CRANFIELD UNIVERSITY

MARINA MAGNABOSCO

SELF LOCALIZATION AND MAPPING USING OPTICAL AND THERMAL IMAGERY

SCHOOL OF ENGINEERING

MSc by Research THESIS

Academic Year: 2010-11

Supervisor: T. Breckon

February 2011

CRANFIELD UNIVERSITY

SCHOOL OF ENGINEERING

MSc by Research THESIS

Academic Year: 2010-11

MARINA MAGNABOSCO

SELF LOCALIZATION AND MAPPING USING OPTICAL AND THERMAL IMAGERY

Supervisor: T. Breckon

February 2011

This thesis is submitted in partial fulfilment of the requirements
for the degree of Master of Science

Abstract

Given a mobile robot starting from an unknown position in an unknown environment, with the task of explores the surroundings, it has to be able to build an environmental map and localize itself inside that map. Achieving a solution of this problem allows the exploration of area that can be dangerous or inaccessible for humans.

In our implementation we decide to use two primary sensors for the environment exploration: an optical and a thermal camera. Prior work on the combined use of optical and thermal sensors for the Simultaneous Localization And Mapping (SLAM) problem is limited. The innovative aspect of this work is based on this combined use of a secondary thermal camera as an additional visual sensor for navigation under varying environmental conditions. A secondary innovative aspect is that we focus our attention on both cameras, using them as two separate and independent sensors and combine the information in the final stage of environmental mapping.

During the mobile robot navigation the two cameras capture images on the environment and SURF feature points are extracted and matched between successive scenes. Using a prior work on bearing-only SLAM approach as a reference, a feature initialization method is implemented and allows each new good candidate feature (optical or thermal) to be initialized with a sum of Gaussians that represents a set of possible spatial positions of the detected feature. Using successive observations, is possible to estimate the environment coordinates of the feature and adding it to the Extended Kalman Filter (EKF) dynamic state vector. The EKF state vector contains the information about the position of the 6 degree of freedom mobile robot and the environmental landmark coordinates. The update of this information is managed by the EKF algorithm, a statistical method that allows a prediction of the state vector and it updates based on sensor information available.

The final methodology is tested in indoor and outdoor environments with several different light conditions and robot trajectories producing results that are robust in terms of noise in the images and in other sensor data (i.e. encoders and GPS). The use of the thermal camera improves the number of landmarks detected during the navigation adding useful information about the explored area.

Contents

Chapter 1

Introduction

1.1 Motivation

Autonomous vehicles will soon be with us and are currently under development by a range of universities and companies all over the world. The main applications of autonomous vehicles are surveillance and exploration (air, ground, underground and water).

An important application of autonomous vehicles is also the exploration of unknown environments where the human access is not possible or it is dangerous. The combination of exploration of unknown environment and human presence detection can be an interesting novel aspect in the surveillance area [1].

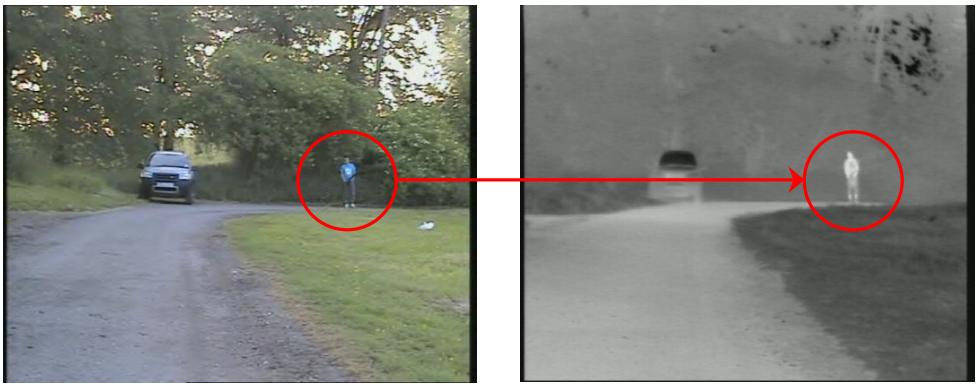

Figure 1.1.1: *Detection of human presence using optical and thermal cameras (Data source: SATURN project, Salisbury Plain, Wiltshire, UK, June 2009).*

In this work by using independent optical and thermal cameras, we develop a system able to explore a small area of the *Cranfield University Campus* and, at the same time, build a map of the environment and self-localize an autonomous vehicle within this map. This type of issue refers to the problem knows as Simultaneous Localization And Mapping (SLAM). Using a thermal camera it allows us to readily detect human presence (e.g. Fig. 1.1.1) and particular thermal characteristics of the explored area (see Fig. 1.1.2). The novel aspect of this project is the combined use of thermal and optical sensing for mobile robot SLAM which is not being carrying out before in the reported literature.

Figure 1.1.2: *Building trade examples.*

Simultaneous Localization And Mapping (SLAM) based on purely optical features is susceptible to changing environmental lighting conditions. Additionally it is somewhat limited for night–based SLAM operations. Combining the use of optical features (SURF [2]) and thermal image features (MSERs [3]) we can overcome these limitations and offer illumination/night – based robust SLAM operation with a possibility to operate during the transition from day to night light without interruption of operation.

1.2 SLAM: State of the Art and Overview

The SLAM problem is a very wide research topic which has become increasingly significant over the last 20 years. Our approach is to use a mobile robot to explore an unknown (or semi-known) environment (here *Cranfield University campus*) and we want to build a map and localize the robot position within this 3D map. Both of these actions form part of the typical SLAM problem previously mentioned.

SLAM has numerous applications both indoor and outdoor. An important application is the mapping of environments where human access is difficult or dangerous. The aims of this type of mapping exploration can be very different and often are related to a specific category of environment. If we think about indoor environment we could find a wide range of household devices as robot vacuum [4] and industry devices where usual applications include welding, painting, assembly, pick and place, packaging and palletizing, product inspection, and testing [4] [5]. Streets mapping, as Google Street View [6], is a new and interesting application for outdoor environments. Others applications are driverless vehicle which will be an aim to improve the safety of the people on the road [7].

A large interest of unmanned vehicles is in military applications where the primary aim of this autonomous robot is to explore, capture surveillance imagery and possibly be equipped to engage ground targets (see Fig. 1.2.1). This type of vehicle is designed for the ground, underground, air and also for underwater environments [8] [9]. Underwater vehicles have also aims of surveillance, maritime situational awareness and also the exploration of dangerous area (e.g. area with mines) [10].

(a) iRobot® Ranger [8] (b) iRobot ®PackBot [8]

Figure 1.2.1: *Examples of an (a) underwater vehicle and (b) bombs disposal robot with manipulator.*

1.2.1 Type of Navigations

As we can expect, for each type of application and such diverse aims we can use a variety of different techniques related to the knowledge we have of the environment. From [11] we can summarize the mobile robot navigation as three principal sub-genres:

- *Map-Based Navigation*: In this case the mobile robot has an *a priori* map, provided by the user, and it is a geometric model or topological map of the environment. This is common in indoor and structured environment where one can build a model or can identify some kind of sensor detected landmarks which can be used by the mobile robot for localization within the environment map.

- *Map-Building-Based Navigation*: A system uses this technique constructs its own map or model of the environment and then uses this for subsequent localization and navigation. This type of navigation refers to the outdoor and unstructured

environment where in general an *a priori* map of landmarks or the general environment is not available.

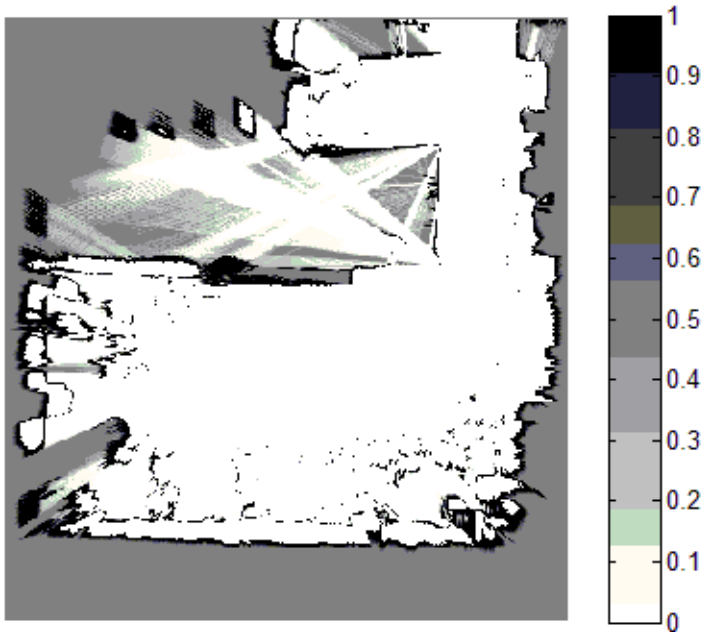

Figure 1.2.2: *Example of a map using a 2D laser, probabilistic approach is used to build an occupancy grid map* [12].

- *Mapless Navigation*: In this last case the system does not use an explicit model and by recognizing or tracking particular objects it is able to generate an environmental map based on these visual observations. Often the main aim is exploration and in general the mobile robot only require to store absolute localization relative to some initial point of reference (lander or spacecraft) [11].

Essentially SLAM is a *Map-Building-Based navigation* where the system has a known kinematic model of the mobile robot. Starting from an unknown position inside the environment, the aim of SLAM is to localize the vehicle within the environment and, at the same time, build and incremental navigation map using the observed landmarks from sensor information.

A useful method to decrease the positioning error of the robot inside the environmental map is the *loop closing method*. Systems that use *Map Navigation* techniques can also take advantage of the SLAM loop closing method. This method is very important in SLAM and consists of re-estimate the map when the robot returns to a previously visited location and, in this case, the robot is able to recognize the SLAM landmarks and increase the accuracy of the overall map. This method permits the construction of map with greater consistency through

reducing the localization and the position errors of the landmarks [13] increasing the robustness of the overall pose estimation within the environment.

