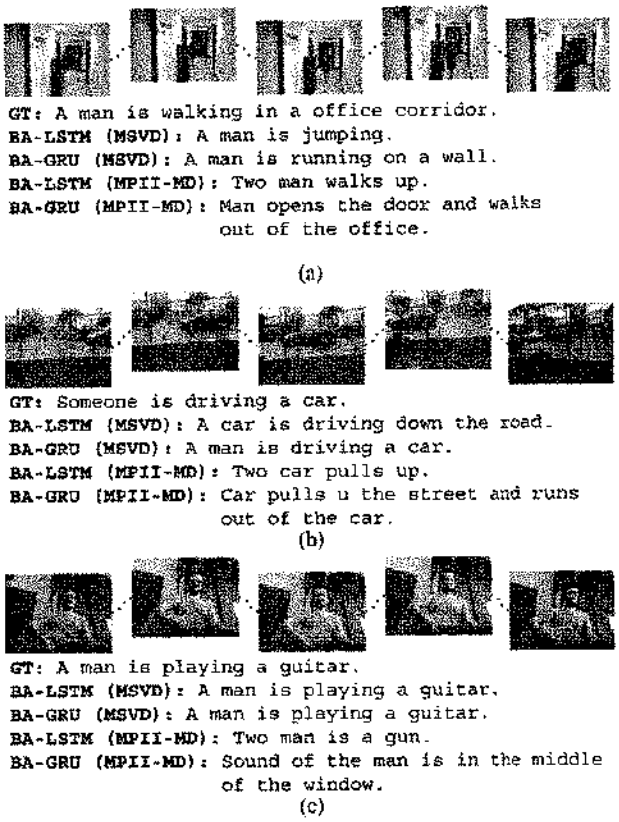


Fig. 4: Example results on videos from the ISARLab-VD dataset. In particular, 4a and 4b refer to videos that have been collected with two different hig resolution cameras, while 4c refers to a low resolution video collected during the experiments with the Anki's COZMO robot.

References

[1] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[2] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation," *arXiv preprint arXiv:1406.1078*, 2014.

[4] http://isar.unipg.it/index.php?option=com_content&view=article&id=46&catid=2&Itemid=188

[3] A. Rohrbach, M. Rohrbach, and B. Schiele, "The Long-Short Story of Movie Description," in *German Conference on Pattern Recognition*. Springer, 2015, pp. 209–221.

[4] A. Barbu, A. Bridge, Z. Burchill, D. Coroian, S. Dickinson, S. Fidler, A. Michaux, S. Mussman, S. Narayanaswamy, D. Salvi, *et al.*, "Video in Sentences Out," *arXiv preprint arXiv:1204.2742*, 2012.

[5] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, "Long-term Recurrent Convolutional Networks for Visual Recognition and Description," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 2625–2634.

[6] Y. Zhu, R. Kiros, R. Zemel, R. Salakhutdinov, R. Urtasun, A. Torralba, and S. Fidler, "Aligning Books and Movies: Towards Story-like Visual Explanations by Watching Movies and Reading Books," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 19–27.

[7] S. Venugopalan, M. Rohrbach, J. Donahue, R. Mooney, T. Darrell, and K. Saenko, "Sequence to Sequence-Video to Text," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 4534–4542.

[8] S. Venugopalan, H. Xu, J. Donahue, M. Rohrbach, R. Mooney, and K. Saenko, "Translating Videos to Natural Language using Deep Recurrent Neural Networks," *arXiv preprint arXiv:1412.4729*, 2014.

[9] M. Rohrbach, W. Qiu, I. Titov, S. Thater, M. Pinkal, and B. Schiele, "Translating Video Content to Natural Language Descriptions," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 433–440.

[10] Y. Pan, T. Mei, T. Yao, H. Li, and Y. Rui, "Jointly Modeling Embedding and Translation to Bridge Video and Language," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4594–4602.

[11] P. Pan, Z. Xu, Y. Yang, F. Wu, and Y. Zhuang, "Hierarchical Recurrent Neural Encoder for Video Representation with Application to Captioning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1029–1038.

[12] L. Baraldi, C. Grana, and R. Cucchiara, "Hierarchical Boundary-Aware Neural Encoder for Video Captioning," *arXiv preprint arXiv:1611.09312*, 2016.

[13] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[14] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[15] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning Spatiotemporal Features with 3D Convolutional Networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 4489–4497.

[16] Y. Bengio, N. Léonard, and A. Courville, "Estimating or Propagating Gradients Through Stochastic Neurons for Conditional Computation," *arXiv preprint arXiv:1308.3432*, 2013.

[17] A. Rohrbach, M. Rohrbach, N. Tandon, and B. Schiele, "A dataset for movie description," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3202–3212.

[18] S. Venugopalan, L. A. Hendricks, R. Mooney, and K. Saenko, "Improving LSTM-based Video Description with Linguistic Knowledge Mined from Text," *arXiv preprint arXiv:1604.01729*, 2016.

[19] L. Yao, A. Torabi, K. Cho, N. Ballas, C. Pal, H. Larochelle, and A. Courville, "Describing Videos by Exploiting Temporal Structure," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 4507–4515.

[20] D. L. Chen and W. B. Dolan, "Collecting Highly Parallel Data for Paraphrase Evaluation," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*. Association for Computational Linguistics, 2011, pp. 190–200.

[21] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: a Method for Automatic Evaluation of Machine Translation," in *Proceedings of the 40th annual meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 2002, pp. 311–318.

[22] S. Banerjee and A. Lavie, "METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments," in *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, vol. 29, 2005, pp. 65–72.

[23] C.-Y. Lin, "ROUGE: A Package for Automatic Evaluation of Summaries," in *Text summarization branches out: Proceedings of the ACL-04 workshop*, vol. 8. Barcelona, Spain, 2004.

[24] R. Vedantam, C. Lawrence Zitnick, and D. Parikh, "CIDEr: Consensus-based Image Description Evaluation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 4566–4575.

[25] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going Deeper with Convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1–9.

[26] S. Ji, W. Xu, M. Yang, and K. Yu, "3D Convolutional Neural Networks for Human Action Recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 1, pp. 221–231, 2013.

[27] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[28] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Advances in Neural Information Processing Systems*, 2012, pp. 1097–1105.

J-MOD^2: Joint Monocular Obstacle Detection and Depth Estimation

Michele Mancini[1], Gabriele Costante[1], Paolo Valigi[1] and Thomas A. Ciarfuglia[1]

***Abstract*—In this work, we propose an end-to-end deep architecture that jointly learns to detect obstacles and estimate their depth for MAV flight applications. Most of the existing approaches rely either on Visual SLAM systems or on depth estimation models to build 3D maps and detect obstacles. However, for the task of avoiding obstacles this level of complexity is not required. Recent works have proposed multi task architectures to perform both scene understanding and depth estimation. We follow their path and propose a specific architecture to jointly estimate depth and obstacles, without the need to compute a global map, but maintaining compatibility with a global SLAM system if needed. The network architecture is devised to jointly exploit the information learned from the obstacle detection task, which produces reliable bounding boxes, and the depth estimation one, increasing the robustness of both to scenario changes. We call this architecture J-MOD^2. We test the effectiveness of our approach with experiments on sequences with different appearance and focal lengths and compare it to SotA multi task methods that perform both semantic segmentation and depth estimation. In addition, we show the integration in a full system using a set of simulated navigation experiments where a MAV explores an unknown scenario and plans safe trajectories by using our detection model.**

***Index Terms*—Range Sensing, Visual Learning, Visual-Based Navigation**

I. Introduction

OBSTACLE avoidance has been deeply studied in robotics due to its crucial role for vehicle navigation. Recently, the demand for faster and more precise Micro Aerial Vehicle (MAV) platforms has put even more attention on it. To safely execute aggressive maneuvers in unknown scenarios, the MAVs need a robust obstacle detection procedure.

Most fruitful approaches rely on range sensors, such as laser-scanner, stereo cameras or RGB-D cameras [1], [2], [3] to build 3D maps and compute obstacle-free trajectories. However, their use results in an increased weight and power consumption, which is unfeasible for small MAVs. Furthermore, their sensing range is either limited by device characteristics (RGB-D and lasers) or by camera baselines (stereo cameras).

Manuscript received: September, 10, 2017; Revised November 8, 2017; Accepted January, 10, 2018.

*This paper was recommended for publication by Editor Cyrill Stachniss upon evaluation of the Associate Editor and Reviewers' comments. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan X GPU used for this research.

[1]All the authors are with the Department of Engineering, University of Perugia, via Duranti 93, Perugia Italy

{thomas.ciarfuglia, paolo.valigi, gabriele.costante, michele.mancini}@unipg.it

Digital Object Identifier (DOI): see top of this page.

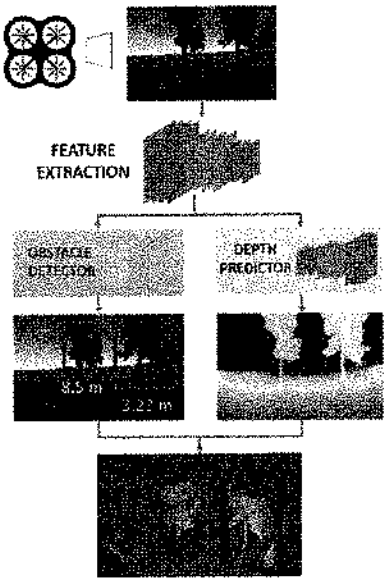


Fig. 1: Overview of the proposed system: the architecture is composed by two networks that perform different, but connected tasks: obstacle detection and pixel-wise depth estimation. The two task are jointly learned and the feature extraction layers are in common. Thus, the resulting model has increased accuracy in depth prediction because of the semantic information received from the detector. On the other hand, the detector learns a better representation of obstacles through depth estimation.

Monocular Visual SLAM (VSLAM) approaches address the above limitations by exploiting single camera pose estimation and 3D map reconstruction [4], [5], [6], [7]. Nevertheless, these advantages come with costs: the absolute scale is not observable (which easily results in wrong obstacle distance estimations); they fail to compute reliable 3D maps on low-textured environments; the 3D map updates are slow with respect to real-time requirements of fast manoeuvres. With careful tuning, these approaches can be used for obstacle avoidance.

At the same time there are other approaches that tackle the problem more specifically. In this respect, a step toward more robust obstacle detection has been made by monocular depth estimation methods based on Convolutional Neural Networks (CNNs) [8], [9], [10]. Compared to standard VSLAM strategies, these works train CNN-based model to quickly compute depth maps from single image, which allows for fast trajectory replanning. However, as any data-driven approach, these depth models are biased with respect to appearance domains and camera intrinsics. Most of the CNN architectures

so far proposed address the more general task of pixel-wise depth prediction and are not specifically devised for obstacle detection. However, recent works [11] [12] have digressed from this trail, proposing multi task network architectures to jointly learning depth and some semantic property of the images. These works show that the mutual information is beneficial to both tasks.

Driven by the previous considerations, in this work we propose a novel CNN architecture that jointly learns the task of depth estimation and obstacle detection. We aim to get, at the same time, the detection speed of CNNs approaches and more robustness to scale and appearance changes, using the joint learning of the depth distribution. The combination of these two tasks gives them mutual advantages: the depth prediction branch is informed with object structures, which result in more robust estimations. On the other hand, the obstacle detection model exploits the depth information to predict obstacle distance and bounding boxes more precisely. Our approach is similar to [11] and [12], but is specifically devised for obstacle detection, and not generic scene understanding, in order to achieve more robustness to appearance changes. We show the comparison with these two aforementioned methods in the experimental part of the work. We demonstrate the detection and depth estimation effectiveness of our approach in both publicly available and brand new sequences. In these experiments, we prove the robustness of the learned models in test scenarios that differ from the training ones with respect to focal length and appearance. In addition, to demonstrate the detection advantages of the proposed detection system, we set up a full navigation avoidance system in a simulated environment with a MAV that detects obstacles and computes free trajectories as it explores the scene.

II. Related Work

The most straight-forward approaches to obstacle detection and depth estimation involve RGB-D or stereo cameras. Unfortunately, these sensors suffer from limited range, in particular stereo systems, that require large baselines to achieve acceptable performances [13]. For example, some authors explored push-broom stereo systems on fixed-wing, high speed MAVs [14]. However, these approaches require too large baselines for small rotary wing MAVs. In addition, while short-range estimations still allows safe collision avoidance, it sets an upper bound to the robot's maximum operative speed. For all these reasons the study of alternative systems based on monocular cameras becomes relevant. Even with the limitation of monocular vision, our method can detect and localize obstacles up to 20 meters and compute dense depth maps up to 40 meters with a minor payload and space consumption.

Monocular obstacle detection can be achieved by dense 3D map reconstruction via SLAM or Structure from Motion (SFM) based procedures [6], [15], [16]. These systems perform a much more complex task though, and usually fail at high speeds, since they reconstruct the environment from frame to frame triangulation. In addition, with standard geometric monocular systems it is not possible to recover the absolute scale of the objects, without using additional information. In [17] the scale is recovered using the knowledge of the camera height from the ground plane, while [18] uses a inference based method on the average size of objects that frequently appear in the images (e.g. cars), then optimize to the whole trajectory. The lack of knowledge of the scale makes the obstacle avoidance a difficult task. For this reason, some approaches exploit optical information to detect proximity of obstacles from camera, or, similarly, detect traversable space, or use hand-crafted image features [19], [20], [21], [22], [23].

However, recently proposed deep learning-based solutions have shown robustness to the aforementioned issues. These models produce a dense 3D representation of the environment from a single image, exploiting the knowledge acquired through training on large labeled datasets, both real-world and synthetic [24], [8], [25], [9]. A few of these methods have been recently tested in obstacle detection and autonomous flight applications. In [26], the authors fine-tune on a self-collected dataset the depth estimation model proposed by [24] and use it for path planning. In [10] the authors exploit depth and normals estimations of a deep model presented in [8] as an intermediate step to train an visual reactive obstacle avoidance system. More recently, [10] proposed a similar approach, regressing avoidance paths directly from monocular 3D depth maps.

However, the aforementioned methods solve the task of depth estimation and from it derive the obstacle map. Another set of approaches use semantic knowledge to strengthen the detection task. On this line the works of [27], [11] and [12] train a multi task architecture for semantic scene understanding that is reinforced by the joint learning of a depth estimation task. However, these methods show better performances on classes such as "ground" or "sky". Our intuition is that current depth estimators overfit their predictions on these classes, as they tend to have more regular texture and geometric structures. On the contrary, in robotic applications we want to train detection models to be as accurate as possible when estimating obstacle distances.

Following this multi task approaches, we propose a novel solution to the problem by jointly training a model for depth estimation and obstacle detection. While each task's output comes from independent branches of the network, feature extraction from their common RGB input is shared for both targets. This choice improves both depth and detection estimations compared to single task models, as shown in the experiments. An approach similar to ours, applied to 3D bounding box detection, is presented in [28], where the authors train a three-loss model, sharing the feature extraction layers between the tasks.

In our system the obstacles bounding box regression part is obtained modifying the architecture of [29] making it fully convolutional. This allows for multiple bounding box predictions with a single forward pass. In addition, we also ask the obstacle detector to regress the average depth and the corresponding estimate variance of the detected obstacles.

Depth estimation is devised following the architecture of [9], improved by taking into account the obstacle detection branch. In particular, we correct the depth predictions by using the mean depth estimates computed by the obstacle detec-

tion branch to achieve robustness with respect to appearance changes. We prove the benefits of this strategy by validating the model in test sequences with different focal length and scene appearance. We compare our method to the ones of [11] and [12], showing a considerable increase of performances over these two baselines.

