



*SAFERA project no. 22*

**Smart PROcess INdustry CranEs**

This  
SAFERA project  
has received funding  
from INAIL (Italy) and  
MESTD (Serbia)

Smart PROcess INdustry CranEs

(project acronym SPRINCE)

# Visual Guidance System Development

Report activity WP1.1



This project is supported by the consortium ERA-NET SAFERA (Coordination of European Research on Industrial Safety towards Smart and Sustainable Growth under the Seventh Framework Programme for Research and Technological Development). Funding institutions are INAIL and MESTD

## Notes

With respect to the version of October 2015, this report includes the experimental activity of the prototype, which took place at the Thermoelectric Power Plant of San Filippo del Mela (Messina) - Company a2a Energie Future.



**University of Messina**



**University of Belgrade**



**University of Defence**

## **DELIVERABLE no. D01-SPRINCE**

*Deliverable number:* D01-SPRINCE

*Title:* Visual guidance system development

*Responsible:* University of Messina (UM)

*Active partners:* University of Messina (UM)

*Delivery dates:* 30<sup>th</sup> September 2015

Document version: 28<sup>th</sup> February 2017

## Table of Contents

Research Team Composition .....	3
1. Objectives .....	3
2. Description of performed activities .....	4
2.1. Development of the Visual Guidance System: basic algorithms .....	4
2.2. Development of the image acquisition system .....	7
2.2.1. Hardware: main characteristics .....	8
2.2.2. Installation of the prototype .....	11
2.2.3. Installation of the prototype on a crane Liebherr LTM 1200-5.1 .....	12
2.2.4. Installation of the prototype on a crane Ivo Lola Ribar KP5/D-0313 .....	15
3. Results .....	17
3.1. Laboratory tests .....	19
3.2. Field tests .....	21
4. Major encountered problems and corrective actions .....	28
4.1. Software tests .....	31
5. Deviations from the work plan .....	33
6. Produced publications .....	33
References .....	34
Annex 1 - Questionnaire for the Company .....	36
Annex 2 – Data collected from Companies .....	41
Annex 3 - Use of the VGS - Tutorial .....	44
Annex 4 - Configuration of Raspberry - Tutorial .....	53

## Research Team Composition

Team component	Current position	Role in the activity	Institution
Milazzo Maria Francesca	Assistant Professor	Research Coordinator & Project Leader	UM
Consolo Giancarlo	Assistant Professor	Team Member	UM
Ancione Giuseppa	SAF€RA Post-Doc Researcher	Team Member	UM
Kavassidis Isaak	SAF€RA Fellowship Researcher	Team Member	UM
Merlino Giovanni	PhD student	Team Member	UM
Scionti Giuseppe	SAF€RA Fellowship Researcher	Team Member	UM

### 1. Objectives

Safety is a fundamental issue during heavy duty operations in several contexts, e.g. industrial and construction sites, etc. In these environments, particular attention should be paid to the protection of operators working with cranes.

The literature shows that cranes are the most dangerous equipment [1,2]. A tipped, dropped or mishandled load can directly injure workers or potentially upset the equipment. Moreover, a significant number of accidents are due to the hindered view of the working area for the crane-operator. Their dangerousness has special relevance in the chemical process industry and the intermodal transport, where accidental events could cause the release of hazardous substances, leading to fires, explosions or toxic dispersions.

In this frame, a real-time computer-aided visual guidance application has been developed based on the idea that crane accidents, caused by obstructed view, can be avoided. The objective of this deliverable is to describe such an application, named *Visual Guidance System* (VGS); it allows predicting whether a dangerous event is going to occur and promptly alerting the crane-operator in order to let her/him taking corrective actions during the execution of crane-assisted shifting duties.

This application can be executed by using a system-prototype, which was designed to offer high execution speed, ease of integration, low cost, low power consumption, low computer memory, good support to navigate the load in the right position, ergonomics and scalability.

## 2. Description of performed activities

The workpackage *WPI* of the SPRINCE project (Smart PROcess INdustrial CranE) concerns to the “Implementation of a real-time object detection solution for industrial cranes” and includes two activities, the “Visual guidance system development” (*WPI.1*) and the “Implementation of the solution on different cranes” located in different industrial contexts and by using different scalability aspects (*WPI.2*). This deliverable reports on the investigation efforts from which a first version of the VGS was designed, implemented and tested (activity *WPI.1*).

The VGS is expected to be realised as a real-time system dealing with different tracking systems and incorporating both a visual and an audio feedback. It will be integrated into industrial cranes and prevent accidents due to obstructed view. The proposed system utilises a colour web-camera in order (i) to obtain the position of the hook (spreader bar) relative to the cabin and of the obstacle relative to the environment, (ii) to compute the distance between them and (iii) to create guidance commands to the operator for the feedback activation. Real-time position measurements obtained from the camera are displayed as a video on a LCD display monitor that is also used to create the feedback to the operator.

The VGS, developed from this activity, will be improved during the following activity *WPI.2* by a testing phase, executed by the project partners *UM* (University of Messina) and *FME-UB* (Faculty of Mechanical Engineering – University of Belgrade).

*UM* is the responsible and the only active partner in the activity *WPI.1*, which has been completed by executing the tasks listed below:

- *T1.1.1 - Object recognition and AR experience implementation*
- *T1.1.2 - Extensions for mesh-based cooperation*
- *T1.1.3 - Cloud-based storage subsystem*

### 2.1. Development of the Visual Guidance System: basic algorithms

Given that the VGS is responsible to inform the crane-operator about something that is trespassing the working area, the motion detection system must be able to identify moving objects in the monitored scene and emit a clear and unambiguous notification to alert the operator.

The area surrounding the moving object is named *background* [3]. As underlined in the literature, the acquisition of a background image, which does not include any moving object, is the most elementary way to model the environment [4]. Simple methods, named *background modelling* for a simultaneous tracking object movement, allow the real-time visual modelling by means of the following steps: (i) modelling the background, (ii) acquiring several images (iii) subtracting each new image from the model of the

background scene and (iv) determining the foreground pixels from the resulting difference image. While in ideal settings (laboratory), motion detection algorithms, based on *background modelling* methods, have good performance, the detection process can be altered/influenced by a large number of parameters in open and unrestricted environments. In fact, false results could be generated due to illumination/weather changes, periodic objects motion in the working area, undesired camera movements etc.

The implementation of the module has to be as much accurate as possible to avoid false positive and negative indications, which could hamper the image acquisition process and endanger human lives. A false negative result occurs when a trespasser is moving inside the monitored area but the algorithm failed in reporting it. This could have devastating consequences because the trespasser could put him/her in a dangerous situation, which may be avoided if it is detected. On the contrary, a false positive result, i.e. the algorithm reports an event but it does not occur, determines that the operator should interrupt his work to control the area and ensures that the operation can be safely continued.

*Statistical background modelling* approaches, compared to the standard *background modelling* ones, usually create a model of the scene built on a longer history of previous frames (statistical model). Based on this statistical model, a pixel in the current frame is classified as belonging to the foreground or background [5]. The ability to maintain more information about the past of the scene makes these approaches more robust in real life applications, but at the same time they require a more complex implementation and additional computational power. In general, there is an initialisation phase, when the first image of a video sequence is used to generate the model; then for each successive frame, each pixel is checked if it belongs to the statistical model of the scene. If there is enough evidence that the model supports the current pixel, then it is characterised as background; on the contrary, if the pixels of the current frame are not supported by sufficient data in the background model, they are considered as moving objects.

The state-of-the-art related to this topic shows that an algorithm that combines all the above characteristics is ViBe [6]. It allows achieving low false positive and negative rates by using very few computational resources to be run in real-time. Other statistical background modelling algorithms are GMM [7] and APMM [8], in which the background is modelled respectively through Gaussian and Poisson distributions, and finally Codebook [9] uses a dictionary of pixel values for a pixel classification.

Within the SPRINCE project, the module has been developed by using the OpenCV programming language and the Qt libraries. OpenCV has been chosen for the following two reasons: (i) it allows creating a great variety of image processing algorithms out-of-the-box and (ii) it contains an open-source version of the ViBe algorithm, which can be freely implemented for research purposes even if it is protected by a patent. The Qt libraries have been used in order to ensure portability between different operating systems. The architecture of a previously developed system is shown in Figure 1. A camera is

positioned within the working area of the crane, the video stream of the area is continuously processed by the *statistical background modelling* algorithm and, in case a moving object is detected in the scene, the operator is warned about its presence.



Figure 1. Architecture of the developed system.

In the context of the SPRINCE project, the system for the object detection will be installed on a moving machine and, given that the aim is not to track obstacles but to figure out if, at any time, one or more of them is close to the load, further considerations have been done to account for this additional complexity in the final version of the VGS. Firstly, to give the motion detection system also the ability to accurately estimate the real dimensions of the item, beyond the distance between a moving object and the camera, *stereoscopic video acquisition* methods have been investigated. If two cameras acquire an image of the same object, each of them is slightly different, thus the stereoscopic acquisition fuses the two images and induces the depth perception [10]; the process simulates the human's stereo vision. Then, the *superpixel segmentation technique* has been used for the 3D reconstruction of the scene [11]. *Image segmentation* is the process of partitioning of a digital image into multiple small patches (sets of pixels, also known as superpixels). The goal of segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyse. The pixels, which are included in a superpixel, are characterised by similar properties such as colour, intensity, or texture.

A stereoscopic image acquisition can be realised by adopting two cameras having the same focal length and optical properties to shoot a same scene from slightly different directions [12]. In this case, the *maximum observing distance* ( $z$ ) is the maximum distance that can be observed by a pair of cameras. Figure 2 shows the dependence of the *maximum observing distance* on the parameters of a stereoscopic vision system. The line  $B$  (*baseline*) denotes the distance between the central points of the cameras' sensors ( $O$  and  $O'$ ),  $X$  is the



observed object,  $z$  denotes the distance of  $X$  from the centre of the line  $B$  and  $F$  and  $F'$  are the focal lengths of the cameras.

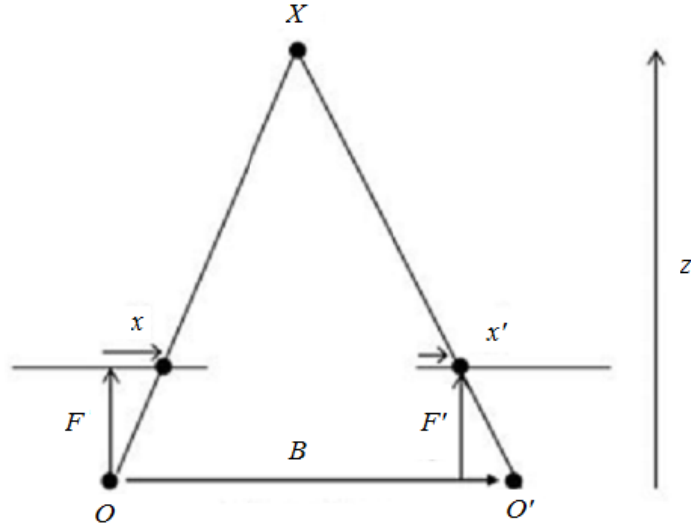


Figure 2. Dependence of the maximum observing distance on the parameters of a stereoscopic vision system.

With respect to Figure 2, the *maximum observable distance* is given by the following formula:

$$z = \frac{B \cdot F}{N \cdot s} \quad (5)$$

where:  $B$  = baseline length;  $N$  = maximum disparity value given by  $|x-x'|$  (difference between the horizontal coordinates of the observed point from the two single cameras with the respect fixation point);  $s$  = pixel size (characteristic of the image sensor).

As discussed above the acquisition of stereoscopic images allows calculating the effective distance between various objects in the three dimensional space. These measurements were used by the processing computer to understand whether the distance of the load from an obstacle gets lower than a predetermined threshold.

## 2.2. Development of the image acquisition system

The designed prototype for the image acquisition is a simple system composed by hardware and software. The software is the VGS application, which was developed based on the algorithms described in Section 2.1.

### 2.2.1. Hardware: main characteristics

The hardware includes a camera and a computer as main elements. The characteristics have been defined by taking into account the need to offer high execution speed, ease of integration, low cost, low power consumption, low computer memory, good support to navigate the load in the right position, ergonomics and scalability.

The development of the application and the design of the prototype are strictly connected. Thus, before the implementation of the system, the following issues have been discussed with the Companies where the solution will be implemented:

1. Potential limits associated with the type/manufacturer/model of cranes.
2. Potential limits in making changes to the force feedback through the joystick, if it is already implemented by the manufacturer.
3. Characteristics of the camera.

A questionnaire has been prepared by *UM* (Annex 1) and answered by Italian and Serbian Companies to mainly clarify the points 1 and 2 (see collected data Annex 2). Concerning the third point, the preference towards an OpenNI type (with advanced functions to capture 3D image) has been expressed by the expert in Informatics. A video taken from the crane has been required, this clip has to reproduce conditions similar to those that will be studied, i.e. the load motion with the entry in the field of view of different foreign elements. It will serve to calibrate the software and test the results, besides it could be also employed as a demo.

Given the higher cost of a stereoscopic camera, compared to the available budget for the acquisition of the complete kit to assemble the prototype for the real-time object vision detection, the use of two cameras to realise the stereoscopic vision has been preferred. This choice requires that cameras have the same focal length and optical properties to shoot an equivalent scene from slightly different positions. An economic appraisal allowed verifying the cheapness of this solution.

The main elements of the hardware are the following:

- no. 2 Wi-Fi cameras OpenNI type
- no. 1 Remote control
- no. 2 Portable power pack 10400 mAh
- no. 2 MicroSDXC Memory Card 64 GB
- no. 1 Computer (see Table 1)
- no. 2 Raspberry pi 3

Taking into account the characteristics of the application to be developed, which will be described in the following Section, the minimum hardware requirements to implement and execute the application are given in Table 1.

Table 1. Minimum hardware requirements

Component	Minimum characteristics
Motherboard	Gigabyte GA-H81N-D2H
CPU	Intel i5 4460
RAM	4 GB DDR3
Case 1 (equipped with a dissipater)	Inter-Tech SY-500
Power Supply	Generic power supply (external switching) 120W
HDD 1	USB Flash Drive / SD Card (32 GB)
HDD 2	HDD 500GB / SDD 250GB

The main elements of the prototype are placed in a box and connected as given in Figure 3. Two cameras are fixed at a given distance and connected by USB cables to two power banks. Both cameras emit a Wi-Fi signal, which will be received through two Raspberry pi 3 and transmitted to the computer. The computer will elaborate the video and, by mean of the VGS, will produce the feedback to the crane-operator when the distance load-obstacle reaches a threshold. As previous pointed, the *maximum observable distance* with this system depends on the distance between the central points of the sensors of the cameras, the focal lengths, the pixel size of the image sensor and the resolution of the camera.