1.2.2 Sensors

Once we have identified the type of environment and what type of *a priori* knowledge we have, we can select or identify the sensors to use on our mobile system. In general there are a wide range of devices that can be used for robot navigation and help us to detect the necessary SLAM landmark information. From the literature we can identify the following types of sensors:

- *Laser range finders*: the main components are a laser light and a photo detector that allows to recover information about the distance of the objects in the scene based on the travelling time of the laser light (e.g. LIDAR [14], Millimetre Wave (MMW) Radar [15]), a laser scanner can be 1D, 2D or 3D [16] [17] [13].

- *Single camera*: a 2D image is the output of a single camera and it allows to recover the direction of feature points in the scene, combination of multiple views from several viewpoints is a possible solution to recover depth information [18] [19] [14] [20] [21].

- *Stereo vision*: the combination of multiple 2D pair of images from which 3D seen information can be recover by geometric triangulation [22] [20].

- *Multiple camera rigs*: as for the stereo vision case but extended the principal to multiple geometrically aligned cameras [22].

- *Catadioptric sensors*: consists in a single camera and two fixed conic mirrors and it provides two views of the scene in a single image [22].

- *Odometry and inertial sensor*: an example are wheel encoders [23] [14], they are transducers of position and are widely used in robotics and in autonomous systems;

- *Global Positioning System (GPS)*: is a satellites based navigation system that provides absolute location and time information [22] [23] [15] [14].

- *Sonar sensor:* evaluates the distance of an object by interpreting the echoes from sound waves [17].

The above devices are in common use in addition to a wide variety of other sensors. A general approach is the use of two or more sensors which data are merged by sensor fusion technique [24]. Using different sensors enables to realize a system which is more robust than using a single type of sensor. In sensor fusion one of the main problems is how merge the information from different types of sensors (for a good overview the reader is directed to [24]).

1.2.3 Visual SLAM

In SLAM common sensing devices are single or stereo cameras which are then combined with one or more other sensors such as laser or GPS, etc. In general this approach is referred to as Visual SLAM because the principal information is in the form of a digital image of the environment (i.e. visual signal).

Within the Visual SLAM problem we can establish another two main classes related to the method used for extracting the information from the sensing devices. From [16] we can identify the first method called *Feature-Based Methods*. This class of methods consists of first extracting a sufficient number of features (e.g. points, line, edge, *etc*) and as a second stage, matching them between successive images. The matching stage is the key to all SLAM algorithms and here a lot of attention must be given – erroneous matching means erroneous pose estimation and hence erroneous map information. The second class of methods is *Direct Methods*. The required information or parameters are directly extracted by the pixels intensity value (such as image brightness, brightness – based cross – correlation, *etc*) [25]. This second class of methods is used, especially in urban environments, where the requirement is to identify certain classes of objects from others such as roads from buildings (e.g. road scene understanding [21]).

Both feature detection approach and direct method require a “tool” for extracting the necessary information from the image and in general we have a lot of techniques from which to choose [26]. From the recent literature some of the most popular and recently developed feature detection approaches are Harris corners [27], scene flow [16] [28], Scale-Invariant Feature Transform (SIFT) [29], Shi–Tomasi corners [19], Speeded Up Robust Feature (SURF) [2] and generalized segmentation [3] [21] . In this work we will specifically look at the use of a feature detection based approach based primarily on the use of image feature points.

New techniques (such as SURF [2]) have been developed to extract information from images and several new methods have applied to solve SLAM problems. SLAM is currently a subject of intense research and there is a lot of research work going on in collaboration both with SLAM problem resolution methods and computer vision feature extraction methods [30] [11] [14].

Within this work we specifically look at the use of a Visual SLAM combining the use of optical SURF features [2] with MSER thermal features [3] within the SLAM framework proposed by [31]. This is realized using a mobile robot platform (Section 3.5) and tested over a range of outdoor and indoor environments.

Chapter 2

In this chapter we are going to outline the key elements of the approach we use to solve the SLAM problem.

Literature review

In Visual SLAM (Section 1.2.3) the *Feature-Based method* is mainly used to extract information from an image [16]. This method consists in first extracting a sufficient number of features (e.g. points, line, edge, *etc*) and as second stage matching them, in a robust way, between successive images. The matching stage is the key to all SLAM algorithms and a lot of attention must be paid – wrong matching means wrong pose estimation and hence wrong map information.

To apply the feature based method a feature detection method needs to be used. Most popular and recently developed feature detection approaches are Harris corners [27], scene flow [16] [28], Scale-Invariant Feature Transform (SIFT) [29], Shi-Tomasi corners [19], Speeded Up Robust Feature (SURF) [2] and Maximally Stable Extremal Regions (MSER) feature [3].

2.1 SURF Features

The Speeded Up Robust Feature (SURF) method [2] is a robust image feature point detector and descriptor partially based on the Scale-Invariant Feature Transform (SIFT) method [29]. As in SIFT, the SURF method is scale and rotation invariant and this is essential in our application because our robot platform is moving within its environment.

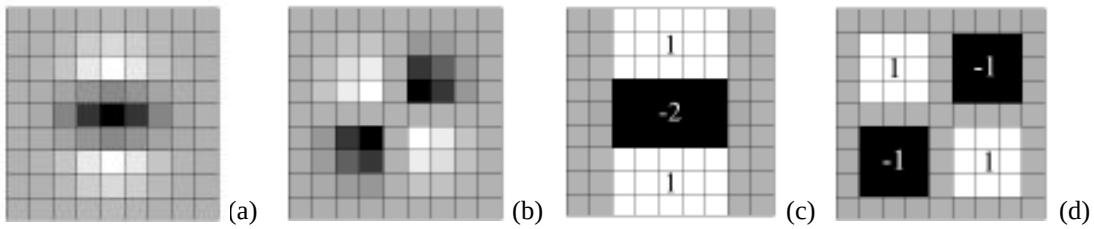

Figure 2.1.1: *The discretised and cropped Gaussian second order partial derivatives in (a) y-direction and (b) xy-direction, and (c)(d) the approximations used in* [2] *(*[2], *Fig. 1, pp.408).*

The first step of the SURF method is the *detection* of characteristic feature point in the image whilst the second step is to assign a unique *descriptor* to each detected point. The descriptor is a parameter vector of the feature point that aims to be unique because it is then used to subsequently match feature points between images. The descriptor has also to be noise robust, invariant to scale and invariant to rotation in order to recover same points in different images.

The *detector* used in SURF [2] is based on the Hessian matrix of the image and that choice is based on its good performance in terms of both computation time and accuracy. Given a point $\boldsymbol{x} = \{x, y\}$ in an image *I*, the Hessian matrix H($\boldsymbol{x}$, σ) in $\boldsymbol{x}$ at scale σ is defined as follows:

$$H(\boldsymbol{x}, \sigma) = \begin{bmatrix} L_{xx}(\boldsymbol{x}, \sigma) & L_{xy}(\boldsymbol{x}, \sigma) \\ L_{xy}(\boldsymbol{x}, \sigma) & L_{yy}(\boldsymbol{x}, \sigma) \end{bmatrix} \tag{2.1}$$

where $L_{xx}(\boldsymbol{x}, \sigma)$ is the convolution of the Gaussian second order derivative $\frac{\delta^2}{\delta x^2} g(\sigma)$ with the image *I* at the point $\boldsymbol{x}$, and similarly for $L_{xy}(\boldsymbol{x}, \sigma)$ and $L_{yy}(\boldsymbol{x}, \sigma)$as described in [2]. The discretised Gaussian second order derivatives are shown in Fig. 2.1.1. To obtain a feature scale invariant, the images are successively smoothed with a Gaussian and then sub-sampled.

As introduced at the beginning of this section, SURF method is based on SIFT feature detector [29] because of the good performance of the descriptor compared to others [32]. The SURF descriptor proposed in [2] is based on similar properties of SIFT descriptor, but less complex and this makes the SURF algorithm faster than the SIFT method. In [2] based on information from a circular region around the interest point, a consistent orientation is set as first step. The SURF descriptor is estimated from a square region aligned to the selected orientation.

SURF is a widely used robust method for features matching with applications in a wide range of computer vision correspondent problems [2] [33]. In practical terms the descriptor is estimated by an *n* by *n* window (square region aligned to the selected orientation). This translates to the dimension of the descriptor vector in the given implementation.

The SURF implementation used in this project [34] allows the selection of the dimension of the descriptor vector as either a *basic descriptor* composed of 64 elements or an *extended descriptor* that uses 128 elements. The difference between the two types of descriptor is just the number of elements used to describe the feature point.

Figure 2.1.2: *Example of SURF features extraction using the basic descriptor.*

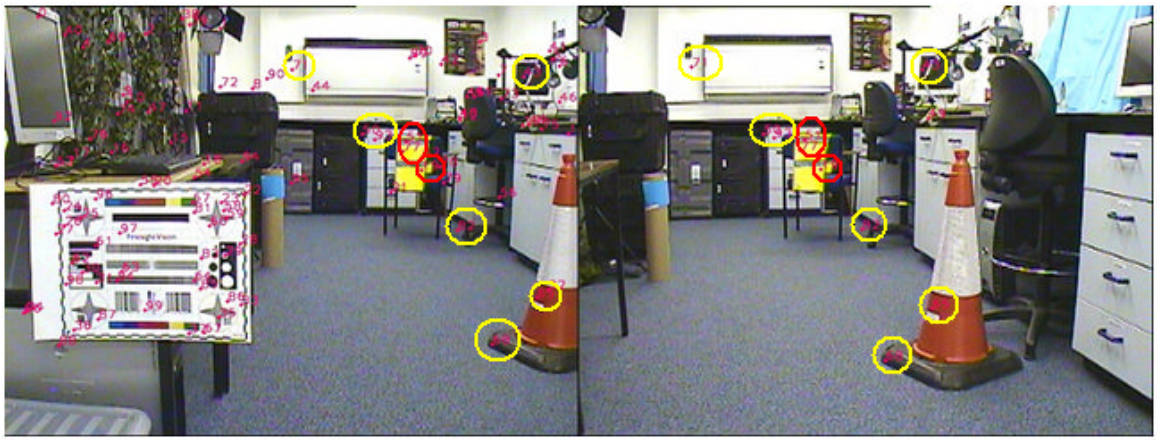

Figure 2.1.3: *Example of SURF features extraction using the extended descriptor.*

In our project, based on the analyzed environments, the *extended descriptor* gives the best performance in terms of matching robustness through different views of the environment as the reader can see form Fig. 2.1.2-2.1.3 where it is shown an example of SURF feature extraction using the two different descriptors available in implementation [34]. The implementation chosen in this project [34] was in general found to be faster in practical performance than others available [33].