III. Network Overview

Our proposed network is depicted in Figure 2. Given an 256×160 RGB input, features are extracted with a fine-tuned version of the VGG19 network pruned of its fully connected layers [30]. VGG19 weights are initialized on the image classification task on the ImageNet dataset. Features are then fed to two, task-dependent branches: a depth prediction branch and a obstacle detector branch. The former is composed by 4 upconvolution layers and a final convolution layer which outputs the predicted depth at original input resolution. This branch, plus the VGG19 feature extractor, is equivalent to the fully convolutional network proposed in [9]. We optimize depth prediction on the following loss:

$$\begin{aligned} L_{depth} = & \frac{1}{n}\sum_i d_i^2 - \frac{1}{2n^2}(\sum_i d_i)^2 \\ & + \frac{1}{n}\sum_i [\nabla_x D_i + \nabla_y D_i] \cdot N_i^* \end{aligned} \tag{1}$$

where $d_i = \log D_i - \log D_i^*$, D_i and D_i^* are respectively the predicted and ground truth depths at pixel i, N_i^* is the ground truth 3D surface normal, and $\nabla_x D_i$, $\nabla_y D_i$ are the horizontal and vertical predicted depth gradients. While the first two terms correspond to the scale invariant log RMSE loss introduced in [24], the third term enforces orthogonality between predicted gradients and ground truth normals, aiming at preserving geometrical coherence. With respect to the loss proposed in [8], that introduced a L2 penalty on gradients to the scale invariant loss, our loss performs comparably in preliminary tests.

The obstacle detection branch is composed by 9 convolutional layer with Glorot initialization. The detection methodology is similar to the one presented in [29]: the input image is divided into a 8×5 grid of square-shaped cells of size 32×32 pixels. For each cell, we train a detector to estimate:

- The (x, y) coordinates of the bounding box center
- The bounding box width w and height h
- A confidence score C
- The average distance of the detected obstacle from the camera m and the variance of its depth distribution v

The resulting output has a 40×7 shape. At test time, we consider only predictions with a confidence score over a certain threshold. We train the detector on the following loss:

$$\begin{aligned} L_{det} = & \lambda_{coord} \sum_{i=0}^{N} [(x_i - x_i^*)^2 + (y_i - y_i^*)^2] \\ & + \lambda_{coord} \sum_{i=0}^{N} [(w_i - w_i^*)^2 + (h_i - h_i^*)^2] \\ & + \lambda_{obj} \sum_{i=0}^{N} (C_i - C_i^*)^2 + \lambda_{noobj} \sum_{i=0}^{N} (C_i - C_i^*)^2 \\ & + \lambda_{mean} \sum_{i=0}^{N} (m_i - m_i^*)^2 + \lambda_{var} \sum_{i=0}^{N} (v_i - v_i^*)^2 \end{aligned} \tag{2}$$

where we set $\lambda_{coord} = 0.25$, $\lambda_{obj} = 5.0$, $\lambda_{noobj} = 0.05$, $\lambda_{mean} = 1.5$, $\lambda_{var} = 1.25$. Our network is trained simultaneously on both tasks. Gradients computed by each loss are backpropagated through their respective branches and the shared VGG19 multi-task feature extractor.

A. Exploiting detection to correct global scale estimations

The absolute scale of a depth estimation is not observable from a single image. However, learning-based depth estimators are able to give an accurate guess of the scale under certain conditions. While training, these models implicitly learn domain-specific object proportions and appearances. This helps the estimation process in giving depth maps with correct absolute scale. As the relations between object proportions and global scale in the image strongly depend on camera focal length, at test time the absolute scale estimation are strongly biased towards the training set domain and its intrinsics. For these reasons, when object proportions and/or camera parameters change from training to test, scale estimates quickly degrade. Nonetheless, if object proportions stay roughly the same and only camera intrinsics are altered at test time, it is possible to employ some recovery strategy. If the size of a given object is known, we can analytically compute its distance from the camera and recover the global scale for the whole depth map. For this reason, we suppose that the obstacle detection branch can help recovering the global scale when intrinsics change. We hypothesize that, while learning to regress obstacles bounding boxes, a detector model implicitly learns sizes and proportions of objects belonging to the training domain. We can then evaluate estimated obstacle distances from the detection branch and use them as a tool to correct dense depth estimations. Let m_j be the average distance of the obstacle j computed by the detector, $\hat{D}_j$ the average depth estimation within the j-th obstacle bounding box, n_o the number of estimated obstacles, then we compute the correction factor k as:

$$k = \frac{\frac{1}{n_o}\sum_j^{n_o} m_j}{\frac{1}{n_o}\sum_j^{n_o} \hat{D}_j} \tag{3}$$

Finally, we calculate the corrected depth at each pixel i as $\tilde{D}_i = kD_i$. To validate our hypothesis, in Section IV-C we test on target domains with camera focal lengths that differ from the one used for training.

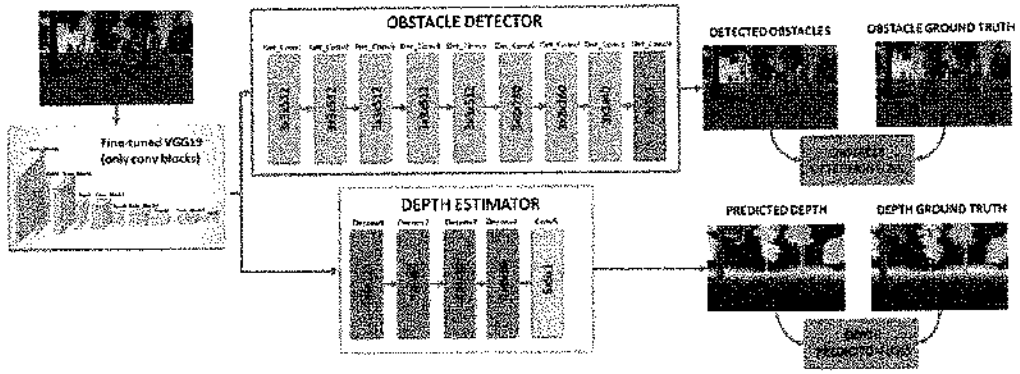


Fig. 2: Architecture of J-MOD^2. Given an RGB input, features are extracted by the VGG19 module and then fed into the depth estimation and obstacle detection branches to produce dense depth maps and osbtacles bounding boxes.

IV. Experiments

A. Datasets

1) UnrealDataset: UnrealDataset is a self-collected synthetic dataset that comprises of more than 100k images and 21 sequences collected in a bunch of highly photorealistic urban and forest scenarios with Unreal Engine and the AirSim plugin [31], which allows us to navigate a simulated MAV inside any Unreal scenarios. The plugin also allows us to collect MAV's frontal camera RGB images, ground truth depth up to 40 meters and segmentation labels. Some samples are shown in Figure 4(a). We postprocess segmentation labels to form a binary image depicting only two semantic classes: obstacle and non-obstacle by filtering these data with corresponding depth maps, we are finally able to segment obstacles at up to 20 meters from the camera and get ground truth labels for the detection network branch (Fig. 3). MAV's frontal camera has a horizontal field of view of 81,5 degrees.

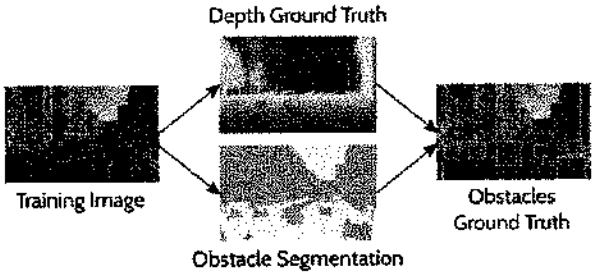


Fig. 3: Given depth and segmentation ground truth, we compute obstacle bounding boxes for each training image. We evaluate only obstacles in a 20 meters range.

2) Zurich Forest Dataset: Zurich Forest Dataset consist of 9846 real-world grayscale images collected with a hand-held stereo camera rig in a forest area. Ground truth depth maps are obtained for the whole dataset through semi-global stereo matching [32]. We manually draw 357 bounding boxes on a subset of 64 images to provide obstacle ground truth and evaluate detection in a real-world scenario.

B. Training and testing details

As baselines, we compare J-MOD^2 with:

- The depth estimation method proposed in [9].
- Our implementation of the multi-scale Eigen's model [8].
- A simple obstacle detector, consisting of our proposed model, trained without the depth estimation branch.
- Our implementation of the multi-modal autoencoder (later referred as Full-MAE) proposed by Cadena et al. [11].
- Our implementation of the joint refinement network (later referred as JRN) proposed by Jafari et al. [12].

We train J-MOD^2 and all the baseline models on 19 sequences of the UnrealDataset. We left out sequences 09 and 14 for testing. All the approaches have been trained on a single NVIDIA Titan X GPU. Training is performed with Adam optimizer by setting a learning rate of 0.0001 until convergence. The segmentation tasks for the Full-MAE and the JRN baselines are trained to classify two classes: "obstacle" and "not obstacle". The JRN is trained to fuse and refine depth estimations from our implementation of [8] with segmentation estimates from the SotA segmentation algorithm of Long et al. [33], as suggested by the authors, with the latter retrained on the 2-class segmentation problem of the UnrealDataset.

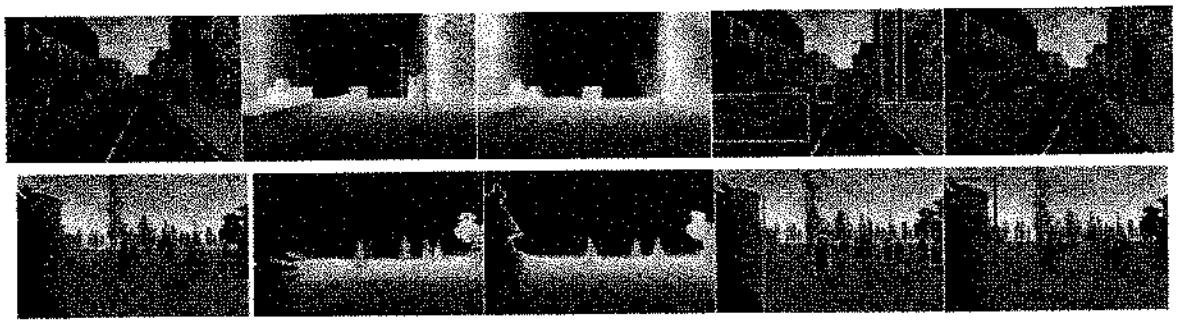
At test time, all baseline methods are tested using only RGB inputs. For both methods, we then infer obstacle bounding boxes from their depth and segmentation estimates applying the same procedure described in Figure 3, allowing direct comparison with our method. All the approaches are tested on the test sequences of the UnrealDataset and on the whole Zurich Forest Dataset. Note that, while testing on the latter, we do not perform any finetuning for both our method and the baselines.

At runtime, estimations require about 0.01 seconds per frame on a NVIDIA Titan X GPU. We also test J-MOD^2 on a NVIDIA TX1 board, to evaluate its portability on a on-board embedded system, measuring an average forward time of about 0.28 seconds per frame. The code for J-MOD^2 and all the baseline methods is available online[1]

To evaluate the depth estimator branch performance, we compute the following metrics:

- Linear RMSE and Scale Invariant Log RMSE ($\frac{1}{n}\sum_i d_i^2 - \frac{1}{n^2}(\sum_i d_i)^2$, with $d_i = \log y_i - \log y_i^*$) on the full depth map.

[1] http://isar.unipg.it/index.php?option=com_content&view=article&id=47&catid=2&Itemid=188



(a) RGB Input (b) Depth GT (c) Depth Estimation (d) Obstacle GT (e) Detected Obstacles

Fig. 4: J-MOD^2 qualitative results on the UnrealDataset.

	DEPTH [9]	DETECTOR	EIGEN [8]	FULL-MAE [11]	JRN [12]	J-MOD^2	
RMSE Full Depth Map	3.653	-	3.785	7.566	7.242	3.473	Lower
Sc.Inv RMSE Full Depth Map	0.042	-	0.043	0.124	0.110	**0.036**	is
Depth RMSE on Obs.(Mean/Var)	1.317 / 37.124	-	1.854 / 50.71	5.355/180.67	2.938 / 87.595	**1.034 / 29.583**	better
Detection RMSE on Obs.(Mean/Var)	-	2.307/ 59.407	-	-	-	**1.754 / 46.006**	
Detection IOU	-	63.11%	-	32.58%	44.19%	**66.58%**	Higher
Detection Precision	-	72.15%	-	75.53%	54.37%	**78.64%**	is
Detection Recall	-	90.05%	-	44.38%	49.55%	**90.85%**	better

TABLE I: Results on the UnrealDataset. For the depth estimation task we report full depth map RMSE and scale invariant errors, obstacle-wise depth and detection branches statistics (mean/variance) estimation errors and detector's IOU, precision and recall.

- Depth RMSE on Obstacles (Mean/Variance): For each ground truth obstacle, we compute its depth statistics (mean and variance) and we compare them against the estimated ones by using linear RMSE.

For the detector branch, we compute the following metrics:

- Detection RMSE on Obstacles (Mean/Variance):For each detected obstacle, we compare its estimated obstacle depth statistics (mean and variance) with the closest obstacle ones by using linear RMSE.
- Intersection Over Union (IOU)
- Precision and Recall.

C. Test on UnrealDataset

We report results on Table I. For [9] and [8] we report results only on depth-related metrics, as they do not perform any detection. Results confirm how J-MOD^2 outperforms all the other baselines in all metrics, corroborating our starting claim: object structures learned by the detector branch improve obstacles depth estimations of the depth branch. At the same time, localization and accuracy of the detected bounding boxes improve significantly compared to our single-task obstacle detector. We achieve good performances on both urban and forest sequences, without any significant discrepancy due to different depicted objects and contexts. We report qualitative results on Figure 4. According to the results on the NYU benchmark reported in [12], we expect JRN to outperform [8] on depth metrics, but this is not observed in this experiment. Our intuition is that the JRN segmentation network deals with a more challenging scenario, since the labels to the different objects are simply "obstacle", "not-obstacle" while in the original NYU there were specific labels for each object category. This makes this task for JRN similar to a semi-supervised learning problem, that is implicitly more difficult. Our system relies on a obstacle detector, that is a much simpler task to train, and therefore has an edge in this scenario.

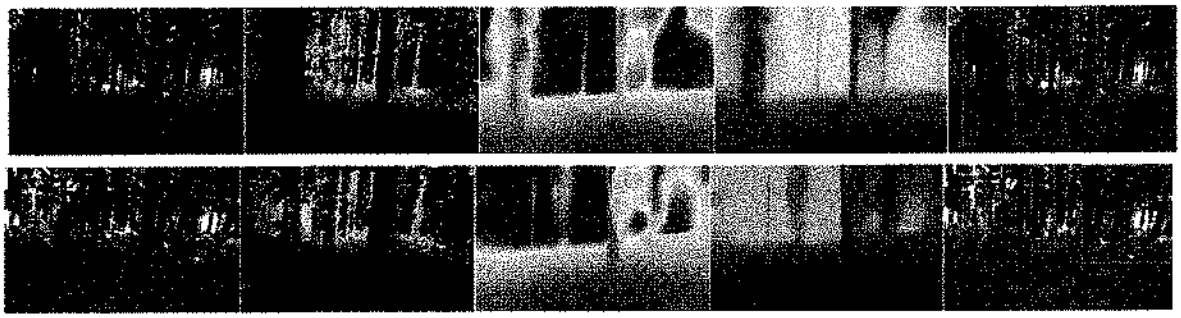
To validate our proposed depth correction strategy introduced in Section III-A, we also simulate focal length alterations by cropping and upsampling a central region of the input images of the UnrealDataset. We evaluate performances on different sized crops of images on the sequence-20, one of the training sequences, comprising of more than 7700 images. We choose to stage this experiment on a training sequence to minimize appearance-induced error and make evident the focal-length-induced error. We report results on Table II. When no crop is applied, camera intrinsics are unaltered and appearance-induced error is very low, as expected. As correction is applied linearly on the whole depth map, when scale-dependant error is absent or low, such correction worsen estimations by 19% on non-cropped images. A 230×144 crop simulates a slightly longer focal length. All metrics worsen, as expected, and correction still cause a 15% higher RMSE error. When 204×128 crops are evaluated, correction starts to be effective, improving performances by $1,45\%$ with respect to the non-corrected estimation. On 154×96 crops, correction leads to a 23% improvement. On 128×80 crops, correction improves performance by 25%. We also observe how the detection branch outperforms the depth estimation branch on obstacle distance evaluation as we apply wider crops to the input. This results uphold our hypothesis that detection branch is more robust to large mismatches between training and test camera focal lengths and can be used to partially compensate the induced absolute scale estimation deterioration.

