

# UNIVERSITA' DEGLI STUDI DI PERUGIA

## DIPARTIMENTO DI INGEGNERIA

Allegati al Verbale n. 1 del 16/02/2017

n.34 allegati suddivisi e numerati per ogni rispettivo punto del seguente Ordine del Giorno:

1. Approvazione verbali
2. Comunicazioni del presidente
3. Convenzioni, contratti e progetti di ricerca
4. Richiesta assegni di ricerca e borse di studio e di ricerca finanziati dal D.I.
5. Approvazione relazioni annuali assegnisti di ricerca
6. Richiesta di contratti di lavoro autonomo
7. Autorizzazioni di spesa
- 7Bis. Attivazione nuovi laboratori didattici
8. Ratifica decreti
9. Varie ed eventuali

### **Riservato ai Professori di Prima e Seconda Fascia, Ricercatori Universitari e Rappresentanti degli Studenti**

10. Approvazione relazione annuale tecnico-scientifica ricercatori a tempo determinato
11. Programmazione didattica
12. Varie ed eventuali

### **Riservato ai Professori di Prima e Seconda Fascia, Ricercatori Universitari a tempo indeterminato**

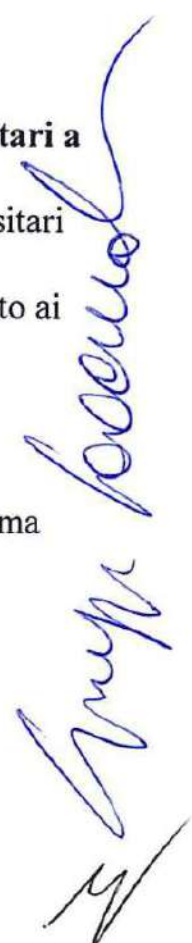
13. Verifica periodica dell'attività didattica e scientifica dei ricercatori universitari – adempimenti previsti dall'art. 33 del D.P.R. 382/1980
14. Richiesta emissione bando per l'assunzione di ricercatori a tempo determinato ai sensi dell'art.24-comma 3 – lettera a) della L.240/2010
15. Varie ed eventuali

### **Riservato ai Professori di Prima e Seconda Fascia**

16. Proposta di chiamata diretta nel ruolo di professore associato, ex art.24-comma 5- della L240/2010
17. Relazioni triennali professori di II fascia
18. Varie ed eventuali

### **Riservato ai Professori di Prima Fascia**

19. Relazioni triennali professori di I fascia
20. Varie ed eventuali



**Relazione Assegno di Ricerca:  
periodo dicembre 2015 - dicembre 2016**



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

**Thomas Alessandro Ciarfuglia**

*Thomas Alessandro Ciarfuglia*

*TC*

## Introduzione

Nell'ambito della ricerca in Robotica e Intelligenza Artificiale i sistemi distribuiti rappresentano un ampio sottoinsieme di sistemi di natura e scopi molto diversi. In particolare si parla di Ubiquitous Robotics quando si ha a che fare con sistemi intelligenti le cui capacità e la cui intelligenza non risulta concentrata in un singolo apparato computazionale, o in cui gli attuatori, o qualunque altro dispositivo in grado di effettuare un'azione o erogare un servizio, non siano fisicamente collegati. Chiaramente questo tipo di definizione abbraccia con naturalezza molti generi di sistemi distribuiti, intersecando l'Internet of Things, ogni genere di Smart Grid e molte altre applicazioni. L'obiettivo di questo progetto di ricerca, affiliato al progetto SEAL (Smart&Safe Energy-Aware Assiste Living), era proprio quello di portare alcune delle tecnologie più avanzate di Intelligenza Artificiale e di Robotica nel settore della domotica al fine di aumentare la sicurezza e la vivibilità degli ambienti in cui l'uomo vive e passa gran parte del proprio tempo. Chiaramente un edificio "intelligente" è sicuramente uno di quei sistemi che possono essere inquadrati come Ubiquitous Robots.

Nell'ambito di questo progetto di ricerca si è portato avanti il lavoro secondo una duplice direttrice. La prima ha portato avanti alcuni aspetti di ricerca applicata, già avviati in precedenza, basati sulla Visione Computazionale, in quanto essa rappresenta uno dei canali di interazione uomo-macchina principali. Questi lavori sono applicati al contesto dei veicoli autonomi per convenienza di argomentazione scientifica (disponibilità di dati, e comunità scientifica più sviluppata), ma la natura degli algoritmi è del tutto generale e può essere applicata anche al contesto dei sistemi domotici (riconoscimento di persone e ambienti noti, ad esempio).

La seconda direttrice ha invece trattato principalmente degli aspetti di sistema e di interconnessione di una rete di sensori e attuatori domotici, sviluppandoli come si sviluppa un sistema robotico distribuito.

L'insieme dei lavori di ricerca sviluppati secondo queste due direttrici rappresenta un passo avanti nella frontiera della ricerca in questo settore. Il passo successivo sarà rappresentato da una migliore integrazione finale di tutti i sottosistemi e algoritmi sviluppati, al fine di ottenere una funzionalità di sistema ancora più complessa.

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. E. Bellocchio, G. Costante, S. Cascianelli, P. Valigi and T. A. Ciarfuglia, "SmartSEAL: A ROS based home automation framework for heterogeneous devices interconnection in smart buildings," *2016 IEEE International Smart Cities Conference (ISC2)*, Trento, 2016, pp. 1-6.
2. G. Costante, M. Mancini, P. Valigi and T. A. Ciarfuglia, "Exploring Representation Learning With CNNs for Frame-to-Frame Ego-Motion Estimation," in *IEEE Robotics and Automation Letters*, vol. 1, no. 1, pp. 18-25, Jan. 2016.
3. M. Mancini, G. Costante, P. Valigi and T. A. Ciarfuglia, "Fast robust monocular depth estimation for Obstacle Detection with fully convolutional networks," *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Daejeon, 2016, pp. 4296-4303.
4. S. Cascianelli, G. Costante, E. Bellocchio, P. Valigi, M. L. Fravolini and T. A. Ciarfuglia, "A robust semi-semantic approach for visual localization in urban environment," *2016 IEEE International Smart Cities Conference (ISC2)*, Trento, 2016, pp. 1-6.





# SmartSEAL: A ROS based Home Automation Framework for Heterogeneous Devices Interconnection in Smart Buildings

Enrico Bellocchio<sup>1</sup>, Gabriele Costante<sup>1</sup>, Silvia Cascianelli<sup>1</sup>, Paolo Valigi<sup>1</sup>, Thomas A. Ciarfuglia<sup>1</sup>

**Abstract**—With this paper we present the SmartSEAL interconnection system developed for the nationally founded SEAL project. SEAL is a research project aimed at developing Home Automation (HA) solutions for building energy management, user customization and improved safety of its inhabitants. One of the main problems of HA systems is the wide range of communication standards that commercial devices use. Usually this forces the designer to choose devices from a few brands, limiting the scope of the system and its capabilities. In this context, SmartSEAL is a framework that aims to integrate heterogeneous devices, such as sensors and actuators from different vendors, providing networking features, protocols and interfaces that are easy to implement and dynamically configurable. The core of our system is a Robotics middleware called Robot Operating System (ROS). We adapted the ROS features to the HA problem, designing the network and protocol architectures for this particular needs. These software infrastructure allows for complex HA functions that could be realized only leveraging the services provided by different devices. The system has been tested in our laboratory and installed in two real environments, Palazzo Fogazzaro in Schio and "Le Case" childhood school in Malo. Since one of the aim of the SEAL project is the personalization of the building environment according to the user needs, and the learning of their patterns of behaviour, in the final part of this work we also describe the ongoing design and experiments to provide a Machine Learning based re-identification module implemented with Convolutional Neural Networks (CNNs). The description of the adaptation module complements the description of the SmartSEAL system and helps in understanding how to develop complex HA services through it.

## I. INTRODUCTION

With many countries aiming to considerably reduce their annual carbon emissions by 2050 [1], energy conservation has become an issue of national importance. Buildings have great impact on human life and global sustainability. They consume a large amount of energy to provide a comfortable, healthy, safe and productive environment for human beings. At the same time, according to [2], Home Automation market was valued at around USD 5.0 billion in 2014 and is expected to reach USD 21.0 billion in 2020, growing at a CAGR (Compounded Average Growth Rate) of around 25% between 2015 and 2020. For these reasons improving building operational performance is of significant importance for energy saving in the construction sector, and Home Au-

This work was supported in part by the M.I.U.R. (Ministero dell'Istruzione dell'Università e della Ricerca) under Grant SCN\_398/SEAL (Program Smart Cities).

<sup>1</sup>The authors are with Department of Engineering, Faculty of Engineering, University of Perugia, via Duranti 93, 06125 Perugia, Italy  
enrico.bellocchio@unipg.it  
thomas.ciarfuglia@unipg.it

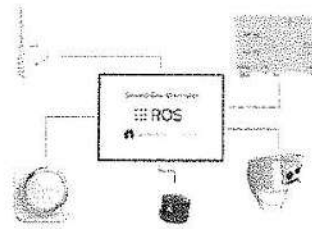


Fig. 1: Overview of SEAL architecture.



Fig. 2: SEAL project logo.

tomation systems can play a key role for decreasing energy consumption while maintaining, or improving, comfort.

Nowadays there is no common standard for HA devices interoperability. This pose an obstacle for the efficient use of every device in an automated home when the devices are produced by different vendors. Some examples of current HA protocols are ZigBee, Z-Wave and KNX [3]. The lack of a common standard poses a serious limit to actual smart building integration. To solve this problem we propose the use of a middle-ware called ROS Robot Operating System that allows the abstraction and the communication of heterogeneous devices. A central server runs the core system, while every device, from single sensors to vendor specific controllers, is a node of the network and is able to communicate with other nodes through a standard TCP/IP network. The communication protocol uses a topic publisher/subscriber architecture that allows for full decentralization of devices, if needed. A standard set of topics and commands have been designed to create the common language for each device to talk to others. On this communication infrastructure more complex services that can act independently may be added. Once the system is set-up, environmental data can be collected and routed with speed and ease. This allows, for example, the collection of temperature, humidity and air-quality profiles over time, build prediction models and adopt strategies according to user habits in the use of the building, thus improving comfort and safety. In particular, as the SEAL project requires HA system adaptation to different users, the use of Machine Learning techniques to model users behaviours and habits is essential. We developed a

ROS-compatible smart-camera and a state-of-the-art People Detection system in order to provide person specific services. In this work we describe both the architecture of the ROS-based HA architecture and protocols, and the HW-SW design of the people detection system.

## II. RELATED WORKS

Smart buildings appliances fall under the broader category of the Internet of Things (IoT) [4]. In [5] a survey about HA systems, illustrating advantages and disadvantages, is given. It illustrates four main barriers that prevent a wider adoption of these systems: high cost of ownership, inflexibility, poor manageability, and difficulty in achieving security.

In literature there are many examples of automation systems installed in smart buildings. In [6] an agent based smart-home called MAVHome is presented. Combining various technologies, such as Artificial Intelligence, Mobile Computing, Robotics and Multimedia Computing, the system is able to perceive home rooms status and act on the environments using actuators. The project combines a wide range of Machine Learning approaches to predict mobility patterns and ambient usage of the inhabitants, with the focus on maximizing the comfort and minimizing the operational costs. In [7] a set of "tape on and forget" sensors that can be installed in home environments is proposed. These devices are then used for Activity Recognition. In [8] a smart building called SMLsystem is described. SMLsystem is a solar-powered house that integrates a whole range of different devices and technologies, such as solar light irradiance sensor, CO<sub>2</sub> and humidity sensors, in order to improve energy consumption. This experimental set-up is used to implement an On-line Learning algorithm, based on a Neural Network architecture, for the production of short-term forecast of indoor temperatures. The work proposed in [9] uses Data Mining approaches to analyze samples collected by a Building Automation System (BAS) present in a commercial building in Hong Kong. Sensors samples, such as indoor and outdoor temperatures, CO<sub>2</sub> concentration and power consumption measurements, were collected for a period of six months. This dataset was then refined with data-mining techniques, that includes clustering approaches and association rules extracted from data. These rules are then used to improve building operational performance. In [10] electric energy consumption samples from three kind of buildings situated at the University of León were collected. In particular from the School and Administrative buildings, from the Research buildings, and from the Special Purpose buildings. These samples were then filtered with a dimensionality reduction algorithm to obtain electricity consumption profiles.

However, the approach proposed in previous works have been developed for a specific set of devices and had a fixed set of services focused on a specific task. In contrast, our SmartSEAL system provides an efficient way to add novel heterogeneous devices (*i.e.*, developed by different vendors) and services on the fly, without requiring a complete system re-design. Our work proposes the use of Robotics

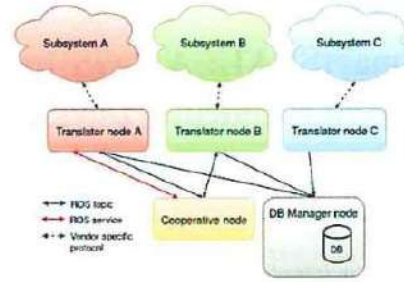


Fig. 3: Seal architecture overview.

technologies to create an adaptive environment that easily integrates different kinds of software and hardware agents. In addition we show the future perspectives of the system, that will feature the use of Machine Learning techniques for user personalization, energy consumption forecasting and energy management optimization, describing the people re-identification module that has been already developed for this project.

## III. CONTRIBUTION AND OUTLINE

The SmartSEAL system aims to interconnect heterogeneous devices and software entities in order to provide complex cooperative services that would not be available in other ways. Devices are connected across a LAN network, and ROS framework is used as an integration layer. To the best of our knowledge, our work is the first example of HA system developed using ROS framework. To summarize, the main contributions of this work are:

- The implementation of a messaging protocol and an abstraction layer for typical sensors and actuators used in HA appliances.
- The deployment and test of the proposed system on two real world buildings in collaboration with local governments.
- The initial design and implementation of a user habits learning module, based on state-of-the-art Machine Learning people identification algorithm.
- The design of custom embedded hardware needed to run the people identification algorithm.

The outline of the paper is the following: In Section IV we give an overview of the ROS middle-ware, while in Section V we describe in details our messaging protocol architecture and the actual deployment and functionalities of the system in the two test sites. We also provide a discussion of the network performance. In Section VI we present the initial design of the CNN-based habits learning module, describing current functionalities and future development. In Section VII we draw the conclusions.

## IV. ROS OUTLINE

In this section we discuss the system architecture. As mentioned, we base our system on the ROS middle-ware [11]. ROS defines a computational environment and network functionalities for software agents, called nodes. These nodes are software processes that range from lower-level hardware



drivers to sophisticated algorithms. The nodes work together in a producer-consumer fashion: the output of a node is written in a named topic that is broadcast over the network, while other nodes can subscribe it if they need this information. The node network and communication is managed by a "master node", called **roscore**, that provides naming and registration services to the rest of the nodes in the ROS environment. Thanks to roscore, the ROS network structure can change during execution, since nodes can be added or removed from the network at run time. As mentioned before, ROS network is based on TCP/IP protocol stack, so it can be either hosted in a local machine, or over a network such as a LAN or a VPN. ROS provides two main mechanism of communication between nodes:

- **Topic:** is the synchronous communication mechanism. This type of communication follow a publisher/subscriber mechanism. A node sends a message by publishing it on a topic, and a node receives message by subscribing it. Each Topic is defined by a simple text file, called *message file*, that describes the format and data structure of the message. In these terms a topic can be considered a one-way communication channel and its name identifies the meaning and the content of the message. There could also be multiple publisher and subscriber nodes for each topic.
- **Service:** is the asynchronous communication mechanism, it uses a client/server communication paradigm. The server node offers a service with a descriptive name and a client needs to send a request message to the server node whenever it wants the service to be provided.

Roscore acts like a DNS server, tracking publishers and subscribers to topics and services, and enabling ROS nodes to locate each other. Once nodes have located they can communicate peer-to-peer. Is possible to use, for the topics and services, standard messages structures provided by ROS libraries, as well as customized structures.

## V. SMARTSEAL MESSAGING ARCHITECTURE

### A. Architecture Overview

In the SmartSEAL system we have a network of devices of different kind and vendors, mainly sensors and actuators, connected through Ethernet LAN or WiFi to a central host machine. This machine is called **SmartSEAL controller** and it is the machine who communicates with the rest of the SEAL network.

In the SmartSEAL system, buildings are divided in conceptual zones. A zone represents a common environment for a group of devices, such as a room, a garden or a courtyard. Every device can be assigned to one or more zones. Sets of devices from the same vendor can be grouped into subsystems, if needed. Within each subsystem, devices communicate with each other through the vendor-specific low-level protocols and policies. The connection to the ROS network is provided by a device capable of acting as a translator to the ROS specific protocol stack. This device is

called concentrator and could either be a dedicated embedded system, or a software node running on a machine of the ROS network.

Following these definitions, the concentrator node uses the ROS communication features in the following way:

- **Topics:** are used for sending sensors measurements.
- **Services:** are used for actuators commands.

Each Topic message definition provides both sensor type, in order to bring measurement information, such as sample value, measurement unit, resolution and accuracy, and device placement information, e.g. device and zone id. Also service definition is customized and depends on command type, e.g. integer or binary command.

Sensors in the system can be divided in two main categories: sensors that give analog measurement and devices that give digital ones. Hence, we have defined two basic structures containing informations about measurement: **AnalogMeasurement** and **DigitalMeasurement**. **AnalogMeasurement** contains information about resolution, accuracy, minimal and maximal range, and measurement unit and value. **DigitalMeasurement** contains similar informations: accuracy, minimal and maximal range, step size and value. We have also defined an header part with informations about devices, *i.e.*, device id, zone id and additional support information. Basic structures and header part are composed obtaining complex message definitions, used for ROS topics:

- **AnalogDevice** and **DigitalDevice**, used for sensors that stream analog or digital measurements.
- **BinaryFlagDevice**, used for sensors with boolean measurements.
- **IntegralMeasurementDevice**, used for devices with integral measurement, such as energy consumption sensors.
- **CameraState**, specifically designed for describing state of robotic cameras.

Actuators command structure has two fields, one for the service request and one for the response. We have defined ROS service structures according to the command type:

- **SetInteger**, used for command containing integer values.
- **SetFloat**, used for commands that involves analog values.
- **SetBool**, used for boolean commands.
- **MoveCamera**, specific for moving robotic cameras.
- **GetStatus**, this command type is implemented by every device and is used for polling last sensors measures. Measure will be sended in sensor topic.

