3D modeling, reconstruction and analysis of environments assisted by multi-sensorial data processing

Department of Electrical and Information Engineering
ELECTRICAL AND INFORMATION ENGINEERING
Ph.D. Program
SSD: ING-INF/03 – TELECOMMUNICATIONS

Final Dissertation

3D modeling, reconstruction and analysis of environments assisted by multi-sensorial data processing

---

by
Vito Renò

Referees:

Prof. Danilo Mandic

Prof. Ioannis Pitas

Supervisors:

Prof. Cataldo Guaragnella

_____

Dr. Ettore Stella

_____

Coordinator of Ph.D Program:

Prof. Vittorio Passaro

XXIX cycle, 2014-2016

to everyone who believed in me

Abstract

The work described in this thesis falls in the general category of computer vision. More specifically, 3D modeling, reconstruction and analysis of environments is treated from multiple points of view in order to provide effective and efficient methods to capture data and perform complex processing tasks. Building a model by means of an automated machine vision system induces the research of constantly new techniques to make the final system both able to fulfill the requirements and optimized to efficiently perform proper tasks.

The problems that need to be solved relatively to these topics spread from the background modeling of a scene to moving object tracking, from 3D point cloud analysis to the identification of a motion, a trajectory or a particular feature of an object in the three dimensional space. All these tasks are related to open problems in the image/video processing field, since their efficient implementation is strictly related to the ability of a system to correctly represent a 3D complex scene or to the effective understanding of the semantics of an acquired video. For this reason, the main focus of this thesis is on the analysis of complex situations (i.e. indoor, outdoor, with and without controlled illumination, with many moving subjects) by means of innovative data acquisition and processing techniques.

Two types of point clouds actually exist – dense and sparse – depending on the number of 3D points that are contained in the acquisition. Both of them will be treated in this thesis, along with proper methodologies to deal with the specific type of data.

Laser based sensors are largely employed to create a 3D model of a generic scene, since they are able to provide detailed information (the so called Dense point clouds) about the depth of an object that is illuminated by the light source, guaranteeing one of the two following capabilities: high data throughput or large sensor field of view, up to 360°.

The first objective of this thesis will be the design, prototype and test of

a miniaturized omnidirectional 3D catadioptric sensor capable of both high throughput and large field of view. Also, a new methodology to perform 3D dense point clouds registration will be investigated and detailed. Such systems are of relevant interest and can be effectively employed in industrial applications for monitoring purposes, to perform non destructive tests, quality control or – more generally – objects analysis.

Another well known technique used to solve the 3D modeling problem is stereovision, that is used to evaluate the depth information about a point that is simultaneously captured by two or more sensors (cameras). Stereovision avoids the use of laser sources, but produces highly redundant data that needs to be exploited to compute a sparse point cloud. In fact, it is mandatory to collect and process at least two distinct videos to obtain 3D information about the scene, regardless the effort that is needed to estabilish the correct correspondence between points from distinct views.

The other objective of this thesis will be the design and development of a semi-engineered stereo system prototype for evaluating complex situations, applied to the sportive context (in particular, the tennis one). This system will be able to analyze game tactics of a specific player by logical inferences that will take place after having executed specific queries that should properly combine data extracted from each software module.

In summary, the aim of this thesis is the design and development of intelligent systems for the analysis of complex scenes by using 3D information. This leads to the study of novel techniques, as well as the optimization of known algorithms. In fact, once the 3D point cloud is extracted, it needs to be appropriately processed to perform, for example, the identification of an object or a subject, its tracking in the 3D space or the semantic analysis of the scene. The ability of interpreting a scene via software starting from the output of a camera or a depth sensor is an ambitious objective of certain scientific interest. Nevertheless, it is necessary to develop new methodologies as well as optimize and revise the known ones to achieve this goal, because semantic analysis highly depends on all the other software modules of the vision system (both 2D or 3D). Good models and effective processing algorithms are the keys to enable reliable high level modules on complex systems.

Contents

Chapter 1

Introduction

1.1 Dense Point Clouds

The problem of 3D reconstruction of environments founds application in many fields of computer science and engineering [1] – like video surveillance or robot automation [2] – and is particularly interesting for researchers and engineers coming from both academia and industry. The systems that are mostly used to solve this problem are typically based on laser profilometry or laser scanners, since both approaches are not affected by typical optical based system problems like, for example, illumination changes in a scene [3], [4]. Systems based on laser profilometry [5, 6, 7, 8, 9, 10] are basically composed of a laser source and a camera, whose position with respect to the source is properly calibrated. Supposing that the laser emits a line, the camera captures an image of that line that is altered by the object that is hit, allowing the system to compute the distance of each illuminated point of the line from the camera. These techniques are based on trigonometry and are known as laser triangulation. Even if these approaches can reach very high precision in the reconstruction task, the field of view is relatively small. This drawback can be overcome by using multiple pairs of laser and camera, but the resulting complexity in terms of calibration increases significantly [11, 12].

On the contrary, laser scanners measure the reflection of a point-like laser beam by means of a photo diode [13]. The difference of phase of the reflected beam with respect to the emitted one, or the time of flight estimation of the reflected beam are both techniques that enable the sensor to compute the distance of the illuminated object with respect to the laser source. One of

the limits of this technique is the emission of a point-like beam, that implies the measurement of one point at a time. However, this issue is easily resolved by employing a mechanical setup like a rotating mirror that reflects the beam on a circular trajectory, increasing the throughput of the sensor in terms of returned measures over time. Such systems do not show a limited field of view, but data acquisition speed is lower than laser profilometers whose throughput basically depends on the chosen camera. At the moment, these two approaches are the most used to obtain 3D point clouds [14] that are processed and employed in multiple applications, like for example:

- implementation of collision avoidance algorithms [15];
- resolution of navigation and localization problems [16];
- object tracking in the 3D space [17];
- or, more generally, robot control and 3D scene reconstruction [18, 19, 20, 21].

From a comparative analysis of these two approaches it is reasonable to assert that:

- laser scanners can inspect an environment with a field of view of about 360°, despite the low speed of data acquisition (a commercially available example is [22]);
- laser profilometers can reach a very high throughput that depends only on the camera, but with a sensibly reduced field of view.

Finally, another approach that can be used to deal with these problems is represented by catadioptric systems [23], that combine a certain number of mirrors and lenses along with laser and cameras. The union of a laser profilometer with a parabolic mirror and a camera can represent an effective solution to exploit the high acquisition rate of profilometers (limited only by camera temporal resolution) combined with the parabolic mirror that enlarges the field of view reaching performance typical of laser scanners [24, 25, 26]. This way, the strengths of both approaches can be exploited to perform three dimensional environment reconstruction. An example of such catadioptric sensor is the one proposed in [27], that is mainly composed by a laser source, a high framerate camera, a telecentric lens, a parabolic mirror and a moving vehicle (whose motion is supposed along the optical axis). However, the

practical use of this solution is limited by its huge dimensions. For this reason, one of the objective of this thesis will be the study and development of a miniaturized solution of this catadioptric sensor. The whole setup will be optimized by integrating the camera and the lens on a specific support created *ad hoc*, equipping a mobile robot and making it able to perform real acquisition. Also the calibration phase of this sensor will be regarded, as it is a mandatory step to obtain the desired level of precision and efficiency.

A miniaturized omnidirectional high throughput 3D sensor can be effectively employed in industrial applications for monitoring purposes, to perform non destructive tests, quality control or – more generally – objects analysis. The railway scenario can be one example, as trains represent a good test bed for such systems. Another example can be a car (or mobile robot) equipped with such sensor that should help in monitoring road surfaces (especially after adverse climatic events), subways, bridges or buildings. For this reason, a prototype will be tested on real life cases by acquiring specific targets, like:

- the 3D profile of a road or a building, in order to highlight roughness, holes or, more generally, defects or dissimilarities between multiple observations in time;
- the detail of scenarios that can be scanned along a pathway (railway tunnels or a covered parking), to perform structural monitoring.

along with a new methodology to perform 3D dense point clouds registration.

1.2 Sparse Point Clouds

Another particularly interesting research topic in the field of 3D image processing is stereovision, that consists of observing the same scene from multiple points of view in order to evaluate the 3D representation of what is being observed. Compared to the other methods discussed beforehand, this one is certainly applicable both in indoor and outdoor setups (because it only relies on cameras) but generally produces sparse point clouds, since it is mandatory to establish the exact correspondence of the same point across all the views that should be used to extract depth information [28]. This mechanism is the numerical equivalent of the human depth representation, since everyone that observes the world with two eyes (cameras) is able to estimate the distance of an object with respect to himself. Nowadays, many

systems rely on stereovision for telemedicine, robotics, video surveillance or cinema applications. However, regardless of the specific application, each stereo system needs some mandatory steps in order to be properly configured and perform an acceptable 3D scene reconstruction. Thus, artificial vision systems should be able to perform fundamental tasks [29, 30, 31], such as:

- background modeling;
- background/foreground segmentation;
- morphological filtering of foreground masks;
- construction of the ground truth of a scene;
- analysis and labeling of connected components;
- feature extraction;
- object tracking.

It is worth observing that all the information need to be properly combined to infer new knowledge. Among all the application contexts, the sportive one represents a good benchmark for developing and testing innovative algorithms that implement such tasks that have been listed beforehand. In fact, in literature there are many systems or algorithms that are employed to analyze tactics and teams in multiple sports – for example soccer – that start from the identification and tracking of both ball and players, the active entities during game. For example, in [32] a system that extracts useful information for coaches is presented, in [33] and [34] players' skills are evaluated while in [35] and [36] the overall game trend is analyzed. A multicamera (six cameras) approach to the activity recognition problem is presented in [37], where the authors estimate how long a player has been active during the game by means of posture analysis, ball control and eventual actions in which he has a strategic role.

Another objective of this thesis is the development of a semi-engineered system prototype for evaluating tennis game scenes, based on an artificial vision system. The principal requirements of such system are:

- recording of all game actions;
- identification and 3D tracking of active game entities (ball/players);

- identification and understanding of the whole game.

This system will be able to analyze game tactics of a specific player by logical inferences that will take place after having executed specific queries that should properly combine data extracted from each software module. The first problem to deal with is the hardware choice and setup that will be used to equip a private tennis test court. Hence, several factors need to be considered in order to correctly identify:

- the number of cameras to be installed;
- the type of cameras to install (in terms of sensor type, sensor size, framerate, etc...);
- the lenses;
- the position of the cameras with respect of the tennis court;
- the proper video acquisition and synchronization sub systems;
- the connection and communication protocol between camera and recording system;
- a custom illumination system, if needed.

Proper attention should be payed to the data acquisition task, since it is strictly related to the study of appropriate methodologies that will be used to process 3D data. It will be necessary to:

- identify and track the ball, i.e. recognize the object among multiple views at the same time in order to estimate its 3D position and reconstruct its trajectory in the 3D space;
- identify and track players to extract statistical parameters that enable further analyses, such as position and speed;
- recognize players' behaviors by combining high level data acquired during the game play.

The effective implementation of these features requires the application of advanced image processing techniques to perform segmentation, tracking and analysis of the interactions between the objects of the scene. Specifically it will be necessary to define and optimize:

- a proper background subtraction algorithm;
- the methodology to robustly identify the ball and the players in the videos;
- the algorithms to extract a sparse point cloud;
- the algorithms to track the ball in the 3D space;
- the algorithm to perform the scene understanding task and save the information in a database for further exploitation.

The description of these requirements gives a clear idea of a contextualized application of this research topic. However, it is worth observing that each software module can be also encapsulated on other kind of systems, not only related to sports. As a matter of fact, the 3D information extraction modules and data processing techniques that will be presented can be seen as a part of more complex high level expert systems that can be employed to analyze the behaviors of one or more subjects acting in a specific context.

Structure of the thesis

The thesis is organized as follows:

Chapter 2 contains the literature review related to each topic that will be covered in the subsequent chapters;

Chapter 3 presents an accurate miniaturized catadioptric range sensor that has been designed and developed. It combines a high-resolution and high-frame-rate camera with a telecentric lens that collects the laser light reflected by a parabolic mirror and is employed to perform the three-dimensional reconstruction of environments. The results presented in this chapter have been published in [38];

Chapter 4 presents an accurate method for the registration of point clouds returned by a 3D rangefinder. The method modifies the well-known iterative closest point (ICP) algorithm by introducing the concept of deletion mask. In this way, spatial regions of implicit ambiguities, due to edge effects or systematical errors of the rangefinder, are automatically found and neglected, lowering the errors made during the 3D registration task. The results presented in this chapter have been published in [39];

Chapter 5 is devoted to the design and implementation of real time algorithms that represent the building blocks of more complex 3D vision systems. In particular, three background models and a 3D tracking method will be detailed. The results presented in this chapter have been published in [40, 41, 42, 43];

Chapter 6 shows a system for the automated analysis of a sportive event applied to the tennis context. The system consists of a dedicated hardware setup (cameras and computer) and a number of software modules for the automatic processing of the recorded video sequences. The aim of the work is to support coaches in the evaluation of tennis players performance properly interpolating 3D data and semantic information about the context. The results presented in this chapter have been published in [44, 45];

Chapter 7 presents the conclusions of the thesis along with a description of future works.

Chapter 2

Literature review

2.1 Dense Point Clouds

2.1.1 3D Omnidirectional Range Sensor

The world today can be inspected in detail by exploiting three dimensional data thanks to the more recent technology developments. In fact, the information contained in a 3D image can be effectively used to address details about a specific target or an entire environment that should be analyzed. 3D data have been initially employed in robotics, where it is mandatory to build a complete map of the robot surroundings to perform self-localization or collision avoidance [46, 47, 48, 49]. Moreover, these kind of 3D images (or range images) gained increasing importance and interest among private companies to implement efficient quality control tasks. Industrial processes can be dramatically improved and speeded-up via unsupervised inspection techniques that are the basis for industrial standardization and automatized production of manufactured goods [50, 51, 52]. Finally, it is worth mentioning other kind of applications that today benefit from the exploitation of 3D data, such as medicine [53, 54], biology [55], archaeology [56, 57, 58], geology [59] or reverse engineering [60, 61]. Range sensors for three dimensional mapping of both indoor and outdoor scenes have been developed in the last years, basically relying on stereo imaging, time of flight principles, structured light or laser triangulation. Some commercially available products, along with their main features like acquisition rate, resolution, accuracy and precision are shown in Table 2.1.

Table 2.1: Detail of commercially available devices for 3D environment reconstruction

Model name	Accurange AR4000 [22]	RIEGL VQ-250 [62]	RIEGL VZ-400 [62]	Optech ILRIS [63]	Bumblebee BB2-08S2 [64]	Kinect v2 [65]
Type	Laser Scanner	Laser Scanner	Terrestrial Scanner	Terrestrial Scanner	Depth camera	Depth camera
Acquisition rate	50 kHz	50 kHz	122 kHz	10 kHz	1032 x 776 @ 20 Hz	512 x 424 @ 30 Hz
Maximum distance	2 m	180 m	350 m	400 m	10 m	4.5 m
Resolution	5 mm	Not reported	Not reported	Not reported	Not reported	2 mm
Accuracy	5 mm @ 9 m	10 mm	5 mm	4 to 7 mm @ 100 m	Not reported	1 mm
Precision	Not reported	5 mm	3 mm	Not reported	Not reported	Not reported
(I)n – (O)utdoor	I/O	I/O	O	O	I/O	I
Applications	3D environment reconstruction	Mobile mapping from moving platforms.	Large environments reconstruction end inspection	Large environments reconstruction end inspection	3D environment modeling	3D environment modeling

As highlighted in the previous chapter, stereo imaging [66] computes depth information starting from multiple views of the same scene, but its applicability its limited due to the need of point correspondence between the images and the complexity of mathematical models that are employed to triangulate the points. A certain number of sensors based on this principle has been proposed (e.g. [64, 67, 68]), but they are hardly employable in the context of dense point cloud extraction and environment inspection that involves precise measurements. For this reason, stereo vision is often used for qualitative real time analysis of sparse point clouds extracted by dynamic scenes [69, 70] and will be discussed further in Chapters 5 and 6.

Time of flight (ToF) range finders measure the distance of a target estimating the time elapsed between the emission of a laser beam and its reflection on the sensor. Many devices (e.g. [71, 72]) known as lidar, deflect the beam

with moving mirrors and are able to scan wider areas. For example, the AccuRange AR4000 [22] reflects a single laser spot with a rotating mirror that allows it to describe circular profiles. However, as introduced in the previous chapter, whenever a mechanical component is used, the throughput in terms of sample rate of the whole 3D system decreases. In fact, although the single spot could be acquired up to 50 kHz, the value is limited by the rotating mirror to 1 kHz, that correspond to few tens of profiles per second. ToF sensors try to overcome this limitation increasing the number of detectors – i.e. adding redundancy – as in the commercial products [73, 74]. In this case, the emitting lasers shed light over wider areas, whereas the matrix of detectors compute the phase difference between the sent signals and the returned ones. A depth image having the size of the matrix of detectors is thus computed in a single shot. Increasing the number of camera pixel, the corresponding equivalent sample rate can surge of orders of magnitude. On the contrary, the achievable field of view is implicitly bounded, so that multiple acquisitions are necessary to get a full mapping of the surroundings, with problems residing in the registration of the different unknown views. Moreover the cost of such systems is still impressive because of the number of laser sources illuminating the environment. Further commercial sensors, devoted to the home entertainment (Microsoft Kinect v2 [65]), employ diffused modulated light to illuminate the scene, thus downing the overall costs at expanses of the final measurement resolution.

Terrestrial Laser Scanners (TLSs) are devices used for the modelling of complex targets under outdoor conditions, with maximum ranges of hundreds of meters [62]. Such systems are based on the principles of time-of-flight or of phase difference and typically return range data as a function of the angular position of the emitted laser line. Their typical applications fall in the monitoring of extended areas for the detection of landslides and terrestrial deformations, or in the field of 3D reconstruction of cultural heritage sites [75, 76]. However, the main drawbacks reside in the huge cost of TLSs, their dimensions and weights and the limited Field-of-View (FoV) which makes them suitable mostly for long range measurements, and often not adaptable for several applications which require environmental scans of complex scenes.

Structured light patterns are often used to compute the 3D shape of objects, since they are deformed in accordance with the profile of the surface under investigation. Light patterns can be made of stripes (as in [77, 78]) or points (see [79]), whose distribution in the camera image is preliminary determined with reference to a calibration plane. Each alteration of the target

surface with reference to this plane returns a shift of the detected pattern, depending on the change of depth. The main limit of this technique resides in the mere indoor use, where fringes and spots are highly distinguishable. Outdoor application requires the use of coherent light, such as laser beams.

Laser profilometers follow the same principles of structured light, for which a laser line impinging a target is accordingly deformed. Knowing the relative position of laser and camera, triangulation laws can derive the position of the line in an absolute reference system [80]. As for the ToF range camera, the weakness of this technique is related to the bounded field of view of the sensor, which does not permit the full mapping of the sensor surroundings. For this reason mirrors can be added to collect a wider sight of the environment in a single frame. These complex systems belong to the category of sensors based on catadioptrics [81, 82, 83] and will be investigated to develop a miniaturized sensor, as described in Chapter 3.

2.1.2 3D Point cloud registration

The research on the use of laser scanners as a tool to produce 3D point clouds of complex scenes for structural engineering applications has received a great impulse thanks to the continuous improving of laser scanning technology. 3D geometric models from building, terrains, and infrastructure systems, can be used for preventing geological hazards, such as landslides, debris-flows, rockfalls and floods [84, 85]. At the same time, the high accuracy of measurements achievable with 3D models permits the reliable check of the conditions of existing buildings and roads [86, 87, 88, 89, 90, 91, 92, 93, 94].

In the context of infrastructures monitoring, registration of point clouds is a crucial preliminary step to compare data acquired at different epochs and to document changes and geometric deformations of the observed surfaces. The capability of the processing methods to detect variations is strictly dependent on the registration process which has to transform all acquired point clouds to a common coordinates system. In the chapter, the problem – crucial for infrastructures monitoring – of developing a point cloud registration approach which improves the accuracy of 3D data alignments when reliable results are required will be addressed.

Generally speaking, point cloud registration refers to two categories of problems:

- the precise localization of navigation systems during the acquisition of

the dense 3D models of targets;

- the exact matching of datasets acquired at different epochs for structural monitoring.

The registration of laser scans for the creation of dense 3D models can be increasingly performed by matching the newest scan over the acquired ones while the surroundings are sensed. In this context, the literature on 3D scene recovery is mainly related to trajectory-based methods. Among these methods for laser scan matching, those based on Self Localization And Mapping (SLAM) are the most used [95, 96, 97]. They can produce dense 3D models in real time by updating an even more detailed map of the scene together with the information on the sensor position. As an example, a more sophisticated method [98] exploits the knowledge on the topology of the scene to simultaneously update both the 3D map and the sensor pose by means approximate surface reconstructions.

On the other hand, point cloud registration for structural monitoring is aimed to align different datasets, even acquired with SLAM methods, in order to achieve meaningful comparisons on a common reference system. These approaches can be classified as follows [99]: marker-based, sensor-based and data-driven registration methods.

Marker-based registrations can be very precise but require that artificial control points, with an easy-to-recognize pattern, are placed in the scene [100, 101, 102].

In the sensor-based category [103, 104], the position and orientation of the scanner is determined by GPS and an Integrated Measurement Unit (IMU), limiting the application of these methods to outdoor contexts, where the lines of sight to the GPS satellites are not occluded.

Data-driven approaches use the point clouds properties to find the registration parameters. A widely used algorithm belonging to this category is the ICP (Iterative Closest Point), originally introduced in [105, 106]. Given two clouds of points (a reference and a source), the algorithm finds 3D correspondences between them and tries to determine the translation and rotation matrices whose application to the source can lead to the best match on the reference in terms of minimum distance. Although the method is simple and easy to implement, a drawback resides in the need of a user control for the validation of results, since it often reaches a wrong convergence. Specifically, an erroneous point correspondence between the source and the reference can

increase the value of the distance function under optimization, even if the models are overlapping.

Many techniques have been presented to overcome this problem, such as: using the calibration equation of the sensor [107]; weighting the input surface depth data for the integration of the views in a continuous surface [108]; including color information, if available, or more generally intensities, in the comparison of the datasets [109, 110, 111]); extracting invariant features for the selection of points [112]; applying geometrical constraints on the point collinearity and closeness [113]; employing a global consistency measure to detect incorrect, but locally consistent matches [114]; using general-purpose non-linear optimization, such the Levenberg–Marquardt algorithm [115]. At the same time many speeded-up variants of this method have been also presented [116], including the approximation of the nonlinear optimization problem with a linear least-squares one [117] and an efficient evaluation of the meaningful points [118]. All these techniques can be also used in the case of registration of scans which are individually subjected to local deformations [119].

In Chapter 4 a data-driven approach for the 3D registration of point clouds acquired at different epochs will be detailed.

2.2 Sparse point clouds

2.2.1 Real time algorithms for high throughput data processing

Background models

The analysis of high-throughput data has always been a challenging task in computer science, especially in the computer vision field. Any artificial vision system must deal with the background (BG) modeling as a low level computational task: the model must be fast, reliable and fault adaptive if a real time processing constraint has to be satisfied. Moreover, the background/foreground segmentation is generally the first operation that an intelligent system has to implement. It represents the input for many other modules that can do for example object tracking, gesture analysis or semantic interpretation of the scene. Therefore, an artificial vision system has to be improved by doing this segmentation in real time. Generally speaking, background models

can be classified as *Temporal difference methods* and *Background subtraction methods*: the first ones are based on the idea that the foreground objects can be obtained subtracting two consecutive frames and applying a threshold to the output image; the second ones build a dynamic model and subtract it to the following frames that have to be processed. The BG model is usually updated over time in order to adapt it to the environment changes.

One of the most used BG methods is the Adaptive Mixture of Gaussians (MoG) proposed by Stauffer and Grimson [120] and its subsequent improvements. This algorithm uses Gaussians distributions to represent the variation of pixel intensity, so it is suitable to model complex and dynamic backgrounds. Moreover, Zivkovic [121] has introduced a technique to adaptively update the parameters of the MoG algorithm. Other BG algorithms known in literature are, for example:

- the Eigenbackground [122] introduced by Oliver et al., where the model is a PCA-based subspace representation of a certain number of static frames that represent the background;
- the Codebook [123] proposed by Kim et al., that implements a quantization of the pixel values using codebooks in order to compress the model size;
- models that use Hidden Markov Models (HMMs) [124] to represent pixel intensity variations as discrete states.

Many algorithms work well under varying light conditions or during dynamic scenes, but are computationally complex or threshold-dependent. Nonparametric models as the GMG by Godbehere et al. [125] try to resolve this dependence estimating the entire pixel intensity distribution rather than its parameters, using dynamic information. Only the probability distributions associated with background pixels are finally updated.

The advent of smart cameras with embedded processing units involves intelligent vision systems design improvements, because some tasks, such as the background/foreground segmentation, can be directly implemented on the camera. An example is the Adaptive light-weight algorithm presented in [126], whose application in athletic video processing is of relevant research interest. According to [127], this type of scenes can be used to:

- detect relevant events, for example offsides or goals during football matches [128, 129, 130];

- analyse and track relevant objects, for example ball and players;
- do 3D reconstruction;
- analyse tactics and so on.

Moreover, the athletic video processing represents one of the most challenging real time applications, because, in this context, there are many critical aspects that should be taken into account when designing a BG model: no a-priori knowledge of the static scene, sudden illumination changes and many moving objects that slow down the upgrade phase. In order to solve these issues a reliable computer vision system has to:

- build the model without bootstrapping frames;
- be responsive to light variations;
- be fast in the updating phase.

In Chapter 5 particular attention will be given to the design and implementation of three background algorithms able to deal with high throughput data in real time.

3D Tracking

Research in sports field considerably gained importance in the last decades due to meaningful technology improvements that totally changed our way of thinking. A huge number of sensors is now available for almost everyone and reasonably cost-effective solutions are provided to bring digital innovation inside sports. In addition, the constant growth of devices' pervasiveness provide "fertile ground" for investigating new and optimized techniques for extracting significant information from the integration of multiple data streams [131]. Many applications like tactics analysis, automatic key events identification and highlights extraction as well as statistical approaches to game evaluation and human performance analysis have been developed and represent the *desiderata* for specialized software that helps coaches, players and involved people in both supporting and enjoying innovation in sports [132] [133] [134] [129].

A key role is undoubtedly played by computer vision because cameras – either monocular or arranged in stereoscopic configurations – represent one

of the most used sensor technology in almost every sport [135]. It is worth pointing out that problems such as object recognition and tracking are well known in scientific literature, but the need of achieving high performance is moving researchers towards the integration of domain knowledge inside machine vision algorithms [42] [136]. Sports involving moving ball and players like soccer or tennis represent a good test-bed for intelligent agents that have been successfully implemented to extract and understand active entities and their relationships during the game [137]. In addition, whenever a sport is followed by a large audience both broadcast and private videos are available and can be used as feeds for algorithms [138]. These videos represent a vast dataset that can be exploited to perform a certain number of tasks by working on single camera bi-dimensional recordings.

A remarkable work in this field has been done by Yan et al., that introduced a methodology to process low quality single camera videos by enhancing low level elliptical features to identify and track a tennis ball with the aid of a modified particle filter [139]. Moreover, the same authors improved tracking efficiency overcoming the issues introduced by Robust Data Association (RDA), a non-iterative tracking algorithm inspired by the well known RANSAC approach [140] [141]. Basically, the authors claim that the RDA approach suffers from growing complexity issues in cluttered environments and is based on independent models estimated over time. For this reason, they introduce the concept of seed triplet instead of pure random sampling, fitting the models only after evaluating a small ellipsoid that should embrace points that are likely to be part of the same trajectory. The final step is a layered data association method that exploits graph theory to link pieces of trajectories that have been recognized. In recent years other sophisticated approaches have been developed, trying to combine machine learning techniques with computer vision ones [142] or extending Kalman filter theory and employing a neural network approach to predict ball trajectories [143].

Figure 2.1: Example of a private tennis court equipped with a monocular camera.

Bi-dimensional approaches can be useful to solve tracking problems at image plane level or coarse key events annotation, but more sophisticated applications may not fulfill the requirements imposed by coaches. An algorithm that runs only on a single camera that covers a large court can not guarantee the same performance on every zone of the court. Looking at Fig. 2.1 it is immediate to notice that the zone opposite to the camera is penalized with respect to the nearest one. Moreover, the lack of the third dimension makes data not usable for high precision applications such as line calling or player performance analysis in terms of posture or behavior. For this reason, in these cases dealing with 3D data become a mandatory step. Pingali et al. can be considered as pioneers of multi camera real time ball tracking system based on six cameras arranged to form four stereo pair per each side of the court [144]. In that work player-ball interactions and ball occlusions were left as future works, as well as the improvement of tracking accuracy. Today, the *de facto* standard for 3D tracking and line calling is represented by the Hawk-eye technology, whose preliminary results were presented in [145] and now is available as a service provided by a dedicated company [146]. The system achieves extremely precise results (mean error rating of 2.6 mm) as it employs 6 to 10 cameras to equip the court. It is reasonable to say that with such setup the ball is visible at almost every frame.

However, in this work particular attention is focused on semantic data fusion for ball recognition and tracking purposes applied to tennis, adding domain knowledge to 3D data coming from a stereo system made of only 4 cameras arranged in two stereo pairs, one per each half court. This configuration is less precise than the Hawk-eye system, but hardware reduction enables

to lower the costs in terms of time, space and money. Many applications aimed to semi-professional players can take the advantage of a 3D system, for example if the coach needs to analyze how the player is performing a particular stroke. Of course the ball is not always available in the three dimensional space, since the lack of information in just one camera makes the 3D estimation not possible. For this reason, an algorithm that recovers and interpolates the most probable trajectory is essential. Details will be given in Chapter 5.

2.2.2 A technology platform for automatic high-level tennis game analysis

Sports analysis can provide a complete survey of sport events to interested parties. This kind of systems produces objective feedback helping players and coaches to improve performance in a field that is competitive by nature. For this reason, several commercial solutions are available, sometimes addressing analysis in more than one sport discipline. Most of them provide only support for manual annotations of video sequences. Manual annotations can either be done off-line or in real-time. Dartfish Video Analysis Software [147] and Sportscode Performance Analysis [148] are examples of commercial systems in which video sequences are manually annotated off-line on desktop-class computers with the latter being used by important football clubs [149]. Match Analysis [150] is a further example where manual annotations are created, although in this case the operation is remotely outsourced to other companies. TenniVis is a tennis match visualization system that relies entirely on data such as score, point outcomes, point lengths, service information, that can be easily collected by a human operator watching videos that are captured by one consumer-level camera [151]. Other systems [152, 153] offer support for smart-phones and tablets, while still requiring manual annotations.

Few systems try to provide a solution to sport analysis without requiring human supervision. A recent trend is represented by the acquisition of data directly from the player using wearable devices in several sports in general [154, 155] and in tennis in particular [156, 157]. However, intrusive systems can be either sensible to signal collisions and interferences for operating and communicating in real-time [158], or are limited to off-line processing [159, 160]. Additionally they are rarely accepted by players as they have to be small enough to be comfortable and not perceived as an obstacle to their

movements and performance [161]. Non intrusive solutions are based on broadcast cameras or dedicated cameras placed around the game court and use computer vision techniques to process the acquired videos. ProZone [162] provides automatic video analysis for soccer and rugby. This system is based on the automatic processing of NTSC/PAL video. The system can operate in almost every professional match broadcasted in TV. Human intervention is sometimes required to correct errors done by the system. TennisSense [136] has been developed to use custom-installed cameras, optimized for automatic processing. The system has been designed and developed by Dublin City University in partnership with Tennis Ireland, the Irish tennis governing body, using a UbiSense spatial localization system and requiring the installation of nine IP cameras with pan, tilt and zoom capabilities, surrounding the instrumented tennis court. Cameras position and setup are optimized to cover specific areas and perform specific tasks. Ball and players tracking is therefore performed synchronizing and fusing these data streams. A system, operating in real time and aimed at enhancing broadcasts as well as coaching activities, is proposed in [163, 144], where computed motion trajectories, along with compressed video streams, are stored in a database system. The system proposed in [163] also provides a way to customize information to be shown using a proprietary Application Programming Interface (API).

Other works focus their attention on a more limited set of topics, such as stroke detection or ball trajectories reconstruction. In [164] strokes are detected and recognized through player tracking and skeletonization, although under restrictive assumptions. Ball trajectory is the focus of the work described in [165], that is performed on soccer matches using broadcast video. Novel in this work is their focus on recognizing the ball through the evaluation of the followed trajectory rather than its low-level visual features. The ability of discerning event cues starting from the evaluation of ball trajectories is the focus of the work [139] on broadcasted tennis matches, enabling therefore automatic annotation of broadcasted videos. Issues on the reconstruction of ball trajectories are also common in table tennis games, with the aggravating problem imposed by frequent occlusions between ball and racquet. The paper [166] addresses this challenge through the evaluation of trajectory planes. Misdetection and abrupt changes of ball trajectories are addressed in [140] using a layered data association scheme. Last but not least, ball tracking can be done in 3D using a physics-based approach (as in [167]), when sports events are acquired using multiple synchronized views.

In this work an innovative approach for event recognition such as strokes,

bounces and serves will be proposed. It is based on the analysis of the reconstructed 3D ball trajectory which can be used for automatic annotations of video sequences and high level semantic analysis. The extracted action sequences with the associated data can effectively support coaches for the evaluation of game tactics and for improving players performance. Details about this system will be given in Chapter 6.

Chapter 3

3D Omnidirectional Range Sensor

As introduced beforehand, the main idea of the proposed sensor setup comes from the contributions described in [26, 25, 27] – where an omnidirectional sensor for high-resolution 3D mapping has been proposed – but adding compactness and applicability to the whole approach. Here a laser profilometer assisted by a parabolic mirror is designed to reconstruct spaces when a mobile robot flows through them. The achievable resolutions ($10mm$ at $5m$ of distance from the laser source) have demonstrated the capability of the previous approach to precisely model both indoor and outdoor scenes, going beyond the mere 3D mapping devoted to robot navigation and obstacle avoidance. Previous results have enabled novel applications, such as the detection of wall cracks or the prevention of geological hazards, as landslides and rockfalls, just to mention a few. A step forward in the sensor development consists of reducing the size of the whole experimental setup, without altering the final accuracy. In fact, downsizing the setup enables the possibility of using it in further applications, including pipe inspection and monitoring of dangerous and confined spaces. For this reason the prototype has been completely redesigned with state-of-art devices (lasers and camera) able to further increase the acquisition rate. Furthermore, the calibration phase has been eased by means of a novel numerical approach for the exact computation of the actual parameters involved in the measurements. In this way, precise mechanical alignment of the optical components which constitute the system is no longer required.

3.1 Working principle

This Section aims to describe the working principles of the presented setup, showing how the components cooperate to sense the surroundings. Starting with the description of the setup components, this Section flows through the investigation of the mathematical formulations that lead to the design of the sensor prototype.

3.1.1 Geometry description

The proposed sensor is designed to map environments with high resolutions and high frame rates, exploiting the principles of laser triangulation. Although profilometry is a rather simple way to retrieve the 3D shape of objects, or more generally of any surrounding, its fundamental limit resides in the short available FoV. In the simplest case of a laser generating a line over the target and a receiving camera, which directly looks at the illuminated surface, the FoV is limited by the sensor width times the lens magnification. Since short-focal lenses are not suitable for measuring because of the huge distortions, the magnification is not small enough to increase the FoV to a full representation of the environment. To achieve this result exploiting the advantages of laser profilometry, it is mandatory to increase the component redundancy or combine one or more mirrors with one or more cameras. These systems are referred as catadioptric systems.

In general, catadioptric systems are made of a standard camera, with perspective or orthographic projection models, pointing upward a convex mirror (parabolic, hyperbolic, conic, etc.). As a consequence the FoV of the camera is opened to the surrounding regions beyond the limit imposed by the camera lens. On the other hand, such systems introduce deformations of the acquired images. As a consequence, image distortions have to be compensated to produce effective measurements, taking advantage of the knowledge of the mirror equation.

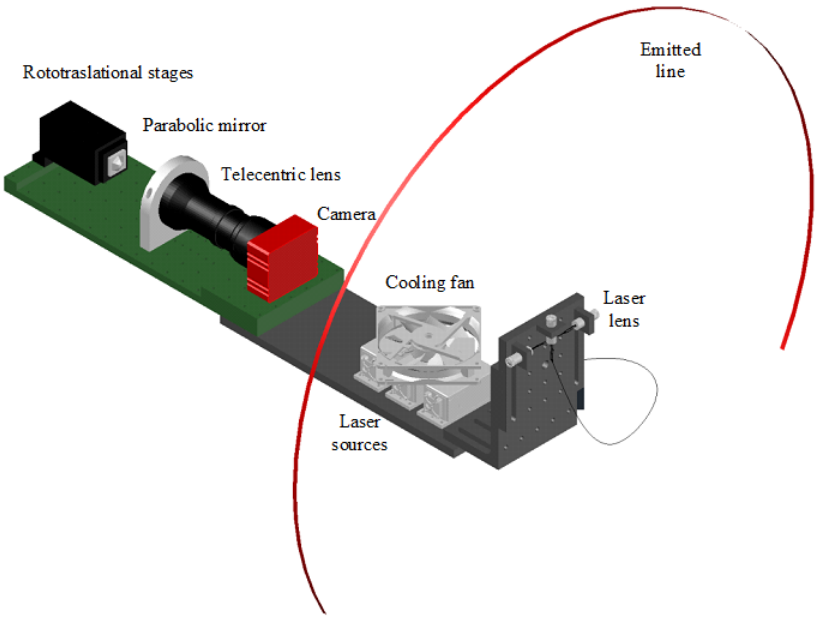

Figure 3.1: Sketch of the presented laser profilometer. Note the parabolic mirror is mounted onto micrometric rototranslational stages, whereas the camera is fastened on the metallic stand. Lasers are placed across from the parabolic mirror, behind the camera.

Following the approach described in [27], the proposed sensor falls in the category of the catadioptric laser profilometers, since it is made of a laser source, a parabolic mirror and an optical receiver. With reference to Figure 3.1, three laser sources are used to emit light, forming a plane with an overall fan angle of 270° (90° each laser). When the light strikes a target of the surrounding environment, a complete line is displayed on the surfaces. Each point of the line describes a measurement sample where the scene will be mapped in the global reference system. The parabolic mirror deflects light on the camera plane, throughout the lens. Since a parabolic mirror reflects light always following directions parallel to its axis of symmetry, a telecentric lens is the best candidate for the image formation. The resulting image has information about the position only of the illuminated targets. It is worth noticing that the sensor must be aided by an encoded movement to perform a complete scan of the whole environment. For this reason a mobile vehicle is used to carry the sensor through the scene, sense its spatial pose via standard odometry and send this information to the data collector.

Once the fundamental active devices are chosen and arranged in the setup,

it is mandatory to derive the triangulation laws that govern the process of image formation on the camera. This aspect will be treated in the next section.

3.1.2 Triangulation equations

The aim of the proposed range sensor is the measurement of distances starting from the inspection of where the laser line is displayed in the image. The next steps are derived following the notation reported in Figure 3.2, where the setup scheme is proposed. Here the reference system (x, y, z) is centered in the vertex of the parabolic mirror, having symmetry axis along the z-direction and focus at coordinates $(0, 0, F)$. It follows that the parabolic mirror has equation:

$$z = \frac{1}{4F}(x^2 + y^2) \tag{3.1}$$

The laser plane intercepts the z-axis at b (baseline), whereas the camera plane intercepts the z-axis in WD (working distance). For the sake of simplicity, Cartesian (x, y, z) and polar (ρ, Θ, z) coordinates are both used within the next lines to refer the points in the world absolute system. Finally, the camera plane has a proper 2D reference system (x', y'), assisted by the corresponding polar coordinates (r, ϕ).

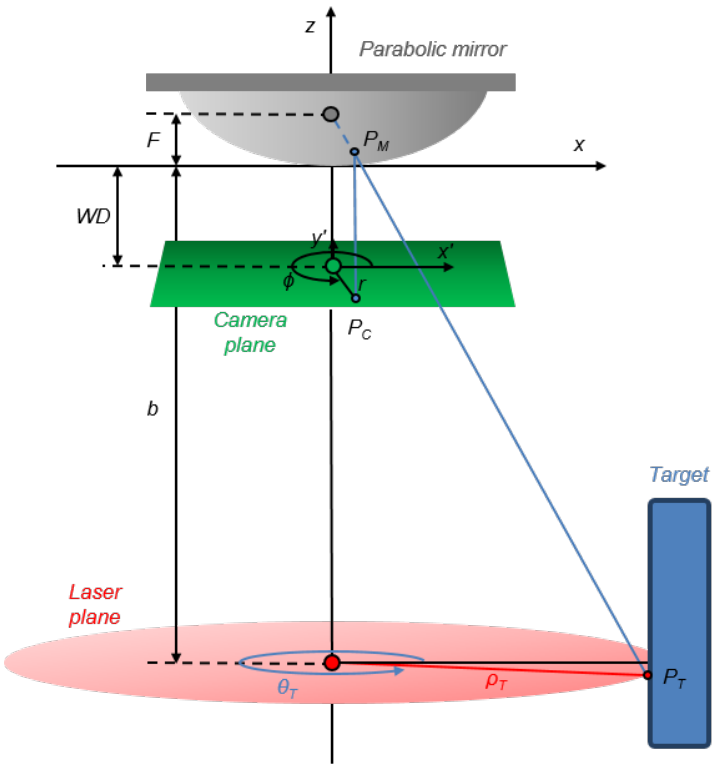

Figure 3.2: Schematic view of the proposed setup. The final goal of the analytic model is the translation of the known coordinates (r_C, ϕ_C), recovered on the camera plane, in world coordinates (ρ_T, Θ_T, z_T), having origin in the mirror vertex.

When the laser plane strikes a target, a line emerges. Each point P_T belonging to the line has coordinates (ρ_T, Θ_T, z_T) and is detected on the camera plane in P_C, having coordinates (r_C, ϕ_C) in the reference system of the camera (x', y'). In summary, the problem can be formulated in deriving the world coordinates (ρ_T, Θ_T, z_T) knowing the terms (r_C, ϕ_C). The following steps start from two fundamental initial hypotheses:

1. Both the laser and camera planes suffer from negligible alterations of their normal vectors with respect to the z-axis. This implies that the laser line is always across from the focus of the paraboloid, behind the camera. Equivalently each point of the line can be detected on the camera plane whenever its sight is not occluded by other objects;

2. The z-axis crosses the image plane in its center, or equivalently the

vertex of the mirror is displayed in the center of the image plane. This condition will lead to a simplification of the model, as the image projection can be referred in both absolute and camera reference systems. In other words, the point P_M is projected in P_C keeping the transversal coordinates $(x_C, y_C)|_{(x,y,z)} = (x_C, y_C)|_{(x',y')}$. The last condition is valid when the magnification M of the lens does not scale the metric coordinates. Otherwise, the term M has to be added to the formulation as a multiplicative factor.