D. Test: Zurich Forest Dataset

In this experiment we test our models, trained on synthetically generated data, on a real world scenario without performing any finetuning, to verify the generalization capabilities of the models when tested on unseen domains. Depth metrics (Linear RMSE and Scale Invariant MSE) refer to the whole dataset, while all the other metrics refer to the labelled subset, as described in Section IV-A2. Results are reported

	ORIGINAL SIZE		CROP 230X144		CROP 204X128		CROP 154X96		CROP 128X80	
	NoCor	Cor	NoCor	Cor	NoCor	Cor	NoCor	Cor	NoCor	Cor
RMSE Full Depth Map	2.179	2.595	**2.632**	3.042	4.052	**3.991**	8.098	**6.234**	10.825	**8.045**
Sc. Inv RMSE Full Depth Map	**0.096**	0.115	**0.121**	0.134	0.173	**0.164**	0.274	**0.217**	0.305	**0.250**
Depth RMSE on Obs.(Mean)	**0.185**	0.676	**1.293**	1.458	2.465	2.219	4.865	**3.583**	6.148	**4.485**
Detector RMSE on Obs.(Mean)	0.404		1.079		1.998		4.124		5.450	

TABLE II: Results of J-MOD^2 on the sequence-20 of the UnrealDataset on different-sized central crops. For each crop, we report in bold the better estimation between unchanged (labeled as NoCor) and corrected depths (labeled as WithCor).



(a) RGB Input (b) Depth GT (c) Non Corrected Depth Estimation (d) Corrected Depth Estimation (e) Detected Obstacles

Fig. 5: J-MOD^2 qualitative results on the Zurich Forest Dataset.

	DEPTH [9]		DETECTOR		EIGEN [8]		FULL-MAE [11]		JRN [12]		J-MOD^2	
	Cor	NoCor	Cor	NoCor	Cor	NoCor	Cor	NoCor	Cor	NoCor	Cor	NoCor
RMSE	-	12.421	-	-	-	14.640	-	17.581	-	10.114	**9.009**	12.569
Sc. Inv RMSE	-	0.873	-	-	-	1.025	-	1.711	-	0.702	**0.429**	0.954
Depth RMSE on Obs.(Mean)*	-	**4.378**	-	-	-	8.060	-	10.488	-	4.783	4.510	4.847
Detector RMSE on Obs.(Mean)*	-		6.277		-		-		-		**3.702**	
Detector IOU*	-		14.4%		-		2.13%		9.19%		**26.32%**	
Detector Precision*	-		25.32%		-		11.4%		13.18%		**48.36%**	
Detector Recall*	-		10.80%		-		1.12%		6.72%		**20.49%**	

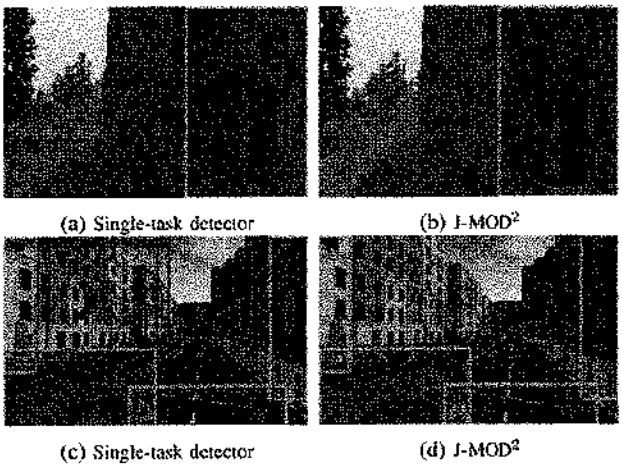
TABLE III: Results on the Zurich Forest Dataset. Metrics marked with a * symbol are evaluated on a subset of 64 images with ground truth bounding boxes.

on Table III. J-MOD^2 outperforms all baselines in almost all metrics, which suggests improved generalization capabilities. Furthermore, we show how the correction factor introduced in Section III-A improves J-MOD^2 depth estimation by about 28% on the RMSE metric, reducing the scale-induced errors on the estimates caused by the different camera parameters. We report qualitative results on Figure 5. The performance of all the approaches are lower with respect to the UnrealDataset. This is expected, since the synthetic textures and general appearance are different from the ones in this dataset. In addition, the camera characteristics do not match the ones of the UnrealDataset sequences.

E. Qualitative analysis of the multi-task interaction

Besides the advantages given by J-MOD^2 in terms of numerical performance, in the following, we qualitatively discuss the benefits of our joint architecture compared to its single task counterparts.

Figure 6 shows a comparison between the estimated obstacle bounding boxes of the detector-only architecture and the J-MOD^2 ones. It can be observed that, by exploiting the auxiliary depth estimation task, J-MOD^2 learns a detector that is aware of scene geometry. This results in an architecture that models a better concept of obstacle and, thus, is more precise in detecting what really determines a threat for the robot. Hence, it avoids wrong detections, such as ground surfaces

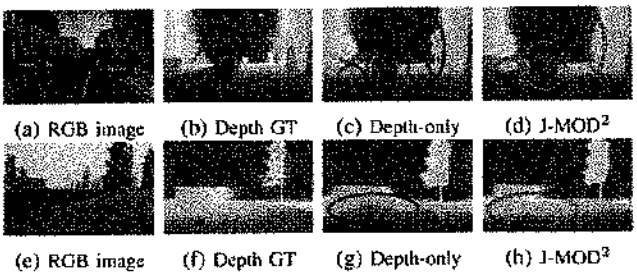


(a) Single-task detector (b) J-MOD^2

(c) Single-task detector (d) J-MOD^2

Fig. 6: For each row, we compare J-MOD^2 obstacle detections with the detector-only architecture. Ground truth bounding boxes are reported in green, predictions in red. In the first example (first row), the single-task detector erroneously detects a false obstacle on the ground. Similarly, in the second example (second row), the single-task wrongly considers the whole building on the left as an obstacle while only its closest part is an immediate threat for robot navigation.

(see Figures 6(a) and 6(b)), or full buildings of which only the closest part would constitutes an immediate danger for navigation (see Figures 6(c) and 6(d)).



(a) RGB image (b) Depth GT (c) Depth-only (d) J-MOD^2

(e) RGB image (f) Depth GT (g) Depth-only (h) J-MOD^2

Fig. 7: For each row, we compare J-MOD^2 depth maps with the ones predicted by the depth-only architecture. J-MOD^2 estimations are sharper and more defined. Consider, for example, the bollard and the lamppost in Figures 7(a)-7(d) or the ground surface in Figures 7(e)-7(h), whose depth is wrongly estimate by the depth-only estimator.

Similarly, depth estimation branch of the proposed J-MOD^2 approach takes advantage from the obstacle detector task to refine the estimation of the scene geometry. The representation learned by the J-MOD^2 depth estimation stream contains also visual clues about object shapes and proportions, which gives it the capability to integrate object semantics when estimating the scene depths. Compared to the depth-only architecture [9], our approach predicts sharper and more precise depth maps. This is more evident if we consider very thin elements and objects that could be mistaken for ground surfaces (*e.g.* consider the lamppost and the bollard in Figures 7(a)7(d) or the ground estimates in Figures 7(e)7(h)).

F. Navigation experiments

We further validate J-MOD^2 effectivness for obstacle detection applications by setting up a simulated full MAV navigation system. We depict the system architecture in Figure 8. We create a virtual forest scenario on Unreal Engine, slightly different from the one used for dataset collection. The line-of-sight distance between the takeoff point and the designed landing goal is about 61 meters. Trees are about 6 meters tall and spaced 7 meters from each other, on average. An aerial picture of the test scenario is reported in Figure 8.

A simulated MAV is able to navigate into the scenario and collect RGB images from its frontal camera. We estimate depth from the captured input and we employ it to dynamically build and update an Octomap [34]. We plan obstacle-free trajectories exploiting an off-the shelf implementation of the RRT-Connect planner [35] from the *MoveIt!* ROS library, which we use to pilot the simulated MAV at a cruise speed of $1m/s$. Trajectories are bounded to a maximum altitude of 5 meters. As a new obstacle is detected along the planned trajectory, the MAV stops and a new trajectory is computed. The goal point is set 4 meters above the ground. For each flight, we verify its success and measure the flight distance and duration. A flight fails if the MAV crashes or gets stuck, namely not completing its mission in a 5 minute interval. We compare J-MOD^2 with the Eigen's baseline, both trained on the UnrealDataset.

While planning, we add a safety padding on each Octomap obstacles. This enforces the planner to compute trajectories not too close to the detected obstacles. For each estimator, we set this value equal the average RMSE obstacle depth error on the UnrealDataset test set, as reported in Table I: 1.034 meters for J-MOD^2, 1.854 meters for Eigen. We refer to this value as a reliability measure of each estimator; the less accurate an estimator is, the more padding we need to guarantee safe operation. We perform 15 flights for each depth estimator and report their results on Table IV.

	EIGEN [8]	J-MOD^2
Success rate	26.6%	**73.3%**
Failure cases	8 stuck / 3 crash	2 stuck / 2 crash
Avg. flight time	147s	**131s**
Std. Dev. Flight Time	18.51s	**12.88s**
Avg. flight distance	78m	**77m**
Std. Dev. Flight Distance	**4.47m**	9.95m

TABLE IV: Results of the navigation experiment. We compare the navigation success rate when using J-MOD^2 and Eigen's approach as obstacle detection systems.

J-MOD^2 clearly performs better in all metrics, proving that how our method is effective for monocular obstacle detection. By analyzing failure cases, for 6 times the MAV using Eigen as obstacle detector got stuck in the proximity of goal point because ground was estimated closer than its real distance, causing planner failure in finding an obstacle-free trajectory to the goal. J-MOD^2 failures are mostly related on erratic trajectory computation which caused the MAV to fly too close to obstacles, causing lateral collisions or getting stuck in proximity of tree's leaves.

V. Conclusion and Future Work

In this work, we proposed J-MOD^2, a novel end-to-end deep architecture for joint obstacle detection and depth estimation. We demonstrated its effectiveness in detecting obstacles on synthetic and real-world datasets. We tested its robustness to appearance and camera focal length changes. Furthermore, we deployed J-MOD^2 as an obstacle detector and 3D mapping module in a full MAV navigation system and we tested it on a highly photo-realistic simulated forest scenario. We showed how J-MOD^2 dramatically improves mapping quality in a previously unknown scenario, leading to a substantial lower navigation failure rate than other SotA depth estimators. In future works, we plan to further improve robustness over appearance changes, as this is the major challenge for the effective deployment of these algorithms in practical real-world scenarios.

References

[1] S. Grzonka, G. Grisetti, and W. Burgard, "A fully autonomous indoor quadrotor," *IEEE Transactions on Robotics*, vol. 28, no. 1, pp. 90–100, 2012.

[2] F. Fraundorfer, L. Heng, D. Honegger, G. H. Lee, L. Meier, P. Tanskanen, and M. Pollefeys, "Vision-based autonomous mapping and exploration using a quadrotor mav," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*. IEEE, 2012, pp. 4557–4564.

[3] A. Bachrach, S. Prentice, R. He, P. Henry, A. S. Huang, M. Krainin, D. Maturana, D. Fox, and N. Roy, "Estimation, planning, and mapping for autonomous flight using an rgb-d camera in gps-denied environments," *The International Journal of Robotics Research*, vol. 31, no. 11, pp. 1320–1343, 2012.

[4] M. W. Achtelik, S. Lynen, S. Weiss, M. Chli, and R. Siegwart, "Motion- and uncertainty-aware path planning for micro aerial vehicles," *Journal of Field Robotics*, vol. 31, no. 4, pp. 676–698, 2014.

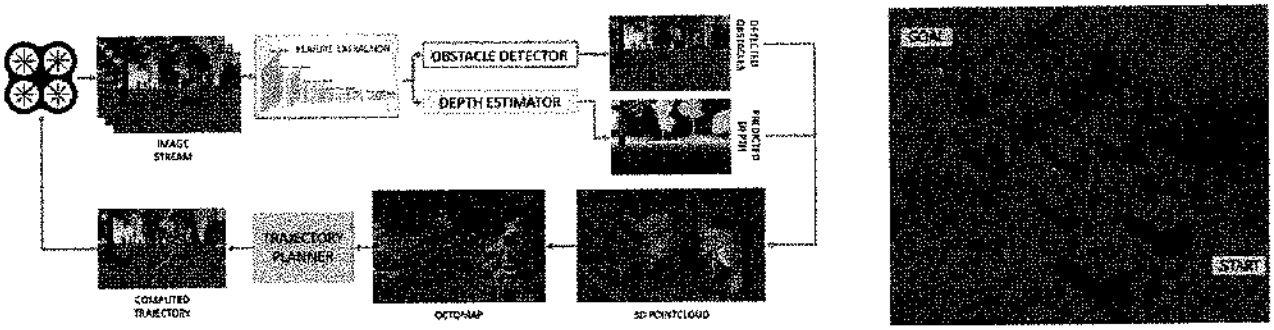


Fig. 8: Architecture of the full navigation pipeline (on the left) and a aerial picture of the test scenario (on the right). For each RGB image captured by the MAV frontal camera, a depth map is computed and converted into a point cloud used to update the 3D map and compute an obstacle-free trajectory. The MAV then flies along the computed trajectory until a new obstacle is detected.

[5] D. Scaramuzza, M. C. Achtelik, L. Doitsidis, F. Friedrich, E. Kosmatopoulos, A. Martinelli, M. W. Achtelik, M. Chli, S. Chatzichristofis, L. Kneip, *et al.*, "Vision-controlled micro flying robots: from system design to autonomous navigation and mapping in gps-denied environments," *IEEE Robotics & Automation Magazine*, vol. 21, no. 3, pp. 26–40, 2014.

[6] J. Engel, T. Schöps, and D. Cremers, "Lsd-slam: Large-scale direct monocular slam," in *European Conference on Computer Vision*. Springer, 2014, pp. 834–849.

[7] C. Forster, M. Pizzoli, and D. Scaramuzza, "Svo: Fast semi-direct monocular visual odometry," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE, 2014, pp. 15–22.

[8] D. Eigen and R. Fergus, "Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 2650–2658.

[9] M. Mancini, G. Costante, P. Valigi, T. A. Ciarfuglia, J. Delmerico, and D. Scaramuzza, "Towards domain independence for learning-based monocular depth estimation," *IEEE Robotics and Automation Letters*, 2017.

[10] S. Yang, S. Konam, C. Ma, S. Rosenthal, M. Veloso, and S. Scherer, "Obstacle avoidance through deep networks based intermediate perception," *arXiv preprint arXiv:1704.08759*, 2017.

[11] C. Cadena, A. Dick, and I. Reid, "Multi-modal auto-encoders as joint estimators for robotics scene understanding," in *Proceedings of Robotics: Science and Systems*, AnnArbor, Michigan, June 2016.

[12] O. H. Jafari, O. Groth, A. Kirillov, M. Y. Yang, and C. Rother, "Analyzing modular CNN architectures for joint depth prediction and semantic segmentation," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, May 2017, pp. 4620–4627.

[13] C. Nous, R. Meertens, C. De Wagter, and G. de Croon, "Performance evaluation in obstacle avoidance," in *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*. IEEE, 2016, pp. 3614–3619.

[14] A. J. Barry and R. Tedrake, "Pushbroom stereo for high-speed navigation in cluttered environments," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, May 2015, pp. 3046–3052.