In the following, we show some examples of ROS topic message and ROS service command definitions used in SmartSEAL system:

Listing 1: AnalogMeasurement message definition

```
string measurement_unit
float accuracy
float resolution
float range_min
```

```
float range_max
float value
```

Listing 2: AnalogDevice topic definition

```
Header header
int id_device
AnalogMeasurement measurement
string info
int[] zone
```

Listing 3: SetInteger service definition

```
int id_device
int value

bool done
int error_code
```

For the SmartSEAL system we defined four types of agents:

- **Roscore:** it runs in the SmartSEAL controller and manages the ROS network.
- **Translator nodes:** acts as translator between ROS protocol and one of the vendor-specific protocols, thus providing subsystem topics and services to the ROS network. It runs either on a dedicated embedded system or on the SmartSEAL controller, depending on the specific implementation chosen by each vendor.
- **Cooperative nodes:** use actuators and sensors of different vendors to provide complex functions. Cooperative nodes run mainly on the SmartSEAL controller.
- **Database Manager node:** it is a special node used to log sensor data and states. This node subscribes every sensor topic, collecting measures over time. The Database Manager runs on the SmartSEAL controller.

Cooperative nodes provide complex HA functionalities to specific zones, such as specific heating and lighting condition according to the current user. Since they can be started at runtime, Cooperative nodes are developed to be self-configurable. In order to do this two special topics were defined: **Command Discovery Request (CDRq)** and **Command Discovery Response (CDRp)**. Using this two topics a Cooperative node is able to scan the network, checking for devices able to publish the topics it requires to provide a specific service or feature in a specific zone. For example, a gate control node could ask for the topic provided by a camera that is able to see the gate, and the topic related to the gate status. The combination of the two topics, CDRq and CDRp, allow for the handshaking between the devices providing the information and the Cooperative node that uses this information to implement a complex service.

### B. System Deployment

To test the SmartSEAL framework we conducted extensive tests in three real environments, that represents different use-cases. The first test site is a student facility located in a wing of the Fogazzaro Palace, in the city of Schio, that is



Fig. 4: Palazzo Fogazzaro in Schio (left) and "Case" childhood school in Malo (right).

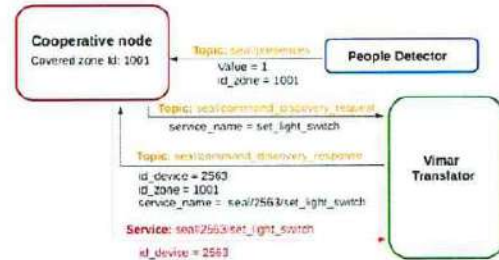


Fig. 5: Cooperative node execution.

commonly used as a study room by students of any grade. The second test building is a kindergarten located in the city of Malo. The last test site is the robotics lab in the Department of Engineering of the University of Perugia. These three sites have very different pattern of use, for example the kindergarten has a very stable pattern, with defined schedules, while the lab has a schedule that depends mostly on the personal schedule and habits of the lab staff, and is very changeable. Figure 6 shows the system deployment pattern, in the first two of these buildings a Videotec robotic camera is installed, together with a smart boiler with energy consumption monitoring system (built by Tecnowatt), automated gate actuators (built by BFT), smart sensors kits (built by Ecam Ricert) and smart light and heating plant (built by Vimar). In addition to this commercial equipment, we installed one of the smart-cameras described in Section VI in each site, together with a computer running the SmartSEAL controller.

Videotec, Tecnowatt and Vimar Translators nodes run on the same machine running the SmartSEAL controller and talk directly with their respective subsystems. BFT subsystem is quite similar, and the HA central device is a dedicated embedded system called Magistro. Ecam Ricert HA central device is represented by a normal computer that runs ROS and its own Translator node. Finally, our smart camera is equipped with an on-board Linux operating system and with ROS, so it also runs its own Translator node.

Figure 5 shows how cooperative service node configuration phase works. Cooperative node interacts with environment lights according to zone id and presence messages value. Presence message indicates person presence in the specified zone. When the execution starts the node does not know which devices are present in the system, it only know the zone id where the functionality have to be provided. The node subscribe the presence topic and command discovery



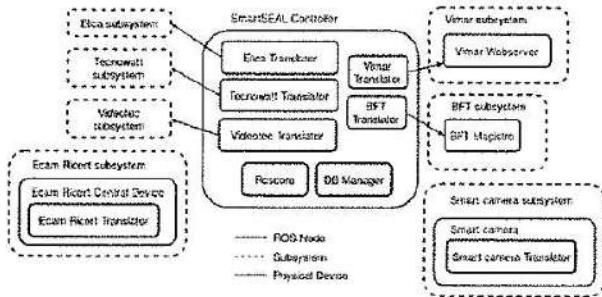


Fig. 6: Deployment scheme.

TABLE I: Ping Performance

Ping type	Average Latency
ICMP Ping	1.13 ms
ROS Ping (XML-RPC)	8.04 ms

TABLE II: Topics Performance

Topic type	Bandwidth (10Hz)	Average Latency
AnalogDevice	600.00 Byte/s	0.021 s
DigitalDevice	560.00 Byte/s	0.020 s
BinaryFlagDevice	370.00 Byte/s	0.019 s
IntegralMeas.Device	720.00 Byte/s	0.023 s
CameraState	1.10 KByte/s	0.026 s
CDRq	200.00 Byte/s	0.018 s
CDRp	320.00 Byte/s	0.020 s

response topic. In this scenario, people detector node in execution in the smart camera publishes presence messages. Cooperative node filters sensor messages according to zone id. To know which light devices are present, cooperative node publish a command discovery request message containing the command type. Every device present on the SEAL network that can accept this command type will reply, publishing a message on command discovery, containing device information such as deployment zone id and device id. Cooperative node filters command discovery response and now can contacts interested zone devices. It is important to notice that this flow of operation is independent of devices vendor.

### C. Network Performance

In this section, we discuss the network performance. Tests are performed on a point-to-point ad-hoc Ethernet network with two hosts. Table I shows the latency comparison of the ICMP and ROS ping packets. We want to measure the overhead due to the headers of the packet introduced by the ROS middle-ware, so we sent the simple ICMP ping packet over the network and we compared the latency with the ROS ping packet, which uses the XML-RPC protocol. In particular, the ROS ping routine consists of a XML coded request sent using the HTTP protocol. Despite of the overhead of 6.91ms, this experiment proves that the ROS framework does not compromise network functionalities.

Table II gives an overview of the bandwidth and the latency for different topic message definitions. Messages on topics are published with a rate of 10 Hz. Latency time is quite similar for the various message definitions with an

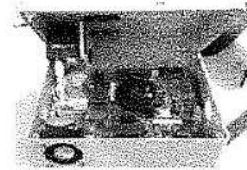


Fig. 8: Smart camera internal view.

average value of 21ms, mainly depending of the network throughput.

## VI. BEHAVIOUR LEARNING WITH CONVOLUTIONAL NEURAL NETWORK

The second aim of the SmartSEAL system is to provide the foundation for a self-configuring environment that is able to learn the behaviours of specific individuals that, over the time, regularly uses the buildings equipped with the system. This Section describes the module that provides the camera-based people re-identification feature that is instrumental for the future development of the behaviour learning feature.

In recent years the Deep Networks has become the tool of choice for many computer vision tasks [12], [13]. Also the core of our re-identification system is a Convolutional Neural Network (CNN), trained to recognize people and other common object present in the building environment. In addition, to run such algorithm a good amount of processing power is required. For this reason we developed a custom smart camera based on a CUDA capable device. The smart camera is showed in figure 8 and is composed by two main part:

- **Nvidia Jetson TK1:** is an embedded prototyping system equipped with Nvidia Kepler GPU and Quad-Core ARM Cortex-A15 CPU, it runs Ubuntu Linux operating system and developing tools include Cuda libraries for GPU optimization, optimized OpenCV libraries and Deep learning frameworks, such as Caffe [14] and Darknet. It also has got a ethernet interface and it supports ROS middle-ware.
- **Matrix Vision BlueFox3 BF3-1012bc Camera:** is a reliable industrial camera, with an USB3 connection, it can support maximum resolution of 1280x960 pixels and a maximum framerate of 40 frames per second.

The main people detecting algorithm is an implementation of the state-of-the-art YOLO object detector [12]. A CNN is composed by a large set of sequential layers that take images from camera video stream and are trained to automatically extract the best set of features for optimal people and object detection, as shown in Figure 7. Convolutional layers are composed of large sets of filters that perform sequential convolution operations on blocks of input data:

$$h_c = W_c * x_c + b_c \quad (1)$$

where  $h_c$  and  $x_c$  are respectively the output and the input matrices, while  $W_c$  and  $b_c$  are the weights and the biases matrices. The weights of these convolutional layer are learned according to the task at hand, through a backpropagation algorithm [15]. In between convolutional layers, often other

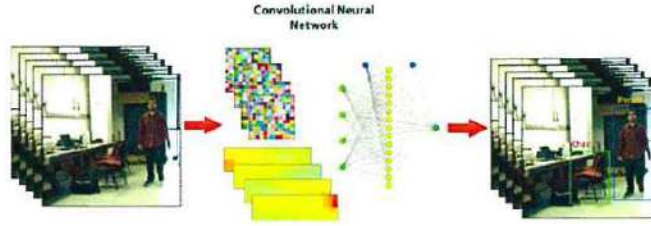


Fig. 7: Overview of detection system based on Convolutional Neural Network.

kind of layers are inserted to reduce the data to prevent overfitting. The maxpool layer is one example of such layers. It performs a down-sampling operation on the input data, simplifying successive computations. Down-sampling is performed extracting the maximum neuron activation  $x_{(i,j)}$ :

$$x_{(i,j)} = \max_{\forall(p,q) \in \Omega_{(i,j)}} x_{(p,q)} \quad (2)$$

where  $\Omega_{(i,j)}$  is the sliding pooling region and  $x_{(p,q)}$  is the neuron activation value. The convolutional and maxpooling layers are trained in order to learn the best filter weights for the detection task, and to complete the algorithm pipeline a fully connected neural network is placed at the end of the cascaded filters. This layer is a standard NN that performs the actual detection task using the feature computed by the CNN layers. In detail, YOLO CNN architecture is composed by twelve layers, where the first 8 layers are 4 of convolutional layers paired with a maxpool layer each. The final four layers are composed by two convolutional layer followed by two fully connected layer. Fully connected layer conduces a matrix multiplication of the input:

$$h_{fc} = W_{fc}x_{fc} + b_{fc} \quad (3)$$

where  $h_{fc}$  and  $x_{fc}$  are respectively the output and the input parameter, and  $W_{fc}$  and  $b_{fc}$  are weights and biases.

The YOLO NN is trained to detect twenty classes of objects, including person, chairs, cats, planes, and others.

In a SmartSEAL system the YOLO detector can be considered as a sophisticated sensor that can detect people presence with good precision, and this information can be used for various HA functionalities, such as temperature and CO2 level forecasting, interaction with lights, windows and radiators, and more. At the current stage of development of the project, the YOLO detector is used to automatically actuate lights and temperature control. The behaviour learning features are under development in these days and will be described in future works.

## VII. CONCLUSIONS AND FUTURE WORKS

In this paper we presented the SmartSEAL Home Automation framework. We discussed the system messaging protocol, the network architecture and the deployment on actual buildings in the cities of Schio and Malo. We demonstrated through experiment that the system allows for the seamless communication of devices from different vendors in order to provide new and complex services. We gave examples of

these services, focusing in particular on the complex CNN-based people detection module that will soon develop in the habits learning module. This module will provide the customization and adaptability required for better energy management and improved user experience.

## REFERENCES

- [1] K. Kern, G. Alber, S. Energy, and C. Policy, "Governing climate change in cities: modes of urban climate governance in multi-level systems," *Competitive Cities and Climate Change*, vol. 171, 2008.
- [2] ZionResearch, "Home automation market - global industry perspective, comprehensive analysis, and forecast, 2014-2020," Dec 2015.
- [3] M. Tariq, Z. Zhou, J. Wu, M. Macuha, and T. Sato, "Smart grid standards for home and building automation," in *Power System Technology (POWERCON), 2012 IEEE International Conference on*, pp. 1-6, Oct 2012.
- [4] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of things (IoT): A vision, architectural elements, and future directions," *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645-1660, 2013.
- [5] A. Brush, B. Lee, R. Mahajan, S. Agarwal, S. Saroiu, and C. Dixon, "Home automation in the wild: challenges and opportunities," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 2115-2124, ACM, 2011.
- [6] S. K. Das, D. J. Cook, A. Battacharya, E. O. Heierman, and T.-Y. Lin, "The role of prediction algorithms in the mavhome smart home architecture," *Wireless Communications, IEEE*, vol. 9, no. 6, pp. 77-84, 2002.
- [7] E. M. Tapia, S. S. Intille, and K. Larson, *Activity recognition in the home using simple and ubiquitous sensors*. Springer, 2004.
- [8] F. Zamora-Martínez, P. Romeu, P. Botella-Rocamora, and J. Pardo, "On-line learning of indoor temperature forecasting models towards energy efficiency," *Energy and Buildings*, vol. 83, pp. 162-172, 2014.
- [9] F. Xiao and C. Fan, "Data mining in building automation system for improving building operational performance," *Energy and buildings*, vol. 75, pp. 109-118, 2014.
- [10] A. Morán, J. J. Fuertes, M. A. Prada, S. Alonso, P. Barrientos, I. Díaz, and M. Domínguez, "Analysis of electricity consumption profiles in public buildings with dimensionality reduction techniques," *Engineering Applications of Artificial Intelligence*, vol. 26, no. 8, pp. 1872-1880, 2013.
- [11] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3, p. 5, 2009.
- [12] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," *arXiv preprint arXiv:1506.02640*, 2015.
- [13] 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.
- [14] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proceedings of the ACM International Conference on Multimedia*, pp. 675-678, ACM, 2014.
- [15] R. Hecht-Nielsen, "Theory of the backpropagation neural network," in *Neural Networks, 1989, IJCNN., International Joint Conference on*, pp. 593-605, IEEE, 1989.



# Exploring Representation Learning With CNNs for Frame-to-Frame Ego-Motion Estimation

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

**Abstract**—Visual ego-motion estimation, or briefly visual odometry (VO), is one of the key building blocks of modern SLAM systems. In the last decade, impressive results have been demonstrated in the context of visual navigation, reaching very high localization performance. However, all ego-motion estimation systems require careful parameter tuning procedures for the specific environment they have to work in. Furthermore, even in ideal scenarios, most state-of-the-art approaches fail to handle image anomalies and imperfections, which results in less robust estimates. VO systems that rely on geometrical approaches extract sparse or dense features and match them to perform frame-to-frame (F2F) motion estimation. However, images contain much more information that can be used to further improve the F2F estimation. To learn new feature representation, a very successful approach is to use deep convolutional neural networks. Inspired by recent advances in deep networks and by previous work on learning methods applied to VO, we explore the use of convolutional neural networks to learn *both* the best visual features and the best estimator for the task of visual ego-motion estimation. With experiments on publicly available datasets, we show that our approach is robust with respect to blur, luminance, and contrast anomalies and outperforms most state-of-the-art approaches even in nominal conditions.

**Index Terms**—Visual Learning, Visual-Based Navigation.

## I. INTRODUCTION

**E**GO-MOTION estimation is a fundamental building block of any robotic system, needed for localization and route planning, and for the more complex task of mapping an unknown environment. When vision comes into play, the task of estimating the ego-motion of cameras is referred to as Visual Odometry (VO). In recent years, impressive results have been shown in the context of monocular visual odometry [1], [2], [3], [4].

Most visual odometry approaches are grounded on the estimate of the camera motion between pairs of consecutive frames. This frame to frame (F2F) estimate is in most cases computed with geometric methods, *i.e.* through the use of projective geometry relations between 3D points of the scene and their projection on the image plane, or by minimizing the gradient of the pixel intensities across consecutive images [5]. The initial F2F estimate is then refined with different strategies, such as

bundle adjustment on a sliding window of previous frames [6], or loop closure detection [7].

However, the initial feature extraction process is critical to the whole estimation process and, because of that, while in structured and controlled environments (*e.g.*, with a large amount of texture and without dynamic objects) these standard approaches provide good results, their performance drops quickly when facing challenging and unpredicted scenarios. These uncertainties can have various causes: (i) illumination changes over time and scenes; (ii) presence of dynamic objects; (iii) different camera calibrations; (iv) low-textured environments, noise and blur. Strengthening the estimation process against these issues requires a twofold action. On one side more informative and robust features are needed, on the other the estimation algorithms are required to better handle noise and unpredicted input anomalies. As far as the estimation aspect goes, new approaches have recently been proposed, that start from the perspective of a statistical learning problem [8], [9], and show many desirable properties. Learning an estimator from data requires good labelled datasets, but when these are available, the learned estimator is robust to illumination changes, noise and blur. From the point of view of what features to use for robust ego-motion estimation, recent advances in Deep Networks applied to representation learning have shown a lot of potential [10]. The core of these approaches in Computer Vision is to use deep Convolutional Neural Networks (CNNs) to learn the best convolutional filters to apply to input image for a given task.

Guided by the previous considerations, in this work we explore a different strategy for performing visual ego-motion estimation. We do not assume any pre-defined procedure to compute the transformation between frames. Instead, following the deep network learning paradigm, we allow the system to autonomously select both the portion of input data that is crucial to achieving robust F2F motion estimation and the strategy for computing it. In particular, inspired by the results achieved with deep architectures, we propose a novel F2F estimation strategy that predicts the camera motion using a CNN (see Figure 1). By learning the CNN, our approach is able to autonomously select the *most important visual cues* and the *best strategy* to compute F2F estimates that are *robust to blur, luminance and contrast anomalies*.

## II. RELATED WORK

To our knowledge, algorithms that compute VO can be divided into different categories, according to the kind of processing used for computing ego-motion: Geometric Methods,

Manuscript received August 29, 2015; accepted November 18, 2015. Date of publication December 4, 2015; date of current version December 28, 2015. This work was supported by the NVIDIA Corporation with the donation of the Tesla K40 GPU. This paper is recommended for publication by Associate Editor B. Caputo and Editor D. Borra upon evaluation of the reviewers' comments.

The authors are with the Department of Engineering, University of Perugia, Perugia 06125, Italy (e-mail: thomas.ciarfuglia@unipg.it; paolo.valigi@unipg.it; gabriele.costante@studenti.unipg.it; michele.mancini1@studenti.unipg.it).

Digital Object Identifier 10.1109/LRA.2015.2505717

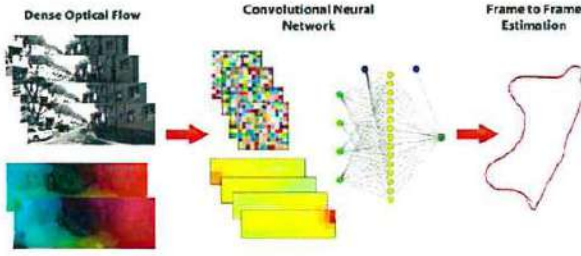


Fig. 1. Overview of the proposed Convolutional Neural Network VO estimator. First, dense optical flow is extracted from consecutive images and then fed into a chain of convolutional filters that extract more visual information. Finally, the new features are used by a fully connected neural network to estimate the F2F camera motion.

that are further divided in Feature-based and Direct Methods, and Learning methods, that learn the estimation function from data.