It is easy to understand that the calibration phase has to be run to ensure the meeting of the initial hypotheses. As an example, the mirror has to be placed properly in order to achieve its centering in the image plane. These procedures will be further described in the next section. Any generic point P_T of the laser line produces a reflection on the parabolic mirror at coordinates defined by the point P_M. Because of the properties of a parabolic mirror, the projection of the laser spot onto the mirror is equal to the intersection of the ray that connects the spot itself with the focus of the paraboloid with the paraboloid itself shown in Equation 3.1. This ray has equations:

$$\begin{cases} x &= \dfrac{\rho_T \cos\Theta_T (F - z)}{F + b} \\ \\ y &= \dfrac{\rho_T \sin\Theta_T (F - z)}{F + b} \end{cases} \tag{3.2}$$

The corresponding analytical system, result of the ray incidence on the mirror, admits two solutions of P_M:

$$P_{M,1} = \begin{pmatrix} -\dfrac{2F\cos\Theta_T}{\rho_T}\left(F + b + \sqrt{\rho_T^2 + (F+b)^2}\right) \\ -\dfrac{2F\sin\Theta_T}{\rho_T}\left(F + b + \sqrt{\rho_T^2 + (F+b)^2}\right) \\ \dfrac{F}{\rho_T^2}\left(F + b + \sqrt{\rho_T^2 + (F+b)^2}\right)^2 \end{pmatrix} \tag{3.3}$$

$$P_{M,2} = \begin{pmatrix} -\dfrac{2F\cos\Theta_T}{\rho_T}\left(F+b-\sqrt{\rho_T^2+(F+b)^2}\right) \\ -\dfrac{2F\sin\Theta_T}{\rho_T}\left(F+b-\sqrt{\rho_T^2+(F+b)^2}\right) \\ \dfrac{F}{\rho_T^2}\left(F+b-\sqrt{\rho_T^2+(F+b)^2}\right)^2 \end{pmatrix} \tag{3.4}$$

Both solutions are valid in the set of real numbers, but only one of them is physically possible. In particular, the geometry of the system imposes a strict constraint: only that point that hits the mirror at the lowest z-coordinate is solution of the analytical system. It follows that $P_{M,2}$ (from now on P_M) solves the specific problem. Consequently, the coordinates of P_C on the camera plane are:

$$P_C = \begin{pmatrix} M\dfrac{2F\cos\Theta_T}{\rho_T}\left(\sqrt{\rho_T^2+(F+b)^2}-(F+b)\right) \\ M\dfrac{2F\sin\Theta_T}{\rho_T}\left(\sqrt{\rho_T^2+(F+b)^2}-(F+b)\right) \\ -WD \end{pmatrix} \tag{3.5}$$

being WD the nominal working distance of the lens-camera set. The transversal coordinates can be also expressed in polar coordinates, thus giving (r_C, ϕ_C) equal to:

$$\begin{cases} r_C &= M\dfrac{2F}{\rho_T}\left(\sqrt{\rho_T^2+(F+b)^2}-(F+b)\right) \\ \phi_C &= \Theta_T \end{cases} \tag{3.6}$$

Since the final goal of the presented framework is the estimation of (ρ_T, Θ_T, z_T) knowing the terms (r_C, ϕ_C), the relationships in Equation 3.6 have to be

inverted, thus obtaining:

$$\begin{cases} \rho_T & = \dfrac{4MF(F+b)}{4M^2F^2 - r_C^2} r_C = \dfrac{1+4ab}{1-4a^2r_C^2} r_C \\ \\ \Theta_T & = \phi_C \end{cases} \tag{3.7}$$

where a is the curvature of the parabolic mirror, equal to $\frac{1}{4F}$.

The results in Equation 3.7 are thus able to transfer the points belonging to the laser line detected on the camera plane in an absolute reference system.

3.1.3 Design strategy

Starting from the deep knowledge of the triangulation laws in 3.7, a prototype can be designed in terms of selection of active devices, namely camera and laser sources, and passive components, i.e. telecentric lens and parabolic mirror. The geometrical and physical parameters involved in the actual design of the experimental setup are reported in Table 3.1.

Table 3.1: List of geometrical and physical parameters involved in the range measurements.

Components		Parameter name	Description
Passive	Mirror	a	Curvature of the mirror $[m^{-1}]$
	Lens	M	Magnification of the lens
Active	Camera	$W \times H$	Camera resolution
		p	Pixel size $[m]$
		f	Frame rate $[s^{-1}]$
	Laser	b	Baseline $[m]$

The choice of the model parameters in Table 3.1 is linked to a set of initial specifications:

- the maximum measurable range d_{MAX};
- the maximum acceptable uncertainty in range estimation $\Delta\rho_{T,MAX}$ obtained at $\rho_T = d_{MAX}$;
- the number of profiles per second that are returned by the sensor (herein Profile Acquisition Rate, PAR).

The estimation of the device parameters starts with the analysis of the specified PAR. In particular, this requirement defines a first and unavoidable constraint on the choice of the camera, which is the only device responsible for the measurement rate. On the other hand, the requirements on the measurement quality have effects on the choice of the mirror equation, in terms of its curvature a, and of the baseline b between mirror and lasers. Also the lens magnification M has to be defined properly in order to adapt the properties of the camera (pixel size and resolution) to the specific problem under analysis.

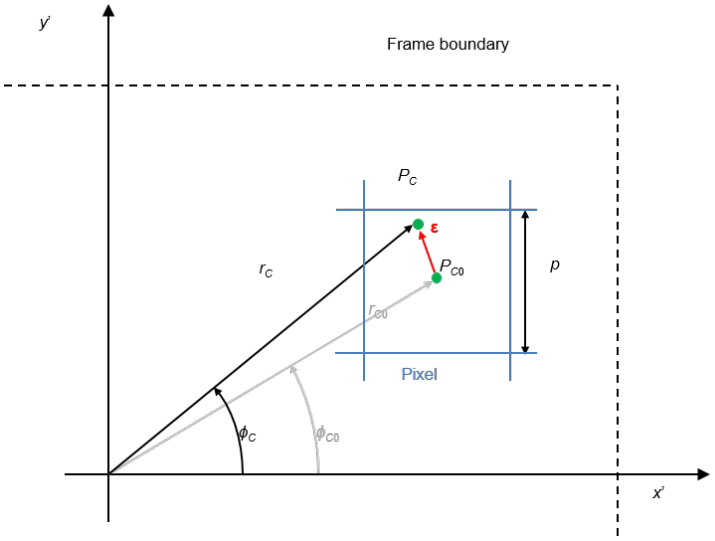

Figure 3.3: Error components related to the quantization of the image plane due to the finite area of pixels. Each pixel of the plane is square and has side equal to the pixel pitch p

In this context, errors are ascribable to the quantization induced by the matrix of pixels on the camera plane. Figure 3.3 shows a sketch of the quantization and the corresponding effects on the determination of the beam coordinates. In particular, for any point P_C, projection of the laser line within the pixel area, the resulting actual coordinates (r_C, ϕ_C) are always associated to the coordinates of the center of the illuminated pixel (r_{C0}, ϕ_{C0}). The error contribution can be described by the vector ε, which has origin in the center of the pixel P_{C0} and ends in P_C, corresponding to the actual range measurement. The pixel area determines a region of uncertainty. This region

can be shifted in the absolute reference system, thus defining an ambiguous spatial region where differences in (ρ_T, Θ_T) cannot be resolved. In this case, the measurement is:

$$\begin{cases} \rho_T &= \rho_{T0} + \Delta\rho_T \\ \\ \Theta_T &= \Theta_{T0} + \Delta\Theta_T \end{cases} \tag{3.8}$$

where $\Delta\rho_T$ and $\Delta\Theta_T$ refer to the range and angular uncertainties. The following formulations aim to detect the worst condition for the measurement, or equivalently the highest contribution of the error vector ε to the coordinates (r_C, ϕ_C). It is easy to understand that the vector ε has maximum modulus when the point P_C exactly lies on the corners of the pixel area. In this case the modulus is equal to half the diagonal of a pixel, i.e.:

$$|\varepsilon|_{MAX} = \frac{p\sqrt{2}}{2M} \tag{3.9}$$

It is mandatory to observe that the pitch term p in Equation 3.9 has been divided by M before being reported in the world reference system. In the following lines, the ratio $\frac{p}{M}$ will be named as effective pixel size p'. In a similar manner, when P_C lies on the corners of the pixel area, the uncertainty in the determination of Θ_T experiences its lowest or highest values. Also in this case, the peak of uncertainty is reached along the pixel diagonal, which represents the maximum range of angles that can be spanned within the pixel itself. In summary, given the extension of the pixel diagonal and the analytic model derived before, the maximum error can be directly estimated at a specific region of the mirror, or, equivalently, at every distance from the laser sources. Following Equation 3.7, the generic pixel of coordinates (r_{C0}, ϕ_{C0}) corresponds to a target placed at position:

$$\begin{cases} \rho_{T0} &= \dfrac{1 + 4ab}{1 - 4a^2 r_{C0}^2} r_{C0} \\ \\ \Theta_{T0} &= \phi_{C0} \end{cases} \tag{3.10}$$

As effect of the image quantization, the returned measurement is affected by the two contributions of uncertainty, $\Delta\rho_T$ and $\Delta\Theta_T$. Given the hypothesis in Equation 3.9, the expression of $\Delta\rho_T$ can be easily derived as:

$$\Delta\rho_T = \frac{1 + 4ab}{2} \frac{2r_{C0} + p'\sqrt{2}}{1 - a^2\left(2r_{C0} + p'\sqrt{2}\right)^2} - \rho_{T0} \tag{3.11}$$

which leads to:

$$\Delta\rho_T = \frac{p'\sqrt{2}}{2}\frac{1+4ab}{1-4a^2r_{C0}^2}\frac{1+2a^2\left(2r_{C0}+p'\sqrt{2}\right)r_{C0}}{1-a^2\left(2r_{C0}+p'\sqrt{2}\right)^2} \tag{3.12}$$

Equation 3.12 can be further manipulated to derive the expression of $\Delta\rho_T$ as a function of the measurement ρ_{T0}. This result can be easily obtained inverting the first equation of 3.10:

$$r_{C0} = \frac{\sqrt{(4ab)^2 + 16a^2\rho_{T0}^2} - (1+4ab)}{8a^2\rho_{T0}} \tag{3.13}$$

At the same time, the maximum angular uncertainty in target measurements can be derived knowing the coordinates of the point P_C and P_{C0} in the $x'-y'$-plane and how they are related to the pixel size. For instance, if P_C lies on the north-west corner of the pixel depicted in Figure 3.3, it is possible to derive $\Delta\Theta_T$ as follows:

$$\Delta\Theta_T = \arctan\left(\frac{2r_{C0}\sin\Theta_{T0}+p}{2r_{C0}\cos\Theta_{T0}-p}\right) - \Theta_{T0} \tag{3.14}$$

which can be further developed as a function of the range measurement, by replacing the expression in Equation 3.13. Equations 3.12, 3.13 and 3.14 are necessary but not sufficient to achieve the complete design of the sensor, which requires the last constraint: the mirror has to be in the FoV of the selected camera. When the mirror is acquired by the camera, its edges define a circle of diameter D_M. It is easy to understand that this area has to be included within the camera plane in order to be captured, and, consequently, the mirror diameter has to be at least equal to the smallest size of the camera sensor. Specifically, being W and H the number of pixels along the horizontal and vertical directions ($H \leq W$), D_M has to be equal to $h = H \cdot p'$. Since the sensor has to return measurements at a maximum distance d_{MAX} from the laser sources, equation 3.13 can be rewritten imposing that a laser beam, impinging on a target at distance d_{MAX}, is detected on the most external regions of the mirror. Mathematically, this condition leads to impose that $r_{C0} = \frac{D_M}{2}$ when $\rho_{T0} = d_{MAX}$. This can be exploited to define the unknown baseline b as a function of the mirror curvature a:

$$b = \frac{2\left(1-a^2h^2\right)d_{MAX} - h}{4ah} \tag{3.15}$$

As a consequence, the design can be shifted to the evaluation of the unknown a, which is the only term that has to be dimensioned to match the specification on the maximum error. Equation 3.12 can be developed considering $r_{C0}|_{\rho_{T0}=d_{MAX}} = \frac{D_M}{2} = \frac{h}{2}$, together with the expressions 3.9 and 3.13, thus obtaining:

$$a = \sqrt{\frac{h\Delta\rho_{T,MAX} - p'\sqrt{2}d_{MAX}}{h\left(h + p'\sqrt{2}\right)\left(h\Delta\rho_{T,MAX} + p'\sqrt{2}(d_{MAX} + \Delta\rho_{T,MAX})\right)}} \tag{3.16}$$

Note that only the positive solution of a has been considered, accordingly with the sketch in Figure 3.2, where a concave up paraboloid is presented. In summary, the first specifications on the maximum measurement range and the maximum acceptable error define univocally the geometrical parameters that determine the shape of the parabolic mirror, in terms of its curvature a in equation 3.16, and the position of the laser sources along z, assessed by the baseline b in equation 3.15.

3.2 Experimental analysis

3.2.1 Prototype description

As described in details, the proposed range sensor is based on the principle of laser triangulation. Following the early idea given in [26, 25, 27], the triangulation process is assisted by a parabolic mirror in order to achieve a wide FoV of 270°.

The aim of this investigation is a further improvement of the previous setup in terms of the reduction of the sensor size and the increase of the measurement rate. Specifically, the first prototype implements a parabolic mirror whose radius is equal to 60 mm. The corresponding telecentric lens, chosen to capture the whole mirror area, has the same radius of the reflector, and a length of about 600 mm. At the same time the distance between the vertex of the mirror and the laser plane, from now on baseline b, has been dimensioned equal to 1.5 m to acquire measurements with a maximum relative error of 0.1% at a distance of 3 m from the emitters. Finally the PAR, which determines the number of slices per seconds that maps the environment, is equal to 5.

The novel design fixes new initial specifications. As a first step the setup has to be reduced in size to a maximum total length of 1 m, keeping the

measurement resolution $\Delta\rho_{T,MAX}$ to 10 mm at a maximum distance d_{MAX} of 3 m. At the same time the PAR has to be improved reaching 25 profiles per second. These aspects imply the use of state-of-art devices, together with the redefinition of the design parameters, to fit the new requirements.

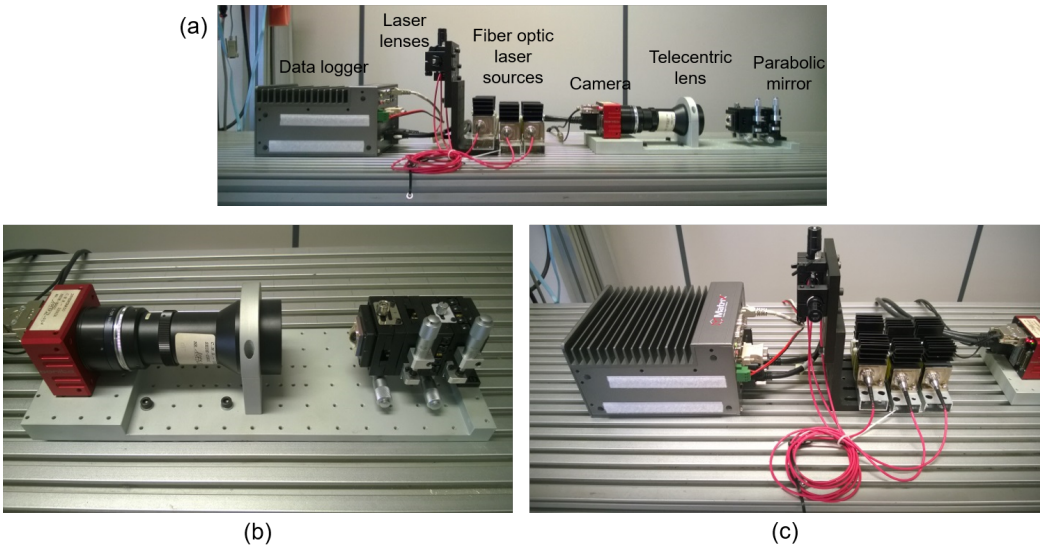

Figure 3.4: Picture of the actual prototype: (a) Overall setup; (b) Optical receiver made of the parabolic mirror and the lens-camera set; (c) Laser sources and lenses and data logger connected to the camera.

With reference to Figure 3.4, where a first prototype is presented, the sensor exploits fiber optic lasers, namely CUBE Laser by Coherent [168], with a built-in thermal management. Furthermore, the use of fiber tails assisted by cylindrical lenses enables the reduction of the space required for its mechanical assembling. At the same time, the initial specification of high measurement rate is ensured by the use of the camera CL-400 Bonito by Allied Vision Technology [169], which exploits the double and full Cameralink protocol with frame rates f up to 386 frames per second. The main features of this camera are reported in Table 3.2.

Table 3.2: Specification of the implemented camera (AVT CL-400 Bonito [169]).

Parameter Description	Value
Interface	$2 \times$ 10-tap CL Full+
Image resolution ($W \times H$)	2320×1728
Sensor size	4/3"
Pixel size (p)	$7\mu m$
Max frame rate at full resolution	386 fps

Once the camera has been selected, the unknowns a, b and M have to be properly dimensioned to match the initial specifications on the measurement error and sensor size. As stated previously, the error analysis leads to Equations 3.15 and 3.16 which can be easily exploited to derive the mirror curvature and the baseline, as a function of the magnification of the telecentric lens, implicitly held in p'. Typical values of the magnification M are 0.75, 1 and 2 (e.g. see Ref. [170]). These numbers have been tested, producing the results in Figure 3.5, where the maximum error $\Delta\rho_{T,MAX}$ is reported as a function of the mirror curvature and the laser-mirror distance. The presented plots are computed for realistic values of a and b. Specifically, the mirror curvature spans describing a maximum mirror depth of about 13 mm, corresponding to $a = 100m^{-1}$ with $M = 0.75$. It is important to notice that high-curvature mirrors are not suitable for the specific application, since their depths are much over the limit imposed by the depth of field of the lens, typically close to few millimeters. In this case, the telecentric lens would not be able to focus the mirror over its entire depth, or equivalently over its whole area. At the same time, the trial values of the baseline are limited to 1.3 m, anyway higher than the desired maximum length of the sensor. Before going through the inspection of Figure 3.5, it is worth observing that, within the considered boundaries of a and b, a magnification M equal to 2 does not produce visible reflections on the mirror, i.e. in the camera FoV, when the target is 3 m far from the laser sources. This value defines the maximum range of the proposed device, which makes it most suitable for indoor applications. At the same time, those outdoor applications where the main interest is focused on the closest targets (see railway monitoring) can be faced, taking advantage of the coherent nature of the laser line, which is highly recognizable against the ambient light. Nevertheless, also higher maximum ranges can be reached by changing properly the optical components

involved in the presented setup.

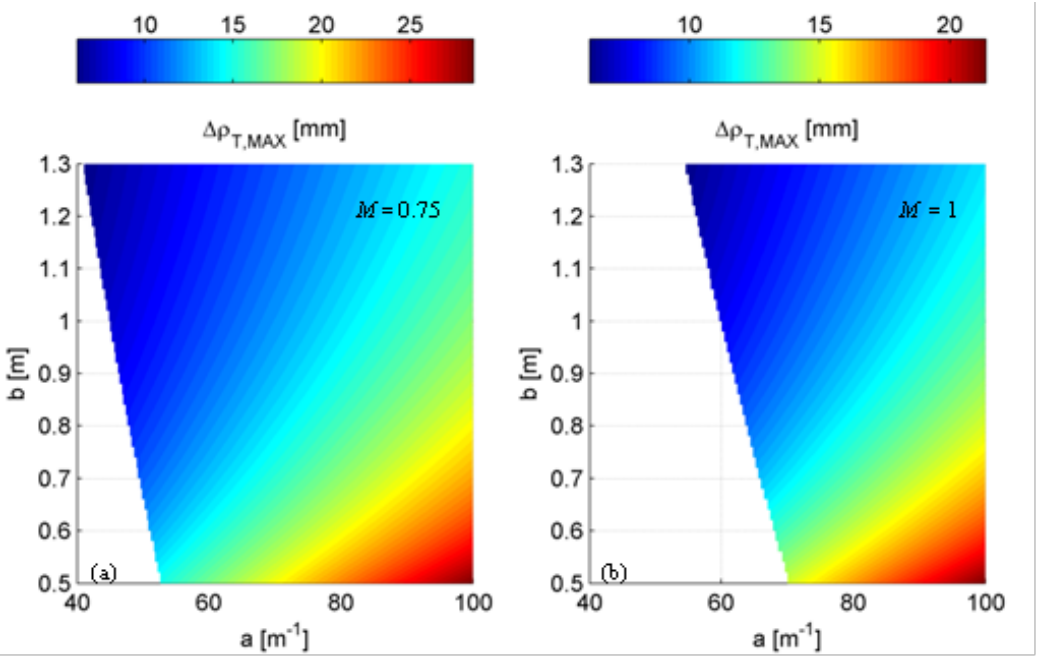

Figure 3.5: Maximum errors $\Delta\rho_{T,MAX}$ at d_{MAX} = 3 m as a function of the mirror curvature and the baseline, computed for magnification equal to: (a) 0.75, (b) 1. Regions where the laser incidence is out of the camera FoV are displayed in white.

The insight of Figure 3.5 demonstrates that when the magnification is equal to 0.75, the lower values of a and b that allows $\Delta\rho_{T,MAX} = 10$ mm are $48.22m^{-1}$ and $756.9mm$, respectively. On the other, the same specification is matched for $a = 64.29m^{-1}$ and $b = 758.2mm$, when $M = 1$. Although baselines are almost equal, the mirror curvatures change considerably. As stated previously, a conscious design would prefer lower curvatures, since the corresponding mirrors have shorter depths. In this way, the telecentric lens can extend its working distance over increasing areas of mirror, keeping the laser line focused. Hence, the telecentric lens VS-LTC075-70-35/FS by VS-Technology [170], having magnification equal to 0.75, has been chosen for the presented prototype. The final design parameters that allow the specification compliance are thus summarized in Table 3.3.

Table 3.3: List of design parameters that allow maximum error of 10 mm at a distance of 3 m.

Parameter	Value
a	$48.22m^{-1}$
b	$756.9mm$
M	0.75

Once the design parameters have been selected, the maximum error in the computation of the angular component of P_T can be estimated. With reference to equation 3.12, this error contribution depends on the exact angular component of the point P_{C0}. Figure 3.6 (a) shows the dynamics of the error term $\Delta\Theta_{T,MAX}$ as a function of the angle Θ_{T0}, equal to ϕ_{C0}.

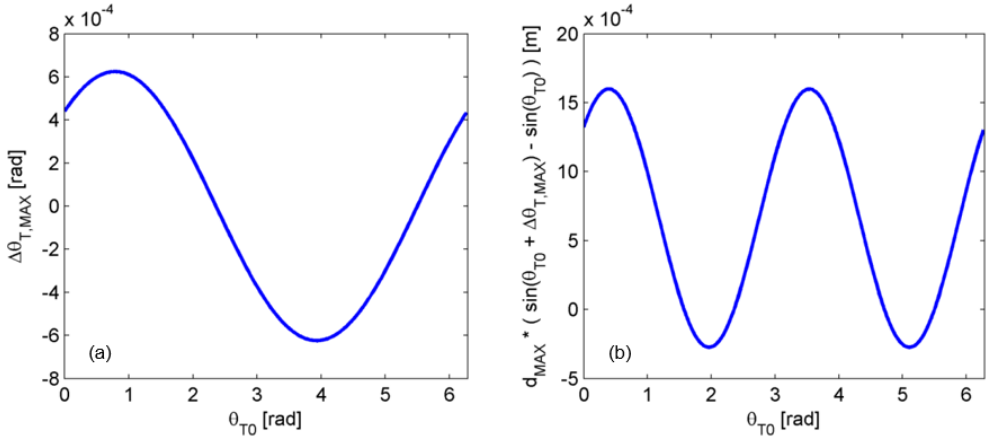

Figure 3.6: (a) Angular component of the maximum measurement error at $d_{MAX} = 3m$ as a function of the estimated angle Θ_{T0}. (b) Maximum estimated shift, due to the presence of angular uncertainty $\Delta\Theta_{T,MAX}$, in the computation of the y-coordinate of the point P_C at d_{MAX} equal to 3 m.

The analysis of Figure 3.6 demonstrates that the angular component of the maximum error due to image quantization is always below 3.5×10^{-2} degrees. As a consequence, the estimation of the target position in the (x, y, z) system of coordinates is altered as results of the application of sine and cosine functions to the term $\Theta_{T0} + \Delta\Theta_{T,MAX}$. Quantitatively the maximum error due to $\Delta\Theta_{T,MAX}$ in determining the x and y coordinates of the point P_C is

at most equal to $1.6mm$ at a distance of $3m$ (see Figure 3.6 (b)), i.e. about one order of magnitude lower than the specified $\Delta\rho_{T,MAX}$.

3.2.2 Setup calibration

However precise and mechanically stable the experimental setup can be, the actual geometrical parameters differ from the nominal one. As a consequence, the setup calibration has to compensate for it, estimating the unknown parameters F (or equivalently a) and b that govern the triangulation process. This task is mandatory within a calibration phase, which is driven by the inspection of a completely-known target.

Before going through the estimation, it is important to mention the preliminary assumption of the model, regarding the relative position of the mirror and the image plane. Specifically, the camera plane has to intercept the axis of symmetry of the mirror in its center. Since the camera has greater sizes than the mirror, it is more convenient to change the position of the latter, keeping the camera fixed at a distance from the mirror close to WD. Consequently, the mirror has been bracketed on mechanical handles, taking advantages of micrometric shifts in the xy-plane and rotations around the x- and y-axis.

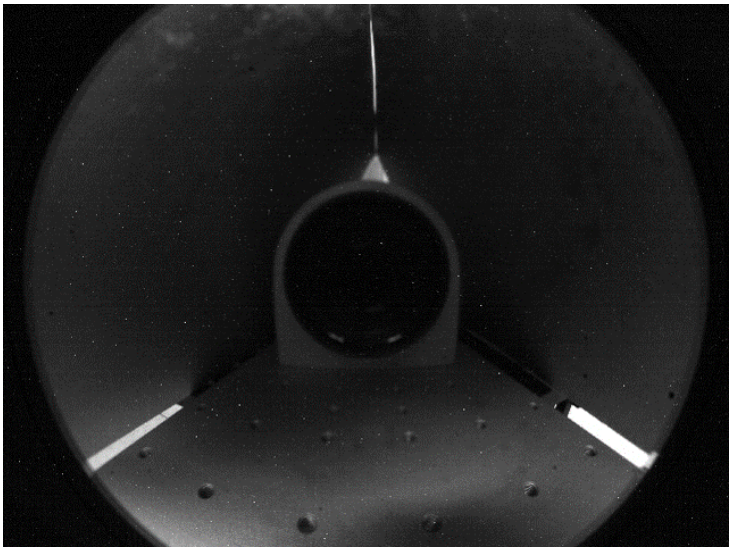

Figure 3.7: Example of frame captured by the camera for the estimation of the mirror position in the image plane.

Once the mirror has been equipped with micrometric rototranslational stages, a processing pipeline is needed to estimate its position within the image plane. The algorithm of mirror identification has been developed in the MVTech Halcon 11 [171] environment. In this case, the position of the mirror vertex can be estimated by searching for the mirror circular boundary in a set of sample frames captured by the camera. Figure 3.7 shows an example of image returned by the camera, where a self-reflection of the telecentric lens can be observed in the image center, whereas the mirror boundary can be easily recognized on the outer regions.

With reference to Figure 3.8, where the contour extraction is presented step by step, the implemented algorithm processes the returned frames (e.g. Figure 3.7) to estimate the mirror position through the following steps:

1. A process of image threshold highlights the pixels of intensity higher than 20, returning the green area in Figure 3.2 (a);

2. Given the areas of high intensity, the method extracts the region contours in Figure 3.2 (b);

3. The longest boundary is selected and fitted on a two-dimensional ellipse in the least squares sense, producing the green curve in Figure 3.2 (c);

4. The center coordinates are consequently derived (red cross in Figure 3.2 (d)), whereas the eccentricity of the estimated ellipse is evaluated to measure the alteration of the normal vector of the image plane with respect to the z-axis.

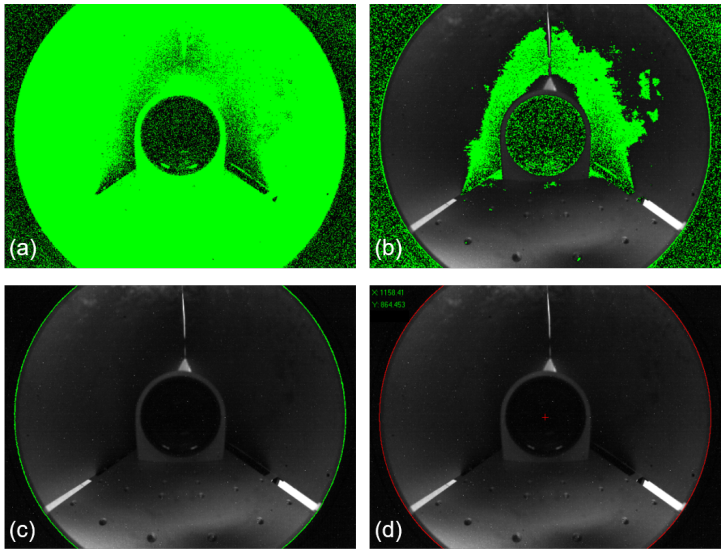

Figure 3.8: Image processing steps for the determination of the mirror position in the camera plane. (a) Threshold image; (b) Boundaries extracted from the threshold regions of high intensity; (c) Start image and corresponding fitting ellipse (in green); (d) Final results with the estimation of the center coordinates (red cross).

The presented algorithm controls the mirror position in real time, thus enabling the direct use of the micrometric stages for its exact placement. In this way, the initial hypothesis that leads to the model in equation 3.7 is verified.

The calibration phase can thus proceed with the estimation of the unknown model parameters. For this purpose, a wood structure made of 45-mm-thick strips has been realized and scanned by the proposed sensor in order to frame couples of laser points belonging to the strip corners. The Euclidean distance computed in the image plane between corresponding corner points is then compared to the actual corner distance, implicitly equal to the thickness of the laths. Figure 3.9 reports an example of an acquired frame used for the setup calibration, whereas the inset shows the corresponding couple of points named as the structure edges.

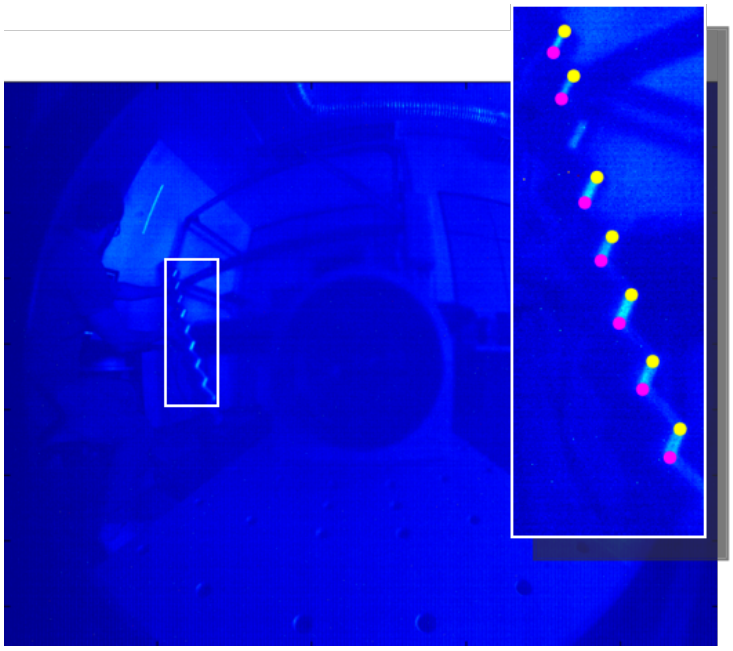

Figure 3.9: Example of a frame captured by the camera during the setup calibration. The laser line illuminates a target of completely known geometry. The inset highlights the extracted points belonging to the structure edges, whose distance corresponds to the thickness of the wood strips.

The experimental calibration is treated as an optimization problem. As a first step the couples of edges are extracted, passing through the following steps:

1. The image is cropped in order to eliminate secondary reflections due to the presence of external light sources, returning the region enclosing the sample target (see the inset of Figure 3.9);

2. The ROI is treated by a threshold process to highlight the laser points. This step generates a binary image where white pixels are candidate laser points;

3. A region growing approach is applied, after a morphological dilation filter, to detect continuous region that resamples the laser line;

4. The resulting regions are individually fitted on an ellipse. The limits of the major axis determine the edges of the laser line impinging on the

sample target. These points are derived with subpixel resolution.

Once the edges are extracted, these are transformed in world coordinates, using trial parameters. An objective cost function is thus defined as the square error between the computed edge distances and their nominal counterparts. The problem is thus solved in the non-linear least squares sense.

An overdetermined system is built exploiting more than 100 frames and solved in the model parameters, thus obtaining the resulting values in Table 3.4, with a corresponding residual of the cost function of 3.205×10^{-4}.

Table 3.4: Results of the calibration process.

Parameter	Calibration results
F	$5.166mm$
b	$702.12mm$

The actual values of the model parameters determine a drift of the measurement obtained under ideal conditions. From a quantitative point of view, compensating for the presence of setup alterations allows the deletion of additional systematic errors. With more details, considering the nominal parameters instead of the calibrated ones generates a peak error of $144.19mm$ at the maximum range of $3m$, i.e. one order of magnitude higher than the required range resolution ($10mm$).

3.2.3 Experiments and discussions

The experimental validation of the sensor setup can be performed in two different ways:

1. Inspecting the movement of a target, which is mechanically controlled via encoded slits and rotational stages. This technique requires the perfect understanding of the mathematical relationships between the world reference system, where the target shift is defined, and the mirror reference system, where actual results are determined;

2. Scanning the shape of a known object, placed at increasing distances from the laser sources. This method returns relative measurements, which are characteristics of the target itself. It follows that the knowledge

of the object pose in the mirror system of coordinates is no longer required. The comparison is self-consistent, given the shape of the target.

For these reasons, several acquisitions have been performed with the aim of determining the size of a square board, placed at increasing distances. Moreover, experiments have been run changing the direction of the radial shifts in order to cover many spatial regions. This results will be of interest since the goal of the proposed system is the inspection of surroundings, wrapped around the range sensor. In this case, it is mandatory to ensure that the measurements are always reliable, regardless the target position. Figure 3.10 reports an example of frame, acquired when the laser line impinges on a square paperboard having side equal to $310mm$. The base of the board has been perfectly aligned to the ground, in order to ensure that the line crosses it parallel to its vertical sides.

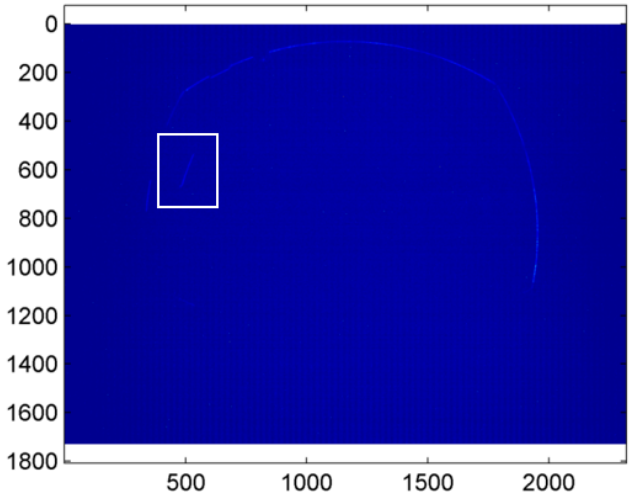

Figure 3.10: Example of frame acquired by the camera for testing the sensor accuracy. The rectangle encloses the laser line impinging on the board.

The edges of the laser line have been extracted by means of the same algorithm used in the calibration phase for the corner extraction from the known target. In summary, a ROI including the laser line is extracted and a

binary image is built by means of a threshold process; after the application of a dilation filter, a region growing approach is used to determine the actual laser line, which is fitted on an ellipse. The edges of the laser line on the board sample are equal to the limits of the major axis of the fitting ellipse.

The results of the proposed algorithm, applied to the frame of Figure 3.10, are shown in Figure 3.11.

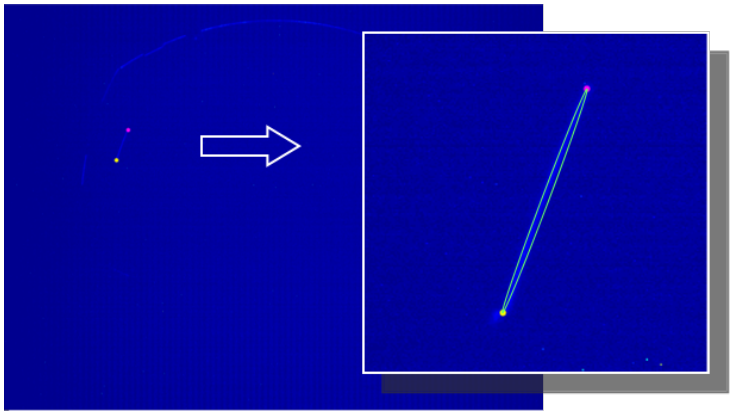

Figure 3.11: Results of the edge extraction algorithm used for the detection of the board sizes. The inset shows a magnified view of the extracted points; the green line identifies the fitting ellipse.

Once the edges of the laser line are extracted from the image, they can be reported in the (x, y, z) reference system, thus obtaining their positions in space. It is evident that the spatial distance between the edges is implicitly equal to the side of the panel. Figure 3.12 points out the estimated dimension of the board as a function of the target distance. Plots are obtained spanning the target movement around the sensor for discrete angles α, which defines the direction of the target shifts with reference to the ground (assumed parallel to xz-plane).

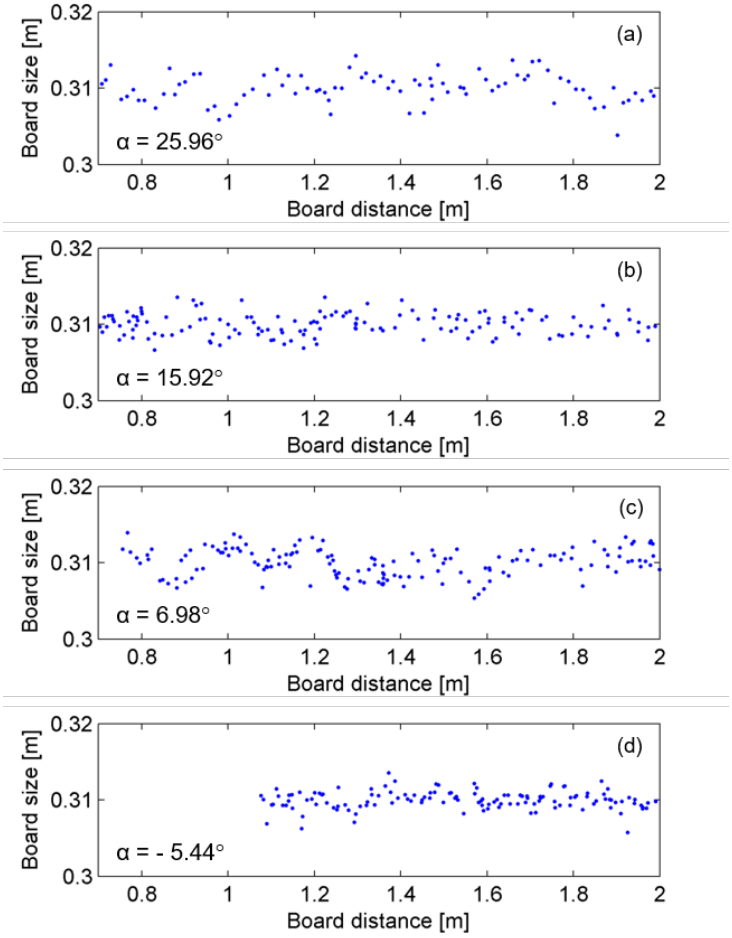

Figure 3.12: Estimation of the size of the sample board as a function of the distance of the target from the light sources. Measurements have been performed changing the direction of the radial shifts, accordingly with the axis defined by the angle α, referred to the ground plane: (a) $\alpha = 25.96°$; (b) $\alpha = 15.92°$; (c) $\alpha = 6.98°$; (d) $\alpha = -5.44°$.

Results clearly show the good agreement of results in computing the dimension of the board side, regardless the target position, which qualitatively does not alter the measurement error. In particular the average values of the estimated dimensions of the board are reported in Table 3.5.

Table 3.5: Average values of the measured size of the square paperboard under analysis. The target has a nominal dimension of $310mm$.

α	Paperboard size $[mm]$
$25.96°$	309.77
$15.92°$	310.62
$6.98°$	310.23
$-5.44°$	309.39

Moreover, range samples have been collected in equally spaced bins in order to derive information about the noise statistics, leading to Figure 3.13. At this stage, quantization errors are compensated by the process of point extraction, which computes the position of the panel borders with subpixel precision. Here, the main contributions to the measurement errors are related to a superposition of two mechanisms of degradation. First the laser line is defocused on the camera plane as effect of the finite depth of field of the camera and the divergence of the laser light. Then, the image processing introduces implicit approximations, since curve lines corresponding to straight segments are actually fitted by ellipses. Nevertheless, the data collection in Figure 3.13 follows a Gaussian-shaped function centered on the expected measurement, thus proving the good accuracy of the proposed sensor. Measurements are altered by noise contribution with standard deviation of $1.74mm$ and a consequent $\sim 99\%$ confidence interval of about $10.44mm$.

Furthermore, the presented error estimation is uncorrelated with respect to the camera frame rate, till the limit fixed by the inverse of the exposure time used in the presented experiments ($30ms$). When the frame rate is higher than 33 fps, the exposure time has to be reduced properly, thus downing the intensity amplitude of the detected laser line. As a consequence, the decreasing signal-to-noise ratio can produce effects on the measurement quality. Nevertheless, the initial requirement of fast acquisitions (25 fps) can be matched within the limit of precision discussed before.

The presented results can be compared with those returned by the AccuRange AR4000 rangefinder, whose range measurements are affected by a statistical white noise with standard deviation of $2.5mm$ when the target is placed $1m$ far from the emitter [172]. Although noise contributions seem comparable, the frame rate of the AR4000 rangefinder imposed for these experiments is equal to 1 kHz. On the other hand, the presented sensor

produces about 5×10^4 samples per second at the current frame rate of 25 fps. This behavior is due to the camera resolution which allows the proper decomposition of the detected laser line of a single frame in more than 2000 samples, without any degradation of the measurements.

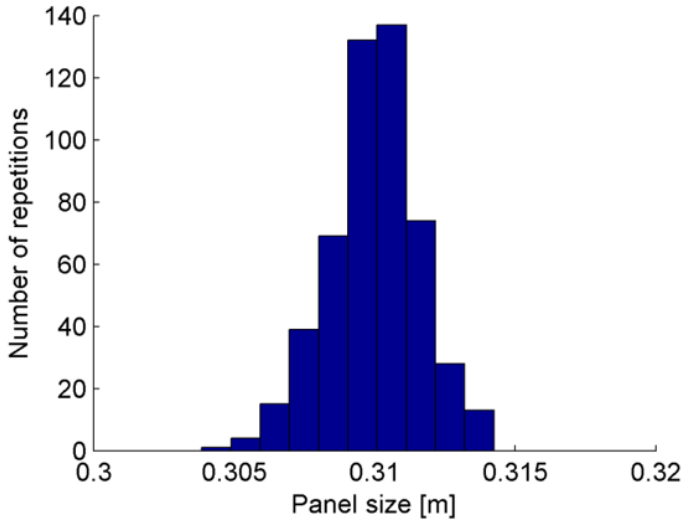

Figure 3.13: Collection of samples returned by the analysis of the board side.

3.2.4 3D reconstruction

As a proof of the actual capabilities of the presented range sensor in 3D reconstruction, an example of acquisition is briefly reported in this Section.

The sensor is fastened on a mobile robot, which flows through an indoor environment (in this example a corridor) following straight trajectories at a constant speed of $400mm/s$. The camera is triggered by a TTL signal generated by the robot encoders. Given the resolution of the encoders and the robot speed, the camera sends a frame to the data receiver every $5mm$, exploiting the full camera link protocol. This data is a raw matrix with 1728×2320, full of unsigned char representing the image intensities. Frames are then processed to extract the position of the laser line in the image plane. At this stage, the image is sectioned following 2048 radial directions, starting from the image center. Each section can include at most one laser peak, whose position can be easily computed applying the standard center of mass

approach [173]. Knowing the exact position of the laser line with subpixel accuracy and the robot pose returned by odometry, it is possible to derive the corresponding coordinates in three-dimensions. These samples are finally ordered in a Wavefront .obj file, filled by the vertex of the dataset. The reconstruction has produced a point cloud having size equal to 2.4×10^6 points. This outcome is shown in Figure 3.14.