[15] M. Pizzoli, C. Forster, and D. Scaramuzza, "Remode: Probabilistic, monocular dense reconstruction in real time," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 2609–2616.

[16] H. Alvarez, L. M. Paz, J. Sturm, and D. Cremers, "Collision avoidance for quadrotors with a monocular camera," in *Experimental Robotics*. Springer, 2016, pp. 195–209.

[17] A. Geiger, J. Ziegler, and C. Stiller, "Stereoscan: Dense 3d reconstruction in real-time," in *Intelligent Vehicles Symposium (IV), 2011*.

[18] D. P. Frost, O. Kähler, and D. W. Murray, "Object-aware bundle adjustment for correcting monocular scale drift," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, May 2016, pp. 4770–4776.

[19] S. Ross, N. Melik-Barkhudarov, K. S. Shankar, A. Wendel, D. Dey, J. A. Bagnell, and M. Hebert, "Learning monocular reactive uav control in cluttered natural environments," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 2013, pp. 1765–1772.

[20] S. Daftry, S. Zeng, A. Khan, D. Dey, N. Melik-Barkhudarov, J. A. Bagnell, and M. Hebert, "Robust monocular flight in cluttered outdoor environments," *CoRR*, vol. abs/1604.04779, 2016.

[21] A. Beyeler, J.-C. Zufferey, and D. Floreano, "Vision-based control of near-obstacle flight," *Autonomous robots*, vol. 27, no. 3, pp. 201–219, 2009.

[22] C. Bills, J. Chen, and A. Saxena, "Autonomous mav flight in indoor environments using single image perspective cues," in *Robotics and automation (ICRA), 2011 IEEE international conference on*. IEEE, 2011, pp. 5776–5783.

[23] T. Mori and S. Scherer, "First results in detecting and avoiding frontal obstacles from a monocular camera for micro unmanned aerial vehicles," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 2013, pp. 1750–1757.

[24] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," in *Advances in neural information processing systems*, 2014, pp. 2366–2374.

[25] F. Liu, C. Shen, G. Lin, and I. Reid, "Learning depth from single monocular images using deep convolutional neural fields," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 10, pp. 2024–2039, Oct 2016.

[26] P. Chakravarty, K. Kelchtermans, T. Roussel, S. Wellens, T. Tuytelaars, and L. Van Eycken, "Cnn-based single image obstacle avoidance on a quadrotor," in *Robotics and Automation (ICRA), 2017 IEEE International Conference on*. IEEE, 2017, pp. 6369–6374.

[27] A. Kendall, Y. Gal, and R. Cipolla, "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics," *CoRR*, vol. abs/1705.07115, 2017.

[28] A. Mousavian, D. Anguelov, J. Flynn, and J. Košecká, "3d bounding box estimation using deep learning and geometry," in *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*. IEEE, 2017, pp. 5632–5640.

[29] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 779–788.

[30] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[31] S. Shah, D. Dey, C. Lovett, and A. Kapoor, "Airsim: High-fidelity visual and physical simulation for autonomous vehicles," *arXiv preprint arXiv:1705.05065*, 2017.

[32] H. Hirschmuller, "Stereo processing by semiglobal matching and mutual information," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 30, no. 2, pp. 328–341, 2008.

[33] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3431–3440.

[34] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "Octomap: An efficient probabilistic 3d mapping framework based on octrees," *Autonomous Robots*, vol. 34, no. 3, pp. 189–206, 2013.

[35] J. J. Kuffner and S. M. LaValle, "Rrt-connect: An efficient approach to single-query path planning," in *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, vol. 2. IEEE, 2000, pp. 995–1001.

LS-VO: Learning Dense Optical Subspace for Robust Visual Odometry Estimation

Gabriele Costante[†,1] and Thomas A. Ciarfuglia[†,1]

***Abstract*—This work proposes a novel deep network architecture to solve the camera Ego-Motion estimation problem. A motion estimation network generally learns features similar to Optical Flow (OF) fields starting from sequences of images. This OF can be described by a lower dimensional latent space. Previous research has shown how to find linear approximations of this space. We propose to use an Auto-Encoder network to find a non-linear representation of the OF manifold. In addition, we propose to learn the latent space jointly with the estimation task, so that the learned OF features become a more robust description of the OF input. We call this novel architecture Latent Space Visual Odometry (LS-VO). The experiments show that LS-VO achieves a considerable increase in performances with respect to baselines, while the number of parameters of the estimation network only slightly increases.**

***Index Terms*—Computer Vision for Transportation, Deep Learning in Robotics and Automation, Visual Learning, Visual-Based Navigation**

I. Introduction

Learning based Visual Odometry (L-VO) in the last few years has seen an increasing attention of the robotics community because of its desirable properties of robustness to image noise and camera calibration independence [1], mostly thanks to Convolutional Neural Networks (CNNs) representational power, which can complement current geometric solutions [2]. While current results are very promising, making these solutions easily applicable to different environments still presents challenges. One of them is that most of the approaches so far explored have not shown strong domain independence and suffer from high dataset bias, i.e. the performances considerably degrade when tested on sequences with motion dynamics and scene depth significantly different from the training data [3]. In the context of L-VO this bias is expressed in different Optical Flow (OF) field distribution in training and test data, due to differences in scene depth and general motion of the camera sensor.

One possible explanation for the poor performances of learned methods on unseen contexts is that most current learning architectures try to extract both visual features and motion estimate as a single training problem, coupling the appearance and scene depth with the actual camera motion information contained in the OF input. Some works have addressed the problem with an unsupervised, or semi-supervised approach, trying to learn directly the motion representation and scene depth from some kind of frame-to-frame photometric error [4] [5] [6]. While very promising, these approaches are mainly devised for scene depth estimation and still fall short in terms of general performances on Ego-Motion estimation.

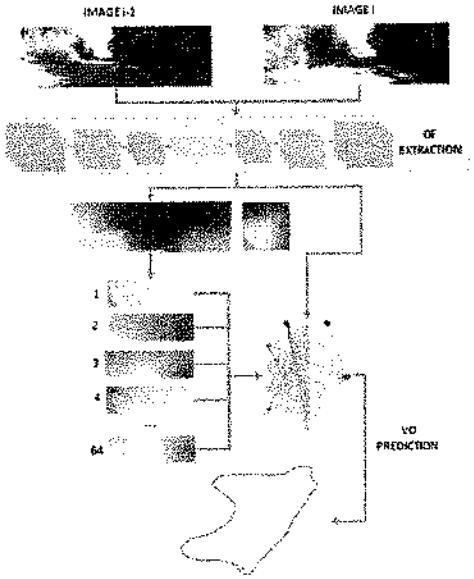


Fig. 1: Overview of the method: We propose a network architecture that jointly learn a latent space representation of the Optical Flow field and estimates motion. The joint learning makes the estimation more robust to input domain changes. The latent representation is an input to the estimation network together with the lower level features.

At the same time, previous research has shown how OF fields have a bilinear dependence on motion and inverse scene depth [7]. We suggest that this is the main reason for the low generalization properties shown by learned algorithms so far. Past research has shown that the high dimensional OF field, when scene depth can be considered locally constant, can be projected on a much lower dimensional linear space [8] [9]. However, when these conditions do not hold, the OF field subspace exists but is highly non-linear.

In this work we propose to exploit this knowledge, estimating the latent OF representation using an Auto-Encoder (AE) Neural Network architecture as a non-linear subspace approximator. AE networks are able to extract latent variable representation of high dimensional inputs. Since our aim is

Manuscript received: September, 10, 2017; Revised December, 12, 2017; Accepted January, 30, 2018.

This paper was recommended for publication by Editor Jana Kosecka upon evaluation of the Associate Editor and Reviewers' comments.

We gratefully acknowledge the support of NVIDIA Corporation with the donation of the *Titan Xp* GPU used for this research.

† The authors contributed equally to the work.

[1]Department of Engineering, University of Perugia, Italy
{thomas.ciarfuglia, gabriele.costante}@unipg.it

Digital Object Identifier (DOI): see top of this page.

to make the Ego-Motion estimation more robust to OF fields that show high variability in their distribution, we do not simply use this subspace to directly produce motion prediction. Instead, we propose a novel architecture that jointly trains the subspace estimation and Ego-Motion estimation so that the two network tasks are mutually reinforcing and at the same time able to better generalize OF field representation. The conceptual architecture is shown in Figure 1. To demonstrate the increased performances and reduced dataset bias with respect to high dynamical variation of the OF field, we test the proposed approach on a challenging scenario. We sub-sample the datasets, producing sequences that simulate high speed variations, then we train and test on sequences that are both different in appearance and sub-sampling rate.

II. Related Works

A. Ego-Motion estimation

1) Geometric Visual Odometry: G-VO has a long history of solutions. While the first approaches were based on sparse feature tracking, mainly for computational reasons, nowadays direct or semi-direct approaches are preferred. These approaches use the photometric error as an optimization objective. Research on this topic is very active. Engel at al. developed one of the most succesful direct approaches, LSD SLAM, both for monocular and stereoscopic cameras [10], [11]. Forster et al. developed the Semi-Direct VO (SVO) [12] and its more recent update [13], which is a direct method but tracks only a subset of features on the image and runs at very high frame rate compared to full direct methods. Even if direct methods have gained most of the attention in the last few years, the ORB-SLAM algorithm by Mur-Artal et al. [14] reverted to sparse feature tracking and reached impressive robustness and accuracy comparable with direct approaches.

2) Learned Visual Odometry: Learned approaches go back to the early explorations by Roberts et al. [8], [15], Guizilini et al. [16], [17], and Ciarfuglia et al. [18]. As for the geometric case, the initial proposal focused on sparse OF features that, faithful to the *there's no free lunch theorem*, explored the performances of different learning algorithms such as SVMs, Gaussian Processes and others. While these early approaches already showed some of the strengths of L-VO, it was only more recently, when Costante et al. [1] introduced the use of CNNs for feature extraction from dense optical flow, that the learned methods started to attract more interest. Since then a couple of methods have been proposed. Muller and Savakis [19] added the FlowNet architecture to the estimation network, producing one of the first end-to-end approaches. Clark et al. [20] proposed an end-to-end approach that merged camera inputs with IMU readings using an LSTM network. Through this sensor fusion, the resulting algorithm is able to give good results but requires sensors other than a single monocular camera. The use of LSTM is further explored by Wang et al. in [21], this time without any sensor fusion. The resulting architecture gives again good performances on KITTI sequences but does not show any experiments on environments with different appearance from the training sequences. On a different track is the work of Pillai et al. [22], that, like [17], looked at the problem as a generative probabilistic problem. Pillai proposes an architecture based on an MDN network and a Variational Auto-Encoder (VAE) to estimate the motion density given the OF inputs as a GMM. While Frame to Frame (F2F) performances are on a par with other approaches, they also introduce a loss term on the whole trajectory that mimics the bundle optimization that is often used in G-VO. The results of the complete system are thus very good. However, they use as input sparse KLT optical flow, since the joint density estimation for dense OF would become computationally intractable, meaning that they could be more prone to OF noise than dense methods.

Most of the described approaches claim independence from camera parameters. While this is true, we note that this is more an intrinsic feature of the learning approach than the merit of a particular architecture. The learned model implicitly learns also the camera parameters, but then it fails on images collected with other camera optics. This parameter generalization issue remains an open problem for L-VO.

B. Semi-supervised Approaches

Since dataset bias and domain independence are critical challenges for L-VO, it is not surprising that a number of unsupervised and semi-supervised methods have been recently proposed. However, all the architectures have been proposed as a way of solving the more general problem of joint scene depth and motion estimation, and motion estimation is considered more as a way of improving depth estimation. Konda and Memisevich [23] used a stereo pair to learn VO but the architecture was conceived only for stereo cameras. Ummenhofer and Zhou [4] propose the DeMoN architecture, a solution for F2F Structure from Motion (SfM) that trains a network end-to-end on image pairs, levering motion parallax. Zhou et al. [5] proposed an end-to-end unsupervised system based on a loss that minimizes image warping error from one frame to the next. A similar approach is used by Vijayanarasimhan et al. [6] with their SfM-Net.

All these approaches are devised mainly for depth estimation and the authors give little or no attention to the performances on VO tasks. Nonetheless, the semi-supervised approach is one of the more relevant future directions for achieving domain independence for L-VO, and we expect that this approach will be integrated in the current research on this topic.

C. Optical Flow Latent Space Estimation

The semi-supervised approaches described in Section II-B make evident an intrinsic aspect of monocular camera motion estimation, that is, even when the scene is static, the OF field depends both on camera motion and scene depth. This relationship between inverse depth and motion is bilinear and well known [24] and is at the root of scale ambiguity in monocular VO. However, locally and under certain hypothesis of depth regularity, it is possible to express the OF field in terms of a linear subspace of OF basis vectors. Roberts et al. [15] used Probabilistic-PCA to learn a lower dimensional dense OF subspace without supervision, then used it

to compute dense OF templates starting from sparse optical flow. They then used it to compute Ego-Motion. Herdtweck and Cristóbal extended the result and used Expert Systems to estimate motion [25]. More recently, a similar approach to OF field computation was proposed by Wulff and Black [9] that complemented the PCA with MRF, while Ochs et al. [26] did the same by including prior knowledge with an MAP approach. These methods suggest that OF field, which is an intrinsically high dimensional space generated from a non-linear process, lies on an ideal lower dimensional manifold that sometimes can be linearly locally approximated. However, modern deep networks are able to find latent representation of high dimensional image inputs, and in this work we use this intuition to explore this OF latent space estimation.

III. Contribution

Inspired by the early work of Roberts on OF subspaces [7], and by recent advances in deep latent space learning [27], we propose a network architecture that jointly estimates a low dimensional representation of dense OF field using an Auto-Encoder (AE) and at the same time computes the camera Ego-Motion estimate with a standard Convolutional network, as in [1]. The two networks share the feature representation in the decoder part of the AE, and this constrains the training process to learn features that are compatible with a general latent subspace. We show through experiments that this joint training increases the Ego-Motion estimation performances and generalization properties. In particular, we show that learning the latent space and concatenating it to the feature vector makes the resulting estimation considerably more robust to domain change, both in appearance and in OF field dynamical range and distribution.

We train our network both in an end-to-end version, using deep OF estimation, and with standard OF field input, in order to explore the relative advantages and weaknesses. We show that while the end-to-end approach is more general, precomputed OF still has some performance advantages.

In summary our contributions are:

- A novel end-to-end architecture to jointly learn the OF latent space and camera Ego-Motion estimation is proposed. We call this architecture Latent Space-VO (LS-VO).
- The strength of the proposed architecture is demonstrated experimentally, both for appearance changes, blur, and large camera speed changes.
- Effects of geometrically computed OF fields are compared to end-to-end architectures in all cases.
- The adaptability of the proposed approach to other end-to-end architectures is demonstrated, without increasing the chances of overfitting them, due to parameters increase.

IV. Learning Optical Flow Subspaces

Given an optical flow vector $\mathbf{u} = (\mathbf{u}_x^\intercal, \mathbf{u}_y^\intercal)^\intercal$ from a given OF field $\mathbf{x}$, [7] [9] approximate it with a linear relationship:

$$\mathbf{u} \approx \mathbf{W}\mathbf{z} = \sum_{i=1}^{l} z_i \mathbf{w}_i \tag{1}$$

where the columns of $\mathbf{W}$ are the basis vectors that form the OF linear subspace and $\mathbf{z}$ is a vector of latent variables. This approximation is valid only if there are some regularities of scene depth and is applicable only to local patches in the image. The real subspace is non-linear in nature and, in this work, we express it as a generic function $\mathbf{u} = \mathcal{D}(\mathbf{z})$ that we learn from data by using the architecture described in the following.