In the following, the related works for each kind of approach are presented.

#### A. Geometric Methods

1) *Feature-Based Methods*: Feature-based methods rely on the extraction of visual salient points from each image in the stream, tracking them frame after frame until they disappear from view. Typically, the process matches 2D point features across two or more frames, then reconstructs their 3D position using triangulation. Examples of these approaches are [11], [12], [13] for stereo and [14], [15] for mono.

To address the problem of scale estimation a number of solutions were proposed. In [16] the scale is recovered using the extra information from an IMU, while in [1] and, more recently, in [2] an optimization approach using loop closing information is able to recover scale errors. In most cases, to address the problem of scale uncertainty, extra-information is needed in the form of an extra sensor, loop closing data or metric set up information.

2) *Dense Methods*: The main limit of feature based VO estimators is the extraction of the features. Feature-based methods are fast but can easily fail in contexts where feature extraction is difficult, such as low textured or blurred images. Dense methods try to use the whole image instead. The process is similar to dense optical flow extraction [17], but instead of extracting the motion of each pixel, it extracts the underlying camera motion. Clearly, dense methods can achieve better accuracy than feature-based ones, but are more computationally intense, and only recently some have reached real-time operation on common hand-held devices or embedded systems, enabling their use on Micro Aerial Vehicle (MAV) platforms [2], [3], [4].

#### B. Learning Methods

While geometric methods mainly make strong assumptions about what to extract from images and how to use this information to compute motion estimation, learning methods try to infer them from data. The first examples of learning-based VO are [18] and [19], where the authors divide each frame into cells



Fig. 2. Optical flow fields a) feature based (sparse) b) intensity based (dense).

and compute an average optical flow for each block, then they train a K-Nearest Neighbor (KNN) regressor in the first work and an Expectation Maximization (EM) algorithm in the second. In [9], [20] and [8] a similar feature parametrization of optical flow is used together with Coupled Gaussian Processes (CGP) as a regression algorithm.

The common aspect of these previous works is the use of quantized sparse optical flow, such as Histogram of Optical Flow (HOF). As far as we know, no previous work on this problem has proposed learning the representation from data. Since the recent rise of deep architectures has shown that it is possible to give to the learning algorithm the plain input and let it learn the correct representation for the task with impressive results [21], [22], [23], we propose to apply the same techniques to the feature extraction part of ego-motion estimation problem. In order to do so there is the need to stack many stages of successive filter banks [24], whose coefficients are learned using unsupervised or supervised methods. In this work, we use Convolutional Neural Networks (CNN), applying them to dense optical flow, to learn new visual features at the same time with a non-parametric motion estimator. The proposed CNN architectures outperform the state-of-the-art F2F estimation approaches and guarantee robustness with respect to image anomalies (*e.g.*, blur, contrast and luminance).

#### C. Contribution and Overview

Our contribution summarizes to this points:

- A) We explore feature selection for ego-motion estimation using different CNN architectures. The CNN architectures are used to extract new input features starting from dense optical flow (Figure 2). Three different architectures are proposed: two of them investigate the influence of global and local optical flow fields with respect to the ego-motion estimation (*i.e.*, considering both the full flow image and its different sub-blocks); the last one combine the advantages of the others in a parallel CNN that exploits both global and local information.
- B) We show that the presented learned estimators are able to estimate motion outperforming other SotA geometrical and learned methods. In addition the proposed methods are able to use global information to extract camera motion and scale information, while dealing with noise in input.
- C) Finally, we show the performances of the presented method in difficult scenarios, using images with very different contrast and blur parameters, to show the robustness of the new features extracted by the CNN.

It is important to note that, in this work, we propose to improve specifically F2F estimation performance. Hence, to



provide a precise and in-depth discussion about this fundamental block of any VO approach, we do not consider bundle adjustment or, in general, procedures that refine the motion estimates. However, our approach can be easily embedded into a full key-frame based VO system to reach even better performances.

This work proceeds as follows: Section III describes the foundations of CNNs and the proposed networks architecture; Section IV presents the experimental set-up, the dataset and the performance parameters used; finally, Section V draws the conclusions and the path of future work.

### III. NETWORK STRUCTURE

In the following, we first describe how we extract dense optical flow information and then we discuss the CNN network architectures used to learn the F2F estimation models.

#### A. Dense Optical Flow

The input to our network is a *dense* optical flow (OF) extracted with Brox algorithm [17]. The Brox strategy computes the optical flow between two images at time  $t$  and  $t + 1$  using a variational formulation that penalizes the total variation of the flow field by minimizing the following energy function:

$$V(u, v) = V_d + \alpha V_s \quad (1)$$

with

$$V_d(u, v) = \int_{\Omega} \Phi(|I(\mathbf{x} + \mathbf{w}) - I(\mathbf{x})|^2 + \gamma |\nabla I(\mathbf{x} + \mathbf{w}) - \nabla I(\mathbf{x})|^2) dx$$

$$V_s(u, v) = \int_{\Omega} \Phi(|\nabla_3 u|^2 + |\nabla_3 v|^2) dx$$

and where  $I: \Omega \subset \mathbb{R}^3 \rightarrow \mathbb{R}$  denotes a rectangular image sequence,  $\mathbf{w} := (u, v, 1)^T$ ,  $\Phi$  is a concave function (as described by [17]) and  $\alpha$  and  $\gamma$  are weighting parameters.

The computed flow field is then quantized in the common RGB encoding, as shown in Figure 2(b), so the input data is a 3 channel, 8-bit depth image.

#### B. Proposed Network Architecture and Training Procedure

In object recognition and people detection tasks the input images are smaller than the ones typically used in VO. Simply applying one of the already proposed architectures is not straightforward. Down-sampling the image could discard important information for motion estimate. For this reason, we tested three different architectures and compared their performances:

- 1) **CNN-1b VO**: As a basic exploratory approach we train a deep network on the entire OF image after down-sampling it 8 times with average pooling to reach a dimension of  $155 \times 48$ .

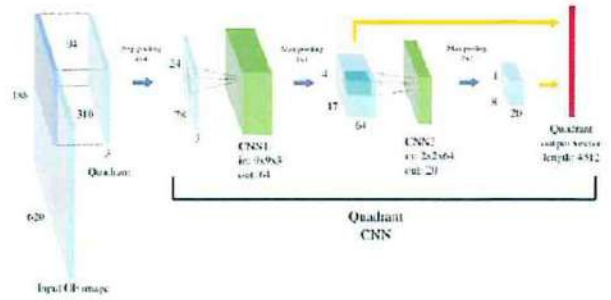


Fig. 3. Quadrant CNN architecture. The image is divided into four quadrants and each one passes through a chain of filter banks (CNN1-pooling, CNN2-pooling). To produce stronger visual features we concatenate the output of CNN1 and CNN2.

- 2) **CNN-4b VO**: The first alternative configuration tries to exploit local information. We divide the OF image into four sub-images. Each quadrant is down-sampled 4 times and then passed through a series of CNN filters analogous to CNN-1b ones. The final layer is trained to use the output of the four CNN networks to give a global F2F estimate.
- 3) **P-CNN VO**: The last architecture uses the CNN filters of both CNN-1b and CNN-4b feeding their output to a fully connected network. We do so to explore the performances of a network that merges the global information of CNN-1b with the local information of CNN-4b.

Our hypothesis for P-CNN is that the information of the two other networks is partially different, so can be combined to train a better estimator. In Section IV, we will show that this hypothesis is verified by our experiments. We start the description of these architectures from CNN-4b, since the structure of CNN-1b can be described in terms of the first one, and that P-CNN is the composition of the two.

The architecture of CNN-4b network is shown in Figure 3. The first section of the network is composed of four branches, identical in complexity, but trained separately, that perform the first two convolutional steps (CNN1 and CNN2). Note that each of the four quadrants of the image contains some motion information to compute a motion estimate, with ambiguity between simple turns and forward moving motion. We then link the output of the first CNN-pooling pair with the second one. We do so because exploratory experiments on a down-sampled version of the OF images showed that VO estimators using only CNN1 output, or only the cascade of the two CNNs, were both able to learn good estimators, but the VO estimator learned on the concatenation of the two outputs performed better. This result shows that CNN1 and CNN2 extract different information from the OF images. We presume that CNN1 extracts finer details, while the CNN2 extracts coarser ones, and that this information is not completely overlapping. After this stage the four convoluted features are put back together to form an image that contains the global information and thus is able to solve the motion ambiguities with symmetry information. The last layer computes a fully connected network that uses the information of all four quadrants at both resolutions, as shown in the upper part of Figure 4.



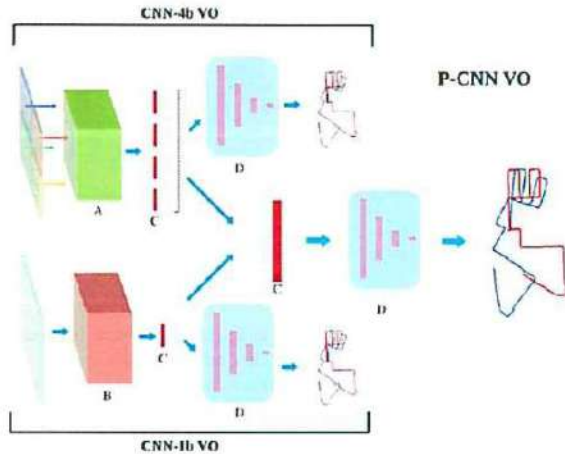


Fig. 4. Global P-CNN network architecture. The capital letters indicate: A) 4xQuadrant CNNs; B) 1-branch CNN; C) Extracted feature vectors of different sizes; D) Fully Connected Networks for estimation. The output of the Convolutional layers of CNN-4b and CNN-1b are concatenated and fed to the FCN of P-CNN.

CNN-1b architecture is similar in shape to a single quadrant of CNN-4b, but with different image dimensions. The initial down-sampling brings the image to a size of  $155 \times 48$ , so the CNN1 and CNN2 layers are wider, but maintaining the number of outputs to 64 and 20 respectively. The max pooling layers are the same. The last architecture, P-CNN, is a composition of the other two networks as shown in Figure 4.

### C. Training the Networks

Since it is known that training deep architectures in a global way is an open problem [25], we use the standard *greedy-layerwise* method to find good local minima for the filter coefficients of each layer and then we perform a global training to fine-tune them. More in detail, for each branch we train the CNN1 filter using a fully connected layer next to it to train a first estimator, then we drop the fully connected layers, feed the output of CNN1 to CNN2 and train only this one with a new fully connected estimator. Again we drop the fully connected layers, concatenate the two outputs of the CNNs and train a third estimator. We repeat this procedure for each branch, then we drop the last fully connected layers and concatenate the four quadrant outputs into a last fully connected network that trains the final estimator and fine tunes the CNNs coefficients. The final fully connected layers for the CNN-1b, CNN-4b and P-CNN have two hidden layers with (4500, 1000), (600, 2000) and (9000, 3000) nodes, respectively. The whole process requires a few days with a modern GPU or a few hours with a Tesla K40.

### D. Estimation Problem Formulation

We want to model a function  $f$  that given the OF of a pair of consecutive  $n \times m$  images, is able to output the camera motion that has generated it, filtering out the OF disturbances due to dynamic objects in the scene. The output of the function is the motion vector  $\mathbf{y} \in \mathcal{Y} \subset \mathbb{R}^6$  that encodes the displacement

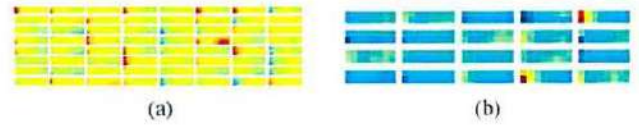


Fig. 5. Application of the convolutional filters to input optical flows. 5(a) and 5(b) show the output of CNN1 and CNN2 when processing the optical flow depicted in Figure 2(b).

of the camera centre and the three euler angles that represent the camera orientation, while the input  $\mathbf{x} \in \mathcal{X} \subset \mathbb{R}^{n \times m \times 3}$  is the RGB representation of the dense OF  $\mathcal{I} \in \mathbb{R}^{n \times m \times 2}$  that is a matrix with OF modulus and phase for each pixel. Thus, our function is defined as  $f: \mathcal{X} \rightarrow \mathcal{Y}$ .

The first convolutional layer performs  $K_1$  convolutions on each input  $\mathbf{x}_i$ ,  $i = 1 \dots N$  of a motion sequence of length  $N$ , producing an output  $\mathbf{h}_{i,k} \in \mathcal{Y} \subset \mathbb{R}^{(n-l+1) \times (m-l+1)}$  in this way:

$$\mathbf{h}_{i,k} = \mathbf{W}_k^1 * \mathbf{x}_i + b_k \quad (2)$$

where  $\mathbf{W}_k^1 \in \mathbb{R}^{l \times l \times 3}$ ,  $k = 1 \dots K_1$  are the filter coefficients,  $b_k$  is a bias and  $*$  is the convolution operator. The final output of the network is  $\mathbf{h}_i \in \mathcal{Y} \subset \mathbb{R}^{(n-l+1) \times (m-l+1) \times K_1}$ , that is the composition by the third dimension of every  $\mathbf{h}_{i,k}$ . This output is like an image slightly smaller in width and height, but with a number of channels equal to the number of convolution filters. After the first CNN1 block we put a max-pooling operation, that is a highly non-linear function that reduces the size of the CNN1 output image by 16 times and selects the maximal response for each non overlapping  $4 \times 4$  pixel tiles that compose the image. The second convolutional layer performs an analogous operation, but with different filter sizes:  $\mathbf{W}_k^2 \in \mathbb{R}^{q \times q \times 64}$ ,  $k = 1 \dots K_2$  and  $2 \times 2$  max-pooling.

The coefficients of  $\mathbf{W}_k^1$  and  $\mathbf{W}_k^2$  are learned in a supervised greedy layer-wise procedure, as explained in Section III-B, using fully connected NN with rectified-linear activation functions

$$\text{ReLU}(\mathbf{x}) = \max(0, \mathbf{x}) \quad (3)$$

and using root mean squared loss:

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^N \|\mathbf{f}(\mathbf{x}_i) - \mathbf{y}_i\|_2 \quad (4)$$

A sample representation of the effect of the convolutional filters after the training phase is depicted in Figure 5.

## IV. EXPERIMENTS

To evaluate the proposed approach we run experiments on a publicly available dataset. In particular, we compare our ego-motion estimation method based on CNNs (presented in Section III) with different SotA baselines. To further explore the robustness of these architectures, we also perform tests on artificially modified sequences, adding blur and changing contrast and luminance, to simulate adverse recording conditions such as low-light and motion blur.

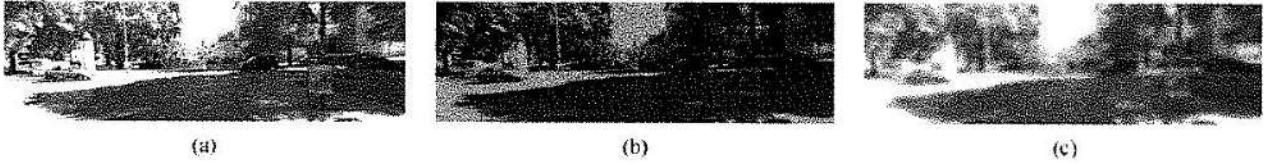


Fig. 6. Example of artificially darkened and blurred images a) standard image b) darkened with max contrast 0.4 and gamma 1.5 c) blurred with Gaussian blur radius of 10 pixels.

In the following, we describe the dataset used to evaluate the performances and the baselines used for comparison. Afterwards, we discuss the results and draw conclusions.

### A. Dataset

We test the performances of the CNN-based visual odometry on sequences taken from the KITTI benchmark suite [26], which is a common test bench for many vision algorithms. The sequences are gathered by a car traveling in the streets of the Karlsruhe city equipped with a Pointgrey Flea2 firewire stereo camera. The images are given already undistorted, with a resolution of  $1240 \times 386$  and a frame rate of 10 Hz (in some sequences the resolution is slightly higher: in these cases we perform a simple crop to uniform all the frames). Each pair of frames is associated with an absolute position with respect to the world reference frame computed with a high precision differential GPS and a Velodyne laser scanner. We used only 11 sequences since the remaining ones are not provided with ground truth. The first 7 are used as training input for the learned methods, described in the following Subsection, and performances of all the estimators are evaluated on the last three sequences 08, 09 and 10. Sequence 08 is filmed in the narrow streets of a peripheral neighbourhood, with some cyclists moving and lots of shadows. Sequence 09 presents a path on a very winding road, with background varying from countryside to suburbs. Sequence 10 presents a paved winding road with high slopes and a number of trucks and vans manocuvring in front of the camera.

In addition, to test the robustness of the baseline and presented methods, we produced 5 transformed versions of each test sequence. To do so we changed contrast and gamma to simulate different light conditions, and applied Gaussian blur of different radii to simulate defocus or motion blur. We call *darkened 1* the sequence with max contrast 0.4 and gamma 1.5, *darkened 2* the one with max contrast 0.6 and gamma 5.0, *lightened* the sequence with min contrast 0.2, max contrast 0.7 and gamma 0.2, and *blurred s3* and *blurred s10* the sequences with Gaussian blur of radius 3 and 10 pixels respectively. An example of these images is shown in Figure 6.

### B. Baselines

We compare the proposed approaches (CNN-1b VO, CNN-4b VO and P-CNN VO) to three different state-of-the-art baselines: a geometric monocular visual odometry, namely VISO2-M, described in [13], a regression model based on Support Vector Machine (SVR VO) as in [27], and a regression model based on standard Neural Network (FCN VO). The

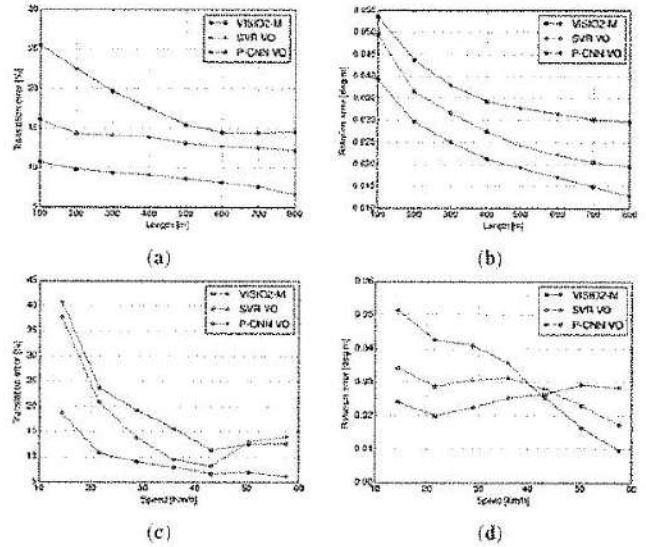


Fig. 7. Average errors across sequence lengths (a and b) and speeds (c and d) on test sequences for baseline and proposed methods.