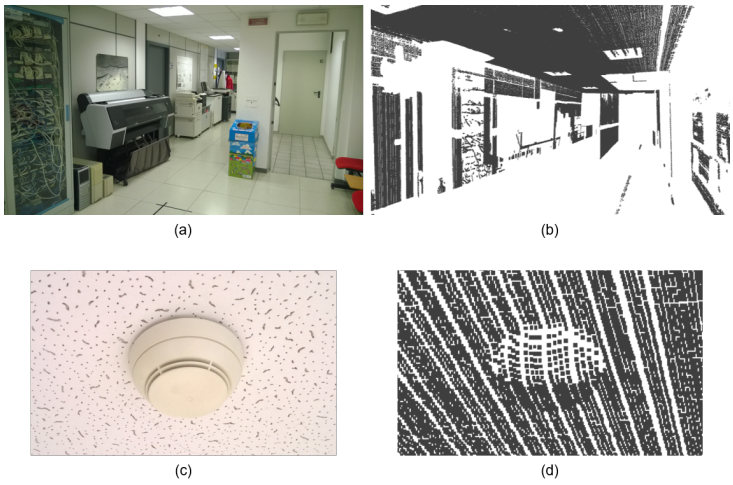

Figure 3.14: (a) Acquired corridor and (b) corresponding 3D reconstruction; (c) picture of a particular object with maximum size of $10cm$ and (d) corresponding 3D model.

3.3 Summary

In this chapter, an omnidirectional range sensor for the inspection of surrounding spaces has been developed. Following the principles of laser profilometry, the range sensor estimates the distance of targets by looking at the displacements of a laser line projected onto the environment. When the vision system is assisted by a parabolic mirror, high FoV can be reached in a single scan, i.e. a camera frame, thus increasing the number of profiles, up to the limit fixed by the camera electronics. The experimental setup has been designed following analytic expressions to meet initial specification on its overall size and the measurement resolution at a distance of $3m$ from the emitters. A novel calibration phase devoted to the alignment of the optical component involved in the acquisition has been described, together with the estimation

of the actual geometrical parameters that lead to the range measurements. Several experiments had been run in order to establish whether the proposed system can inspect accurately the surface of known calibrated targets, using effective image processing techniques. Measurements returned by the sensor for the estimation of the size of the known target had been compared with nominal values. Experimental results have demonstrated that the noise contribution follows a Gaussian shape with standard deviation of $1.74mm$ and negligible systematic error (mean value close to $0.31mm$), regardless of the target distance from the sensor. All noise sources are ascribable to the defocusing effect induced by the finite depths of field of both emitting lasers and receiving system. Keeping the same exposure of the camera, the profile acquisition rate can reach 33 profiles per second, as required by the specifications, without increasing the maximum error.

Chapter 4

3D Point cloud registration

In this Chapter a modified ICP algorithm for the registration of datasets acquired at different epochs for structural monitoring is described. The proposed approach belongs to the data-driven category, i.e. it uses information within point clouds, without artificial markers or GPS/inertial information. As a matter of fact, computer aided methods that do not use markers can speed up registrations, since the time spent for structuring the environment is no longer required. Furthermore, in this way alignments of point clouds are always enabled, also when the environments can not be structured or the GPS information is not available (e.g. indoor scenes).

The underlying idea comes from the observation of some limitations common to many ICP approaches. First of all, most of them neglect the properties of the acquisition and the environment under investigation. In fact, when laser rangefinders are used, the mechanisms of ray projection can induce the presence of different shades when objects are observed from altered points of view. As an example, pillars and columns, typical of civil infrastructures (buildings, road and underground infrastructures, such as covered parking, metro tunnels, etc.) can introduce implicit artifacts in the measurements and hence errors in the registration process. In Figure 4.1, two 3D models of the same environment acquired from different points of view are shown: the red points represent the differences, due to the change of the view-point, which can not be matched in the registration process. Furthermore, when point clouds are acquired at different epochs for structural monitoring, the inspected scene can experience changes (object shifts, plane rotations, etc.) with respect to the reference point cloud. If both implicit artifacts and actual changes are neglected, and all the points are considered in the registration

process, wrong registration parameters can be obtained.

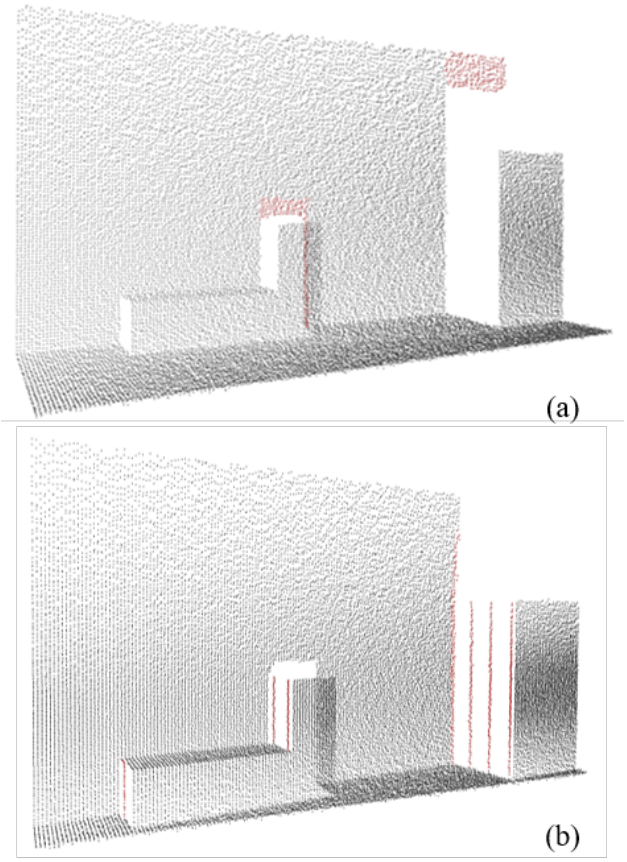

Figure 4.1: Comparison of two 3D models of the same environment. Red dots are implicit differences due to the change of the sensor point of view. (a) Reference and (b) source point clouds.

These critical aspects are the main topic of the methodology described in this chapter, which modifies the standard ICP implementation by introducing deletion masks, i.e. binary weighting matrices made of 0s and 1s. This strategy is able to remove the measurement artifacts due to the changes of the sensor point of view, reaching higher robustness against the possible environmental changes between the two different acquisitions. Deletion masks are defined at each iteration as a function of the estimated sensor positions and are applied before the evaluation of the distance between the source

and reference point clouds. The aim of this mask is the deletion of pairwise comparisons altered as effect of estimated changes of the sensor point of view. Experimental evidence demonstrates that the proposed method can improve the accuracy of the standard ICP method and its variants, also in presence of alterations of the environments under inspection.

4.1 Methodology

Whenever the processing of 3D models is aimed to monitoring infrastructures, high accuracy and high resolution are necessary. Laser rangefinders are the best sensors to achieve this goal since they can reach and measure hardly-positioned structures in narrow spaces, without any difficulty and regardless the lighting conditions. Typically, laser rangefinders are bracketed on mobile vehicles, which proceed through the environment, and collect distance measurements ρ as a function of the vehicle position. As a result, the position in space of the samples gives a representation of the acquired targets, namely of their external surfaces. Two examples of point clouds acquired in indoor environments are reported in Figure 4.2. In particular, Figure 4.2a represents a generic entrance hall, whereas Figure 4.2b models a covered parking. These environments will be the specific case studies for the presented algorithm of point cloud registration. The arrows in Figure 4.2 display the directions followed by the mobile vehicle during the acquisitions. At this stage, it is important to notice that the method, and its underlying ideas, can be applied to any dataset produced by a generic laser scanner. Nevertheless, for the sake of simplicity, the following treatments will refer to the specific case of a moving sensor which collects samples as the vehicle moves through the environment.

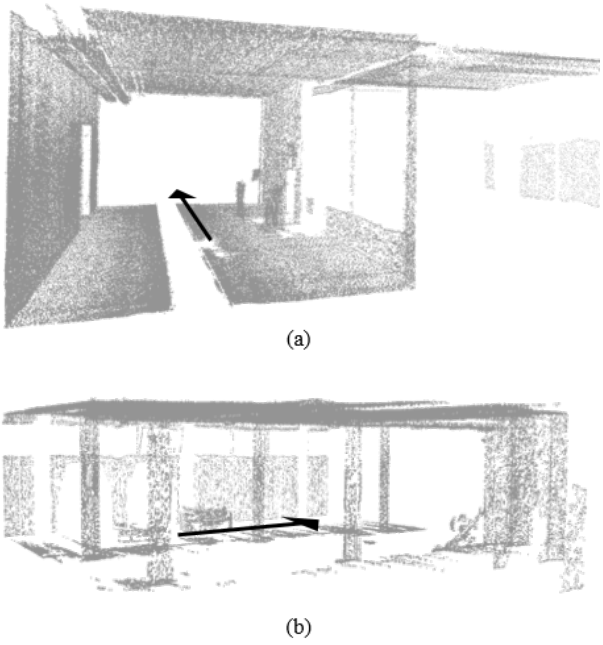

Figure 4.2: Example of point clouds derived from (a) a generic indoor environment and (b) a covered parking. The arrows represent the directions followed by the mobile vehicle which carries the sensor through the environment under analysis.

The following subsections will describe the best processing for the alignment of two or more datasets modelling the same environment, i.e. an indoor infrastructure. The presented method can find application for any kind of measurement scheme aiming the environmental modeling. Attention will be focused to the reduction of the size of the point clouds, together with the description of the main limits of the existing algorithms. Then, the method will be explained in details, pointing out the most important features that will carry to the improvement of the results.

4.1.1 Preprocessing steps

The first step in the processing of point clouds aims to the datasets simplification, which is often mandatory to derive lighter datasets, full of information, that can be easily treated by the algorithms. Well-known techniques and methods are often used to extrapolate the meaningful parts of the datasets

and to simplify them without any loss of information ([174, 175, 176]). This phase can be summarized in the following steps: outlier removal, reduction of useless samples and surface analysis.

Typically, secondary reflections or high-absorbing targets can lead to noisy measurements. As a consequence, many points acquired by the range sensor are outliers which can be removed exploiting the dataset statistics [177]. Since the point clouds are dense of samples, a point is an outlier when it belongs to a low-density region. In practice, a sphere is centered on the investigated point in order to compare the number of samples within this region with the expected one. In a more efficient way, the acquired samples are clustered following a distance criterion and then processed in order to find isolated points, i.e. those points, or sets of points, which have a small number of neighbors, lower than a threshold S_{th}. This processing is general and can be applied regardless the kind of scene under analysis. Its effectivity only depends on the properties of the point cloud produced by the sensor: size, resolution and accuracy, which implicitly define the threshold parameters. For instance, laser rangefinders able to produce tens of samples of a surface of $1cm^2$ at $1m$ of distance, can return dense point clouds. In this case, setting the radius of the sphere to $1cm$ and the threshold $S_{th} = 5$ can ensure the removal of the only outliers due to measurement errors.

Since the method is defined for processing indoor datasets, it is possible to design smart filters, able to exploit this domain knowledge for the removal of those samples that do not add significant information to the model. This result can be achieved by extending the principles of the Split and Merge algorithm (also known as Ramer-Douglas-Peucker algorithm, RDP [178]) to the input dataset. In more details, the range values belonging to an ordered vector of indices generate a curve which is decomposed in a set of line segments, whose edges define a subset of the exact samples. The simplified curve is derived by deleting the points that have a distance from the corresponding line segment lower than a tolerance value, named as RDP_{tol}. Some results, obtained by changing the tolerance value, are shown in Figure 4.3. The method operates searching for the most informative points and deleting the ones which are unnecessary. In this way, range sets extracted from indoor transport infrastructures, which are of interest in this framework, are approximated by line segments with low residuals. This representation is the most suitable for the processing of the specific environments, since scenes are usually made of planes. Finally, it is important to observe that the tolerance RDP_{tol} can be chosen proportional to the range measurement,

since many sensors produce results with different resolutions, depending on the distance of the target.

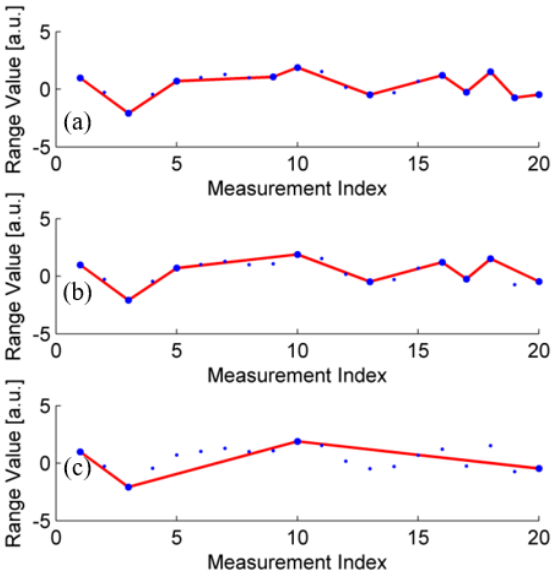

Figure 4.3: RDP results on a pseudo-random array of range values. The tolerance value is constant and equal to: (a) $RDP_{tol} = 0.5$, (b) $RDP_{tol} = 1$ and (c) $RDP_{tol} = 2$.

As a final step in the model creation, information about the point position in space are merged with surface data [179]. The task of surface reconstruction from 3D range data has been deeply developed and many algorithms have been already proposed. Among them, the most important are the Ball Pivoting Algorithm [180], the Powercrust [181], the Poisson Surface Reconstruction [182, 183] and the Multi-level Partition of Unity Implicits (MPU) [184]. When the point cloud is ordered, this goal can be achieved easily. Whenever each range value ρ that belongs to the i-th point cloud P_i (i identifies the acquisition) is obtained at specific discrete indices, a surface mesh S_i, made of triangular patches, is directly created by linking consecutive indices (points become vertices of the triangles of the mesh). It is clear that the preliminary simplification produces holes in the map of range values given by the sensor. In this context, Figure 4.4 reports an example of the generation of holes in the triangular mesh. When the green dot in Figure 4.4a is deleted by the

previous simplification, the reconstruction of the triangular patches fails since the theoretical correspondence of adjacent indices is no longer valid. These issues are overcome by connecting vertices whose indices satisfy a criterion of minimum distance, instead of connecting vertices which are close in space, thus reaching the final result in Figure 4.4c. In this way, the construction of wrong patches made of edges that actually belongs to different surfaces is avoided.

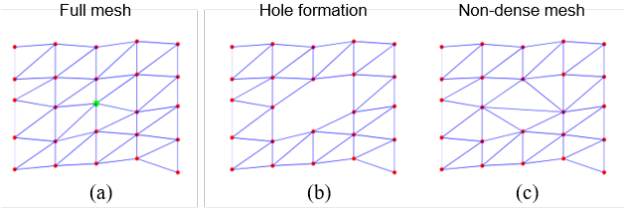

Figure 4.4: Example of the process of hole generation due to the simplification of the point cloud. (a) Starting mesh; (b) Hole formation due to the simplification of the green dot in (a); (c) Final result of the surface reconstruction.

Once the set of ordered connections defined by the surface mesh is defined, it is used to create point normal vectors, which are defined as the average value among all the normal vectors of the triangular patches that include the specific point. Each sample is further compared with the closest ones in terms of normals and it is deleted from the dataset if all surroundings have the same properties.

4.1.2 ICP and its drawbacks

The task of registration of clouds of points is mostly per-formed applying the Iterative Closest Point (ICP) algorithm [105, 106]. A simplified scheme of the standard ICP algorithm is summarized in the flow chart in Figure 4.5.

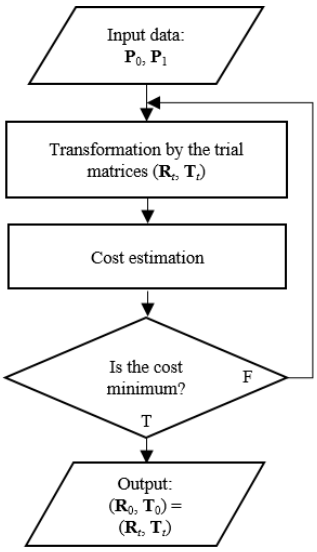

Figure 4.5: Flow chart of the standard ICP implementation.

Starting from its first formulation, the method considers two point clouds, a reference P_0 and a source P_1, each one constituted by a set (vector) of distance measurements ρ_0 and ρ_1, respectively. The ICP tries to establish the transformation parameters which carry to the best matching of the overlapping regions, solving an optimization problem in the least squares sense. In summary, the point clouds are first subsampled uniformly or trying to extrapolate the most significant points (discontinuities). Then, the ICP algorithm establishes Γ point correspondences between the two datasets and transforms the source point cloud, following the rotation R and translation T guess matrices. Then it directly computes the cost in terms of sum of squared differences between the Γ range values of the matched samples. The cost function is defined as follows:

$$C(R,T) = \sum_{j=1}^{\Gamma} \left(\rho_{0,j} - \rho_{1,j}(R,T)\right)^2 \tag{4.1}$$

where $\rho_{1,j}(R,T)$ are the range values extracted from the source P_1, after the transformation defined by the guess matrices (R,T). The cost is thus optimized as a function of the trial matrices, which are full of entries. As a consequence, the alteration of the point of view can be compensated exploiting six degrees of freedom.

Further ICP variants exploit a different estimation of the cost function. As an example, the use of a point-to-plane metrics is used in [117] in order to weight the correspondences between homologous points by means of the surface properties. In this case each addend of the cost function is multiplied by a weighting term w_j, equal to the normal vector of the specific j-th point of the reference. Here the problem of estimating point normals deserves attention since its accuracy and reliability are mandatory to achieve good results [185].

Although the ICP formulation is very simple and often allows a closed form solution, many drawbacks can emerge in actual contexts [186]. First of all, it is straightforward to understand that this cost term is also linked to the possible modifications of the environment. If the environment is heavily altered or the points of view significantly change, the perfect alignment of the datasets produces higher values of the cost function. As an effect, the ICP algorithm can fail since it reaches a minimum of C for incorrect entries of the matrices (R, T).

Moreover, the different points of view of the sensor among the acquisitions can further weight this aspect, since they generate measurement artifacts near the object edges, even if the environment is not altered. This issue is of great importance, since the ICP algorithm filters out these wrong correspondences before the cost estimation by means of median filters. However, when the datasets are obtained from altered environments, the distances among points are higher in values. As a consequence, the median value raises till the limit of being comparable with the distance between points in wrong correspondence, which can not be deleted by the median analysis. In this case, the effective contribution of this approach vanishes.

Moreover, if the dataset is firstly subsampled non-uniformly to preserve information, i.e. discontinuities [187], the comparison can be additionally affected by errors, since edge regions are the ones carrying the main contributions of implicit ambiguities. Rejecting edges from the comparison, without any smart control, removes almost all the information, inducing registration uncertainty. For this reason the strategy must be improved by taking into account the three-dimensions in order to understand how view-points differ in space, and remove the spatial regions that lead to errors.

4.1.3 Point cloud registration with deletion mask

As described before, the proposed method intends to overcome the drawbacks of the standard ICP technique and its variants. The method modifies the standard implementation of the ICP algorithm following the processing steps depicted in the flow diagram in Figure 4.6. Specifically, deletion masks, or DMs (see the dashed box in Figure 4.6), are introduced to remove the erroneous point correspondences which are extracted from ambiguous regions, where implicit differences can raise as a consequence of the change of the sensor view-point.

Referring to the nomenclature of the previous section, the two datasets P_0 (reference) and P_1 (source) differ by the goal rotation R and translation T matrices. Moreover, the starting dataset is completed by the corresponding surface meshes S_0 and S_1.

Each box of the flow chart describes a specific operation on the input data, made of the full model (vertices and faces). In summary, the reference and the source point clouds are compared by means of a cost estimation, after that the reference point cloud is analyzed to derive the deletion mask. Since the problem is solved iteratively, trying to find the values of the objective matrices R and T that approximate the alteration of the sensor point of view, it is possible to exploit R and T to find the implicit differences due to the change of the sensor trajectory.

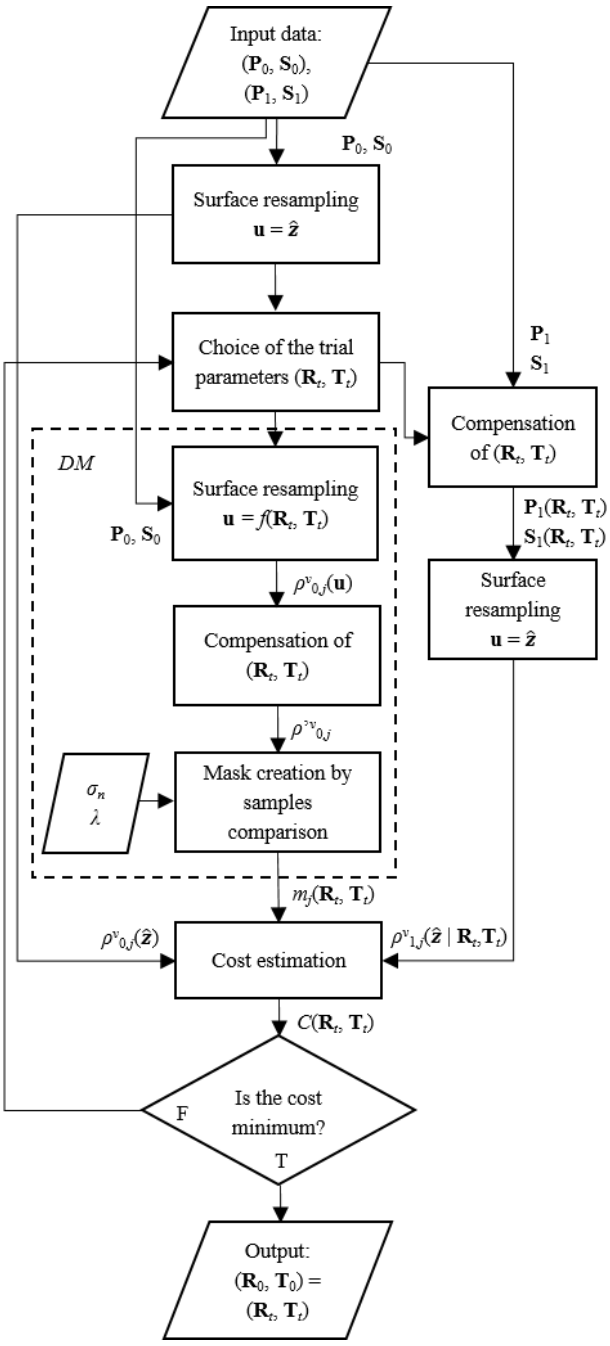

Figure 4.6: Flow chart of the presented method for point cloud registration.

S_0 is scanned in synthetics, exploiting the concepts of virtual measurement to replicate the expected point of view of P_1, iteratively defined by R and T. This virtual point cloud is then compensated by the same parameters R and T and compared to the actual reference. This comparison generates the deletion mask for the specific parameters of R and T. The mask is thus applied in product in the cost estimation, removing wrong correspondences due to implicit and unavoidable alterations of the point clouds under registration.

It is important to notice that the use of deletion masks in the selection of suitable point correspondences prevents the task from being solved in a closed form. As for the vast majority of the ICP variants, the hypotheses that lead to a close analytical solution are no longer valid, and thus its solution has to be found by means of a trial-and-error approach.

The following subsections highlight the bases of the proposed algorithm, focusing on the two main concepts of virtual measurement and deletion mask.

4.1.4 Virtual measurements

Before going through the description of the methodology, it is mandatory to focus on a preliminary task. Actually, the implementation of the deletion masks follows the definition of virtual measurements. The aim of this task is the extraction of a new arrangement of Q samples of the starting surface mesh S_i from a user-defined point of view.

Since the environment is scanned with the aim of a complete reconstruction, it is possible to suppose that the whole surroundings are modeled by a set of surfaces wrapped around a specific direction (e.g. the arrows in Figure 4.2). Under this hypothesis, the processing intends to create a novel set of points by looking at the whole surfaces from specific positions.

In summary, the virtual scan resamples the reconstructed surfaces starting from positions defined by the direction of a unit vector $u = [u_x, u_y, u_z]^T$, having origin in a specific initial point p_0. The direction τ of this vector is sampled in $(S + 1)$ points, labelled as p_s, from the origin of $u(p_0)$ till the end of the spatial domain (p_S). Consequently, $(S + 1)$ planes π_s, orthogonal to τ in the 3D positions of p_s, can be defined. The intersection between these planes and the surface mesh S_i returns a closed curve, which can be further sampled at discrete angular steps around the direction of u.

As a result, the process gives a new set of range measurements $\rho^v_{i,j}$, where $j = 1, \ldots, Q$. Here the apex v underlines the virtual nature of this measurement.

It is important to notice that this process is intended to replace the original measured points with equivalent ones coming from the intersection with the three-dimensional mesh. Then the registration of datasets will be performed over this new set of points. This process adds an advantage to the methodology in terms of a better response against noise. Specifically, if noise mainly follows a zero-mean Gaussian statistic, each patch takes into account the influence of three points experiencing different corruptions. Consequently, the virtual resampling of the triangular patches acts as a smoothing filter since the noise over the three ranges is "averaged" by the patches. Further numerical analyses have proven a reduction of the dispersion of the set of samples of about 30%.

4.1.5 Deletion masks

Virtual measurements constitute the basis for deletion masks, which are the focus of interest of the presented method. With reference to the diagram in Figure 4.6, the starting mesh S_0, made of a set of contiguous triangular surfaces, is first resampled at the beginning of the algorithm along a reference path which identifies the direction over which the source point clouds will be registered. Although any direction can be equivalently set as the reference, for the sake of simplicity, S_0 is resampled along the z-axis. The resulting set of range values is labelled as $\rho^v_{0,j}(\hat{z})$. This task is out of the iterative process and thus is computed once when the algorithm starts and aims to determine Q reference samples which will be used to create the deletion masks.

The iterative process begins with the choice of the trial compensation matrices (R_t, T_t), full of non-vanishing entries. The reference P_0 is scanned virtually from the view-point u, defined accordingly with the trial parameters (R_t, T_t). The resulting dataset of range values $\rho^v_{0,j}(u)$ is further rototranslated to compensate for the superimposed changes defined by (R_t, T_t), giving a new set of range samples $\rho'^v_{0,j}$. It is easy to understand that the pairwise comparison of $\rho^v_{0,j}(\hat{z})$ and $\rho'^v_{0,j}$ highlights the only ambiguous regions which can introduce an overestimation of the cost function. Equivalently, $\rho'^v_{0,j}$ is found numerically by looking at exactly the same scene of $\rho^v_{0,j}(\hat{z})$, but from a different point of view. This replicates on P_0 the same corrupted conditions that are iteratively estimated to affect the source P_1, responsible for the unavoidable implicit differences between the two acquisitions.

Given the information on the ambiguous regions, a deletion mask can be

created to prevent these points from entering in the computation of the cost function. Analytically, the entries of the deletion mask are:

$$m_j(R_t, T_t) = \begin{cases} 0, & \left|\rho^v_{0,j}(\hat{z}) - \rho^{'v}_{0,j}\right| > \lambda \cdot \sigma_n \\ 1, & \left|\rho^v_{0,j}(\hat{z}) - \rho^{'v}_{0,j}\right| \leq \lambda \cdot \sigma_n \end{cases} \tag{4.2}$$

being σ_n the noise standard deviation, whose amplitude will be discussed in the next sections, and λ a positive number identifying the mask strength. The latter term should be chosen properly in accordance with the noise statistics. As an example, if range measurements are mostly degraded by white noise, a value of this product greater than three times the variance ($\lambda = 3$) is enough to ensure that differences between couples $\rho^v_{0,j}(\hat{z})$ and $\rho^{'v}_{0,j}$ are only due to implicit alterations, out of the statistics with a confidence equal to 99.7%.

The iteration process is finally completed by the surface resampling of the source input mesh. This dataset is first rototranslated applying the trial parameters at each iteration. Then, it is resampled following the procedure of virtual measurements with $u = \hat{z}$ (misalignments have been already compensated). It is important to notice that the surface resampling of the source mesh still gives Q range values, named as $\rho^v_{1,j}(\hat{z}|R_t, T_t)$, which are implicitly in pairwise correspondence with those extracted from the reference dataset $\rho^v_{0,j}(\hat{z})$. As a consequence, the point matching is guaranteed without the application of any a priori condition.

Starting from its ICP formulation in Equation 4.1, the cost function can be finally redefined as:

$$C(R_t, T_t) = \sum_{j=1}^{Q} m_j(R_t, T_t) \cdot \left(\rho^v_{0,j}(\hat{z}) - \rho^v_{1,j}(\hat{z}|R_t, T_t)\right)^2 \tag{4.3}$$

The method can be thus iterated improving the solutions for the cost optimization, within a termination criterion. The final trial matrices R_0 and T_0 that give the minimization of the cost are those of the refined transformation that best approximates the actual values of R and T.

4.2 Experiments and discussion

4.2.1 Case study

The proposed technique has been developed for the registration of point clouds acquired in the context of indoor infrastructures, where GPS localization

is no longer available. The following subsections describe the experimental setup used for the acquisitions, the choice of the preprocessing parameters and the error metrics that will be used for the comparison of results with further ICP variants.

Experimental setup

In the presented experiments, 3D datasets are referred to a local reference system (x_i, y_i, z_i) of the i-th acquisition, where the $x_i z_i$-plane is assumed parallel to the ground. A mobile vehicle proceeds through the environment following straight trajectories along the z_i-axis, and carries a laser rangefinder which samples the surroundings by slices. The origin of the local reference system is placed on the position assumed by the sensor when it acquires the first slice of points. Each slice has N distance measurements expressed in terms of pairs (ρ_k, Θ_k), $k = 1, \ldots, N$, belonging to planes parallel to (x_i, y_i). Therefore, the resulting point cloud is implicitly ordered in discrete indices, since each range value ρ_k can be labeled by increasing angles Θ_k and slices. Without any loss of generality, the registration is applied to distance measurements performed using the time-of-flight laser scanner AccuRange AR4000-LIR [22] in Figure 4.7.

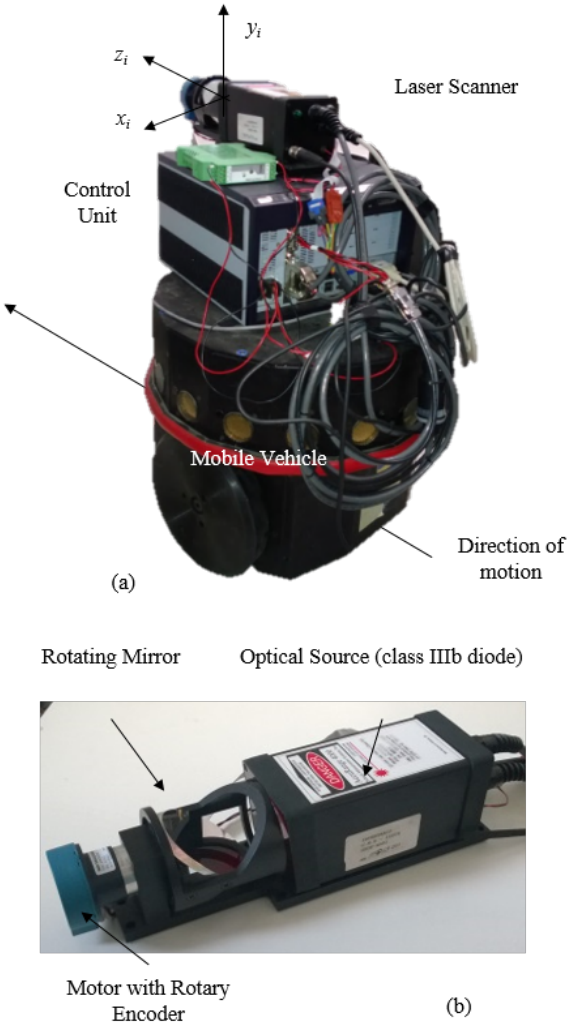

Figure 4.7: (a) Experimental setup used for actual inspections. (b) Picture of the laser rangefinder, underlining the optical source and the rotating mirror.

It is made of a laser source working at a wavelength of $780nm$. The generated beam is deflected of 90° by a rotating mirror ($2600rpm$) and then swept through 360°, to sample the environment by slices with a maximum range distance of $15m$. It is worth noticing that acquisitions are actually obtained following a helix, having axis along z_i. Nevertheless, range values are assumed to lie on sampling slices, which are formed anytime the mirror

performs a 360° revolution. The position of the slice origin on the ground is equal to the average value of the $x_i z_i$-coordinates returned by the vehicle odometry. The vehicle speed has been set to $0.2m/s$, whereas the sample rate of the rangefinder has been fixed to $1kHz$. Virtual measurements have been performed following the approach described in Section 4.1.4 in order to resample the point cloud. This way, a number of total slices between 200 and 250 has been collected, obtaining a spatial resolution along the direction of the vehicle movement of about $75mm$. Finally the range resolution is equal to $0.25mm$.

Preliminary analyses on the collected samples had demonstrated the existence of three noise sources [172]:

- statistical white noise with standard deviation equal to $2.5mm$ at a distance of $1m$;
- colored noise due to the temperature control with a slow time constant of about $2.1s$;
- an amount of failed acquisitions (5% of the total number of the acquired samples).

Knowing the statistics of the point cloud, it is possible to determine the parameters of the preprocessing steps described in Section 4.1.1. With more details, the threshold value S_{th} for the choice of the poorer cluster made of outlier candidates is equal to 5. The Ramer-Douglas-Peucker algorithm has been applied to the range values belonging to each slice of the dataset with a tolerance value $RDP_{tol} = 1mm$ at $1m$ of distance from the sensor source.

It is worth noticing that these parameters are chosen in order to prevent the lack of information due to the dataset simplification. This ensures that the application of the preprocessing steps does not impact in the results of the point cloud alignment.

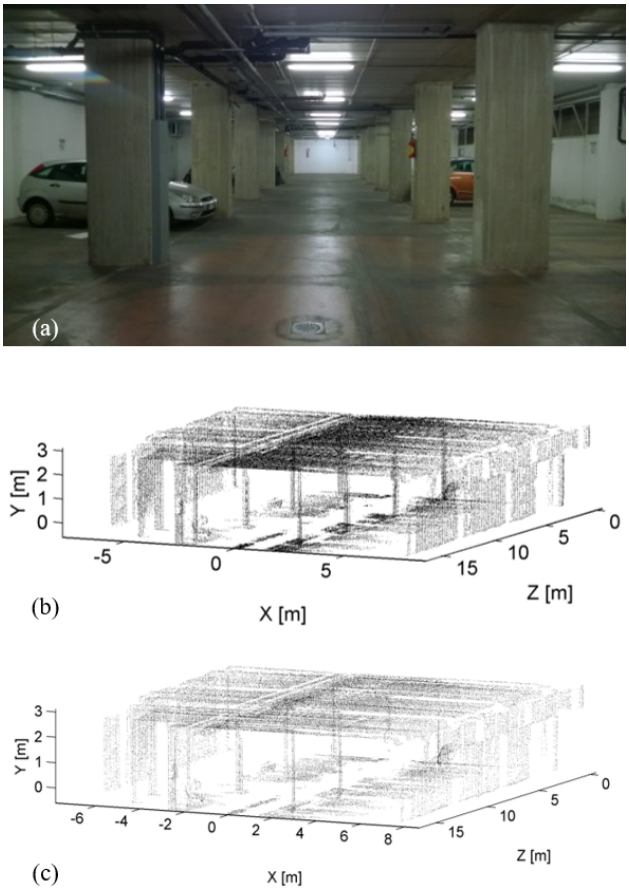

Figure 4.8: (a) Covered parking acquired by the laser rangefinder. (b) Corresponding reference dataset obtained by the AccuRange AR-4000 laser rangefinder. (c) Simplified point cloud obtained by the application of the preprocessing procedures.

To prove the efficiency of the preprocessing procedures, Figure 4.8a displays an example of indoor environment acquired with the proposed setup. A covered parking is modeled by the point cloud in Figure 4.8b, whose samples are referred to a local system of coordinates, having origin in the center of the first slice of points. The application of the preprocessing steps produces the point cloud in Figure 4.8c, which is almost three times smaller in size than the starting one. In summary, as an effect of the preprocessing steps, all point clouds considered in these experiments have sizes in the range between 7×10^4 and 10^5 points.

Error metrics for result comparison

The results of the registration processes will be compared with those returned by other ICP algorithms. In this case three variants of the ICP implementation have been considered: the standard linear ICP (Lin-ICP) solved by means of the Single Value Decomposition (SVD) [105, 106], the non-linear ICP (NL-ICP) proposed in [115] which is directly solved as a Levenberg-Marquardt (LM) optimization problem, and an optimized linear ICP variant with the point-to-plane (Pt2Pl) metrics [117]. All algorithms used for the comparison are available online as a part of the point cloud library (PCL) [188].

Following the same strategy adopted by marker-based approaches, several landmarks are used to obtain an effective comparison with a reliable ground truth. In the proposed experiments, the environment under analysis has been structured with seven high-reflection markers (see Figure 4.9), named as M_k, $k = 1, \ldots, 7$, whose position is chosen in order to investigate all degrees of freedom (four markers on the side walls of the parking area, two on the ground, and one on the floor) and to obtain their detection from each point of view. These markers can be easily distinguished within the datasets by looking at the intensity of the laser spot (this value is returned by the sensor for each range sample). Hence, the error metric is defined as the distance (d_x, d_y, and d_z along the three corresponding axes) between homologous markers extracted from the reference cloud P_0 and the source one, after its registration. Specifically, the exact marker position is assumed as the center of mass of the cluster which models the marker. In this way the measurement uncertainty is divided by the number of points of the cluster, thus becoming negligible in the evaluation of the registration error.

Figure 4.9: Reflective marker used for the point-by-point comparison of registrations.

Furthermore, it is worth noting that the position of the markers in the point clouds is established through odometry, since the dataset creation makes use of the position of the vehicle to translate range values into spatial 3D coordinates. This gives in turns an error in the localization of such points, since the vehicle position is determined with the measurement uncertainty of the encoders. Nevertheless, the comparison of results obtained by the proposed method and the others ICP variants is consistent, since all the methods are applied on the same datasets. As a consequence, the uncertainty will produce the same bias errors in the distance measurements between the homologous markers.

Model optimization

Although the formulation of the presented method is general and can be applied in any context, given the specific case of study, some simplifications are imposed to increase efficiency, downing computational requirements. As a first step, it is possible to take advantage of the measurements purposes. As stated before, environmental monitoring aims to understand whether changes affects the scene under analysis. Consequently, the vehicle has to sense the environment from points of view that have to be close to the one of the reference. In this case the comparison makes sense since the same targets, which are constitutive of the scene, can be compared. As a consequence, all paths followed by the vehicle are almost comparable, but not equal.

Moreover, in the scenario of environmental monitoring, 3D reconstructions will be performed exploiting the same experimental setup, i.e. with fixed

elevation of the sensor on the mobile vehicle. Hence, consecutive measurements are affected by relative alterations of the reference system in the xz-plane. Analytically, any alteration of the vehicle trajectory can be compensated by means of two translations X, Z along the x_i- and z_i-axis, respectively, and a rotation H around the y_i-axis. This hypothesis introduces a simplification in the registration scheme and consequently reduces the computational time required to perform the cost optimization described in Section 4.1.3, without a corresponding degradation of the final results, as it will be shown in the next subsections.

4.2.2 Experiments and results

Several experiments have been run to compare the results of the registration obtained with the proposed method with those returned by three ICP variants.

Two different conditions are discussed in the next subsections to prove the robustness of the registration. In the first case the datasets are extracted from the same environment (static environment), sensed from different points of view, i.e. trajectories. Then, the acquisitions will be performed still on the same environment, but introducing some alterations (changing environment).

Finally, acquisitions of an indoor environment, the entrance hall of a building under construction, will be registered, to further compare the proposed methodology with the existing ones.

Acquisitions of static environments

In the first experiments, the dataset registration is performed on static environments, i.e. perfectly equal scenes. As an effect, although surroundings do not change with respect to the reference dataset, relative differences among the point clouds arise because of the alteration of the vehicle trajectories and the measurement noise.

Three source datasets P_1, P_2, and P_3, in addition to the reference P_0, have been acquired at different epochs following different trajectories. Here, to prove the robustness of the method, P_3 has different spatial resolution along the direction of motion of the vehicle. In particular its size is almost halved with respect to P_0. As an example, the comparison of the datasets P_0 and P_3 is shown in Figure 4.10.

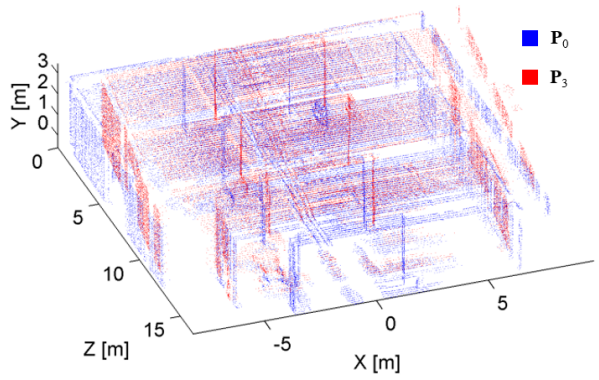

Figure 4.10: Original datasets referred on the local reference system of the laser rangefinder. Blue and red dots belong to different datasets to be registered, namely P_0 and P_3, respectively.

Following the theoretical description in Section 4.1.3, the deletion masks have been determined starting from the choice of the trial parameters X_t, Z_t and H_t. In this case the number of points Q drawn from the input datasets in the surface resampling task has been imposed equal to 38400, corresponding to 160 slices having 240 samples. An example of deletion mask is reported in Figure 4.11, where the points of the resampled data extracted from P_0 are colored accordingly with the values assumed by the mask.

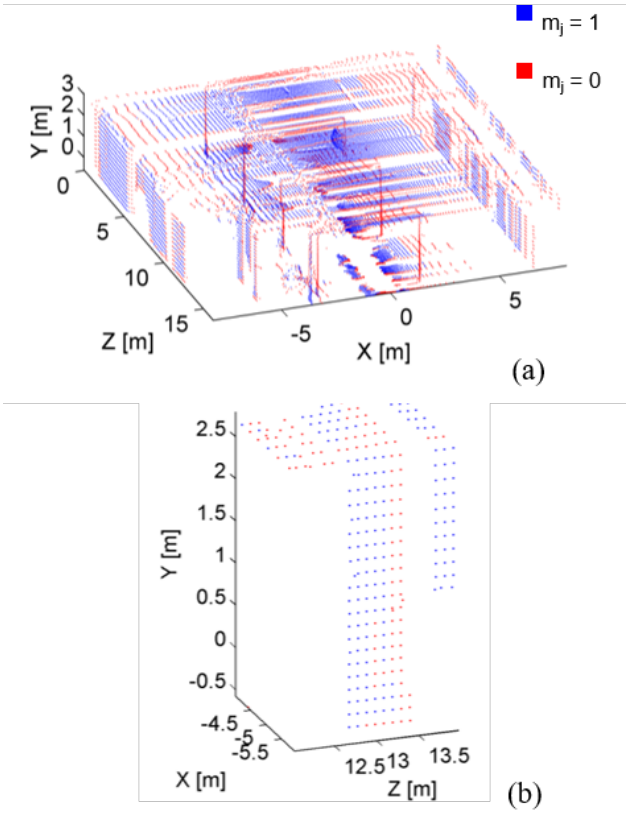

Figure 4.11: (a) Effects of the masking process: red points are neglected in the pair-wise registration of datasets; (b) Magnified view of the deletion mask applied to the samples extracted from the reference.

Figure 4.12 shows the first results of the registration of P_3 on the reference P_0 performed by the linear ICP algorithm and the proposed variant, which employs the deletion masks (see Figure 4.12a and 4.12b, respectively).