A. Latent Space Estimation with Auto-Encoder Networks

Let $\mathbf{y} \in \mathbb{R}^6$ be the camera motion vector and $\mathbf{x} \in \mathbb{R}^{2\times w\times h}$ the input OF field, computed with some dense method, where $\mathbf{x}_{(i,j)} = \mathbf{u}_{(i,j)}$ is a 2-dimensional vector of the field at image coordinates (i, j). Both can be viewed as random variables with their own distributions. In particular, we make the hypothesis that the input images lie on a lower dimensional manifold, as in [28], and thus also the OF field lies on a lower dimensional space $\mathbb{O} \subset \mathbb{R}^{2\times w\times h}$ with a distance function $S(\mathbf{x}^{(a)}, \mathbf{x}^{(b)})$, where $\mathbf{x}^{(a)}, \mathbf{x}^{(b)} \in \mathbb{O}$. The true manifold is very difficult to compute, so we look for an estimate $\hat{\mathbb{O}} \approx \mathbb{O}$ using the model extracted by an encoding neural network.

Let $\mathbf{z} \in \mathbb{Y} \subset \mathbb{R}^l, l \ll w \times h$ be a vector of latent random variables that encodes the variabilities of OF field that lies on this approximate space. The decoder part of the AE can be seen as a function

$$\mathcal{D}(\mathbf{z}; \theta_d) = D(\mathbf{z}; \{\mathbf{W}_k, \mathbf{b}_k\}, k = 1 \cdots K) \tag{2}$$

where $\theta_d = (\{\mathbf{W}_k, \mathbf{b}_k\}, k = 1 \cdots K)$ is the set of learnable parameters of the network (with K upconv layers), that is able to generate a dense optical flow from a vector of latent variables $\mathbf{z}$. Note that the AE works similarly to a non-linear version of PCA [27]. We define the set $\hat{\mathbb{O}} = \{D(\mathbf{z}; \theta_d) \mid \mathbf{z} \in \mathbb{Y}\}$ as our approximation of the OF field manifold and use the logaritmic Euclidean distance (as described in Section IV-B as a loss function) as an approximation of $S(D(\mathbf{z}^{(a)}), D(\mathbf{z}^{(b)}))$. Using this framework the problem of estimating the latent space is carried out by the AE network, where the Encoder part can be defined as the function $\mathbf{z} = E(\mathbf{x}; \theta_e)$.

While in [22] the AE is used to estimate motion, and $\mathbf{z}$ are the camera translation and rotations, here we follow a different strategy. We compute the latent space for a two-fold purpose: we use the latent variables as an input feature to the motion estimation network and we learn this latent space together with the estimator, thus forcing the estimator to learn features compatible with the encoder representation. Together these two aspects make the representation more robust to domain changes.

B. Network Architecture

The LS-VO network architecture in its end-to-end form is shown in Figure 2. It is composed of two main branches, one is the AE network and the other is the convolutional network that computes the regression of motion vector $\mathbf{y}$. The OF extraction section is Flownet [29], for which we use the pre-trained weights. We run tests fine-tuning this part of the network on KITTI [30] and Malaga [31] datasets, but the result was a degraded performance due to overfitting.

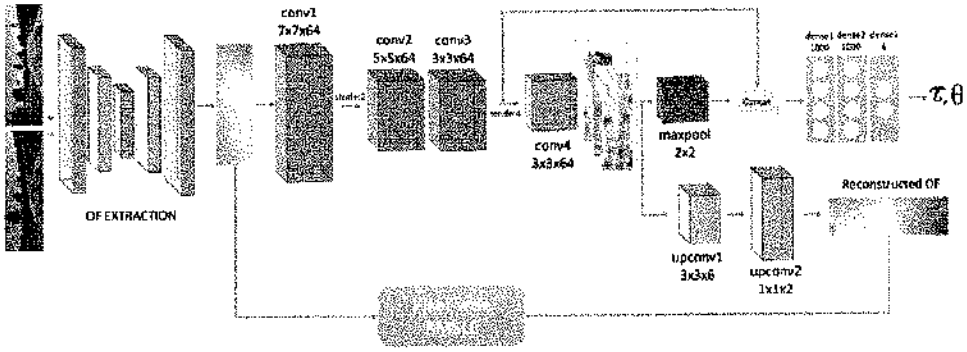


Fig. 2: LS-VO network architecture. The shared part is composed of Flownet OF extraction, then three convolutional layers that start the feature extraction. The last layer of the Encoder, conv4, is not shared with the Estimator network. From conv4 the latent variables are produced. The Decoder network takes these variables and reconstructs the input, while the Estimator concatenates them to conv3 output. Then three fully connected layers produce the motion estimates.

The next layers are convolutions that extract features from the computed OF field. After the first convolutional layers (conv1, conv2 and conv3), the network splits into the AE network and the estimation network. The two branches share part of the feature extraction convolutions, so the entire network is constrained in learning a general representation that is good for estimation and latent variable extraction. The Encoder is completed by another convolutional layer, that brings the input $\mathbf{x}$ to the desired representation $\mathbf{z}$, and its output is fed both in the Decoder and concatenated to the feature extracted before. The resulting feature vector, composed of latent variables and convolutional features is fed into a fully connected network that performs motion estimation. The details are summarized in Table I.

The AE is trained with a pixel-wise squared Root Mean Squared Log Error (RMSLE) loss:

$$\mathcal{L}_{AE} = \sum_i || \log(\hat{\mathbf{u}}^{(i)} + \mathbf{1}) - \log(\mathbf{u}^{(i)} + 1)||_2^2 \qquad (3)$$

where $\hat{\mathbf{u}}^{(i)}$ is the predicted OF vector for the i-th pixel, and $\mathbf{u}^{(i)}$ is the corresponding input to the network, and the logarithm is intended as an element-wise operation. This loss penalizes the ratio difference, and not the absolute value difference of the estimated OF compared to the real one, so that the flow vectors of distant points are taken into account and not smoothed off.

We use the loss introduced by Kendall et al. in [32]:

$$\mathcal{L}_{EM} = \sum_i ||\hat{\tau} - \tau||_2^2 + \beta||\hat{\theta} - \theta||_2^2 \qquad (4)$$

where the τ is camera translation vector in meters, θ is the rotation vector in Euler notation in radians, and β is a scale factor that balances the angular and translational errors. β has been cross-validated on the trajectory reconstruction error ($\beta = 20$ for our experiments), so that the frame to frame error propagation to the whole trajectory is taken into account. The use of a Euclidean loss with Euler angle representation works well in the case of autonomous cars, since the yaw angle is the only one with significant changes. For more general cases, is better to use a quaternion distance metric [33].

TABLE I: LS-VO and ST-VO network architectures

	Layer name	Kernel size	Stride	output size
Input	-	-	-	(94, 300, 2)
	LS-VO			
Shared Features Layer	conv1	7×7	2×2	(47, 150, 64)
	conv2	5×5	1×1	(47, 150, 64)
	conv3 $\star$	3×3	4×4	(12, 38, 64)
Auto-Encoder	conv4	3×3	1×1	(12, 38, 64)
	upconv1	3×3	1×1	(48, 152, 6)
	crop	-	-	(47, 150, 6)
	upconv2	1×1	1×1	(94, 300, 2)
Estimator	maxpool $\dagger$	2×2	2×2	(6, 19, 64)
	concat $\star$ and $\dagger$	-	-	(36480)
	dense1	-	-	(1000)
	dense2	-	-	(1000)
	dense3	-	-	(6)
	ST-VO			
Feature Extraction	st-conv1	3×3	2×2	(46, 149, 64)
	st-maxpool1 $\bullet$	4×4	4×4	(11, 37, 64)
	st-conv2	4×4	4×4	(9, 35, 20)
	st-maxpool2 $\diamond$	2×2	2×2	(4, 17, 20)
Estimation	concat $\bullet$ and $\diamond$	-	-	(27408)
	st-dense1	-	-	(1000)
	st-dense2	-	-	(6)

In Section V-B, we compare this architecture both with SotA geometrical and learned methods. The baseline for the learned approaches is a Single Task (ST) network, similar to the 1b network presented in [1], and described in Table I.

C. OF field distribution

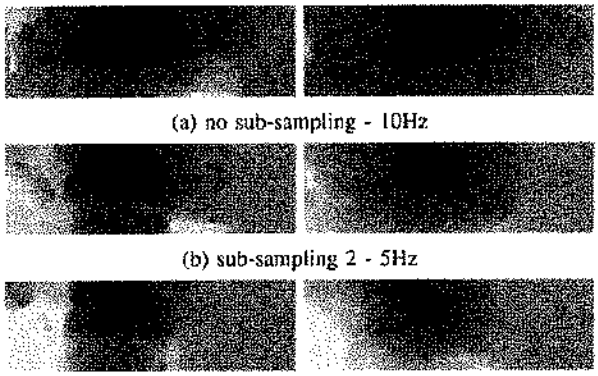
As mentioned in Section IV-A, the OF field has a probability distribution that lies on a manifold with lower dimensionality than the number of pixels of the image. We can argue that the actual density depends on the motion of the camera as much as the scene depth of the images collected. In this work, we test generalization properties of the network for both aspects:

i For the appearance we use the standard approach to test on completely different sequences than the ones used in training.

ii For the motion dynamics, we sub-sample the sequences, thus multiplying the OF dynamics by the same factor.

iii To further test OF distribution robustness, we also test the architecture on downsampled blurred images, as in [1].

Examples of the resulting OF field are shown in Figure 3, while an example of a blurred OF field is shown in Figure 4.

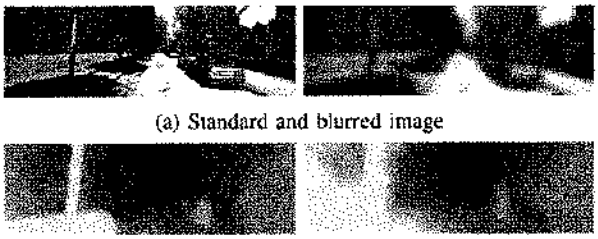


(a) no sub-sampling - 10Hz

(b) sub-sampling 2 - 5Hz

(c) sub-sampling 3 - 3.33Hz

Fig. 3: Examples of the OF field intensity due to different sub-sampling rates of the original sequences. In the left are the OF field extracted with Brox algorithm (BF) [34], while on the right the ones extracted with Flownet [29]. While the BF fields look more crisp, they require parameter tuning, while the Flownet version is non-parametric at test time.



(a) Standard and blurred image

(b) Standard and blurred OF field

Fig. 4: Examples of OF fields obtained applying gaussian blur to image sequences. (a) The image and its blurred variant is shown, with blur radius 10. (b) The corresponding OF fields. Note the huge change in OF distribution.

In both images there are evident differences both in hue and saturation, meaning that both modulus and phase of the OF vectors change.

V. Experimental Results

A. Data and Experiments set-up

We perform experiments on two different datasets, the KITTI Visual Odometry benchmark [30] and the Malaga 2013 dataset [31]. Both datasets are taken from cars that travel in city suburbs and countryside, however the illumination conditions and camera setups are different. For the KITTI dataset we used the sequences 00 to 07 for training and the 08, 09 and 10 for test, as is common practice. The images are all around 1240×350, and we resize them to 300×94. The frame rate is 10Hz. For the Malaga dataset we use the sequences 02, 03 and 09 as test set, and the 01, 04, 06, 07, 08, 10 and 11 as training set. In this case the images are 1024×768 that we resize to 224×170. The frame rate is 20Hz. For the Malaga dataset there is no high precision GPS ground truth, so we use the ORBSLAM2 stereo VO [14] as a Ground truth, since its performances, comprising bundle adjustment and loop closing, are much higher than any monocular method.

The networks are implemented in Keras/Tensorflow and trained using an Nvidia Titan Xp. Training of the ST-VO variant takes $6h$, while LS-VO $27h$. The ST-VO memory occupancy is on average 460MB, while LS-VO requires 600MB. At test time, computing Flownet and BF features takes on average 12.5ms and 1 ms per sample, while the prediction requires, on average, $2 - 3$ms for both ST-VO and LS-VO. The total time, when considering Flownet features, amounts to 14.5ms for ST-VO and 15.5ms for LS-VO. Hence, we can observe that the increased complexity does not affect much computational performance at test time.

For all the experiments described in the following Section, we tested the LS-VO architecture and the ST-VO baseline. Furthermore, on all KITTI experiments we tested with both Flownet and BF features. While the contribution of this work relates mainly on showing the increased robustness of the proposed method with respect to learned architectures, we also sampled the performances of SotA geometrical methods, namely VISO2-M [35] and ORBSLAM2-M [14] in order to have a general baseline.

B. Experiments

As mentioned in Section IV-C, on both datasets we perform three kinds of experiments, of increasing difficulty. We observe that the original sequences show some variability in speed, since the car travels in both datasets at speeds of up to 60Km/h, but the distribution of OF field is still limited. This implies that the possible combinations of linear and rotational speeds are limited. We extend the variability of OF field distribution performing some data augmentation. Firstly, we sub-sample the sequences by 2 and 3 times, to generate virtual sequences that have OF vectors with very different intensity. In Figure 3, an example of the different dynamics is shown. In both KITTI and Malaga datasets we indicate the standard sequences by the $d1$ subscript, and the sequences sub-sampled by 2 and 3 times by $d2$ and $d3$, respectively. In addition to this, we generate blurred versions of the $d2$ test sequences, with gaussian blur, as in [1]. Then we perform three kinds of experiment and compare the results. The first is a standard training and test on $d1$ sequences. This kind of test explores the generalization properties on appearance changes alone. In the second kind of experiment we train all the networks on the sequences $d1$ and $d3$ and test on $d2$. This helps us to understand how the networks perform when both appearance and OF dynamics change. The third experiment is training on $d1$ and $d3$ sequences, and testing on the on the blurred versions of the $d2$ test set (Figure 4).

The proposed architecture is end-to-end, since it computes the OF field through a Flownet network. However, as a baseline, we decided to test the performances of all the architecture on a standard geometrical OF input, computed as in [34], and indicated as BF in the following.

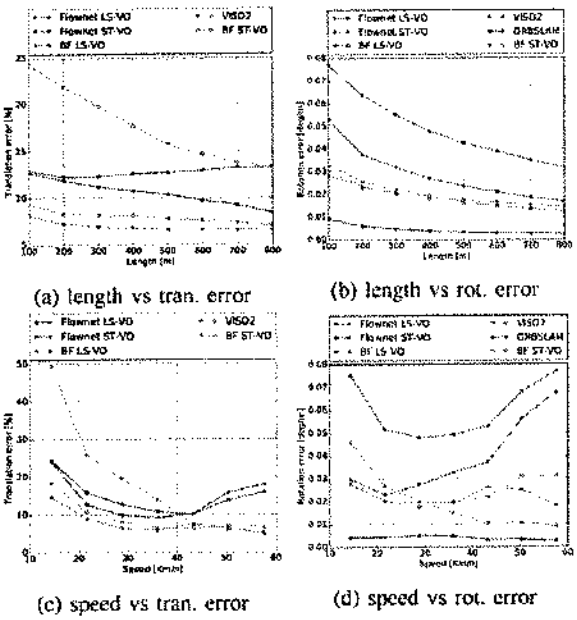
In addition, we train the BF version on the RGB representation of OF, since from our experiments performs slightly better than the floating point one.

	VISO2-M [35]		ORBSLAM2-M [14]		ST-VO (Flow)		ST-VO (BF)		LS-VO (Flow)		LS-VO (BF)	
	Trasl.	Rot.	Trasl.	Rot.	Trasl.	Rot.	Trasl.	Rot.	Trasl.	Rot.	Trasl.	Rot.
KITTI *d1*	18.13%	0.0193	62.71%	**0.0058**	12.73%	0.0507	8.06%	0.0205	10.71%	0.0290	**6.98%**	0.0199
KITTI *d2*	19.08%	**0.0090**	fail	fail	12.30%	0.0383	9.43%	0.0360	10.85%	0.0320	7.71%	0.0205
KITTI *d2* + blur	52.54%	0.0688	fail	fail	18.35%	0.0502	16.39%	0.0627	14.47%	0.0375	8.13%	**0.02710**

TABLE II: Performances summary of all methods on the Kitti experiments. The geometrical methods perform better on the angular rate estimation (in deg/m) on both datasets at standard rate, but usually fail on others (loss of tracking). Learned methods are consistent in their behaviour in all cases: even if the general error increases, they never fail to give an output even in the worst conditions tested, and the trajectories are always meaningful.