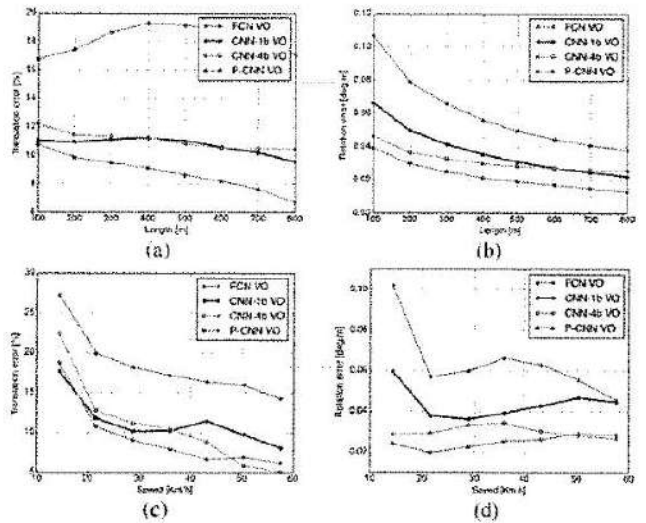


Fig. 8. Average errors across (8(a) and 8(b)) and speeds (8(c) and 8(d)) on test sequences for different network architectures.

implementation of VISO2-M [26] performs frame to frame estimation with some scale recovery using the known distance from the ground plane, but without bundle adjustment and feature tracking on other frames, so it is comparable to our method. The input features used for the training and testing of the SVR



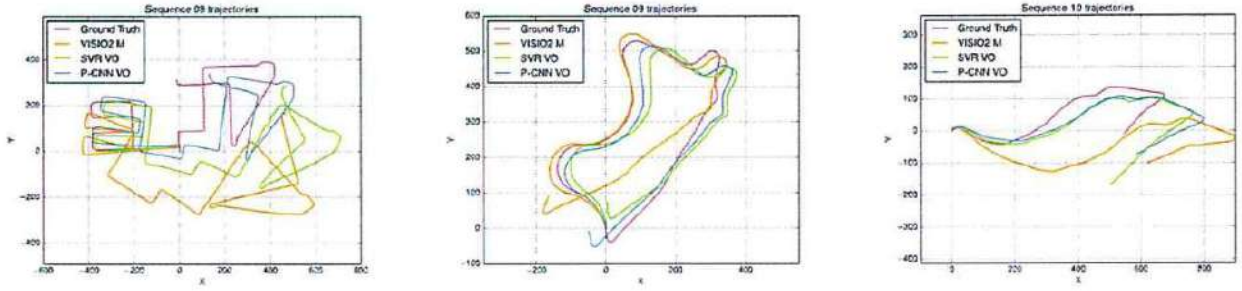


Fig. 9. Comparison of the proposed P-CNN approach with different baseline methods. The P-CNN has better performances almost at every path length and speed, except for high rotational speeds, where sparse features methods perform better. We explain this fact with the difficulty of dense optical flow to converge to meaningful results when the frame rate (in KITTI is 10Hz) is too low compared to the camera speed.

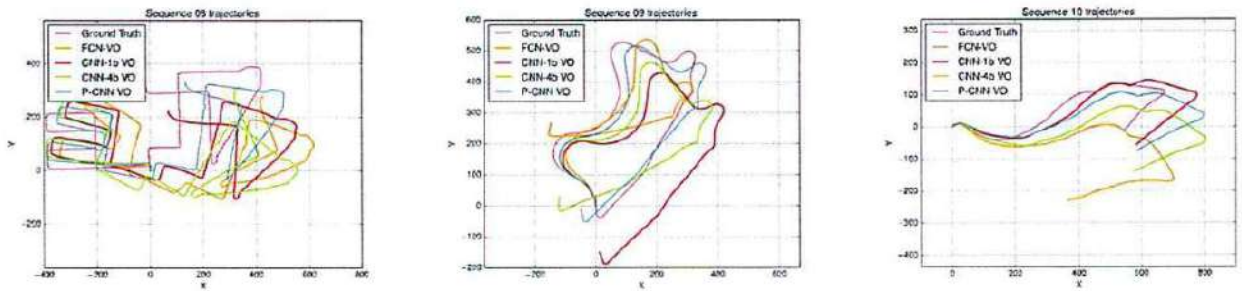


Fig. 10. Trajectories estimated for different network architectures. In this figure, we compare the parallel CNN (P-CNN) with respect to other network configurations.

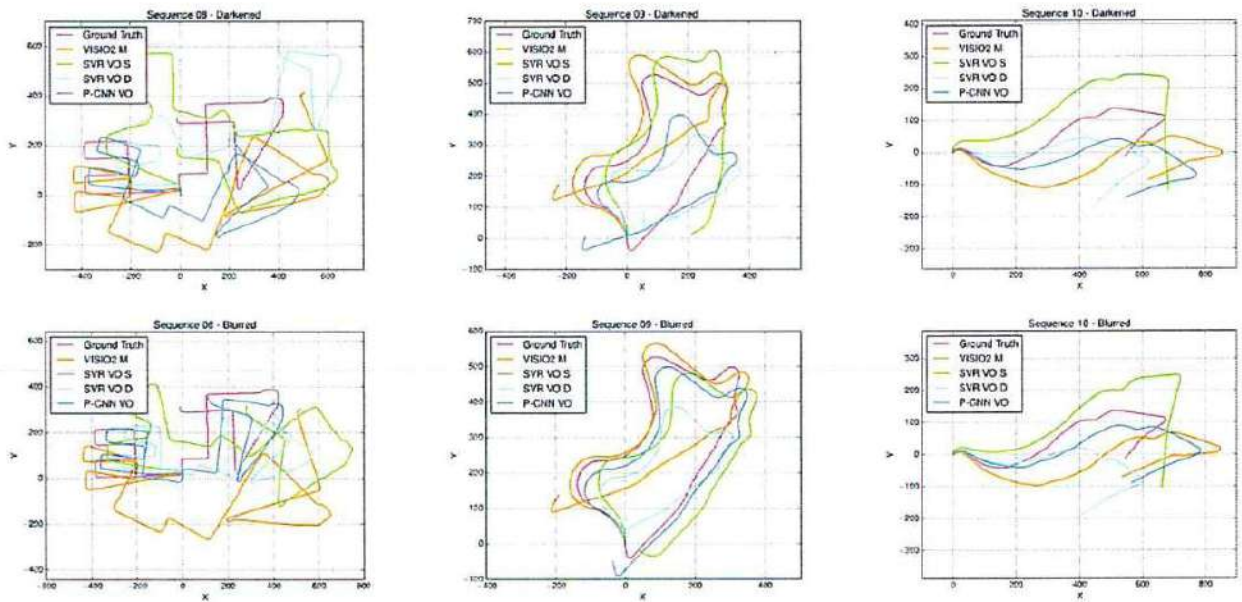


Fig. 11. Trajectories computed with different methods on artificially modified sequences: the first row shows the reconstructed trajectories when images are darkened (0.0-0.4 contrast, 1.5 gamma), while the second row depicts the resulting estimations with respect to blurred sequences (blur radius 3).

VO are computed following the strategy proposed in [27]: optical flows are computed on image pairs and quantized to obtain Histogram of Optical Flow (HOF). The training data, and test code are available on the accompanying web page [28]. The other regression baseline is a Fully Connected Neural Network (FCN VO) that we use to have a direct performance comparison with the CNN+FCN implementations. The training input is the same as for SVR VO. The network is composed by two

hidden layers of 1000 nodes and 6 output nodes. SVR VO, FCN VO and the proposed CNN VO are all trained using the KITTI sequences 00 to 07, and tested on 08 to 10.

The error metric we use to train the models and evaluate the performance is the *Root Mean Square Error* (RMSE) of the difference between predicted and true translations and rotations. We test the estimators using the evaluation code proposed in [26]: for each sequence the performances on different length



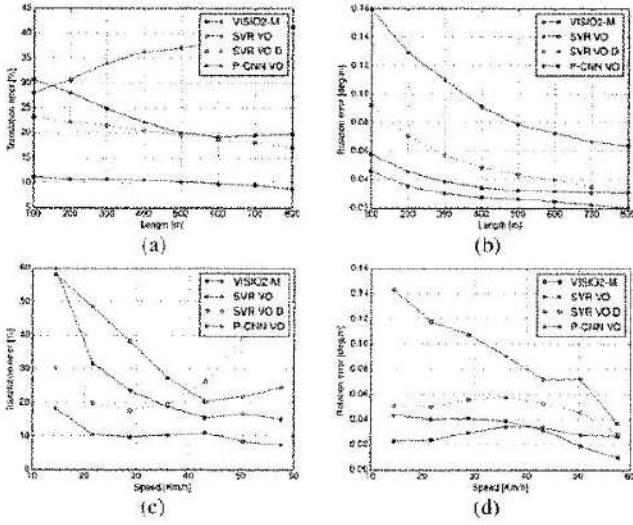


Fig. 12. Average errors across sequence lengths (12(a) and 12(b)) and speeds (12(c) and 12(d)) on artificially darkened sequences for different estimation algorithms (max contrast reduced to 0.4, and gamma increased to 1.5).

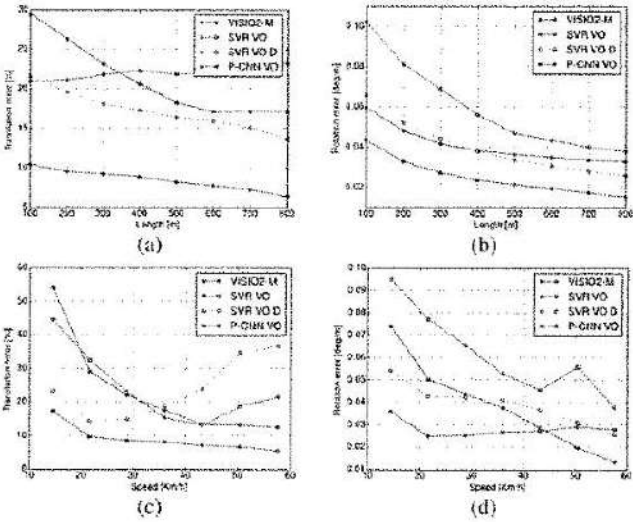


Fig. 13. Average errors across sequence lengths (13(a) and 13(b)) and speeds (13(c) and 13(d)) on artificially blurred sequences for different estimation algorithms (blur radius of 3 pixels). The general behaviour is similar to the one shown in Figure 12.

sub-sequences, ranging from 100 to 800 frames, are computed. Then the average error is computed as the average of the errors divided by the length of the sub-sequence. In addition, the translation and rotation errors corresponding to different speeds are computed. The results of the experiments on the standard sequences are shown in Figures 7 and 8, and the reconstructed trajectories in Figures 9 and 10. Some of the results of the experiments on the darkened and blurred sequences are shown in Figures 11, 12 and 13, while the rest are included in the accompanying web page [28].

### C. Discussion

Table I shows the performances of the baseline methods and the proposed P-CNN method on the three unmodified test

TABLE I  
COMPARISON OF TRANSLATIONAL AND ROTATIONAL ERRORS BETWEEN THE PROPOSED P-CNN AND THE BASELINES APPROACHES

	VIS02-M		SVR VO		P-CNN VO	
	Trans [%]	Rot [deg/m]	Trans [%]	Rot [deg/m]	Trans [%]	Rot [deg/m]
08	19.39	0.0393	14.44	0.0300	7.60	0.0187
09	9.26	0.0279	8.70	0.0266	6.75	0.0252
10	27.55	0.0409	18.81	0.0265	21.23	0.0405
Avg	18.55	0.0376	13.81	0.0302	8.96	0.0235

TABLE II  
COMPARISON OF TRANSLATIONAL AND ROTATIONAL ERRORS BETWEEN DIFFERENT CNN ARCHITECTURES

	CNN1b VO		CNN4b VO		P-CNN VO	
	Trans [%]	Rot [deg/m]	Trans [%]	Rot [deg/m]	Trans [%]	Rot [deg/m]
08	9.42	0.0363	9.91	0.0301	7.60	0.0187
09	11.33	0.0422	8.83	0.0286	6.75	0.0252
10	17.57	0.0319	23.45	0.0467	21.23	0.0405
Avg	10.77	0.0389	11.14	0.0324	8.96	0.0235

sequences. Results show that these sequences have different characteristics, and some are more challenging than others. However, in all the experiments we can observe that the SVR VO and the proposed P-CNN VO outperform VIS02-M, a SotA geometric F2F estimator. We omit the detailed results of the FCN VO because they under-perform all the other methods (Avg. transl. error 18.17%, Avg. rot. error 0.0626 deg/m), but its performances can be observed from Figures 8 and 10.

P-CNN performs better than SVM VO, except in sequence 10. However, the average errors of P-CNN VO are much lower than the other two methods. In Figure 7 the error contributions to the average errors divided per sequence length and speed range are depicted. From this figure, we see that P-CNN has a consistently lower error for each length and speed, except for rotation errors at high speeds, shown in Figure 7(d), where the trend is inverted. We explain this fact observing that rotations at more than 40Km/h are rare in the KITTI dataset, and the learning algorithms have few training examples to learn this behaviour.

In Table II the performances of different network architectures are shown. We compare CNN-1b, CNN-4b and the parallel combination of the two. P-CNN outperforms all the other architectures, except again for sequence 10. The fact that P-CNN has average errors much lower than the two single networks that compose it validates our hypothesis that CNN-1b and CNN-4b extract different information from the images. Thus, their combination in P-CNN is better than the performance of each one. In Figures 9 and 10, the trajectories of each method are shown for qualitative comparison.

Performance on darkened and blurred sequences show that P-CNN performs almost always better than the baseline methods in these scenarios. In the comparison we added an SVR VO using dense optical flow to explore the differences in performances with the features learned by the CNNs. From Figure 12(c) and 13(c) it is interesting to note that the SVR performances with dense optical flow degrades with higher translational speeds, while the P-CNN are always good. This result suggests that the increase in robustness is due to the features learned by the CNN, and not to the dense OF *per se*.

TABLE III

COMPUTATIONAL TIME COMPARISON BETWEEN THE PROPOSED P-CNN ARCHITECTURE AND THE BASELINE METHODS. THE TABLE SHOWS THE TIME REQUIRED TO COMPUTE A SINGLE F2F ESTIMATION, AVERAGED WITH RESPECT TO EACH SEQUENCE

	08 [s/img]		09 [s/img]		10 [s/img]	
	Standard	Blurred s10	Standard	Blurred s10	Standard	Blurred s10
VISIO2-M	0.0634	0.054	0.0682	0.0524	0.0697	0.0157
SVR VO Dense	0.1118	0.1123	0.1142	0.1113	0.113	0.1112
P-CNN VO (CPU)	0.311	0.209	0.205	0.301	0.309	0.307
P-CNN VO (GPU)	0.051	0.049	0.041	0.045	0.052	0.0504

Finally, we compare the computational costs between our P-CNN approach and the baseline methods (*i.e.*, VISIO2 and SVR VO Dense). The performance of VISIO2, SVR VO Dense and P-CNN (CPU) are evaluated using a i7-4720HQ 2.60 GHz processor, while an NVIDIA Tesla K40 GPU is used to run P-CNN (GPU). Table III shows that our approach takes 50 ms on average (30 ms to compute the optical flow and 20 ms to perform the CNN prediction) and, thus, it is well-suited for most real-time applications. As a final remark, VISIO2 runs at 0.0157 seconds per image on the blurred version of sequence 10 because it completely fails to extract features and perform the F2F estimation.

V. CONCLUSION AND FUTURE WORK

In this letter, we explored the architecture and performances of an ego-motion estimation approach based on Convolutional Neural Networks. We showed that this powerful learning paradigm is able to learn both new visual features and a high performing estimation model to achieve robust ego-motion estimation. We studied three different network architectures, comparing them with state-of-the-art ego-motion estimation methods on publicly available sequences. These experiments showed that all these architectures were able to autonomously select a new data representation that allowed the learned estimators to outperform other methods. We also tested the estimators on artificially degraded sequences and showed that the new features learned by the CNNs make the estimation pipeline more robust. We find this behaviour very promising, and future work will explore the performances of the integration of CNNs feature extractors with other direct and semi-direct SotA VO estimators. In addition, our experiments suggest that deep networks are very promising in general for VO estimators learning, and we will conduct further research to experiment with deeper networks and different architectures. In addition, since in this work we focused on F2F estimation, future work will also explore the improvements that can be achieved by integrating strategies such as bundle adjustment, scale estimation, or loop closing.

REFERENCES

[1] H. Strasdat, J. M. M. Montiel, and A. Davison, "Scale drift-aware large scale monocular slam," in *Proc. Robot. Sci. Syst. (RSS)*, Jun. 2010.  
 [2] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-scale direct monocular SLAM," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2014, pp. 834-849.  
 [3] C. Forster, M. Pizzoli, and D. Scaramuzza, "SVO: Fast semi-direct monocular visual odometry," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Jun. 2014, pp. 15-22.