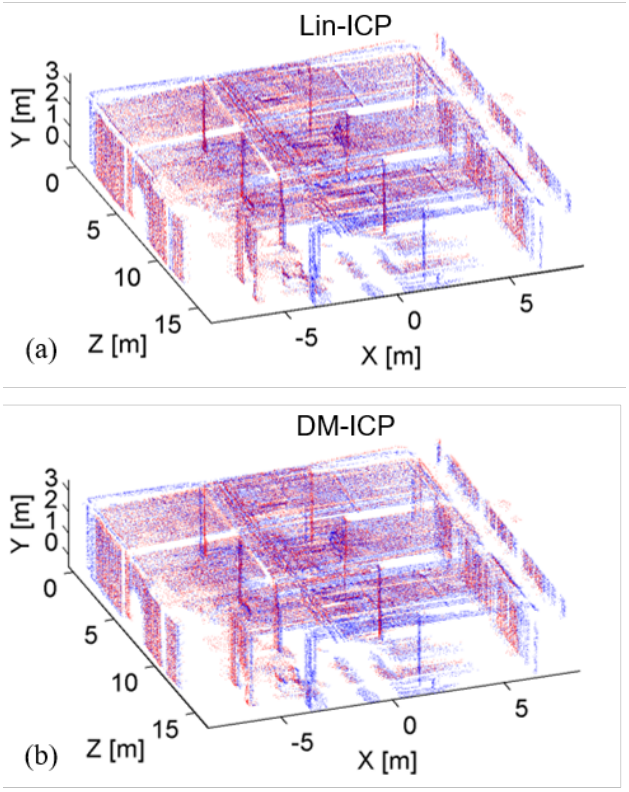

Figure 4.12: (a) Results of the dataset registration performed with the Lin-ICP algorithm. (b) Point clouds registered by the use of the proposed algorithm based on the use of deletion masks.

Although the results seem to be comparable, the estimated correction parameters differ in values. This consideration is further proved by the analysis of Table 4.1, which reports the correction parameters estimated by the four considered ICP algorithms. In particular, the parameters obtained by the Lin-ICP and the DM-ICP produce the vehicle trajectories described by the vectors in Figure 4.13.

Table 4.1: Results of the registrations of the source datasets P_1, P_2 and P_3 on the reference dataset P_0. X_0 and Z_0 are expressed in millimeters (Lin: standard linear ICP; NL: non-linear ICP; Pt2Pl: Point-to-Plane metrics; DM: Deletion Mask).

			X_0	
	Lin	NL	Pt2Pl	DM
P_1	-1121.1	-998.06	-1239.06	-1230.1
P_2	61.14	29.25	109.75	138.11
P_3	-1341.7	-1118.27	-1570.83	-1598.3
			Z_0	
	Lin	NL	Pt2Pl	DM
P_1	152.81	118.53	191.85	242.61
P_2	259.08	196.4	441.96	461.68
P_3	240.38	174.2	356.87	370.72
			H_0	
	Lin	NL	Pt2Pl	DM
P_1	4.94°	4.29°	5.67°	5.49°
P_2	-2.35°	-2.01°	-2.75°	-3.09°
P_3	4.73°	3.5°	6.13°	6.19°

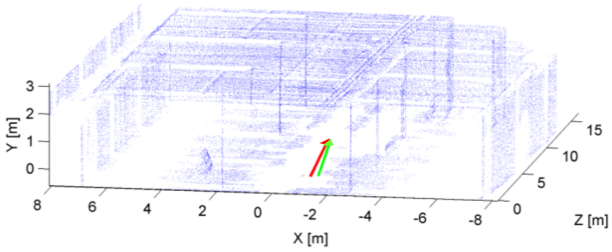

Figure 4.13: The red and green arrows are the robot trajectories within the reference point cloud, estimated by the Lin-ICP and the proposed method, respectively.

Table 4.2 summarizes the minimum, maximum and average values of the distances computed between corresponding reflecting markers extracted from the reference dataset and the registered ones. Bold values indicate the best results achieved by the comparisons.

The insight into the results of Table 4.2 reveals that the use of the deletion

Table 4.2: Minimum, maximum and mean distance values [mm] between corresponding reflective markers extracted from the registrations of P_1, P_2 and P_3 on P_0. The best results are highlighted in bold (Lin: standard linear ICP; NL: non-linear ICP; Pt2Pl: Point-to-Plane metrics; DM: Deletion Mask).

		d_x			
		Lin	NL	Pt2Pl	DM
P_1	Min	69.85	92.93	4.75	**2.23**
	Max	343.17	251.66	**75.41**	76.33
	Mean	192.77	156.33	39.88	**37.94**
P_2	Min	**2.82**	5.89	6.37	19.48
	Max	**63.44**	112.96	77.78	125.19
	Mean	**30.01**	38.7	30.48	44.77
P_3	Min	57.73	157.39	17.26	**10.05**
	Max	246.7	426.76	89.62	**63.89**
	Mean	152.39	273.28	45.9	**36.85**
		d_y			
		Lin	NL	Pt2Pl	DM
P_1	Min	**0.24**	3.32	0.53	1.08
	Max	**8.92**	21.94	21.43	21.59
	Mean	**2.34**	8.55	6.41	6.45
P_2	Min	**1.99**	5.98	3.89	3.76
	Max	40.27	33.87	29.05	**28.72**
	Mean	20.43	16.16	17.47	**15.06**
P_3	Min	**0.08**	0.85	3.02	1.7
	Max	**23.9**	43.01	50.01	33.05
	Mean	**6.92**	22.54	17.23	13.83
		d_z			
		Lin	NL	Pt2Pl	DM
P_1	Min	1.62	2.49	13.75	**0.41**
	Max	104.88	205.21	105.53	**58.25**
	Mean	38.47	118.1	60.55	**34.31**
P_2	Min	69.7	175.78	1.04	**6.43**
	Max	301.18	348.24	136.18	**130.88**
	Mean	186.78	260.21	41.15	**39.5**
P_3	Min	**10.35**	99.12	14.6	10.82
	Max	237.19	321.71	133.61	**112.55**
	Mean	117.02	194.91	68.2	**62.97**

masks can improve the estimation of the registration parameters, since the distance components d_x and d_z are always lower when the deletion masks are used. This scenario is altered only in the case of the analysis of P_2, whose registration performed by the linear ICP induces the lowest values of the term d_x. However, the decrease of the mean value of d_x is much lower in magnitude than the improvements produced by the DM-ICP in the remaining cases (see the registrations of P_1 and P_3).

At the same time, results in Table 4.2 show a different behavior of the distance term computed along the y-axis (d_y). The linear ICP often carries to the best results in comparison with the other methods, although improvements are in any case below those obtained for the comparison of the other two components d_x and d_z. This behavior is mainly ascribable to the experimental setup used for the experiments. In fact, the only contribution responsible for the distance component d_y is the measurement noise. It is clear that changing the algorithm, making it heavier, with the sole intention of compensating for noise would not produce appreciable improvement of the overall results. In other terms, although the analytical formulation of the Lin-, Nl- and Pt2Pl-ICP considers rototranslation matrices full of non-vanishing entries, it does not improve significantly the results.

An easier comparison can be derived by means of the figure of merit ε_M which depends on the global average value of the distance vectors made of the three components (d_x, d_y, d_z). Analytically, it is equal to:

$$\varepsilon_M = \underset{M_k}{\text{mean}}\{\sqrt{d_x^2 + d_y^2 + d_z^2}\} \tag{4.4}$$

where the mean function is first computed among the corresponding marker distances. This figure of merit is plotted in Figure 4.14.

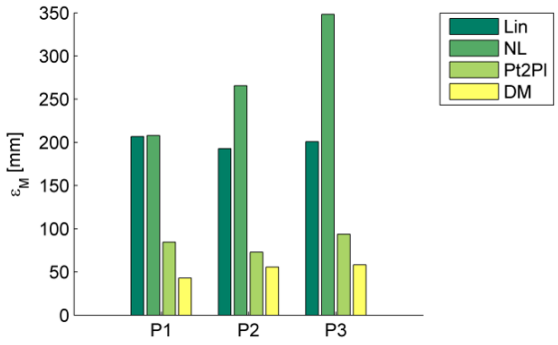

Figure 4.14: Comparison of results obtained by the use of the four ICP variants. The bar plot displays the values of the figure of merit ε_M, defined to compare the registration outcomes.

The analysis of the results states a clear reduction of the distance errors. Specifically, averaging the ε_M values among the three registrations, the mean values of ε_M are equal to $197.02mm$ for the Lin-ICP, $273.94mm$ for the NL-ICP and $83.75mm$ for the Pt2Pl-ICP, whereas the homologous term for the proposed algorithm is equal to $62.46mm$. Also, this result proves that the initial hypothesis of alteration of the vehicle trajectory in the xz-plane, does not lead to appreciable registration errors.

Finally, it is important to notice that the numerical gap found by the comparison of the ε_M values is much higher than the measurement uncertainty produced by the sensor, close to few millimeters. As a consequence, this result is only attributable to the effective contribution brought by the methods to the registration process.

Acquisitions of changing environments

The comparison of changing environments, i.e. scenes with small differences, is the most challenging problem in the dataset registration, since the cost function takes into account also the presence of scene alterations. In this case, the distance between the two considered point clouds can be significantly different from zero, till the limit of turning into a local minimum. In this case, the ICP algorithm reaches the convergence with registration parameters which can be significantly different from the correct ones.

These experiments have considered three new acquisitions, namely P_4,

P_5 and P_6, acquired at different epochs within the same environment, after that the position of several foreground objects has changed. In particular, another car is added in the parking, producing an alteration of 2% of points of the reference dataset. Quantitatively, the point cloud P_4 shows 1463 altered points over the total size of 79073 samples. A comparison between the dataset P_4 and the reference P_0 is reported in Figure 4.15, where the two point clouds are displayed.

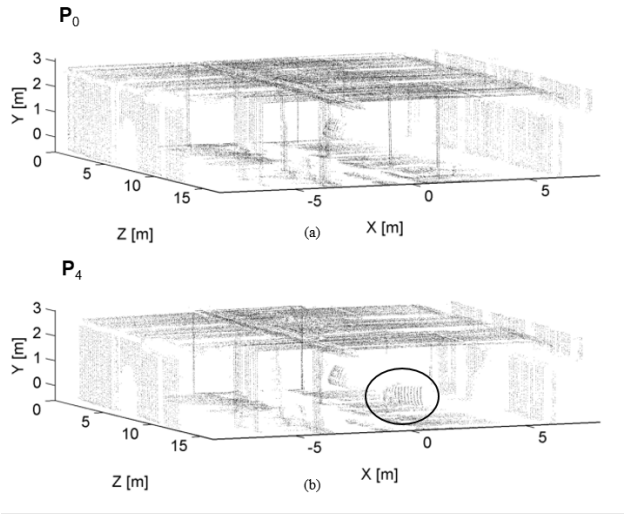

Figure 4.15: . Comparison between (a) the reference P_0 and (b) the source point cloud P_4. The circle includes the altered points.

Also in this case, the P_6 dataset has been created by halving the spatial resolution along the straight trajectory followed by the vehicle, i.e. doubling the speed of the vehicle that carries the sensor, but keeping the remaining set of measurement parameters. The results of the registration process are thus reported in Table 4.3, where the estimated parameters derived by the four methods under analysis are shown.

Results are once more in contrast and produce different distances between corresponding markers. As shown in Table 4.4, which describes the minimum, maximum and mean distance contributions computed among the homologous markers of different datasets, the use of deletion masks can reduce the registration errors. This consideration is verified for the analysis of the d_x component, whose values obtained by the DM-ICP are better than the

Table 4.3: Results of the registrations of the source datasets P_4, P_5 and P_6 on the reference dataset P_0. X_0 and Z_0 are expressed in millimeters (Lin: standard linear ICP; NL: non-linear ICP; Pt2Pl: Point-to-Plane metrics; DM: Deletion Mask).

			X_0	
	Lin	NL	Pt2Pl	DM
P_4	-154.06	-123.95	-173.29	-237.29
P_5	-935.13	-820.66	-1062.38	-1085.5
P_6	118.53	-253.26	107.22	86.2
			Z_0	
	Lin	NL	Pt2Pl	DM
P_4	219.71	167.16	515.26	482.02
P_5	142.2	109.1	232.78	291.26
P_6	315.91	207.09	674.43	674.42
			H_0	
	Lin	NL	Pt2Pl	DM
P_4	1.5°	1.27°	1.65°	2.3°
P_5	2.3°	1.8°	2.96°	2.95°
P_6	-5.54°	-3.6	-8.05	-7.81°

Table 4.4: Minimum, maximum and mean distance values [mm] between corresponding reflective markers extracted from the registrations of P_4, P_5 and P_6 on P_0. Best results are highlighted in bold (Lin: standard linear ICP; NL: non-linear ICP; Pt2Pl: Point-to-Plane metrics; DM: Deletion Mask).

		d_x			
		Lin	NL	Pt2Pl	DM
P_4	Min	18.02	38.55	4.9	**1.19**
	Max	55.66	83.35	72.35	**72.08**
	Mean	36.46	51.36	35.67	**33.96**
P_5	Min	9.62	17.29	**6.89**	7.91
	Max	205.36	294.92	109.33	**83.49**
	Mean	95.14	141.38	41.32	**38.21**
P_6	Min	45.8	71.47	**5.06**	14.12
	Max	126.27	149.42	49.6	**49.23**
	Mean	78.55	110.44	28.23	**25.93**
		d_y			
		Lin	NL	Pt2Pl	DM
P_4	Min	2.90	4.06	1.26	**1.25**
	Max	**24.29**	28.57	26.58	33.32
	Mean	13.95	14.48	**10**	14.24
P_5	Min	0.46	**0.34**	2.67	2.81
	Max	**38.03**	59.04	63.96	62.47
	Mean	**18.16**	20.13	24.07	22.13
P_6	Min	**0.7**	24.75	3.23	13.81
	Max	**14.38**	27.31	14.96	16.4
	Mean	**5.59**	26.03	8.14	15.72
		d_z			
		Lin	NL	Pt2Pl	DM
P_4	Min	73.88	141.06	**1.46**	9.73
	Max	302.65	364	166.24	**160.59**
	Mean	200.59	275.02	59.76	**59.02**
P_5	Min	34.09	36.48	**11.26**	59.33
	Max	126.49	457.62	190.22	**132.22**
	Mean	91.96	148.09	**38.3**	71.48
P_6	Min	132.43	149.82	**68.56**	90.3
	Max	422.07	524.92	270	**233.8**
	Mean	310.31	337.37	**136.53**	174.24

others in most cases. On the contrary, the inspection of results shows again comparable values of d_y obtained by the four methods, although Lin-, NL- and Pt2Pl-ICP exploit the full transformation matrices, whereas the proposed technique simplifies the problem to the optimization of only three terms. Although measurement noise determines a contribution to the overall cost function, which cannot be compensated by the DM-ICP, its outcomes are in any case comparable. Once again it justifies the initial downgrade of the problem to the compensation of the vehicle trajectory with only three degrees of freedom.

Results obtained by the Pt2Pl-ICP and the DM-ICP in terms of the d_z component are highly comparable in magnitude. In this case, it is important to observe that the point-to-plane metrics allows the reduction of the contributions to the objective cost function of erroneous correspondences between samples. This filtering effect is noticeable especially in these last experiments, when the environments under testing show relative changes. In principle, the method weights such correspondences, exploiting the surface similarity. With more details, the point distance is multiplied by a term (dot product of surface normals) which is 0 when the two surfaces are orthogonal and 1 when the two surfaces are parallel. At a first glance, this metrics seems to limit wrong correspondences in the cost computation in a similar way to the DM approach, thus producing comparable results. Actually, given the weight formulation, the Pt2Pl metrics is not able to discriminate the presence of scene changes due to the movement of objects having parallel surfaces to the ones placed in the corresponding regions of the reference point cloud. Consequently, the point-to-plane metrics fails and the cost term can grow as much as the objects position changes. On the contrary, the use of virtual resampling and DMs can automatically and perceptively remove wrong point correspondences, regardless the relative direction of the surface normals. Nevertheless, results in Table 4.4 demonstrate that the point-to-plane error metrics can add reliability to convergence of ICP algorithms. Its implementation in the proposed method will be the aim of future investigations.

Then, with reference to the outcomes displayed in Figure 4.16, where the figure of merit ε_M is presented, the use of deletion masks improves the registration process in two cases out of three. Quantitatively, the average value among registrations of ε_M reaches $223.28mm$ in the case of the Lin-ICP, $297.6mm$ for the NL-ICP, $86.27mm$ for the Pt2Pl-ICP and $65.06mm$ for the proposed algorithm. By a comparison of these results with those displayed in the previous subsection, it can be stated that the proposed method is robust

against consistent scene alterations, since the average value of ε_M does not change as scene differences arise.

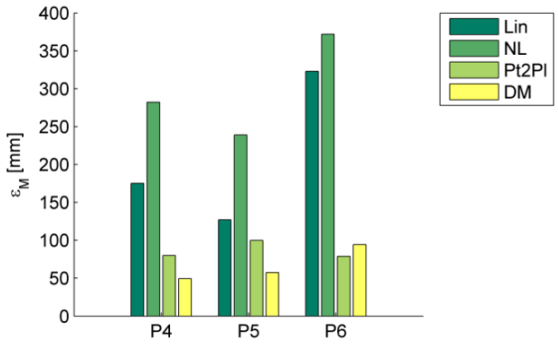

Figure 4.16: Comparison of results obtained by the use of the four ICP variants. The bar plot displays the values of the figure of merit ε_M, defined to compare the registration outcomes.

Further analysis of an indoor environment

The proposed method has been further tested for the registration of two point clouds obtained by the inspection of another environment, namely the entrance hall of an under-construction building, in order to prove the quality of the algorithm. The specific entrance hall constitutes a challenging indoor environment because of its spatial uniformity due to the lack of pillars, whose shapes and position were highly informative in the previous registrations.

As already discussed for the previous investigations, the experiment has been performed by changing the pose assumed by the vehicle before starting its movement: two point clouds model the same environment from different points of view. Figure 4.17 shows the two point clouds extracted from the inspection of the entrance hall and referred to the local reference system of the corresponding acquisition.

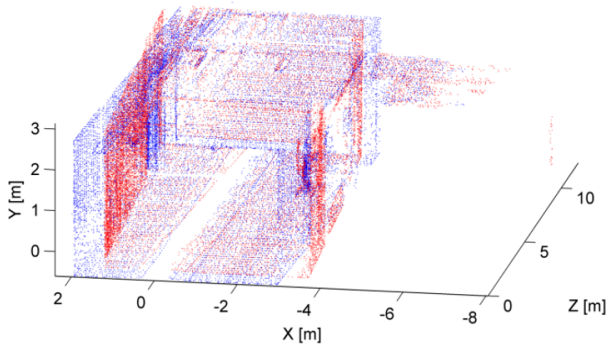

Figure 4.17: Original datasets acquired from an entrance hall. Point clouds are referred on the local reference system of the sensor.

Also in this case, the proposed method for point cloud registration has been compared with the three considered ICP implementations (Lin, NL and Pt2Pl), giving raise to the results in Figure 4.18, which plots separately the top views of the source point clouds, registered on the reference one.

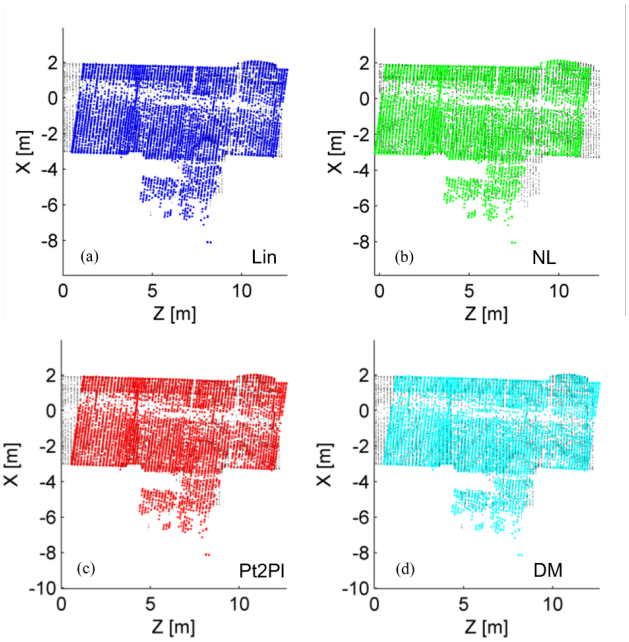

Figure 4.18: Top views of the source point clouds (colored data) registered on the reference one (black data). The registration process is performed exploiting the (a) Linear ICP, (b) the Non-linear ICP with kD-tree representation of points, (c) the standard ICP with point-to-plane metrics and (d) the proposed ICP with deletion mask.

With more details, Figure 4.19 highlights the differences between the reference dataset and the source one registered by means of the Lin-ICP, the Pt2Pl-ICP and the proposed method. The focus on Figure 4.19 reveals that the Lin-ICP can poorly register the input datasets. On the other hand, the Pt2Pl-ICP and the proposed DM-ICP are in good agreements with comparable results, although the DM-ICP makes use of a simpler distance metrics and a registration scheme dealing with three parameters. Quantitatively, the three correction parameters found by the Pt2Pl-ICP are $X_0 = 810.92mm$, $Z_0 = 1089.96mm$, and $H_0 = -8.12°$, whereas the DM-ICP returns $X_0 = 813.48mm$, $Z_0 = 1071.09mm$, and $H_0 = -8.18°$. Here, differences between corresponding parameters are negligible, since these terms are slightly higher than the measurement uncertainty of the sensor. In summary, as stated by the inspection of the previous experiments, it is possible to envisage even better results by implementing in the proposed algorithm the point-to-plane distance metrics, which weights correspondences between points on the similarities between planes.

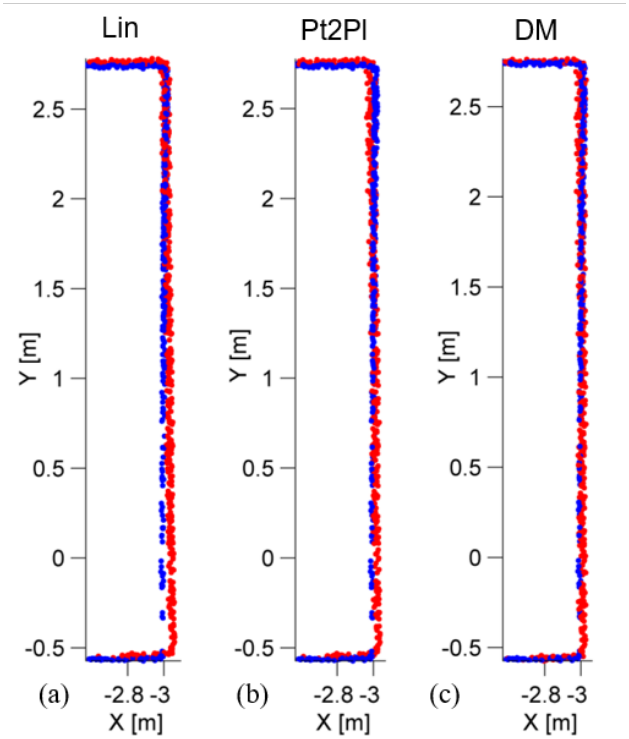

Figure 4.19: Comparison of the reference point cloud (blue dots) and the registered one (red dots) extracted from the results in Figure 4.18. Results of (a) the Lin-ICP, (b) the Pt2Pl-ICP and (c) DM-ICP.

4.3 Summary

In this work, a numerical approach for point cloud registration returned by a laser rangefinder has been presented. The analysis has been focused on the topic of remote sensing of indoor civil infrastructures, where standard approaches based on GPS are no longer available. Acquisitions are thus referred to a local reference system having origin in the starting position of the vehicle that carries the sensor. In this case, occlusions can emerge when the point of view of the sensor changes, and thus consecutive reconstructions of the same environment can suffer from implicit differences. For this reason deletion masks have been introduced iteratively within the standard ICP technique to delete those points that can induce erroneous registrations.

The method has been applied for the registration of datasets extracted from actual environments, namely a covered parking and an entrance hall, where scenes are equal or slightly altered. Several comparisons with three well-known ICP variants have been performed by computing the distances between distinguishable markers extracted from the reference dataset and the registered ones. Outcomes have proved a reduction of the registration errors, with respect to the other implemented ICP variants. Only the use of the point-to-plane distance metrics between the datasets has lowered the negative effects of erroneous correspondences, with results often similar to those of the presented method, which implements the simpler point-to-point metrics. This behavior suggests that future developments of the method will use a more effective error metrics to further minimize the negative effects of wrong point correspondences.

Chapter 5

Real time algorithms for high throughput data processing

As introduced in Chapter 1, there is a certain number of features that needs to be implemented in an artificial vision system to perform high level tasks. The technology available nowadays is capable of producing high throughput data and opens new perspectives in data processing and analysis, since algorithms should be properly designed and implemented to meet the most challenging system requirements. Special attention needs to be given to the development of real time algorithms that represent the building blocks of more complex 3D vision systems. In particular, the methodologies used to pre process the images that lead to the extraction of sparse point clouds will be presented, as well as the methodology defined to perform three dimensional tracking.

For this reason, three background models have been developed with the aim of fulfilling both real time and effectiveness constraints and will be presented in this Chapter. The first, named Parallel Integer Incremental Background (PIIB) is an adaptive background model for high frame rate video applications suitable for smart cameras embedding due to its implementation. The second, named Likelihood Bayer Background (LBB), is a BG model based on statistical likelihood that directly works on Bayer images taking into account the intrinsic variance of each gray level of the sensor. The last one, named Global Intrinsic VariancE BACKground (GIVEBACK), starts from the formulation of LBB and adds some processing modifications essential to address the specific problems related to the tennis context. Finally, a tennis ball tracking method that exploits domain knowledge to effectively recognize ball positions and trajectories starting from a sparse but cluttered

point cloud that evolves over time – basically working on 3D samples only – will be introduced.

5.1 PIIB

5.1.1 Algorithm description

The proposed algorithm, named *Parallel Integer Incremental Background* (PIIB), can be divided into three main steps, as it is shown in figure 5.1:

1. background initialization (in blue);
2. foreground extraction (with energy and threshold processing) (in green);
3. background update (in red).

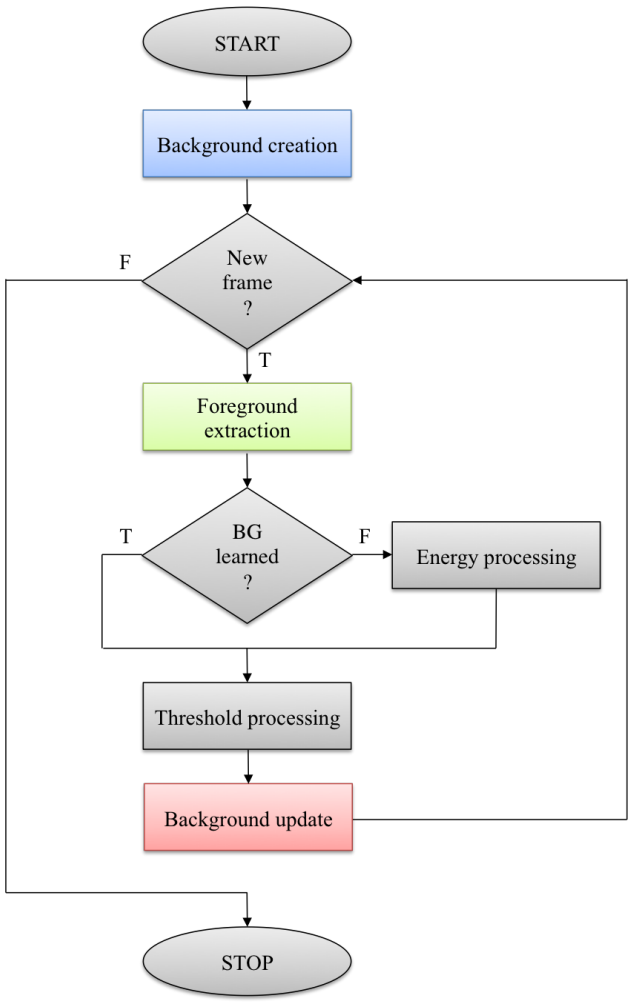

Figure 5.1: PIIB high level flowchart.

The first step is executed only once and prepares the environment. In this phase, the width (w) and the height (h) of the frames are saved and the new data structures are allocated in memory taking advantage of the Streaming SIMD Extensions 2 (SSE2) [189], an instruction set that enables multiple data operations on Intel CPUs. Then, the whole frame is divided in $\left(\frac{w \cdot h}{16}\right)$ Update Rectangle Structures (URSs), which are row vectors of 16 entries, assisted by a corresponding boolean flag which sets the pixel update (initially set to true). Finally, if a background file that contains a valid preprocessed

background is missing, the model is initialised to 128. This *all gray* logic is due to the absence of any *a priori* knowledge about the scene.

The other steps need to be extremely efficient in order to match the real time, high frame rate constraint of this procedure. The foreground extraction takes place for every incoming frame, calculating the binary foreground mask at the time t (M_t). Each frame is processed $\left(\frac{w \cdot h}{16}\right)$ times because SSE2 instructions can execute 16 operations simultaneously, boosting the performances of the whole procedure. Therefore, this approach makes the algorithm parallel because it works on 16 pixels at the same time. Assuming that I_t is the frame at the time t, BG_{t-1} is the background image at the time $(t-1)$ and Thr is the gray threshold, the Addition and Subtraction Vector (ASV) can be defined as follows:

$$ASV = (S_1 \ominus Thr) \oplus (S_2 \ominus Thr) \tag{5.1}$$

where

$$S_1 = (BG_{t-1} \ominus I_t) \tag{5.2}$$

$$S_2 = (I_t \ominus BG_{t-1}) \tag{5.3}$$

$\ominus$ and $\oplus$ are the saturated version of the subtraction and the addition operations in the range $[0 \dots 255]$. Then, the values of the foreground mask corresponding to the considered URS are calculated from the ASV and are defined as follows:

$$M_t = [M_t^0 \dots M_t^{15}] \tag{5.4}$$

$$M_t^k = \begin{cases} 0 & if ASV^k = 0 \\ 255 & if ASV^k \neq 0 \end{cases} \tag{5.5}$$

where k is the index of the pixel in the URS.

If the background is not completely learned and the frame number is lesser than 128, the energy processing is invoked in order to evaluate the energy of the background signal. Then, the image histogram is used to calculate the best gray threshold value. Further explications will be provided in the next paragraph.

The third step is the background model update, which works according to the 16 bytes logic presented beforehand. The basic idea is that every pixel

of BG_t can increase or decrease by 1 its gray intensity depending on S_1 and S_2. The generic background pixel of coordinates (x, y) will be:

$$BG_t(x,y) = \begin{cases} BG_{t-1}(x,y) - 1 & \text{if} BG_{t-1}(x,y) > I_t(x,y) \\ BG_{t-1}(x,y) & \text{if} BG_{t-1}(x,y) = I_t(x,y) \\ BG_{t-1}(x,y) + 1 & \text{if} BG_{t-1}(x,y) < I_t(x,y) \end{cases} \tag{5.6}$$

This extremely simple and fast instruction is adapted to the SSE2 logic, through the use of the URSs. If and only if the allowance flag is set to true, the 16 pixels of the structure are updated as shown before. The flag value can be set by a procedure that identifies stopped objects on the scene, making the algorithm able to update the background only if a foreground object is stationary.

5.1.2 Energy and threshold processing

The energy processing is a task that is executed during the background learning phase monitoring the energy $\varepsilon = ||BG_{t-1} - I_t||$ in order to stop the learning phase when it reaches its minimum value. In the worst case, 128 update frames must be analyzed to obtain a stable background model, since the initial background is supposed gray (see the previous paragraph). Nevertheless the background can become stable after $n < 128$ iterations and the learning process can be stopped earlier than the worst case. The evaluation of ε sets up a flag which is true if the energy computed in two consecutive frames does not decrease. In this case the learning process is stopped.

The threshold processing with reference to the flowchart in figure 5.2 and calculates the threshold value Thr defined as the gray intensity limit used to classify a background/foreground pixel. The PIIB initially sets $Thr = 10$ and then adaptively updates it during the threshold processing. This task is performed evaluating the gray scale image histogram with respect to the SSE2 instructions programming rules, i.e. approximating it using 1×16 vectors.

Let I_{t-1} be the previous frame, I_t be the incoming frame at time t and H_t the generic 256 bins histogram of $|I_{t-1} - I_t|$ at the same time t. Once the histogram is computed, this is normalized and smoothed via low-pass filtering in order to deal with isolated peaks and avoid local maxima. The smoothed curve S_t is used to calculate dynamically the gray threshold with the following

procedure: for every $i = 2, \ldots, 255$, the quantity $y = \frac{S_t(i+1) - S_t(i-1)}{2} + S_t(i-1)$ is evaluated and if $y > S_t(i)$, the threshold value is set to $Thr = i + 1$.

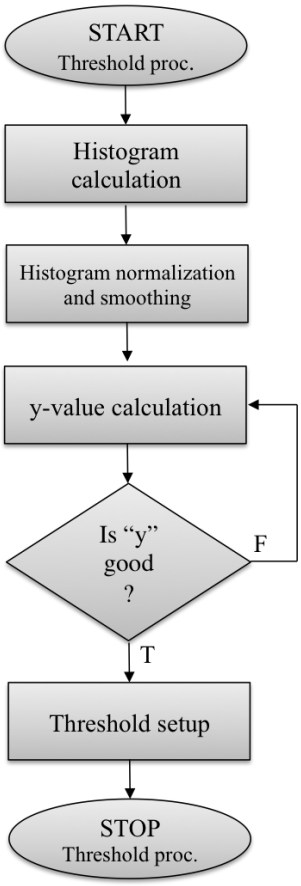

Figure 5.2: Threshold processing flowchart.

5.1.3 Computational complexity

The computational complexity of PIIB can be calculated splitting the analysis into the three building blocks, according to the flowchart in figure 5.1. Let $(w \cdot h) = n$ be the single frame dimension, where w is the frame width and h is the frame height, then the complexities are the following:

1. **Background creation** has a complexity of $\mathcal{O}(n)$ in the worst case, i.e.

when the BG is not known and the *all-gray* logic is implemented with n assignments;

2. **Foreground extraction** requires $\mathcal{O}(n)$ operations because the main loop repeats a small number of operations (absolute values and saturated differences) $\left(\frac{n}{16}\right)$ times;

3. **Background update** is done in $\mathcal{O}(n)$ operations, similarly to what happens in the previous phase

For these reasons, PIIB has a linear computational complexity that makes possible the actual implementation.

5.1.4 Experiments and results

PIIB is evaluated comparing its performances with the GMG and MoGv2 algorithms implemented in the BGS Library [190]. It has been tested on five different athletic videos taken with a Dalsa Pantera SA 2M30 camera and representing a football match, named AR1, AR2, AR3, FG1 and FG2. The AR- sequences are focused on the penalty area and each frame has a size of 1600×736 pixels. In the FG- sequences a larger area is filmed and the frame size is 1920×1280 pixels because the cameras are used to monitor the offside. Every background model is evaluated after 20, 40, 60 and 80 seconds after the starting frame f_0. At least 128 frames are used to build the BG model. Here it is a brief description of the scenes:

AR1 in this video the referee gives the signal for a penalty kick and there are many players on the scene. The illumination is changing due to a cloud on the outdoor field;

AR2 in this video a penalty kick is shooted. The scene is occupied by the attacking and defending players, together with the referee and about 10 other players from the two teams;

AR3 in this recording a free kick is shooted from the limit of the penalty area;

FG1 in this video the frames are taken from a wider point of view. The sequence starts with the goal keeper alone and ends while an action is being played;

FG2 this sequence is similar to the FG1 one, but here some players are doing the warm up, making the scene more dynamic.

The first column of table 5.1 displays some examples of the acquired frames. In every football match the advertising behind the touchline periodically changes over time modifying the background. The foreground masks are not postprocessed with morphological filters and the quantitative results are obtained calculating the F-Measure, Precision and Recall on four different frames, representing the considered time interval. Each evaluated frame has been manually segmented in order to obtain the ground truth. Let TP be the number of true positives pixels, FP be the number of false positives pixels, TN be the number of true negatives pixels and FN be the number of false negatives pixels on the foreground mask. Accordingly, Precision P, Recall R and F-Measure F are defined as:

$$P = \frac{TP}{TP + FP} \tag{5.7}$$

$$R = \frac{TP}{TP + FN} \tag{5.8}$$

$$F = 2 \cdot \frac{P \cdot R}{P + R} \tag{5.9}$$

Table 5.1 contains the qualitative analysis of some frames in terms of ground truth and foreground masks, while table 5.2 summarizes the metrics calculated for every video sequence. The comparison of PIIB Precision and Recall values against the best value among GMG and MoGv2 shows that the average PIIB R value is 22% better than the others, even if the P value is generally lower. This result demonstrates that PIIB is robust to false negative outputs. Figures 5.3 to 5.7 show the bar chart representation of the F-measure. In the AR1 and FG2 sequences (figures 5.3, 5.7) the F-Measure of PIIB and GMG is generally comparable. In the AR2 and AR3 sequences (figures 5.4, 5.5) PIIB is able to model a complex dynamic situation with many moving people in foreground, in fact the F-Measure is 20% better than GMG and MoGv2 on average. In the FG1 sequence (figure 5.6), PIIB starts with a low F-Measure (frame 1) because the background is noisy due to an advertising change, but the model is correctly updated in the subsequent frames (the F-Measure increases).

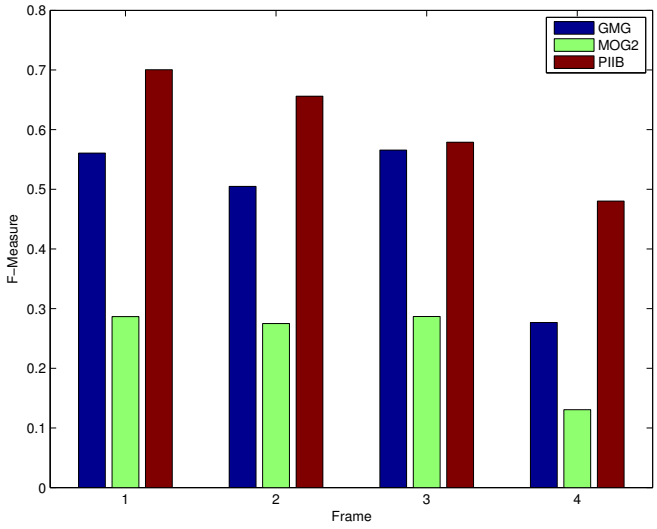

Figure 5.3: Bar chart representing the F-Measure comparison for the AR1 sequence.

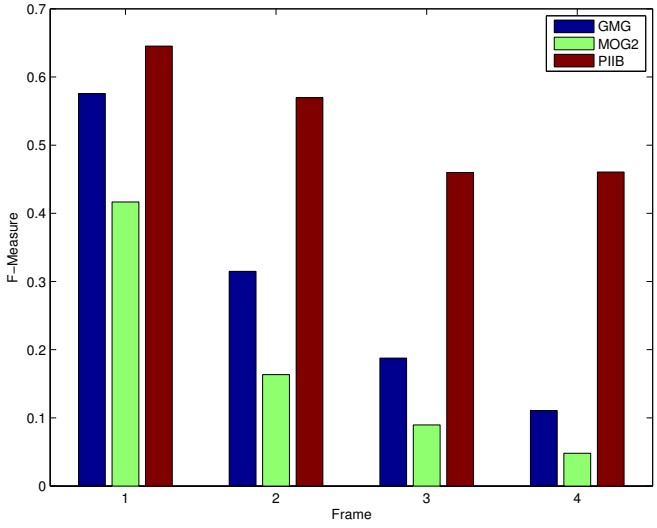

Figure 5.4: Bar chart representing the F-Measure comparison for the AR2 sequence.

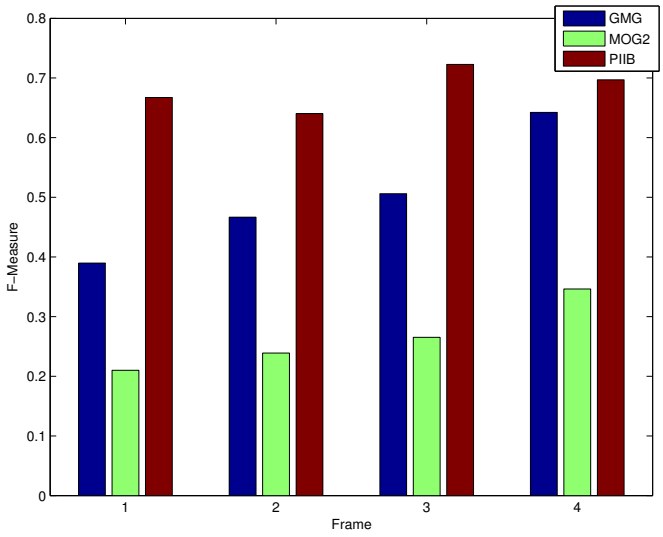

Figure 5.5: Bar chart representing the F-Measure comparison for the AR3 sequence.

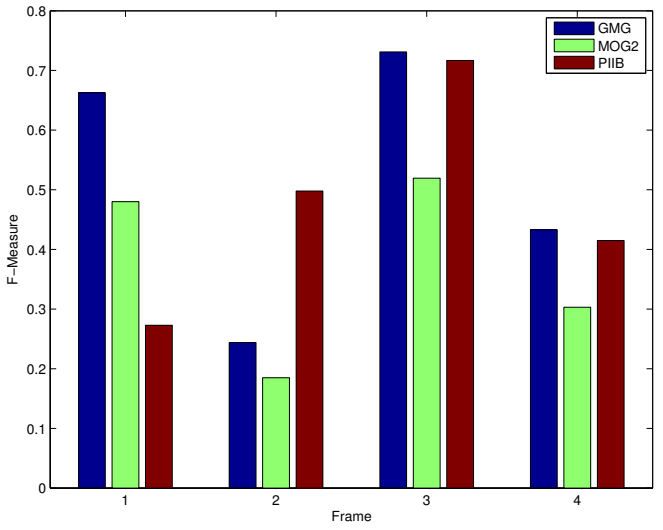

Figure 5.6: Bar chart representing the F-Measure comparison for the FG1 sequence.

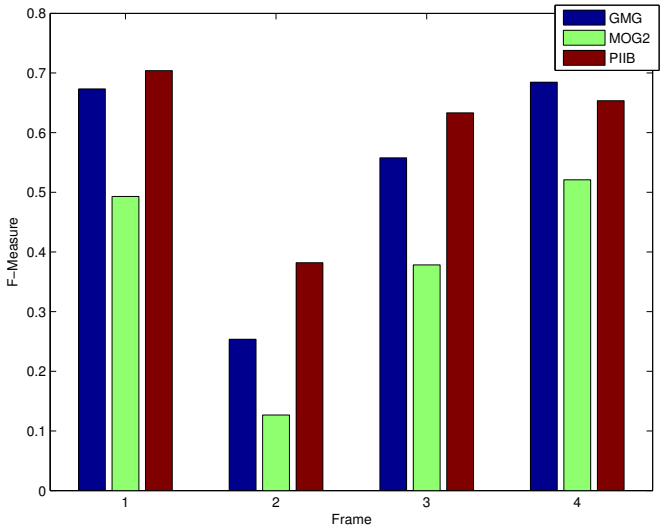

Figure 5.7: Bar chart representing the F-Measure comparison for the FG2 sequence.

Table 5.1: This table contains the qualitative analysis of some frames in terms of ground truth and foreground masks.

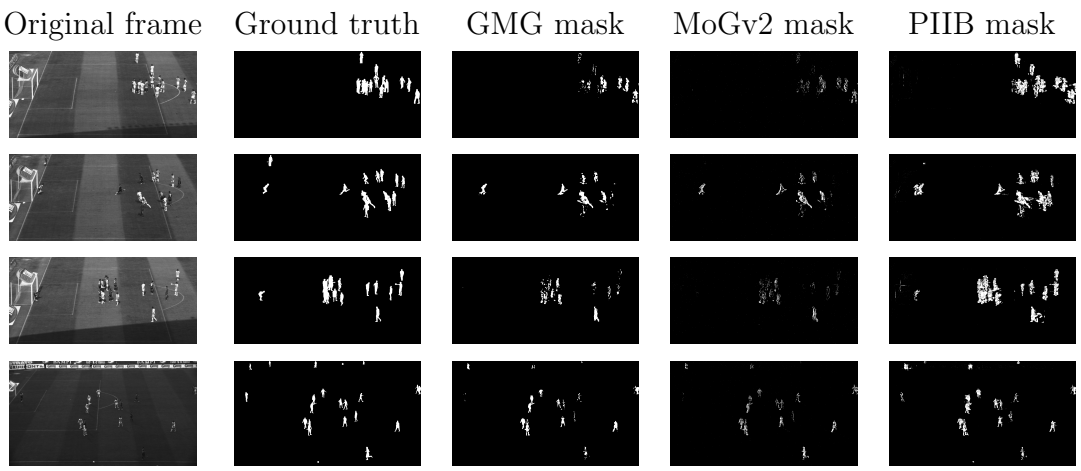

Table 5.2: This table summarizes the results in terms of F-Measure, Precision and Recall for the five scenes. In bold the best value among the three algorithms.