2.2 MSERs Features

Maximally Stable Extremal Regions (MSERs) are affinely-invariant stable regions within an image [3] and are regions in the image that are either darker or brighter than the confined regions stable across a range of intensity thresholds. Extremal regions have two useful properties for tracking features from image to image: they are invariant respect to affine transformations and are "*closed under monotonic transformation*" [3]. This makes them a good candidate feature for robust matching during robot navigation. In addition, extremal

regions have all the properties required of a stable local detector. The concept of the MSER features can be explained as follows [3]:

> "*Imagine all possible thresholdings of a gray-level image I. We will refer to the pixels below a threshold as 'black' and to those above or equal to it as 'white'. If we were shown a movie of thresholded images* I_t*, with frame t corresponding to threshold t, we would see first a white image. Subsequently black spots corresponding to local intensity minima will appear and grow. At some point regions corresponding to two local minima will merge. Finally, the last image will be black. The set of all connected components of all frames of the movie is the set of all maximal regions; minimal regions could be obtained by inverting the intensity of I and running the same process.*" ([3], p.386)

Many images have certain regions where the local binarization will be stable over a large range of such thresholds and this are the regions of interest we desiderate since they have the following important properties:

- Invariance to affine transformation of image intensities.
- Stability.
- Multi-scale detection.
- Fast to compute (the set of all extremal regions can be enumerated in O(*n* log log *n*), where *n* is the number of pixels in the image).

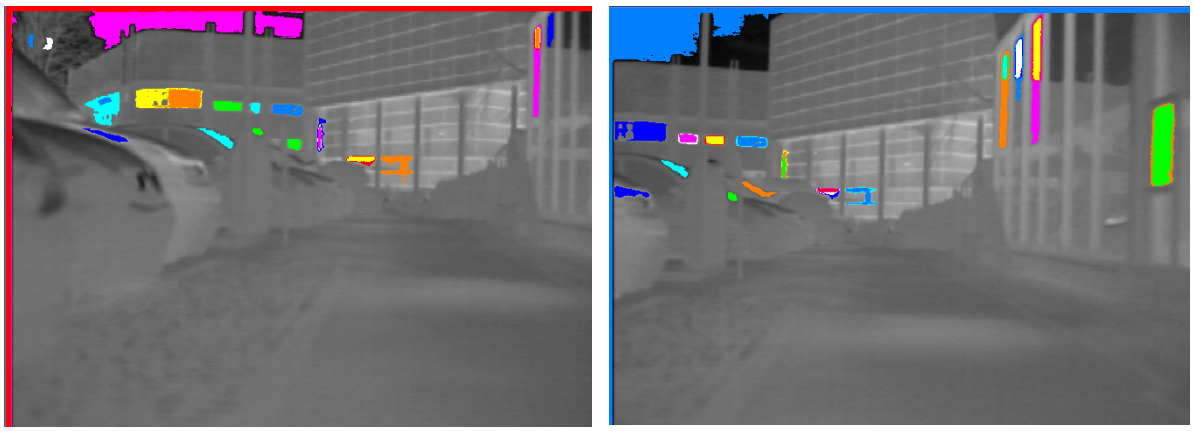

Figure 2.2.1: *Example of MSERs features extraction in successive views.*

As the reader can see form Fig. 2.2.1 the MSERs features are not in a large number and as the field of view of the thermal camera is wide the detected region are very far from the robot and this is not helpful for the navigation purpose. For this reason we decided to concentrate

on the use the SURF feature detector described in Section 2.1 for the thermal images. MSER features were originally considered for this task but as shown in Fig. 2.2.1 the feature detection result was generally unsuitable for SLAM.

2.3 Matching Process

After the feature extraction stage we need to match the feature points between successive images and this is the key aspect in the approach of [31]: by matching features and using successive observations it is possible to recover the 3D coordinates using the information from a single camera.

The output of the feature extraction algorithms (Section 2.1 and 2.2) is a set of points in the image (2D coordinates in pixel) and the associated descriptor. The descriptor is an array of float numbers that characterized the point based on the region around the point (i.e. for the SURF feature point).

Given two sets of points (a first set from past views – i.e. based on the *feature database* – and a second set based on the points extracted in the current view) and the associated descriptors the matching stage is shown in Fig. 2.3.1 and it can be described in three steps as follows:

1. Given a point from the first set (i.e. *query point*) we compute the Euclidean distance (in an n dimensional space – where n is the dimension of the descriptor, see Section 2.1) between its descriptor and all the point descriptors of the second set. The output of this first stage is a subset of points from the second set for each *query point* that has the estimated Euclidean distance, computed between descriptors, under a given threshold τ (see Section 2.3.1).

2. Given the *query point* and the associated subset of matched points (i.e. output of the first stage) the Euclidean distance based on the 2D pixel coordinates of the points is computed between the *query point* and each matched point of the subset. The *query point* is finally associated with the point from the subset of points from the current view that has the minimum Euclidean distances from the *query point* computed using the descriptors as first and the 2D pixel coordinated as second. This choice is based on the assumption that the robot does not move very far between frames.

3. Given a vector of *query points* – from the *feature database* – and the matched points – from the current view – we eliminate the outliers using RANSAC [35] methodology (see Section 2.3.2).

The described approach is used to match *features* through successive images. Unfortunately we cannot apply a parallel identical matching process for the landmarks. The reason of this is because the number of landmarks increases during the image analysis and until the number of stored landmarks is low the matching procedure does not ensure a robust matching (i.e. the RANSAC methodology at least 6 corresponding points are required). The solution adopted in this project consists in adding the 2D pixel coordinates of each landmark to the *query point* vector of feature points. In this way a landmark point is thread as a feature point and matched to a point in the current view using the methodology just described. At the end of the matching process each landmark and its matched point are extracted from the output vector forming two separate matching vectors: one for features and another for landmarks. Details about the techniques used for the matching are described in the following section.

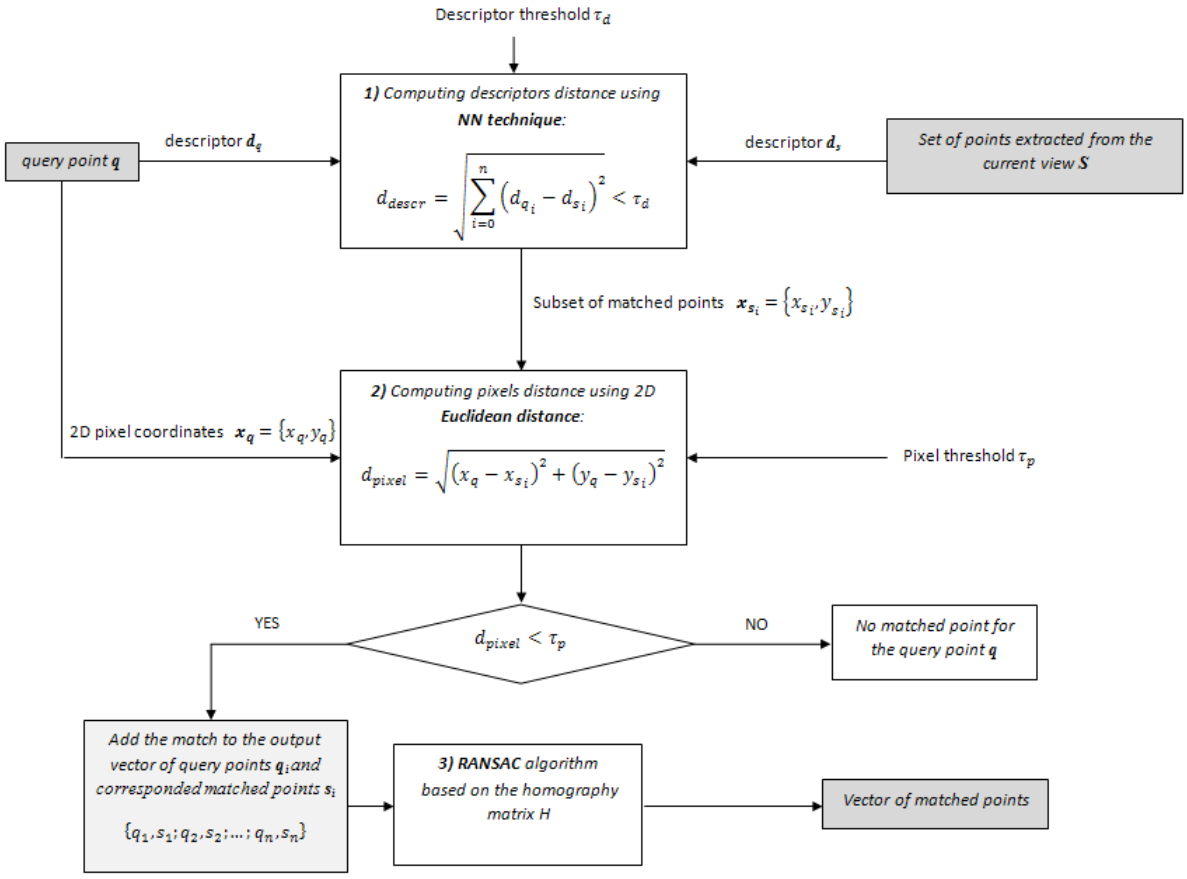

Figure 2.3.1: *Matching process description scheme.*

2.3.1 Nearest Neighbor Technique

Nearest neighbor technique is an optimization problem: given a query point $q \in K$ and a set of points $S \in K$, it finds the closest point in *S* to *q*, where *K* is a *n*-dimensional Euclidean space and, as previously mentioned. The distance between the query point q and the set of points *S* is computing using the *Euclidean distance* in a *n*-dimensional space as follows:

$$d_{qs} = \sqrt{\sum_{i=0}^{n}(q_i - s_i)^2}, \tag{2.2}$$

where q_i is the i^{th} component of the query point *q*, and s_i is the i^{th} component a point $s \in S$.