[4] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, "DTAM: Dense tracking and mapping in real-time," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Nov. 2011, pp. 2320-2327.  
 [5] D. Scaramuzza and F. Fraundorfer, "Visual odometry [tutorial]," *IEEE Robot. Autom. Mag.*, vol. 18, no. 4, pp. 80-92, Dec. 2011.  
 [6] S. Song, M. Chandraker, and C. Guest, "Parallel, real-time monocular visual odometry," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2013, pp. 4698-4705.  
 [7] T. A. Ciarfuglia, G. Costante, P. Valigi, and E. Ricci, "A discriminative approach for appearance based loop closing," in *Proc. IEEE Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2012, pp. 3837-3843.  
 [8] V. Guizilini and F. Ramos, "Semi-parametric learning for visual odometry," *Int. J. Robot. Res.*, vol. 32, no. 5, pp. 526-546, Apr. 2013.  
 [9] V. Guizilini and F. Ramos, "Semi-parametric models for visual odometry," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2012, pp. 3482-3489.  
 [10] Y. Bengio, "Learning deep architectures for AI," *Found. Trends Mach. Learn.*, vol. 2, no. 1, pp. 1-127, 2009.  
 [11] D. Nister, O. Naroditsky, and J. Bergen, "Visual odometry," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, vol. 1, Jul. 2004, pp. 652-659.  
 [12] P. F. Alcantarilla, J. Yebes, J. Almazán, and L. M. Bergasa, "On combining visual slam and dense scene flow to increase the robustness of localization and mapping in dynamic environments," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2012, pp. 1290-1297.  
 [13] A. Geiger, J. Ziegler, and C. Stiller, "Stereoscan: Dense 3D reconstruction in real-time," in *Proc. IEEE Intell. Veh. Symp. (IV)*, Jun. 2011, pp. 963-968.  
 [14] A. Davison, I. Reid, N. Molton, and O. Stasse, "Monoslam: Real-time single camera slam," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 6, pp. 1052-1067, Jun. 2007.  
 [15] E. Eade and T. Drummond, "Scalable monocular slam," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, Jun. 2006, pp. 469-476.  
 [16] G. Nützi, S. Weiss, D. Scaramuzza, and R. Siegwart, "Fusion of IMU and vision for absolute scale estimation in monocular slam," *J. Intell. Robot. Syst.*, vol. 61, nos. 1-4, pp. 287-299, 2011.  
 [17] T. Brox, A. Bruhn, N. Papenber, and J. Weickert, "High accuracy optical flow estimation based on a theory for warping," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, May 2004, pp. 25-36.  
 [18] R. Roberts, H. Nguyen, N. Krishnamurthi, and T. R. Balch, "Memory-based learning for visual odometry," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2008, pp. 47-52.  
 [19] R. Roberts, C. Potthast, and F. Dellaert, "Learning general optical flow subspaces for egomotion estimation and detection of motion anomalies," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, Jun. 2009, pp. 57-64.  
 [20] V. Guizilini and F. Ramos, "Visual odometry learning for unmanned aerial vehicles," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2011, pp. 6213-6220.  
 [21] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, Dec. 2012, pp. 1097-1105.  
 [22] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, Jun. 2014, pp. 580-587.  
 [23] X. Zeng, W. Ouyang, M. Wang, and X. Wang, "Deep learning of scene-specific classifier for pedestrian detection," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2014, pp. 472-487.  
 [24] K. Jarrett, K. Kavukcuoglu, M. Razafato, and Y. LeCun, "What is the best multi-stage architecture for object recognition?" in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Sep. 2009, pp. 2146-2153.  
 [25] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Trans. Pattern Analysis Mach. Intell.*, vol. 35, no. 8, pp. 1798-1828, Aug. 2013.  
 [26] A. Geiger, F. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. Comput. Vis. Pattern Recog. (CVPR)*, Jun. 2012, pp. 3354-3361.  
 [27] T. A. Ciarfuglia, G. Costante, P. Valigi, and E. Ricci, "Evaluation of non-geometric methods for visual odometry," *Robot. Autom. Syst.*, vol. 62, no. 12, pp. 1717-1730, 2014.  
 [28] G. Costante, M. Mancini, P. Valigi, and T. A. Ciarfuglia. (2015). *Extra Materials for This Work* [Online]. Available: [http://www.sira.diei.unipg.it/supplementary/Deep\\_VO/extra.html](http://www.sira.diei.unipg.it/supplementary/Deep_VO/extra.html).



# Fast Robust Monocular Depth Estimation for Obstacle Detection with Fully Convolutional Networks

Michele Mancini<sup>1</sup>, Gabriele Costante<sup>1</sup>, Paolo Valigi<sup>1</sup> and Thomas A. Ciarfuglia<sup>1</sup>

**Abstract**—Obstacle Detection is a central problem for any robotic system, and critical for autonomous systems that travel at high speeds in unpredictable environment. This is often achieved through scene depth estimation, by various means. When fast motion is considered, the detection range must be longer enough to allow for safe avoidance and path planning. Current solutions often make assumption on the motion of the vehicle that limit their applicability, or work at very limited ranges due to intrinsic constraints. We propose a novel appearance-based Object Detection system that is able to detect obstacles at very long range and at a very high speed ( $\sim 300\text{Hz}$ ), without making assumptions on the type of motion. We achieve these results using a Deep Neural Network approach trained on real and synthetic images and trading some depth accuracy for fast, robust and consistent operation. We show how photo-realistic synthetic images are able to solve the problem of training set dimension and variety typical of machine learning approaches, and how our system is robust to massive blurring of test images.

## I. INTRODUCTION

Obstacle Detection (OD) is a challenging and relevant capability for any autonomous robotic system required to operate in real world scenarios, for safe path planning tasks and reaction to unexpected situations. Obstacle pose estimation must be fast enough to allow robot control system to react and perform required corrections. Since higher robot speeds require longer range detection to timely react, OD in automotive and autonomous aerial vehicle applications is particularly challenging. Obstacle definition changes according to the specific application. In automotive and ground-based robotic applications an obstacle is usually any vertical object raising from the ground, such as cars, pedestrian, traffic lights poles, garbage bins, trees etc. When Micro aerial Vehicles (MAVs) are considered, some other assumptions are required. For example horizontal structures, such as tree branches and overpasses, signs become relevant obstacles, since robot motion is no more constrained to a well defined street environment. In these cases the OD system has to be able to detect any physical object present in the scene.

The main techniques to address the OD problem are based on visual stereo systems. However, such systems are limited in detection range and accuracy by camera set-up and baselines [1], [2], which in turn pose a limit on maximum speed, and this is a tough constraint both in automotive

<sup>\*</sup>We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Tesla K40 GPU used for this research.

<sup>1</sup>All the authors are with the Department of Engineering, University of Perugia, via Duranti 93, Perugia Italy  
{michele.mancini, gabriele.costante,  
thomas.ciarfuglia, paolo.valigi}@unipg.it

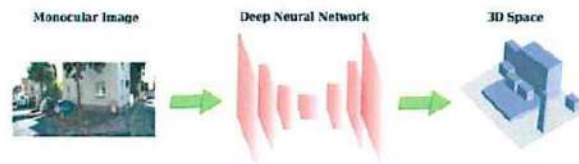


Fig. 1: We propose a fully convolutional network fed with both images and optical flows to obtain fast and robust depth estimation, with a robotic applications-oriented design.

and MAV applications. To overcome this limitations some systems exploited geometric knowledge about obstacles relationships with ground plane and assuming a limitation in the degrees of freedom of the vehicles movement, allowing long range obstacle detection up to 200 meters [3] or a real-time construction of a raw 3D obstacle mapping [4], [5], [6]. Unfortunately, these methods can't work in application where their geometric assumptions are violated and the robot does not operate at ground level.

Monocular based vision detection systems have been proposed to bypass both stereo vision limitations and geometric assumptions. Since monocular vision does not allow accurate and robust distance geometric measurement, often machine learning based solutions have been proposed [7], [8]. Since learning methods are limited by the training set samples and these methods have been trained using datasets with ground truths collected through stereo vision or laser rigs, these solutions still have limitations on range and accuracy as stereo systems.

To develop an OD system that is capable of detecting obstacles at high speed, allowing fast motion without geometrical assumptions, we propose a hybrid monocular approach that trades some detection accuracy for speed and general applicability. We decided to use monocular images to be able to apply the method on small or micro aerial vehicles that are able to move up to speeds of 10-20 m/s, for which a stereo approach would not be viable. In addition, we address the problem using Deep Neural Networks (DNNs) to learn an algorithm that is accurate and fast enough to allow fast reaction to unexpected obstacles on the vehicle path. To solve the limitations of machine learning approaches, namely the lack of data and generality of the solution, we extend the dataset with artificial sequences created using a state-of-the-art graphic engine capable of producing photo-realistic outdoor environments. This allows us to add an arbitrary number of sequences with perfect ground truth at very long distances (200m), that would not have been



possible to collect with a laser or stereo based ground truth system. Through our experiments we show that our algorithm is capable of doing fast estimation of depth with an accuracy that is sufficient for motion planning and that the learning on simulated photo-realistic environments is a viable way to extend datasets on robot vision problems.

This work is focused on depth estimation for obstacle perception and does not assess planning and control strategies to achieve effective obstacle avoidance. These aspects will be considered in future works.

## II. RELATED WORK

Most of traditional vision-based obstacle avoidance works rely on stereo vision. The most trivial solutions are based on finding disparities between the two matched images, compute point clouds and apply heuristics to detect obstacles. This methodology suffers from range limitations, produces sparse maps and may be not robust to pixel matching errors [9]. Many of these methods are based on *v-disparity* computation. Labayrade et al. [10], using a planarity assumption on stereo cameras, formulates a more robust analytical ground-plane estimation method based on *v-disparity* computation. Benenson et al. [5] use *v-disparity* and *u-disparity* to generate at high rate a fast obstacle representation on 3D space, while Harakeh et al. [11] build a probability field based on *v-disparity* to get a precise ground segmentation and occupancy grid of the scene. Pinggera et al. [3] improve range and accuracy detection using stereo vision to compute local ground normal as a statistical hypothesis testing problem, getting detection range up to 200 meters. Pillai et al. [6] propose a tunable and scalable stereo reconstruction algorithm which allows scene depth comprehension with very high frame rates, which may be usable for real time obstacle detection purposes. The main issue of these methods is that they make geometric assumptions, requiring planarity between stereo images. Also, methods such as [10], [5], [11] and [3] utilize ground model to position obstacles on 3D space, so obstacles posed over the ground won't be detected by these methods. Stereo vision has also been used as a data acquisition method for machine-learning approaches. Hadsell et al. [12] use a stereo rig to assign labels to close-range obstacles, detecting them using geometric techniques cited above. Features are extracted from image patches containing those obstacles through an offline-trained convolutional autoencoder. A classifier is trained using these features and obtained labels as reference. Distant obstacles are then detected by the online trained classifier. Ball et al. [13] apply a similar approach optimized for agricultural applications, based on a novelty-based obstacle detector. To overcome stereo methods geometric constraints, monocular vision-based methods have been proposed. Mori et al. [14] extract SURF features from monocular images and use template matching to detect frontal obstacles from their change in relative size between consecutive frames. This method makes no geometric assumption on the scene, but it has no capability to detect lateral obstacles and has limited range, which makes it unsuitable for high speed operations. Optical flow based

obstacle detection has been explored in [15], but it tends to be noisy in images with far away backgrounds, where optical flow tends to assume values close to zero. Day et al. [8] implement a imitation learning based reactive MAV controller based on monocular images, training a non-linear regressor to detect obstacles distances from features extracted from several patches of the image, as optical flow, histogram of oriented gradients, Radon transform, Laws' Masks and structure tensors. Ground truth for obstacles is obtained through a stereo rig. Being a machine learning based algorithm, it cannot perform better than the hardware used for training, so detection capability are limited by stereo vision weaknesses.

In the set of monocular methods, we also consider depth map estimators. These methods solve a different problem, as they try to find an accurate 3D reconstruction of the scene, but we use them as benchmarks for our methods. Michels [7] implements a reinforcement learning based 3D model generator with real-time capability. It relies on horizontal alignment of the images and does not generalize in less controlled settings. Eigen et al. [16] develop a deep learning based architecture for single image 3D reconstruction trained, on different experiments, on NYUDepth dataset [17] and KITTI dataset [18], obtaining state-of-the-art performance in terms of depth estimation accuracy. Although, for robotic application, we're not interested into obtaining state-of-the-art precision as it would require higher computational costs and it would be even unnecessary for our purposes, we consider these methods as reference for estimation performance.

We share with some of these methods the learning approach based on Deep Neural Networks (DNNs), but we fetch them not only with monocular images, but also with the optical flow of consecutive frames. Our architecture is inspired by recent works in semantic segmentation and optical flow estimation ([19], [20], [21], [22]), where Fully Convolutional Networks (FCNs) have been trained to make pixel-wise estimations, obtaining outputs of the same size of the input image. Differently from standard Convolutional Neural Network approaches, FCNs do not make use of fully connected layers, which account for most of the parameters of the network (e.g. on VGG-16 [23] architecture fully connected layers parameters are about 120M, out of the 134M parameters describing the whole network), and for this reason they improve training speed and reduce the amount of data required to train the deep network [21]. In addition, since convolution operations can be strongly optimized on GPU, these networks can generate estimates with very high frame rates.

## III. NETWORK STRUCTURE

For the network structure we propose an *encoder-decoder* architecture, similar to [22], [21] and [20]. Since the problem we tackle is depth estimation for Obstacle Detection and not for 3D reconstruction, we design network structure and complexity to be a good compromise between accuracy and execution speed.



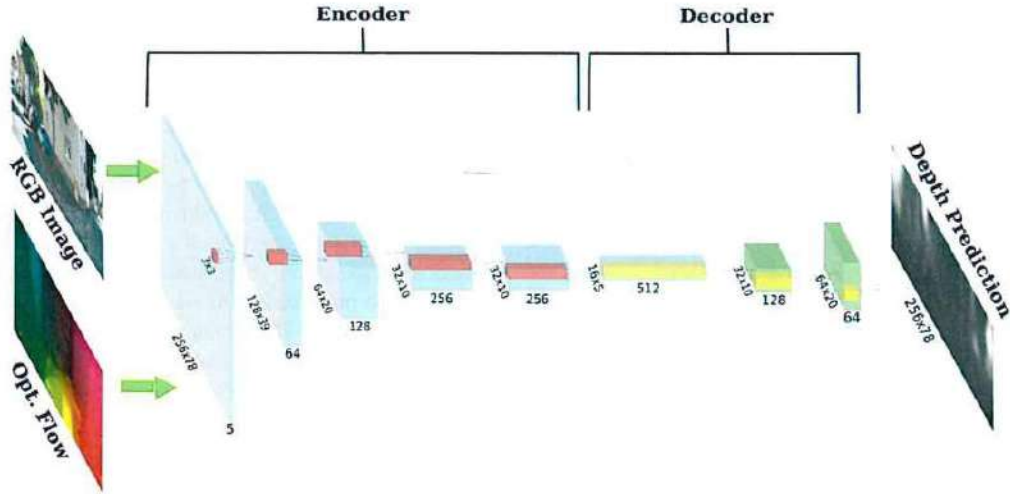


Fig. 2: Network architecture. Blue boxes: Encoder feature maps. Green boxes: Decoder feature maps. Convolutional filters are reported in red, deconvolutional filters in yellow.

#### A. Depth pixel-wise estimation as an encoder-decoder network

Our proposed architecture is reminiscent of fully convolutional architectures as [22], [21] and [20]. The encoder section is composed by a stack of convolutional layers, which apply learned filters on their input and extract relevant synthetic features. We do not apply naive pooling with an a-priori chosen strategy, such as max or average pooling. We choose instead to conveniently stride convolutions in order to obtain a downsampled version of its input. Convolution output's dimensions  $h_{conv}$  and  $w_{conv}$ , are determined, given an input of size  $h \times w$ , defining  $k$  as the convolution kernel size,  $p$  as convolution pad and  $s$  as applied stride, by the following equations:

$$h_{conv} = \frac{h + 2 * p_h - k_h}{s} + 1 \quad (1)$$

$$w_{conv} = \frac{w + 2 * p_w - k_w}{s} + 1 \quad (2)$$

From these equations we can infer how, choosing appropriate stride and padding, we are able to downsample information directly from convolutions, allowing the network to learn the optimal scaling strategy according to the task at hand. The decoder section is composed by a stack of deconvolutional layers, which learns to upsample from the features computed in the encoder section to obtain a final output of the same resolution of the input, containing pixel-wise predictions. Other works as [21] or [20] place unpooling layers between each deconvolutional layer to reverse pooling operations done in the encoder section; since downsampling is performed by convolution layers, we model our deconvolutional layers and learn the most effective upsampling strategy, as an inverse operation. Detailed network implementation is shown on Image 2.

Encoder section is composed by five  $3 \times 3$  convolutional layers. Strided convolutions are applied at the first, second, third and fifth layer of the encoder section to downsample feature maps. Padding is added accordingly to maintain desired feature maps size. ReLU non-linearity is applied after each convolution output. At the end of the encoder section, we obtain feature maps downsampled by a factor of 16 compared to network input.

Decoder section is composed by three deconvolutional layer. Each deconvolutional layer learn to upsample encoder feature maps by, respectively, a factor of 2 for the first two layers and a factor of 4 for the final layer, in order to obtain a final upsampling factor of 16. In [22] and [19] feature maps computed in intermediate convolutional layers in the encoder section are concatenated to each intermediate deconvolutional layer output to improve upsampling quality and edges definitions. We experimented this strategy in preliminary experiments; although we came upon a slight improvement on upsampling quality, we also experimented a performance degradation in terms of inference time, so we did not apply this strategy in successive experiments.

#### B. Image and optical flow as network input

In order to choose appropriate network input, we compare in our experiments two possible strategies: feeding the network with a single image, currently captured by the camera, or concatenate current image with optical flow information between current frame and the previous one. Optical flow has been used previously as raw feature for obstacle detectors [8] [15]. It is known how relative motion information between each pixel in two consecutive frames contains some implicit information about object dimensions and locations in 3D space. As previous works stated, optical flow alone is not sufficient to obtain a complete and long-range depth

estimation. Our intuition is to use it as additional information and let convolutional filters learn optimal strategy to extract useful information from it. Mixing together optical flow and raw image as network input, we expect them to overcome each other's limitations and improve performances and generalization capability in real world scenarios. Optical flow during our experiments has been computed off-line using widely-used and robust Brox algorithm [24], but faster and effective algorithms as [22] may be used as well to improve whole software pipeline real-time performance.

### C. Virtual Dataset

Collection of a sufficient amount of training data is a typical problem for every deep learning work. Considering how we formulate our problem of finding obstacles, we explore existing datasets containing depth ground truth. NYUDepth v2 indoor dataset [17], Make3D [25] and KITTI outdoor datasets [18] are typical choices for depth estimation-related problems. Make3D and NYUDepth datasets contain still images of a scene with no sequentiality between them, since they are thought for 3D image reconstruction, and this makes our optical flow-based approach not applicable. KITTI sequences are grabbed by a camera mounted on a moving car, making it more appropriate for robotic applications. KITTI depth ground truth, collected through a LiDAR unit, is sparse and does not cover the whole image scene. Moreover, images are generally aligned with ground plane, which may be a limitation according to the desired operating scenarios. Motivated by these reasons, we explored the possibility to collect data from virtual scenarios, utilizing development tools generally used in gaming industry, exploiting capabilities of the newest graphic engines. We utilize Unreal Engine 4 with Urban City pack developed by PolyPixel to build an urban scenario sized about 0.36 km<sup>2</sup>. No car, person, or dynamic object is present due to development package limitations; their inclusion will be considered as future work. We move a camera, collecting images and dense scene depth ground truth. Camera moves around the virtual world with six degrees of freedom, simulating non-trivial movements that rarely are present in real-world datasets, such as huge roll or pitch angles with respect to the ground plane. Depth is stored as a grayscale image: it is converted into metric depth by scaling each pixel value by a factor obtained through placing objects in a toy scenario at known metric distance from the camera. For preliminary experiments, depth is collected firstly with a maximum range of 40 meters; in a second phase, we collected depth measures up to 200 meters. Depth measures are spherical with respect of the camera. More than 265k images have been collected and stored in PNG format, with a resolution of 1241 × 376 pixels. Light conditions are changed from time to time to achieve brightness robustness. Haze is added in some sequences as well, and motion blur is simulated through graphic engine's tools, in order to better simulate real scenarios. Images are collected as sequences of consecutive frames captured as the camera moves around the world, at a rate of 10 Hz. We move the camera both on-road and off-road environments,

for example between trees or light poles, to better simulate possible realistic application scenarios.