	F-Measure			**Precision**			**Recall**		
	GMG	MOG2	PIIB	GMG	MOG2	PIIB	GMG	MOG2	PIIB
AR1	0.56	0.29	**0.70**	**0.89**	0.76	0.69	0.41	0.18	**0.71**
	0.50	0.27	**0.66**	**0.93**	0.88	0.68	0.35	0.16	**0.64**
	0.57	0.29	**0.58**	**0.74**	0.61	0.51	0.46	0.19	**0.67**
	0.28	0.13	**0.48**	**0.98**	0.88	0.85	0.16	0.07	**0.33**
AR2	0.58	0.42	**0.65**	**0.83**	0.82	0.66	0.44	0.28	**0.63**
	0.31	0.16	**0.57**	**0.95**	0.82	0.68	0.19	0.09	**0.49**
	0.19	0.09	**0.46**	0.58	0.59	**0.62**	0.11	0.05	**0.36**
	0.11	0.05	**0.46**	**0.90**	0.62	0.72	0.06	0.03	**0.34**
AR3	0.39	0.21	**0.67**	**0.93**	0.89	0.74	0.25	0.12	**0.61**
	0.47	0.24	**0.64**	**0.92**	0.83	0.70	0.31	0.14	**0.59**
	0.51	0.27	**0.72**	**0.92**	0.86	0.69	0.35	0.16	**0.76**
	0.64	0.35	**0.70**	**0.88**	0.83	0.64	0.50	0.22	**0.77**
FG1	**0.66**	0.48	0.27	**0.87**	0.51	0.18	0.54	0.45	**0.58**
	0.24	0.18	**0.50**	**0.99**	0.60	0.74	0.14	0.11	**0.38**
	0.73	0.52	0.72	**0.91**	0.84	0.75	0.61	0.38	**0.69**
	0.43	0.30	0.41	0.66	**0.81**	0.36	0.32	0.19	**0.48**
FG2	0.67	0.49	**0.70**	**0.92**	0.79	0.80	0.53	0.36	**0.63**
	0.25	0.13	**0.38**	**0.93**	0.70	0.59	0.15	0.07	**0.28**
	0.56	0.38	**0.63**	**0.84**	0.71	0.78	0.42	0.26	**0.53**
	0.68	0.52	0.65	**0.86**	0.77	0.60	0.57	0.39	**0.72**

5.2 LBB

5.2.1 Algorithm Description

LBB is divided in three main building blocks that are summarized in Listing 5.1, namely initialization, processing and update and has been designed to work directly on raw data coming from a camera, namely a Bayer image [191]. The first step is executed only once and initializes the BG image setting each pixel to half intensity. This all gray logic is due to the absence of any *a priori* knowledge about the scene. The processing phase is composed of: variance, likelihood, fine tuning and energy. The last one is the same presented in the previous Section, while the other are detailed singularly in the following Sub sections. The BG image is updated according to PIIB logic, but it is enriched by a binary update mask M. Hence, each BG pixel value is increased or decreased by κ if the corresponding M value is set to true (in our implementation $\kappa = 1$). In addition, LBB calculates a second version of the background that does not take care of M (BG_{nu}) with the aim of avoiding ghosts on the scene, as it will be described later.

Listing 5.1: Algorithm pseudocode

```
Background Initialization
for each frame
    Variance process
    for each patch
        Likelihood process
    if(Background is learned)
        Fine tuning process
    Background Update
    Energy Process
```

5.2.2 Variance Process

The variance considered in the this method is not related to the observations of a single pixel over time, but is a function of the gray level and so it models the different responses of the sensor to different light intensities. Therefore, for each frame, the location of the occurrences of each generic gray value γ is first stored in a set

$$\mathrm{Obs}(\gamma) = \{k = (u, v) | BG(u, v) = \gamma\} \tag{5.10}$$

Then, the variance V at the time t, associated to the γ-th gray level is iteratively updated with the following formula:

$$V_t(\gamma) = \frac{V_{t-1}(\gamma) \cdot N_{t-1}(\gamma) + \sum_k |I_t(k) - BG(k)|^2}{N_t(\gamma)} \tag{5.11}$$

where $k \in \text{Obs}(\gamma)$, $N(\gamma)$ is the number of times the γ-th gray level occurred over time and BG is the background. In the equations BG is substituted with the latest available frame (I_{t-1}) while the BG is being learned, namely until the energy gradient descent reaches its minimum value. Figures 5.8 and 5.9 show an example of convergence of this model while estimating μ and σ values of known normal distributions, that will be discussed later.

5.2.3 Likelihood Process

This task is executed for each Bayer squared patch $P_i = (p_1, \ldots, p_4)^T$ of the image, so that P_i contains two green level values, a red one and a blue one. Considering the pixels as normal independent random variables, the likelihood of observing a background patch given a set of parameters $\theta = (\mu_1, \ldots, \mu_4, \sigma_1, \ldots, \sigma_4)$ can be calculated with the formula:

$$\mathcal{L}(\theta|P_i) = \prod_{j=1}^{4} f_{\mu_j,\sigma_j}(p_j) = \ell_i \tag{5.12}$$

where $\mu_j = BG(p_j)$, $\sigma_j = V_t(BG(p_j))^{\frac{1}{2}}$ and $f_{\mu_j,\sigma_j}(p_j)$ is the normal probability density function with mean μ_j and standard deviation σ_j computed in p_j. Therefore, the mean value of a pixel is its corresponding BG value, while the variance depends on its gray level, since different intensity values might have different variances. Following the same steps described in the previous section, the BG is substituted with the latest captured frame until the model is in the learning phase. Formally, a threshold $\tau_L = 0$ is used to classify each patch as background or foreground, but in our implementation $\tau_L = 10^{-10}$, considering that 0 can not be achieved due to noise and floating point representation issues. Experiments show that the value is small enough to guarantee a stable and reliable BG. The binary update mask of a BG patch is set to true, while it is false in case of a foreground patch. This selective update is useful to achieve robustness and to avoid useless updates when an object is moving on the scene.

5.2.4 Fine Tuning Process

The fine tuning task takes place only when the BG has been learned by the system and enriches the pipeline with two modules: a cosine similarity filter [192] and a ghost filter. The first one exploits the dot product between two vectors, specifically a foreground Bayer patch (P_f) and its corresponding background (P_b), both $\in \mathbb{N}^4$. The cosine of the angle between the two patches is filtered to blacken the foreground if it is similar to the background according to the following equation:

$$P_f = 0 \text{ if } \frac{P_f \cdot P_b}{|P_f||P_b|} > \tau_{\text{S}} \tag{5.13}$$

where $\tau_{\text{S}} \sim 1$.

The ghost filter is needed when there are no stable background frames at the beginning of a video, i.e. when the bootstrap phase contains almost stationary objects that are likely to be inserted in the background. In these cases, a movement of the object when the BG has been learned causes the presence of a ghost in the foreground. This phenomenon is removed comparing the incoming frame I_t with the background fully updated at each iteration BG_{nu} in correspondence of the ghost patch P_g. If $|I_t(P_g) - BG_{nu}(P_g)| = 0$, then the background is updated setting $BG(P_g) = BG_{nu}(P_g)$.

5.2.5 Experiments and Results

The model presented in Section 5.2 has been first tested in Matlab in order to numerically confirm its correctness. For this reason, samples from ~ 200 normal distributions with known (μ, σ) have been extracted. Figure 5.8 shows that, starting from 128 (half intensity for 1 byte unsigned variables), each estimated mean tends to the input one in ~ 100 frames in the worst case. The distribution with input mean $\mu = 119$ (magenta) converges immediately in a couple of iterations, while for $\mu = 20$ (blue) more iterations are needed to achieve the result. This is due to the update process that consists of unary increments or decrements at each iteration, as pointed out in Section 5.2.1. Figure 5.9 shows the convergence of the standard deviation estimator after $\sim$ 10M iterations. In particular, the estimated σ tends to the input one subtracted by a bias due to the iterative formulation showed in Equation 5.11.

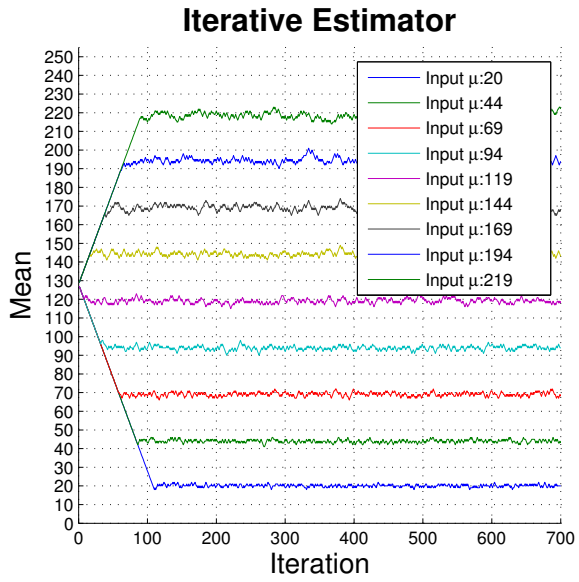

Figure 5.8: Example of convergence of the Mean Iterative Estimator (μ)

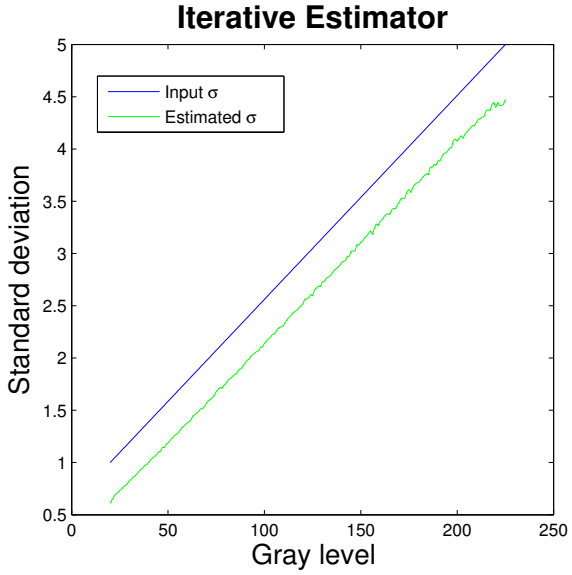

Figure 5.9: Example of convergence of the Standard deviation Iterative Estimator (σ)

Moreover, LBB has been evaluated against the GMG and MoGv2 algorithms implemented in the BGS Library [190]. The test has been conducted on the same dataset presented in the previous Section, that contains five videos that represent a football match. AR- scenes are focused on the penalty area and the size of each frame is 1600×736, while a larger area of 1920×1080 pixels is captured in the FG- ones. The five scenes contain some typical situations of a soccer match, for example: a cluttered scene with illumination changes (AR1); the shoot of a penalty kick that implies almost all players around the penalty area (AR2); the shoot of a free kick (AR3) and two actions that are filmed from a wider point of view (FG1 and FG2). In FG2 some players are warming up, so the scene is more dynamic than the one in FG1. Each BG model is evaluated after 20, 40, 60 and 80 seconds after the starting frame f_0.

Figure 5.10 contains the qualitative analysis of some frames in terms of ground truth and foreground masks. Rows 1 and 3 contain a cluttered scene where a high number of players is moving in in the penalty area. Here, the MoGv2 (Figure 5.10 (d)) shows a weak output due to the constant update of model parameters that is including the players in the BG, while the other approaches (Figures 5.10 (c) - (e)) produce a more stable output. The frames in the second row show a scene where the advertising is changing. Here, LBB is updating the BG mask while the other algorithms already did it, due to the updating speed implemented by the unary increment described in Section 5.2.1. This corresponds to the LBB outlier in Figure 5.11 with low precision and high recall, that gradually tends to a stable configuration when the BG mask is updated. The quantitative results have been extracted calculating the F-Measure, Precision and Recall (see Equations 5.9, 5.7 and 5.8) on four different frames per each sequence, representing the considered time interval. Each ground truth frame has been manually obtained starting from the raw video. Figures 5.11 and 5.12 summarize the metrics calculated for each video sequence. The comparison of LBB Precision and Recall values against the best value among GMG and MoGv2 shows that the average LBB R value is 39% better than the others, while the P value is generally comparable. This result is shown in Figure 5.11 where each point in the P-R plane is referred to a run of a specific algorithm (red for MoGv2, blue for GMG and green for LBB). According to this representation, the ground truth has coordinates $(1, 1)$, therefore points in the upper right part of the figure correspond to the best results. High R values for LBB demonstrate that the approach is robust to false negative outputs. Figure 5.12 shows the 3D bar chart representation

of the F-measure. The AR sequences represent a complex situation with a high number of moving people in foreground and here the F-Measure of LBB is higher than the other methods used for the comparison. In the FG ones LBB and GMG behave in a similar way and show comparable results in terms of F-Measure. In particular, the FG1 sequence starts with an advertising change and here LBB has a low F-Measure in the first frame (FG1 - 1) because the foreground mask is noisy, but then the F-Measure increases, so the model is correctly updated in the subsequent frames. The overall average of the LBB F-Measure is 18% better than GMG and MoGv2, thus confirming that the proposed method is capable of modeling such scenarios.

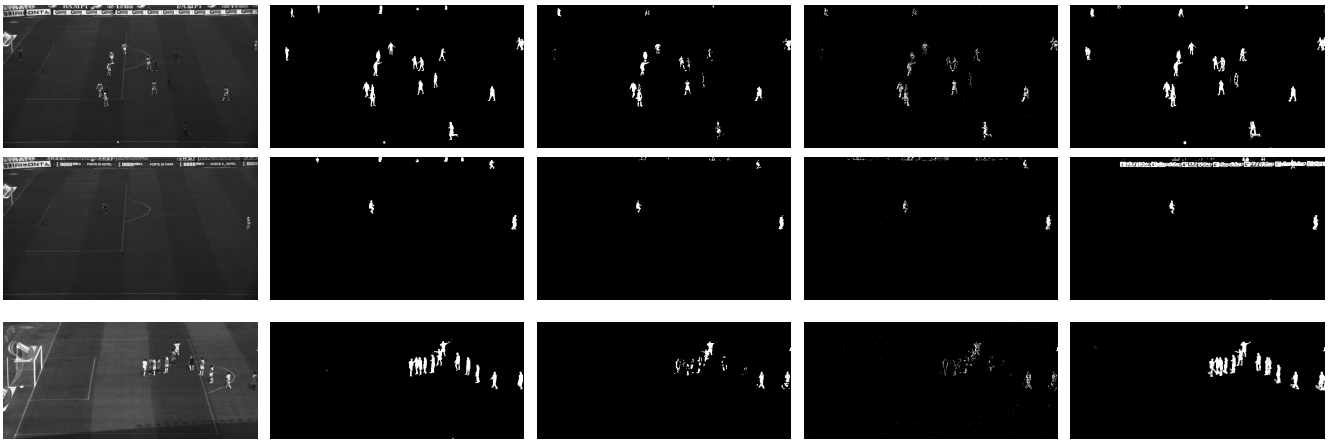

Figure 5.10: Qualitative results for some frames. The columns contain, respectively, the original frame (a), the ground truth (b) and the foreground masks obtained with GMG (c), MoGv2 (d) and LBB (e).

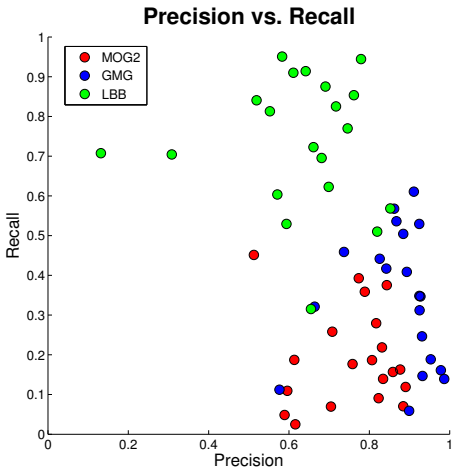

Figure 5.11: Quantitative results on the dataset in terms of Precision and Recall.

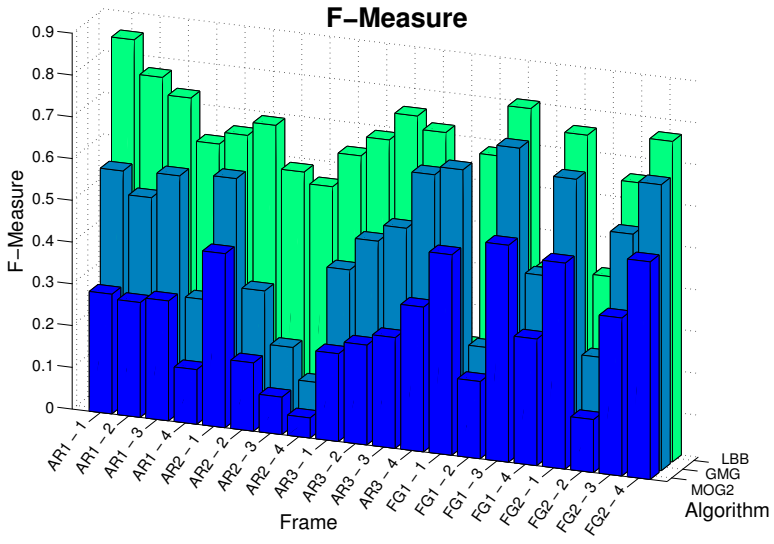

Figure 5.12: Quantitative results on the dataset in terms of F-Measure

5.3 GIVEBACK

5.3.1 Algorithm Description

The proposed algorithm can be divided in three main building blocks, as shown in Listing 5.2: initialization, processing and update. The first step is executed only once and initializes the BG image setting each pixel to half intensity. Therefore, the produced image is gray and reflects the absence of any *a priori* knowledge about the scene. The processing phase is composed of: variance, one step frame differencing, fine tuning and energy. The ones marked with an asterisk in Listing 5.2 are referred to PIIB and described in Section 5.1 and variance process is related to the formulation shown in Equation 5.11. The other modifications are detailed singularly in the following Sub sections. Also here the update phase is assisted by a binary update mask M_{upd} that increases or decreases by $\kappa = 1$ a background pixel value if its corresponding M_{upd} value is set to true. Finally, the fine tuning phase exploits the output of two blob analyses — one on the foreground mask and the other on the one step frame differencing one — with the aim of giving robustness to the BG model, as it will be described later.

Listing 5.2: Algorithm pseudocode

```
Background Initialization*
for each frame
   Variance process
   One step frame differencing
   if(Background is learned)
      Foreground extraction
      Fine tuning process
   Background Update*
   Energy Process*
```

5.3.2 One step frame differencing

This task is executed at each iteration and produces a binary mask obtained by thresholding the absolute difference of the last captured frame and the one being processed. First, the absolute difference image is calculated with the formula:

$$AD = |I_t - I_{t-1}| \tag{5.14}$$

Then, for each pixel (u, v) the binary mask M_{os} is calculated in the following way:

$$M_{os} = \begin{cases} 0 & \text{if } AD(u,v) \leq \tau\left(I_{t-1}(u,v)\right) \\ 255 & \text{if } AD(u,v) > \tau\left(I_{t-1}(u,v)\right) \end{cases} \tag{5.15}$$

Each pixel is considered as a normal random variable: the mean value is represented by its corresponding value in the last captured frame, while the variance depends on its gray level, since different intensity values might have different variances. The threshold $\tau(\cdot)$ used to classify each pixel as background or foreground is a function of a specific gray value and in our implementation it is set to $\tau(\gamma) = 3.5\sigma_\gamma$, where $\sigma_\gamma = \sqrt{V(\gamma)}$. Hence, each black pixel lies in an interval $[\gamma - 3.5\sigma_\gamma, \mu + 3.5\sigma_\gamma]$ while the white ones represent the tails of the corresponding normal distribution. The binary mask obtained at this stage is useful to achieve robustness during the subsequent phases, for example avoiding the BG model update in correspondence of a moving player.

5.3.3 Foreground extraction

The foreground extraction phase is similar to the one step frame differencing one, except from the fact that the background image is exploited instead of the last captured frame. The output of this module is a binary mask M_{fg} obtained with the same thresholding process presented in Eq. 5.15, in which the absolute difference image is $AD = |I_t - BG_{t-1}|$. M_{fg} is the mask used to compare the model with other approaches in the next section.

5.3.4 Fine Tuning Process

After the BG model has been learned by the system, the fine tuning process module is switched on to calculate the binary update mask M_{upd}. This task is achieved by means of a blob analysis done on both M_{os} and M_{fg} to obtain two sets of connected regions — namely $B_{os} = \{b_1, b_2, \ldots, b_n\}$ and $B_{fg} = \{b_1, b_2, \ldots, b_m\}$ — that are processed according to the following rule:

$$M_{upd} = \{\ulcorner(b_i, b_j)\urcorner | b_i \in B_{os}, b_j \in B_{fg}, b_i \cap b_j \neq \emptyset\} \tag{5.16}$$

where $\ulcorner(b_i, b_j)\urcorner$ is the minimum circumscribed rectangle that embeds both b_i and b_j. Each region extracted from the foreground mask is compared to each region extracted by the one step frame differencing process in order to

find overlapping blobs that do not produce an empty set when intersected. As a consequence, the update mask keeps trace of robust foreground areas in which the BG update does not take place, allowing the algorithm to easily filter ghosts or static subjects that stand still on the scene.

5.3.5 Experiments and Results

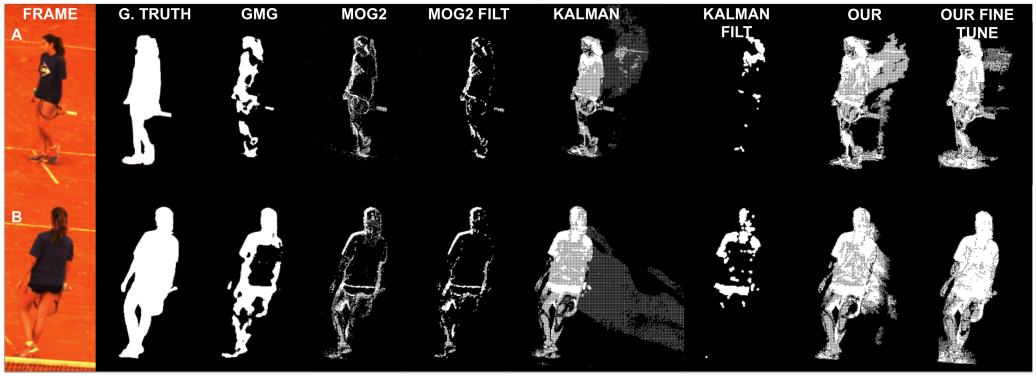

Figure 5.13: Example of silhouette extracted with all the algorithms tested in the experiment. Here, the proposed approach is the one that better preserves the entire silhouette of the player finding a trade off between computational load and reliable results. The amount of false positive or negative pixels in the proposed approach is reduced when compared to the other statistical methods considered.

Two variants of the methodology described in the previous section have been tested and compared with other statistical based background models available in the BGS library [190] (GMG and MOGv2) and the adaptive background estimator based on kalman filtering [193] implemented in MVTec Halcon suite[194]. The first variant of the proposed algorithm models the background skipping the fine tuning process, while the complete method — with the fine tuning process in place — is tested separately as well.

Both qualitative and quantitative tests have been done on recorded sequences that represent a tennis training session. Four raw videos have been taken with AVT Prosilica GT1920C cameras capable of acquiring 1936×1456 frames at $40Hz$ and configured to capture 1920×1024 frames at $50Hz$ in order to avoid flickering issues exploiting the hardware setup. Moreover, cameras are equipped with auto iris lenses which enable to ensure a constant

brightness level in the whole recordings. As a consequence, results obtained on a single camera are reproducible on the other ones when recording the same event from different points of view. Starting from a reference frame f_0, ten images sampled every 500 frames have been manually annotated and quantitatively analysed exploiting the corresponding ground truth masks. Only moving players and balls have been segmented on the ground truth image, while inactive balls (always present in tennis courts, especially during training sessions) have not been annotated as foreground objects.

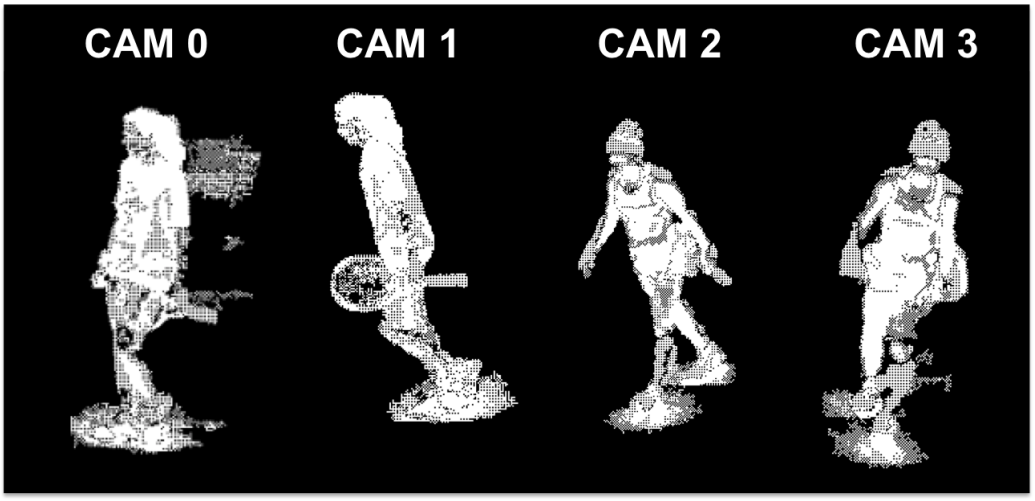

Figure 5.14: Example of player silhouette extracted from four syncronized views. CAM0 and CAM1 refer to Player 1, while CAM2 and CAM3 to Player 2. The performance of the proposed approach is the same independently from the specific point of view, thus confirming that the reproducibility of the results.

Qualitative results in terms of player silhouette segmentation can be inferred from the visual inspection of the foreground objects resulting from different algorithms, as reported in Figure 5.13. Here, GMG algorithm handles effortlessly shadows near the players feet and shows a tendency to consider background some parts of the legs, performing poorly on the lower parts of the player body because of color similarity between the court and the skin of the player. Kalman fitering based backgound estimator is sensitive to ghosting issues that appear when the player moves after having stationed elsewhere. The proposed approach is able to produce a well-cut

player silhouette, especially in the fine tuned variant where the ghost is being reduced while preserving the whole shape of the player.

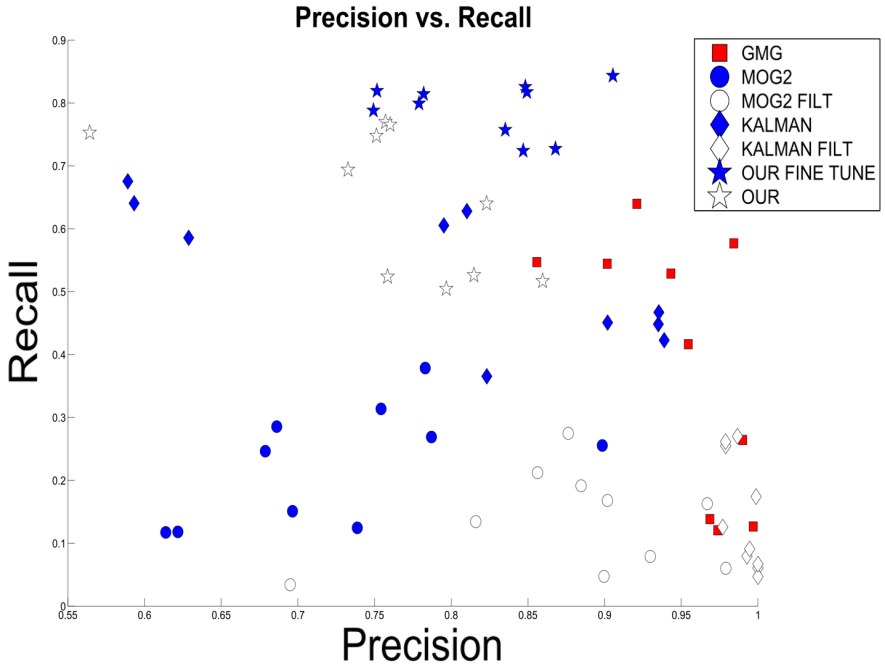

Figure 5.15: Quantitative results on the dataset in terms of Precision and Recall. Each point corresponds to a comparison between a foreground mask obtained with a specific algorithm and the corresponding ground truth. The upper right corner represents a FG mask that that is exactly the same as the ground truth (both P and R values equal 100%). Points that tend to $(1, 1)$ are the best among the considered ones.

Figure 5.15 summarizes the algorithms performance in terms of Precision P and Recall R for each annotated frame. Here, each point in the $P-R$ plane refers to a run of a specific background subtraction method where different algorithms are shown with different marker shapes and colors, while variants are presented as color-filled or white-filled. According to this representation the ground truth has coordinates $(1, 1)$, therefore points that lie in the upper right part of the figure correspond to the best results.

In the comparison, both MOGv2 and the Kalman filter based background FG masks have been post processed with a morphological opening operation employing a circular structuring element of 2 pixels radius. The GMG

algorithm did not require any additional filtering operation since the method already produces salt-and-pepper noise filtered foreground masks.

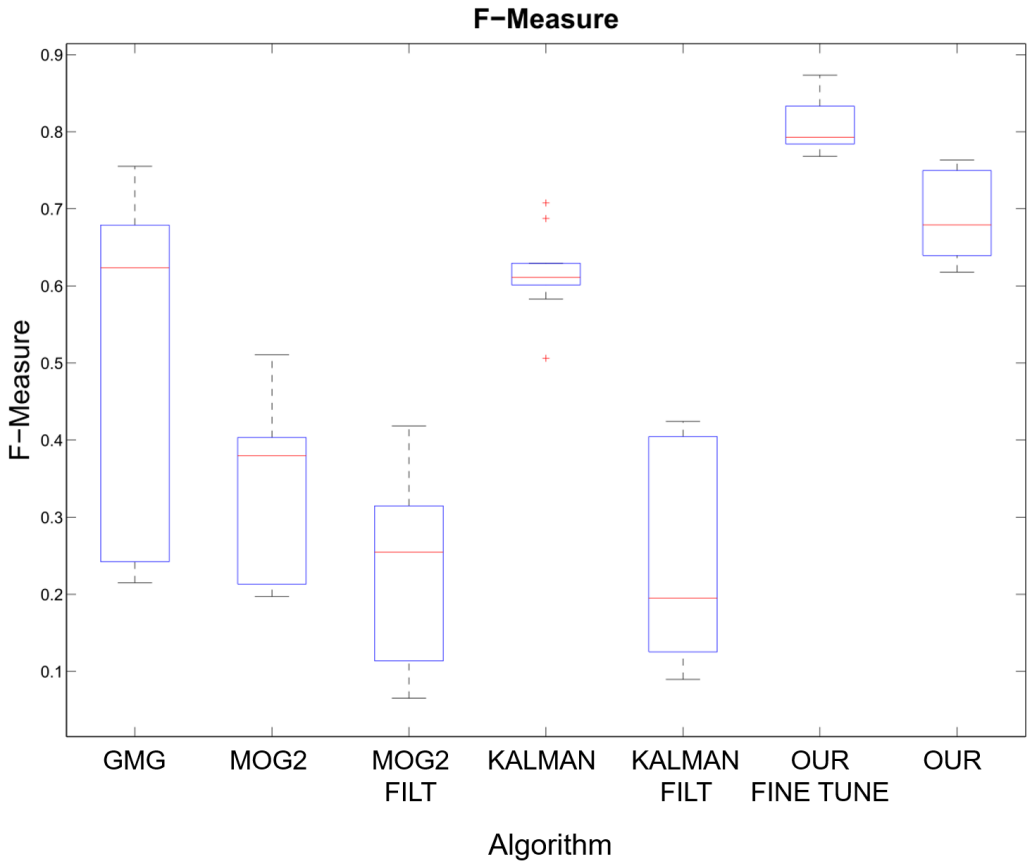

Figure 5.16: Quantitative results in terms of F-Measure organized in a boxplot. The figure summarizes the overall performance of each algorithm. The red marks represent the median value among the executions, while blue boxes go from the 25^{th} to the 75^{th} percentile. Small boxes are better because of low variance and high repeatability of the experiments.

Figure 5.15 shows that both the variants proposed in this paper have noticeable performance. The variant related to the fine tuned algorithm shows the best overall results, with average higher scores on both axes (better precision and recall performances at the same time). MOGv2 is particularly

sensitive to salt-and-pepper noise and has a global tendency to show low recall values. This implies a high number of false negatives pixels in the FG masks, as it can be seen analysing the silhouettes in Figure 5.13. The Kalman filtering based background results (highlighted by blue diamonds) in terms of precision are not constant during the acquisition. This means that the approach is affected by the production of false positive pixels in the form of player ghosts, as shown in Figure 5.13. Finally, GMG algorithm shows a precision comparable with the reference performance of the Kalman based one, trading some precision for better recall scores.

In some respects, the GMG algorithm and the "complete" variant of the algorithm described in this paper perform similarly well, with the GMG algorithm being better in the precision score and the ones proposed here showing better recall. However, as will be shown shortly, the adaptive BG model presented here seems more dependable, with a uniform behavior while working on different frames, while scores obtained by the GMG algorithm are more scattered.

Figure 5.16 shows a boxplot of the F-Measure calculated during the experiment. There are seven boxes, one for each algorithm. Inside each box, the median value is highlighted with a red line, while the edges of the box are the 25^{th} and 75^{th} percentiles. The whiskers extend to the most extreme data points not considered outliers, and outliers are marked individually with a red cross. Hence, small boxes refer to algorithms whose results are repeatable over time, while big boxes show that the range of F is wide (reflecting high variance in the results). The only algorithm that produces outliers is the Kalman filter-based one, due to not constant precision values among the executions as highlighted beforehand. However, it is the one with the smaller box.

In summary, the best algorithm among the ones tested is the proposed method enriched by the fine tuning module, as its median F value is 80%, the associated box is the second smallest and there are no outliers in the statistic.

5.4 Real time 3D tracking

5.4.1 Methodology

The tennis ball detection and tracking algorithm presented in this Section makes use of three dimensional data and domain knowledge to effectively identify and understand ball positions during tennis training sessions and matches. The input of our algorithm is a 3D point cloud enriched by temporal information about ball samples, therefore each point can be seen as a quadruple $P_i = (f_i, x_i, y_i, z_i)$, where f_i is the frame index of the specific ball candidate of coordinates (x_i, y_i, z_i). An example is shown in Figure 5.17, where each blue point represents an observation of a ball candidate coming from a stereoscopic system.

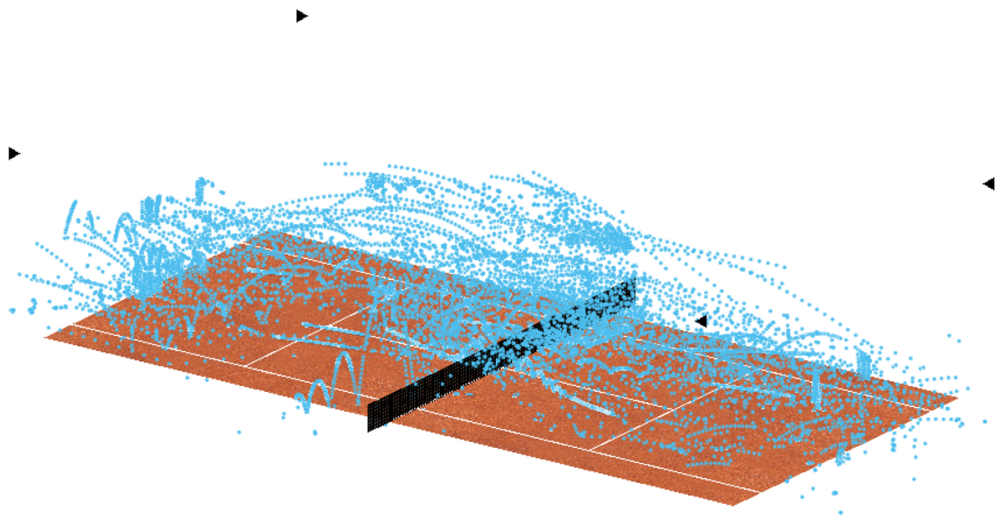

Figure 5.17: Example of input point cloud for the proposed algorithm. The black triangles represent the position of the four cameras used to extract 3D information. Blue points are the ball candidates collected over time.

The proposed method can be summarized in five steps, as reported in Figure 5.18:

1. Ground points removal;

2. Tracklets initialization;

3. Compatible ground points recall;

4. Sub-tracklets identification and polynomial model definition;

5. Final trajectory assembly.

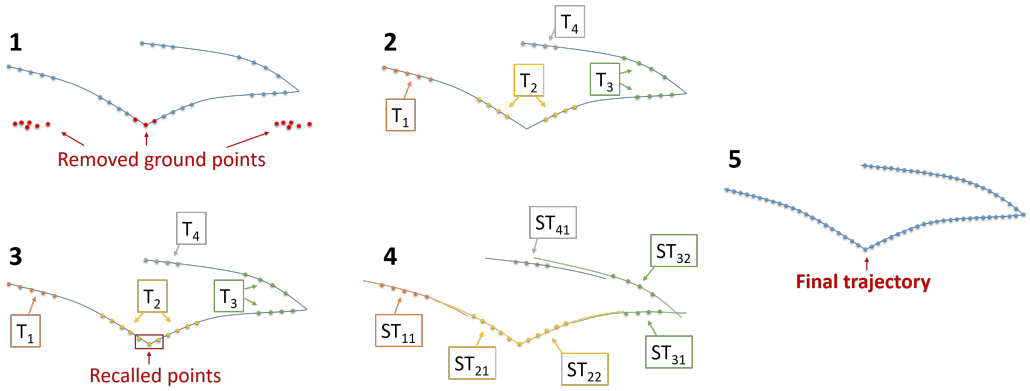

Figure 5.18: Graphical explanation of the proposed tracking method. In all the subfigures the straight line represents the ground truth trajectory while points represent input data for the algorithm. First, all 3D points near to the ground plane are removed from the set of input values. This way, noise due to the misclassification of players' feet or to swipes on clay court is negligible. Then, a nearest neighbor approach is responsible for choosing candidate tracklets that are labelled as T_1, T_2, T_3, T_4. Once tracklets are initialized, ground points compatible with them are recalled in the respective sets of points as reported in the third step. Tracklets can be now split in sub-tracklets to deal with bounces on the ground and changes of direction: both T_2 and T_3 are effectively divided in two distinct subsets of points. For each sub-tracklet the coefficients of a polynomial model are used to project it forward and backward. The final trajectory can be then assembled choosing the appropriate points from each interpolated sub-tracklet.

The first step basically consists of a temporarily removal of points near to the ground plane (i.e. $z_i < \tau_{ground}$) and is a necessary pre processing step in our algorithm since 3D information is retrieved from a stereoscopic imaging system and is affected by noise. Image processing by background subtraction and foreground analysis can generate false positive candidates when players' feet are misclassified, when the player swipes to make a strike leaving a sign on the clay court or simply when slow balls are left on the ground. False positive ball candidates are then included in the 3D point cloud until the background model is able to update itself, even if this can lead to the formation of small

red near ground clusters as the ones in Figure 5.18.1. In our implementation τ_{ground} is set to 0.2 meters.

The resulting point cloud is then scanned in order to initialize pieces of valid trajectories, called tracklets. All the points observed at the initial frame f_0 are associated to different tracklets. Then, a frame counter increased at each iteration filters all the candidates observed at a frame index to perform temporal and spatial distance filtering. Given two candidates P_i and P_j, their temporal distance is defined as $\phi_{ji} = |f_j - f_i|$, while their distance in space is the Euclidean one $\delta_{ji} = [(x_j - x_i)^2 + (y_j - y_i)^2 + (z_j - z_i)^2]^{1/2}$. Assuming that P_i is the last valid point of a tracklet, P_j is linked to P_i if

$$\begin{aligned} \phi_{ji} &< \tau_{time} \\ \delta_{ji} &< \tau_{space} \cdot (2 - 2^{1-\phi_{ji}}) \end{aligned} \tag{5.17}$$

In our implementation $\tau_{time} = 10$ frames and $\tau_{space} = 0.7$ meters. Values have been experimentally chosen as they depend on both hardware setup (cameras temporal resolution) and players' skills (ball speed during the gameplay). At the end of this stage, 3D balls that are close both in time and space are labelled with the same tracklet number as shown in Figure 5.18.2.

Each identified tracklet is then scanned again to detect any hole that is of up to two frames and near to the ground plane. Ball candidates filtered in the first step that are compatible with tracklet segments (with respect to their x, y, z position) and temporarily coherent are then joined inside the tracklet as depicted in Figure 5.18.3. Thanks to this two-steps check, ball candidates near to the ground – excluded at the beginning of the algorithm – are effectively reinserted in the most probable tracklet without introducing noise as described beforehand.

Now, tracklets need to be divided in sub-tracklets before evaluating polynomial models for reducing the computational complexity of the whole procedure. This step leads to the reconstruction of information when they are not available and is motivated by the absence of ball candidates in a certain number of frames. There are situations that for several reasons hinder the detection of a ball in the 3D space and produce a partially filled point cloud, for example due to the ball being visible only from one camera of the pair, or due to a false negative affecting the ball detection phase. Three curves are therefore modelled for each sub-tracklet, one for each axis. A polynomial degree is chosen separately for each one. Polynomial models in the vertical direction are always chosen to be parabolic, while models for the remaining

directions are chosen relatively to the number of points of the considered segment: segments with more points are considered more reliable and hence a parabolic description is computed. On the other hand if there are just a few points, linear interpolation is preferred. To do this, we have chosen the center of our reference system exactly in the middle of the court so that the X-Y plane matches the ground plane ($z = 0$) and z coordinate represents the ball height. Under this assumption, each sub-tracklet ST_{ij} with at least three points can be processed in order to find its kinetic model with respect to the time (frame index f). It is immediate to notice that this step is concerned with the identification of domain specific events affecting the ball: collision with the ground (local minima around zero on the Z axis), collision with the net (intersection on the X-Z plane at a specific height) or collision with a player's racquet (changes of the sign of speed on the Y axis). All these cases affect ball trajectory and ultimately are events that start a new course for the ball. Looking at the example in Figure 5.18.4, T_2 and T_4 are divided in two sub-trackets, respectively ST_{21} and ST_{22} (ball bounce), and ST_{31} and ST_{32} (stroke). Model evaluation is then followed by an interpolation step. Each sub-tracklet is extended in both directions until either the extended curve collides with the ground or a maximum number of interpolated frames is met.

Finally, the extended sub-tracklets are checked for both contiguity and validity in the 3D space and are assembled in the final trajectory described in Figure 5.18.5.

5.4.2 Experiments and results

The tracking approach described in this paper has been tested on real data coming from a multi camera 3D stereoscopic system. A private club has been instrumented with four high resolution cameras that have been used to capture raw data from tennis matches played on a clay court. 3D information of the ball provided by image processing techniques are the input for the proposed tracker. The software has been coded in Matlab 2014 and runs on a Intel i7 CPU @ 2.7GHz, 16 GB RAM in 7.44 seconds for the proposed experiment. The reference dataset comprises about 38000 raw frames (roughly 280 gigabytes of data) that represent a friendly match. Due to the dataset's length in terms of throughput, both qualitative and quantitative experiments have been conducted.

The first took place on the whole sequence by evaluating key events such as strokes and bounces, in fact it is possible to indirectly evaluate the capabilities

of the approach by examining the frames in which those events are recognized. A total amount of 106 strokes and 96 bounces is correctly reported within 10 frames, meaning that the event is recognized in ± 0.2 seconds (because the frame rate is 50 Hz).