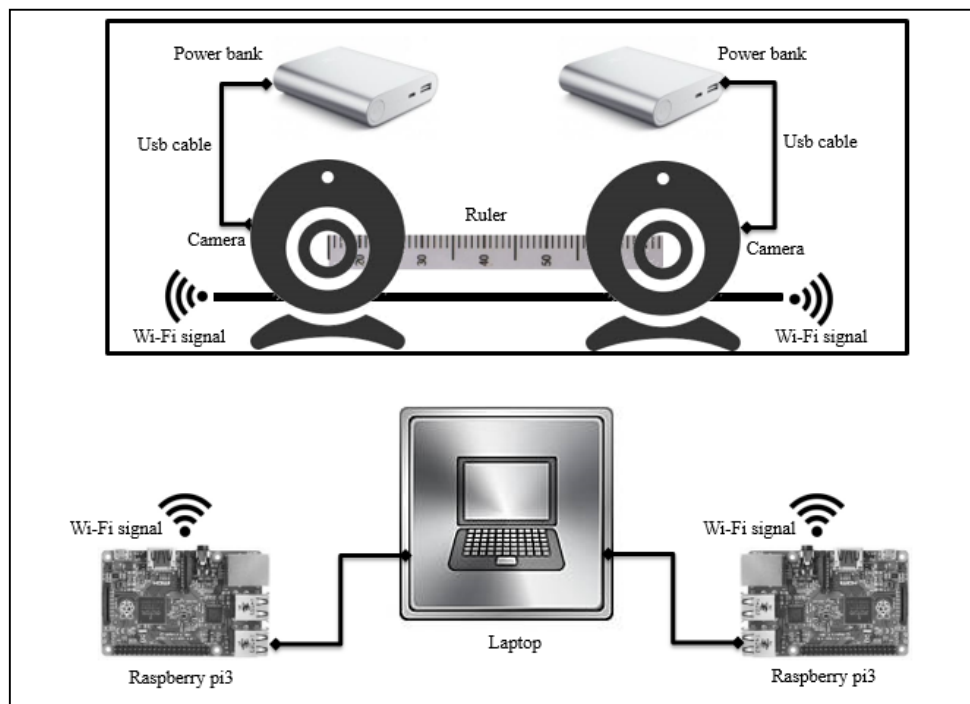
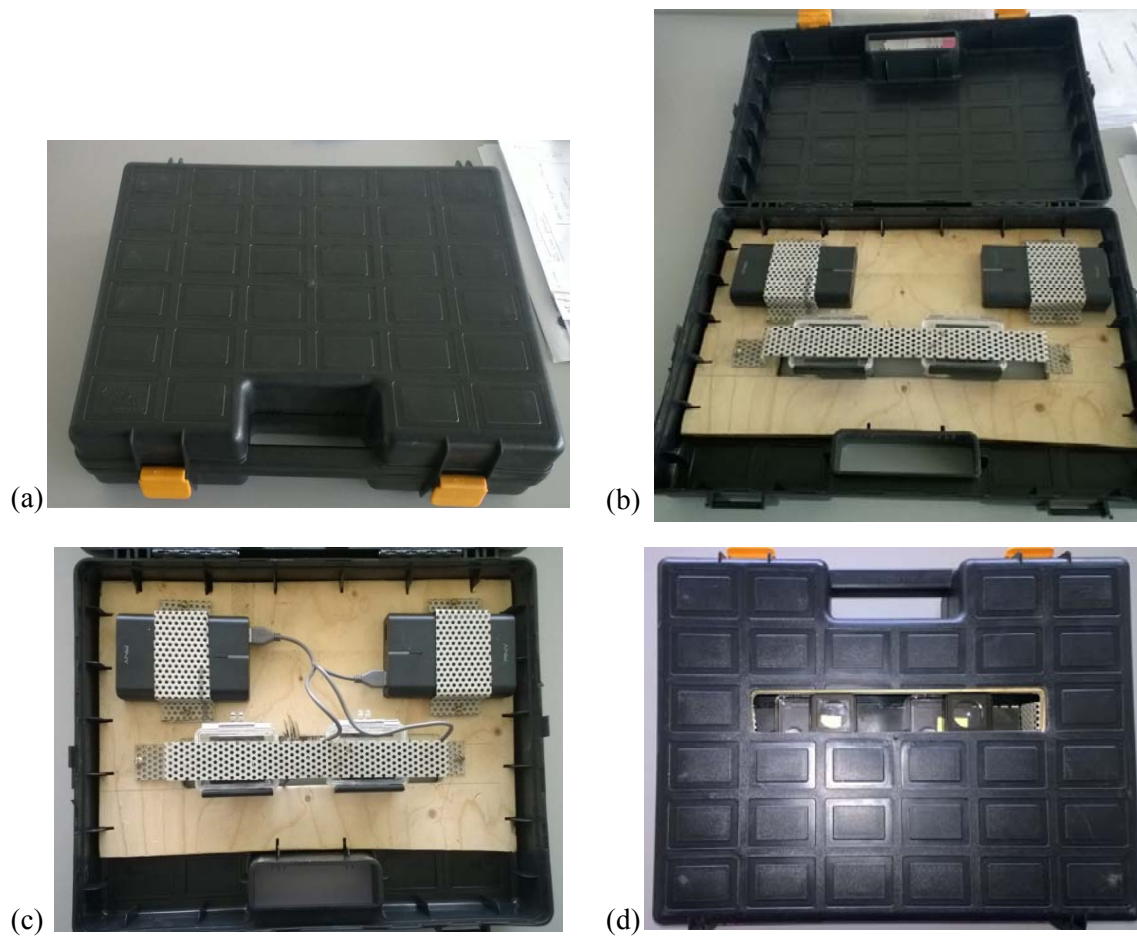


Figure 3. Prototype camera-computer.

The elements of Figure 3 are placed in a box, except the two Raspberry and the laptop; this latter element is the interface of the crane-operator with the image detection system. Figure 4 shows some pictures of such a box, in particular: (a) the back of the closed box; (b) the opened box; (c) the inside of the box, i.e. the cameras, the power banks and related connections; (d) the front of the closed box and the opening from with the image is acquired from the cameras.



*Figure 4. Box containing cameras and two Raspberry.*

### 2.2.2. Installation of the prototype

The location of the prototype has been discussed and chosen by taking into account the following: (i) the need to simplify the motion detection algorithm implementation, without compromising the accuracy of the detection process and, thus, avoiding false and negative results and (ii) the characteristics of the crane.

After a brainstorming, the following considerations have been done:

- (i) The camera-detected coverage, offered by a single static camera positioned at a given distance from the crane, is limited.
- (ii) A wider camera-detected coverage could be obtained by positioning the camera in front of the hook, anyway this would mean that the other half working area is not covered and, thus, it is not monitored at all. This has been deemed unacceptable from the safety point of view.
- (iii) A possible solution could be to deploy more cameras in the surrounding the crane, to cover as much as possible the working area, but this makes the applicability of the VGS dependent on a number of additional factors, e.g. time for the image elaboration, complexity of the image elaboration, particularities identified in the different camera-detected areas, analysis of the dislocation of the cameras to ensure the best coverage, etc. Moreover the acquisition of several cameras makes the solution not economically convenient.
- (iv) Another solution could be that the camera is to locate close to the hook and follow its movement. As discussed in Section 2.1, the movement of the crane, during the lifting operations, is a problem when acquiring images. Due to the movement, the previously implemented *background modelling* algorithm (used for static cameras) would need also of an object recognition algorithm. Moreover, given that cranes could operate in various environments, which may be substantially different, the object recognition algorithm should be trained each time, in order to be adapted to the different working areas. To overcome these difficulties, the previous system based on the *background modelling* approach has been abandoned and substituted by a *statistical background modelling* method.

Based on these considerations, the prototype location is strongly conditioned by the type of crane. In any case, the subsequent settings must be guaranteed:

1. the most extended visibility of the crane's working area
2. a visibility focused on the moving load
3. the adaptability to different cranes and environments
4. a comfortable use of the system (hardware-software) from the crane-operator

Within the activities of the SPRINCE project, two cranes have been analysed:

- a *mobile crane* (manufacturer: LIEBHERR; model: LTM 1200. 5.1)
- an *overhead crane* (manufacturer: Ivo Lola Ribar; model: KP5/D-0313).

### 2.2.3. Installation of the prototype on a crane Liebherr LTM 1200-5.1

A *mobile crane* is a *cable-controlled crane*, mounted on crawlers or rubber-tired carriers (trucks). These are designed to easily transport the crane to a site and use with different types of load and cargo with little or no setup or assembly.

The crane Liebherr LTM 1200-5.1 has a telescopic boom (the longest on the market) and a lattice jib (Figure 5). The telescopic boom includes 7 sections, with a complete extent of 72 m; it can furthermore be extended with a 7 m lattice jib and a swing-away jib up to 36 m of length. Table 2 gives the main characteristics of the crane. The crane can be lifted up at a maximum angle of  $82^\circ$  with respect to the vehicle and its shaft can be elongated at up to 90 m (Figure 6). By using simple trigonometry formulas, the highest position of the hook with the respect to the ground has been calculated; it reaches about 88 m (Figure 7).

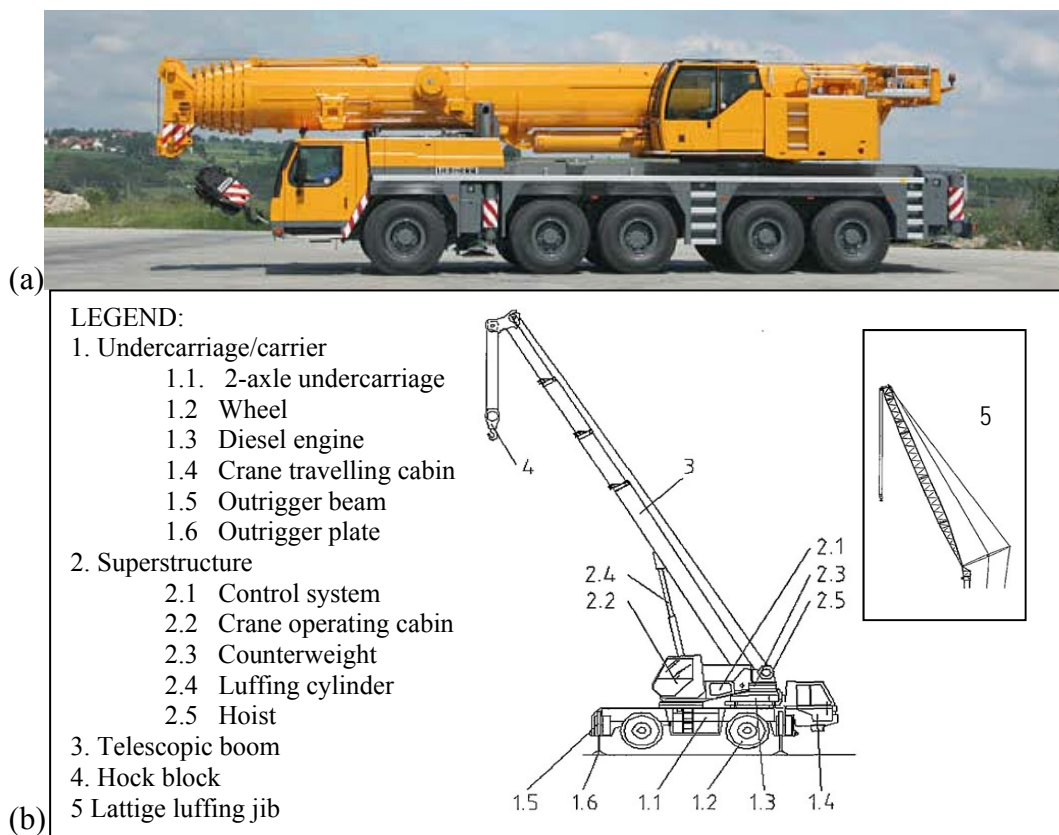


Figure 5. Mobile crane: (a) crane Liebherr LTM 1200-5.1 (5-axle undercarriage); (b) main parts of telescopic cranes (2-axle undercarriage) [13].



Table 2. Technical characteristics of the crane Liebherr LTM 1200-5.1.

Parameter	Value
Model	LTM 1200-5.1 all-terrain
Maximum lifting capacity	200 t at 3 m radius
Telescopic boom	13.2 ÷ 72 m
Lattice jib	7 m ÷ 36 m
Carrier engine/output	Liebherr, 6-cylinder, turbo-Diesel, 370 kW
Crane engine/output	Liebherr, 4-cylinder, turbo-Diesel, 145 kW
Travel speed	80 km/h
Total counterweight	74 t

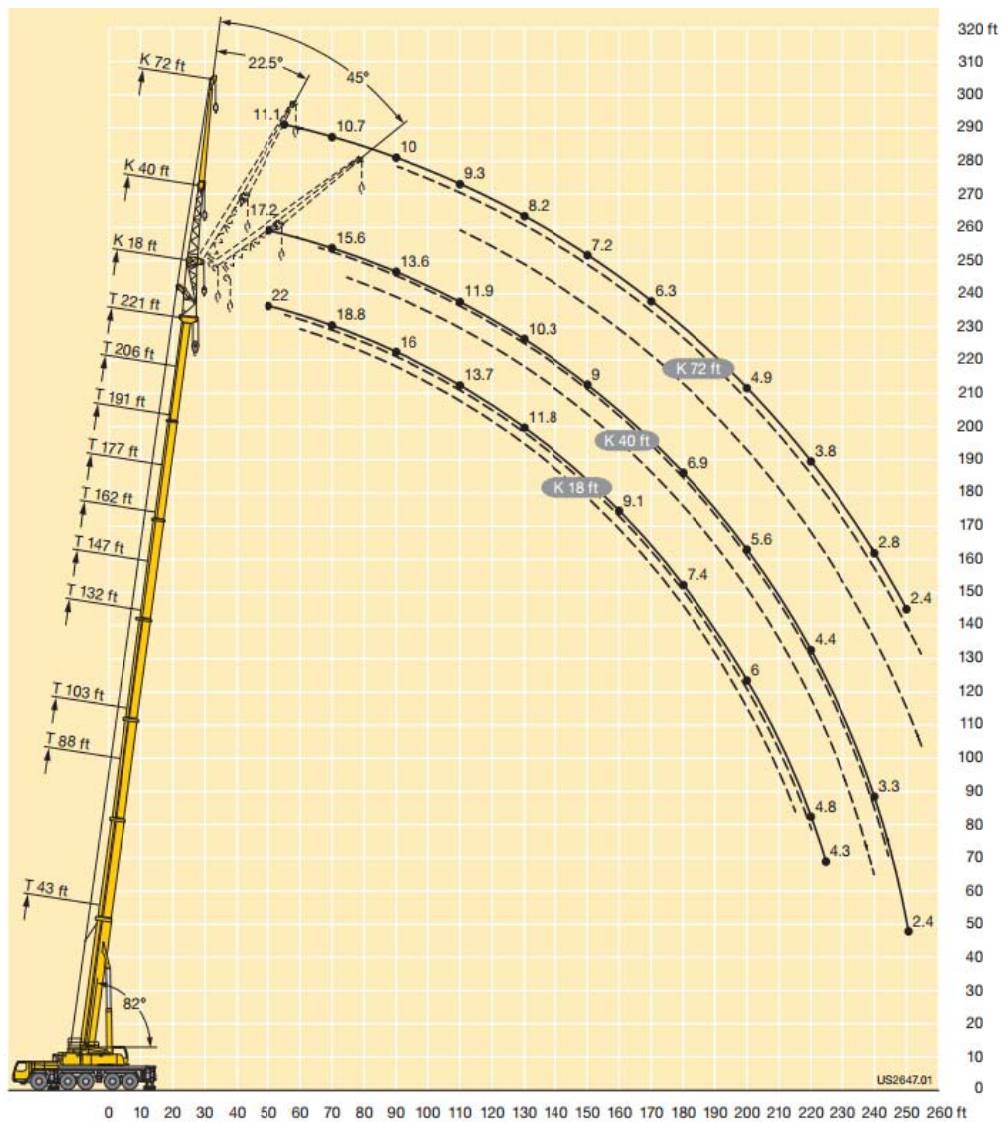


Figure 6. Maximum extension of the telescopic boom for the crane Liebherr LTM 1200-5.1.