In our implementation we use the Approximate Nearest Neighbor (ANN) searching described in [36] that provides also a c++ implementation. The output of the algorithm [36], given a query point *q* and a set of points *S*, can be the closest point or a set of closed points.

As we match the feature points using their descriptors, we had noticed that some points association had a very small distance respect to their descriptors, but a large distance in terms of pixels coordinate in the images. To solve this problem and improve the matching process we use a set of 10 closed points as output of the ANN [36] in combination with a second matching process based on the image coordinates of the query point and the subset of points. The subset of 10 points is sorted by the Euclidean distance computed by the ANN [36]. Given the 2D coordinates of the query point $q = \{x_q, y_q\}^T$ (in pixels) and the 2D coordinates of a point $p = \{x_p, y_p\}^T$ from the set of 10 closed points S_{10}, the two points are matched together if:

1. The Euclidean distance between the point descriptors (computed using [36]) is under a certain threshold τ_d;
2. The Euclidean distance (in a 2-dimentional space) between the images coordinates of the points are under a certain pixel threshold τ_c (empirically set to 5 pixels based on the input video frame available – see Section 3.5).

After these two stages a vector of matched points is generated. During the experiment is being seen that, also after these processes, few outliers remain in the set of matches. To overcome this problem a RANSAC technique [35] is added as a final step of the entire matching process and it is introduced in the following section.

2.3.2 RANSAC

RANdom SAmple Consensus (RANSAC [35] is a robust iterative procedure to estimate parameters of a mathematical model from a set of observed data in which outliers are present. The main idea behind the method is to use a minimum amount of data from the set of observed data needed to estimate the model and then count how many of the others data are compatible with the model (i.e. *inliers*), using a certain threshold τ, and how many do not match the model (i.e. *outliers*). This procedure is repeated until the best model is selected and the number of iteration depends on the percentage of outliers in the set of points (i.e. observed data).

The underlying model used in this work is the *homography matrix H* [37] and it can be obtained using four image point correspondences. In this case the *observed data set* is the output vector of point correspondences from the previous matching stage. The relation between the point correspondences and the homography matrix is expressed as:

$$\boldsymbol{x'}_i = sH\boldsymbol{x_i}, \tag{2.3}$$

where $\boldsymbol{x'}_i = \{x'_i, y'_i, 1\}^T$ is a point in the second image (in homogeneous coordinates) and $\boldsymbol{x}_i = \{x_i, y_i, 1\}^T$ is the correspondent point in the first image and *s* is an arbitrary scalar factor.

In this work the homography matrix *H* is chosen as a mathematical model for the set of matched points and the reader can find the details of the algorithm to compute the matrix *H* in [37]. An OpenCV implementation [34] is also available and it compute the homography matrix *H* using the RANSAC [35] procedure using a set of point correspondences as an input. As the homography matrix *H* is estimated, the set of points from the first image (i.e. x_i,, for *i*=1,2,...*n*) is passed through it computing the estimated point correspondences $\tilde{\boldsymbol{x}}'_{i=1,2,\dots n}$:

$$\tilde{\boldsymbol{x}}'_i = H\boldsymbol{x_i}. \tag{2.4}$$

To remove the outliers from the initial vector of matches, the Euclidean distance between the real matched points in the second image and the estimated matched points is computed as follows:

$$d = \sqrt{(\tilde{x}_i - x_i)^2 + (\tilde{y}_i - y_i)^2}. \tag{2.5}$$

If the resulted distance is under a given threshold τ (empirically chosen to be 5 pixels) the points are selected as inliers, otherwise are selected as outliers. After the RANSAC

procedure, the set of correspondences is ready to be use for the successive elaboration as feature initialization and updating process (Section 3.2) or landmarks and robot positions update (Section 2.4).

2.4 Extended Kalman Filter (EKF)

The Kalman filter is a linear estimator that fuses information obtained from a model of a system, a practical observation of that system in operation and an *a priori* knowledge of the system in an optimal manner:

> "*...under a strong but reasonable set of assumptions, it will be possible – given a history of measurements of a system – to build a model for the state of the system that maximizes the a posteriori probability of those previous measurements.*" ([34], p. 350).

In most real case applications the system itself or the observation models do not have a linear behaviour. To overcome this problem the extension of the Kalman filter is used and it is known as the Extended Kalman Filter (EKF) [38].

The *process model* describes the non-linear state of a system at time *k* as:

$$\boldsymbol{x}(k) = F(\boldsymbol{x}(k-1), \boldsymbol{u}(k), k) + \boldsymbol{w}(k), \tag{2.6}$$

where $F(\boldsymbol{x}(k-1), \boldsymbol{u}(k), k)$is the non – linear state transition function at time *k* which is used to compute the current state $\boldsymbol{x}(k)$ based on the previous state $\boldsymbol{x}(k-1)$ and the current control input $\boldsymbol{u}(k)$. The variable $\boldsymbol{w}(k)$ is a random variable called the *process noise* associated with the forces that has a direct effect on the current state of the system. Assuming a Gaussian distribution for the process noise $\boldsymbol{w}(k)$, it is possible to represent $\boldsymbol{w}(k)$ inside the entire process by the covariance matrix $Q(k)$.

For some components of the system state, $\boldsymbol{z}(k)$ measurements are made in a direct or indirect way. The non-linear *observation model* is represented as:

$$\boldsymbol{z}(k) = H\big(\boldsymbol{x}(k)\big) + \boldsymbol{v}(k), \tag{2.7}$$

where $H\big(\boldsymbol{x}(k)\big)$is the measurement function that relates the state with the observations and $\boldsymbol{v}(k)$ is the associated measurement error. The measurement error $\boldsymbol{v}(k)$ it is assumed to have a Gaussian distribution and it is represented by the covariance matrix $R(k)$.

After the definition of the process and observation models it is possible to apply the Extended Kalman Filter equations to recursively estimate the system state in two stages of *Prediction* and *Update* [38] [39]. These stages are discussed in subsequence sections.

2.4.1 Prediction Stage

The prediction stage generates a prediction of the state *x* using the information available up to time *k:*

$$\boldsymbol{x}^{-}(k) = F\big(\boldsymbol{x}(k-1), \boldsymbol{u}(k)\big). \tag{2.8}$$

The uncertainty of the prediction stage is evaluated using the covariance matrix $P^{-}(k)$:

$$P^{-}(k) = \mathrm{J_F}(k) \cdot P(k-1) \cdot \mathrm{J_F^T}(k) + Q(k), \tag{2.9}$$

where $\mathrm{J_F}(k)$ is the Jacobian of the current non-linear state transition function $F(k)$ with respect to the predicted state $\boldsymbol{x}^{-}(k)$.

2.4.2 Update Stage

As soon an observation of the state of the system is available, the state $\boldsymbol{x}^{-}(k)$ it is updated with the measurements obtained at time *k.* The update stage permits to estimate the state vector $\boldsymbol{x}(k)$ that is usually read as the current state given all the information up to time *k.* The update equation is:

$$\boldsymbol{x}(k) = \boldsymbol{x}^{-}(k) + \boldsymbol{K}(k)\big(\boldsymbol{z}(k) - J_H(k) \cdot \boldsymbol{x}^{-}(k)\big), \tag{2.10}$$

where $J_H(k)$ is the Jacobian of the observation function $H(k)$ with respect to the estimated state $\boldsymbol{x}(k)$ and $\big(\boldsymbol{z}(k) - J_H(k) \cdot \boldsymbol{x}^{-}(k)\big)$ represents the non-linear innovation. The variable $\boldsymbol{K}(k)$ is called Kalman gain [38] [39] and is estimated as follows:

$$\boldsymbol{K}(k) = P^{-}(k) \cdot J_H^T(k) \cdot S^{-1}(k), \tag{2.11}$$

$$S(k) = J_H(k) \cdot P^{-}(k) \cdot J_H^T(k) + R(k). \tag{2.12}$$

The updated covariance matrix is:

$$P(k) = P^{-}(k) - K(k) \cdot S(k) \cdot K^T(k). \tag{2.13}$$

2.5 System observations

To estimate the robot positions it necessary to have some measurements of its displacement during the navigation. The main sensors used in autonomous vehicle are encoders that are usually already equipped in the mobile vehicles. The second well known positioning sensor used is the Global Positioning System (GPS). In this project we use integrate the observation data from wheels encoders and a GPS receiver.

2.5.1 Encoders data

The easiest way to have displacement measurements is to use the encoders that are usually already equipped in the robot wheels and they give information about the position of the robot (i.e. *X* and *Y* values) in absolute coordinates and the orientation (i.e. θ angle around the *z* axis). The inconvenient of using the encoders is that the measurements error is increasing with the time so they are not a good method of sensing for long duration navigation due to accumulative error.