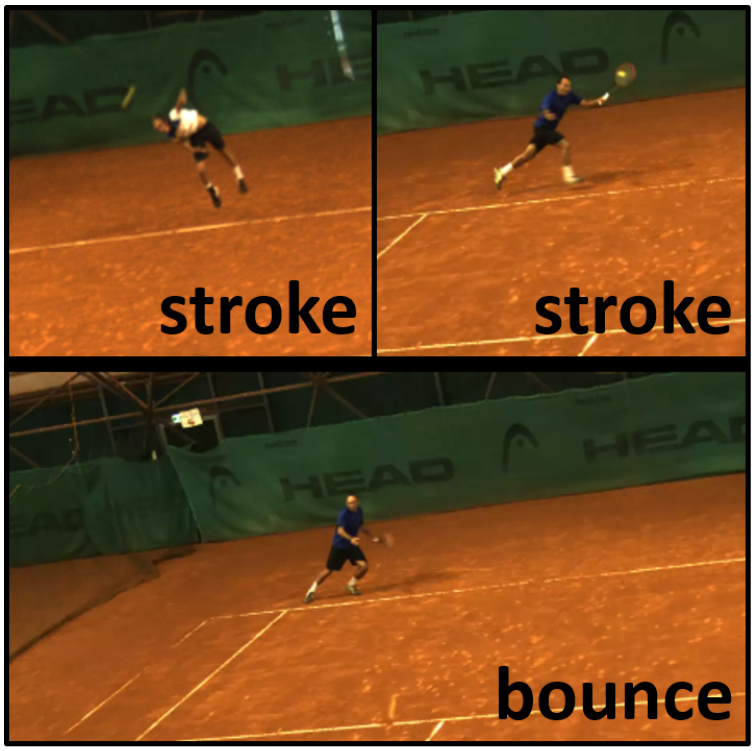

Figure 5.19: Qualitative evaluation of trajectory reconstruction. Since ball trajectory in a tennis game is mostly determined from interactions with other objects, like the tennis court ground, the net or the player's raquets, a qualitative evaluation of trajectory reconstruction can be achieved by determining key frames in which the ball changes its course and verifying if they correspond to meaningful events or happen elsewhere. Three key frames that are representative of the accuracy of an event detection strategy based only on tracklets evaluation, are shown in the figure. Key frames are recognized within ± 0.2 seconds from the real events.

On the contrary, for the quantitative analysis 893 3D points have been manually annotated as ground truth that has been acquired by highlighting the center of mass of the ball in the raw images and then applying the same projective transformations built in the stereoscopic system to extract

3D information. Then, each processed point $\hat{P}_i = (f_i, \hat{x}_i, \hat{y}_i, \hat{z}_i)$ is compared with its homologous ground truth point $P_i = (f_i, x_i, y_i, z_i)$ to compute the residual error $E_i = (x_i - \hat{x}_i, y_i - \hat{y}_i, z_i - \hat{z}_i)$. Finally, the figure of merit $\varepsilon_i = \frac{1}{3} \cdot [(x_i - \hat{x}_i) + (y_i - \hat{y}_i) + (z_i - \hat{z}_i)]$ is computed and analyzed in the following figures and table in order to quantify the accuracy of the proposed tracking algorithm.

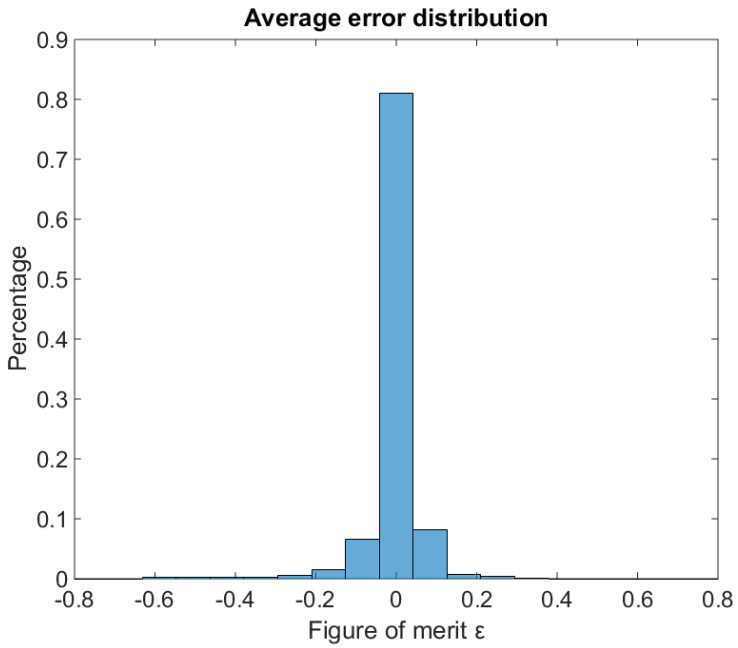

Figure 5.20: Average error distribution histogram with respect to the figure of merit ε. An intrinsic uncertainty is related to the employed cost-effective stereoscopic system setup. The use of low degree polynomial models due to real time constraints can lead to some time misalignments as well.

Figure 5.20 shows the average error distribution histogram in which bars represent the ε percentage evaluated for each error displacement value. The ideal result for such a plot should be a 100% histogram exactly located in $\varepsilon = 0$, but in our real case experiment a distribution of roughly 80% samples is located around zero and an exponential decrease in the range $[-0.2, 0.2]$ is observed. This behavior is motivated by two important phenomena: first, an intrinsic uncertainty is introduced by the stereoscopic system used for evaluating 3D data; second, real time constraints induced the choice of low

degree polynomial models that can lead to misalignments when processing high speed balls (typically serves) with few observed points, as it will be shown in Figure 5.22.

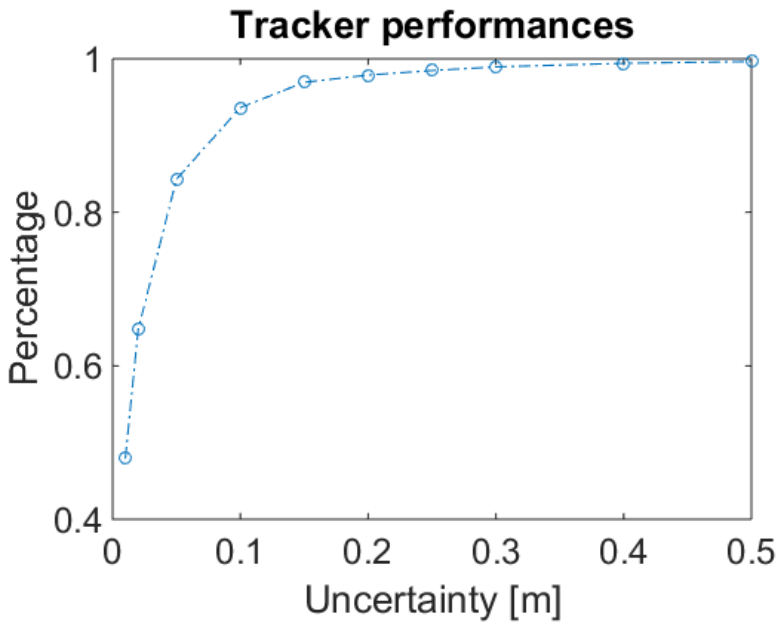

Figure 5.21: Tracker performance. A plot of the tracker performance is shown here. This shows that almost 94% of the tracklets points are accurately reported with a maximum distance 0.1m from the real ball position, as labeled in the ground truth.

Data are also presented in the tracker performances curve shown in Figure 5.21, where the results' uncertainty can be immediately observed by looking at the plot. In this case, the uncertainty is defined as the absolute value of the figure of merit ε so that for each 3D point a portion of the 3D space – i.e. a cube – is delimited as the region in which the predicted/observed point should lie. Low uncertainty values for a high number of points reflect a high tracker accuracy during the experiments, but since the samples are affected by noise introduced by cameras and projective transformations, we can reasonably assume that 0.1 meters of uncertainty is an acceptable threshold to evaluate performances. For this reason, the proposed tracking system achieves 93.6% accuracy on real data.

Precise quantitative details about the two plots described beforehand are reported in Table 5.3, where percentages are sampled started from $|\varepsilon| < 0.01$

Table 5.3: Details of the average error distribution histogram. Tracker performance accuracy is presented here in a tabular form.

ε_{min}	ε_{max}	%
-0.01	0.01	48.0
-0.02	0.02	64.7
-0.05	0.05	84.3
-0.10	0.10	93.6
-0.15	0.15	96.9
-0.20	0.20	97.9
-0.25	0.25	98.5
-0.30	0.30	98.9
-0.40	0.40	99.4
-0.50	0.50	99.6

meters to $|\varepsilon| < 0.5$ meters. The first thing to notice is that half of the points are almost coincident with their ground truth as the uncertainty cube defined by $|\varepsilon| < 0.01$ contains 48% of the samples. Then, the curve increases logarithmically until the figure of merit reaches 0.1 meters. In fact, increasing $|\varepsilon| < 0.1$ to the upper bound of 0.5 meters enhances the performances of only 6%, from 93.6% to 99.6%.

Finally, qualitative results are provided in Figures 5.22 and 5.23, where two distinct actions are plotted directly on a virtual tennis court. The first action is also split in four parts (a), (b), (c), (d) for a better presentation, while in the second case the whole action is plotted. Black triangles indicate the position of cameras that are located behind the side line at approximately 6 meters in height, while dots are used to mark ball positions. Yellow balls are the ground truth and red circles represent the output of the tracker. These results confirm what has been observed in the previous plots because red circles are close to their ground truth corresponding points. Moreover, it is worth focusing on the effects of numerical approximations that are highlighted by three green rectangles:

1. in the first case the reconstructed trajectory follows a line while the ground truth is represented by a slightly pronounced curve. This is due to the fact that only a few high speed points have been correctly identified on the net, so a first order model has been chosen to reconstruct

missing data;

2. in the second case the reconstructed trajectory is over estimated, in fact the ground truth shows that the ball has followed a shorter path with respect to the output of our algorithm. Two different factors cause this behavior: the side line is the furthest zone in the field of view of the stereo pair – approximately 24 meters far from the camera (therefore losing some accuracy) – and the ball can be occluded by the player in correspondence of strokes losing a certain number of input points for the tracker;

3. the third case is similar to the second one, but here the computed trajectory is under estimated. The causes of this behavior can be considered the same of the previous case, because the main problem is that ball candidates are lost when the ball intersects the player's blob. Whenever there is no ball correspondence between image pairs, no 3D point can be extracted.

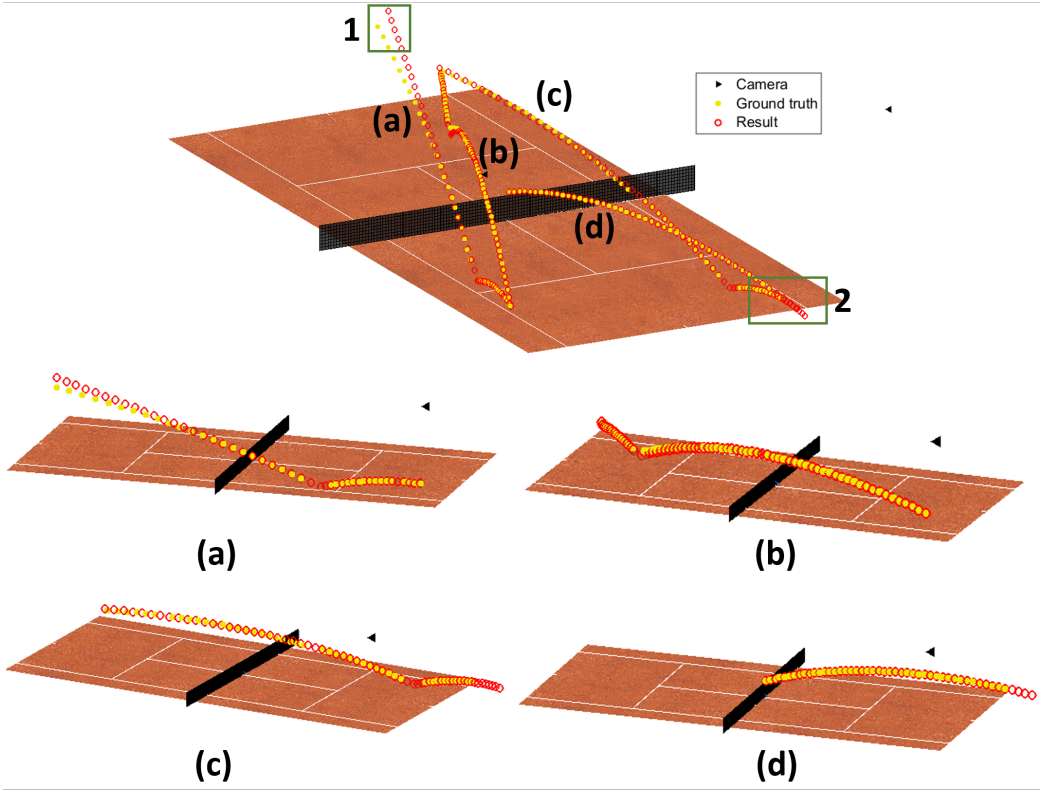

Figure 5.22: This figure shows an example action that has been reconstructed with the proposed approach. Ball candidates (red circles) are plotted against their respective ground truth (yellow dots). The first image represents the whole sequence that starts with a serve and ends with a rewarded point. Green rectangles are used to highlight the effects of numerical approximations that have been commented in the paper. Four sub plots are also reported for clarity purposes.

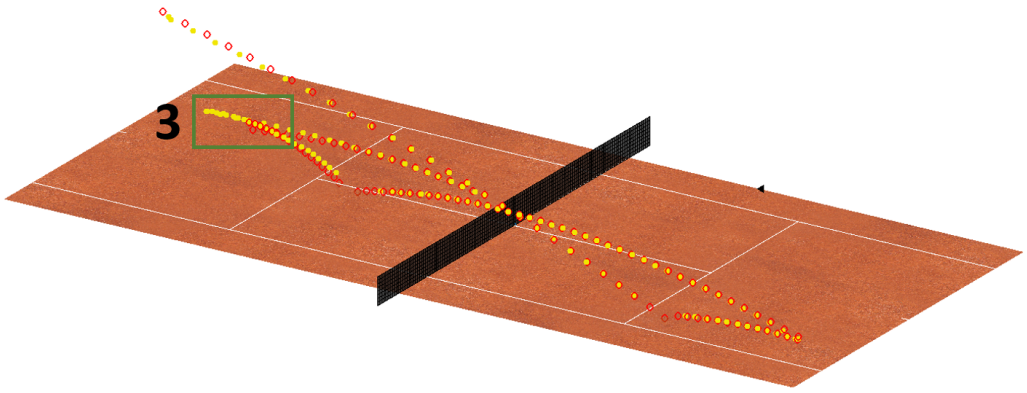

Figure 5.23: Another example of reconstructed action.

Even if the lack of some points could produce unpredictable results, experimental results shown in Figures 5.22 and 5.23 confirm the correct path following capability of the proposed approach. In fact, our algorithm accurately follows the ground truth as soon as newly observed points are available, as trajectories reconstructed in blind spots are effectively close to their correspondent ground truth value.

5.5 Summary

In this chapter multiple algorithms for real time processing of high throughput data have been detailed. In particular, three background models and a 3D tracking method have been presented.

The first one, namely PIIB, is an adaptive background model for high frame rate video applications suitable for smart cameras embedding. The algorithm is designed with respect to the SSE2 instruction set programming rules and works on grayscale images for speed reasons, but the logic can be extended also to RGB images. The computational complexity of PIIB is linear and it shows a really fast responsiveness that enable its implementation on smart cameras. Moreover, it obtains a good overall performance indicator in athletic video processing due to the high recall value shown in the experiments, even if its precision is generally lesser than the other analyzed methods. Precision can be improved adding some modules to the algorithm, such as a *selective background update* or a *shadow removal* module, in order to enhance the

overall performances, as it has been done in the other approaches.

The second one, namely LBB, is a likelihood-based background model for real time processing of CFA images. The algorithm is designed with respect to the state of the art output format for vision cameras and implements a statistical model that takes into account the BG as the mean image while modeling the variance of each gray level processing its occurrences in the whole frames. For this reason, the variance is not calculated with respect to the observations of a single pixel over time, but is related to the intrinsic nature of the sensor. The results obtained in terms of F-Measure overcome the ones obtained with PIIB and confirm the robustness of the proposed approach in the athletic video processing context.

The third one, namely GIVEBACK, extends the results obtained with LBB to the specific context of tennis and represents an efficient method to segment active entities — players and balls — in this context. The proposed approach is based on simple but effective operations (from a computational load point of view) that allow its employment on real time systems. Moreover, it operates directly on raw videos thus encouraging its implementation directly on smart cameras. Experiments on tennis training video sequences demonstrate its effectiveness in tennis players silhouettes processing, even if there usually is a strong similarity between players skin and the tennis court. The fine tuned version of the algorithm shows good scores in terms of Precision and Recall and F-Measure. Its performance on different frames are very similar on each ground truth annotated test image. These results confirm the robustness of the proposed method when compared to other statistical approaches evaluated in the benchmark.

Finally, an effective method to detect and track a tennis ball in a 3D space has been presented. This approach exploits domain knowledge to recognize ball positions and trajectories from a sparse but cluttered point cloud that evolves over time. The tracker has been tested on real data collected by a stereoscopic system during a friendly match. The results demonstrate that 93.6% accuracy can be achieved within 0.1 meters of uncertainty in the 3D Euclidean space. This is compatible with the specifications of the stereo system, that has been developed using the minimum number of cameras to obtain an acceptable resolution on the side lines of the court. The majority of errors are due to missing 3D data, as shown in the experiments, because if the ball is not recognized correctly in one of the cameras, triangulation can not take place.

Chapter 6

A technology platform for automatic high-level tennis game analysis

6.1 The proposed system

The proposed system consists of a dedicated hardware setup (cameras and computer) and a number of software modules for the automatic processing of the recorded video sequences. The aim is to record tennis video sequences and performs the segmentation and the analysis of significant tennis actions in order to support coaches in the evaluation of tennis players performance during training sessions or official matches.

We propose the use of dedicated cameras in order to collect data that cover all the court and are able to observe simultaneously the positions of players and ball during actions. Broadcast cameras (which commonly show a single point of view of the match) are not suitable for this kind of tasks first because 3D reconstruction of the ball trajectory is necessary to evaluate events, and also because positions of the two teams and the ball in the court are necessary to evaluate tactics and performance. Moreover, broadcast camera videos are often chosen for entertainment purposes then they are not suitable to record all the events necessary for automatic game and players evaluation.

Here, four synchronized cameras are placed on the corners of the court and connected to a central node, provided with suitable algorithms for processing the acquired videos. The system reconstructs the 3D ball trajectory and

recognizes key events by the concatenation of simple action parts which concern ball rebounds, shots or faults. The main contribution of this work is, from one side, the system architecture, in terms of number and positions of dedicated cameras, frame rate and resolutions which respect the constraints imposed by the tennis domain. On the other side a number of processing modules has been implemented to perform low level image processing and high level semantic interpretation and to recognize key events automatically assigning a score. Finally, large attention has been put on designing a technology platform that can effectively support coaches with relatively low cost equipments. The results demonstrate that the proposed system is able to effectively reconstruct 3D ball trajectories, recognize serves, strokes and bounces and make a decision about score assignment for each action. Moreover, coaches can perform strategic queries to analyze players intentions, behaviors and performance using a combination of both 3D data and key events annotations.

6.2 System overview

The proposed system is primarily designed to address coaching needs. For this reason, it is designed to operate in structured environments, typically indoor, although it can operate outdoor as well. The system can be schematically decomposed in three sub-systems (see Figure 6.1): data acquisition, data processing and data storage.

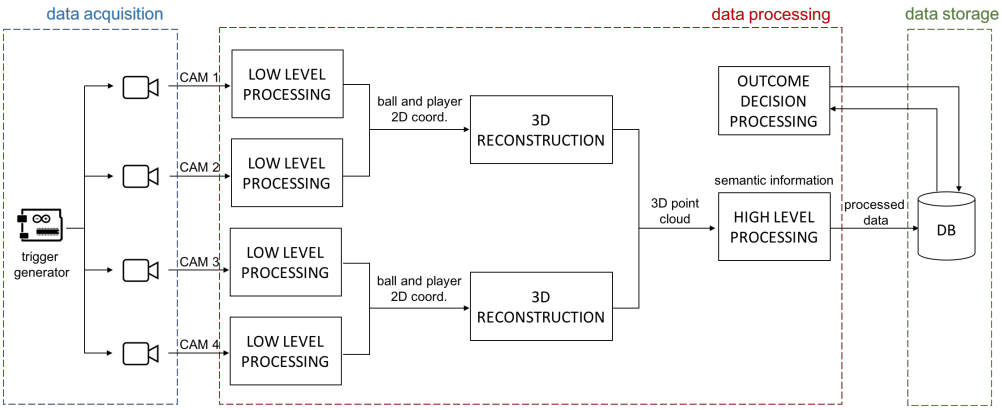

Figure 6.1: Block diagram of the system.

6.2.1 Data acquisition

In order to design a platform which can operate in every condition, both indoor and outdoor, the most general conditions have been considered to choose the proper equipments and functionalities. Most of the tennis training sessions are performed in indoor environments, generally under shelter structures. Shelter structures create particularly challenging conditions for image processing, since they are made to let sunlight in, therefore with varying illumination from sunny to cloudy as for outdoor condition, and yet allow to switch on artificial lights when sunlight proves insufficient. In addition, artificial lights introduce flickering effects which can greatly modify the quality of the acquired images. For this reason the platform has been designed to operate in the most challenging situation, i.e. indoor environments, but the same equipments can be used or either simplified to operate on outdoor courts.

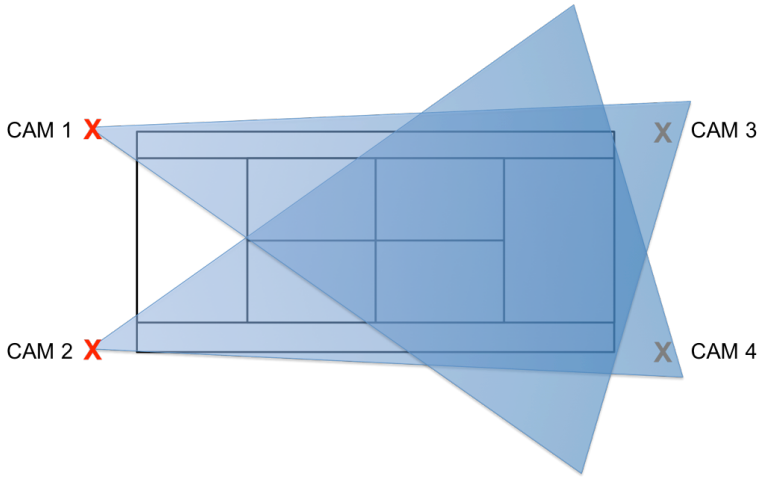

Figure 6.2: The figure depicts the position of the four cameras on the corners of the tennis court with X marks. Each pair of cameras can acquire the opposite part of the court. The fields of view of two cameras are highlighted in blue.

The proposed architecture makes use of four high-definition cameras that are synchronized using a trigger circuit. The model chosen is an AVT Prosilica GT 1920C. This is a Gigabit camera with a maximum resolution of 1936×1456 with a maximum frame rate of 40 fps. However, since the system has been designed to operate also with artificial lights, some horizontal scanlines have been cropped to achieve a fixed frame rate of 50 fps, which is the same of the

power lights fluctuations (which operates at 50Hz in Europe). This frame rate boost essentially removes any flickering issues due to the use of artificial lights and enables the system to operate better during serves, where ball usually travels at its maximum speed. Under these conditions acquired frames have a resolution of 1920×1024 pixels. These cameras are also equipped with auto-iris lens control, enabling to stabilize brightness even during extended recording sessions, spanning several hours.

Figure 6.3: Example of synchronized acquisition. Four frames are captured exactly at the same time to make the system capable of observing the whole court from at least two different points of view.

Cameras arrangement requires the use of two pairs of cameras. Each pair is positioned to cover the half-court on their opposite side. The fields of view of a pair of these cameras is shown in Figure 6.2 as the blue regions. The intersection of both fields of view is the area in which 3D reconstruction can take place by means of triangulation techniques. A central node, equipped with the trigger generator, synchronizes the four cameras and collects data. The synchronization task has been implemented with the aid of a low cost programmable micro controller, namely an Arduino, that has been set to operate at 50 Hz. A squared wave is then generated accordingly to the selected

frequency. This way, every installed camera can acquire an image exactly at the same time, meeting the requirement of each stereo vision system that needs a synchronized pair of images to compute the 3D information. Cameras are connected to the central node by using a Power Over Ethernet (PoE) Gigabit card. The central node is also equipped with dedicated SATA hard disks hosted on a SAS controller, independent from the operating system boot disk, providing storage capabilities that are both sized for extended playing sessions, and shielded from the interference of operating systems tasks. Each camera stream is physically stored on a dedicated partition of a different drive, thus enabling the system to store multiple raw data streams in parallel at the maximum available speed. An example of four synchronized images acquired by the four cameras is reported in Figure 6.3.

6.2.2 Data processing

The data processing consists of several modules (see Figure 6.1): low level processing, 3D reconstruction, high level processing, outcome decision processing.

Thanks to the nature of the dedicated installation, using fixed cameras with fixed zoom, low level processing performs moving blobs detection on each camera aided by a background subtraction approach. In this way, low level entities (i.e. ball and player candidates) present in each camera are extracted. Therefore for each camera, a temporal and spatial blob analysis is performed for filtering false candidates due to noise, for recognizing ball and players and remove artifacts due to shadows or net movements.

The resulting ball and players candidates of the corresponding pair of cameras of the same side of the court, are then forwarded to the 3D reconstruction module. For all these candidates, the 3D point cloud is extracted by applying a triangulation approach which exploits known geometrical relationships between cameras, and fixed positions of entities and corners of the court in the real world.

Finally, the 3D point cloud is processed by the high level processing module which applies semantic analysis for identifying ball trajectories and game events by using the domain knowledge such as the expected trajectory of the ball or the change of its direction and speed.

A finite state machine can be eventually used to perform the last step of outcome decision, by using semantic data stored in the previous step and by associating to the sequential evolution of the recognized events the score

assignment dictated by the tennis rules.

6.2.3 Data Storage

A huge amount of redundant data is available, but only few processed data are stored in a database in a fine-graded fashion: ball positions, events that change ball trajectory (such as impacts with the ground field or the players' racquet), players' positions, and score assignments (the last one resulting from the outcome decision process). A relational database, namely PostgreSQL, has been chosen to record the information in order to exploit the hierarchical structure of tennis domain data. The purpose is to obtain effective storage while enabling subsequent statistical analyses. Structured meta-data storage enables fast access to key match events, and, at the same time, provides a foundation for combining data in more meaningful ways in the future, thus enabling the system to be customized to coaches needs and easily expandable to accommodate ever-evolving requirements.

6.3 Processing modules

In this Section all the steps that are performed will be described, from the low level processing of the raw data coming from the four synchronized cameras up to the final decision process which collects semantic information and assigns a score.

6.3.1 Low level Processing

This module is responsible for ball and players detection. Several steps are needed to identify them:

1. Image acquisition and storage;
2. Robust background estimation and subtraction;
3. Moving region filtering by connectivity and morphological analysis;
4. Ball and Players candidate selection by spatial and temporal filtering.

The chosen sensors produce data in the form of color filter arrays using the Bayer pattern [191]. Thus, images coming from the camera are stored in their

raw form for archiving purposes. No conversion is done, so that data requires fewer resources to be stored. Color conversion is applied only on demand for display purposes, while for the subsequent processing raw images are used.

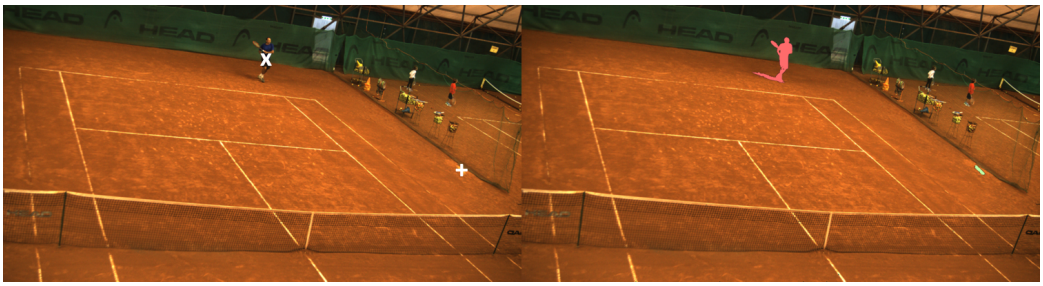

Figure 6.4: Graphical output of the low level processing in terms of entities coordinates and player silhouette. On the left side, two different symbols are used to mark the entities: X for the player and + for the ball. On the right side, the silhouette of the player is highlighted in red and ball in green. Data from this stage is essential to reconstruct three dimensional coordinates and proceed with the analysis of the match performing the subsequent tasks.

A suitable algorithm has been chosen to provide a background estimation and hence extract ball and players candidates with a high confidence level. In particular, the GIVEBACK algorithm is applied as described in Chapter 5. After the background model is learned, the fine tuning procedure continues with a selective mask update, keeping trace of robust foreground areas in which the background is not updated. This enables the algorithm to easily filter ghosts or subjects that stand still on the scene and to retain good player silhouettes.

After the background modeling and subtraction, a connected component analysis is applied to the resulting moving regions, to allow an initial estimation of the area dimensions. Then, morphological operations such as dilatation and erosion are used to remove holes and merge neighbor regions. These preliminary steps allow an initial filtering of noisy regions.

Ball candidates are chosen from blobs that have compatible dimensions with the tennis ball. In particular the radius of the inscribed and circumscribed circles of each region is estimated, selecting those having an inner radius between 5 and 10 pixels and an outer radius between 5 and 30 pixels. This is necessary to allow the detection of balls even at high speed, that, due to motion blur, present an ellipsoidal shape. Correct corresponding balls between consecutive frames are found by evaluating the shape features extracted from

the two images. For each candidate in one frame, the most similar are first chosen from the other frame. Then, the selection is refined by associating the closest balls in space. Moreover, a speed threshold is applied to filter ball candidates that are not likely to be part of the game. Whenever a match is played, only the fastest ball candidate is chosen, but during training sessions many balls can be thrown simultaneously and processed accordingly.

The selection of player candidates is done with a different set of operations. Since players move at a lower speed and sometimes are in "idle" state, for example waiting for the serve, there is a chance that parts of the silhouette might be considered as background. Even if GIVEBACK algorithm reduces the probability of observing such phenomenon, a morphological closing operation can be necessary to consider only big foreground areas as players candidates. It is worth noticing that the chosen background algorithm is able to deal with such situations, as it was developed specifically for this task.

An example of low level processing result is reported in Figure 6.4. On the left image different marks are used to sign the player and the ball, while on the right image the segmented player silhouette is shown in red and the ball in green.

6.3.2 3D reconstruction

This module is essentially responsible for providing 3D information of ball and players candidates. Each pair of synchronized images is exploited to produce a sparse point cloud that embeds information about the active entities on the court during the game. The algorithm is mainly composed of the following steps:

1. Homography computation;
2. Entity projection on the ground plane;
3. 3D information retrieval.

First of all, for each pair of cameras observing the opposite half of the field, the homography matrices which map the transformation between the image planes and the ground plane are estimated. A set of reference points placed on the ground plane is measured by a theodolite sensor in a global reference system, and their correspondences in the two image planes are annotated accordingly. Let (X, Y, Z) with $Z = 0$ be the coordinate of the a reference

point in the world reference system and (u, v) its corresponding coordinate on the image plane. The general transformation is given in the following equation:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} sX \\ sY \\ s \end{bmatrix} \tag{6.1}$$

that can be expressed in Cartesian coordinates as:

$$\begin{aligned} u &= \frac{h_{11}X + h_{12}Y + h_{13}}{h_{31}X + h_{32}Y + h_{33}} \\ v &= \frac{h_{21}X + h_{22}Y + h_{23}}{h_{31}X + h_{32}Y + h_{33}} \end{aligned} \tag{6.2}$$

To estimate the coefficients $h_{i,j}$ at least four corresponding points are needed thus solving the resulting equation system in the least squares sense. Additionally, the theodolite sensor is used to measure the position in the world reference system of the centers of projection (CP) of all the cameras. It is worth observing that this procedure is necessary only when installing the system to generate the four homography matrices for the four cameras. Given a point observed in the image plane it is possible to detect the corresponding position P on the ground plane and construct the viewing lines between CP and P as shown in Figure 6.5. Whenever the same point is observed by two cameras simultaneously, the intersection of the two viewing lines will then give a 3D point that represents its position in the world reference system.

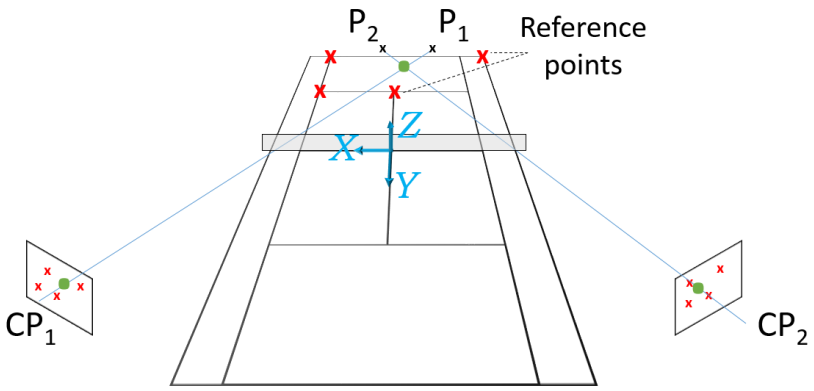

Figure 6.5: Example of 3D information retrieval from a pair of homologous cameras. The red points represent some of the reference points chosen on the ground plane which are used to estimate the homography matrices. The green dots represent the ball whose position is determined by the intersection of the two viewing lines (depicted in blue). The global reference system is also shown at the center of the court, on the ground plane.

In real cases, each pair of 3D lines constructed this way will not perfectly intersect in a specific point because of noise or numerical approximations. For this reason, the segment of minimum distance between the two skew lines is computed and the 3D point is finally associated to its midpoint.

6.3.3 High level processing

Once 3D positions of ball and players have been retrieved, this module is responsible to segment the acquired sequence into actions. This is initially done by detecting idle parts during the game, where no ball is moving, and splitting the whole sequence accordingly. These chunks of sequences might contain valid game actions or just show a suspended game where inactive balls are collected from the tennis court or exchanged between players while preparing for the next round of serves, a situation very frequent during training sessions. This classification is done after the ball trajectories evaluation.

The temporal analysis of the 3D coordinates of all ball candidates allows the reconstruction of the most complete trajectories which respect the kinematic constraints of the tennis ball. An interpolation procedure is used to approximate the real 3D measures and fill the gap due to missing data and filter out false measures due to noise.

Trajectory breaking is instead concerned with the identification of events affecting the ball. They can be classified as collision with the ground or collision with a player's racquet. These cases modify the ball trajectories and ultimately are events that start a new course for the ball. They are also the key events that are considered by game rules for assigning a score to the action. They can also be easily classified on the basis of their position on the court and of the principal axis where the change of direction is detected. In particular we have that:

- *Ball bounces* are on the ground, and they can be detected as local minima around zero on the Z axis.
- *Strokes* change the trajectory as well. They affect the ball in all directions and a player must be nearby.
- *Serves* happen on the border of the court when both players assume a specific position and are characterized by the initial height of the trajectory and the high speed.

Particular care in the detection of these trajectory changes must be taken so that noise due to errors of the background subtraction module or to wrong association in the triangulation procedure do not add false positives events. A Gaussian smoothing operation (whose parameters have been experimentally set) is applied before breaking trajectories.

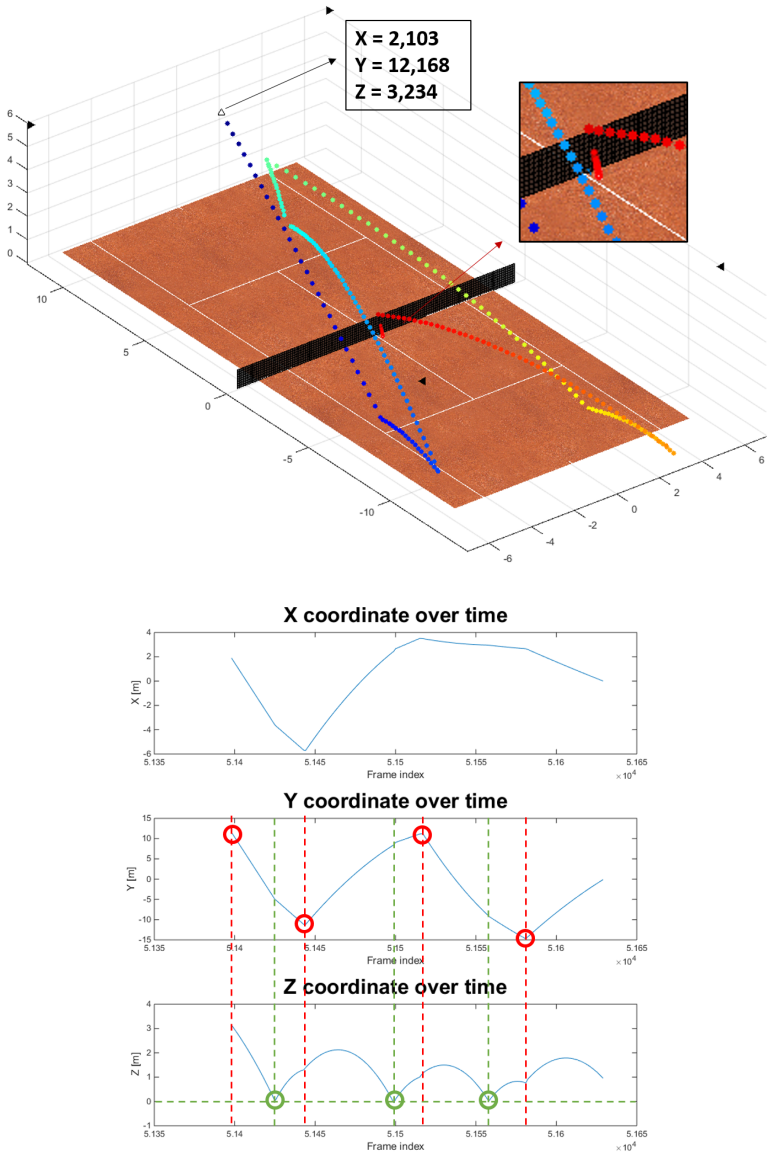

Figure 6.6: Example of the 3D ball trajectory and its corresponding X,Y,Z plots studied separately, called action plot. Bounces can be easily recognized looking at the third subplot and searching for local minima around zero (reported as green circles). Strokes can be found exploiting the changes in the sign of the acceleration along the Y-axis (reported as red circles). The box shows the coordinates of the ball in correspondence of the serve. The observations of the ball at the end of the rally are reported in the zoomed rectangle.

Figure 6.6 reports an example of the 3D ball trajectory enriched by separate plots for the (X, Y, Z) coordinates. The plot is referred to a complete action that starts with a serve and ends with a scored point, thus the evolution over time of a multiple rally is shown. As reported in Figure 6.5, the reference system has the origin in the middle of the court on the ground plane. Trajectory changes along Y are related to *strokes* and local minima and maxima, determined by changes in the sign of the acceleration, can be exploited to understand when a stroke happened. It is worth noting that the Gaussian smoothing operation is useful when dealing with real data, as it reduces the effects due to noise. *Bounces* on the ground are recognizable searching for local minima around 0 in the Z coordinates. *Serves* can be recognized by evaluating players positions on the court along with the height of the ball at the beginning of a tracked trajectory and its Y coordinate (that should be behind the side line), as reported in the box in the figure. Different ball colors in the 3D map of Figure 6.6 represent the trajectory over time: blue (serve), cyan, green, yellow, orange and finally red (point). Finally, the dots between the net and the ground (also zoomed in the figure) represent the observations of the ball at the end of the rally.

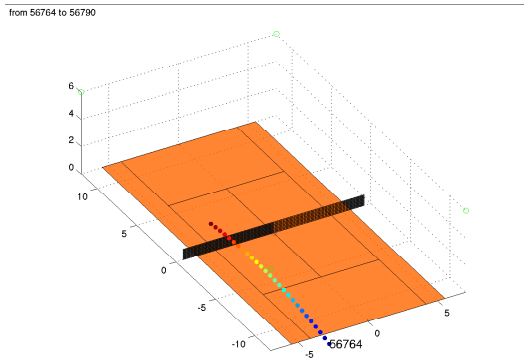

Figure 6.7: Example of invalid action: a ball pass between two valid points.

The resulting trajectory classification is stored in the database together with the estimated frame number and the ball coordinates in the court reference system. At the same time, in correspondence of these events players positions are checked and saved in the database. All this data is used by the following decision process that assigns a score, but can be used for

further applications such as query of specific key events, statistics and so on. Trajectories are considered as not valid when there are no strokes in a relatively short period of time, as in Figure 6.7 that describes the trajectory of a simple ball pass event between two valid points. Moreover, it is worth noting that the hardware setup described in Section 6.2 can also introduce time duplicated trajectories when the same ball is captured by both camera pairs (specifically above the net, where all the fields of view are overlapped).

6.3.4 Outcome decision processing

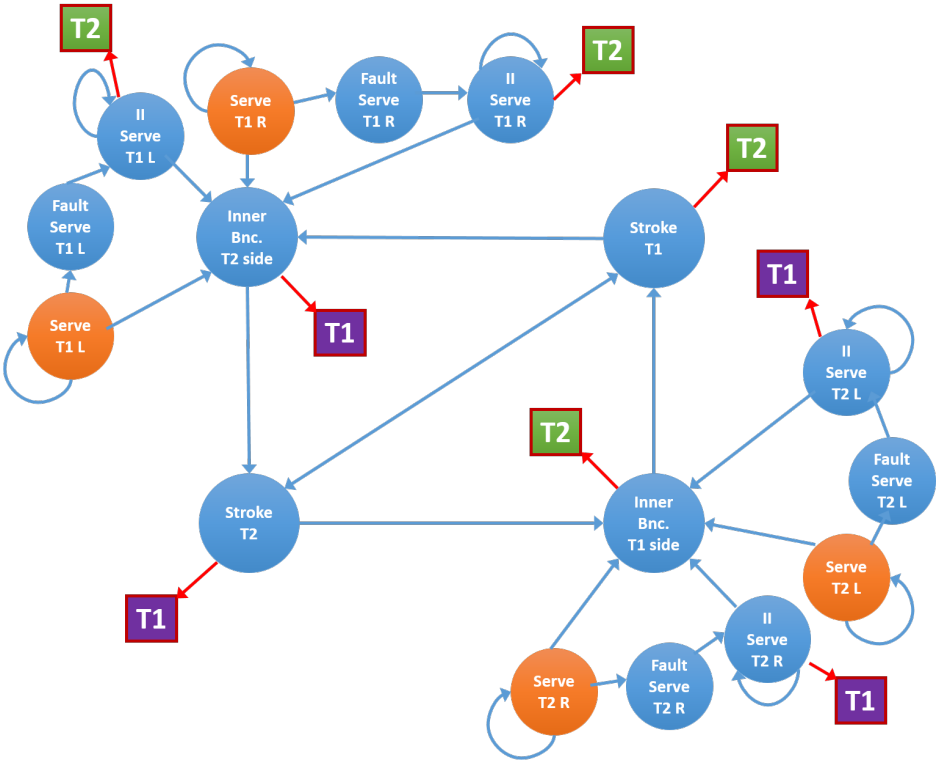

Figure 6.8: Graphical model of the finite state machine used to assign a point at the end of each action. The orange filled nodes represent the four initial states that can be found during a tennis match (the serve), while the blue ones are the inner states that describe the progress of the action. The connection between the nodes represent the allowed transitions only. The outcome is reported in squared boxes (purple for team T1 and green for team T2). Each proper event must fire a transition, i.e. the FSM must change its state at each iteration if the action is not concluded. Otherwise, the point is assigned to the green/purple highlighted team in correspondence of the node that did not changed its state.