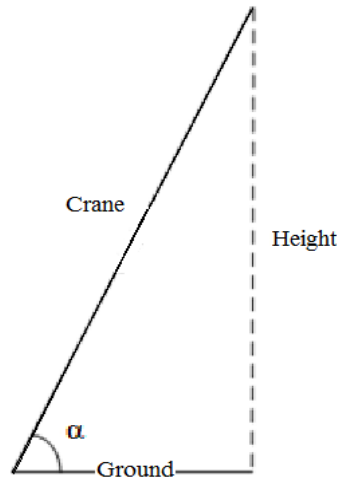


Figure 7. Calculation of the maximum height of the hook.

The prototype (image acquisition system) is going to be installed on the top of the crane (Figure 8). A *maximum observing distance* equal to 100 m has been set to make conservative assessments. The distance determines the parameters of the stereoscopic vision system which will be realised.

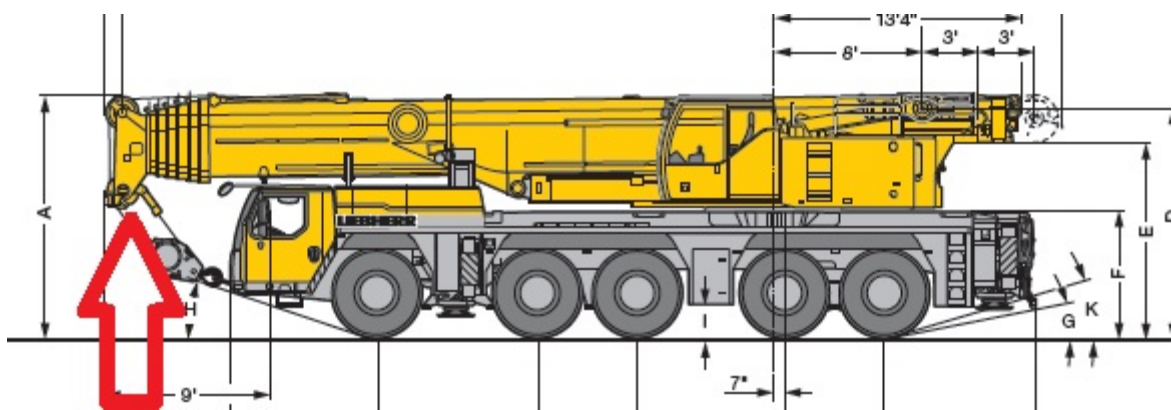


Figure 8. Prototype positioning on the crane Liebherr LTM 1200-5.1.



### 2.3.2. Installation of the prototype on a crane Ivo Lola Ribar KP5/D-0313

An *overhead crane* (commonly called *bridge crane*) consists of parallel runways with a travelling bridge spanning the gap. A hoist (the lifting component of the crane) travels along the bridge. Unlike other types, overhead cranes are typically used for either manufacturing or maintenance applications, where efficiency or downtime is critical factors.

As shown in Figure 9(a), the crane Ivo Lola Ribar KP5/D-0313 is a *cab-operated crane*. Table 3 lists its main characteristics.

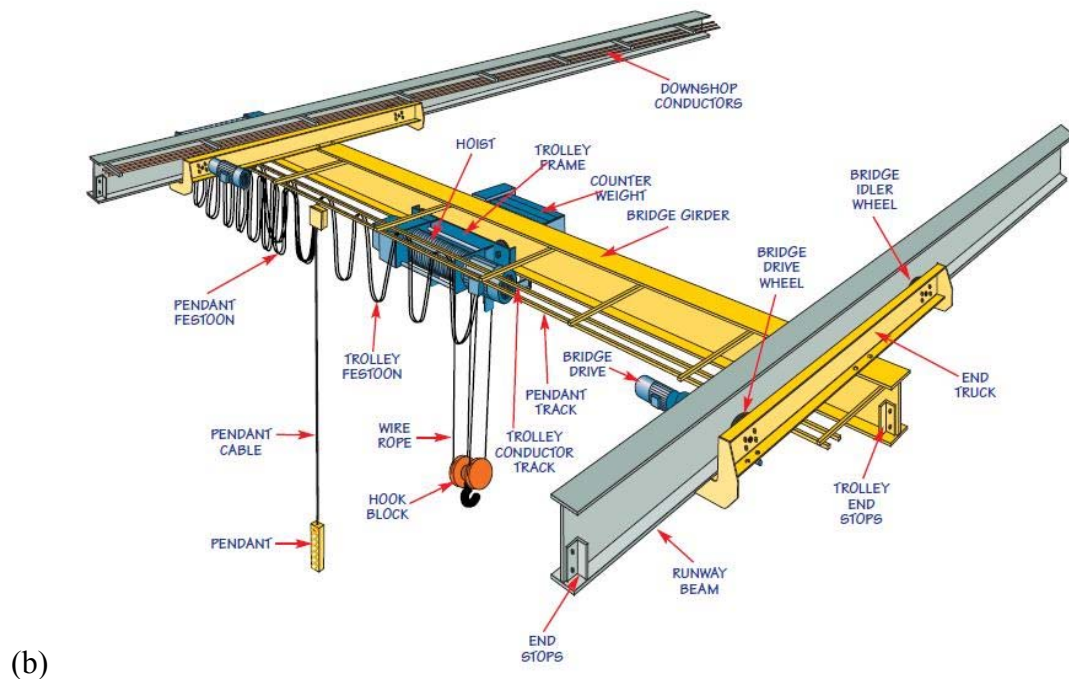
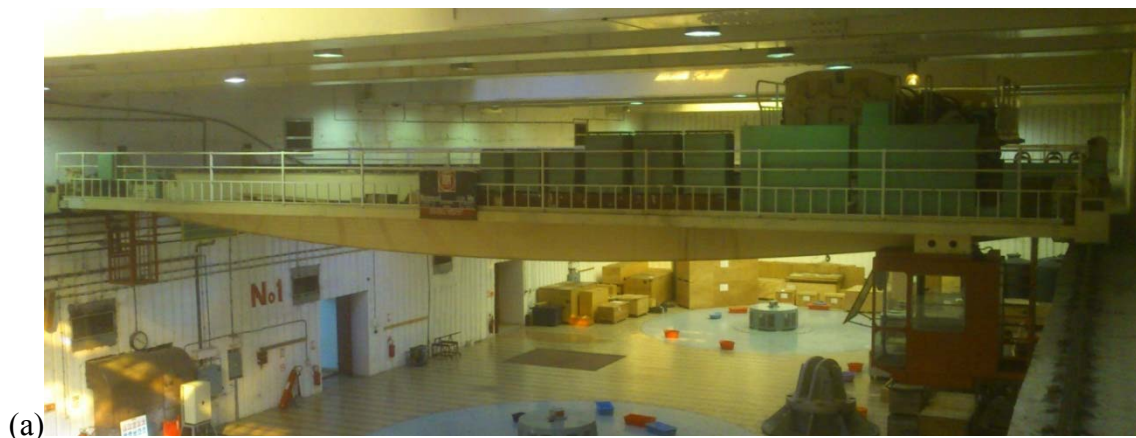


Figure 9. Overhead crane: (a) crane Ivo Lola Ribar KP5/D-0313; (b) main parts of an overhead crane.

Table 3. Technical characteristics of the crane Ivo Lola Ribar KP5/D-0313.

Parameter	Value
Structure group	DIN H2B3
Capacity	63/20 t
Span*	22 m
Lifting height	25 m
Lifting speed main hook	0.5÷5 m/min
Auxiliary hook	6.3 m/min
Trolley speed	16 m/min
Bridge speed	25 m/min

\*Distance between the rail centers from left side 1 to right side 2.

The prototype (image acquisition system) is going to be installed close to the hook (Figure 10) and will have an integral motion with the trolley. Its *maximum observing distance* is about 20 m.

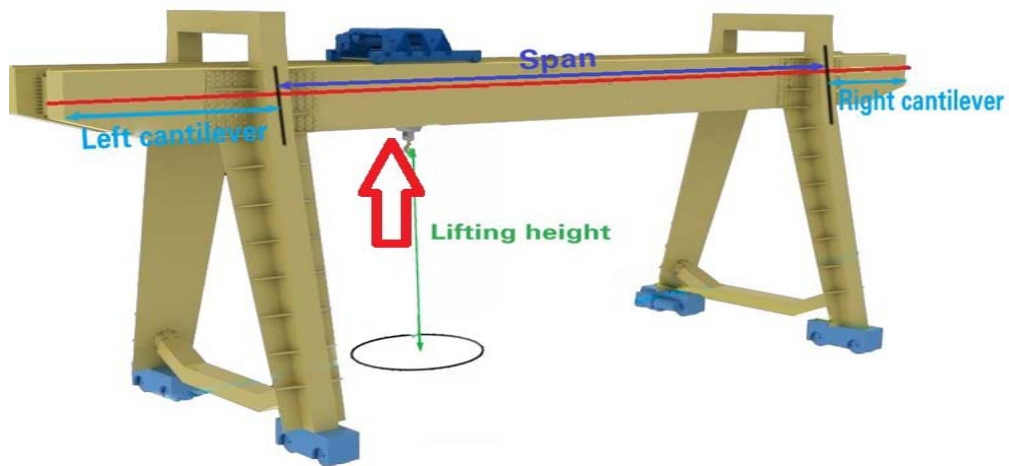


Figure 10. Prototype positioning on the crane Ivo Lola Ribar KP5/D-0313.

### 3. Results

The product of this activity is the Visual Guidance System (VGS), which can be run once the image acquisition system is positioned on the crane. Figure 11 shows the architecture of the final version of the system. In this case a stereoscopic camera is mounted on the top part of a mobile-crane; after the image is acquired, it is processed in order to reconstruct the depth map, which represents the object distances from the camera using different levels of grey colour (values near white, i.e. grey level = 255, mean that the object is close to the camera; values near black, i.e. grey level = 0, mean that the object is far from the camera). Black parts indicate that the area could not be represented (misconfiguration of the camera or long distance from the object).

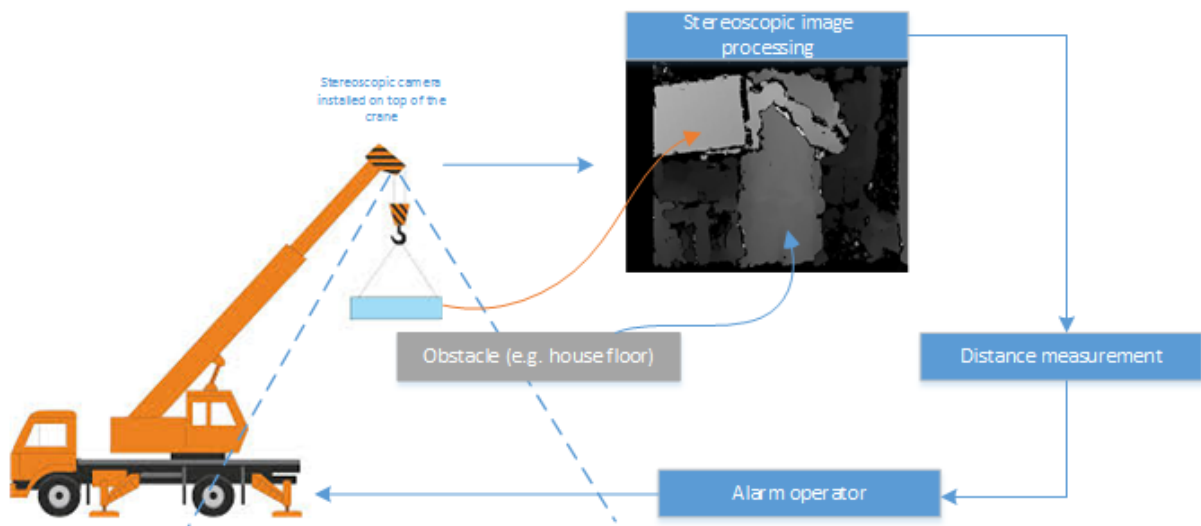


Figure 11. Architecture of the developed system by using stereoscopic images acquisition.

The GUI (Graphic User Interface) of the application is given in Figure 12. The *Start*, *Stop* and *Reset* buttons are respectively used to start, end and reset the monitoring process; The *Beep on intrusion* checkbox enables or disables the acoustic signal alerting that an object is detected; the other two controls are inserted for debugging purposes, but they should not be included in the final version of the product.

If the *Start* button is pressed the system continuously registers the video stream coming from the camera and, if an object that does not belong to the scene enters the view of the camera, the main window of the GUI starts to blink in a red colour (Figure 13), warning the operator about a potential dangerous situation. Figure 14 shows the monitoring window in which the object causing the alert of Figure 13 is visualised. This preliminary test has been executed on a laboratory scale and the object is a pen.

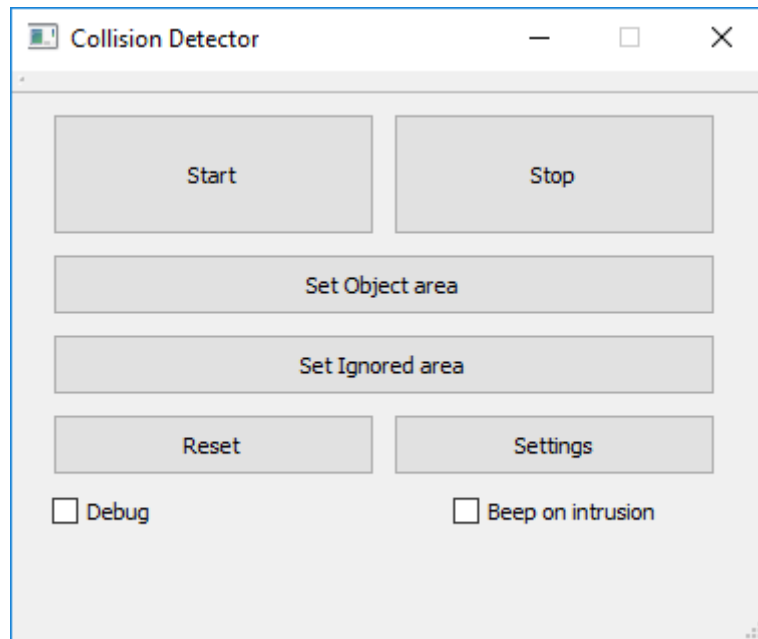


Figure 12. Main GUI of the motion detection system.