As soon the encoder data from the encoders are available they are used to update the system state vector during the *Update Stage* of the extended Kalman filter [38]. The measurements vector is composed as follows:

$$z(k) = \{X_{cmR} \quad Y_{cmR} \quad \theta_{cmR}\}^T, \tag{2.14}$$

where X_{cmR} is the displacement of the mass centre of the robot along the *x* axis of the global reference frame of the robot (i.e. first reference frame of the robot; it is coincident with the global coordinate system up to a constant displacement), Y_{cmR} is the displacement of the mass centre of the robot along the *y* axis and θ_{cmR} is the rotation respect to the *z* axis respect to the same coordinate system. In order to be able to use these measurements we have to transform them from the robot frame to the global frame that refers to the image reference frame. Given the measurement vector $z(k)$ we can express it in the global coordinate system as follows:

$$\begin{Bmatrix} X \\ Y \end{Bmatrix} = \begin{Bmatrix} X_{cmR} \\ Y_{cmR} \end{Bmatrix} - \begin{Bmatrix} x_{cam} \\ y_{cam} \end{Bmatrix} + R_x(\gamma) \cdot R_y(\beta) \cdot R_z(\alpha) \cdot \begin{Bmatrix} x_{cam} \\ y_{cam} \\ z_{cam} \end{Bmatrix} and\ \theta = \theta_{cmR}, \tag{2.15}$$

where $\{x_{cam} \quad y_{cam} \quad z_{cam}\}^T$ is the translation vector that represents the position of the centre of the optical camera respect to the mass centre of the robot and $R_x(\gamma) \cdot R_y(\beta) \cdot R_z(\alpha)$

represents the rotation matrices related to the current orientation of the camera respect to the global reference frame. As the encoder's data are expressed in the image reference frame it is possible to insert them in the *Update Stage* of the Kalman filter.

2.5.2 Global Positioning System data

Additional robot position data can be recovered from a GPS sensor. As previously mentioned, our system is also equipped with a GPS receiver connected to the laptop of the mobile robot. The information available from the GPS device used is in the National Marine Electronics Association (NMEA) format. The NMEA data are strings of characters related to the latitude, longitude, altitude of the GPS and others useful information as how many satellites are being viewed and the kind of signal available (e.g. not valid, single signal, differential).

```
$GPRMC,130135.000,V,3817.6779,N,00730.1174,E,0.0,0.0,040510,,,N*76
$GPGGA,130135.000,3817.6779,N,00730.1174,E,0,02,0.0,,M,,M,,0000*6F
$GPVTG,0.0,T,,M,0.0,N,0.0,K,N*02
$GPGLL,3817.6779,N,00730.1174,E,130135.00,V,N*71
$GPGSA,A,1,,,,,,,,,,,,,,*1E
$GPRMC,130136.000,V,3817.6779,N,00730.1174,E,0.0,0.0,040510,,,N*75
$GPGGA,130136.000,3817.6779,N,00730.1174,E,0,02,0.0,,M,,M,,0000*6C
$GPVTG,0.0,T,,M,0.0,N,0.0,K,N*02
$GPGLL,3817.6779,N,00730.1174,E,130136.00,V,N*72
$GPGSA,A,1,,,,,,,,,,,,,,*1E
$GPRMC,130137.000,V,3817.6779,N,00730.1174,E,0.0,0.0,040510,,,N*74
$GPGGA,130137.000,3817.6779,N,00730.1174,E,0,02,0.0,,M,,M,,0000*6D
$GPVTG,0.0,T,,M,0.0,N,0.0,K,N*02
$GPGLL,3817.6779,N,00730.1174,E,130137.00,V,N*73
$GPGSA,A,1,,,,,,,,,,,,,,*1E
$GPRMC,130138.000,V,3817.6779,N,00730.1174,E,0.0,0.0,040510,,,N*7B
```

Figure 2.5.1: *Example of the output for a GPS.*

The NMEA library is used [40] to read the GPS data. It is an open source library written in c++ and it helps to convert the strings information from the RS-232 serial port (see Fig. 2.5.1) to a useful data as the direct value of latitude, longitude and altitude of the device.

As soon data from the GPS are available, the current robot position is stored as a reference for the GPS. Given the GPS data of two points it is possible to recover the distance (in metre) and the bearing (i.e. angle between the direction and the meridian). For each point it is also possible to recover the XYZ coordinates in the Earth Centred Earth Fixed (ECEF) coordinate system. The ECEF coordinate system has the centre in the earth centre (as the acronym suggests), the x axis passes in the intersection between the equator and the Greenwich meridian, the z axis passes through the North Pole and the y axis completes the tern. The

problem in using this coordinate system is that the global robot reference frame is in unknown position respect to the ECEF system and there is no possibility to recover the orientation with just the latitude, longitude and altitude information. Instead we use the distance information computed using Equations (2.16a-e) merged with the information of the state vector after the *Prediction Stage.*

$$\Delta lat = lat_2 - lat_1, \tag{2.16a}$$

$$\Delta long = long_2 - long_1, \tag{2.16b}$$

$$a = \sin^2(\Delta lat/2) + \cos(lat_1) \cdot \cos(lat_2) \cdot \sin^2(\Delta long/2), \tag{2.16c}$$

$$c = 2 \cdot \text{atan2}\left(\sqrt{a}, \sqrt{1-a}\right), \tag{2.16d}$$

$$d = R_e \cdot c, \tag{2.16e}$$

where:

lat_1, lat_2 are the latitudes of the initial and final locations,

$long_1, long_2$ are the longitudes of the initial and final locations,

R_e is the radius of the earth (= 6371 km).

To insert the distance information inside the Kalman filter we used a combination of data available from the *Prediction Stage* and from the first robot position when the GPS receives a valid signal.

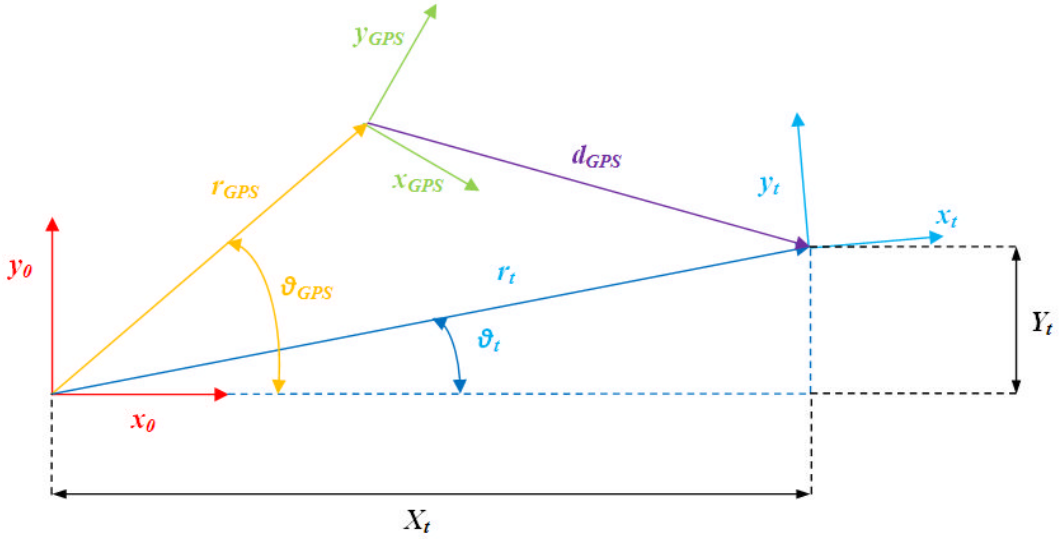

Figure 2.5.2: *Coordinate systems and quantities used to update the Kalman filter with the data available from the GPS receiver.*

As shown in Fig. 2.5.2 as soon a GPS valid signal is received, the current robot position and orientation from the state vector and the GPS information are stored (i.e. $\{X_{GPS}, Y_{GPS}, Z_{GPS}, \gamma_{GPS}, \beta_{GPS}, \alpha_{GPS}, lat_{GPS}, long_{GPS}\}$).

As soon a new GPS valid signal is available, the distance between the first GPS available position and the new point is computed using the Equations (2.16a-e) where $\{lat_1, long_1\}$ refer to the latitude and longitude of the first GPS available position (i.e. $\{x_{GPS}, y_{GPS}\}$ in Fig. 2.5.2) and $\{lat_2, long_2\}$ refer to the latitude and longitude of current robot position (i.e. $\{x_t, y_t\}$ in Fig. 2.5.2).

As previously mentioned, it is just possible to recover the absolute distance between these two points (see d_{GPS} in Fig. 2.5.2). As shown in Fig. 2.5.2 it is possible to compute the distance r_{GPS} of the first GPS available position respect to the global reference frame (i.e. $\{x_0, y_0\}$ in Fig. 2.5.2) and the relative angle ϑ_{GPS}. At this point we use the information of the state vector available from the *Prediction stage*. The current position (i.e. new valid GPS data, $\{x_t, y_t\}$ reference frame in Fig. 2.5.2) is known with some uncertainty and we can estimate the angle ϑ_t. From the trigonometry we can compute the measurement $\{X_t, Y_t\}$ for the robot position starting from the estimated distance d_{GPS} as follows:

$$\widehat{r_t r_{GPS}} = |\vartheta_{GPS} - \vartheta_t|, \tag{2.17a}$$

$$\widehat{d_{GPS} r_t} = \text{asin}\,(r_{GPS}/d_{GPS} \cdot \sin\,(\widehat{r_t r_{GPS}})), \tag{2.17b}$$

$$\widehat{r_{GPS} d_{GPS}} = \pi - \widehat{r_t r_{GPS}} - \widehat{d_{GPS} r_t}, \tag{2.17c}$$

$$r_t = \left(r_{GPS}^2 + d_{GPS}^2 - 2 \cdot r_{GPS} \cdot d_{GPS} \cdot \cos\left(\widehat{r_{GPS} d_{GPS}}\right)\right)^{1/2}, \tag{2.17d}$$

$$X_t = r_t \cdot \cos(\vartheta_t), \quad Y_t = r_t \cdot \cos(\vartheta_t). \tag{2.17e}$$

The measurement vector $\{X_t, Y_t\}$ is then inserted in the *Update Stage* of the Kalman filter without any others manipulations.