A score assignment can start only when a complete action has been identified. A complete action is a sequence of frames which starts and ends with idle phases (a number of consecutive frames where no movement of the ball is perceived). The first step searches for a "serve" event, recognized as the first stroke after an idle period that happens near the side line. Then, the ball trajectory is analyzed until the end of the action, when a point is assigned to one of the players. A finite state machine (FSM), which embeds the rules of the game, has been designed. The finite state machine changes the state if the ball follows a valid trajectory with respect to the rules. When the FSM can not reach another valid state in response to an event, the action is considered completed and a point is assigned. Particular attention should be given to the repetition of a serve (first or second) that is allowed only when the served ball touches the net and bounces inside a valid area of the court. In that case, the particular service should not count and the service needs to be repeated without cancelling any previous fault. It should be noted that net events are important in this context only, otherwise they can safely be ignored to correctly assign a score. Figure 6.8 shows a graphical overview of the FSM, which resumes all the possible situations that can assign a score, starting from simple aces (for example Serve T1 L, Inner Bounce T2 side $\rightarrow$ score T1) to more complex actions with several strokes and bounces. In Table 6.1 all the possible states with their correspondent outcome are reported. The states are extracted by the events stored in the database in the previous step by analyzing both the type of events and the corresponding 3D ball coordinates. It is worth noting that valid court boundaries depend on both game type (single/double) and stroke type (serve or other strokes), therefore the meaning of "inside" and "outside" changes according to the rules of the game. In Figure 6.8 square boxes represent the key-value map that associates an outcome to the action: if the FSM is not able to change its state, then the point is assigned to the appropriate team.

Table 6.1: List of all possible FSM states with the correspondent outcome. The * means that no point can be assigned in the specific state. This is true only when a fault occurs and the serve can be repeated.

State	**Possible Outcome**
Serve T1 L/R	*
Fault Serve T1	*
II Serve T1 L/R	T2
Serve T2 L/R	*
Fault Serve T2	*
II Serve T2 L/R	T1
Inner Bounce T1 side	T2
Inner Bounce T2 side	T1
Stroke T1	T2
Stroke T2	T1

In order to explain the decision process by the FSM two examples are analyzed. Figure 6.9 shows two actions, represented by the sets of events $A_1 = [ev_1, ev_2, ev_3]$ and $A_2 = [ev_1, ev_2, ev_3, ev_4, ev_5]$. Blue lines represent ball trajectories between valid states, while the red lines depict the last state transition in which the decision about the point assignment is made. In the first case the events are: serve, bounce and another bounce. The associated states are respectively: Serve T1 R, Inner Bounce T2 side and Inner Bounce T2 side again. The latter state is due to the fact that the second bounce takes place on the same side of the court, therefore since there is no valid state transition, the FSM remains in the previous state. As shown in Figure 6.8 the point is assigned to T1.

Following the same approach, the events of the second action are: serve, bounce, stroke, bounce, bounce. The corresponding states are detailed in Table 6.2. The first stroke event ev_1 is a Serve made from team T1 on the right side of the court, so the initial state is "Serve T1 R". Then, the following events are responsible for allowed state transitions as specified in the Table 6.2, until the last state Inner bounce T1 side is repeated. Also in this case

the last event is another bounce on the same side of the court. Then there is no valid state transition and the point is assigned to T2.

It should be noted that the entrance state of the FSM cannot be necessarily a service, as the proposed system can be used also during training sessions with any kind of action. The FSM in this cases is used to label the observed actions and to store in the data base all the information about number of strokes, positions, bounces, with the resulting scores. These data can be used by coaches to perform useful queries and evaluate player performances both during training and official matches.

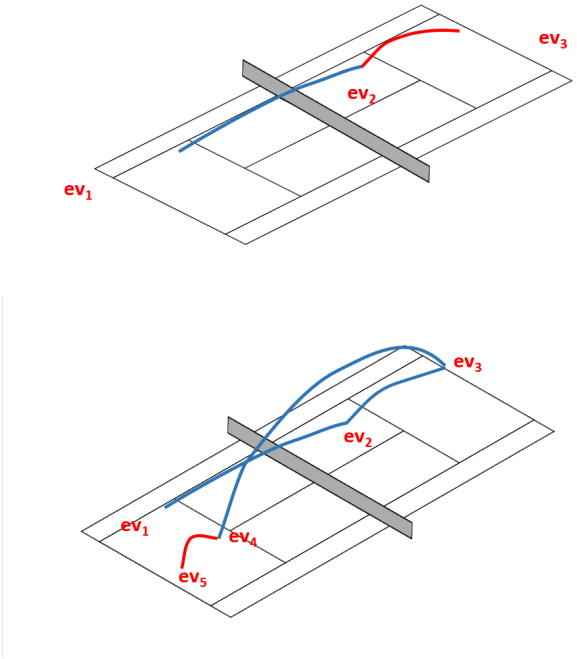

Figure 6.9: Examples of actions that can be processed with the FSM to decide the outcome. In the first case, the player on the left side performs an ace and wins the point, while in the second case a longer action is depicted.

Table 6.2: Analysis of the longer example action in Figure 6.9. The first event initializes the FSM with a Serve played by team T1 from the right side. Then, other events are received and the machine changes its state according to the representation in Figure 6.8. The last state is "Inner bounce T1 side", meaning that the point should be assigned to the team T2 according to the key-map representation in Table 6.1. The last event is not able to fire a transition, implying the end of the action.

Initial state	**Event**	**Arrival state**
*	(ev_1)	Serve T1 R
Serve T1 R	(ev_2)	Inner bounce T2 side
Inner bounce T2 side	(ev_3)	Stroke T2
Stroke T2	(ev_4)	Inner bounce T1 side
Inner bounce T1 side	(ev_5)	*No valid transition*, *end of action*

6.4 Experiments and results

The experiments have been performed on a clay court hosted by a private tennis club that has been equipped with the hardware described in Section 6.2.

Two different experiments have been conducted: the first one to demonstrate the performances of the proposed approach to recognize serves, strokes and bounces, by the analysis of the 3D ball trajectory variations; the second one to test the whole chain, with the FSM that assigns a final score to each action.

In the first case, a number of 225650 frames have been recorded by four synchronized cameras during different training sessions for a total of about 75 minutes. In these sessions, players are free to move on the court without strictly respecting rules and under the supervision of their coach, they can improve the technique on particular/unusual strokes, control the length of an action or repeat particular sequences of strokes in order to enhance tactics. The registration of these video sequences has been used to test the ability of the proposed system to segment subsequences which contain actions, track the ball and reconstruct the 3D trajectories in order to recognize services, bounces and strokes.

Before proceeding in the analysis of the results, it should be observed that

in this kind of real experiments is actually difficult to establish the ground truth for 3D ball trajectory evaluation. It would be necessary to employ external measurement sensors able to exactly measure the ball trajectory. What is generally done is to manually label the ball position in the images and reconstruct the 3D ball coordinates by triangulation. In this case also the resulting measures are affected by errors due to the optical projections of the system, the matrices approximations, and so on. The scope of this work is not to exactly evaluate the 3D ball trajectory, but only to establish the ability of the system to recognize events. The ground truth has been then generated by manually labeling the frames in which users recognize strokes, bounces and services, and comparing these results with those produced by the system.

Table 6.3: Experiment 1: the table reports the results in terms of TP, FP, and FN for the recognition of strokes, bounces, and service during a training session of 75 minutes. In the first column the numbers of events manually labelled during the training sessions. The values of TP, FP, FN, P, R are all percentages.

	GT	TP	FP	FN	P	R
Serve	112	85.8	0.0	14.2	100	85.8
Shot	409	89.6	5.6	4.8	94.1	94.9
Bounce	467	85.9	9.2	4.9	90.3	94.6

In Table 6.3 the results in terms of True Positives, False Positives, False Negatives, Precision and Recall are provided for a total of 112 services, 409 strokes, and 467 bounces. A detection has been considered correct when an event is recognized within a temporal window of 20 frames corresponding to $0.4s$ which seems a reasonable interval in which different events cannot be found. The detection in terms of true positives are satisfactory for all the kinds of events. In particular, the number of services which are not recognized are due to the fact that the ball tracking starts later than the actual ones, and the constraints for the service detection are not met anymore (a service is a stroke that happen near the bottom line of the tennis court and with a certain elevation with respect to the ground). In the shot detection there is a percentage of False Positive due to the fact that the analysis of the changes in the sign of acceleration can be caused by local minima which were not filtered

by the smoothing step. Also for the bounce detection, some false positives are due to strokes which happen very close to the field and the constraint on the minimum around 0 in the Z coordinate fails. False negatives are generally caused by a not precise trajectory reconstruction due to failures in the ball detection in some frames. For example in Figure 6.10 one of the cameras cannot see the ball as it is saturated by the lights on the advertisement. The system overall performance can be then evaluated in terms of Precision and Recall percentages that have been obtained during the experiment. The first value indicates how many selected items are relevant, while the second one expresses how many relevant items have been output by the system. For the categories Shot and Bounce the system is able to achieve values greater than 90%, proving its capability to output a high number of True Positive values along with a low number of False Positive or Negatives. The Serve Recall value scores 85.8% and is degraded by the False Negative values that have been discussed before.

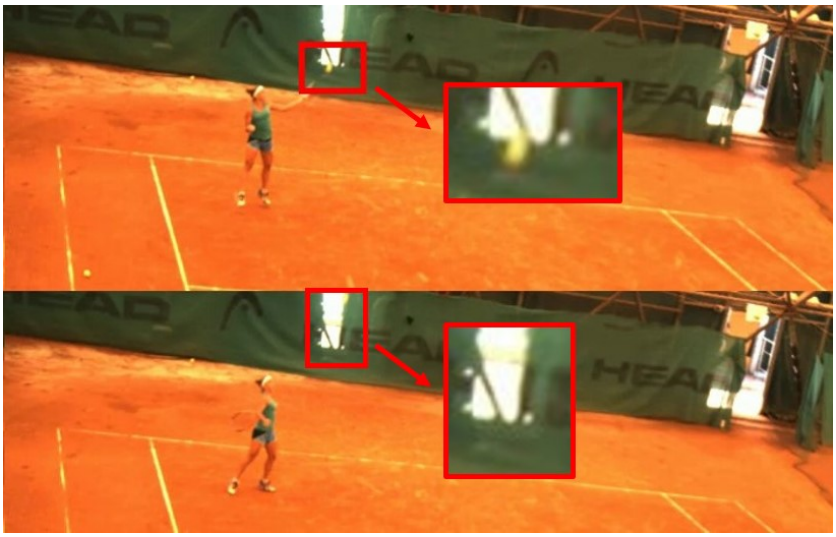

Figure 6.10: Examples of images in which the ball is not visible by one of the cameras.

Another point which should be considered for the evaluation of the system is the precision in the evaluation of the ball position when bounces are recognized as these data are necessary to assess if the ball is outside or inside the valid court. Certainly the performances of the proposed system cannot be compared to complex commercial ones (such as Hawkeye [145]) based

on a greater number of cameras which observe only the lines and perform 3D trajectory reconstruction. Anyway in order to have a general idea about these measurement errors, the position of the ball in correspondence of the observed bounces has been manually labeled, thus comparing the ball positions obtained by our system with those estimated by the manually annotation procedure. The same observation made beforehand about the ground truth is still valid. Also these measures are affected by noise, as the variation of one pixel in the ball manual labeling produces variations in the ball localization of several centimeters. For this reason, comparative results can be considered only in a qualitative way. Indeed, for the 91% of bounces the ball position error is estimated under $15cm$. As a consequence, when the ball is close to border of the valid field, the system could fail in determining inside or outside situations.

At this point the actions can be preliminarily analyzed for statistical purposes grouping them with respect to the number of strokes that occur during the play of a single point. Figure 6.11 shows the distribution of the actions according to this representation:

- short duration ones are reported in blue and cover about 43% of the total number of actions and generally refer to faults, aces or points that finish just after a couple of strokes;
- medium duration actions are the green ones, represent about 39% and are related to well balanced points in which both players are playing similarly;
- long duration actions, the red ones, are only 17% and can be exploited similarly to the previous ones.

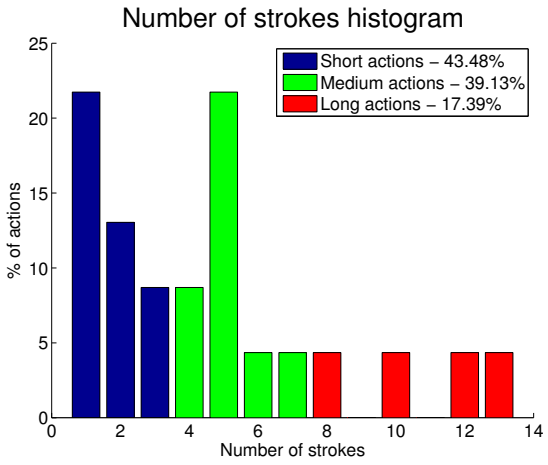

Figure 6.11: Number of strokes histogram.

This kind of statistics can be used by coaches to evaluate performances and extract video sequences containing specific events such as actions with one stroke (probably corresponding to fault or winning return), actions with long exchanges, and so on.

The second experiment consisted of 38000 synchronized frames that cover a real match made of four games. In this experiment the finite state machine has been tested to assign points and keep track of the score. It is important to put in evidence that the FSM just embeds information on game rules that are fixed. Therefore, its behavior is deterministic. For this reason, the purpose of this second part of the experiments is to understand whether this system will be able to effectively assist a coach with a high confidence level. A total number of 45 actions automatically tagged as valid or invalid by the system have been identified.

Table 6.4 contains the details about each action. It reports the manually labeled ground truth (Start frame, End frame, Srv, Str, Bnc), the corresponding system output, and the final evaluations carried out by our system. The ground truth in terms of start frame and end frame has been reported only for the valid actions, while for ball pass actions or idle periods are not reported. Strokes and bounces are correctly detected, while in three cases serves are not recognized (actions 4, 7 and 42). Ground truth start frame is correctly identified within a certain number of frames, showing that the system can effectively segment actions. Only for the three actions (those in which serves

have not been recognized) the starting frames has been found later than the actual ones. This is a clear indication that the ball was not initially visible and the tracked trajectory started later, causing a failure of the constraints for the serve detection. Moreover, the $1:1$ correspondence between the evaluated outcome and ground truth data confirms the capability of identifying bounces and strokes correctly. Action number 27 is an example of short trajectory fully contained in action 26 that is neglected because it is a duplicate one. Actions are considered valid if they start with a serve. When a serve is not recognized and actions have a long duration with a high number of strokes and bounces, actions are considered still valid but labeled with Missing serve.

Table 6.4: This table shows the details about actions, one per each row. Validity and events count are reported as well as temporal information in terms of starting frame, ending frame and duration, compared with manually annotated ground truth. Invalid actions are characterized by one of the following: absence of strokes in combination with a relatively short duration, sub actions whose temporal boundaries are intersected with a main action (as explained in Section 6.4) or simple ball pass events between two points. Finally, three missing serves have been highlighted in bold.

	Ground truth					System output							
#	Start frame	End frame	Srv	Str	Bnc	Start frame	End frame	Srv	Str	Bnc	Delta start	Delta end	Decision
01	47855	47964	1	1	1	47852	47916	1	1	1	3	48	Valid
02	48387	48559	1	2	2	48380	48592	1	2	2	7	-33	Valid
03	49524	49693	1	2	2	49519	49651	1	2	2	5	42	Valid
04	50569	50918	1	3	4	50589	50875	0	3	4	-20	329	**Missing serve**
05	51402	51658	1	3	3	51393	51629	1	3	3	9	29	Valid
06	-	-	0	0	2	52015	52211	0	0	2	-	-	Ball Pass
07	52648	52840	1	4	5	52667	52984	0	4	5	-19	-144	**Missing serve**
08	54751	55722	1	12	12	54747	55763	1	12	12	4	-41	Valid
09	-	-	0	0	0	54879	54912	0	0	0	-	-	Overlap
10	56425	57129	1	9	9	56416	57135	1	9	9	9	-6	Valid
11	-	-	0	0	0	56764	56790	0	0	0	-	-	Overlap
12	58133	58178	1	0	1	58126	58178	1	0	1	7	0	Valid
13	-	-	0	1	1	58424	58546	0	1	1	-	-	Ball pass
14	59002	59360	1	4	5	58945	59323	1	4	5	57	37	Valid
15	-	-	0	0	0	59181	59223	0	0	0	-	-	Overlap
16	-	-	0	1	2	59546	59758	0	1	2	-	-	Ball pass
17	60731	61350	1	4	5	60673	61099	1	4	5	58	251	Valid
18	-	-	0	1	1	61095	61333	0	1	1	-	-	Overlap
19	-	-	0	0	1	61790	61925	0	0	1	-	-	Ball pass
20	-	-	0	1	1	61677	61791	0	1	1	-	-	Ball pass
21	-	-	0	0	0	62348	62483	0	0	0	-	-	Ball pass
22	63615	64008	1	5	4	63612	64001	1	5	4	3	7	Valid
23	-	-	0	1	1	64521	64664	0	1	1	-	-	Ball pass
24	-	-	0	0	1	64633	64782	0	0	1	-	-	Ball pass
25	65213	65561	1	4	5	65206	65562	1	4	5	7	-1	Valid
26	-	-	0	1	3	65892	66106	0	1	3	-	-	Ball pass
27	-	-	0	0	0	65944	66065	0	0	0	-	-	Ball pass
28	66525	66567	1	0	1	66521	66553	1	0	1	4	14	Valid
29	66968	67189	1	0	1	66970	67002	1	0	1	-2	187	Valid
30	-	-	0	1	2	67612	67758	0	1	2	-	-	Ball pass
31	68113	68956	1	11	11	68107	68928	1	11	11	6	28	Valid
32	-	-	0	0	0	68581	68609	0	0	0	-	-	Overlap
33	70092	70197	1	1	1	70088	70171	1	1	1	4	26	Valid
34	70891	71003	1	1	2	70895	71017	1	1	2	-4	-14	Valid
35	72449	72509	1	0	2	72443	72499	1	0	2	6	10	Valid
36	72705	73241	1	7	7	72702	73228	1	7	7	3	13	Valid
37	74776	75287	1	6	7	74711	75271	1	6	7	65	16	Valid
38	-	-	0	0	0	75055	75087	0	0	0	-	-	Overlap
39	-	-	0	1	1	75551	75738	0	1	1	-	-	Ball pass
40	76289	76357	1	0	1	76227	76335	1	0	1	62	22	Valid
41	-	-	0	0	2	76447	76604	0	0	2	-	-	Ball pass
42	78415	78835	1	4	5	78444	78747	0	4	5	-29	88	**Missing serve**
43	-	-	0	1	1	84498	84627	0	1	1	-	-	Match end
44	-	-	0	0	2	85286	85469	0	0	2	-	-	Match end
45	-	-	0	2	1	85821	85933	0	2	1	-	-	Match end

Table 6.5: The table reports the action number, the initial state of the FSM, the outcome of each action (Win) assigned by the FSM, and the score of the actions that could be assigned automatically. The notation W is used to identify the game point.

Action	**In. State**	**Win**	**Game**	**T1**	**T2**
01	Serve T2 R	Fault	1	0	0
02	II Serve T2 R	T2	1	0	15
03	Serve T2 L	T1	1	15	15
04	In Bnc T1	T2	1	15	30
05	Serve T2 L	T2	1	15	40
07	In Bnc T1	T2	1	15	W
08	Serve T2 R	T1	2	15	0
10	Serve T2 L	T2	2	15	15
12	Serve T2 R	T1	2	30	15
14	Serve T2 L	T2	2	30	30
17	Serve T2 R	T1	2	40	30
22	Serve T2 L	T1	2	W	30
25	Serve T1 R	T2	3	0	15
28	Serve T1 L	Fault	3	0	15
29	II Serve T1 L	T1	3	15	15
31	Serve T1 R	T2	3	15	30
33	Serve T1 L	T2	3	15	40
34	Serve T1 R	T2	3	15	W
35	Serve T1 R	Fault	4	0	0
36	II Serve T1 R	T1	4	15	0
37	Serve T1 L	T1	4	30	0
40	Serve T1 R	T1	4	40	0
42	In Bnc T2	T1	4	W	0

Then, the 23 valid actions (20 valid and 3 missing serves) extracted by the previous high level processing, have been analyzed in order to automatically annotate the score. Table 6.5 shows the action number, the initial state of the FSM, the outcome of each action (Win) assigned by the FSM, and the score of the games that could be assigned automatically. Although in the actions 4, 7 and 42 the initial serves were missed and the initial states of the

FSM were Inner bounces T1 or T2, the scores were correctly assigned to the correct team. In this case, as the initial preprocessing correctly recognizes the other events, the score assignment in the considered sequence of games is performed correctly as well.

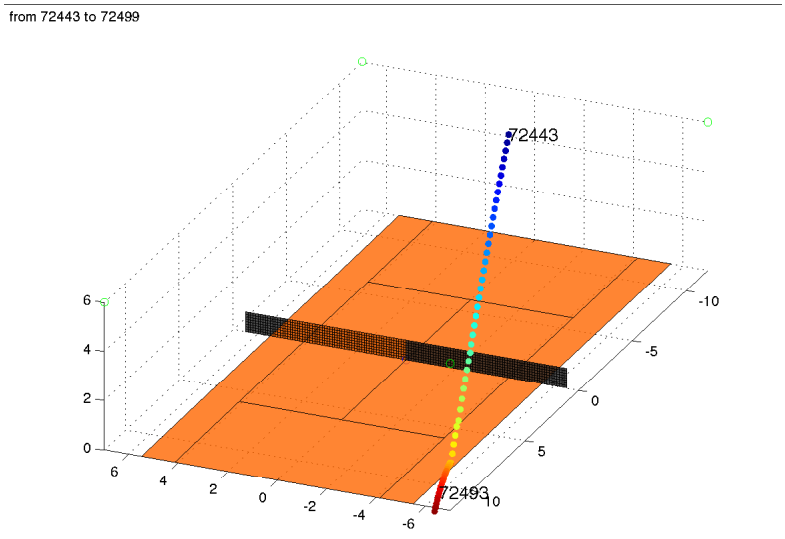

Figure 6.12: Example of wrong serve in which the bounce is outside the allowed area.

Some examples of the reconstructed actions with the ball trajectories plotted in a 3D Euclidean space are reported in Figures 6.12 and 6.13. Figure 6.12 represents a fault where the ball bounce is outside the side line. Figure 6.13 represents action number 22. The colors change according to the frame index (blue, cyan, green, yellow, orange and finally red). Some relevant strokes are highlighted in the Figure 6.13 and represented as players are seen by the respective cameras. The serve is the starting event of the action and is depicted with a blue mark: the player is dressed in blue and assumes the typical serve position (similar to a smash) behind the side line. Other three examples of strokes are provided in the same Figure: the return, that takes place outside the single sideline; a shot (highlighted in green) played by the white-dressed player on the extreme right side of the court and finally a stroke between the service line and the net (yellow rectangle). The miniatures of players demonstrate the potential use of data: coaches can perform intelligent

queries to the database, can extract specific actions and analyze just the frames in which players hit the ball and perform a stroke.

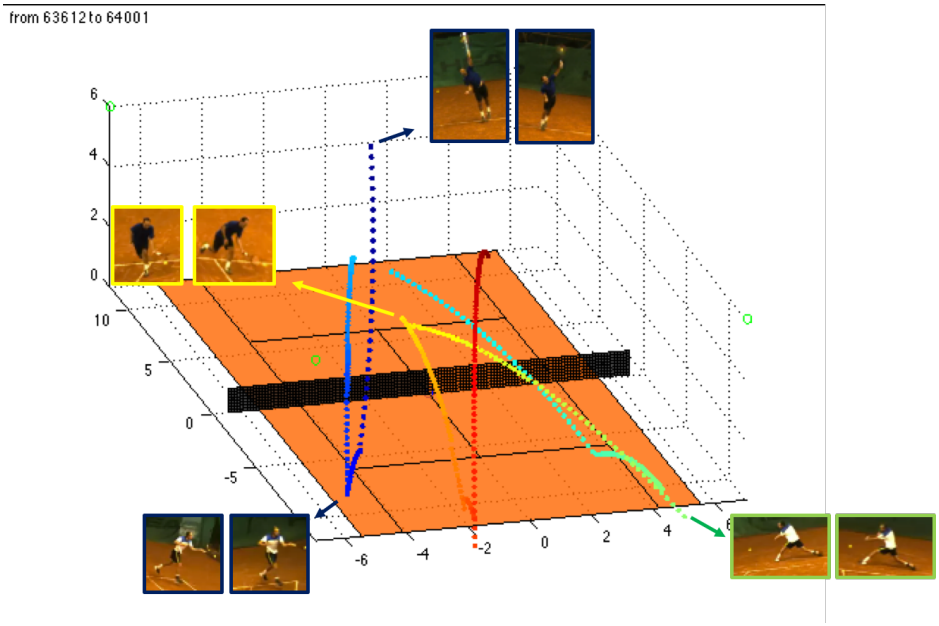

Figure 6.13: Example of full action with two relevant strokes highlighted per each player. Since the database contains an event-indexed representation of the match, it is extremely simple to seek and view key events during a match.

The considered sequence of games was used to demonstrate that the FSM is able to correctly decide the score assignment whenever the preliminary decision process (i.e. event recognition) is correct. Anyway, as discussed beforehand, in some cases bounces can be confused or assigned erroneously inside/outside. In these cases the score could be wrong. However, since the scope of the work is neither to do extremely precise measures nor automatic score assignment but to allow coaches to extract interesting video sequences for player performances analysis, it is not relevant if some actions terminate with a wrong score assignment. The wrong decision can be manually adjusted by the coach in a second time. As the FSM can be used also during training sessions to label extracted actions with scores, coaches can save long time analyzing only short video sequences which contain significant exchanges instead of observing all the recorded sequences that can potentially last for hours. In particular, coaches can exploit this system functionality to filter

relevant parts of the recorded sequences according to their training strategies. As illustrative examples, to improve the attacking capabilities of players, all the events that occur in the attack zone can be selected, or to evaluate the reacting capabilities, consecutive actions with lost scores can be analyzed.

Software modules have been developed in C++ and Matlab languages. In particular, low level processing is the most computationally expensive task, as it must run on each raw video frame. The current implementation of this module runs at 30 fps and will certainly benefit from further optimizations. Once 3D information are extracted from the low level processing module, the high level processing (trajectories processing, events recognition and outcome decision) are performed in Matlab environment and do not require further optimization. However, the whole system architecture will likely benefit from the integration of all these modules in the same language.

6.5 Summary

The work described in this chapter is a visual system based on four synchronized cameras which is able to record training and official tennis matches, segment action in frame sequences, recognize significant events such as strokes, bounces or services, and eventually assign a final score. The system has been designed to meet requirements coming from domain experts. It can be used by coaches and players to analyze long training sessions, and without observing all the sequences, extract significant actions, such as those ending with a positive score or containing at least a certain number of strokes, etc... This is one of the first systems which tries to automatically segment video sequences while adding semantic information useful for player performance analysis.

The system integration phase involved an accurate hardware choice for making the proposed solution modular, scalable and flexible at the same time. Design and implementation of software integrated solutions have been investigated as well, in order to obtain an event based indexed representation of a match starting from big raw data acquired from cameras. A remarkable feature of the whole approach consists in the absence of invasiveness: players are simply free to behave like they already do while the system does acquisition and processing differently from wearable-based solutions. Finally, coaches can exploit the proposed system to filter relevant parts of the recorded sequences according to their training strategies saving time and maximizing their productivity.

Chapter 7

Conclusions and future works

In this thesis, the problem of 3D modeling, reconstruction and analysis of environments has been addressed from multiple points of view in order to provide effective and efficient methods to capture data and perform complex processing tasks. The work has been focused on the mechanisms that can be used to produce both dense and sparse point clouds, as well as on real time data processing techniques. As a matter of fact, the design and development of efficient algorithms is mandatory in systems that need to correctly represent three dimensional complex scenes to perform semantic high level analyses. Both dense and sparse point clouds respectively produced by active and passive sensors have been analyzed, along with proper methodologies that can deal with the specific type of data. In details, the following objectives have been pursued:

1. Design and development of a miniaturized catadioptric sensor capable of high throughput acquisitions with large field of view (for 3D reconstruction purposes);

2. Design and development of a proper methodology to perform three dimensional registration of point clouds acquired at different epochs (with application to the structural monitoring);

3. Design and development of suitable algorithms for the real time processing of high throughput data coming from passive systems (with application to the athletic scene processing);

4. Design and development of a technology platform for the high level analysis of complex scenes (applied to the tennis context).

Chapter 3 presents an innovative device for the inspection of surrounding spaces based on catadioptrics. The vision system equipped with a telecentric lens is assisted by a parabolic mirror in order to inspect the environment with high resolution and large field of view. Moreover, the initial specification of compactness has been met, providing a compact device able to perform 3D reconstruction with the resolution of $10mm$ for targets that are located $3m$ far from the laser emitters. The presented device constitutes one of the best options for many applications, such as inspection of pipes or monitoring of confined spaces, where reliable range measurements with high acquisition rates, high resolution and accuracy are mandatory, especially when miniaturized solutions are needed to achieve the goal. The flexibility of such sensor is one of its strengths and suggests a wide variety of future works, as an example it could be employed in hostile environments – like foggy or low-visibility ones – if equipped with proper illuminators and mirrors. Also, technology improvements on cameras will make such sensor capable of better resolutions with respect to those obtained with the presented prototype. This means that point clouds will be certainly produced with higher number of samples, suggesting that also the research on proper processing methodologies is mandatory to be ready to deal with the challenges that will be faced in the next years.

The numerical approach for point cloud registration returned by a laser rangefinder presented in chapter 4 has been focused on the topic of remote sensing of indoor civil infrastructures at different epochs, where standard approaches based on GPS are no longer available. It is based on the newly introduced deletion masks, that are able to iteratively discard the points that can induce erroneous registrations due to slightly different points of view of the sensors. Additional contributions will be dedicated to the reduction of the computational time required by the creation of the deletion masks, that represent one of the most time consuming tasks in the actual formulation. As an example, pre-computed look-up tables can be loaded at each iteration in order to speed up the algorithm. This is strictly connected to the concept introduced beforehand, because both the academic and industrial world will certainly benefit from smarter algorithms able to deal with the extremely high throughput of data coming from newly developed sensors. This is true not only for active devices, but also for passive ones.

Chapter 5 shows the details about multiple algorithms for real time processing of high throughput data streams. In particular, for the three background models, the results prove the effectiveness of the proposed approaches applied

to the sportive context, along with their robustness when compared to other statistical non parametric algorithms evaluated in the benchmarks. It is also reasonable to assert that the presented background algorithms should be effectively applied to other contexts, for example the intelligent video surveillance. Future works will certainly regard the optimization of the proposed algorithms with an implementation directly on smart cameras (e.g. on FPGA cards or ARM cpus), in order to further speed up computations and move some computational load directly on intelligent cameras. This is a point of particular interest because of its strong connection with the technological progress in terms of throughput because sensors will be able to dramatically increase the amount of produced raw data, as remarked beforehand. The proposed tracking method, on the contrary, has been developed particularly for the tennis context starting from a sparse and cluttered point cloud obtained by a stereo system. It has been designed to follow a point particle that evolves over time – namely a tennis ball – in order to associate a label to each trajectory made by the ball. Future direction of this research will certainly regard extensive tests to prove the multiple balls tracking capability of the algorithm. If each identified tracklet is treated separately and joined to a compatible one when the domain constraints are met, the algorithm will be able to deal with multiple trajectories that are likely to be observed during the training sequences. Moreover, accuracy improvements of sample estimations in absence of observed data will be investigated, working on more robust models for each tracklet.

Finally, the technology platform shown in chapter 6 is a complex system that meets the requirements coming from tennis domain experts. Particular attention has been put on maximizing the performance while keeping the costs as lower as possible. The proposed solution is able to automatically extract an event based indexed representation of a tennis match starting from data captured by four cameras assisted by modular and scalable software layers. Highly redundant data have been processed to compute a sparse point cloud that embeds information about the active entities of the game in order to save the results on a database. This kind of architecture enables the exploitation of high level data to enrich the analysis of game tactics and players intentions by properly combining the information saved on the database and thus assisting the coach during intensive training sessions. Future works will involve machine learning techniques in order to analyze relevant game patterns made by a specific player during multiple matches, as well as the automatic inference of winning tactics given specific initial conditions.

Bibliography

[1] M. Abdelguerfi. *3D Synthetic Environment Reconstruction*. Springer Science & Business Media, Apr. 30, 2001. 180 pp. ISBN: 978-0-7923-7321-6.

[2] F. Blais. "Review of 20 Years of Range Sensor Development". In: *Journal of Electronic Imaging* 13.1 (2004). URL: http://nparc.cisti-icist.nrc-cnrc.gc.ca/npsi/ctrl?action=rtdoc&an=5763160 (visited on 10/04/2016).

[3] W.-F. Xie, Z. Li, X.-W. Tu, and C. Perron. "Switching Control of Image-Based Visual Servoing with Laser Pointer in Robotic Manufacturing Systems". In: *IEEE Transactions on Industrial Electronics* 56.2 (2009), pp. 520–529. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4602717 (visited on 10/04/2016).

[4] L. Deng, F. Janabi-Sharifi, and W. J. Wilson. "Hybrid Motion Control and Planning Strategies for Visual Servoing". In: *IEEE Transactions on Industrial Electronics* 52.4 (2005), pp. 1024–1040. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1490692 (visited on 10/04/2016).

[5] C.-C. Wang and D. Tang. "Seafloor Roughness Measured by a Laser Line Scanner and a Conductivity Probe". In: *IEEE Journal of Oceanic Engineering* 34.4 (2009), pp. 459–465. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5233843 (visited on 10/04/2016).

[6] R. Taylor, N. Hancock, and T. Tran-Cong. "Non-Contact Extrudate Profilometer-Introductory Paper". In: *Mechatronics and Machine Vision in Practice, 1997. Proceedings., Fourth Annual Conference on*. IEEE, 1997, pp. 158–162. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=625315 (visited on 10/04/2016).

[7] A. Sinha, W. Sun, and G. Barbastathis. "Surface Profilometry at Large Working Distances Using Volume Holographic Optics". In: *Conference on Lasers and Electro-Optics*. Optical Society of America, 2004, CTuX4. URL: https://www.osapublishing.org/abstract.cfm?uri=CLEO-2004-CTuX4 (visited on 10/04/2016).

[8] M. J. Baker, J. Xi, J. F. Chicharo, and E. Li. "Optimisation of Triangulation Based Optical Profilometers Utilising Digital Video Projection Technology". In: (2005). URL: https://works.bepress.com/jxi/3/ (visited on 10/04/2016).

[9] S. Mohottala, S. Ono, M. Kagesawa, and K. Ikeuchi. "Fusion of a Camera and a Laser Range Sensor for Vehicle Recognition". In: *Machine Vision Beyond Visible Spectrum*. Springer, 2011, pp. 141–157. URL: http://link.springer.com/chapter/10.1007/978-3-642-11568-4_6 (visited on 10/04/2016).

[10] S. Kriegel, C. Rink, T. Bodenmüller, A. Narr, M. Suppa, and G. Hirzinger. "Next-Best-Scan Planning for Autonomous 3d Modeling". In: *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 2850–2856. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6385624 (visited on 10/04/2016).

[11] H. Alismail, L. D. Baker, and B. Browning. "Automatic Calibration of a Range Sensor and Camera System". In: *2012 Second International Conference on 3D Imaging, Modeling, Processing, Visualization & Transmission*. IEEE, 2012, pp. 286–292. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6375006 (visited on 10/04/2016).

[12] R. Tsai. "A Versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using off-the-Shelf TV Cameras and Lenses". In: *IEEE Journal on Robotics and Automation* 3.4 (1987), pp. 323–344. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1087109 (visited on 10/04/2016).

[13] Z. Nemoto, H. Takemura, and H. Mizoguchi. "Development of Small-Sized Omni-Directional Laser Range Scanner and Its Application to 3D Background Difference". In: *Industrial Electronics Society, 2007. IECON 2007. 33rd Annual Conference of the IEEE*. IEEE, 2007, pp. 2284–2289. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4460106 (visited on 10/04/2016).

[14] Y.-L. Chen and S.-H. Lai. "An Orientation Inference Framework for Surface Reconstruction from Unorganized Point Clouds". In: *IEEE Transactions on Image Processing* 20.3 (2011), pp. 762–775. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5570956 (visited on 10/04/2016).

[15] T. Sasaki, D. Brscic, and H. Hashimoto. "Human-Observation-Based Extraction of Path Patterns for Mobile Robot Navigation". In: *IEEE Transactions on Industrial Electronics* 57.4 (2010), pp. 1401–1410. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5226588 (visited on 10/04/2016).

[16] S. H. Cho and S. Hong. "Map Based Indoor Robot Navigation and Localization Using Laser Range Finder". In: *Control Automation Robotics & Vision (ICARCV), 2010 11th International Conference on*. IEEE, 2010, pp. 1559–1564. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5707420 (visited on 10/04/2016).

[17] F. He, Z. Du, X. Liu, and Y. Ta. "Laser Range Finder Based Moving Object Tracking and Avoidance in Dynamic Environment". In: *Information and Automation (ICIA), 2010 IEEE International Conference on*. IEEE, 2010, pp. 2357–2362. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5512068 (visited on 10/04/2016).

[18] K. Saito, N. Yata, and T. Nagao. "Three-Dimensional Scene Reconstruction Using Stereo Camera and Laser Range Finder". In: *SICE Annual Conference 2010, Proceedings of*. IEEE, 2010, pp. 1515–1520. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5602848 (visited on 10/04/2016).

[19] N. J. Chen and J. S. Chen. "3D Scenes Registration Using a 2D Laser Range Finder". In: *2010 IEEE International Conference on Automation and Logistics*. IEEE, 2010, pp. 457–463. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5585328 (visited on 10/04/2016).

[20] C. Poullis and S. You. "3d Reconstruction of Urban Areas". In: *2011 International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission*. IEEE, 2011, pp. 33–40. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5955340 (visited on 10/04/2016).

[21] S. Barba and F. Fiorillo. "3D Modeling for Documentation and Monitoring of Landslide Risk". In: (2012). URL: http://www.academia.edu/download/42548263/4710a134.pdf (visited on 10/04/2016).

[22] *Laser Line Scanner — 2D Laser Scanning Device — Laser Measuring Sensor*. URL: http://www.acuitylaser.com/products/item/accurange-line-scanner (visited on 10/04/2016).

[23] J. Gluckman and S. K. Nayar. "Rectified Catadioptric Stereo Sensors". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24.2 (2002), pp. 224–236. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=982902 (visited on 10/04/2016).

[24] "Sensore per La Ricostruzione Ambientale Tridimensionale Ad Alta Precisione". Pat. RM 2009 A 0003 67B. Stella, E., Marino, F., De Ruvo, P., Nitti, M., and Distante, A.

[25] P. De Ruvo, G. De Ruvo, A. Distante, M. Nitti, E. Stella, and F. Marino. "An Omnidirectional Range Sensor for Environmental 3-D Reconstruction". In: *2010 IEEE International Symposium on Industrial Electronics*. IEEE, 2010, pp. 396–401. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5637870 (visited on 10/04/2016).

[26] P. De Ruvo, G. De Ruvo, A. Distante, M. Nitti, E. Stella, and F. Marino. "An Environmental 3-D Scanner with Wide Fov Geometric Parameters Set up". In: *IEEE International Conference on Imaging Systems and Techniques (IST) 2010*. IEEE, 2010, pp. 111–114. URL: https://iris.poliba.it/handle/11589/15217 (visited on 10/04/2016).

[27] F. Marino, P. De Ruvo, G. De Ruvo, M. Nitti, and E. Stella. "HiPER 3-D: An Omnidirectional Sensor for High Precision Environmental 3-D Reconstruction". In: *IEEE Transactions on Industrial Electronics* 59.1 (2012), pp. 579–591. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5754573 (visited on 10/04/2016).

[28] S. Ouddane, S. A. Fezza, and K. M. Faraoun. "Stereo Image Coding: State of the Art". In: *Systems, Signal Processing and Their Applications (WoSSPA), 2013 8th International Workshop on*. IEEE, 2013, pp. 122–126. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6602348 (visited on 10/05/2016).

[29] T. D'Orazio, M. Leo, P. Spagnolo, P. L. Mazzeo, N. Mosca, and M. Nitti. "A Visual Tracking Algorithm for Real Time People Detection". In: *Image Analysis for Multimedia Interactive Services, 2007. WIAMIS'07. Eighth International Workshop on*. IEEE, 2007, pp. 34–34. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4279142 (visited on 10/05/2016).

[30] P. L. Mazzeo, P. Spagnolo, M. Leo, and T. D'Orazio. "Visual Players Detection and Tracking in Soccer Matches". In: *Advanced Video and Signal Based Surveillance, 2008. AVSS'08. IEEE Fifth International Conference on*. IEEE, 2008, pp. 326–333. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4730434 (visited on 10/05/2016).

[31] T. D'Orazio, M. Leo, N. Mosca, P. Spagnolo, and P. L. Mazzeo. "A Semi-Automatic System for Ground Truth Generation of Soccer Video Sequences". In: *Advanced Video and Signal Based Surveillance, 2009. AVSS'09. Sixth IEEE International Conference on*. IEEE, 2009, pp. 559–564. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5280004 (visited on 10/05/2016).

[32] G. Zhu, C. Xu, Y. Zhang, Q. Huang, and H. Lu. "Event Tactic Analysis Based on Player and Ball Trajectory in Broadcast Video". In: *Proceedings of the 2008 International Conference on Content-Based Image and Video Retrieval*. ACM, 2008, pp. 515–524. URL: http://dl.acm.org/citation.cfm?id=1386418 (visited on 10/05/2016).

[33] R. M. Barros, M. S. Misuta, R. P. Menezes, P. J. Figueroa, F. A. Moura, S. A. Cunha, R. Anido, and N. J. Leite. "Analysis of the Distances Covered by First Division Brazilian Soccer Players Obtained with an Automatic Tracking Method". In: *Journal of Sports Science and Medicine* (2007), pp. 233–242. URL: http://repositorio.unesp.br/handle/11449/69706 (visited on 10/05/2016).

[34] C.-H. Kang, J.-R. Hwang, and K.-J. Li. "Trajectory Analysis for Soccer Players". In: *Sixth IEEE International Conference on Data Mining-Workshops (ICDMW'06)*. IEEE, 2006, pp. 377–381. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4063657 (visited on 10/05/2016).

[35] P.-S. Tsai, T. Meijome, and P. G. Austin. "Scout: A Game Speed Analysis and Tracking System". In: *Machine Vision and Applications* 18.5 (2007), pp. 289–299. URL: http://link.springer.com/article/10.1007/s00138-006-0058-7 (visited on 10/05/2016).

[36] M. Beetz, B. Kirchlechner, and M. Lames. "Computerized Real-Time Analysis of Football Games". In: *IEEE pervasive computing* 4.3 (2005), pp. 33–39. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1495388 (visited on 10/05/2016).

[37] M. Leo, T. D'Orazio, and M. Trivedi. "A Multi Camera System for Soccer Player Performance Evaluation". In: *Distributed Smart Cameras, 2009. ICDSC 2009. Third ACM/IEEE International Conference on*. IEEE, 2009, pp. 1–8. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5289343 (visited on 10/05/2016).

[38] R. Marani, V. Renò, M. Nitti, T. D'Orazio, and E. Stella. "A Compact 3D Omnidirectional Range Sensor of High Resolution for Robust Reconstruction of Environments". In: *Sensors* 15.2 (2015), pp. 2283–2308. URL: http://www.mdpi.com/1424-8220/15/2/2283/htm (visited on 10/24/2016).

[39] R. Marani, V. Renò, M. Nitti, T. D'Orazio, and E. Stella. "A Modified Iterative Closest Point Algorithm for 3D Point Cloud Registration". In: *Computer-Aided Civil and Infrastructure Engineering* (2016). URL: http://onlinelibrary.wiley.com/doi/10.1111/mice.12184/pdf (visited on 10/24/2016).

[40] V. Renò, R. Marani, T. D'Orazio, E. Stella, and M. Nitti. "An Adaptive Parallel Background Model for High-Throughput Video Applications and Smart Cameras Embedding". In: *Proceedings of the International Conference on Distributed Smart Cameras*. ICDSC '14. New York, NY, USA: ACM, 2014, 30:1–30:6. ISBN: 978-1-4503-2925-5. DOI: 10.1145/2659021.2659059.