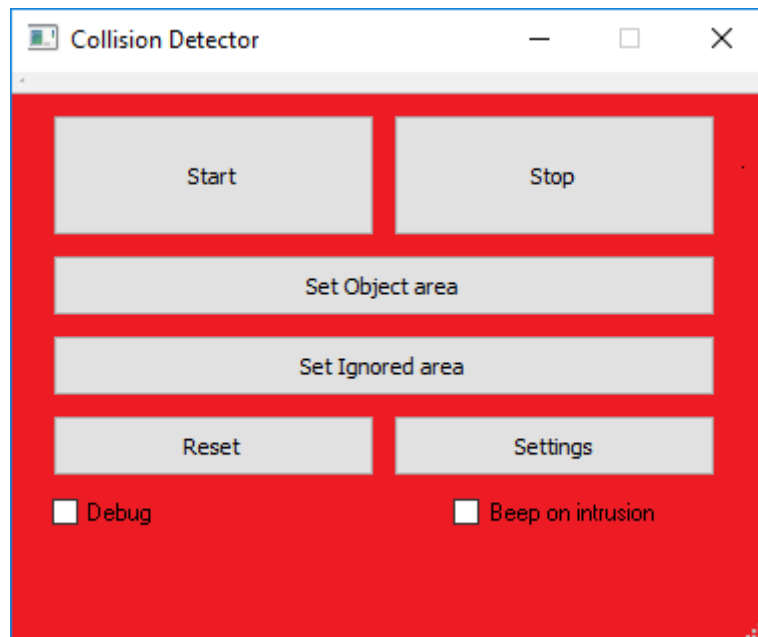
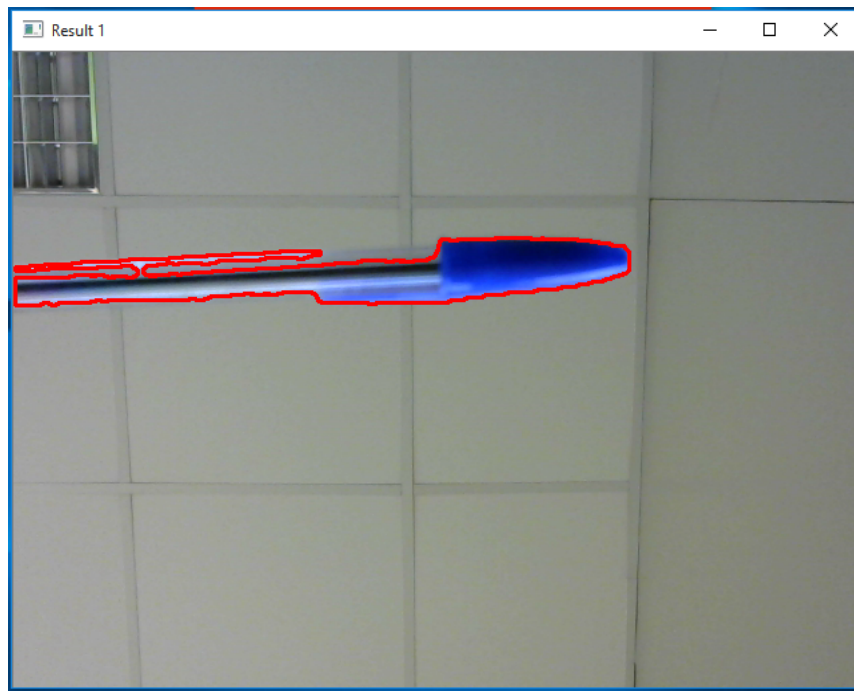


Figure 13. Main GUI of the VGS when an intrusion occurs.



*Figure 14. Monitoring window shows a the object entering in the visual field of the camera (in this case a pen).*

The application was tested on a Desktop PC with an Intel I7-3930K CPU and 16 GB of RAM and also on a laptop Asus Aspire s3-391 with Window 7 Ultimate 64Bit as Operative System, a Intel Core i7-3517U CPU, 4 GB Ram and a graphic card: Intel Graphics 4000. Anyway the VGS can smoothly run with computers having low performance (max. CPU utilization 15%, max occupied RAM 150 Mb).

A tutorial on the use of the VGS for users has been prepared (see Annex 3).

### ***3.1. Laboratory tests***

Some pictures have been taken in the laboratory to show how the VGS works. Figure 15(a) gives the working area. The red rectangle highlights the load and represents the bounders inside which obstacles (moving objects) have not to enter. Figure 15(b) shows the entrance of a potential intrusion in the scene, it is underlined by means of the smaller blue rectangle. Figures 16(a)-(b) show the same images; these are the result of the elaboration made by the VGS. All elements are identified by using a grey colour scale, where lighter shades indicate objects that are closer to the cameras compared to the darkest ones. In Figure 16(a), the rectangle indicates the load position; it is located closer to the cameras compared to any other element present in the scene, as the surrounding has the darkest colours; this means that there are no objects placed at the same distance. Similarly, by considering the  $x$  direction of a two-dimensional plane (2D), the rectangle of Figure 16(b) shows an object

(intrusion) that is close to the previous underlined rectangle. The light colour indicates that the intrusion and the load are located at the same distance from the cameras ( $z$  direction of a 3D representation). Figures 17(a)-(b) show the processing of the images by the *superpixel segmentation technique* to reconstruct the depth provided in Figures 16(a)-(b).

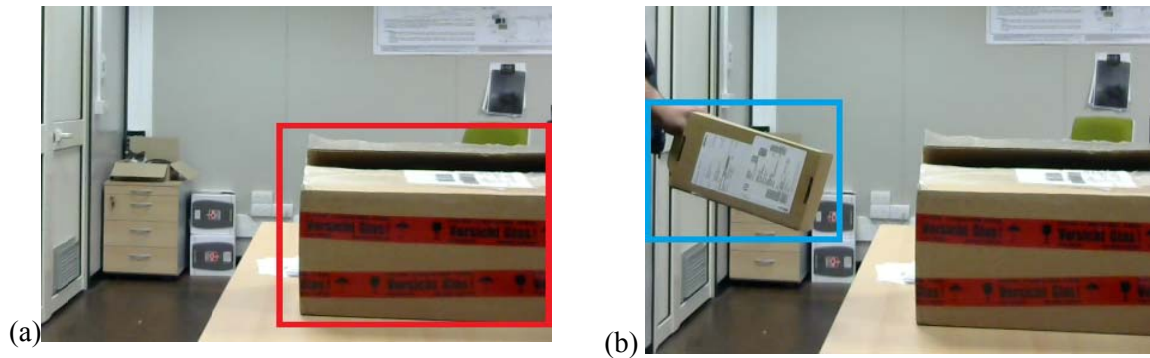


Figure 15. Working area: (a) the rectangle underlines the load; (b) the rectangle underlines an intrusion.

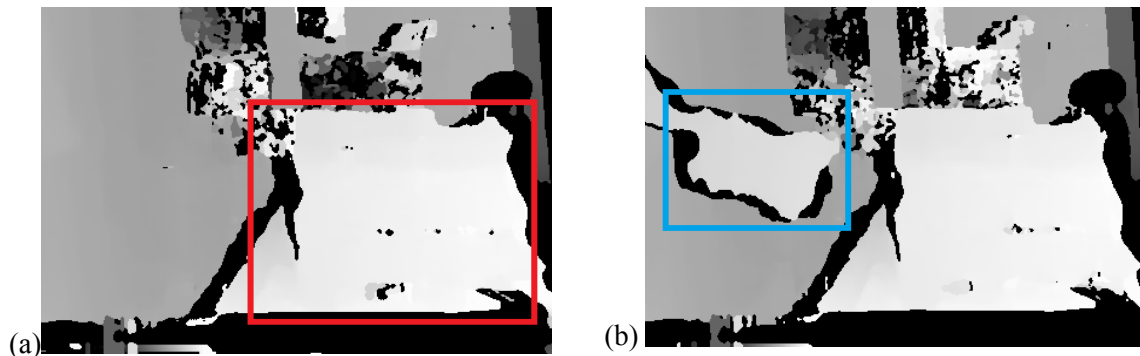


Figure 16. Image depth: (a) the rectangle underlines the load; (b) the rectangle underlines an intrusion.

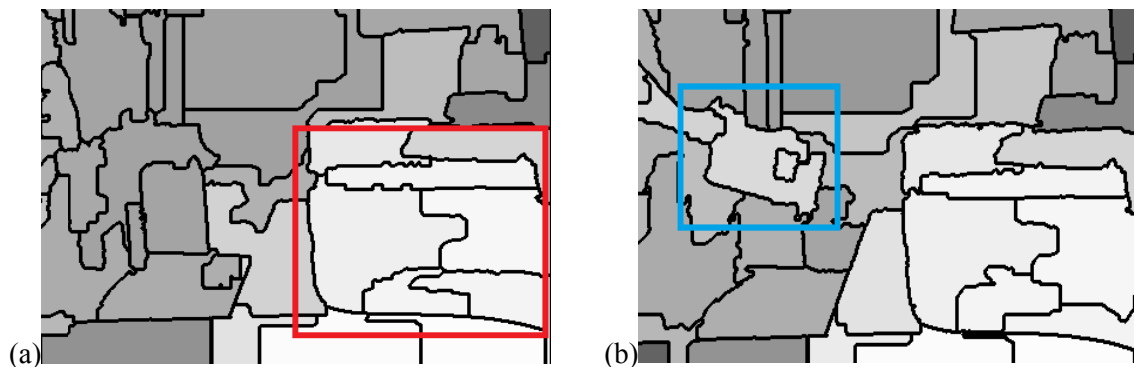


Figure 17. Segmentation of the depth images: (a) the rectangle underlines the load; (b) the rectangle underlines an intrusion.



### 3.2. Field tests

Several tests have been executed in an industrial context, thanks to the cooperation of the Edipower Company - a2a Group. The a2a Group is now the largest Italian multi-utility Company [14], it was available to test the SPRINCE prototype at the Thermoelectric Power Plant “Edipower” (located in San Filippo del Mela – Sicily, Italy).

The prototype has been installed on an overhead crane, which is shown in Figures 18 and 19. In Figure 20, the operator hooks the load (cover of a vapour turbine) to be moved by means of the crane.



*Figure 18. Overhead crane (Edipower – a2a Group).*



*Figure 19. Trolley of the overhead crane (Edipower – a2a Group).*





*Figure 20. Load positioning on the hook.*

Figure 21 shows the “custom made” metallic support, which has been properly designed and constructed in order to position the prototype on the crane. The image acquisition system is located on the support by the operator (Figure 22). Figure 23 shows the positioned system, which is ready to be used. The operator controls the load lifting by means of the VGS, which has previously been installed on the laptop (Figure 24). The computer communicates with the prototype by a remote desktop protocol.



*Figure 21. Metallic support for the prototype positioning.*

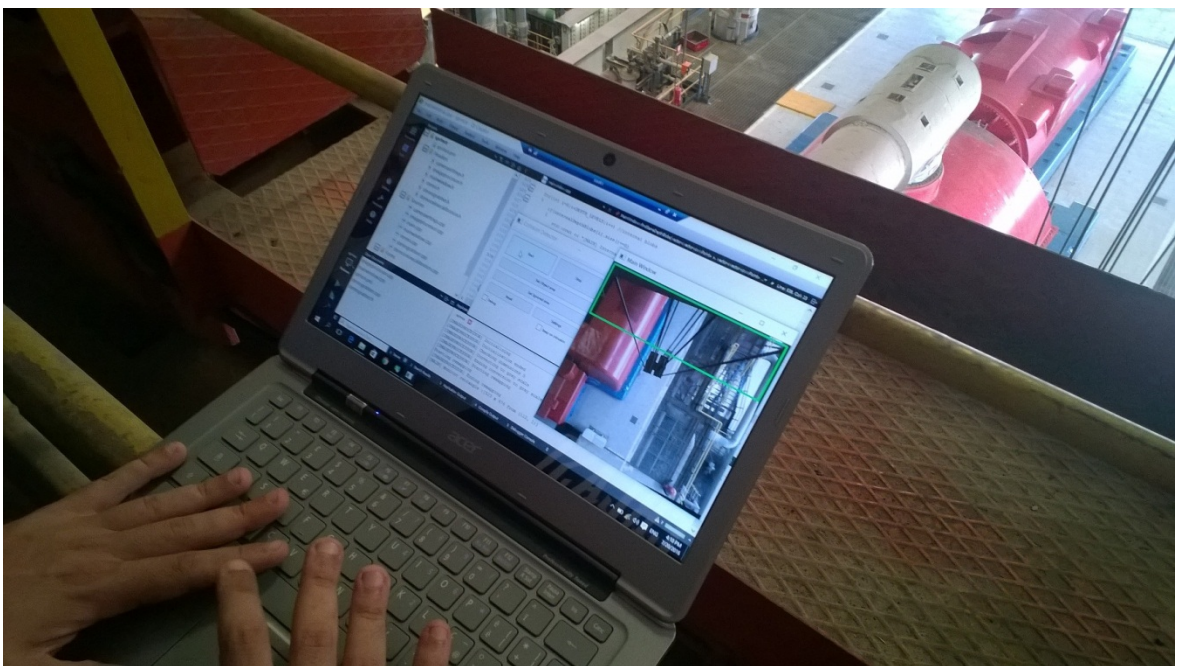


*Figure 22. Installation of the prototype by the crane-operator.*





*Figure 23. Positioned prototype.*



*Figure 24. Use of the VGS to control the crane-operation.*

Figure 25(a) shows the area to be controlled in the red box, i.e. the location of the load (in this case the hook as no load is positioned). Whereas in the Figure 25(b) the green rectangle highlights the area that has not to be detected because it is not part of the background to be processed and cannot cause collisions with the load. In this case, the portion of excluded area includes the cables supporting the hook.