	VISO2-M [35]		ORBSLAM2-M [14]		ST-VO (Flow)		LS-VO (Flow)	
	Trasl.	Rot.	Trasl.	Rot.	Trasl.	Rot.	Trasl.	Rot.
Malaga *d1*	43.90%	0.0321	86.60%	**0.0156**	23.20%	0.1241	**15.56%**	0.0690
Malaga *d2*	47.37%	0.0530	fail	fail	23.35%	0.1088	21.44%	**0.0472**
Malaga *d2* + blur	fail	fail	fail	fail	25.14%	0.1262	24.06%	**0.0657**

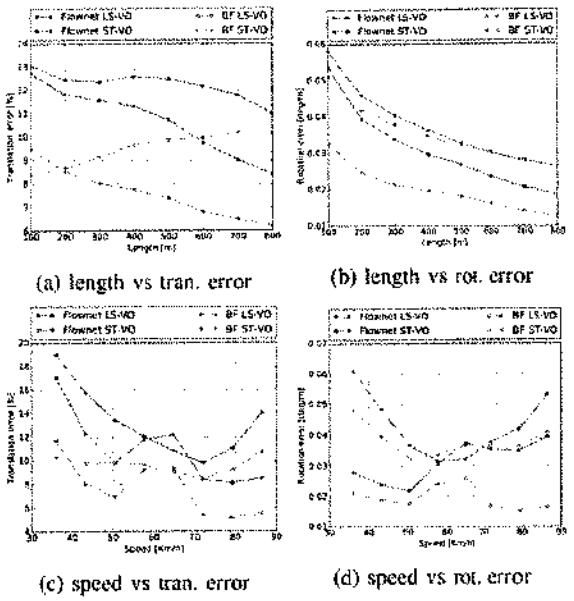
TABLE III: Performances summary of all methods on the Malaga experiments. The same considerations of Table II apply. In this set of experiments we analysed only the end-to-end architecture, for the sake of simplicity.



(a) length vs tran. error (b) length vs rot. error

(c) speed vs tran. error (d) speed vs rot. error

Fig. 5: Comparison between all methods on KITTI dataset, with no sequence sub-sampling. It is evident that the LS-VO network outperforms the ST equivalent, and in the case of the BF OF inputs it is almost always better by a large margin. Geometrical methods outperform learned ones on angular rate. ORBSLAM2-M is not shown in 5a and 5b for axis clarity, since the error is greater than other methods.



(a) length vs tran. error (b) length vs rot. error

(c) speed vs tran. error (d) speed vs rot. error

Fig. 6: Comparison between the four network architectures on KITTI *d2* dataset. Again, the LS-VO architecture outperforms the other, except for speed around 60Km/h.

C. Discussion

The experiments described in Section V-C on both datasets have been evaluated with KITTI devkit [30], and the output plots have been reported in Figures 5, 6, 7, 8 and 9. In all Figures except 7, the upper sub-plots, (a) and (b), represent the translational and rotational errors averaged on sub-sequences of length 100m up to 800m. The lower plots represent the same errors, but averaged on vehicle speed (Km/h). The horizontal axis limits for the lower plots, in Figures relative to *d2* downsampled experiments are different, since the sub-sampling is seen by the evaluation software as an increase in vehicle speed. In Table II and III the total average translational and rotational errors for all the experiments are reported. Figure 5 summarises the performances of all methods on KITTI without frame downsampling. From Figures 5a and 5b we observe that the BF-fed architectures outperform the Flownet-fed networks by a good margin. This is expected, since BF OF fields have been tuned on the dataset to be usable, while Flownet has not been fine-tuned on KITTI sequences. In addition, the LS-VO networks perform almost always better than, or on a par with, the corresponding ST networks. When we consider Figures 5c and 5d, we observe that the increase in performance from ST to LS-VO appears to be slight, except in the rotational errors for the Flownet architecture. However, the difference between the length errors and the speed errors is coherent if we consider that the errors are averaged. Therefore, the speed values that are less represented in the dataset are probably the ones that are more difficult to estimate, but at the same time their effect on the general trajectory estimation is consequently less important.

The geometrical methods do not work on frame pairs only, but perform local bundle adjustment and eventually scale estimation. Even if the comparison is not completely fair with respect to learned methods, it is informative nonetheless.

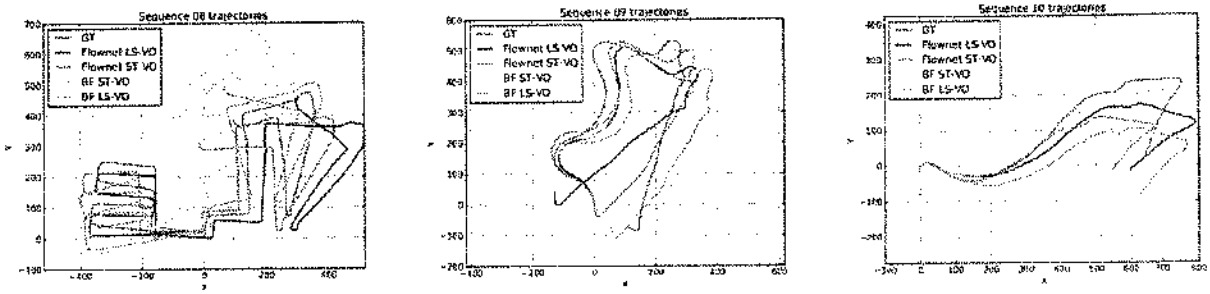
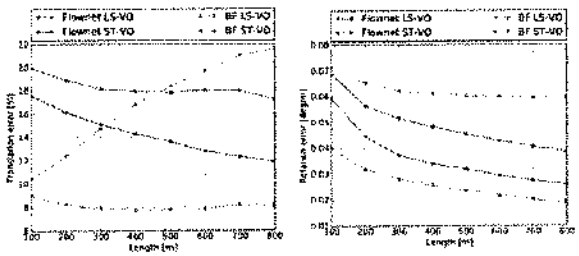
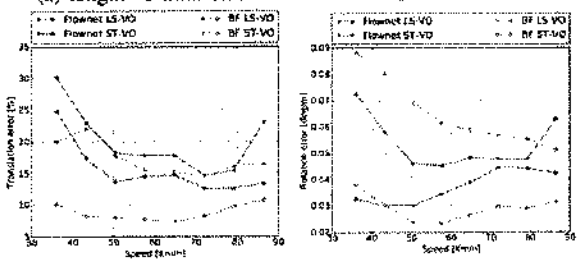


Fig. 7: KITTI *d*2 trajectories: Trajectories computed on the sub-sampled sequences for all architectures (*d*2 - 5Hz).

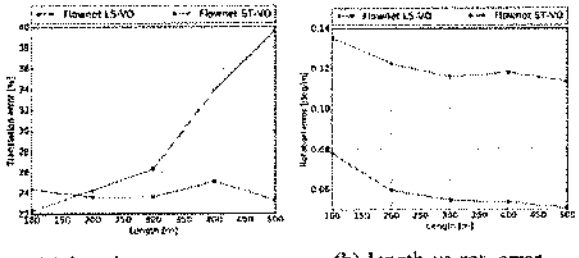


(a) length vs tran. error (b) length vs rot. error

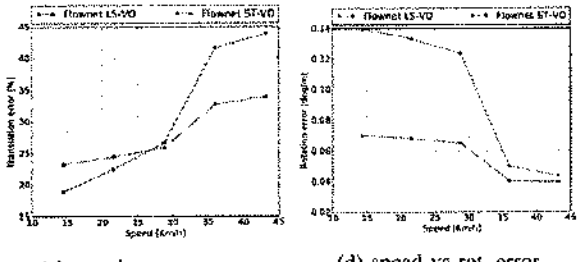


(c) speed vs tran. error (d) speed vs rot. error

Fig. 8: Performances of the four architectures on blurred KITTI *d*2 sequences. The difference in performances between the ST and LS-VO networks is huge. VISO2-M has been omitted, for axis scale reasons.



(a) length vs tran. error (b) length vs rot. error



(c) speed vs tran. error (d) speed vs rot. error

Fig. 9: Performances of the end-to-end architecture on blurred Malaga *d*2 sequences. The lack of samples at high speed make the LS-VO network slightly overfit those samples, as shown in 9c, but in all other respects the behaviour is similar to Figure 8.

In particular we observe (see Figure 5) that the geometrical methods are able to achieve top performances on angular estimation, because they work on full-resolution images and because there is no scale error on angular rate. On the contrary, on average, they perform sensibly worse than learned methods for translational errors. This is also expected, since geometrical methods lack in scale estimation, while learned methods are able to infer scale from appearance. Similar results are obtained for the Malaga dataset. The complete set of experiments is available online [36].

When we consider the second type of experiment, we expect that the general performances of all the architectures and methods should decrease, since the task is more challenging. At the same time, we are interested in probing the robustness and generalization properties of the LS-VO architectures over the ST ones. Figure 6 shows the KITTI results. From 6a and 6b we notice that, while all the average errors for each length increase with respect to the previous experiments, they increase much more for the two ST variants. If we consider the errors depicted in Figures 6c and 6d, we observe that the LS-VO networks perform better than the ST ones, except on speed around 60Km/h, where they are on par. This is understandable, since the networks have been trained on *d*1 and *d*3, that correspond to very low and very high speeds, so the OF in between them are the less represented in the training set. However, the most important consideration here is that the LS-VO architectures show more robustness to domain shifts. The plots of the performances on Malaga can be found online [36], and the same considerations of the previous one apply.

The last experiment is on the downsampled and blurred image. On these datasets both VISO2-M and ORBSLAM2-M fail to give any trajectory, due to the lack of keypoints, while Learned methods always give reasonable results. The results are shown in Figure 8 and 9 for the KITTI and the Malaga dataset, respectively. In both KITTI and Malaga experiments we observe a huge improvement in performances of LS-VO over ST-VO. Due to the difference in sample variety in Malaga with respect to KITTI, we observe overfitting of the more complex network (LS-VO) over the less represented linear speeds (above 30Kmh). This experiments demonstrate that the LS-VO architecture is particularly apt to help end-to-end networks in extracting a robust OF representation. This is an important result, since this architecture can be easily included in other end-to-end approaches, increasing the estimation performances by a good margin, but without significantly increasing the number of parameters for the estimation task, making it more

robust to overfitting, as mentioned in Section IV-B.

VI. Conclusions

This work presented LS-VO, a novel network architecture for estimating monocular camera Ego-Motion. The architecture is composed by two branches that jointly learn a latent space representation of the input OF field, and the camera motion estimate. The joint training allows for the learning of OF features that take into account the underlying structure of a lower dimensional OF manifold. The proposed architecture has been tested on the KITTI and Malaga datasets, with challenging alterations, in order to test the robustness to domain variability in both appearance and OF dynamic range. Compared to the data-driven architectures, LS-VO network outperformed the single branch network on most benchmarks, and in the others performed at the same level. Compared to geometrical methods, the learned methods show outstanding robustness to non-ideal conditions and reasonable performances, given that they work only on a frame to frame estimation and on smaller input images. The new architecture is lean and easy to train and shows good generalization performances. The results provided here are promising and encourage further exploration of OF field latent space learning for the purpose of estimating camera Ego-Motion. All the code, datasets and trained models are made available online [36].

References

[1] G. Costante, M. Mancini, P. Valigi, and T. A. Ciarfuglia, "Exploring Representation Learning with CNNs for Frame-to-Frame Ego-Motion Estimation," *Robotics and Automation Letters, IEEE*, vol. 1, no. 1, pp. 18–25, 2016.

[2] R. Gomez-Ojeda, Z. Zhang, J. Gonzalez-Jimenez, and D. Scaramuzza, "Learning-based image enhancement for visual odometry in challenging hdr environments," *arXiv preprint arXiv:1707.01274*, 2017.

[3] T. Tommasi, N. Patricia, B. Caputo, and T. Tuytelaars, "A deeper look at dataset bias," in *Pattern Recognition: 37th German Conference, GCPR 2015, Aachen, Germany, October 7-10, 2015, Proceedings*. Springer International Publishing, 2015, pp. 504–516.

[4] B. Ummenhofer, H. Zhou, J. Uhrig, N. Mayer, E. Ilg, A. Dosovitskiy, and T. Brox, "DeMoN: Depth and Motion Network for Learning Monocular Stereo," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[5] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, "Unsupervised learning of depth and ego-motion from video," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[6] S. Vijayanarasimhan, S. Ricco, C. Schmid, R. Sukthankar, and K. Fragkiadaki, "SfM-Net: Learning of structure and motion from video," *arXiv preprint arXiv:1704.07804*, 2017.

[7] R. J. W. Roberts, "Optical flow templates for mobile robot environment understanding," Ph.D. dissertation, GIT, 2014.

[8] R. Roberts, H. Nguyen, N. Krishnamurthi, and T. R. Balch, "Memory-based learning for visual odometry," in *2008 IEEE International Conference on Robotics and Automation (ICRA)*, 2008.

[9] J. Wulff and M. J. Black, "Efficient sparse-to-dense optical flow estimation using a learned basis and layers," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[10] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-scale direct monocular SLAM," in *European Conference on Computer Vision (ECCV)*. Springer, 2014, pp. 834–849.

[11] J. Engel, J. Stückler, and D. Cremers, "Large-scale direct SLAM with stereo cameras," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 1935–1942.

[12] C. Forster, M. Pizzoli, and D. Scaramuzza, "SVO: Fast semi-direct monocular visual odometry," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 15–22.

[13] C. Forster, Z. Zhang, M. Gassner, M. Werlberger, and D. Scaramuzza, "SVO: Semidirect visual odometry for monocular and multicamera systems," *IEEE Transactions on Robotics*, vol. 33, no. 2, pp. 249–265, 2017.

[14] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM: a versatile and accurate monocular slam system," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[15] R. Roberts, C. Potthast, and F. Dellaert, "Learning general optical flow subspaces for egomotion estimation and detection of motion anomalies," in *2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[16] V. Guizilini and F. Ramos, "Visual odometry learning for unmanned aerial vehicles," in *2011 IEEE International Conference on Robotics and Automation (ICRA)*, 2011, pp. 6213–6220.

[17] ——, "Semi-parametric models for visual odometry," in *2012 IEEE International Conference on Robotics and Automation (ICRA)*, 2012.

[18] T. A. Ciarfuglia, G. Costante, P. Valigi, and E. Ricci, "Evaluation of non-geometric methods for visual odometry," *Robotics and Autonomous Systems*, vol. 62, no. 12, pp. 1717 – 1730, 2014.

[19] P. Muller and A. Savakis, "Flowdometry: An optical flow and deep learning based approach to visual odometry," in *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2017.

[20] R. Clark, S. Wang, H. Wen, A. Markham, and N. Trigoni, "VINet: Visual-inertial odometry as a sequence-to-sequence learning problem." in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.*, 2017, pp. 3995–4001.

[21] S. Wang, R. Clark, H. Wen, and N. Trigoni, "DeepVO: Towards end-to-end visual odometry with deep recurrent convolutional neural networks," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, May 2017, pp. 2043–2050.

[22] S. Pillai and J. J. Leonard, "Towards visual ego-motion learning in robots," *arXiv preprint arXiv:1705.10279*, 2017.

[23] K. R. Konda and R. Memisevic, "Learning visual odometry with a convolutional network." in *International Conference on Computer Vision Theory and Applications (VISAPP)*, 2015, pp. 486–490.

[24] D. J. Heeger and A. D. Jepson, "Subspace methods for recovering rigid motion I: Algorithm and implementation," *International Journal of Computer Vision*, vol. 7, no. 2, pp. 95–117, 1992.

[25] C. Herdtweck and C. Curio, "Experts of probabilistic flow subspaces for robust monocular odometry in urban areas," in *2012 IEEE Intelligent Vehicles Symposium, 2012 IEEE*, June 2012, pp. 661–667.