[41] V. Renó, R. Marani, N. Mosca, M. Nitti, T. D'Orazio, and E. Stella. "A Likelihood-Based Background Model for Real Time Processing of Color Filter Array Videos". In: *New Trends in Image Analysis and Processing–ICIAP 2015 Workshops*. Springer, 2015, pp. 218–225.

[42] V. Reno, N. Mosca, M. Nitti, T. D'Orazio, D. Campagnoli, A. Prati, and E. Stella. "Tennis Player Segmentation for Semantic Behavior Analysis". In: *Proceedings of the IEEE International Conference on Computer Vision Workshops*. 2015, pp. 1–8.

[43] V. Renò, N. Mosca, M. Nitti, C. Guaragnella, T. D'Orazio, and E. Stella. "Real-Time Tracking of a Tennis Ball by Combining 3D Data and Domain Knowledge - International Conference on Technology and Innovation in Sports, Health and Wellbeing December 1-3, 2016 - UTAD, Vila Real, Portugal - In Press". In: ().

[44] "System for the Automated Analisys of a Sporting Match". Pat. PCT/IB2016/051883. D. Campagnoli, A. Prati, E. Stella, N. Mosca, V. Renò, and M. Nitti. 2016.

[45] V. Renò, N. Mosca, M. Nitti, T. D'Orazio, C. Guaragnella, D. Campagnoli, A. Prati, and E. Stella. "A technology platform for automatic high-level tennis game analysis". In: *Computer Vision and Image Understanding* (2017).

[46] W. Song, K. Cho, K. Um, C. S. Won, and S. Sim. "Complete Scene Recovery and Terrain Classification in Textured Terrain Meshes". In: *Sensors* 12.8 (2012), pp. 11221–11237. URL: http://www.mdpi.com/1424-8220/12/8/11221/htm (visited on 10/10/2016).

[47] A. Torres-González, J. R. Martinez-de Dios, and A. Ollero. "An Adaptive Scheme for Robot Localization and Mapping with Dynamically Configurable Inter-Beacon Range Measurements". In: *Sensors* 14.5 (2014), pp. 7684–7710. URL: http://www.mdpi.com/1424-8220/14/5/7684/htm (visited on 10/10/2016).

[48] G. N. DeSouza and A. C. Kak. "Vision for Mobile Robot Navigation: A Survey". In: *IEEE transactions on pattern analysis and machine intelligence* 24.2 (2002), pp. 237–267. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=982903 (visited on 10/10/2016).

[49] C. S. Andersen, C. B. Madsen, J. J. Sorensen, N. O. Kirkeby, J. P. Jones, and H. I. Christensen. "Navigation Using Range Images on a Mobile Robot". In: *Robotics and Autonomous Systems* 10.2 (1992), pp. 147–160. URL: http://www.sciencedirect.com/science/article/pii/092188909290023R (visited on 10/10/2016).

[50] Q. Li, L. Zhang, Q. Mao, Q. Zou, P. Zhang, S. Feng, and W. Ochieng. "Motion Field Estimation for a Dynamic Scene Using a 3D LiDAR". In: *Sensors* 14.9 (2014), pp. 16672–16691. URL: http://www.mdpi.com/1424-8220/14/9/16672/htm (visited on 10/10/2016).

[51] R. Marani, M. Nitti, G. Cicirelli, T. D'Orazio, and E. Stella. "High-Resolution Laser Scanning for Three-Dimensional Inspection of Drilling Tools". In: *Advances in Mechanical Engineering* 5 (2013), p. 620786. URL: http://ade.sagepub.com/content/5/620786.full (visited on 10/10/2016).

[52] R. Marani, G. Roselli, M. Nitti, G. Cicirelli, T. D'Orazio, and E. Stella. "A 3D Vision System for High Resolution Surface Reconstruction". In: *Sensing Technology (ICST), 2013 Seventh International Conference on*. IEEE, 2013, pp. 157–162. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6727634 (visited on 10/10/2016).

[53] C. H. Kau, S. Richmond, A. I. Zhurov, J. Knox, I. Chestnutt, F. Hartles, and R. Playle. "Reliability of Measuring Facial Morphology with a 3-Dimensional Laser Scanning System". In: *American Journal of Orthodontics and Dentofacial Orthopedics* 128.4 (2005), pp. 424–430. URL: http://www.sciencedirect.com/science/article/pii/S0889540605006761 (visited on 10/10/2016).

[54] S. C. Aung, R. C. K. Ngim, and S. T. Lee. "Evaluation of the Laser Scanner as a Surface Measuring Tool and Its Accuracy Compared with Direct Facial Anthropometric Measurements". In: *British journal of plastic surgery* 48.8 (1995), pp. 551–558. URL: http://www.sciencedirect.com/science/article/pii/0007122695900438 (visited on 10/10/2016).

[55] C. Bellasio, J. Olejníčková, R. Tesař, D. Šebela, and L. Nedbal. "Computer Reconstruction of Plant Growth and Chlorophyll Fluorescence Emission in Three Spatial Dimensions". In: *Sensors* 12.1 (2012), pp. 1052–1071. URL: http://www.mdpi.com/1424-8220/12/1/1052/htm (visited on 10/10/2016).

[56] S. Barone, A. Paoli, and A. V. Razionale. "3D Reconstruction and Restoration Monitoring of Sculptural Artworks by a Multi-Sensor Framework". In: *Sensors* 12.12 (2012), pp. 16785–16801. URL: http://www.mdpi.com/1424-8220/12/12/16785/htm (visited on 10/10/2016).

[57] M. Kampel and R. Sablatnig. "Rule Based System for Archaeological Pottery Classification". In: *Pattern Recognition Letters* 28.6 (2007), pp. 740–747. URL: http://www.sciencedirect.com/science/article/pii/S0167865506002030 (visited on 10/10/2016).

[58] M. Levoy et al. "The Digital Michelangelo Project: 3D Scanning of Large Statues". In: *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*. ACM Press/Addison-Wesley Publishing Co., 2000, pp. 131–144. URL: http://dl.acm.org/citation.cfm?id=344849 (visited on 10/10/2016).

[59] H. Surmann, A. Nüchter, and J. Hertzberg. "An Autonomous Mobile Robot with a 3D Laser Range Finder for 3D Exploration and Digitalization of Indoor Environments". In: *Robotics and Autonomous Systems* 45.3 (2003), pp. 181–198. URL: http://www.sciencedirect.com/science/article/pii/S0921889003001556 (visited on 10/10/2016).

[60] S. Son, H. Park, and K. H. Lee. "Automated Laser Scanning System for Reverse Engineering and Inspection". In: *International Journal of Machine Tools and Manufacture* 42.8 (2002), pp. 889–897. URL: http://www.sciencedirect.com/science/article/pii/S0890695502000305 (visited on 10/10/2016).

[61] M. J. Milroy, D. J. Weir, C. Bradley, and G. W. Vickers. "Reverse Engineering Employing a 3D Laser Scanner: A Case Study". In: *The International Journal of Advanced Manufacturing Technology* 12.2 (1996), pp. 111–121. URL: http://link.springer.com/article/10.1007/BF01178951 (visited on 10/10/2016).

[62] *RIEGL - Terrestrial Scanning*. URL: http://www.riegl.com/nc/products/terrestrial-scanning (visited on 10/10/2016).

[63] *ILRIS Terrestrial Laser Scanner*. URL: http://www.teledyneoptech.com/wp-content/uploads/ILRIS-Spec-Sheet-140730-WEB.pdf (visited on 10/04/2016).

[64] *Bumblebee2 FireWire Stereo Vision Camera Systems*. URL: https://www.ptgrey.com//bumblebee2-firewire-stereo-vision-camera-systems (visited on 10/10/2016).

[65] *Kinect - Windows App Development*. URL: https://developer.microsoft.com/en-us/windows/kinect (visited on 10/10/2016).

[66] D. Marr and T. Poggio. "A Computational Theory of Human Stereo Vision." In: *Proceedings of the Royal Society of London. Series B, Biological sciences* 204.1156 (1979), p. 301. PMID: 37518.

[67] *Bumblebee XB3 FireWire Stereo Vision Camera Systems*. URL: https://www.ptgrey.com/bumblebee-xb3-1394b-stereo-vision-camera-systems-2 (visited on 10/10/2016).

[68] *SILICON VIDEO® 2KS*. URL: http://www.epixinc.com/products/sv2ks.htm (visited on 10/10/2016).

[69] M. Bertozzi, A. Broggi, A. Coati, and R. I. Fedriga. "A 13,000 Km Intercontinental Trip with Driverless Vehicles: The VIAC Experiment". In: *IEEE Intelligent Transportation Systems Magazine* 5.1 (2013), pp. 28–41. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6420052 (visited on 10/10/2016).

[70] M. Bertozzi and A. Broggi. "GOLD: A Parallel Real-Time Stereo Vision System for Generic Obstacle and Lane Detection". In: *IEEE transactions on image processing* 7.1 (1998), pp. 62–81. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=650851 (visited on 10/10/2016).

[71] *3D Laser Scanners / SICK*. URL: https://www.sick.com/de/en/product-portfolio/detection-and-ranging-solutions/3d-laser-scanners/c/g282752 (visited on 10/10/2016).

[72] *Photo Sensor — PRODUCTS — HOKUYO AUTOMATIC CO.,LTD.* URL: http://www.hokuyo-aut.jp/02sensor/index.html (visited on 10/10/2016).

[73] *Product Overview – FOTONIC*. URL: http://www.fotonic.com/product-overview/ (visited on 10/10/2016).

[74] *SwissRanger*. URL: http://hptg.com/industrial/ (visited on 10/10/2016).

[75] Z. Xu, L. Wu, Y. Shen, F. Li, Q. Wang, and R. Wang. "Tridimensional Reconstruction Applied to Cultural Heritage with the Use of Camera-Equipped UAV and Terrestrial Laser Scanner". In: *Remote Sensing* 6.11 (2014), pp. 10413–10434. URL: http://www.mdpi.com/2072-4292/6/11/10413/htm (visited on 10/10/2016).

[76] G. Teza, A. Galgaro, N. Zaltron, and R. Genevois. "Terrestrial Laser Scanner to Detect Landslide Displacement Fields: A New Approach". In: *International Journal of Remote Sensing* 28.16 (2007), pp. 3425–3446. URL: http://www.tandfonline.com/doi/abs/10.1080/01431160601024234 (visited on 10/10/2016).

[77] *G2 Series - ShapeDrive*. URL: http://www.shape-drive.com/index.php/g2.html (visited on 10/10/2016).

[78] J. C. Pedraza-Ortega, E. Efren Gorrostieta-Hurtado, M. Delgado-Rosas, S. L. Canchola-Magdaleno, J. M. Ramos-Arreguin, M. A. Aceves Fernandez, and A. Sotomayor-Olmedo. "A 3D Sensor Based on a Profilometrical Approach". In: *Sensors* 9.12 (2009), pp. 10326–10340. URL: http://www.mdpi.com/1424-8220/9/12/10326/htm (visited on 10/10/2016).

[79] *Kinect — Xbox 360*. URL: http://www.xbox.com/en-US/xbox-360/accessories/kinect (visited on 10/10/2016).

[80] J.-H. Wu, C.-C. Pen, and J.-A. Jiang. "Applications of the Integrated High-Performance Cmos Image Sensor to Range Finders—from Optical Triangulation to the Automotive Field". In: *Sensors* 8.3 (2008), pp. 1719–1739. URL: http://www.mdpi.com/1424-8220/8/3/1719/htm (visited on 10/10/2016).

[81] X. Ying, K. Peng, R. Ren, and H. Zha. "Geometric Properties of Multiple Reflections in Catadioptric Camera with Two Planar Mirrors". In: *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 1126–1132. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5540088 (visited on 10/10/2016).

[82] S.-h Chang and others. "Fundamental Matrix of Planar Catadioptric Stereo Systems". In: *IET Computer Vision* 4.2 (2010), pp. 85–104. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5475470 (visited on 10/10/2016).

[83] R. A. Hicks and R. Bajcsy. "Reflective Surfaces as Computational Sensors". In: *Image and Vision Computing* 19.11 (2001), pp. 773–777. URL: http://www.sciencedirect.com/science/article/pii/S0262885600001049 (visited on 10/10/2016).

[84] S. S. Deshpande. "Improved Floodplain Delineation Method Using High-Density LiDAR Data". In: *Computer-Aided Civil and Infrastructure Engineering* 28.1 (2013), pp. 68–79. URL: http://onlinelibrary.wiley.com/doi/10.1111/j.1467-8667.2012.00774.x/full (visited on 10/11/2016).

[85] M. Jaboyedoff, T. Oppikofer, A. Abellán, M.-H. Derron, A. Loye, R. Metzger, and A. Pedrazzini. "Use of LIDAR in Landslide Investigations: A Review". In: *Natural hazards* 61.1 (2012), pp. 5–28. URL: http://link.springer.com/article/10.1007/s11069-010-9634-2 (visited on 10/11/2016).

[86] H. Cai and W. Rasdorf. "Modeling Road Centerlines and Predicting Lengths in 3-D Using LIDAR Point Cloud and Planimetric Road Centerline Data". In: *Computer-Aided Civil and Infrastructure Engineering* 23.3 (2008), pp. 157–173. URL: http://onlinelibrary.wiley.com/doi/10.1111/j.1467-8667.2008.00518.x/abstract (visited on 10/11/2016).

[87] A. Cord and S. Chambon. "Automatic Road Defect Detection by Textural Pattern Recognition Based on AdaBoost". In: *Computer-Aided Civil and Infrastructure Engineering* 27.4 (2012), pp. 244–259. URL: http://onlinelibrary.wiley.com/doi/10.1111/j.1467-8667.2011.00736.x/full (visited on 10/11/2016).

[88] F.-A. Moreno, J. Gonzalez-Jimenez, J.-L. Blanco, and A. Esteban. "An Instrumented Vehicle for Efficient and Accurate 3D Mapping of Roads". In: *Computer-Aided Civil and Infrastructure Engineering* 28.6 (2013), pp. 403–419. URL: http://onlinelibrary.wiley.com/doi/10.1111/mice.12006/full (visited on 10/11/2016).

[89] T. Nishikawa, J. Yoshida, T. Sugiyama, and Y. Fujino. "Concrete Crack Detection by Multiple Sequential Image Filtering". In: *Computer-Aided Civil and Infrastructure Engineering* 27.1 (2012), pp. 29–47. URL: http://onlinelibrary.wiley.com/doi/10.1111/j.1467-8667.2011.00716.x/full (visited on 10/11/2016).

[90] H. S. Park, H. M. Lee, H. Adeli, and I. Lee. "A New Approach for Health Monitoring of Structures: Terrestrial Laser Scanning". In: *Computer-Aided Civil and Infrastructure Engineering* 22.1 (2007), pp. 19–30. URL: http://onlinelibrary.wiley.com/doi/10.1111/j.1467-8667.2006.00466.x/full (visited on 10/11/2016).

[91] S. W. Park, H. S. Park, J. H. Kim, and H. Adeli. "3D Displacement Measurement Model for Health Monitoring of Structures Using a Motion Capture System". In: *Measurement* 59 (2015), pp. 352–362. URL: http://www.sciencedirect.com/science/article/pii/S0263224114004436 (visited on 10/11/2016).

[92] L. Truong-Hong, D. F. Laefer, T. Hinks, and H. Carr. "Combining an Angle Criterion with Voxelization and the Flying Voxel Method in Reconstructing Building Models from LiDAR Data". In: *Computer-Aided Civil and Infrastructure Engineering* 28.2 (2013), pp. 112–129. URL: http://onlinelibrary.wiley.com/doi/10.1111/j.1467-8667.2012.00761.x/full (visited on 10/11/2016).

[93] S. B. Walsh, D. J. Borello, B. Guldur, and J. F. Hajjar. "Data Processing of Point Clouds for Object Detection for Structural Engineering Applications". In: *Computer-Aided Civil and Infrastructure Engineering* 28.7 (2013), pp. 495–508. URL: http://onlinelibrary.wiley.com/doi/10.1111/mice.12016/full (visited on 10/11/2016).

[94] C. Zhang and A. Elaksher. "An Unmanned Aerial Vehicle-Based Imaging System for 3D Measurement of Unpaved Road Surface Distresses1". In: *Computer-Aided Civil and Infrastructure Engineering* 27.2 (2012), pp. 118–129. URL: http://onlinelibrary.wiley.com/doi/10.1111/j.1467-8667.2011.00727.x/full (visited on 10/11/2016).

[95] A. Diosi and L. Kleeman. "Laser Scan Matching in Polar Coordinates with Application to SLAM". In: *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2005, pp. 3317–3322. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1545181 (visited on 10/11/2016).

[96] D. Holz and S. Behnke. "Sancta Simplicitas-on the Efficiency and Achievable Results of SLAM Using ICP-Based Incremental Registration". In: *Robotics and Automation (ICRA), 2010 IEEE International Conference on*. IEEE, 2010, pp. 1380–1387. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5509918 (visited on 10/11/2016).

[97] J. Röwekämper, C. Sprunk, G. D. Tipaldi, C. Stachniss, P. Pfaff, and W. Burgard. "On the Position Accuracy of Mobile Robot Localization Based on Particle Filters Combined with Scan Matching". In: *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 3158–3164. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6385988 (visited on 10/11/2016).

[98] D. Holz and S. Behnke. "Registration of Non-Uniform Density 3D Point Clouds Using Approximate Surface Reconstruction". In: *ISR/Robotik 2014; 41st International Symposium on Robotics; Proceedings of*. VDE, 2014, pp. 1–7. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6840170 (visited on 10/11/2016).

[99] N. Pfeifer and J. Böhm. "Early Stages of LiDAR Data Processing". In: *Advances in Photogrammetry, Remote Sensing and Spatial Information Sciences: 2008 ISPRS Congress Book, Li, Z.* 2008, pp. 169–184. URL: https://books.google.it/books?hl=it&lr=&id=-fQM3buoejcC&oi=fnd&pg=PA169&dq=Early+stages+of+LiDAR+data+processin&ots=AYQj6eUie6&sig=r4lNmmBtGVU7ccmP9Z6l05yY6o0 (visited on 10/11/2016).

[100] J.-Y. Han, J. Guo, and Y.-S. Jiang. "Monitoring Tunnel Deformations by Means of Multi-Epoch Dispersed 3D LiDAR Point Clouds: An Improved Approach". In: *Tunnelling and Underground Space Technology* 38 (2013), pp. 385–389. URL: http://www.sciencedirect.com/science/article/pii/S0886779813001181 (visited on 10/11/2016).

[101] Z. Kang, L. Tuo, and S. Zlatanova. "Continuously Deformation Monitoring of Subway Tunnel Based on Terrestrial Point Clouds". In: *XXII ISPRS Congress, Commission V, Melbourne, Australia, 25 August-1 September 2012; IAPRS XXXIX-B5, 2012*. International Society for Photogrammetry and Remote Sensing (ISPRS), 2012. URL: http://repository.tudelft.nl/view/ir/uuid:3fb4f2cf-29d7-4beb-a494-5dbaefad27c1/ (visited on 10/11/2016).

[102] M. Scaioni, L. Barazzetti, A. Giussani, M. Previtali, F. Roncoroni, and M. I. Alba. "Photogrammetric Techniques for Monitoring Tunnel Deformation". In: *Earth Science Informatics* 7.2 (2014), pp. 83–95. URL: http://link.springer.com/article/10.1007/s12145-014-0152-8 (visited on 10/11/2016).

[103] J. Billingsley, B. Grinstead, S. Sukumar, D. Page, A. Koschan, D. Gorsich, and M. A. Abidi. "Mobile Scanning System for the Fast Digitization of Existing Roadways and Structures". In: *Sensor Review* 26.4 (2006), pp. 283–289. URL: http://www.emeraldinsight.com/doi/abs/10.1108/02602280610691999 (visited on 10/11/2016).

[104] N. E. Cazzaniga, G. Forlani, and R. Roncella. "Improving the Reliability of a GPS/INS Navigation Solution for MM Vehicles by Photogrammetry". In: *5-Th International Symposium on Mobile Mapping Technology, Padova*. Citeseer, 2007. URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.491.7181&rep=rep1&type=pdf (visited on 10/11/2016).

[105] P. J. Besl and N. D. McKay. "Method for Registration of 3-D Shapes". In: *Robotics-DL Tentative*. International Society for Optics and Photonics, 1992, pp. 586–606. URL: http://proceedings.spiedigitallibrary.org/proceeding.aspx?articleid=981454 (visited on 10/11/2016).

[106] Y. Chen and G. Medioni. "Object Modelling by Registration of Multiple Range Images". In: *Image and vision computing* 10.3 (1992), pp. 145–155. URL: http://www.sciencedirect.com/science/article/pii/026288569290066C (visited on 10/11/2016).

[107] G. Blais and M. D. Levine. "Registering Multiview Range Data to Create 3D Computer Objects". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 17.8 (1995), pp. 820–824. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=400574 (visited on 10/11/2016).

[108] C. Dorai, G. Wang, A. K. Jain, and C. Mercer. "From Images to Models: Automatic 3D Object Model Construction from Multiple Views". In: *Pattern Recognition, 1996., Proceedings of the 13th International Conference on*. Vol. 1. IEEE, 1996, pp. 770–774. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=546128 (visited on 10/11/2016).

[109] D. Akca. "Registration of Point Clouds Using Range and Intensity Information". In: *The International Workshop on Recording, Modeling and Visualization of Cultural Heritage*. 2005, pp. 115–126. URL: https://books.google.it/books?hl=it&lr=&id=CPOGPVQHcO8C&oi=fnd&pg=PA115&dq=Registration+of+point+clouds+using+range+and+intensity+in-formation&ots=b8i2ACVVxQ&sig=vFpG3yOpoD4yiCZulGN24IYgpmU (visited on 10/11/2016).

[110] J. Gómez-García-Bermejo, E. Zalama, and R. Feliz. "Automated Registration of 3D Scans Using Geometric Features and Normalized Color Data". In: *Computer-Aided Civil and Infrastructure Engineering* 28.2 (2013), pp. 98–111. URL: http://onlinelibrary.wiley.com/doi/10.1111/j.1467-8667.2012.00785.x/full (visited on 10/11/2016).

[111] A. E. Johnson and S. B. Kang. "Registration and Integration of Textured 3D Data". In: *Image and vision computing* 17.2 (1999), pp. 135–147. URL: http://www.sciencedirect.com/science/article/pii/S0262885698001176 (visited on 10/11/2016).

[112] G. C. Sharp, S. W. Lee, and D. K. Wehe. "ICP Registration Using Invariant Features". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24.1 (2002), pp. 90–102. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=982886 (visited on 10/11/2016).

[113] Y. Liu. "Improving ICP with Easy Implementation for Free-Form Surface Matching". In: *Pattern Recognition* 37.2 (2004), pp. 211–226. URL: http://www.sciencedirect.com/science/article/pii/S0031320303002395 (visited on 10/11/2016).

[114] D. F. Huber and M. Hebert. "Fully Automatic Registration of Multiple 3D Data Sets". In: *Image and Vision Computing* 21.7 (2003), pp. 637–650. URL: http://www.sciencedirect.com/science/article/pii/S026288560300060X (visited on 10/11/2016).

[115] A. W. Fitzgibbon. "Robust Registration of 2D and 3D Point Sets". In: *Image and Vision Computing* 21.13 (2003), pp. 1145–1153. URL: http://www.sciencedirect.com/science/article/pii/S0262885603001835 (visited on 10/11/2016).

[116] S. Rusinkiewicz and M. Levoy. "Efficient Variants of the ICP Algorithm". In: *3-D Digital Imaging and Modeling, 2001. Proceedings. Third International Conference on*. IEEE, 2001, pp. 145–152. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=924423 (visited on 10/11/2016).

[117] K.-L. Low. "Linear Least-Squares Optimization for Point-to-Plane Icp Surface Registration". In: *Chapel Hill, University of North Carolina* 4 (2004). URL: https://www.iscs.nus.edu.sg/~lowkl/publications/lowk_point-to-plane_icp_techrep.pdf (visited on 10/11/2016).

[118] W. Xin and J. Pu. "An Improved ICP Algorithm for Point Cloud Registration". In: *Computational and Information Sciences (ICCIS), 2010 International Conference on*. IEEE, 2010, pp. 565–568. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5709064 (visited on 10/11/2016).

[119] D. Haehnel, S. Thrun, and W. Burgard. "An Extension of the ICP Algorithm for Modeling Nonrigid Objects with Mobile Robots". In: *IJCAI*. Vol. 3. 2003, pp. 915–920. URL: http://ais.informatik.uni-freiburg.de/publications/papers/haehnel-ijcai03.pdf (visited on 10/11/2016).

[120] C. Stauffer and W. Grimson. "Learning Patterns of Activity Using Real-Time Tracking". In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 22.8 (Aug. 2000), pp. 747–757. ISSN: 0162-8828. DOI: 10.1109/34.868677.

[121] Z. Zivkovic. "Improved Adaptive Gaussian Mixture Model for Background Subtraction". In: *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*. Vol. 2. Aug. 2004, 28–31 Vol.2. DOI: 10.1109/ICPR.2004.1333992.

[122] N. Oliver, B. Rosario, and A. Pentland. "A Bayesian Computer Vision System for Modeling Human Interactions". In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 22.8 (Aug. 2000), pp. 831–843. ISSN: 0162-8828. DOI: 10.1109/34.868684.

[123] K. Kim, T. H. Chalidabhongse, D. Harwood, and L. Davis. "Real-Time Foreground–Background Segmentation Using Codebook Model". In: *Real-time imaging* 11.3 (2005), pp. 172–185.

[124] J. Rittscher, J. Kato, S. Joga, and A. Blake. "A Probabilistic Background Model for Tracking". In: *Computer Vision ECCV 2000*. Springer, 2000, pp. 336–350.

[125] A. Godbehere, A. Matsukawa, and K. Goldberg. "Visual Tracking of Human Visitors under Variable-Lighting Conditions for a Responsive Audio Art Installation". In: *American Control Conference (ACC), 2012*. June 2012, pp. 4305–4312.

[126] M. Casares, S. Velipasalar, and A. Pinto. "Light-Weight Salient Foreground Detection for Embedded Smart Cameras". In: *Computer Vision and Image Understanding* 114.11 (2010). Special issue on Embedded Vision, pp. 1223 –1237. ISSN: 1077-3142. DOI: http://dx.doi.org/10.1016/j.cviu.2010.03.023. URL: http://www.sciencedirect.com/science/article/pii/S1077314210001001.

[127] X. Yu and D. Farin. "Current and Emerging Topics in Sports Video Processing". In: *Multimedia and Expo, 2005. ICME 2005. IEEE International Conference on*. July 2005, pp. 526–529. DOI: 10.1109/ICME.2005.1521476.

[128] T. D'Orazio, M. Leo, P. Spagnolo, M. Nitti, N. Mosca, and A. Distante. "A Visual System for Real Time Detection of Goal Events during Soccer Matches". In: *Computer Vision and Image Understanding* 113.5 (2009). Computer Vision Based Analysis in Sport Environments, pp. 622 –632. ISSN: 1077-3142. DOI: http://dx.doi.org/10.1016/j.cviu.2008.01.010. URL: http://www.sciencedirect.com/science/article/pii/S1077314208000441.

[129] T. D'Orazio, M. Leo, P. Spagnolo, P. Mazzeo, N. Mosca, M. Nitti, and A. Distante. "An Investigation Into the Feasibility of Real-Time Soccer Offside Detection From a Multiple Camera System". In: *Circuits and Systems for Video Technology, IEEE Transactions on* 19.12 (Dec. 2009), pp. 1804–1818. ISSN: 1051-8215. DOI: 10.1109/TCSVT.2009.2026817.

[130] R. Hamid, R. Kumar, J. Hodgins, and I. Essa. "A Visualization Framework for Team Sports Captured Using Multiple Static Cameras". In: *Computer Vision and Image Understanding* 118.0 (2014), pp. 171 –183. ISSN: 1077-3142. DOI: http://dx.doi.org/10.1016/j.cviu.2013.09.006. URL: http://www.sciencedirect.com/science/article/pii/S1077314213001768.

[131] T. Michelsen, M. Brand, C. Cordes, and H.-J. Appelrath. "Herakles: Real-Time Sport Analysis Using a Distributed Data Stream Management System". In: *Proceedings of the 9th ACM International Conference on Distributed Event-Based Systems*. DEBS '15. Oslo, Norway: ACM, 2015, pp. 356–359. ISBN: 978-1-4503-3286-6. DOI: 10.1145/2675743.2776775. URL: http://doi.acm.org/10.1145/2675743.2776775.

[132] T. D'Orazio and M. Leo. "A Review of Vision-Based Systems for Soccer Video Analysis". In: *Pattern recognition* 43.8 (2010), pp. 2911–2926.

[133] R. Kapela, A Swietlicka, A. Rybarczyk, K. Kolanowski, and N. O'Connor. "Real-Time Event Classification in Field Sport Videos". In: *Signal Processing: Image Communication* 35 (2015), pp. 25–45.

[134] M. Hughes and I. M. Franks. *Notational Analysis of Sport: Systems for Better Coaching and Performance in Sport*. Psychology Press, 2004.

[135] T. B. Moeslund, G. Thomas, and A. Hilton. *Computer Vision in Sports*. Springer, 2015.

[136] C. Ó. Conaire, P. Kelly, D. Connaghan, and N. E. O'Connor. "Tennissense: A Platform for Extracting Semantic Information from Multi-Camera Tennis Data". In: *Digital Signal Processing, 2009 16th International Conference on*. IEEE, 2009, pp. 1–6.

[137] M. Leo, N. Mosca, P. Mazzeo, M. Nitti, T. D'Orazio, and A Distante. "Real-Time Multiview Analysis of Soccer Matches for Understanding Interactions between Ball and Players". In: *Proceedings of the 2008 International Conference on Content-Based Image and Video Retrieval*. 2008, pp. 525–534.

[138] M. Archana and M. Geetha. "Object Detection and Tracking Based on Trajectory in Broadcast Tennis Video". In: *Procedia Computer Science* 58 (2015), pp. 225–232.

[139] F. Yan, W Christmas, and J. Kittler. "A Tennis Ball Tracking Algorithm for Automatic Annotation of Tennis Match". In: *British Machine Vision Conference*. Vol. 2. 2005, pp. 619–628.

[140] F. Yan, A. Kostin, W. Christmas, and J. Kittler. "A Novel Data Association Algorithm for Object Tracking in Clutter with Application to Tennis Video Analysis". In: *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*. Vol. 1. June 2006, pp. 634–641. DOI: 10.1109/CVPR.2006.36.

[141] F. Yan, W. Christmas, and J. Kittler. "Layered Data Association Using Graph-Theoretic Formulation with Application to Tennis Ball Tracking in Monocular Sequences". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30.10 (Oct. 2008), pp. 1814–1830. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2007.70834.

[142] T. Qazi, P. Mukherjee, S. Srivastava, B. Lall, and N. R. Chauhan. "Automated Ball Tracking in Tennis Videos". In: *2015 Third International Conference on Image Information Processing (ICIIP)*. Dec. 2015, pp. 236–240. DOI: 10.1109/ICIIP.2015.7414772.

[143] Q. Wang, K. Zhang, and D. Wang. "The Trajectory Prediction and Analysis of Spinning Ball for a Table Tennis Robot Application". In: *Cyber Technology in Automation, Control, and Intelligent Systems (CYBER), 2014 IEEE 4th Annual International Conference on*. June 2014, pp. 496–501. DOI: 10.1109/CYBER.2014.6917514.

[144] G. Pingali, A. Opalach, and Y. Jean. "Ball Tracking and Virtual Replays for Innovative Tennis Broadcasts". In: *Pattern Recognition, 2000. Proceedings. 15th International Conference on*. Vol. 4. 2000, 152–156 vol.4. DOI: 10.1109/ICPR.2000.902885.

[145] N. Owens, C. Harris, and C. Stennett. "Hawk-Eye Tennis System". In: *Visual Information Engineering, 2003. VIE 2003. International Conference on*. July 2003, pp. 182–185. DOI: 10.1049/cp:20030517.

[146] "Hawk-Eye Innovations Official Website". In: *http://www.hawkeyeinnovations.co.uk/* ().

[147] "Dartfish". In: *http://www.dartfish.com/en/index.htm* ().

[148] Avenir Sports. "Avenir Sports". In: (2016). URL: http://avenirsports.ie/.

[149] A. Kokaram, N. Rea, R. Dahyot, A. Tekalp, P. Bouthemy, P. Gros, and I. Sezan. "Browsing Sports Video: Trends in Sports-Related Indexing and Retrieval Work". In: *Signal Processing Magazine, IEEE* 23.2 (Mar. 2006), pp. 47–58. ISSN: 1053-5888. DOI: 10.1109/MSP.2006.1621448.

[150] Match Analysis. "Match Analysis". In: (2016). URL: http://matchanalysis.com/.

[151] T. Polk, J. Yang, Y. Hu, and Y. Zhao. "TenniVis: Visualization for Tennis Match Analysis". In: *IEEE Transactions on Visualization and Computer Graphics* 20.12 (2014), pp. 225–232.

[152] "Performa Sports". In: *http://www.performasports.com/* ().

[153] Protracker Tennis. "Protracker Tennis". In: (2015). URL: http://www.fieldtown.co.uk/.

[154] M. Ermes, J. Pärkkä, J. Mäntyjärvi, and I. Korhonen. "Detection of Daily Activities and Sports With Wearable Sensors in Controlled and Uncontrolled Conditions". In: *IEEE Transactions on Information Technology in Biomedicine* 12.1 (Jan. 2008), pp. 20–26. ISSN: 1089-7771. DOI: 10.1109/TITB.2007.899496.

[155] C. Strohrmann, H. Harms, G. Tröster, S. Hensler, and R. Müller. "Out of the Lab and into the Woods: Kinematic Analysis in Running Using Wearable Sensors". In: *Proceedings of the 13th International Conference on Ubiquitous Computing*. ACM, 2011, pp. 119–122.

[156] A. Ahmadi, D. Rowlands, and D. A. James. "Towards a Wearable Device for Skill Assessment and Skill Acquisition of a Tennis Player during the First Serve". In: *Sports Technology* 2 (3-4 2009), pp. 129–136.

[157] D. Connaghan, P. Kelly, and N. E. O'Connor. "Game, Shot and Match: Event-Based Indexing of Tennis". In: *Content-Based Multimedia Indexing (CBMI), 2011 9th International Workshop on*. IEEE, 2011, pp. 97–102.

[158] C. Chen and C. Pomalaza-Ráez. "Monitoring Human Movements at Home Using Wearable Wireless Sensors". In: *Proceedings of the Third International Symposium on Medical Information and Communication Technology*. 2009.

[159] M. Bächlin, K. Förster, and G. Tröster. "SwimMaster: A Wearable Assistant for Swimmer". In: *Proceedings of the 11th International Conference on Ubiquitous Computing*. ACM, 2009, pp. 215–224.

[160] H. Ghasemzadeh, V. Loseu, and R. Jafari. "Wearable Coach for Sport Training: A Quantitative Model to Evaluate Wrist-Rotation in Golf". In: *Journal of Ambient Intelligence and Smart Environments* 1.2 (2009), pp. 173–184.

[161] E. H. Chi. "Introducing Wearable Force Sensors in Martial Arts". In: *Pervasive Computing, IEEE* 4.3 (2005), pp. 47–53.

[162] D. S. Valter, C. Adam, M. Barry, and C. Marco. "Validation of Prozone®: A New Video-Based Performance Analysis System". In: *International Journal of Performance Analysis in Sport* 6.1 (2006), pp. 108–119.

[163] G. Pingali, Y Jean, and I Carlbom. "Lucent Vision: A System for Enhanced Sports Viewing, Volume 1614 of". In: *Lecture Notes in Computer Science* (), pp. 689–696.

[164] T. Bloom and A. P. Bradley. "Player Tracking and Stroke Recognition in Tennis Video". In: *APRS Workshop on Digital Image Computing (WDIC'03)*. Vol. 1. The University of Queensland, 2003, pp. 93–97.

[165] X. Yu, C. Xu, H. W. Leong, Q. Tian, Q. Tang, and K. W. Wan. "Trajectory-Based Ball Detection and Tracking with Applications to Semantic Analysis of Broadcast Soccer Video". In: *Proceedings of the Eleventh ACM International Conference on Multimedia*. ACM, 2003, pp. 11–20.

[166] S. Tamaki and H. Saito. "Reconstruction of 3D Trajectories for Performance Analysis in Table Tennis". In: *Computer Vision and Pattern Recognition Workshops (CVPRW), 2013 IEEE Conference on*. IEEE, 2013, pp. 1019–1026.

[167] A. Poliakov, D. Marraud, L. Reithler, and C. Chatain. "Physics Based 3d Ball Tracking for Tennis Videos". In: *Content-Based Multimedia Indexing (CBMI), 2010 International Workshop on*. IEEE, 2010, pp. 1–6.

[168] *Coherent Inc. CUBE, a High Performance Diode Laser System.* URL: `http://www.coherent.com/products/?1007/CUBE-Lasers` (visited on 10/10/2016).

[169] *Detail*. 2015-05-07T18:30:56+02:00. URL: https://www.alliedvision.com/en/products/cameras/detail/Bonito/CL-400.html (visited on 10/10/2016).

[170] *VS-LTC Series, Telecentric Lens for Line Sensor - VS Technology Corporation*. URL: https://www.vst.co.jp/en/products/machinevision/lenses/line-scan-telecentric-lenses/ (visited on 10/10/2016).

[171] *HALCON – The Power of Machine Vision - MVTec Software GmbH*. URL: http://www.mvtec.com/products/halcon (visited on 10/10/2016).

[172] R. Marani, G. Roselli, M. Nitti, G. Cicirelli, T. D'Orazio, and E. Stella. "Analysis of Indoor Environments by Range Images". In: *Sensing Technology (ICST), 2013 Seventh International Conference on*. IEEE, 2013, pp. 163–168. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6727635 (visited on 10/10/2016).

[173] D. K. Naidu and R. B. Fisher. "A Comparative Analysis of Algorithms for Determining the Peak Position of a Stripe to Sub-Pixel Accuracy". In: *BMVC91*. Springer, 1991, pp. 217–225. URL: http://link.springer.com/chapter/10.1007/978-1-4471-1921-0_28 (visited on 10/10/2016).

[174] C. Moenning and N. A. Dodgson. "Intrinsic Point Cloud Simplification". In: *Proc. 14th GrahiCon* 14 (2004), p. 23. URL: http://neildodgson.com/pubs/GraphiCon04.pdf (visited on 10/11/2016).

[175] H. Song and H.-Y. Feng. "A Progressive Point Cloud Simplification Algorithm with Preserved Sharp Edge Data". In: *The International Journal of Advanced Manufacturing Technology* 45 (5-6 2009), pp. 583–592. URL: http://link.springer.com/article/10.1007/s00170-009-1980-4 (visited on 10/11/2016).

[176] T. Whelan, L. Ma, E. Bondarev, P. H. N. de With, and J. McDonald. "Incremental and Batch Planar Simplification of Dense Point Cloud Maps". In: *Robotics and Autonomous Systems* 69 (2015), pp. 3–14. URL: http://www.sciencedirect.com/science/article/pii/S0921889014001961 (visited on 10/11/2016).

[177] R. B. Rusu, Z. C. Marton, N. Blodow, M. Dolha, and M. Beetz. "Towards 3D Point Cloud Based Object Maps for Household Environments". In: *Robotics and Autonomous Systems* 56.11 (2008), pp. 927–941. URL: http://www.sciencedirect.com/science/article/pii/S0921889008001140 (visited on 10/11/2016).

[178] U. Ramer. "An Iterative Procedure for the Polygonal Approximation of Plane Curves". In: *Computer graphics and image processing* 1.3 (1972), pp. 244–256. URL: http://www.sciencedirect.com/science/article/pii/S0146664X72800170 (visited on 10/11/2016).

[179] B. K. Horn. *Sequins and Quills-Representations for Surface Topography*. DTIC Document, 1979. URL: http://oai.dtic.mil/oai/oai?verb=getRecord&metadataPrefix=html&identifier=ADA078068 (visited on 10/11/2016).

[180] F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva, and G. Taubin. "The Ball-Pivoting Algorithm for Surface Reconstruction". In: *IEEE transactions on visualization and computer graphics* 5.4 (1999), pp. 349–359. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=817351 (visited on 10/11/2016).

[181] N. Amenta, S. Choi, and R. K. Kolluri. "The Power Crust, Unions of Balls, and the Medial Axis Transform". In: *Computational Geometry* 19.2 (2001), pp. 127–153. URL: http://www.sciencedirect.com/science/article/pii/S0925772101000177 (visited on 10/11/2016).

[182] M. Kazhdan, M. Bolitho, and H. Hoppe. "Poisson Surface Reconstruction". In: *Proceedings of the Fourth Eurographics Symposium on Geometry Processing*. Vol. 7. 2006. URL: http://faculty.cs.tamu.edu/schaefer/teaching/689_Fall2006/poissonrecon.pdf (visited on 10/11/2016).

[183] M. Kazhdan and H. Hoppe. "Screened Poisson Surface Reconstruction". In: *ACM Transactions on Graphics (TOG)* 32.3 (2013), p. 29. URL: http://dl.acm.org/citation.cfm?id=2487237 (visited on 10/11/2016).

[184] Y. Ohtake, A. Belyaev, M. Alexa, G. Turk, and H.-P. Seidel. "Multi-Level Partition of Unity Implicits". In: *ACM SIGGRAPH 2005 Courses*. ACM, 2005, p. 173. URL: http://dl.acm.org/citation.cfm?id=1198649 (visited on 10/11/2016).

[185] N. J. Mitra and A. Nguyen. "Estimating Surface Normals in Noisy Point Cloud Data". In: *Proceedings of the Nineteenth Annual Symposium on Computational Geometry*. ACM, 2003, pp. 322–328. URL: http://dl.acm.org/citation.cfm?id=777840 (visited on 10/11/2016).

[186] F. Pomerleau, F. Colas, R. Siegwart, and S. Magnenat. "Comparing ICP Variants on Real-World Data Sets". In: *Autonomous Robots* 34.3 (2013), pp. 133–148. URL: http://link.springer.com/article/10.1007/s10514-013-9327-2 (visited on 10/11/2016).

[187] N. Gelfand, L. Ikemoto, S. Rusinkiewicz, and M. Levoy. "Geometrically Stable Sampling for the ICP Algorithm". In: *3-D Digital Imaging and Modeling, 2003. 3DIM 2003. Proceedings. Fourth International Conference on*. IEEE, 2003, pp. 260–267. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1240258 (visited on 10/11/2016).

[188] R. B. Rusu and S. Cousins. "3d Is Here: Point Cloud Library (Pcl)". In: *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 1–4. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5980567 (visited on 10/11/2016).

[189] "Processors - Define SSE2, SSE3 and SSE4". In: (). http://www.intel.com/support/it/mt/mt_win.htm.

[190] A. Sobral. "BGSLibrary: An OpenCV C++ Background Subtraction Library". In: *IX Workshop de Visão Computacional (WVC'2013)*. Rio de Janeiro, Brazil, June 2013. URL: http://iris.sel.eesc.usp.br/wvc/Anais_WVC2013/Poster/2/15.pdf.

[191] "Color Imaging Array". Pat. Classificazione Stati Uniti 348/276, 359/891, 348/E09.003, 348/E09.01, 313/371; Classificazione internazionale H04N9/07, H04N9/04, G02B5/20; Classificazione cooperativa H04N9/07, H04N9/045, H01L27/14621; Classificazione Europea H04N9/04B, H04N9/07, H01L27/146A8C. URL: http://www.google.com/patents/US3971065 (visited on 10/23/2016).

[192] G. Qian, S. Sural, Y. Gu, and S. Pramanik. “Similarity Between Euclidean and Cosine Angle Distance for Nearest Neighbor Queries”. In: *Proceedings of the 2004 ACM Symposium on Applied Computing*. SAC ’04. Nicosia, Cyprus: ACM, 2004, pp. 1232–1237. ISBN: 1-58113-812-1. DOI: 10.1145/967900.968151.

[193] C. Ridder, O. Munkelt, and H. Kirchner. “Adaptive Background Estimation and Foreground Detection Using Kalman-Filtering”. In: *Proceedings of International Conference on Recent Advances in Mechatronics*. Citeseer, 1995, pp. 193–199.

[194] “MVTec Halcon”. In: *http://www.halcon.com/* ().