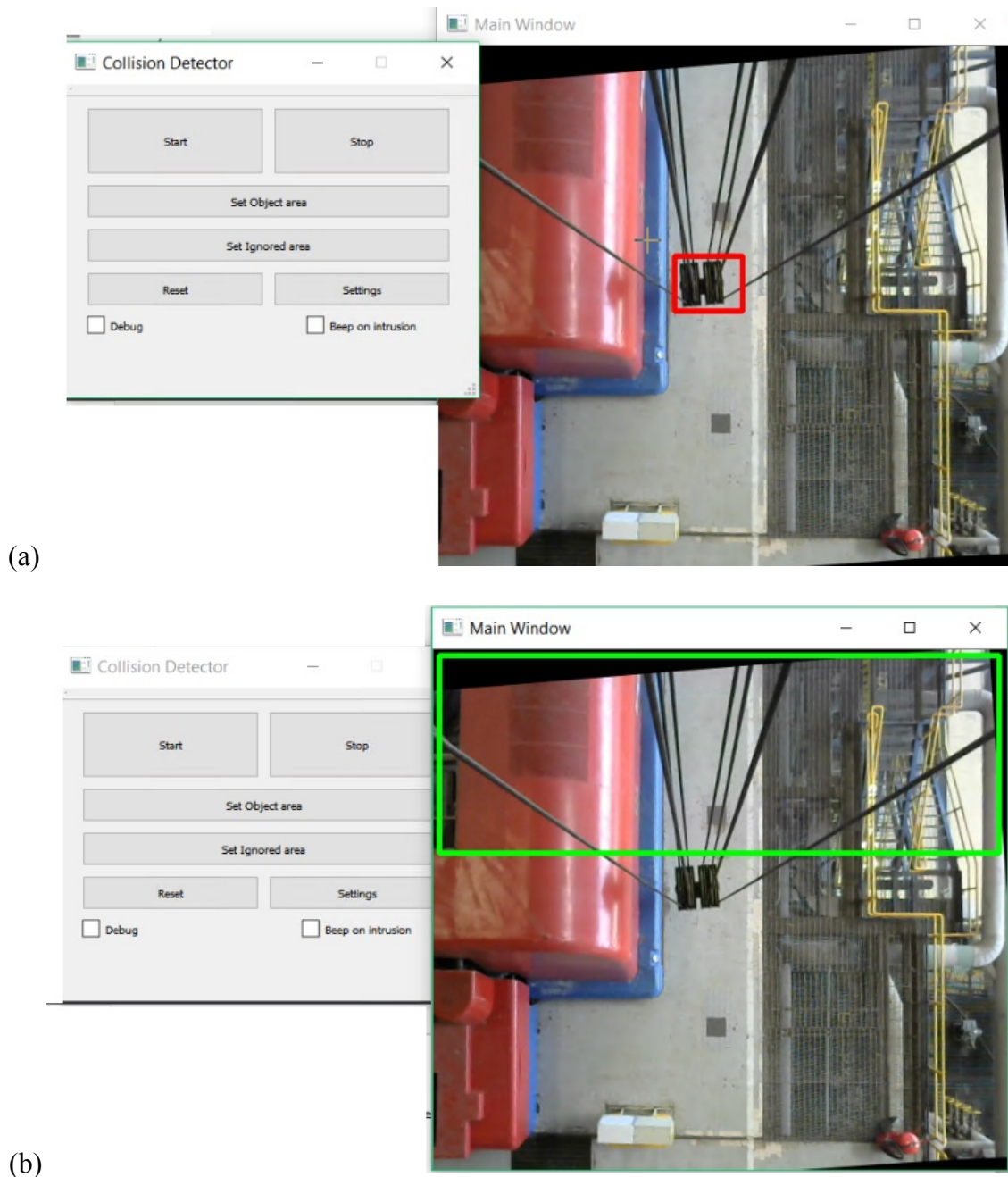


Figure 25. Screen-shoot of the VGS - Stereo Visual Guidance Detection System - (a) area to be controlled in the red box; (b) area that has not to be detected in the green box.

## 4. Major encountered problems and corrective actions

The major problems, encountered during the development of the VGS and the use of the prototype for the image acquisition, are related to the connectivity between the cameras and the laptop. These problems made the execution of the VGS instable, thus these must be solved given the software is the outcome of the activity *WPI.1*. Several tests have been made in order to implement both the prototype and the application. A list of the main problems is given below; each one has been discussed and related corrective actions have been properly addressed.

### 1. *High cost of the stereoscopic camera*

Given the high cost of a stereoscopic camera for the detection of the required depth for the investigated case-studies (~ 25,000 €), the solution of use of two cameras OpenNI type has been preferred (~ 3,500 €) to allow realising an economic system that captures stereoscopic images (see Section 2.2).

This alternative solution was also discussed with the other partners (during the meeting held in Belgrade on the 25<sup>th</sup> September 2015). After that two cameras GoPro Hero4 Black have been acquired.

### 2. *GoPro Hero4 Black cameras act as access points and not as Wi-Fi clients*

Using a pair of GoPro cameras may seem a straightforward task but, upon gaining experience with these cameras, some issues regarding network connectivity begin to emerge. In particular, the main problem is that a GoPro camera can act only as an access point and not as a Wi-Fi client. This implies that the processing computer should act as a client, thus its Wi-Fi card can be connected, at any given time, only at one Wi-Fi network (associated with one GoPro camera). Given that the aim is to connect two GoPro cameras at the same processing computer, in order to create the stereoscopic system, this connectivity problem has been solved by adding a second Wi-Fi card. In this way, each camera is connected to one card.

### 3. *IP addresses of GoPro Hero4 Black cameras are the same and not configurable*

Another problem that emerged with the aforementioned use of the GoPro cameras is that the Internet Protocol (IP) address is always identical in all cameras (10.5.5.9) and is not configurable. Having two or more identical IP addresses in the same network causes a conflict (also called *as IP address conflict*) because there is no way to distinguish between the two cameras and, when making a request, it is unknown which camera responds to it. Changing the IP address does not seem possible without an update of the manufacturer allowing this modification.

The natural approach to solve these problems was to introduce an intermediate layer (e.g. a proxy) that would interface independently with each camera and present a



different IP to the processing computer. Three solutions have been identified and analysed.

- 1) Virtual machines (VM)
- 2) Containers
- 3) Single-Board Computers (SBC)

With regard to the VM-based solution, such a proxy (more precisely, network address translation, NAT) layer would be hosted inside each VM, this means that within each VM this system must have to be setup for each camera. An issue associated with this kind of solution is the need to minimise the number of translations/encapsulations involved, achieved by assigning the wireless network interface devices directly to the guest OS, in turn not always an option for a number of buses, and not widely available as a functionality for a number of hypervisors, even when applicable. Moreover, the overhead of running full-fledged VMs cannot be underestimated, especially when running on a machine already loaded with taxing Computer Vision-related processing work.

The second solution has been identified as the use of Linux containers (LXC/Docker) as a way to reduce the footprint of virtualisation, still being able to support two network interfaces installed on the same physical machine with the same (i.e., conflict-inducing) IP address being configured, by means of name-spacing as a way to isolate the respective networking setups and stacks. The trade-off between performance and OS choice: compared to VMs, performance is higher, with typically negligible overhead when compared to the bare metal. Unfortunately, this choice brings a specific dependency on a container-enabled UNIX derivative as host OS, unlike any VM-based solution.

Then, the third choice has been to offload these networking translation duties to (external) physical machines, by resorting to extremely cheap and compact SBCs, like the Raspberry Pi family of embedded platforms. Such SBCs are already quite powerful enough to cope with network translation without any hiccups or excessive load on the CPU; moreover these have already an integrated Ethernet-class interface, which is used to connect them by means of Eth/USB adapters to the computer, whereas the integrated Wi-Fi of the SBCs connects them to the GoPro cameras. The scheme of this solution is shown Figure 3. In case of this third choice, the two Raspberry need to be configured within the system (see tutorial in [Annex 4](#)). The use of such devices requires additional components (with respect to those listed at page 8), these are: n.2 USB adapter /Ethernet, n.2 Ethernet cable and n.2 microSD 8GB

### 5. *Low resolution of the image acquisition*

Even if the problem of connectivity has been solved there is still a limitation: Wi-Fi transmission of HD signal is not possible with a sustainable rate without expensive and sophisticated hardware that goes beyond the budget of this project. GoPro cameras are able to transmit video wirelessly at a 432 x 240 resolution. This limits the performance, in terms of accuracy of the algorithm.

The suggested solution for this problem would be the deployment of a professional high-end video encoders and decoders (e.g. the Teradek Inc Cube-255 200 Cube) in order to make full use of the HD capabilities of the cameras. The drawback is the price, as for each camera an additional cost of about 1500 \$ would incur.

### 6. *High electrical noise in the working area*

Typical equipment in the working area consumes large quantities of electric current and, subsequently, emits a high electrical noise. For this reason, the Wi-Fi connection to the GoPro cameras resulted unreliable and this causes missing or malformed frames. It would be possible to solve this problem by applying an encoding error detection algorithm [15]; this approach has some drawbacks due to the following reason. The packet loss is proportional to the distance of the cameras from the Raspberry receivers. This means that, while this approach would perform well when the crane is close to the receivers, the drop rate would become too high when the crane is far from them, making this solution inappropriate. Nevertheless the use of an encoding error detection algorithm would not be applicable because it requires the GoPro's firmware modification.

### 7. *Problems of cameras' synchronisation*

The tests' execution in the industrial context has revealed another issue, which was not present during the laboratory tests, it is related to the cameras' synchronisation. In particular, in order to derive accurate depth estimation, the two images acquired from the cameras must be exactly shot at the same time, with a tolerance of a maximum equal to the frame duration (e.g. 40 ms for a frame rate equal to 25 frames/s). Given that the transmission channel has not been unreliable (Wi-Fi transmission) the synchronisation has not been easy; maybe this problem is connected to the problem no. 5 and the same solution discussed above could be applied.

In order to deal with the problems associated with the connectivity, which have been underlined at the point 4, another possible solution is to transfer the largest part of the workload via cabled connection instead of a wireless one. The largest part of the workload is the transmission of clear and coloured images. For this reason, two USB cameras have connected to a laptop and these have been positioned in a box on the cranes. Another

computer (even a tablet or a mobile device), which acts as a client, is connected to the box's content via remote desktop (see Figure 26).

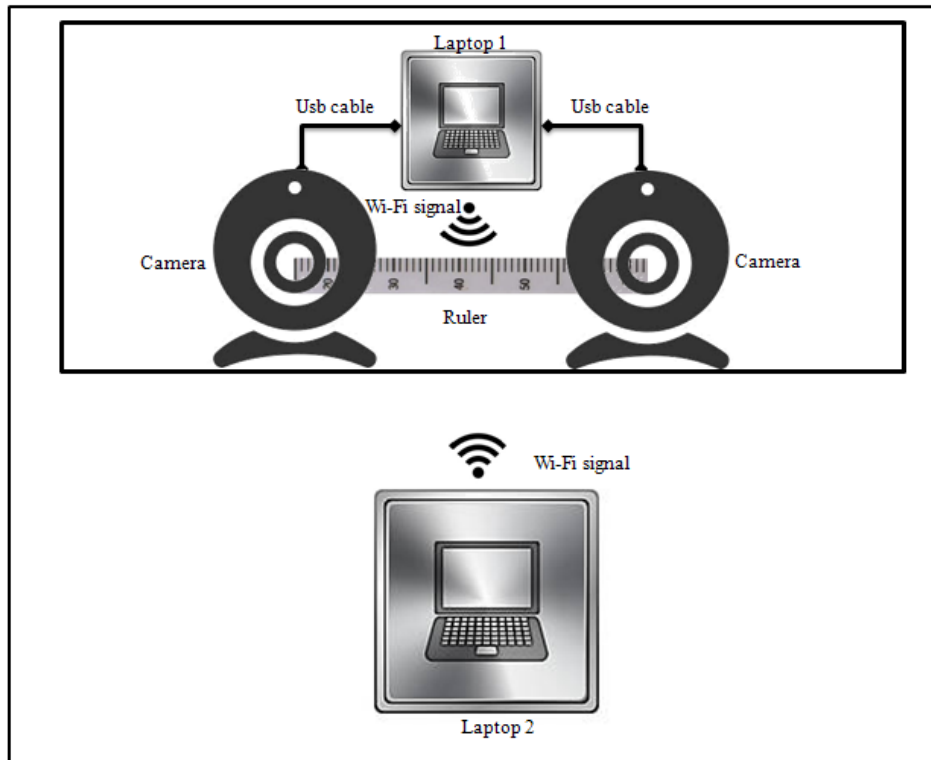


Figure 26. Prototype camera-computer (alternative configuration).

This solution has the advantage that the developed application is not modified at all and that the heavy burden of transmission and computations is done on the laptop placed inside the box. Beyond this, it must be added that no additional costs to the final solution except the cost of a pair of webcams and of the client device. In this case, the wireless connection still occurs but it is limited to the data transmission of the remote desktop protocol.

Such a solution has been also tested in the industrial context, but other problems have been encountered. The most relevant has been that the overheating of the laptop inside the box, due to the hot day occurred when the test was performed (around 40°C). Such an overheating caused the automatic computer shut down to avoid the system damage. This inconvenience has invalidated the work done and caused also time loss. Thus an insulating box should be used to place the components of the prototype.



#### 4.1. Software tests

The software has been tested in order to understand its behaviour in real-time conditions. While the system has been able to identify many possible collisions, initially the false positive rate has been very high, meaning that the system alerted the user very often of collisions that are not taking place. After further investigation, the problem has been found to be associated with the calibration. It has been seen that the depth maps at medium distances ( $> 10$  m) were not accurate, even if the calibration result was very good in near distances. It has been tried to increase the resolution of the cameras but this has brought the processing efficiency of the system under the threshold of 1 frame per second and made the system non-real time. This could be dealt with more processing power (powerful CPU), but the higher heat emission and power requirements must be taken into account.

Figures 27 and 28, respectively, show a false positive due to the cables which were still not excluded at the time of the acquisition and other unknown false positives. False positives are evidenced as blue rectangles.

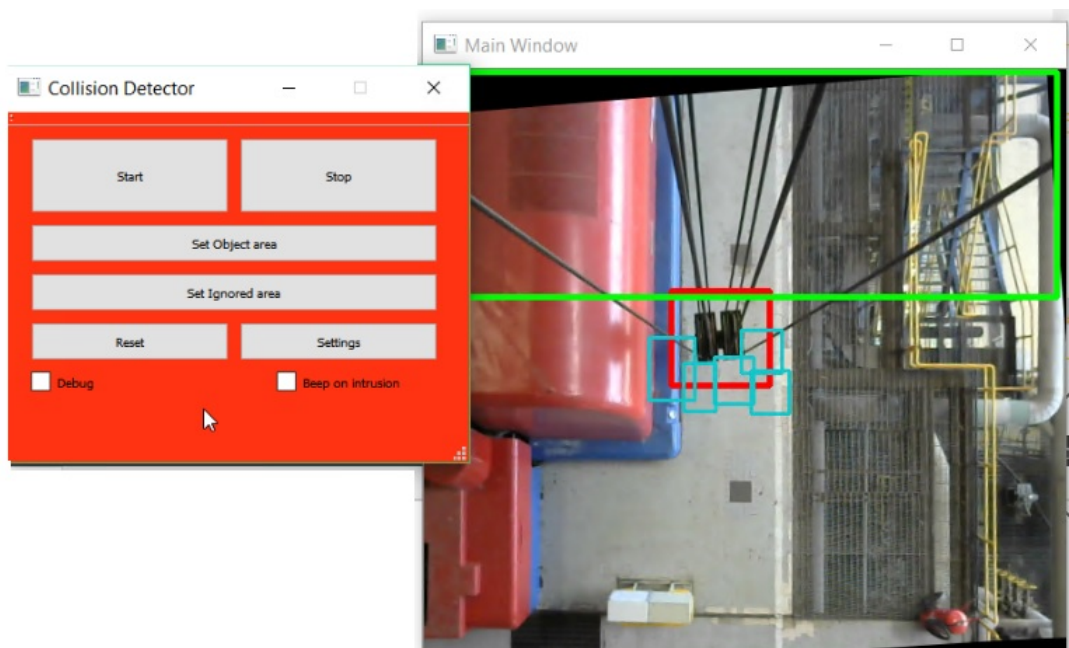


Figure 27. False positives due to the cables, which were still not excluded.