An important aspect to underline is the error in the GPS data. The positioning data provided by the satellites is extremely precise but there are several factors that can make the errors in the data considerable. In situations where it is required a high accuracy it is necessary to understand and compensate these sources of error. The main sources of error for the GPS data are atmospheric distortion (in particular reference to the ionosphere), satellite clock inaccuracies, and the travel delays of the satellite signals. The error can also be reduced using the technique known as Differential GPS. The DGPS uses a static base station and allows to reduce the final error of the GPS receiver position and it is able to partially reduce the ionosphere errors.

Chapter 3

From a Video Sequence to a Map

The approach that is being chosen to solve the SLAM problem is based on [31] which gives an overview of both stereo and monocular cameras approaches. Using the reported results of [31] we construct a system based on a single camera sensor capable of building a 3D environmental map and self-localization within the landmarks of that map over a given period of time.

3.1 3D Map and Robot Localization management

Imagery from an optical camera in combination with wheel encoders data are the initial inputs of our system. Based on this initial information we used the approach of [31] to develop the initial solution for monocular SLAM. The overall methodology we use is summarized in Figure 3.1.1. As we can see from Figure 3.1.1 the whole research project is divided into six different subtasks. Each has a different topic and objective within the overall monocular SLAM goal. Furthermore we describe this methodology in detail based on these six different areas as set out in Fig. 3.1.1.

1. From the optical camera we acquire images of the environment and using the SURF feature detection method [2] we extract feature information from the camera images. The general approach is to extract SURF feature points and initialize new features detected or otherwise match a given feature against existing features within the *feature database* (based on features extracted in previous images within the sequence *– see point 2*).

2. When a new feature is detected we first need to recover its depth and then establish if it can be selected as a landmark. This step is called the initialization process. The 3D coordinates of each new feature is initialized with a sum of Gaussians. At each new feature observation, the correspondent sum of Gaussians is updated [31]. After several successive observations, the depth of the feature and, in general, the 3D coordinates can be recovered and the feature becomes a candidate landmark to be added into the

3D environmental map. The resulting map contains information about the position estimation of each landmark and the related positional error [31].

3. From these two stages we obtain information regarding what the robot has previously observed and what it is currently observing. In the meantime the robot is still in motion and we need to maintain the information of this motion and merge all these information together. In the first stage, we obtain the information of robot motion using a dead reckoning approach (e.g. wheels encoders). Subsequently GPS information will be integrated where available. To combine all the available data an implementation of the Extended Kalman Filter (EKF, [38]) is used. The EKF is a useful tool to manage all the data available from the images and the encoders (or GPS) reducing the errors related to these data in the final system.

4. Until now we have used features from optical camera images. We now extend this to the use of the thermal camera and in this we mainly modify stage 1 because we need different feature extraction technique. Here the initial idea was to use the Maximally Stable Extremal Region (MSERs, [3]) that appeared to be better suited to thermal images which generally have less details and greater region consistency than the texture information in optical images. After the MSERs implementation we realized that not much more features were detected comparing the use of the SURF feature so we decide to carry on using the SURF feature detector.

5. A GPS hardware integration is also available and it is used merged with the encoders' data. To increase the precision of the robot position estimation we aim to extend the work by the use of an optical flow technique [34] rather than the dead reckoning approach which is prone to error. This stage aims to decrease the overall error in the SLAM outputs.

6. Finally, to decrease the uncertainty of the estimations, a future work is the use of the loop closing method [31]. This involves also the study of different techniques to store and to describe images in an efficient way for successive comparison.

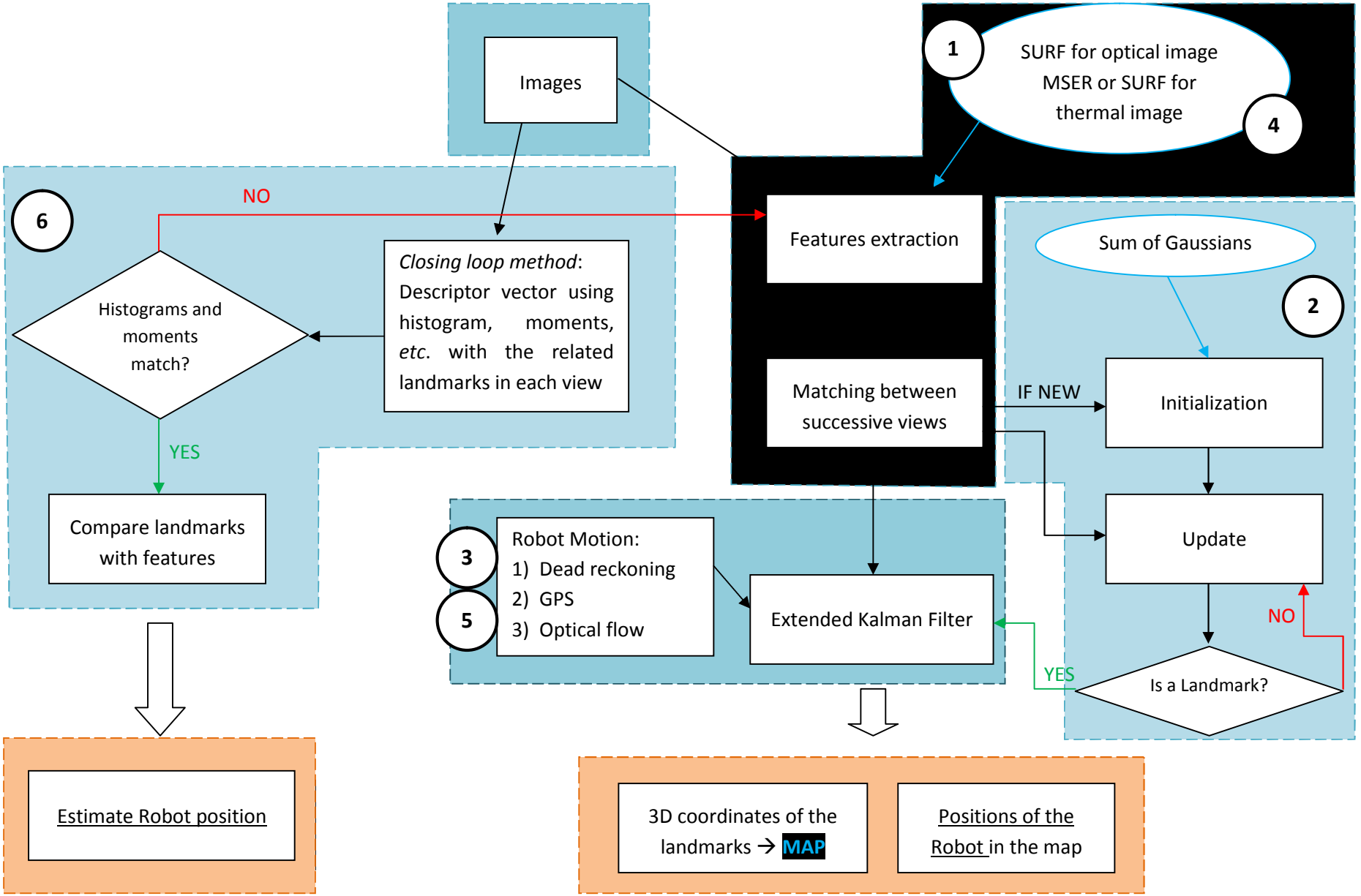

Figure 3.1.1: *Overall design scheme for the six key stages of the project.*

3.2 Features extraction and Initialization

As introduced in the Section 2.1 and 3.1, from each image we extract a number of SURF features. As a first step we select a set of features from the first two frames and use them to build the initial *features database*. This database is then used to track the features from image to image as the robot transits through the environment. This database is also updated every *n* frames with new features based on a technique of "parallel" matching which aims to increase the probability of landmark detection using more stable features through images.

After a feature is detected, following the approach of [31], it is initialized and updated until the initial 3D coordinates are determined.

The information available from each image is the 2D coordinates of a feature in the image respect to the camera reference frame (in pixels). Using the calibration matrix *K* we convert this feature coordinates from pixels to metre units as follows:

$$\lambda \boldsymbol{x} = \lambda \begin{Bmatrix} u \\ v \\ 1 \end{Bmatrix} = \begin{bmatrix} f\, s_x & 0 & o_x \\ 0 & f\, s_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} X \\ Y \\ Z \end{Bmatrix} = K\, \boldsymbol{X}, \tag{3.1}$$

where:

- λ = distance (depth) of a 3D point along the z axis of the camera [m];
- $\{u, v\}$ = 2D pixel coordinates in the image camera plane of the 3D point;
- f = camera focal length [mm];
- s_x = length of the pixel along the horizontal direction [pixels/mm];
- s_y = length of the pixel along the vertical direction [pixels/mm];
- o_x = x coordinate of the optical centre respect to the up – left corner of the image camera plane;
- o_y = y coordinate of the optical centre respect to the up – left corner of the image camera plane;
- $\{X, Y, Z\}$ = 3D coordinates of a point in the camera reference frame [m];

Each parameter is estimated using the Matlab Camera Calibration Toolbox [41], using the technique of [42], that permits to estimate the lens distortion values and the relative errors for each calibration parameters.