## IV. EXPERIMENTS

To validate our work, we perform experiments on our Virtual Dataset, as described on Section III-C, as well as on KITTI dataset [18]. We also test our network's estimation robustness adding artificial blurring and darkening on KITTI images, to evaluate network's performance in presence of noise. For all of our experiments, we train the proposed networks on our Virtual Dataset, divided into a training set composed by about 200k images and a test set of 65k images. In order to evaluate the generalization capabilities of our approach, we do not perform any fine-tuning on the KITTI sequences. Training and testing are performed on a NVIDIA K40 GPU-mounted workstation.

We update weights during training by using Stochastic Gradient Descent (SGD) algorithm with a learning rate  $\alpha = 10^{-3}$ , gradually scaled down during training. Convergence is reached after about 50 epochs on training data. We train our final proposed architecture on Log RMSE (3), in order to penalize more errors on close obstacles than ones committed on long range estimations:

$$\sqrt{\frac{1}{T} \sum_{Y \in T} \|\log y_i - \log y_i^*\|^2} \quad (3)$$

Exploratory experiments are also performed with a linear RMSE loss, as specified later in Section IV-A.

Network inference time, without taking into account optical flow computation, is about 34 ms ( $\sim 300$ Hz) on K40 for each frame, which allows optimal scalability into complete embedded robotic software pipelines. Brox's optical flow algorithm used for our experiments runs at about 10Hz, but much faster algorithms, as [22], could be used as well.

The benchmark metrics for our comparisons are:

- Threshold error: % of  $y_i$  s.t.  $\max(\frac{y_i}{y_i^*}, \frac{y_i^*}{y_i}) = \delta < thr$
- Linear RMSE:  $\sqrt{\frac{1}{T} \sum_{Y \in T} \|y_i - y_i^*\|^2}$
- Scale-invariant Log MSE (as introduced by [16]):  $\frac{1}{n} \sum_i d_i^2 - \frac{1}{n^2} (\sum_i d_i)^2$ , with  $d_i = \log y_i - \log y_i^*$

### A. Virtual Dataset

	Single Image	Opt. Flow+Img.	
thr. $\delta < 1.25$	0.726	<b>0.774</b>	Higher is better
thr. $\delta < 1.25^2$	0.924	<b>0.938</b>	
RMSE	3.819	<b>3.478</b>	Lower is better
Log RMSE	0.246	<b>0.221</b>	
Scale Inv. MSE	0.065	<b>0.055</b>	

TABLE I: Experiments results on virtual dataset for ground truth collected up to 40m.

We initially perform exploratory experiments on Virtual Dataset test set to compare the performance of the two





Fig. 3: Some images from Virtual Dataset, highlighting lighting conditions and captured motion’s diversity comprised into the dataset

proposed architectures (as described in Section III-B). Networks are trained by using sequences with ground truth depth collected up to 40 meters, using Linear RMSE as training loss. Quantitative results are shown in Table I: the network that processes optical flow inputs outperforms single image network with respect to all the metrics, showing the effectiveness of the proposed optical flow+image network.

Afterwards, we compute metric depth up to 200 meters and re-train the optical flow-based architecture on log RMSE and linear RMSE (see Table II). The results show that the network based on the log RMSE loss achieve better performance with respect to the linear one.

	Log RMSE	Linear RMSE	
thr. $\delta < 1.25$	<b>0.643</b>	0.482	Higher
thr. $\delta < 1.25^2$	<b>0.887</b>	0.764	is better
RMSE	<b>6.065</b>	7.004	Lower
Log RMSE	<b>0.292</b>	0.416	is
Scale Inv. MSE	<b>0.085</b>	0.154	better

TABLE II: Comparison between the optical flow+image network trained on Log RMSE and the linear RMSE

### B. KITTI dataset

We perform experiment on KITTI dataset [18]. The sequences are gathered with a Pointgrey Flea2 firewire stereo camera mounted on a car traveling in the streets of the Karlsruhe city. Images are undistorted and collected with a resolution of  $1240 \times 386$  and a frame rate of 10Hz. As the provided depth ground truth is sparse, we compute dense ground truth by using the colourization routine proposed in [17]. Furthermore, since LiDAR provides ground truth measures only for the bottom half of the scene, experiments are performed with respect to that portion of data. Performance are evaluated on a test set composed by 697 images, corresponding to the published results of [16]. We first run exploratory experiments to evaluate the optical flow based architecture, re-using network weights trained on a maximum detection range up to 40 meters, as described in Section IV-A. Results are provided in Table III.

Finally, we perform experiments on our optical flow-based network trained for detection up to 200 meters, and we compare its performance with respect to state of the art depth

	Single Img.	Optical Flow+Img.	
thr. $\delta < 1.25$	0.311	<b>0.421</b>	Higher
thr. $\delta < 1.25^2$	0.572	<b>0.679</b>	is
thr. $\delta < 1.25^3$	0.764	<b>0.813</b>	better
RMSE	7.542	<b>6.863</b>	Lower
Log RMSE	0.574	<b>0.504</b>	is
Scale Inv. MSE	0.206	<b>0.205</b>	better

TABLE III: Results on KITTI Dataset on our architectures trained with a detection range  $< 40$  meters

predictors. In particular, we compare our performances with Eigen et. al [16] and Saxena et al. [25]. It is important to notice that these approaches are both trained and tested with respect to the KITTI sequences. Conversely, in order to prove the generalization capabilities of our approach, we train our network with respect to the Virtual Dataset sequence and test on the KITTI sequences without any fine-tuning procedure.

We report results in Table IV. For Saxena et al. work, we refer to the results provided in [16]. Although we do not perform any fine-tuning with respect to the real sequences, we obtain similar performance with respect to state-of-the-art approaches that are trained and tested on the same scenario. Furthermore, our network outperforms the other methods with respect to the scale invariant Log MSE metric that penalizes relative scale errors without considering absolute scale imperfections. Hence, we infer that our approach provides a accurate estimates with respect to relative depths.

The performance that we achieved with respect to linear RMSE and log RMSE metrics, suggest that our network weaknesses lie on close-range estimations, as log RMSE penalizes more mistakes on small values. We acknowledge that, for obstacle detection tasks, there are more robust methods than can detect close obstacles, such as laser sensors or stereo cameras. Thus, we believe that our approach could definitely improve depth estimation performance if combined with approaches tuned for short range detections.

### C. Testing network robustness

To test the robustness of our approach on different scene conditions, we performed additional experiments on the KITTI sequences, by producing transformed versions of each test sequence. To do so we changed contrast and gamma

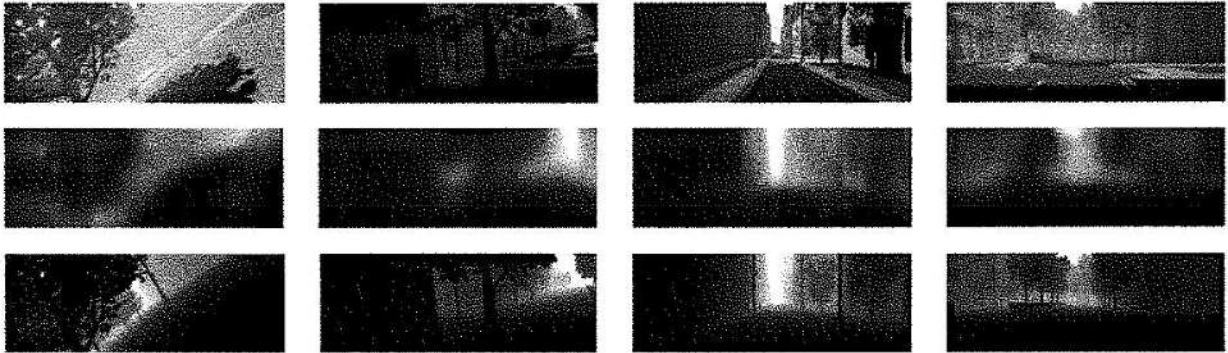


Fig. 4: Qualitative results on the Virtual Dataset. On the first row RGB input images are depicted. The second and the third rows show the network predictions and the dense ground truths, respectively.

	Our network	Eigen et. al [16]	Saxena et al. [25]	
thr. $\delta < 1.25$	0.318	<b>0.692</b>	0.601	Higher
thr. $\delta < 1.25^2$	0.617	<b>0.899</b>	0.820	is
thr. $\delta < 1.25^3$	0.813	<b>0.967</b>	0.926	better
RMSE	7.508	<b>7.156</b>	8.734	Lower
Log RMSE	0.524	<b>0.270</b>	0.361	is
Scale Inv. MSE	<b>0.196</b>	0.246	0.327	better

TABLE IV: Results on KITTI Dataset

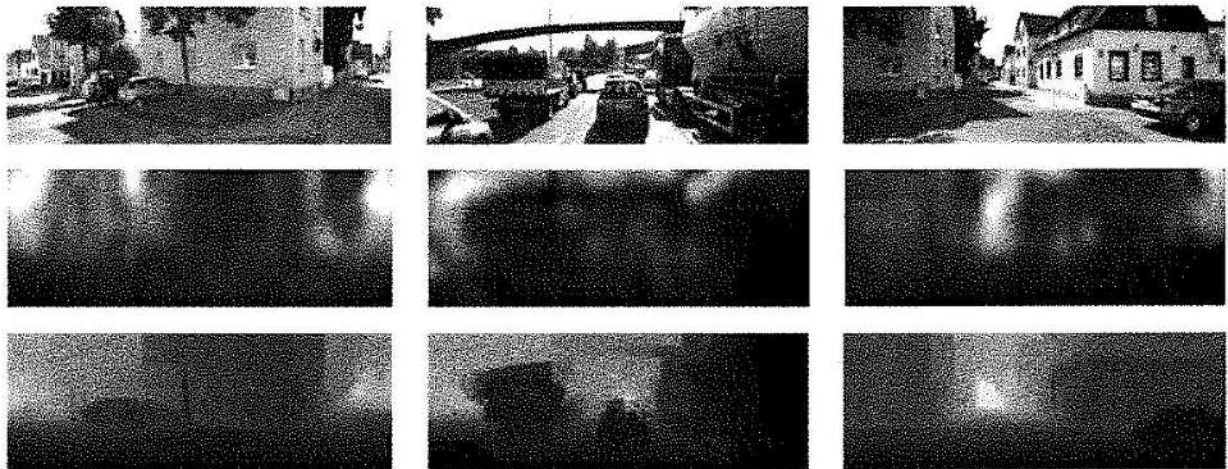
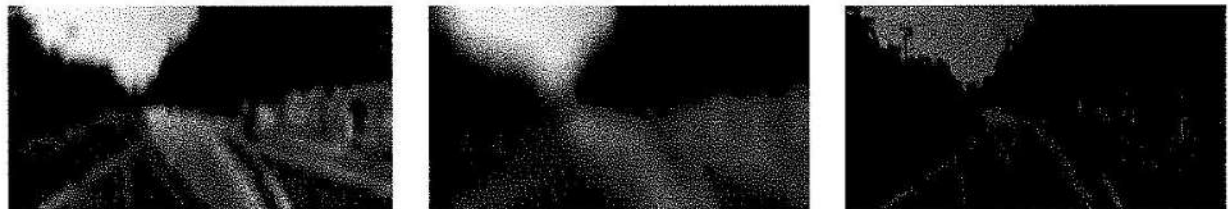


Fig. 5: Qualitative results on KITTI dataset. The first row shows the input RGB image, while the second row and the third rows show the network prediction the dense ground truth obtained by using the colourization routine, respectively.



(a) Blurred image with radius = 3

(b) Blurred image with radius = 10

(c) Darkened image with max contrast = 0.4 and gamma = 1.5

Fig. 6: Some artificial noise-added images as they have been tested in our experiments.



	Plain	Blur rad:3	Blur rad:10	Darkened	
thr. $\delta < 1.25$	0.318	0.244	0.142	0.176	Higher
thr. $\delta < 1.25^2$	0.617	0.525	0.350	0.348	is
thr. $\delta < 1.25^3$	0.813	0.741	0.573	0.509	better
RMSE	7.508	8.126	9.483	9.645	Lower
Log RMSE	0.524	0.606	0.792	0.923	is
Scale Inv.MSE	0.196	0.209	0.240	0.346	better

TABLE V: Results obtained on KITTI dataset applying Gaussian blur to images and changing lighting conditions.

to simulate different light conditions, and applied Gaussian blur of different radius to simulate defocus or motion blur. In particular, we add a gaussian blur with a radius of 3 (we refer to this experiment as *Blurred Image rad:3*, Figure 6(a) ) and 10 pixels (*Blurred Image rad:10*, Figure 6(b) ) and change image lighting by setting max contrast to 0.4 and gamma to 1.5 (*Darkened Image*, Figure 6(c) ). The results of the evaluation with respect to these sequences are shown in Table V. It was not possible to test Eigen et al. method on blurred images since they did not publicly release their network's weights trained on KITTI dataset, so we compare our results with their performance on non-blurred images. On *Blurred Image rad:3* experiment our performance is still better than Eigen et al. in terms of scale invariant log MSE error even after noise addition, and experience just a slight performance deterioration on other metric. On the *Blurred Image rad:10* and *Darkened Image* experiments, the estimations are less accurate, but results remains acceptable and comparable with other techniques on scale invariant log MSE metric. These experiments, thus, demonstrate our network capability to perform acceptable estimations even with very noisy images.

## V. CONCLUSION AND FUTURE WORK

In this paper, we explore the architecture and performances of a depth estimation algorithm based on a Encoder-Decoder Convolutional Neural Networks architecture. The proposed algorithm is intended to be the foundation of an Obstacle Detection system, meant to be run by fast vehicles. We address the limitations of stereo systems using a learning approach trained on synthetic images with long range ground truth. We test two kind of inputs, monocular images and monocular images with optical flow. Both networks trained on synthetic data have shown, compared to state-of-the-art methods, good performances on real data, suggesting that this training strategy is able to overcome some weaknesses of learning approaches, such as generalization and training data availability. In addition, we showed how the proposed algorithm is capable of estimating depth even when the starting images have been corrupted with blur, darkened or lightened. In future work we plan to increase network's robustness augmenting virtual training data, also adding dynamical objects to the scene. In future works we will consider finetuning on real images to improve performance. In addition we plan to integrate our depth estimator together with semantic segmentation algorithms and object detectors to obtain a semantic knowledge of the scene, useful to infer

information about estimation uncertainty and model more robust interpretations of the scene.

## REFERENCES

- [1] E. R. Davies, *Machine vision: theory, algorithms, practicalities*. Elsevier, 2004.
- [2] P. Pinggera, D. Pfeiffer, U. Franke, and R. Mester, "Know your limits: Accuracy of long range stereoscopic object measurements in practice," in *Computer Vision—ECCV 2014*. Springer, 2014, pp. 96–111.
- [3] P. Pinggera, U. Franke, and R. Mester, "High-performance long range obstacle detection using stereo vision," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*. IEEE, 2015, pp. 1308–1313.
- [4] M. Cordts, L. Schneider, M. Enzweiler, U. Franke, and S. Roth, "Object-level priors for stixel generation," in *Pattern Recognition*. Springer, 2014, pp. 172–183.
- [5] R. Benenson, R. Timofte, and L. Van Gool, "Stixels estimation without depth map computation," in *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*. IEEE, 2011, pp. 2010–2017.
- [6] S. Pillai, S. Ramalingam, and J. J. Leonard, "High-performance and tunable stereo reconstruction," *arXiv preprint arXiv:1511.00758*, 2015.
- [7] J. Michels, A. Saxena, and A. Y. Ng, "High speed obstacle avoidance using monocular vision and reinforcement learning," in *Proceedings of the 22nd international conference on Machine learning*. ACM, 2005, pp. 593–600.
- [8] D. Dey, K. S. Shankar, S. Zeng, R. Mehta, M. T. Agetayazi, C. Eriksen, S. Dufry, M. Hebert, and J. A. Bagnell, "Vision and learning for deliberative monocular cluttered flight," *arXiv preprint arXiv:1411.6326*, 2014.
- [9] S. B. Goldberg, M. W. Maimone, and L. Matthies, "Stereo vision and rover navigation software for planetary exploration," in *Aerospace Conference Proceedings, 2002, IEEE*, vol. 5. IEEE, 2002, pp. 5–2025.
- [10] R. Labayrade, D. Aubert, and J.-P. Tarel, "Real time obstacle detection in stereovision on non flat road geometry through" v-disparity" representation," in *Intelligent Vehicle Symposium, 2002, IEEE*, vol. 2. IEEE, 2002, pp. 646–651.
- [11] A. Hankech, D. Asmar, and E. Shammas, "Ground segmentation and occupancy grid generation using probability fields," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*. IEEE, 2015, pp. 695–702.
- [12] R. Hadsell, P. Sermanet, J. Ben, A. Erkan, M. Scoffier, K. Kavukcuoglu, U. Muller, and Y. LeCun, "Learning long-range vision for autonomous off-road driving," *Journal of Field Robotics*, vol. 26, no. 2, pp. 120–144, 2009.
- [13] D. Ball, B. Uperofit, G. Wyeth, P. Corke, A. English, P. Ross, T. Patten, R. Fitch, S. Sukkarieh, and A. Bate, "Vision-based obstacle detection and navigation for an agricultural robot," *Journal of Field Robotics*, 2016.
- [14] 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.
- [15] 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.
- [16] D. Eigen, C. Fuhrsch, 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.

- [17] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from rgbd images," in *Computer Vision—ECCV 2012*. Springer, 2012, pp. 746–760.
- [18] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Computer Vision and Pattern Recognition (CVPR)*, June 2012.
- [19] 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.
- [20] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1520–1528.
- [21] 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.
- [22] P. Fischer, A. Dosovitskiy, E. Ilg, P. Häusser, C. Hazurbay, Y. Golkov, P. van der Smagt, D. Cremers, and T. Brox, "Flownet: Learning optical flow with convolutional networks," *arXiv preprint arXiv:1504.06852*, 2015.
- [23] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014.
- [24] 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)*, May 2004.
- [25] A. Saxena, M. Sun, and A. Y. Ng, "Make3d: Learning 3d scene structure from a single still image," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 31, no. 5, pp. 824–840, 2009.