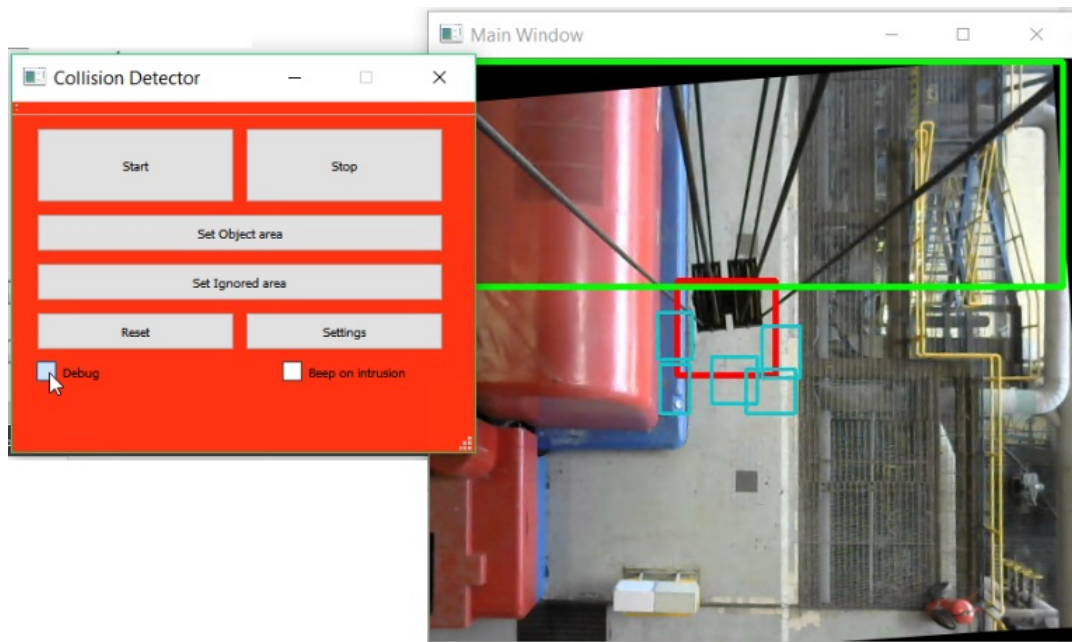


Figure 28. Unknown false positives.

Another problem associated with the software is that when the distance of the object from the camera gets high, it escapes from the designated area (red box in Figures 27-28). In this case the object cannot be monitored anymore and something must be done in order to guarantee the object is always positioned inside the rectangle. To ensure this, algorithms with extraction features of the interested objects can be used. For example, SIFT descriptors [16] can be employed and used in conjunction with GrabCut [17] to accomplish this. During the phase of the selection of the object (where the object rectangle is drawn) the GrabCut algorithm can isolate the object from the rectangle and afterwards the SIFT descriptors extracts it. Once the SIFT descriptors of the object are available, at every successive frame the object can be localised by using the descriptors and moving the object rectangle appropriately. In this way the crane's load can be tracked in every position.

## 5. Deviations from the work plan

In the task breakdown of the SPRINCE project, there was considered a 3D modelling of the scene by using multiple cameras (*T1.1.2*). The main disadvantage of this method is the sensitivity to external disturbance, which are numerous in an industrial context. Therefore, instead of using a three-dimensional modelling, a *stereoscopic video acquisition* method has been investigated and adopted to make application able to accurately estimate the real dimensions of the items, beyond the distance between a moving object and the camera.

## 6. Produced publications

Concerning the activity *WP1.1* “Visual guidance system development”, the following publications have been produced:

- ▶ Milazzo M.F., Ancione G., Spasojevic Brkic V., 2015. Safety in crane operations: an overview on crane-related accidents. Proc. 6<sup>th</sup> International Symposium on Industrial Engineering SIE, 36-39, Belgrade, Serbia (24-25 September 2015).
- ▶ Spasojević Brkić V., Milazzo M.F., Brkić, A. Maneski T., 2015. Emerging risks in smart process industry cranes survey: SAFERA research project SPRINCE. Serbian Journal of Management 10(2): 247-254.
- ▶ Milazzo M.F., Spasojevic Brkic V., Valis D., 2016. Improving Cranes' Safety: Development of a Real-Time Visual Guidance System to Move Loads in Process Industry. Communication at the SAFERA Symposium “Emergence of a New Collaborative Work Programme on Industrial Safety”, Athens, Greece, 11-12 April 2016.
- ▶ Milazzo M.F., Ancione G., Spasojevic Brkic V., Valis D., 2016. Investigation of crane operation safety by analysing main accident causes. Submitted at ESREL 2016 conference.
- ▶ Ancione G., Kavasidis I., Merlino G., Milazzo M.F., 2016. Real-time guidance system for cranes to manage risks due to releases of hazardous materials. Submitted at ESREL 2016 conference.

## References

- [1] NUREG-1774 (2003). *A Survey of Crane Operating Experience at the US Nuclear Power Plants from 1968 to 2002*. US Nuclear Regulatory Commission, Washington, DC, 20555-0001.
- [2] Milazzo M.F., Ancione G., Spasojevic Brkic V. (2015). *Safety in Crane Operations: An Overview on Crane-Related Accidents*. Proc. of the 6<sup>th</sup> International Symposium on Industrial Engineering SIE 2015, 36-39, Belgrade, Serbia (24-25 September 2015). ISBN: 978-86-7083-864-2.
- [3] Amato A., Di Lecce V., Piuri V. (2013). *Semantic Analysis and Understanding of Background modeling and subtraction by codebook construction Human Behavior in Video Streaming*. Springer, London, UK.
- [4] Jeeva S., Sivabalakrishnan M. (2015). *Survey on Background Modeling and Foreground Detection for Real Time Video Surveillance*. Procedia Computer Science, 50: 566-571.
- [5] Hossain, S., Nabendu Chaki K.S. 2014. *Moving Object Detection Using Background Subtraction*. Kolkata: Springer.
- [6] Barnich O., Van Droogenbroeck M. (2011). *ViBe: A universal background subtraction algorithm for video sequences*. Image Processing, IEEE Transactions, 20(6): 1709-1724.
- [7] Stauffer C., Grimson W.E.L. (1999). *Adaptive background mixture models for real-time tracking*. Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2: 246-252
- [8] Faro A., Giordano D., Spampinato C. (2011). *Adaptive background modeling integrated with luminosity sensors and occlusion processing for reliable vehicle detection*. Intelligent Transportation Systems, IEEE Transactions 12(4): 1398-1412.
- [9] Kim K., Chalidabhongse T.H., Harwood D., Davis L. (2004). *Background modeling and subtraction by codebook construction*. Proc. of the International Conference on Image Processing, ICIP, 2: 3061-3064.
- [10] Wang A.H., Wang Q.H., Li D.A. (2010). *Stereoscopic Image Acquisition and Autostereoscopic Display*. Proc. of the Symposium on Photonics and Optoelectronic (SOPO), 1-3.
- [11] Hoiem D., Efros A., Hebert M. (2005). *Automatic photo pop-up*. ACM Transactions on Graphics, 24(3): 577-584.

- [12] United States Patent Application Publication (2005). *Stereoscopic image acquisition device*. Patent no. US 2005/0041095 A1.
- [13] British Standards (2004). *Cranes — Mobile cranes*. Document no. EN 13000.  
Available on-line (accessed 13<sup>th</sup> September 2015):  
<https://law.resource.org/pub/us/cfr/ibr/003/bs.en-13000.2004.html>
- [14] Website Edipower - a2a Group (accessed 13<sup>th</sup> September 2015)  
<http://www.edipower.it/edp/cms/edipower/>
- [15] Wang Y., Qin-Fan Z. (1998). *Error control and concealment for video communication: A review*. Proc. of the IEEE 86.5: 974-997.
- [16] Yan K., Sukthankar R. (2004). *PCA-SIFT: A more distinctive representation for local image descriptors*. Computer Vision and Pattern Recognition (CVPR 2004), Proceedings of the 2004 IEEE Computer Society Conference, 2.
- [17] Carsten R., Kolmogorov V., Blake A. (2004). *Grabcut: Interactive foreground extraction using iterated graph cuts*. ACM Transactions on graphics (TOG), 23(3).

## **Annex 1**

### **Questionnaire for the Company**



University of Messina



University of Belgrade



University of Defence

## QUESTIONNAIRE FOR THE COMPANY

1. Has your crane a real time monitoring system?

- YES       NO

\*If the answer is YES go to question 2, otherwise go to question 10.

2. Which are main elements of the system (camera, computer, joystick, etc.)? Please list them.

---

---

---

3. Does the system give back an alarm in case of dangerous load handling? What type of signal does it give back (audio, visual, etc.)?

- YES       NO

---

---

---

4. Where is/are the camera/s located?

---

---

---

5. Is the position of the camera fixed?

- YES       NO

---

---

---

6. If the position of the camera is not fixed, can the operator move it and which type of technology is used to move it?

- YES       NO

---

---

---

7. If the monitoring system has a monitor, where is it located (in the cab or outside)?

---

---

---



8. Can the existing system be interfaced with an external computer?

- YES       NO

9. Which are the technical characteristics of each element of the system?

- Manufacturer and model of Crane: \_\_\_\_\_
- Manufacturer and model of Camera: \_\_\_\_\_
- Manufacturer and model of Monitor: \_\_\_\_\_
- Manufacturer and model of Joystick: \_\_\_\_\_
- Manufacturer and model of the inside computer: \_\_\_\_\_
- Version of the computer's OS (Operating System): \_\_\_\_\_
- Indicate the manufacturer and model of other elements which have not been mentioned in the list above:

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

## **Annex 2**

### **Data collected from Companies**

## Data collected from Company Loreficie & Ponzio (Italy)

Question	Company 1
Crane's type	<i>Mobile crane</i>
1. Has your crane a real time monitoring system?	<i>NO</i>
2. What are the main elements of the system (camera, computer, joystick, etc.)? Please list them	<i>--</i>
3. Does the system return an alarm in case of dangerous load handling? What type of signal does it return (audio, visual, etc.)?	<i>--</i>
4. Where is/are the camera/s located?	<i>--</i>
5. Is the position of the camera fixed?	<i>--</i>
6. If the position of the camera is not fixed, can the operator move it and which type of technology is used to move it?	<i>--</i>
7. If the monitoring system has a monitor, where is it located (in the cab or outside)?	<i>--</i>
8. Can the existing system be interfaced with an external computer?	<i>--</i>
9. What are the technical characteristics of each element of the system?	<i><u>Manufacturer and model of Crane: LIEBHERR LTM 1200. 5.1 (documents are available)</u></i>
10. Can an external computer be located in the cab?	<i>Missed answer</i>
11. By considering the free space in the crane-cab, what type of external computer is preferred to be located in the cab?	<i>Tablet</i>
12. Can the camera position be chosen/changed?	<i>Missed answer</i>
13. Availability to produce a video during the operations.	<i>YES</i>

## Data collected from Bajina Basta hydropower plant (Serbia)

Question	Company 2
Crane's type	<i>Bridge crane</i>
1. Has your crane a real time monitoring system?	<i>NO</i>
2. What are the main elements of the system (camera, computer, joystick, etc.)? Please list them	--
3. Does the system return an alarm in case of dangerous load handling? What type of signal does it return (audio, visual, etc.)?	--
4. Where is/are the camera/s located?	--
5. Is the position of the camera fixed?	--
6. If the position of the camera is not fixed, can the operator move it and which type of technology is used to move it?	--
7. If the monitoring system has a monitor, where is it located (in the cab or outside)?	--
8. Can the existing system be interfaced with an external computer?	--
9. What are the technical characteristics of each element of the system?	<i>Manufacturer and model of Crane: Ivo Lola Ribar, capacity = 63/20 t, structure group DIN H2B3, span 22m, lifting height 25 m, lifting speed main hook 5/0.5 m/min, auxiliary hook 6.3 m/min, trolley speed 16 m/min, bridge speed 25 m/min. Manufacturer and model of Joystick: producer ISKRA KRANJ model KP5/D-0313</i>
10. Can an external computer be located in the cab?	<i>YES</i>
11. By considering the free space in the crane-cab, what type of external computer is preferred to be located in the cab?	<i>Tablet</i>
12. Can the camera position be chosen/changed?	<i>One camera on hook, other on cabin.</i>
13. Availability to produce a video during the operations.	<i>YES</i>