From Equation 3.1 we can see that the depth λ is the Z coordinates of a feature which is, in general, unknown at this stage of the SLAM operation. What we can do is to multiply the 2D

pixel coordinates $\boldsymbol{x} = (u, v)$ by the inverse of the calibration matrix obtaining the 3D coordinates of a feature in millimetres up to a scalar factor that is the unknown depth:

$$\boldsymbol{X} = \begin{Bmatrix} X \\ Y \\ Z \end{Bmatrix} = \lambda\, K^{-1} \begin{Bmatrix} u \\ v \end{Bmatrix}, \tag{3.2}$$

dividing the above equation by λ we obtain:

$$\boldsymbol{X}/\lambda = \begin{Bmatrix} X/\lambda \\ Y/\lambda \\ Z/\lambda \end{Bmatrix} = \begin{Bmatrix} X/Z \\ Y/Z \\ 1 \end{Bmatrix} = \begin{Bmatrix} x \\ y \\ 1 \end{Bmatrix} = K^{-1} \begin{Bmatrix} u \\ v \\ 1 \end{Bmatrix}. \tag{3.3}$$

Once we calibrate the image using the matrix *K*, we have the coordinates of a point in metres unit and the focal length value became one. The next stage is now to recover the direction of the point respect to the camera reference frame. The direction of a feature is represented by the angles θ and ϕ:

$$\begin{Bmatrix} \theta \\ \phi \end{Bmatrix} = \begin{Bmatrix} \arctan\,(y/x) \\ -\arctan\,(1/\sqrt{x^2 + y^2}) \end{Bmatrix}, \tag{3.4}$$

and these angles refer to a polar reference frame showed in Fig. 3.2.1.

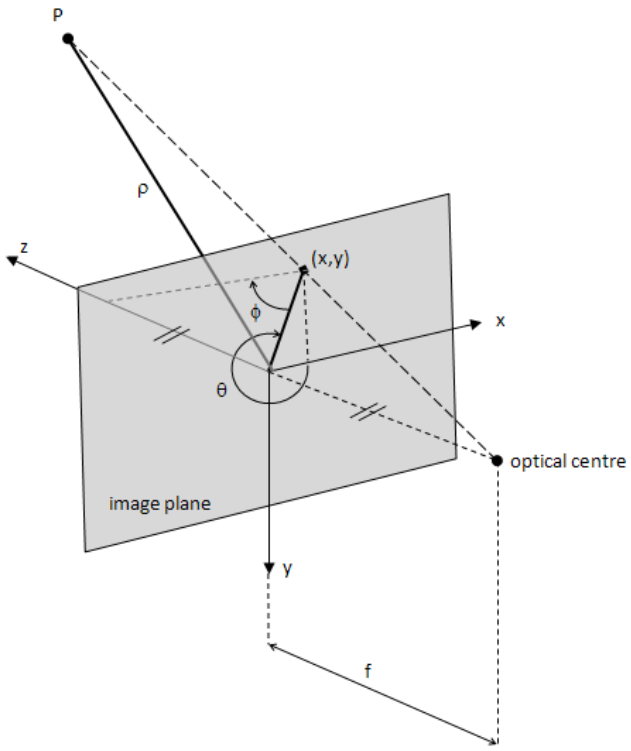

Figure 3.2.1: *Feature direction of a feature in polar coordinates* (θ, ϕ).

Whereas we use images from a single camera, the current depth of a feature point P (i.e. radius ρ in polar coordinates, Fig. 3.2.1) is unknown. As introduced earlier in this chapter, from [31] we estimate the depth of a feature initializing the correspondent feature 3D coordinates by a sum of Gaussians. The sum of Gaussians is then updated by successive observations of the same feature. In this process we use the hypothesis that the robot can see over a specific depth range [ρ_{min}, ρ_{max}] from its sensor. This range is initially defined as a bounded range for initialization. However, the initial value of ρ_{min} and ρ_{max} are arbitrary and can be set empirically based on the environment and sensor capabilities.

To represent a Gaussian we need only two values: its mean value μ and its standard deviation σ. The sum of Gaussians that approximates the *a priori* knowledge on the depth is initialized as follows:

$$P(\theta, \phi, \rho) = \Gamma(\theta, \sigma_\theta) \cdot \Gamma(\phi, \sigma_\phi) \cdot \sum_i \omega_i \Gamma_i(\rho_i, \sigma_{\rho_i}) \tag{3.5}$$

this represents the 3D coordinates of a feature with its relative error. The sum of Gaussians is computed according to [31] using the following geometric series:

$$\rho_0 = \rho_{min}/(1-\alpha), \tag{3.6a}$$

$$\rho_i = \beta^i \cdot \rho_0, \quad \sigma_{\rho_i} = \alpha \cdot \rho_i, \quad \omega_i \propto \rho_i, \tag{3.6b}$$

$$\rho_{n-2} < \rho_{max}/(1-\alpha), \qquad \rho_{n-1} \geq \rho_{max}/(1-\alpha). \tag{3.6c}$$

Once again, according to [31], the values of α (= 0.25) and β (= 2.5) are chosen empirically, but following some constraints related to the distribution of a Gaussian that we want to have. After this initialization each Gaussian $\{\mu_i^p = \{\rho_i, \theta, \phi\}, \sum_i^p = \{\sigma_{\rho_i}^2, \sigma_\theta^2, \sigma_\phi^2\}\}$ is then converted from Polar coordinates to Cartesian coordinates in the current robot reference frame, which is the reference frame for the i^{th} feature also for subsequent observations:

$$\mu_i^c = \boldsymbol{g}\begin{pmatrix}\theta\\ \phi\end{pmatrix} = \begin{Bmatrix}\rho_i \cos\phi \cos\theta\\ \rho_i \cos\phi \sin\theta\\ -\rho_i \sin\phi\end{Bmatrix} = \begin{pmatrix}x_i\\ y_i\\ z_i\end{pmatrix} \qquad \Sigma_i^c = G\Sigma_i^p G^T = \{\sigma_{x_i}^2, \sigma_{y_i}^2, \sigma_{z_i}^2\}, \tag{3.7}$$

$$where\ G = \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{X}} = \begin{bmatrix}\frac{\delta x}{\delta \rho} & \frac{\delta x}{\delta \theta} & \frac{\delta x}{\delta \phi}\\ \frac{\delta y}{\delta \rho} & \frac{\delta y}{\delta \theta} & \frac{\delta y}{\delta \phi}\\ \frac{\delta z}{\delta \rho} & \frac{\delta z}{\delta \theta} & \frac{\delta z}{\delta \phi}\end{bmatrix}. \tag{3.8}$$

After this initialization we have *n* 3D coordinates of the same feature with respect to the camera reference frame where the feature is been seen the first time. As a result of successive observations of the same feature we update that sum of Gaussians and, in this way, we can choose which Gaussian best approximates the feature pose and prune the Gaussian with lesser probability of representing the feature position.

The selection procedure is made by an estimation of the normalized likelihood for each Gaussian. The normalized likelihood is computed every time we have a new observation $z_t = (\theta_t, \phi_t)$ (with covariance R_t) of the feature and, for each time, we prune the Gaussian which has the likelihood under a certain threshold. The likelihood of Γ_i to be an estimation of the observed feature is:

$$L_i^t = \frac{1}{2\pi\sqrt{|S_i|}} \exp\left(-\frac{1}{2}(z_t - \hat{z}_i)^T S_i^{-1}(z_t - \hat{z}_i)\right), \tag{3.9}$$

where S_i is the covariance of the innovation $z_t - \hat{z}_i$. The prediction of the observation $\hat{z}_i = (\theta_i, \phi_i)$ is estimated considering each Gaussian in the current robot frame (i.e. at time *t*). The figure below explains the transformations that we need to correctly compute the value of $\hat{z}_i$.

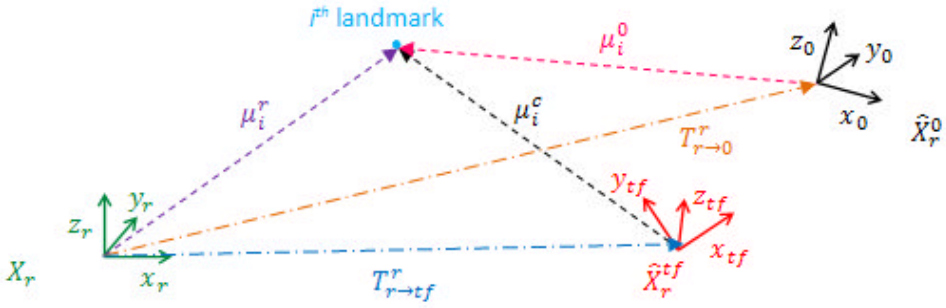

Figure 3.2.2: *Feature/landmark position vector respect to different robot reference frames.*

In Fig. 3.2.2 are represented three different robot reference frames. The X_r reference frame is the global reference frame and is related to the first position of the mobile robot. This decision is been taken because our mobile robot is in an unknown environment and it has not external references. The $\hat{X}_r^{tf}$ frame refer to the frame where the landmark was been seen the first time while the $\hat{X}_r^0$ frame refer to the actual robot frame. In Fig. 3.2.2 we can see the notation of the landmark vector respect to the reference frames (i.e. μ_i^r, μ_i^c and μ_i^0) and the translation vectors that describe the transformation of the last two frames respect to the global reference frame (i.e. $T_{r\to tf}^r$ and $T_{r\to 0}^r$). One notices that the three reference frames refer to a

robot frame. This is because at each time for the mobile robot we have two reference frames: robot reference frame and the image reference frame. From Fig. 3.2.3 we can see the rotation matrix that represents the transformation between these two reference frames.