# A Robust Semi-Semantic Approach For Visual Localization In Urban Environment

Silvia Cascianelli<sup>1</sup>, Gabriele Costante<sup>1</sup>, Enrico Bellocchio<sup>1</sup>, Paolo Valigi<sup>1</sup>, Mario L. Fravolini<sup>1</sup>  
and Thomas A. Ciarfuglia<sup>1</sup>

**Abstract**—This paper provides a new contribution to the problem of vision-based place recognition introducing a novel appearance and viewpoint invariant approach that guarantees robustness with respect to perceptual aliasing and kidnapping. Most of the state-of-the-art strategies rely on low level visual features and ignore the semantical structure of the scene. Thus, even small changes in the appearance of the scene (e.g., illumination conditions) cause a significant performance drop. In contrast to previous work, we propose a new strategy to model the scene by preserving its geometrical and the semantical structure and, at the same time, achieving an improved appearance invariance through a robust visual representation. In particular, to manage the perceptual aliasing problem, we introduce a covisibility graph, that connects semantical entities of the scene preserving their geometrical relations. The method relies on high level patches consisting of dense and robust descriptors that are extracted by a Convolutional Neural Network (CNN). Through the graph structure, we are able to efficiently retrieve candidate locations and to synthesize *virtual locations* (i.e., artificial intermediate views between two keyframes) to improve the viewpoint invariance. The proposed approach has been compared with state-of-the-art approaches in different challenging scenarios taken from public datasets.

## I. INTRODUCTION

Self-localization ability is a crucial feature requested to future robotic systems in order to operate autonomously in complex urban environment. In this context, loop closure detection [1], based on machine vision, is an essential building block. Most of the existing approaches have been developed using low level features and have been tested using datasets having similar viewpoint and lighting conditions. In this simplified scenarios, state-of-the-art algorithms often achieve very good performance. However, when datasets with even small changes in viewpoint and appearance are considered, these algorithms often fail, because approaches based on low level features are typically sensitive to image gradients. For this reason, recent studies are moving toward the adoption of higher level visual features that have a closer relation to a semantic description of the environment, thus providing an increased robustness to viewpoint and appearance changes. In particular, the application of CNNs as object detectors and descriptors has shown promising results in several studies. The rationale behind place recognition

\*We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Tesla K40 GPU used for this research.

<sup>1</sup>All the authors are with the Department of Engineering, University of Perugia, via Duranti 93, Perugia Italy  
{gabriele.costante, thomas.ciarfuglia,  
enrico.bellocchio, paolo.valigi,  
mario.fravolini}@unipg.it  
silvia.cascianelli@studenti.unipg.it

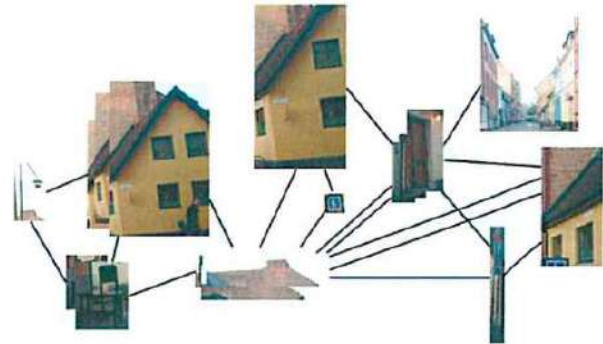


Fig. 1: Covisibility Graph of landmark example: nodes are patches extracted by Edge Boxes, described by the output features from conv3 layer of AlexNet, and are connected by an edge if they have been detected in the same image during environment traversal.

approaches that tend to rely on semantic objects in a scene is that the visual content of the image identifies specific semantic aspects of the place we are looking at and, at the same time, is robust to differences in visual conditions, since semantic information is unaffected by changes in viewpoint and appearance. However, sometimes, the presence of the same set of semantic objects is not enough to unequivocally identify a specific place (consider for instance the presence of objects such as cars of the same model, trees, lamps in a suburban scenario). To address this challenging situation the localization algorithm should be able to discriminate different spatial configurations of multiple objects.

Driven by the previous considerations, we propose a vision based place recognition approach that relies on visual features extracted by an inner layer of a pre-trained CNN run on image patches containing a semantic object, that increases robustness with respect to appearance changes. In addition, we face the viewpoint change problem by modelling the environment as a graph of semantic objects, thus exploiting also their arrangement to facilitate the place recognition task. The main contributions of this work are:

- The combination of appearance invariant features extracted from a pre-trained CNN with a graph based model of the environment that enhances viewpoint robustness
- The decrease of the perceptual aliasing problem by means of a semantic agnostic object extractor
- The automatic incremental building of a patch gallery

of the environment while exploring it

The remainder of this paper is organized as follows: In Section II, we discuss related research works, while the pipeline of the algorithm is described in Section III. Section IV provides experimental results. Conclusion and future development are discussed in Section V.

## II. RELATED WORK

Place recognition and loop closing are strictly related tasks that are particularly important for the autonomous robotic navigation in unknown environments. The main challenges encountered during the visual navigation in real scenarios are viewpoint changes and appearance changes due to illumination and seasonal variations or to the presence in scene of moving parts.

1) *Appearance invariant approaches*: This issue can be addressed via change removal methods, as in [2], or via change prediction, as Neubert et al. in [3], or by directly applying visual descriptors that exhibit invariance properties to appearance, as in [4]. In this work the authors trained a multi-layer perceptron model for learning an appearance invariant feature descriptor. Among appearance invariant descriptors, features obtained from inner layers of pre-trained object recognition CNNs have shown their effectiveness, as demonstrated in [5].

2) *Viewpoint invariant approaches*: Viewpoint changes are usually more critical than appearance changes. This issue is generally addressed in an application dependant fashion, both by applying image rectification methods in case of mild viewpoint changes [6], or by considering the specific type of changes in the viewpoint that will be encountered while performing a specific task, e.g lane traversal in [7], panoramic vision in [8] or air-ground viewpoint change in [9].

3) *Appearance and viewpoint invariant approaches*: Scenarios that feature both viewpoint and appearance changes are particularly challenging for the loop closure detection task. Promising solutions usually rely on CNNs specifically designed for place recognition, as done in [10] or on features extracted from a CNN designed for object recognition, as in [11], or viewpoint synthesis as done by [12], or exploiting robust sequence matching techniques, as in [7].

4) *Graph-based approaches*: Modelling the environment as a graph requires the definition of what a "node is" and of a criterion to link nodes. In order to preserve geometric information, [13], [14] proposed the employment of a geometric graph that is based on the distance between centres of 3D point cloud or 2D patch around a landmark. A recent work by Pepperell et al. [15] 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 by [16] and was 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 in 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 in a Transfer Learning paradigm [17].

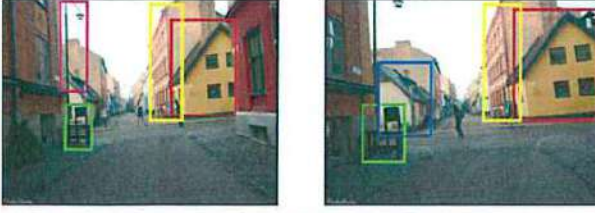
## III. STRUCTURE OF THE ALGORITHM

In this work, we propose a vision based approach that faces, at the same time, viewpoint and appearance changes. Image patches having a semantic content, called Edge Boxes, are extracted using the algorithm proposed by Zitnick et al. in [18]. These visual patches are then processed by a pre-trained CNN and the output of an inner layer is taken as the descriptor in order to obtain features that are invariant to appearance changes. These intermediate CNN landmark objects constitute the nodes of a covisibility graph, that are connected by an unweighted and undirected edge if the corresponding landmarks are observed in the same image along the path (see Fig. 1). Furthermore, landmarks that are labelled as "the same landmark" are mapped in a unique node, thus the resulting incremental graph embeds a sort of a landmarks gallery of the environment. At query time the algorithm extracts the sub-graph of covisible landmarks of the query image and the associated collection of gallery elements. By using the gallery, previous images that share a fraction of the actual features are retrieved and their covisibility graphs are merged to synthesize "virtual images" that can help the matching of the current frame thus facilitating the loop closure. Source code of our approach is available at <http://www.sira.diei.unipg.it/supplementary/GOCCEF>.

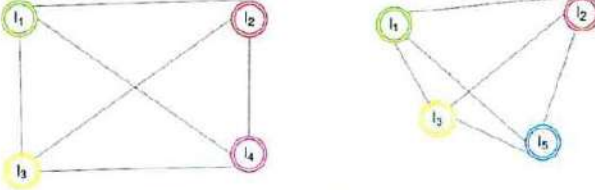
### A. Modelling the environment as a graph

Inspired by the work of Stumm et al. [16], we build a covisibility graph that models the environment as a structured collection of visual patches. Each patch contains an object with high probability, however our method does not rely on the specific class of object, i.e. it works at a semi-semantic level. A user defined number of patches (50 in our experiments) have been extracted with the Edge Boxes algorithm by [18], which are of varying size and contain an intelligible object. These patches are then used as input of the pre-trained AlexNet CNN [19]. Based on the analysis reported in [5] we decided to select the output of the conv3 layer as the descriptor of the patch, because it exhibits appearance invariance properties. Note that this layer produces a vector of  $13 \times 13 \times 384 = 64896$  components for each input patch. In order to reduce computing time we decide to reduce the dimensionality of the above descriptor via Gaussian Random Projection [20], that also approximates the cosine similarity between conv3 outputs, thus obtaining a reduced vector of length 2048. So obtained landmarks are used as nodes of the covisibility graph model of the environment that has to be built. Nodes associated to landmarks in the same image

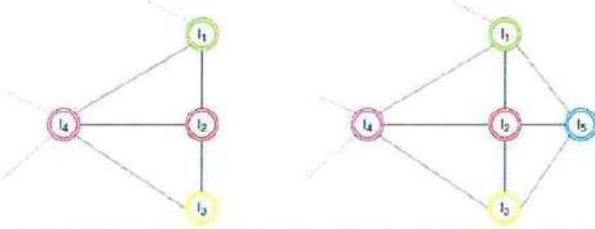




(a) Example of Edge Boxes landmark extracted from images encountered at time  $k - 1$  (left) and at time  $k$  (right) respectively



(b) Landmark covisibility subgraphs of images encountered at time  $k - 1$  (left) and at time  $k$  (right) respectively; landmarks that are seen together are connected in a dense graph



(c) Landmark covisibility whole graph at time  $k - 1$  (left) and at time  $k$  (right) respectively. Sufficiently similar landmarks are mapped to the same node, while different landmarks give rise to new nodes

Fig. 2: Covisibility graph construction while incrementally traversing the environment

are all connected by an undirected edge to encode their copresency. This implies that each image is associated to a connected subgraph. The complete graph is represented by a clique matrix  $M_{clique}$ , whose rows represent landmarks and columns represent image indices, so that a 1 in  $[M_{clique}]_{l,f}$  means that the landmark  $l$  is present in the image  $f$ . An illustration of the graph building process is shown in Fig. 2.

### B. Mapping landmarks in the same node

Landmarks that belong to consecutive images are mapped in the same node of the graph when they are recognized to be the same landmark (patch). The similarity is computed by means of the scalar cosine distance  $d_{ij}$  between the landmark's feature vector  $l_{c,i}$  of the current image and its nearest neighbour  $l_{p,j}$  taken from all the previous images. For each nearest neighbour pair of landmarks we also calculate the similarity of the geometric shape, in terms of width  $w_{c,i}$  and  $w_{p,j}$  and height  $h_{c,i}$  and  $h_{p,j}$  of their bounding boxes. The overall similarity between landmarks in the current image and their nearest neighbour in the previous ones is then computed as:

$$L_{ij} = 1 - d_{ij} \cdot e^{\frac{1}{2} \left( \frac{|w_{c,i} - w_{p,j}|}{\max\{w_{c,i}, w_{p,j}\}} + \frac{|h_{c,i} - h_{p,j}|}{\max\{h_{c,i}, h_{p,j}\}} \right)} \quad (1)$$

Finally, landmarks are labelled as the "same landmark" (and mapped in the same node of the graph) in case the overall similarity  $L_{ij}$  is larger than a user defined threshold (set to 0.3 in our experiments). This mapping is translated into graph clique matrix update. Namely, a new column is added for the current image, which has 1s in the existing rows corresponding to already observed landmarks and, in case the landmarks are recognized to be new, new rows are allocated, that have 1s in the last column, corresponding to the current image, where they first have been observed.

This allows us to consider the clique matrix as an inverted index of the images. In fact, by looking at the row of the clique matrix which is associated to a landmark, positions of ones in the matrix represent the indices of images in which that landmark has been observed. This last aspect is very useful at query time since it allows to efficiently retrieve the subset of stored images that share the at least a landmark with the query image. The subsequent comparison is carried out considering only the subset of retrieved images.

It is noteworthy that, in order to speed up the search of incoming landmark's nearest neighbor, we exploit the KD-Tree algorithm by [21], that organizes landmark descriptors in a tree structure based on their distance. The tree can then be explored in order to find the query landmark's nearest neighbour in logarithmic time. It should be noticed that the above algorithm works only with distance metrics that are component-wise additive and monotonically increasing with components addition, as in case of the Euclidean distance. Cosine Similarity is more suitable than Euclidean distance for high dimensional data, but does not exhibit the characteristics required for the employment in KD-Tree construction. Thus, we calculate the Euclidean distance between  $l_2$ -normalized feature vectors and then applied the following transformation:

$$d_{ij} = 1 - \frac{d_{Euclidean,ij}}{2} \quad (2)$$

where  $d_{Euclidean,ij}$  is the Euclidean distance between the landmarks  $l_{c,i}$  in the current image and  $l_{p,j}$  in the previous images.

### C. Synthesizing virtual locations

When a scene is revisited by the autonomous agent it is reasonable to assume that a modification of camera view point occurred. This implies that in the new scene some detected landmarks can have different relative position, other can be occluded, while some new ones enter in current view. In order to face this important issue, considering matching candidate images (selected and retrieved with the procedure explained in III-B), we compute new virtual locations by fusing their subgraphs if they share a sufficient large number of landmarks, *i.e.* at least the half of the fixed number of patches extracted in each image. In this way, we obtain an interpolated view between contiguous images, that enhances the scene database with additional virtual views as done for instance in [16]. Those virtual locations can facilitate matching and loop closure detection.

#### D. Recognizing places

Retrieved real and virtual locations are then scored to establish if they match with the current query location. The similarity measure is derived considering the landmarks' feature vectors and the shape parameters of their bounding box. In this way, it is possible to compute a similarity score between each pair of landmarks, both in the current and previous images,  $L_{ij}$ , as it has been made for tracking landmarks during graph building phase (see III-B), but in this case in a very limited search space, *i.e.* landmarks in the query image are only matched against those in the candidate (either real or virtual) location under investigation. The score of matching images is assigned as the mean value of individual scores of matching elements. At this point the information embedded in the graph is used to refine the above computed matching scores. Considering the clique matrix columns of query and candidate images, we obtain the adjacency matrix of their landmark covisibility subgraph. This matrix has as many rows and columns as the number of nodes that define the graph and models the connectivity between pairs of nodes with zeros (not connected) and ones (connected) in corresponding positions. Adjacency matrix can be easily obtained as  $A^f = H(M_{clique}^f \cdot M_{clique}^{fT})$ , where  $H(\cdot)$  is the element-wise heaviside step function. Notice that during graph construction nodes are kept ordered, thus rows and columns indices of the adjacency matrix refer to the same node in both query and candidate images subgraphs. This allows subgraphs to be aligned [22] and their direct comparison is possible by means of their adjacency matrix. We then calculate a normalized cross-correlation between candidate and query adjacency matrices as:

$$\gamma_{C,P_n} = \frac{\sum_{ij} A_{ij}^C \cdot E_{ij}^{P_n}}{\sqrt{\sum_{ij} (E_{ij}^C)^2 \cdot \sum_{ij} (E_{ij}^{P_n})^2}} \quad (3)$$

where  $A_{ij}^C$  and  $A_{ij}^{P_n}$  are the adjacency matrix entries relative to landmarks  $l_i$  and  $l_j$  in the graph of query location  $C$  and candidate location  $P_n$  respectively.

We "filter" the similarity score of each candidate location by maintaining normalized cross-correlation values that were lower than 0.1, as:

$$\hat{\gamma}_{C,P_n} = \begin{cases} \gamma_{C,P_n} & \text{if } \gamma_{C,P_n} < 0.1 \\ 1 & \text{if } \gamma_{C,P_n} \geq 0.1 \end{cases} \quad (4)$$

The so obtained  $\hat{\gamma}_{C,P_n}$  is used to weight the similarity score of each candidate location, thus filtering out matching scores of candidate location which landmark arrangement was too different from that one of query location. The resulting matching score is:

$$S_{C,P_n} = \hat{\gamma}_{C,P_n} \cdot \frac{1}{n_C} \sum_{ij} L_{ij} \quad (5)$$

where  $n_C$  is the number of landmarks in the current image. In addition, we assign each image the mean matching score obtained both as a real location and as part (seed) of one or more virtual locations.

## IV. EXPERIMENTS

The experimentation was carried out considering two benchmark loop closing datasets for evaluation, namely the City Centre dataset [23] and the New College dataset [23], that have been subsampled at different rates, while parameter tuning has been made on a different third dataset. Our approach was compared to a "low level features" approach [23] and to a "high level features approach" [11]. The comparison was done in term of a classical precision-recall analysis. Finally, to better highlight the role of virtual locations for loop closure detection, an additional study was conducted by analysing the performance of our approach in case the virtual locations information is excluded.

### A. Datasets

Both datasets consist 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 for the first dataset and a 1.9 km path in a university campus for the second one. Note that, since right and left images collected at the same time are mostly independent from each other, we concatenated each pair and considered the new "panoramic" wider images in our experiments.

1) *City Centre dataset*: This dataset is considered particularly challenging due to the presence of dynamic elements, such as pedestrians and vehicles in the scene and instability due to shadows and foliage.

2) *New College dataset*: In this, the trajectory is particularly articulated and presents many loops and stretches that are traversed also in opposite direction. Also this dataset contains many dynamic elements, such as pedestrians, and repeated features since it was acquired in an area that includes similar repeating walls, archways and bushes.