[26] M. Ochs, H. Bradler, and R. Mester, "Learning rank reduced interpolation with Principal Component Analysis," in *Intelligent Vehicles Symposium (IV), 2017 IEEE*, June 2017, pp. 1126–1133.

[27] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[28] J.-Y. Zhu, P. Krähenbühl, E. Shechtman, and A. A. Efros, "Generative visual manipulation on the natural image manifold," in *European Conference on Computer Vision (ECCV)*, 2016.

[29] P. Fischer, A. Dosovitskiy, E. Ilg, P. Häusser, C. Hazırbaş, V. Golkov, P. van der Smagt, D. Cremers, and T. Brox, "FlowNet: Learning optical flow with convolutional networks," in *2015 IEEE International Conference on Computer Vision (ICCV)*, Dec 2015, pp. 2758–2766.

[30] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *The International Journal of Robotics Research*, 2013.

[31] J.-L. Blanco, F.-A. Moreno, and J. Gonzalez-Jimenez, "The malaga urban dataset: High-rate stereo and lidars in a realistic urban scenario," *International Journal of Robotics Research*, vol. 33, no. 2, pp. 207–214, 2014.

[32] A. Kendall, M. Grimes, and R. Cipolla, "Posenet: A convolutional network for real-time 6-dof camera relocalization," in *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[33] J. J. Kuffner, "Effective sampling and distance metrics for 3d rigid body path planning," in *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*, vol. 4, April 2004, pp. 3993–3998 Vol.4.

[34] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert, "High accuracy optical flow estimation based on a theory for warping," in *European Conference on Computer Vision (ECCV)*. Springer, 2004, pp. 25–36.

[35] A. Geiger, J. Ziegler, and C. Stiller, "StereoScan: Dense 3d reconstruction in real-time," in *2011 IEEE Intelligent Vehicles Symposium (IV)*, June 2011, pp. 963–968.

[36] Isarlab @ unipg website. [Online]. Available: http://isarunipg.it/

Towards Monocular Digital Elevation Model (DEM) Estimation by Convolutional Neural Networks - Application on Synthetic Aperture Radar Images

Gabriele Costante, University of Perugia, Dep. of Engineering, gabriele.costante@unipg.it, Italy
Thomas A. Ciarfuglia University of Perugia, Dep. of Engineering, thomas.ciarfuglia@unipg.it, Italy
Filippo Biondi, University of L'Aquila, Dep. of Engineering biopippoo@gmail.com, Italy

Abstract

Synthetic aperture radar (SAR) interferometry (InSAR) is performed using repeat-pass geometry. InSAR technique is used to estimate the topographic reconstruction of the earth surface. The main problem of the range-Doppler focusing technique is the nature of the two-dimensional SAR result, affected by the layover indetermination. In order to resolve this problem, a minimum of two sensor acquisitions, separated by a baseline and extended in the cross-slant-range, are needed. However, given its multi-temporal nature, these techniques are vulnerable to atmosphere and Earth environment parameters variation in addition to physical platform instabilities. Furthermore, either two radars are needed or an interferometric cycle is required (that spans from days to weeks), which makes real time DEM estimation impossible. In this work, the authors propose a novel experimental alternative to the InSAR method that uses single-pass acquisitions, using a data driven approach implemented by Deep Neural Networks. We propose a fully Convolutional Neural Network (CNN) Encoder-Decoder architecture, training it on radar images in order to estimate DEMs from single pass image acquisitions. Our results on a set of Sentinel images show that this method is able to learn to some extent the statistical properties of the DEM. The results of this exploratory analysis are encouraging and open the way to the solution of single-pass DEM estimation problem with data driven approaches.

1 Introduction

Interferometric Synthetic Aperture Radar (InSAR) allows topographic reconstruction of a physical environment. The technique is performed designing a spatial single-baseline SAR geometry [12], [3],[22] where the result is a digital elevation model (DEM). However, to solve the phase indetermination with a good altitude accuracy, a minimum of two pass are needed, and this usually implies that we need to wait days, or months between the first and the second pass. We propose a method for estimating the topographic reconstruction with machine learning, implemented by Convolutional Neural Networks (CNNs) in order to estimate a DEM using only one single-look-complex (SLC) SAR image. Before getting inside the description of the novel signal processing technique, a brief analysis of the InSAR history is given. It is necessary to go back in time, until 1980, where Walker *et al.* in [21] admits the feasibility of fine Doppler frequency resolution existing for the range-Doppler SAR image. In this context a high energy scattering point target may move through several range-Doppler resolution cells, producing a smeared trace.
SAR data are represented with a three-dimensional Fourier transform of the object reflectivity density. A full three-dimensional environment reconstruction is processable by an inverse Fourier transform. Munson *et al.* [17] show that spotlight SAR, interpreted as a tomographic reconstruction problem, synthesizes high resolution terrain maps observed along multiple observation angles.

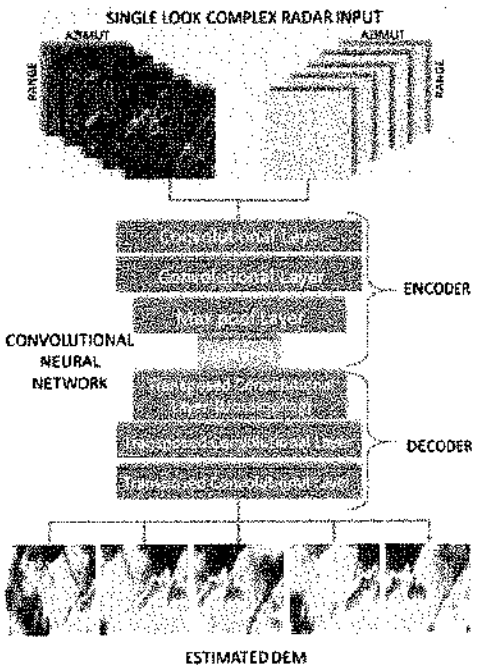


Figure 1: Overview of the proposed DEM estimation approach. The proposed approach does not need multiple radar acquisitions. Instead, it processes a single look complex radar image to estimate the associated elevation model. In order to achieve this, we exploit the convolutional neural network paradigm. In particular, the encoder section extracts highly informative local structures (*i.e.*, features) from the input radar image. Afterwards, the decoder section decodes the features and predicts the DEM image.

Jakowats *et al.* [10] extend the work of Munson *et al.* to a new three-dimensional formulation, making the simplifying assumption that the SAR range-Doppler image is two-dimensional. Unfortunately, this assumption implies the generation of the *layover effect* and, in order to explore target detection in the cross-slant range, multiple observations have to be performed. In [2] the author gave a theoretical explanation of the frequency diversity in SAR-Tomography. A very good introduction to InSAR is given in [15]. The work gives detailed information for combining complex SAR images recorded by antennas positioned at different locations. Recent years saw a refinement of the InSAR technique, trying to remove the need of using multiple satellite passes. InSAR can also be applied using two sensors mounted on the same platform. This configuration is called single-pass interferometry [15]. However, to obtain a digital elevation model with useful accuracy a minimum baseline is needed. Application of InSAR from spaceborne radar prospective is also given in [16]. In Colasanti *et al.* [4] authors performed a precious study regarding ERS-ENVISAT interferometry despite of their carrier frequency having a shift of 31MHz. In [7] the authors gave demonstration in estimating absolute height using a single staring spotlight SAR image using the information of different azimuth defocusing levels generated by scatterers positioned at different heights. The problem of this technique seems being excessively anchored to the nature of the staring-spotlight acquisition which gives a reduced range-azimuth swath of observation and precious absolute height estimation is possible only for few azimuth intra-chromatic high coherency scatterers. However, all the aforementioned methods require complex models and computations to take into account all the atmosphere, sensor and environment conditions. Up to the authors knowledge, the possibility of computing DEM estimates with a standard SAR sensor and with a single-pass acquisition has not be tackled before. In this work, we propose the use of a different paradigm to solve this problem. Since a lot of SAR images has been collected in the past, we adopt a data driven approach. The work has been inspired by recent work on Monocular Depth Estimation performed in the Robotics and Computer Vision communities [13], [14], [19], [20], [9], [8], [18], [11]. Usually, in the Robotics context, depth estimation from standard camera sensors is done by triangulation of information collected through stereo-rigs, or using multiple passes of the same sensor. Recently, Convolutional Neural Networks (CNNs) models have been proposed to perform a reconstruction of a depth map from a single image acquisition. The problem of learning depth from image appearance has similarities with the task of learning DEMs from radar images. In this work, we propose to use the same reasoning, learning the conditional distribution of digital elevation maps from single-pass interferometric imaging. We show that the proposed model is able to learn to some extent the spatial relationships from the input data, even with a moderate amount of data. This preliminary study already shows promising results for future developments.

2 Methodology

In order to perform DEM estimation from single-pass SAR acquisition, we need to infer the structure of the observed Earth portion by only using a single radar image. We achieve this by devising a deep neural network architecture that learns to predict the DEM by extracting structures and high-level information from the input radar image. The key intuition behind this strategy lies in the exploitation of local image structures to infer the DEM value at a certain location (*i.e.*, image pixel). By using multiple stages of convolutional filters, we are able to extract high-level structures (*i.e.*, features) at different scales. These features are then used by the model to resolve ambiguities and estimate the DEM.
In the remainder of this section, we firstly describe more formally the principles behind our approach. Afterwards, we provide details about the proposed convolutional neural network architecture.

2.1 Estimation Problem Formulation

We want to model a function $\mathbf{f}$ that, given a single radar image $\mathcal{I} \in \mathbb{C}^{n \times m}$ represented in the complex range-azimuth domain, is able to estimate the relative DEM, filtering out radar noise and resolving the layover indetermination. The output of the model is the DEM image $\mathcal{D} \in \mathbb{R}^{n \times m}$, where each entry contain the elevation value at that location. In order to evaluate the contribution of the complex components of the radar image, we give the model as input the absolute value $\mathcal{I}_\rho \in \mathbb{R}^{n \times m} = abs(\mathcal{I})$ and the phase $\mathcal{I}_\phi \in \mathbb{R}^{n \times m} = phase(\mathcal{I})$ of the complex image $\mathcal{I}$. Thus, our function is defined as $\mathbf{f} : \mathcal{I}_\rho, \mathcal{I}_\phi \rightarrow \mathcal{D}$.
For the network structure we exploit the *encoder-decoder* paradigm, similar to [1, 6, 13, 14]. This kind of architecture is composed by two main blocks, each one composed by a number of convolutional layers, as shown in Figure 1. The encoder part computes the spatial features and at the same time reduces the image representation size layer after layer, in order to find an encoded representation of the image; the decoder part takes this encoded representation ad decompresses it, with upsamplings and convolutions, to finally reconstruct the original image. The loss that is minimized is the DEM reconstruction error, that is propagated through the decoder and encoder layers. In this way, the network is able to learn a lower dimensional representation (an embedding) of the input radar images, removing noise and increasing the generalization properties for further processing. We propose to use a variation of Encoder-Decoder architecture where the input and output are not the same. In our case the inputs are radar images and the outputs are the DEM reprojected in the radar coordinates (slant-range versus azimuth).
The architecture we propose is a fully convolutional deep network, that is able to handle generic inputs. Fur-

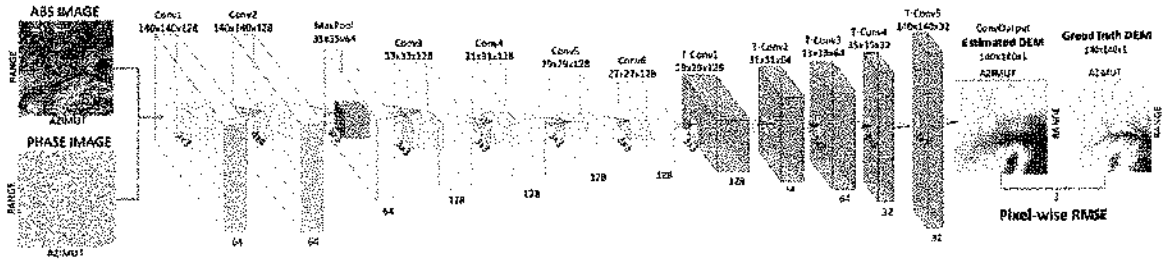


Figure 2: Overview of the proposed Convolutional Neural Network DEM estimator. The encoder section processes the input radar image (*i.e.*, the absolute and the phase images) and extracts features at different scales to detect informative local structures. The decoder sections decodes the features to estimate the associated DEM image.

	Layer name	Kernel size	Stride	Padding	output size	activation
Input	-	-	-	-	$(140, 140, 2)$	
Encoder Section	Conv1	$3 \times 3 \times 64$	1×1	same	$(140, 140, 64)$	ReLU
	Conv2	$5 \times 5 \times 64$	1×1	same	$(140, 140, 64)$	ReLU
	MaxPool	$4 \times 4 \times 64$	4×4	-	$(35, 35, 64)$	-
	Conv3	$3 \times 3 \times 128$	1×1	valid	$(33, 33, 128)$	ReLU
	Conv4	$3 \times 3 \times 128$	1×1	valid	$(31, 31, 128)$	ReLU
	Conv5	$3 \times 3 \times 128$	1×1	valid	$(29, 29, 128)$	ReLU
	Conv6	$3 \times 3 \times 128$	1×1	valid	$(27, 27, 128)$	Linear
Decoder Section	T-Conv1	$3 \times 3 \times 128$	1×1	valid	$(29, 29, 128)$	PReLU
	T-Conv2	$3 \times 3 \times 64$	1×1	valid	$(31, 31, 64)$	PReLU
	T-Conv3	$3 \times 3 \times 64$	1×1	valid	$(33, 33, 64)$	PReLU
	T-Conv3	$3 \times 3 \times 32$	1×1	valid	$(35, 35, 32)$	PReLU
	T-Conv4	$3 \times 3 \times 32$	4×4	valid	$(140, 140, 32)$	PReLU
	ConvOutput	$3 \times 3 \times 1$	1×1	same	$(140, 140, 1)$	Linear

Table 1: Details of the network architecture. The network is composed by two sections, namely the encoder and the decoder section. The encoder section has six convolutional filters and a max pooling layer to extract features at different scale levels. The decoder section decodes the features by using transposed convolutions to estimate the DEM image. Padding *same* is used when we need to preserve the dimensions of a layer input. Conversely, padding *valid* indicates that the convolution operation processes only valid patch of the input (*i.e.*, the output dimension is slightly smaller due to border effects).

thermore, fully convolutional architectures preserve the spatial information both in the encoder and the decoder sections, which is crucial to fully exploit local structure information.

The encoder section is composed by a series of convolutional layers, which sequentially apply learned filters on their input to compute the features.

To extract higher level features, the input is downsampled multiple times. To scale inputs, we use max pooling.

The decoder section is composed by a stack of transposed convolutional layers that learn to reconstruct the pixel-wise predictions of the DEM image from the features computed in the encoder section. Differently from the encoder section, instead of using unpooling layers to reverse pooling operations, we take advantage from the transposed convolutional layers to learn an effective upsampling strategy.

The network is detailed in Table 1 and shown in Figure 2. All the convolutional layers in the encoder section have rectified-linear activation functions (ReLU), except for the last one (Conv6) that has a linear activation function.

The decoder section has five 3×3 Transposed Convolutional (T-Conv) layers to decode the feature extracted by the first section of the network. The last T-Conv layer performs an upsampling by striding the convolutional operations by a factor of 4. All the T-Conv have probabilistic rectified-linear unit (PReLU) to allow for negative activations during the decoding phase. Finally, a single channel 3×3 convolutional layer with a linear activation outputs the predicted DEM image.

All the convolutional filters are regularized with L2 penalty to prevent overfitting.

The objective that is minimized during the learning phase is the pixel-wise linear root mean squared error (RMSE) between the estimated and the GT-DEM images:

$$\sqrt{\frac{1}{T}\sum_{i=0}^{T} \|d_i^{gt} - \hat{d}_i\|^2} \tag{1}$$

where T is the number of pixel of the DEM image $d_i^{gt} \in \mathcal{D}_{gt}$ and $\hat{d}_i \in \hat{\mathcal{D}}$.