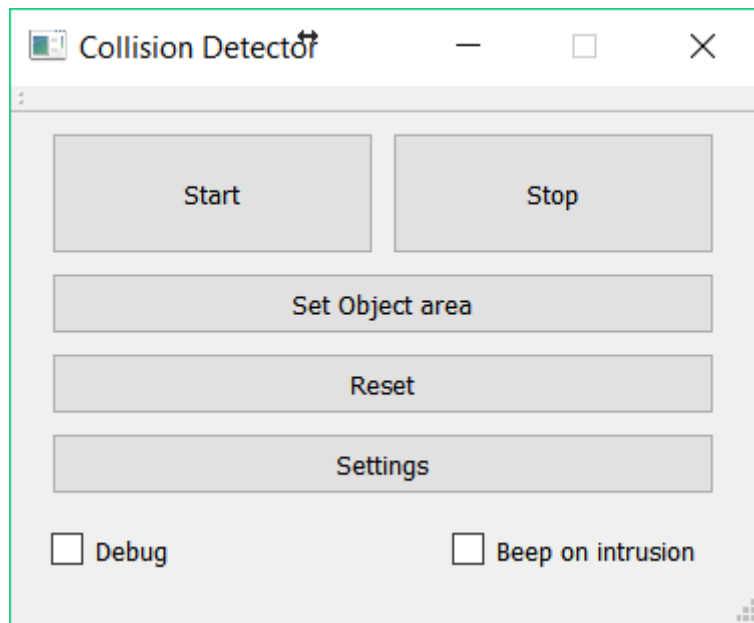
## **Annex 3**

### **Use of the VGS - Tutorial**

# STEREO VISION GUIDANCE SYSTEM (VGS)

## USER'S MANUAL

### 1. The main GUI



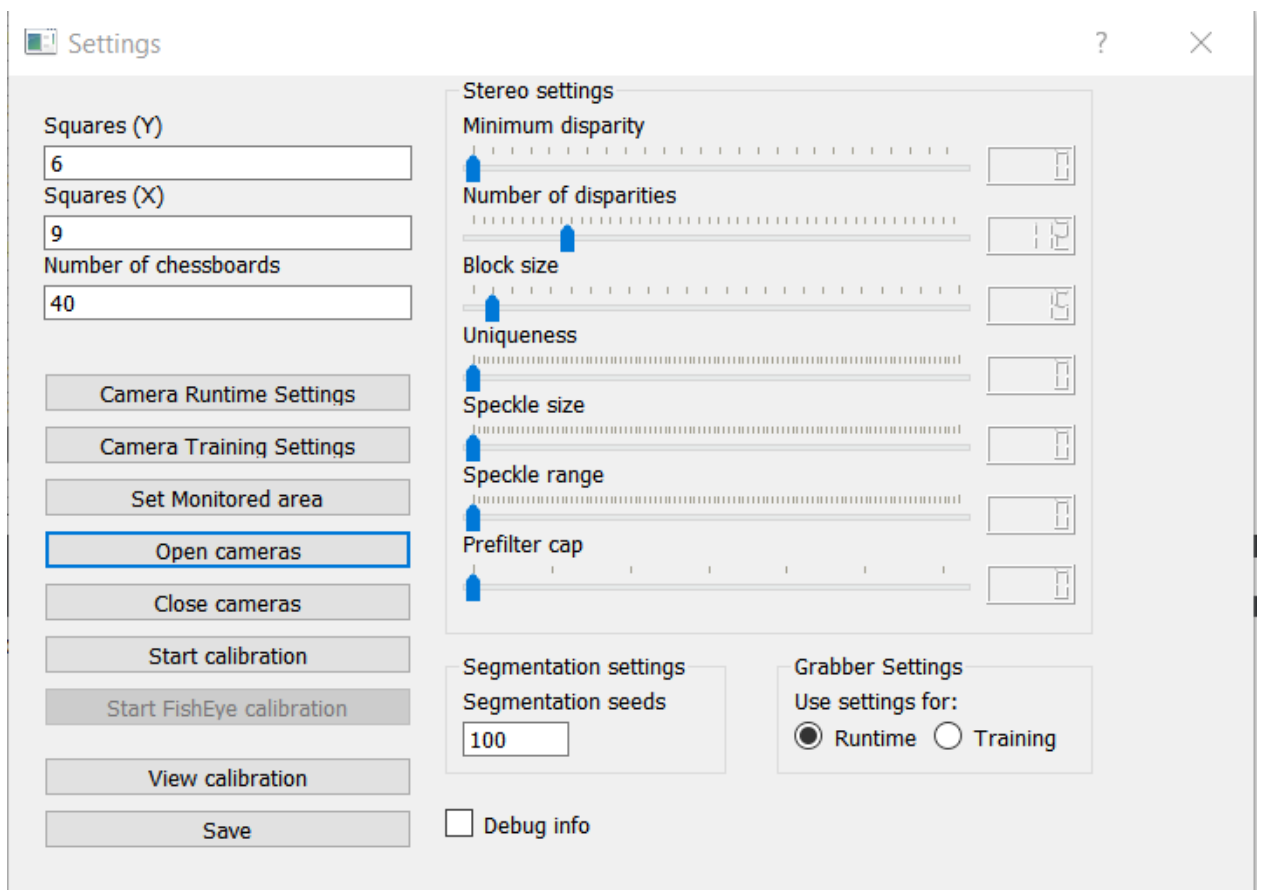
The main graphical user interface features 5 buttons and two checkboxes:

- Start Button: When clicked, the system starts the collision detection process.
- Stop Button: When clicked, the system stops the collision detection process.
- Set Object Area Button: This button is used to create a rectangle on the main window of the object area (i.e. the area where the object is located – red rectangle in the video-demo).
- Reset Button: When clicked, this button restores the system to a starting state (it closes all windows and erases the object rectangle).



- e. Settings Button: When clicked, this button shows the settings GUI.
- f. Debug Checkbox: If checked, the system shows the results of the intermediate results of the algorithms employed during the collision detection system.
- g. Beep on intrusion Checkbox: If checked the form flashes with red colors and produce a sound when an obstacle enters the object area.

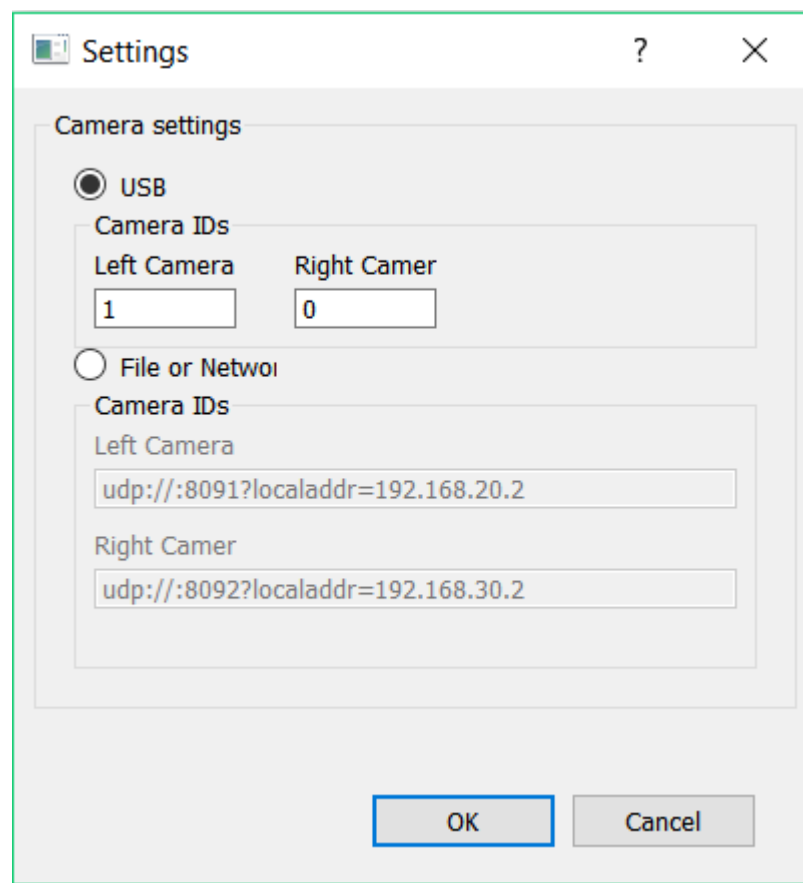
## 2. The settings GUI



The settings form hosts all the necessary controls to fully configure the collision detection system:

- a. Squares (Y) Textbox: The number of internal squares (height) found in the calibration pattern

- b. Squares (X) Textbox: The number of internal squares (width) found in the calibration pattern
- c. Number of chessboards: Number of images to be used in the calibration process. The chessboard pattern must be identified in both the images (left and right) in order to be considered. The bigger is the number of images needed, the longer is the duration of the calibration process (exponential relation). When the calibration fails, and if everything else is correct, it is necessary to increase this number in steps of 5 (45, 50, 55 etc.).
- d. Camera Runtime Settings Button: When clicked, this button shows the camera settings that will be used during the collision detection process. In the case shown in the Figure, the USB cameras are set. If you are going to use WiFi cameras, such as the Gopro Hero Black 4, please select “File or Network” and insert the following address: “udp://:8091?localaddr=192.168.20.2” under the field “Left Camera”. Then, in the field “Right Camera” write: “udp://:8092?localaddr=192.168.30.2”.



- e. Camera Training Settings Button: Same as above but these settings are only used during the calibration and testing phases.
- f. Set Monitored Area Button: After the calibration process, the useful part of the image is almost always smaller than the images themselves. With this button, the user can designate the area where the monitoring occurs (see Section 3 for more information).
- g. Open Cameras Button: When clicked, this button opens the cameras that were configured with the training settings. If the Debug info checkbox is checked, it also shows the acquired videos.
- h. Close Cameras Button: When clicked, it stops the acquisition process and also closes all open image windows.
- i. Start Calibration Button: When clicked it starts the calibration process (see Section 3)
- j. Start FishEye Calibration Button: NOT USED
- k. View Calibration Button: When clicked, it shows the current calibration settings by means of the calculated depth map. If the current calibration settings are good, the user can start the collision detection. Otherwise, the calibration must be performed again.

### 3. Calibration and Settings

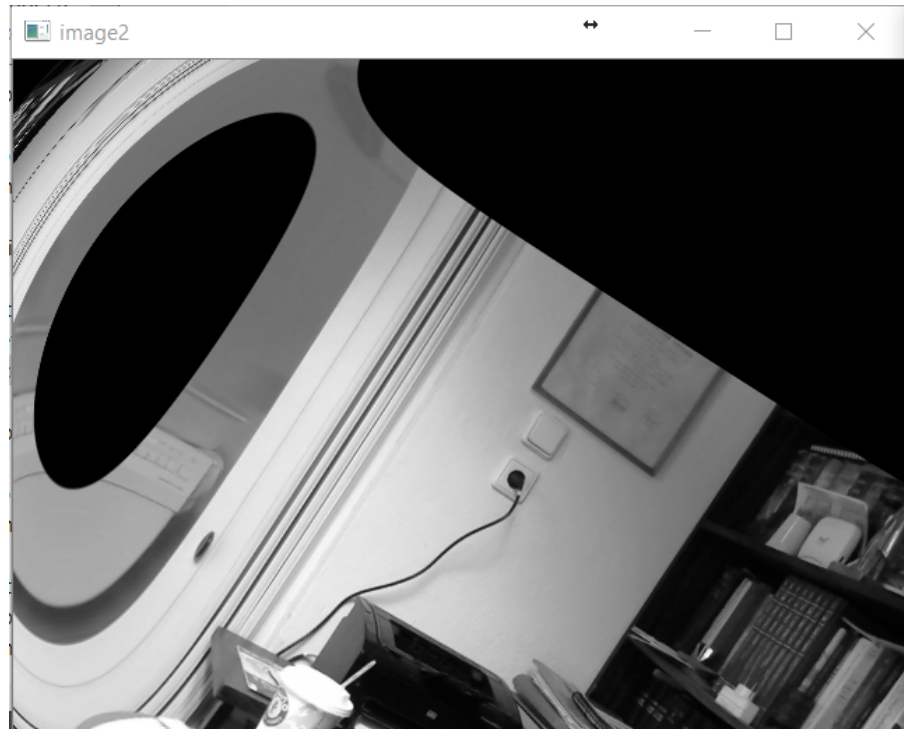
In order to run the Stereo Vision Collision System, firstly a very precise calibration of the cameras must be performed. The calibration is needed because the system has to know the intrinsic (lens distortion) and extrinsic (relative position of the cameras) parameters.

In order to calibrate and set the system, the following steps (**in this precise order as given below**) must be followed:

- a. **Setting up the cameras**: From the Settings GUI, configure the cameras both for training and runtime.
- b. **Check camera settings**: Enable the **Debug Info** checkbox and click on the **Open Camera Button**. The program should show both camera outputs. **During this step, ensure that the window named left shows the left camera output and the**

- window named right shows the right camera output.** If everything is fine, click on the Close Cameras Button.
- c. Print a chessboard pattern (like the one found at the last page of this document) and attach it to a flat surface. **Please ensure that the surface is as flat as possible.**
  - d. Fill the Square(Y) and Square(X) textboxes with the correct values (the internal angles of the chessboard pattern.) For the one supplied the correct values are 6 and 9.
  - e. Fill the number of chessboards parameter. Start with 40 images and start increasing in steps of 5.
  - f. Click on the **Start Calibration Button**. During this phase the cameras are opened, the user must move the chessboard pattern in front of the cameras. **During calibration please ensure that the whole optical view of the cameras is covered and in different depths.**
  - g. After the system acquires a number of images with the chessboard pattern equal to the parameter set during step, the program closes all image windows and performs the calibration process. When the calibration process terminates the program shows the remapped images (undistorted images with lens correction) and the calculated depth maps.