### B. Experimental Setup

Our algorithm was tested by incrementally building the covisibility graph (starting from scratch) as the robot follows its path. The current frame is considered as the query for the loop closing algorithm. In order to asses the benefits of the proposed covisibility graph, we subsampled the test datasets at different rates and compared the results in term of precision and recall. In this framework we have compared performance of our approach (referred to as 'GOCCE' - Graph Of Covisible CNN Extracted features - in the following) with those of:

- A state-of-the-art algorithm that exploits low level features and does not use a graph for environment representation. The selected algorithm is the well-known FABMAP [23]
- A technique that is based on the high level features extracted by Edge Boxes and AlexNet conv3, that were also used in our work, but does not use any graph-based representation of the environment, namely the approach proposed by Sunderhauf et al. in [11]. This algorithm was re-implemented and used in incrementally fashion as done in our approach



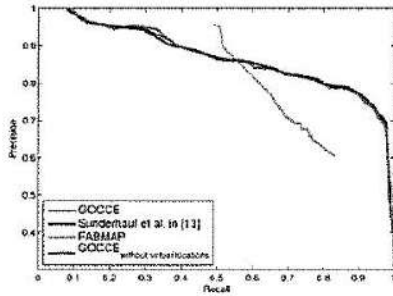


Fig. 3: Precision-recall curves for comparison of different techniques with respect to our approach on not subsampled New College dataset

- A reduced version of our approach that does not include the construction of virtual locations

### C. Results

Fig. 3 shows the precision-recall curves obtained with different approaches on the not subsampled New College dataset. In this case FABMAP obtains the higher recall at high precision, but its performance decrease significantly as the recall grows. It is important to underline that FABMAP was pre-trained using a dataset collected in the same environment as the one of the New College dataset. For this reason its performance expected to be superior. In fact, while tested on a different dataset, namely the City Centre dataset, our method significantly outperforms FABMAP (this trend can be observed in Fig. 4a, where precision-recall curves for not subsampled City Centre dataset are compared) and is slightly better than the graph-free approach. In fact, at 90% precision the method by Sunderhauf et al. in [11] obtains 88.47% recall, our reduced method 87.41% while our complete method 89.03%; in case of 90% recall the graph-free approach obtains 84.63% precision, the reduced version of our approach 86.03% and the complete approach 87.96%. The role of the covisibility graph and, in particular, the role of virtual locations, can be understood by comparing the results achieved with subsampled datasets. In Fig. 4b and Fig. 4c the precision-recall curves obtained in the City Centre dataset with subsampled images (decimated at a rate of 5 and 10) are shown. By analysing the above results we can observe that a severe subsampling enhances the utility of virtual locations, since, due to subsampling, some landmarks arrangements can result not previously seen. Thus, virtual locations including those landmarks allow real locations from which they have been created to be matched to the query image. This translates in a higher recall at maximum precision on subsampled datasets and higher precision at high recall. In this respect, note the precision values at 90% recall: it is 78.85% for the method of Sunderhauf et al. in [11], 80.39% for the version of our approach that does not exploit virtual locations and 87.05% while using virtual locations. Finally, it is noteworthy that in case of higher sampling rate, refining the matching score via candidates retrieval and subgraph comparison causes improvements in both precision and recall, while the performance of the

approach including virtual locations are minor, especially in terms of precision at high recall. This is due to the fact that the virtual locations construction process we applied in this work has no constraints but the number of shared nodes, thus real dataset images are forced to merge in order to build new virtual locations even when not needed.

In Fig. 5 it is shown the planar path traversed by the robot in the City Centre dataset, and it is underlined the GPS position of the subsampled images and that of seeds real locations. Note that virtual locations are often created near curves and angles and in stretches traversed with slight lateral displacement. Furthermore, we observe that large decimation ratios can lead to a significant decrease in the number of new virtual locations, especially near 90° trajectory corners. This because due to decimation the visual overlap between subsequent images can be lost during a turning manoeuvre.

### V. CONCLUSION AND FUTURE WORK

In this work, we proposed an environment agnostic, appearance and viewpoint invariant place recognition system, that is also robust to perceptual aliasing. The method is based only on machine vision images. These desirable characteristics were achieved by effectively and, in some extent, geometry preserving modelling the environment via a covisibility graph, whose nodes are features extracted by the inner convolutional layer of a pre-trained CNN that are remarkably robust to appearance changes.

Experimental validation has shown that our approach provides performance improvements comparable with state-of-the-art place recognition techniques and outperforms these methods in particularly challenging scenarios.

The representation of the environment as a graph of landmarks also eases the integration of additional information from other sensors and can be further enriched with information about the reliability of semantic elements that served as nodes, that could be moving objects.

A possible extension of this work can consider the matching of sequences of images from the datasets, rather than matching of single images. Another interesting direction could be the construction of virtual locations via a parameter-free approach based on local graph clustering via subgraphs connectivity.

### REFERENCES

- [1] T. A. Ciarfuglia, G. Costante, P. Valigi, and E. Ricci, "A Discriminative Approach for Appearance Based Loop Closing," in *IEEE International Conference on Intelligent Robots and Systems (IROS)*, October 2012.
- [2] C. McManus, W. Churchill, W. Maddern, A. D. Stewart, and 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.
- [3] P. Neubert, N. Sünderhauf, and P. Protzel, "Superpixel-based appearance change prediction for long-term navigation across seasons," *Robotics and Autonomous Systems*, vol. 69, pp. 15–27, 2015.
- [4] N. Carlevaris-Bianco and 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.
- [5] N. Sunderhauf, S. Shirazi, F. Dayoub, B. Uperofi, and 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.

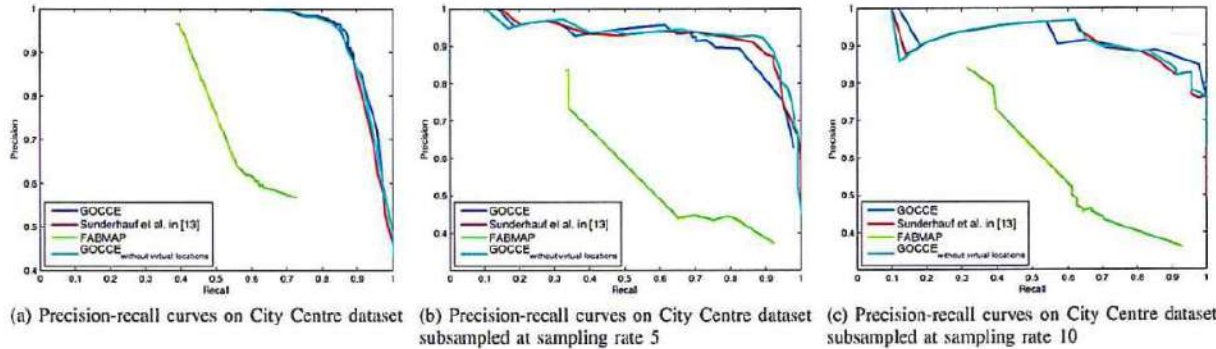


Fig. 4: Precision-recall curves for comparison of different techniques with respect to our approach on City Centre dataset at different sampling rates

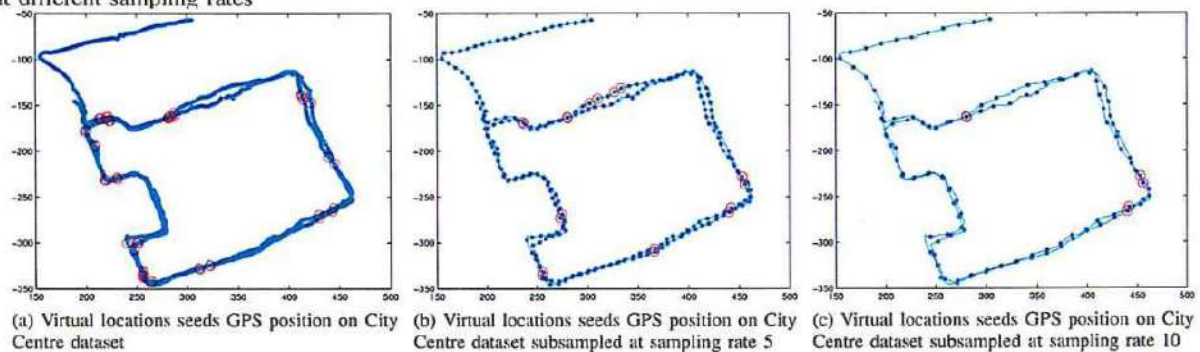


Fig. 5: Virtual locations seeds GPS position on City Centre dataset at different sampling rates. Dots corresponds to sampled images and red circles underline real locations that have been fused to obtain virtual locations

- [6] H. Yang, S. Cai, J. Wang, and 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.
- [7] M. Milford, C. Shen, S. Lowry, N. Sunderhauf, S. Shirazi, G. Lin, F. Liu, E. Pepperell, C. Lerma, B. Upcroft, et al., "Sequence searching with deep-learned 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.
- [8] R. Arroyo, P. F. Alcantarilla, L. M. Bergasa, J. J. Yebes, and S. Gámez, "Bidirectional loop closure detection on panoramas for visual navigation," in *Intelligent Vehicles Symposium Proceedings, 2014 IEEE*. IEEE, 2014, pp. 1378–1383.
- [9] A. L. Majdik, D. Verda, Y. Albers-Schoenberg, and D. Scaramuzza, "Air-ground matching: Appearance-based gps-denied urban localization of micro aerial vehicles," *Journal of Field Robotics*, vol. 32, no. 7, pp. 1015–1039, 2015.
- [10] R. Arandjelović, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "Netvlad: Cnn architecture for weakly supervised place recognition," *arXiv preprint arXiv:1511.07247*, 2015.
- [11] N. Sunderhauf, S. Shirazi, A. Jacobson, F. Dayoub, E. Pepperell, B. Upcroft, and M. Milford, "Place recognition with convnet landmarks: Viewpoint-robust, condition-robust, training-free," *Proceedings of Robotics: Science and Systems XII*, 2015.
- [12] D. Mishkin, M. Perdoch, and J. Matas, "Place recognition with wxbs retrieval," in *CVPR 2015 Workshop on Visual Place Recognition in Changing Environments*, 2015.
- [13] R. Finman, L. Paull, and 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.
- [14] J. Oh, J. Jeon, and B. Lee, "Place recognition for visual loop-closures using similarities of object graphs," *Electronics Letters*, vol. 51, no. 1, pp. 44–46, 2014.
- [15] E. Pepperell, P. Corke, and M. Milford, "Routed roads: Probabilistic vision-based place recognition for changing conditions, split streets and varied viewpoints," *The International Journal of Robotics Research*, 2016.
- [16] E. Stumm, C. Mei, S. Lacroix, and M. Chli, "Location graphs for visual place recognition," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE, 2015, pp. 5475–5480.
- [17] G. Costante, T. A. Ciarfuglia, P. Valigi, and 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.
- [18] C. L. Zitnick and P. Dollár, "Edge boxes: Locating object proposals from edges," in *Computer Vision—ECCV 2014*. Springer, 2014, pp. 391–405.
- [19] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems (NIPS)*, December 2012.
- [20] E. J. Candes and T. Tao, "Near-optimal signal recovery from random projections: Universal encoding strategies?" *Information Theory, IEEE Transactions on*, vol. 52, no. 12, pp. 5406–5425, 2006.
- [21] M. Muja and 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.
- [22] S. Feizi, G. Quon, M. Medard, M. Kellis, and A. Jadbabaie, "Spectral alignment of networks," 2015.
- [23] M. Cummins and P. Newman, "Fab-map: Probabilistic localization and mapping in the space of appearance," *The International Journal of Robotics Research*, vol. 27, no. 6, pp. 647–665, 2008.





**Dipartimento di Ingegneria  
Università degli Studi di Perugia**

**Decreto n. 120/2016**

**Oggetto:**

Approvazione atti, e  
graduatoria di merito per  
il conferimento di n. 1  
borsa di studio procedura  
di selezione comparativa  
D.D. 111/2016 – Resp.  
Prof. Andrea Di Schino

**Il Direttore**

- VISTO** il Regolamento concernente il conferimento di borse di studio per la ricerca e la formazione avanzata, emanato con DR. N. 1527 del 05/07/2005;
- VISTO** il chiarimento interpretativo sull'art.18 c. 5 L. 240/210 espresso dall'Amministrazione Centrale di questo Ateneo con Circolare Prot. 2014/0017480 del 10/06/2014;
- VISTO** il D.L. n. 5/2012, art. 49, comma 1, lettera h), p.5;
- VISTO** il Progetto di Ricerca "Sviluppo di un processo innovativo per la produzione di leghe ad elevata resistenza meccanica", cofinanziato dalla Fondazione Cassa di Risparmio di Terni e Narni, di cui è Responsabile Scientifico il prof. Andrea Di Schino e nell'ambito del quale è previsto il finanziamento di borse di studio (Rif. Richiesta di contributo Dipartimento di Ingegneria del 15/03/2016 e Approvazione richiesta da parte della Fondazione CaRiT del 29/04/2016);
- VISTO** il D.D. n. 103/2016 del 15/11/2016 che autorizza la spesa e l'emissione del bando per l'attribuzione di n. 1 Borsa di Studio Post Lauream dal titolo "**Sviluppo di un processo innovativo per la produzione di leghe ad elevata resistenza meccanica**" per lo svolgimento di attività presso il Dipartimento di Ingegneria;
- VISTO** l'avviso di procedura comparativa D.D. n. 111/2016 pubblicato in data 29/11/2016;
- ESAMINATI** i verbali della riunione della Commissione giudicatrice redatti in data 16/12/2016;
- VERIFICATA** la regolarità della procedura,

**DECRETA**

**Art. 1** – Sono approvati gli atti della procedura di valutazione comparativa D.D. n. 111/2016, per il conferimento di una borsa di studio, per l'espletamento di attività presso il Dipartimento di Ingegneria, della durata e per l'importo ivi indicati;

**Art. 2** – E' approvata la seguente graduatoria di idoneità della procedura di valutazione comparativa di cui all'art. 1 del presente decreto:

**1^ - NAPOLI GIUSEPPE (85/100)**

**Art. 3** – E' dichiarato assegnatario della selezione di cui all'art. 1 del presente decreto il **Dott. NAPOLI GIUSEPPE** a cui si conferisce la borsa di studio oggetto della sopra richiamata procedura comparativa.

Il presente decreto sarà portato a ratifica del prossimo Consiglio di Dipartimento.

Perugia, 19/12/2016



I Direttore  
**Prof. Giuseppe Saccomandi**



Allegato N. ....1..... al punto  
dell'ordine del giorno N. ....8.....

**UNIVERSITA DEGLI STUDI DI PERUGIA**  
**Dipartimento di Ingegneria**

**Oggetto:**  
**Trasferimento**  
**Responsabilità**  
**Scientifica** su  
**progetti di ricerca**  
**del Dott. Marco**  
**Ricci**

**Decreto n. 121 del 22/12/2016**  
**IL DIRETTORE**

Il direttore del DING

**VISTI** gli artt. 41 dello Statuto e 94 del Regolamento Generale di Ateneo;  
**CONSIDERATO** l'art. 10 del Regolamento di funzionamento del Dipartimento di Ingegneria;

**VISTO** La richiesta inoltrata in data 22/12/2016 dal Dott. Marco Ricci, nella quale si richiede di trasferire la responsabilità scientifica del proprio progetto "NDTonAIR" H2020-MSCA-ITN-2016 al prof. Piero Burrascano, mentre per la convenzione di ricerca applicata in essere con la "Società delle Fucine" al Dott. Luca Senni ;

**CONSIDERATO che** dal 30/12/2016 il Dott. Marco Ricci prenderà servizio presso l'Università della Calabria;

**VISTA** pertanto l'urgenza di procedere l'iter degli assegni di ricerca banditi nell'ambito del progetto "NDTonAIR" ;

**DECRETA**

- A) - Di procedere alla variazione della responsabilità scientifica del progetto "NDTonAIR" H2020-MSCA-ITN-2016 individuando nella persona del prof. Pietro Burrascano quale nuovo responsabile;
- B) Di procedere alla variazione della responsabilità scientifica della convenzione di ricerca applicata con la "Società delle Fucine" individuando nella persona del Dott. Luca Senni quale nuovo responsabile;

Il presente decreto sarà sottoposto a ratifica del prossimo Consiglio di Dipartimento.

Perugia, 12/12/2016

  
Il Direttore  
(prof. Giuseppe Saccomandi)



IL DIRETTORE DEL DIPARTIMENTO DI INGEGNERIA

D.D. n. 1/2017

Oggetto:  
Approvazione  
Progetti  
Bando di Ricerca  
2017  
Fondazione Cassa di  
Risparmio di Perugia  
Il direttore del DI

VISTA la pubblicazione del Bando 2017 per la richiesta di finanziamento per progetti di ricerca emanato dalla Fondazione Cassa di Risparmio di Perugia;

VISTI i progetti pervenuti al Dipartimento di Ingegneria dell'Università degli Studi di Perugia:

- OPTO WIND - Metodi innovativi per diagnosi precoce di guasti su macchine eoliche e ottimizzazione della vita a fatica dei componenti" - referente Prof. Francesco Castellani;

- Studio dell'utilizzo di combustibili non convenzionali per lo sviluppo di propulsori a basso impatto ambientale" - referente Ing. Michele Battistoni;

CONSIDERATO che i progetti di ricerca devono essere presentati alla Fondazione Cassa di Risparmio di Perugia entro il termine del 10 Gennaio 2017;

VISTA pertanto l'urgenza di approvare i progetti di ricerca con i relativi piani finanziari con Decreto in quanto entro la data del 10 Gennaio 2017 non sono previste sedute di Consiglio di Dipartimento;

**DECRETA**

di approvare i seguenti progetti di ricerca con i relativi piani finanziari:

OPTO WIND - Metodi innovativi per diagnosi precoce di guasti su macchine eoliche e ottimizzazione della vita a fatica dei componenti - referente Prof. Francesco Castellani, per una richiesta di finanziamento pari ad € 72.651,22 di cui autofinanziamento per € 29.151,26;

- Studio dell'utilizzo di combustibili non convenzionali per lo sviluppo di propulsori a basso impatto ambientale - referente Ing. Michele Battistoni per una richiesta di finanziamento pari ad € 25.000,00 di cui autofinanziamento per € 10.000,00;

di impegnarsi a sostenere tutti gli eventuali oneri non previsti nei progetti.

Il presente decreto sarà sottoposto a ratifica del prossimo Consiglio di Dipartimento.

Perugia, 10/01/2017

Il Direttore  
(Prof. Giuseppe Saccomandi)





Decreto n.6

Master di II livello in "PRO Gettare SMART CITIES Architettura, Bulding Simulation, Energia, Mobilità ICT" - art.3 del Regolamento Didattico: rettifica

**II DIRETTORE**

- Vista la delibera di approvazione della II edizione del Master Universitario di II livello in "PRO Gettare SMART CITIES Architettura, Bulding Simulation, Energia, Mobilità ICT" del consiglio del Dipartimento di Ingegneria del 13.12.2016;
- Visto il Regolamento didattico del Master Universitario di II livello in "PRO Gettare SMART CITIES Architettura, Bulding Simulation, Energia, Mobilità ICT" e il relativo Progetto di Corso;
- Considerato che, per mero errore materiale, all'art.3 del Regolamento didattico del Master Universitario di II livello in "PRO Gettare SMART CITIES Architettura, Bulding Simulation, Energia, Mobilità ICT" è stata indicata la quota d'iscrizione pari a euro 3.000 anziché 3.500, come peraltro previsto nel piano finanziario del Progetto di Corso;
- Ritenuta la propria competenza;

**DECRETA**

di approvare la rettifica, all'art.3 del Regolamento didattico del Master Universitario di II livello in "PRO Gettare SMART CITIES Architettura, Bulding Simulation, Energia, Mobilità ICT", relativa alla quota d'iscrizione pari a euro 3.500.

Il presente decreto sarà portato a ratifica del prossimo consiglio di dipartimento.



IL DIRETTORE  
(Prof. Giuseppe Saccomandi)