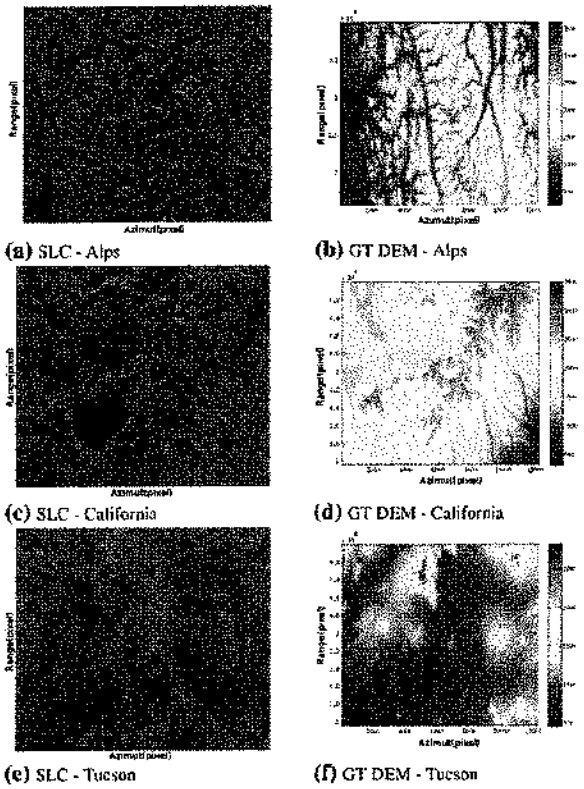
3 Experiments

In this section, we describe the experiments we run to validate our proposed CNN-based DEM estimation approach. In the following, we first describe the experimental setup, providing details about datasets used, preprocessing procedures and details about CNN training.

Afterwards, we discuss the results and draw conclusions.

3.1 Datasets

We test out approach in three different datasets, namely the Alps, the California and the Tucson datasets. The SLC image and the associated GT DEM are depicted in Figure 3(a)-3(b), 3(c)-3(d) and 3(e)-3(f), respectively.



(a) SLC - Alps **(b)** GT DEM - Alps

(c) SLC - California **(d)** GT DEM - California

(e) SLC - Tucson **(f)** GT DEM - Tucson

Figure 3: The datasets used for validating the proposed approach. The first row refers to the Alps dataset, the second to the California dataset and the third to the Tucson dataset. The first column depicts the SLC images, while the second one shows the associated GT DEM.

These datasets are taken from the Sentinel European Space Agency satellite mission. In particular, we use three different acquisitions observing the Alps (Italy), California (USA) and the city of Tucson (USA). The datasets are Single Look Complex (SLC) and vertical-vertical (VV) polarized. Each acquisition is composed by an SLC image with the associated DEM (GT-DEM) computed with standard InSAR techniques. The SLC images are provided as a big complex matrix (typically 12000x20000 entries), while the GT-DEM is a real valued matrix with the same size of its corresponding SLC image.

To learn the CNN model, we generate the training and test samples by sliding a 4000x4000 window on the SLC/GT-DEM pair. The window has a step of 100 pixels with respect to both row and column directions. The size of the window is chosen so that each sample contains enough local structure information to allow the CNN to properly estimate the DEM image. Each sample is downsampled to 140x140 pixels to make the learning task tractable. Depending on the size of the input matrices, we generate up to 22000 samples for each dataset (the exact sample number is discussed in the following sections). The train-test split is generated by randomly selecting the 65% of samples for training and the 35% for testing.

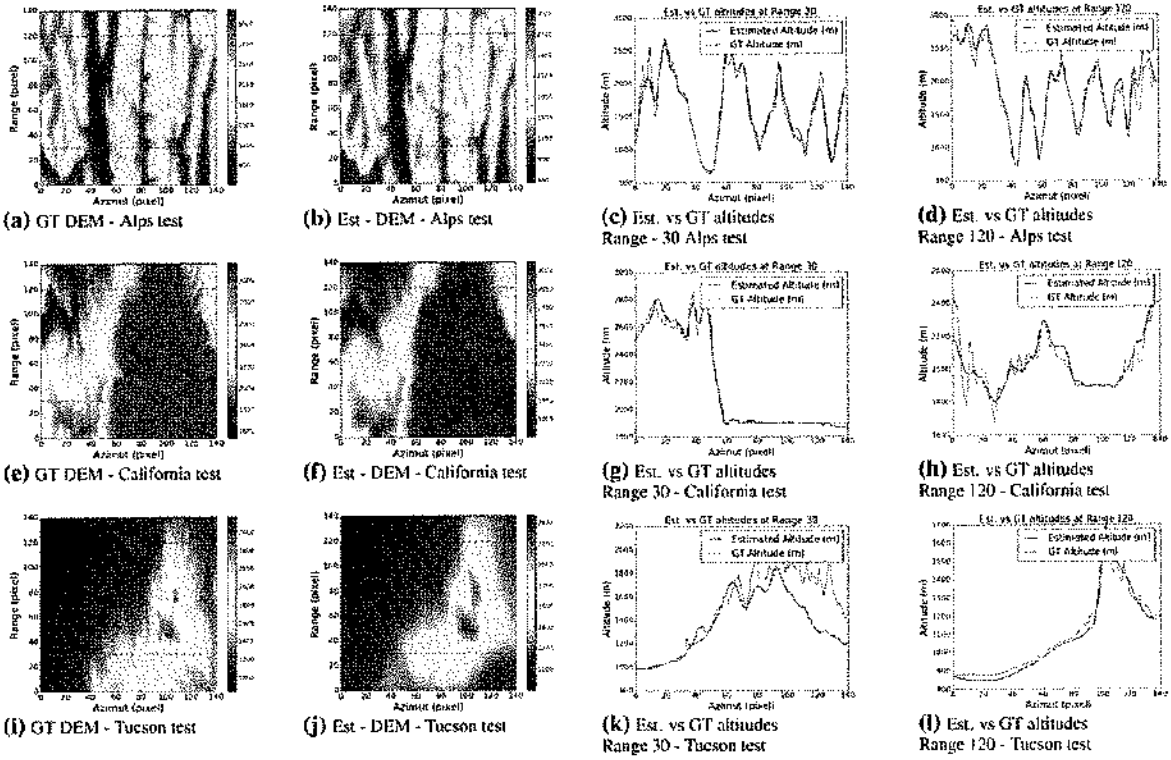
3.2 Training details

The CNN network is trained by using the Adam Optimizer, setting the learning rate $\alpha = 0.001$, the exponential decay rates for the moment estimates $\beta_1 = 0.9$ and $\beta_2 = 0.999$, and $\epsilon = 10^{-08}$. All the L2 regularizer values of the convolutional layer are set to 0.01. The batch size is set to 128 for all the experiments and the training set is randomly shuffled at the end of each epoch. Each model is trained for 500 epochs, which takes approximately four hours with a desktop workstation equipped with a Titan Xp GPU. Once the model is learnt, the predictions run very fast at test time: the computation of the DEM image associated to a 4000x4000 SLC subwindow takes 0.022ms, *i.e.* it runs at approximately 450 Hz.

3.3 Discussion

Figure 4 shows examples of the real DEMs and the estimated ones for each datasets. In addition, the elevation profiles are plotted for two sample range (in pixel), in order to better show the estimation properties of the network. Alps dataset is composed by 11031 train images and 5818 test images, and the average RMSE on all test images is 105.28m. California is composed by 8693 train images and 4755 test images. The average RMSE in this case is 74.46m. Finally, the Tucson dataset has 8846 train and 4849 test images. In this test, the average RMSE error is 43.45m.

It is possible to see qualitatively that the Network learned the altitude statistics, giving a result that closely resembles the ground truth. The main difference is a smoothing effect that the network estimate has in comparison with the original. This is more evident if we compare the GT and predicted profiles for fixed range values (in pixel with respect to the image coordinates) of 30 and 120 pixels, respectively. This is shown in Figures 4(c)-4(d), 4(g)-4(h) and 4(k)-4(l) for the Alps, the California and the Tucson dataset, respectively. From the profiles it is even more apparent that the network is able to output a digital elevation model for the input images that closely resembles the original. The general trends of the GT DEM are closely followed and the main differences between the GT and prediction are due to the smoothing effect on crest ripples, since that for the network these ripples in the ground truth are like a high frequency signal (noise) superimposed to the general elevation model. To better quantify the performances of the network, we quantized the range of elevations in the



(a) GT DEM - Alps test (b) Est - DEM - Alps test (c) Est. vs GT altitudes Range - 30 Alps test (d) Est. vs GT altitudes Range 120 - Alps test

(e) GT DEM - California test (f) Est - DEM - California test (g) Est. vs GT altitudes Range 30 - California test (h) Est. vs GT altitudes Range 120 - California test

(i) GT DEM - Tucson test (j) Est - DEM - Tucson test (k) Est. vs GT altitudes Range 30 - Tucson test (l) Est. vs GT altitudes Range 120 - Tucson test

Figure 4: Comparison between the estimated and the ground truth DEM images. The first row refers to the experiment on the Alps dataset, while the second and the third show the results for the California and the Tucson tests, respectively. The first column depicts the GT DEM of a sample image from the three datasets, while the second column shows the relative estimated DEM. The third and the fourth columns compare the estimated altitude profiles with the ground truth ones at fixed range values.

datasets and computed the average error for each bin, in order to analyse the error distribution given the GT elevation. The resulting plot is shown in Figures 5(a), 5(b) and 5(c) for the Alps, the California and the Tucson datasets, respectively. The three plots, together with the ones in Figure 4 and in consideration of the average RMSE on the test sets show that the estimation network performances degrades when the terrain is mountainous, while are close to the real DEM for slow varying terrain features. This is expected, since the altitude information is not really included in a single pass radar image, so the Network has to extract it from context level information. We hypothesize that, increasing the amount of data given to the network, is possible to further reduce the errors on the high frequency ripples. Furthermore, devising more complex architectures should also help to better model the variabilities of high crests.

4 Conclusions

In this paper, we have proposed an novel method able to estimate DEMs using single SAR images instead interferometric couples. The proposed method uses a data driven approach, implemented through an *Encoder-Decoder* CNNs architecture, and is able to potentially solve the layover indetermination present on the single SLC SAR image using image context information. Our results show that this method is promising, and able to learn useful DEM estimate even with moderate training time and data. For training the CNN a set of Sentinel data has been used.

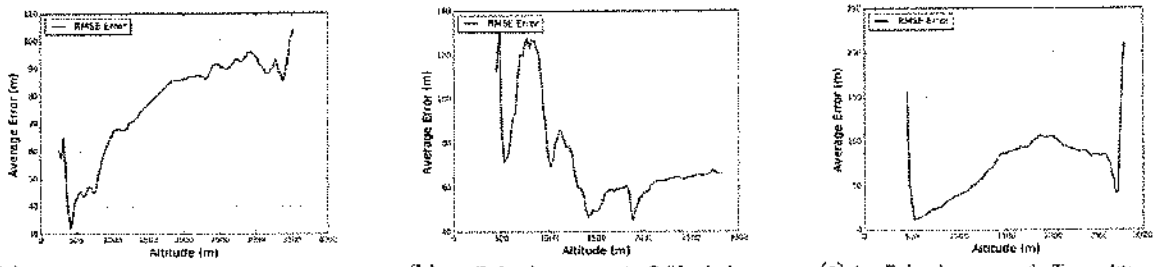
References

[1] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[2] Filippo Biondi. MCA-SAR-Tomography. In *EUSAR 2016: 11th European Conference on Synthetic Aperture Radar, Proceedings of*, pages 1–4. VDE, 2016.

[3] Fabio Bovenga, Dominique Derauw, Fabio Michele Rana, Christian Barbier, Alberto Refice, Nicola Veneziani, and Raffaele Vitulli. Multi-chromatic analysis of sar images for coherent target detection. *Remote Sensing*, 6(9):8822–8843, 2014.

[4] C Colesanti, F De Zan, A Ferretti, C Prati, and F Rocca. Generation of dem with sub-metric vertical accuracy from 30'ers-envisat pairs. In *Proc. FRINGE 2003 Workshop, Frascati, Italy*, pages 1–5, 2003.



(a) Avg Estimation error on the ALPS data (b) Avg Estimation error on the California data (c) Avg Estimation error on the Tucson data

Figure 5: Average estimation error computed on quantized elevations (100 bins) on the ALPS 5(a), the California 5(b) and the Tucson 5(c) data.

[5] Mauro Coltelli. Generation of digital elevation models by using sir-c/x-sar multifrequency two-pass. *IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING*, 34(5), 1995.

[6] A. Dosovitskiy, P. Fischer, E. Ilg, P. Häusser, C. Hazırbaş, V. Golkov, P. v.d. Smagt, D. Cremers, and T. Brox. Flownet: Learning optical flow with convolutional networks. In *IEEE International Conference on Computer Vision (ICCV)*, 2015.

[7] Sergi Duque, Helko Breit, Ulrich Balss, and Alessandro Parizzi. Absolute height estimation using a single terrasar-x staring spotlight acquisition. *IEEE Geoscience and Remote Sensing Letters*, 12(8):1735–1739, 2015.

[8] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2650–2658, 2015.

[9] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Advances in neural information processing systems*, pages 2366–2374, 2014.

[10] Charles V Jakowatz and P Thompson. A new look at spotlight mode synthetic aperture radar as tomography: imaging 3-d targets. *IEEE transactions on image processing*, 4(5):699–703, 1995.

[11] Fayao Liu, Chunhua Shen, Guosheng Lin, and Ian Reid. Learning depth from single monocular images using deep convolutional neural fields. 2015.

[12] Soren N Madsen and Howard A Zebker. Automated absolute phase retrieval in across-track interferometry. 1992.

[13] M. Mancini, G. Costante, P. Valigi, and T. A. Ciarfuglia. Fast robust monocular depth estimation for obstacle detection with fully convolutional networks. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4296–4303, Oct 2016.

[14] M. Mancini, G. Costante, P. Valigi, T. A. Ciarfuglia, J. Delmerico, and D. Scaramuzza. Toward domain independence for learning-based monocular depth estimation. *IEEE Robotics and Automation Letters*, 2(3):1778–1785, July 2017.

[15] Gerard Margarit, Jordi J Mallorqui, and Xavier Fabregas. Single-pass polarimetric sar interferometry for vessel classification. *IEEE transactions on geoscience and remote sensing*, 45(11):3494–3502, 2007.

[16] Joao Moreira, Marcus Schwabisch, Gianfranco Fornaro, Riccardo Lanari, Richard Bamler, Dieter Just, Ulrich Steinbrecher, Helko Breit, Michael Eineder, Giorgio Franceschetti, et al. X-sar interferometry: First results. *IEEE transactions on Geoscience and Remote Sensing*, 33(4):950–956, 1995.

[17] David C Munson, James D O'Brien, and W Kenneth Jenkins. A tomographic formulation of spotlight-mode synthetic aperture radar. *Proceedings of the IEEE*, 71(8):917–925, 1983.

[18] Anirban Roy and Sinisa Todorovic. Monocular depth estimation using neural regression forest. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[19] Ashutosh Saxena, Sung H Chung, and Andrew Y Ng. Learning depth from single monocular images. In *Advances in neural information processing systems*, pages 1161–1168, 2006.

[20] Ashutosh Saxena, Min Sun, and Andrew Y Ng. Make3d: Learning 3d scene structure from a single still image. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(5):824–840, 2009.

[21] Jack L Walker. Range-doppler imaging of rotating objects. *IEEE Transactions on Aerospace and Electronic systems*, (1):23–52, 1980.

[22] Howard A Zebker and Richard M Goldstein. Topographic mapping from interferometric synthetic aperture radar observations. *Journal of Geophysical Research: Solid Earth*, 91(B5):4993–4999, 1986.