### BAD CALIBRATION RESULTS

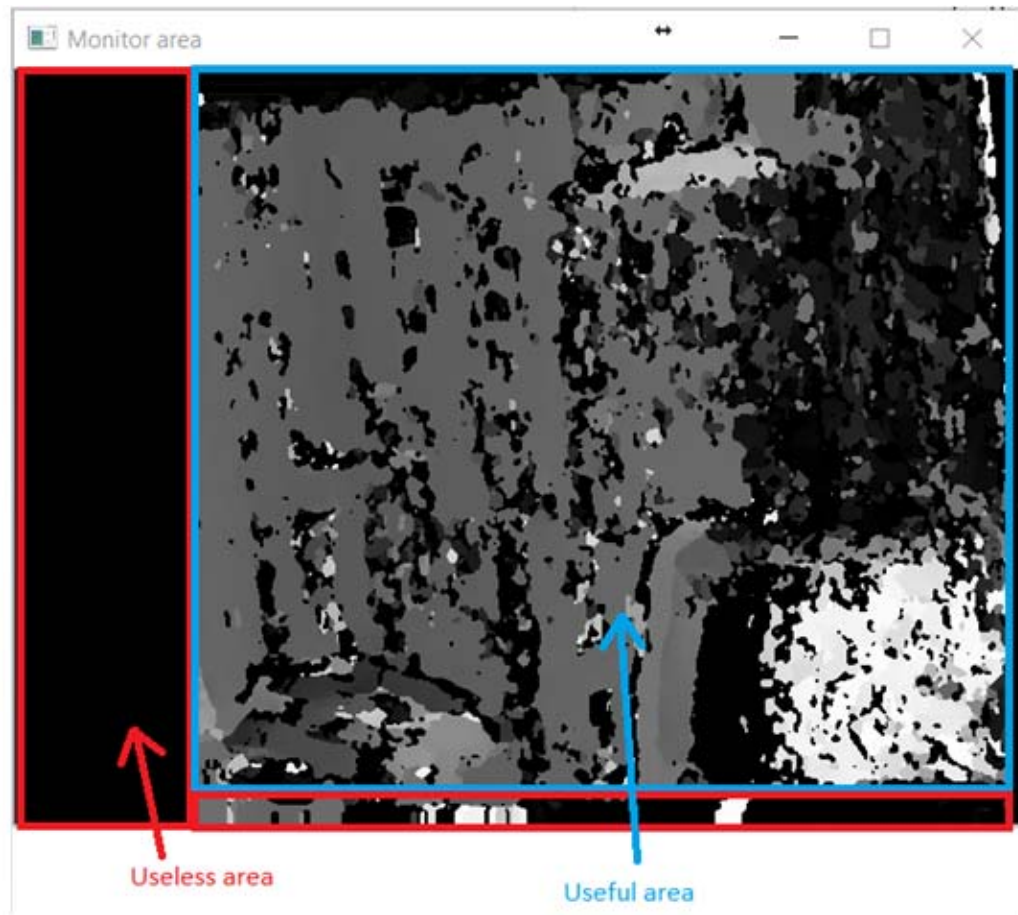


## GOOD CALIBRATION RESULTS

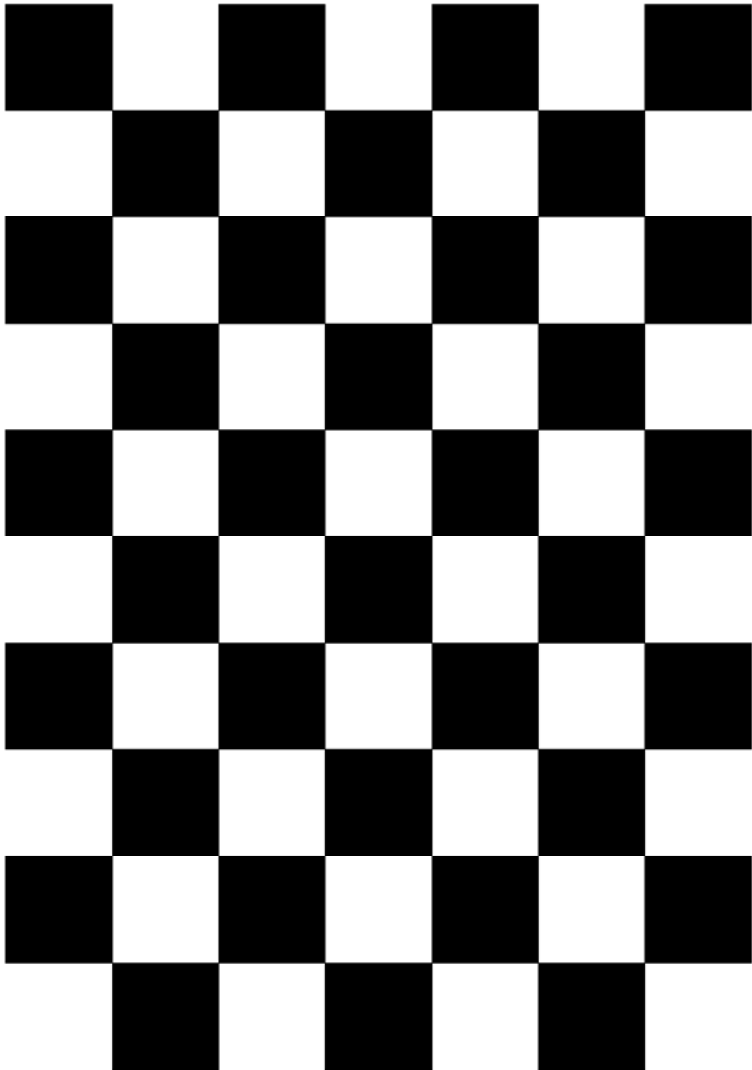


- h. If the corrected images are clear the calibration phase is concluded. Otherwise repeat from step e.
- i. The calculated depth map will not be perfectly clear. For this reason the user must use the **Stereo Settings Sliders** in order to create the most accurate representation of the depth maps. The most important settings to tackle are the **Number of Disparities**, **Block Size**, **Uniqueness** and **Speckle Size** parameters. The **Number of Disparities** should have a minimum value of 92. The **Block Size** should have a minimum value of 13. The **Uniqueness** and the **Speckle Size** can range from zero to their maximum values. Leave all the rest to zero.
- j. Click on the Set Monitoring Area Button. The program responds with a window showing a depth map:





During this step the user must draw a rectangle representing the useful area of the depth map (blue rectangle).



## **Annex 4**

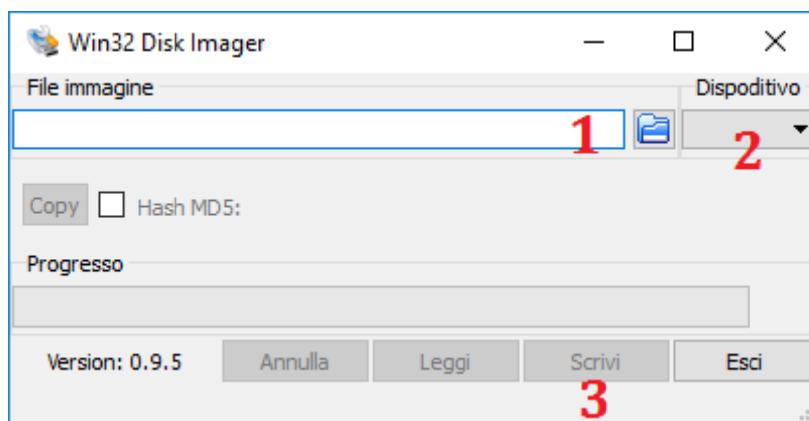
### **Configuration of Raspberry - Tutorial**

## PROCEDURE FOR THE PRE-CONFIGURATIONS OF RASPBERRY PI 3

### TUTORIAL 1

Download the “*Win32DiskImager-0.9.5-install*” and install it. Run this application as administrator.

The execution of the Win32DiskImager-0.9.5 application shows the window:



In **1** insert the path of the file “*rasp1backup.img*”.

In **2** select the unit which refers to the MicroSD having a Capacity 8 GB and that you will insert in Raspberry pi3.

Click on the button WRITE **3**.

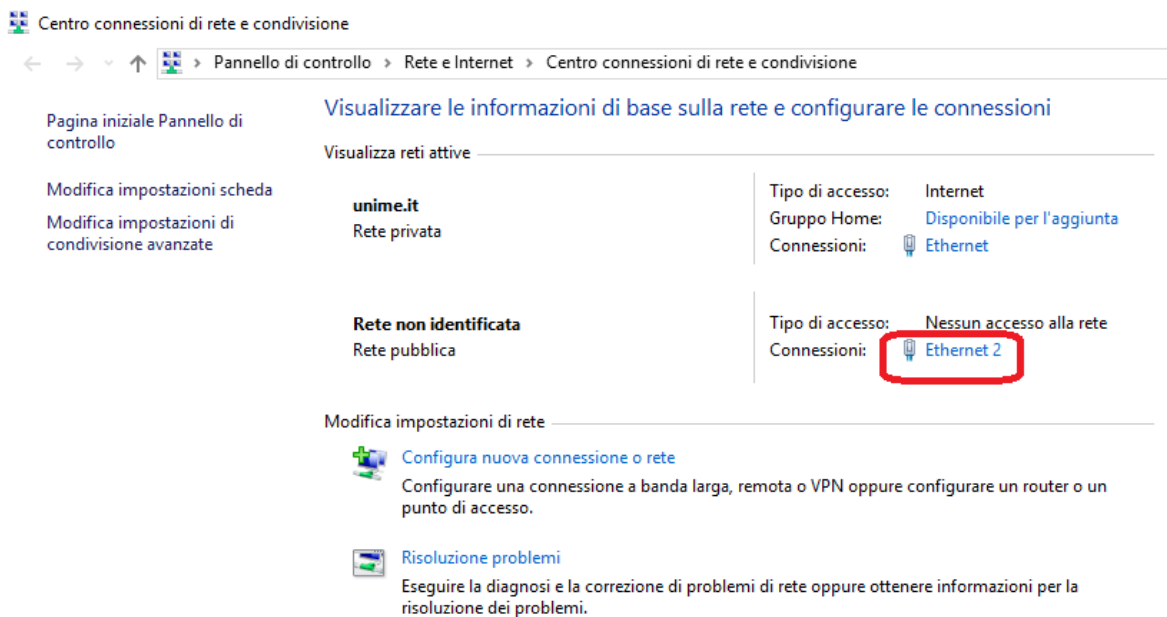
Subsequently repeat the same procedure to register the file “*rasp2backup.img*” on the other MicroSD having a Capacity 8 GB.

Put both MicroSD inside the two Raspberry pi 3.

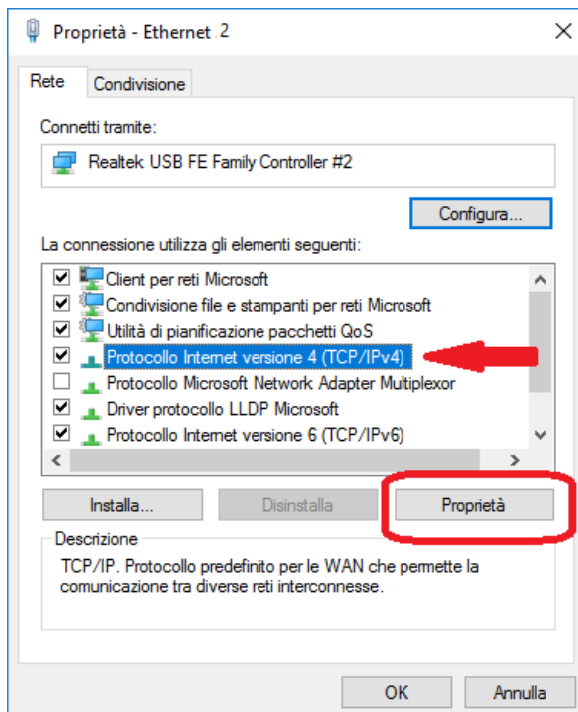
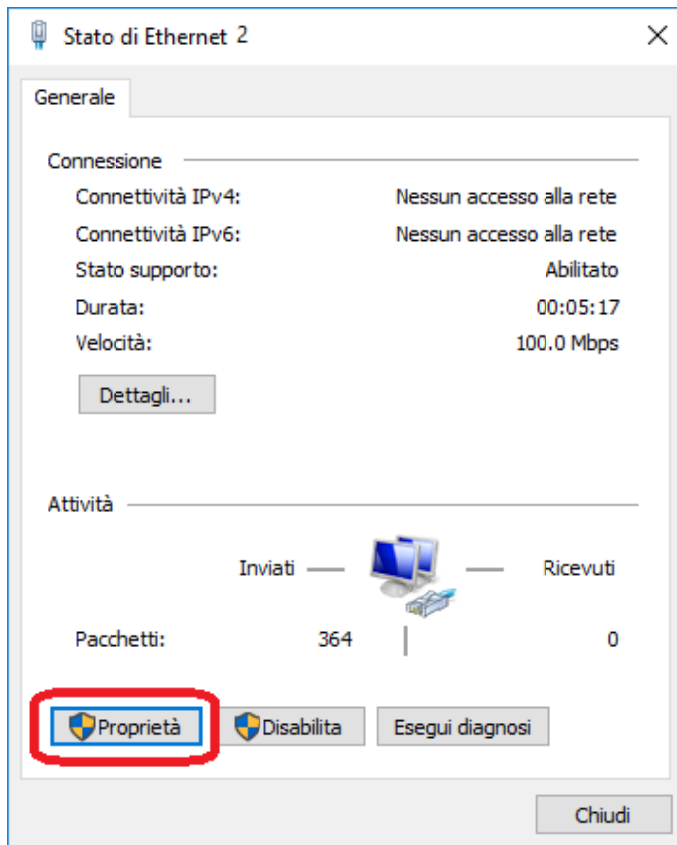
# PROCEDURE OF CONFIGURATION OF THE RASPBERRY PI3 AND GOPRO HERO BLACK4

## TUTORIAL 2

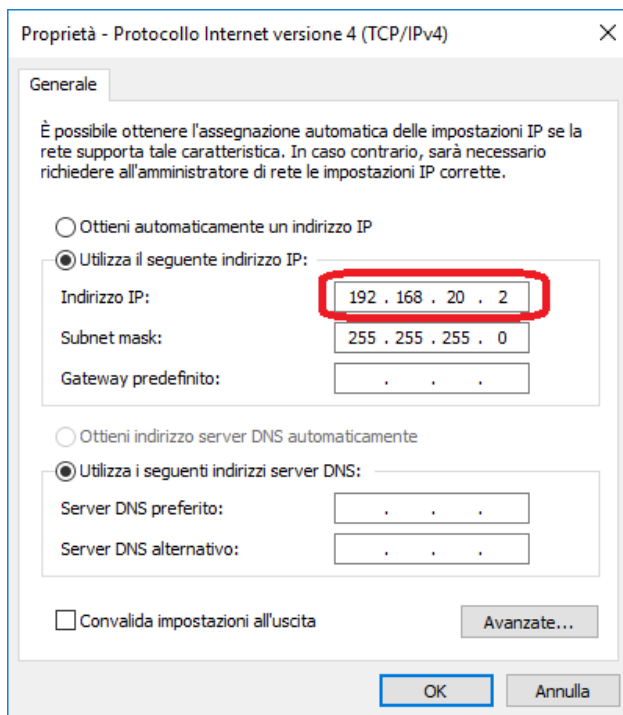
1. Start gopro cameras and activates the “WiFi-app” mode
2. (please, it is important to assign these passwords: “cam1cam1” for the camera 1 and “cam2cam2” for the camera 2)
3. Disable the firewall of your Operative System
4. Connect the first raspberry to the PC via the USB Adapter / Ethernet
5. Connect the raspberry to the power bank
6. Open the “Network and Sharing Center” of you Operative System
7. Click on the new Connection that appears (“Ethernet 2”, in the case of the following Figure)



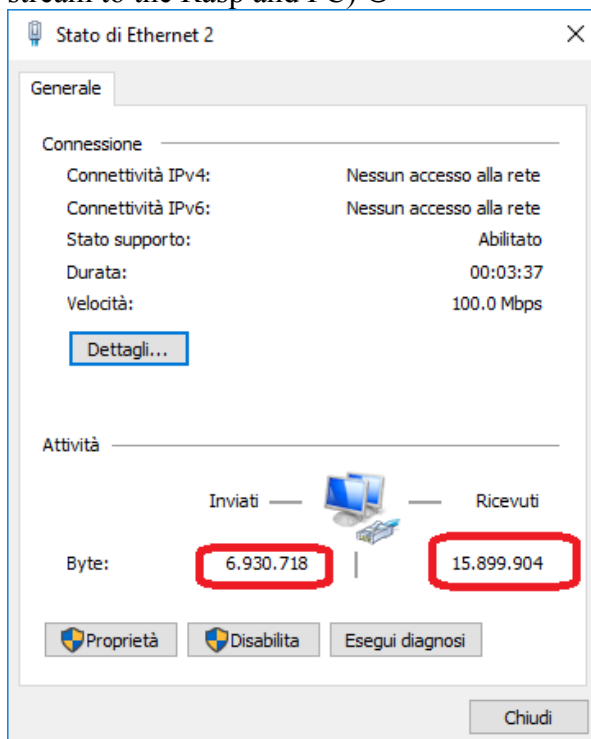
8. Click on the Property button → select TCP/IPv4 → click on Property → enter the IP address “192.168.20.2” and → click on the ok button to finish the procedure.







Wait for few minutes. If the procedure is successful, you will see a continuously increase of the received bytes (this means that gopro camera is transmitting video stream to the Rasp and PC) 😊



If the procedure is not successful, please remove the power supply from the raspberry and then connect them again after few seconds.

After, repeat the same procedure for the second Raspberry by using the IP address 192.168.30.2 for the new card network (see the step 7).