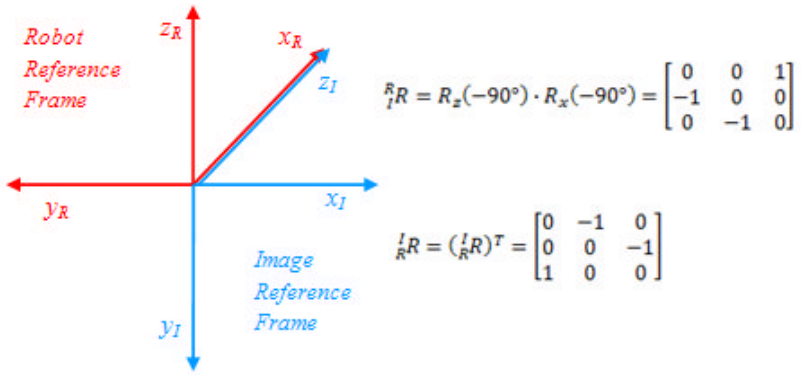

Figure 3.2.3: *Relation between the robot reference frame and the image reference frame.*

All the information from the images is acquired in the image reference frame, while the information stored in the EKF and in the 3D map refers to the robot reference frame.

Nomenclatures	Description
X_r	*Global* (or *world*) *reference frame*, it is taken as the first position of the mobile robot
$\hat{X}_r^{tf} = \{X^{tf}, Y^{tf}, Z^{tf}, \gamma^{tf}, \beta^{tf}, \alpha^{tf}\}$	*Robot frame* where the landmark is seen the first time, its coordinates are expressed respect to the global reference frame
$\hat{X}_r^0 = \{X^0, Y^0, Z^0, \gamma^0, \beta^0, \alpha^0\}$	*Current robot frame*, its coordinates are expressed respect to the global reference frame
$T^r_{r\to tf} = \{X^{tf}, Y^{tf}, Z^{tf}\}$	Translation vector
$T^r_{r\to 0} = \{X^0, Y^0, Z^0\}$	Translation vector
$\mu_i^r = \{x_i^r, y_i^r, z_i^r\}$	3D coordinates of the ith landmark respect to the *Global* reference frame
$\mu_i^c = \{x_i^c, y_i^c, z_i^c\}$	3D coordinates of the ith landmark respect to the *robot* reference frame
$\mu_i^0 = \{x_i^0, y_i^0, z_i^0\}$	3D coordinates of the ith landmark respect to the *current* robot reference frame

Table 3.2.1: *Description of the variables used in our formulation based on* [31].

The variable $\hat{z}_i$ in Equation 3.9 is the Cartesian coordinates μ_i^0 transformed in polar reference frame as described in [31]. The following table describes all the variables used in Equation 3.9 to compute the prediction of the observation. As we can see from it, we use 6 degree of freedom to describe the robot position and 3 to describe a landmark/feature location.

Once we have specified all the variables interested, we can compute $\hat{z}_i$ as described in [31] as follows:

$$\hat{z}_i = \boldsymbol{h}\left(\boldsymbol{to}\left(\hat{X}_r^0, \boldsymbol{from}\left(\hat{X}_r^{tf}, \mu_i^c\right)\right)\right) = H\left(\hat{X}_r^0, \hat{X}_r^{tf}, \mu_i^c\right), \tag{3.10}$$

where:

$$\begin{aligned}\mu_i^r = \boldsymbol{from}\left(\hat{X}_r^{tf}, \mu_i^c\right) &= R_x(\gamma^{tf}) \cdot R_y(\beta^{tf}) \cdot R_z(\alpha^{tf}) \cdot {}_I^R R \cdot \mu_i^c + T_{r\to tf}^r \\ &= {}_{tf}^r R \cdot \mu_i^c + T_{r\to tf}^r,\end{aligned} \tag{3.11}$$

$$\mu_i^0 = \boldsymbol{to}\left(\hat{X}_r^0, \mu_i^r\right) = {}_r^0 R \cdot \mu_i^r - T_{r\to 0}^r, \tag{3.12}$$

$$\hat{z}_i = \begin{Bmatrix}\theta_i \\ \phi_i\end{Bmatrix} = \boldsymbol{h}(R_x(90^\circ) \cdot \mu_i^0) = \begin{Bmatrix}\arctan\left(-z_i^0 / x_i^0\right) \\ -\arctan\left(\frac{y_i^0}{\sqrt{(x_i^0)^2 + (-z_i^0)^2}}\right)\end{Bmatrix}. \tag{3.13}$$

In Equation 3.9 we use the matrix S_i so called covariance of the innovation and it is computed as follows:

$$S_i = H_1 P_{X_r^0} H_1^T + H_2 P_{X_r^{tf}} H_2^T + H_1 P_{X_r^0, X_r^{tf}} H_2^T + H_2 P_{X_r^0, X_r^{tf}}^T H_1^T + H_3 \Sigma_i^c H_3^T + R_t, \tag{3.14a}$$

$$H_1 = \frac{\partial H}{\partial X_r^0} = \begin{bmatrix} \frac{\delta\theta_i}{\delta X^0} & \frac{\delta\theta_i}{\delta Y^0} & \frac{\delta\theta_i}{\delta Z^0} & \frac{\delta\theta_i}{\delta \gamma^0} & \frac{\delta\theta_i}{\delta \beta^0} & \frac{\delta\theta_i}{\delta \alpha^0} \\ \frac{\delta\phi_i}{\delta X^0} & \frac{\delta\phi_i}{\delta Y^0} & \frac{\delta\phi_i}{\delta Z^0} & \frac{\delta\phi_i}{\delta \gamma^0} & \frac{\delta\phi_i}{\delta \beta^0} & \frac{\delta\phi_i}{\delta \alpha^0} \end{bmatrix}, \tag{3.14b}$$

$$H_2 = \frac{\partial H}{\partial X_r^{tf}} = \begin{bmatrix} \frac{\delta\theta_i}{\delta X^{tf}} & \frac{\delta\theta_i}{\delta Y^{tf}} & \frac{\delta\theta_i}{\delta Z^{tf}} & \frac{\delta\theta_i}{\delta \gamma^{tf}} & \frac{\delta\theta_i}{\delta \beta^{tf}} & \frac{\delta\theta_i}{\delta \alpha^{tf}} \\ \frac{\delta\phi_i}{\delta X^{tf}} & \frac{\delta\phi_i}{\delta Y^{tf}} & \frac{\delta\phi_i}{\delta Z^{tf}} & \frac{\delta\phi_i}{\delta \gamma^{tf}} & \frac{\delta\phi_i}{\delta \beta^{tf}} & \frac{\delta\phi_i}{\delta \alpha^{tf}} \end{bmatrix}, \tag{3.14c}$$

$$H_3 = \frac{\partial H}{\partial \mu_i^c} = \begin{bmatrix} \frac{\delta\theta_i}{\delta x_i^c} & \frac{\delta\theta_i}{\delta y_i^c} & \frac{\delta\theta_i}{\delta z_i^c} \\ \frac{\delta\phi_i}{\delta x_i^c} & \frac{\delta\phi_i}{\delta y_i^c} & \frac{\delta\phi_i}{\delta z_i^c} \end{bmatrix}. \tag{3.14d}$$

Following [32], to compare the likelihood of each Gaussian, we use the normalized likelihood that is, for the hypothesis *I*, the product of likelihoods obtained for Γ_i :

$$\Lambda_i = \frac{\prod_t L_i^t}{\sum_j \prod_t L_j^t}. \tag{3.15}$$

After computing the normalized likelihood for each Gaussian, the Gaussian associated with the worst hypotheses is pruned if $\Lambda_i < \tau$ ($\tau = 0.8/n$, n = number of Gaussians last). After few observations we have only one Gaussian and the associate 3D coordinates of the feature are compared with the last observation using the χ^2 test ([31] [43]). If the coordinates pass the test, the associated feature is declared as landmark, inserted in the *landmark database* and deleted from the *features database*. If this is not the case, this means that the feature is not in the depth range [ρ_{min}, ρ_{max}] that we have selected in the preview stage or the observations were not enough consistent and the feature is rejected.

In Fig. 3.2.4 we illustrate the process from a feature to a new landmark stored in the 3D map. The first image represents the initialization stage of a feature as a sum of Gaussians. The successive two images in Figure 3.2.4 show that, thanks to successive observation of the same feature, some Gaussians are pruned.

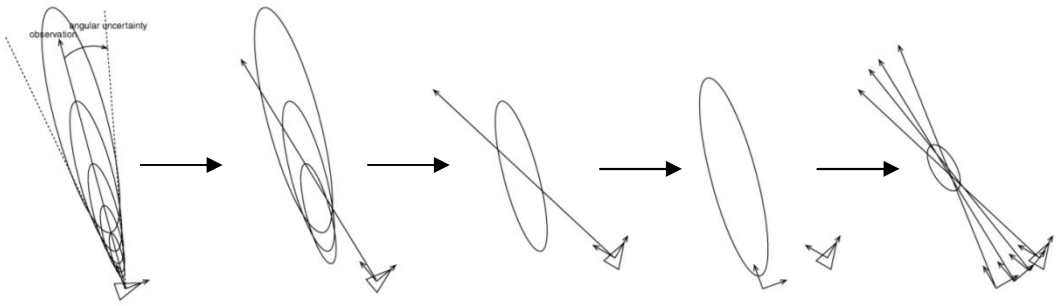

Figure 3.2.4: *"From an observed feature in the images to a landmark in the map"*, *(*[31], *Fig.11, pp.356).*

As we previously mentioned, when only one Gaussian remains, the feature is declared as a landmark and it is projected into the global reference frame. At the end the past observations are used to update the estimation of the landmark position and reduce the overall error related to the Gaussian (see right image in Fig. 3.2.4) and at this point the landmark is added to the 3D map. As the number of landmarks detected from the environment increases, we slowly build up a 3D landmark map of the environment over time.

Once we compute all of these values we can merge them with the information of the robot motion and proceed with the use of the extended Kalman filter (see Section 2.4).

3.3 Camera Calibration

As introduced in Section 3.2, to use the information extracted from the images (i.e. features points) we need to calibrate the cameras using the *calibration matrix K*. To calibrate a camera there are already several implemented methods such as the one in OpenCV [34] or the camera calibration toolbox available for Matlab [41]. Both algorithms are based on the same principles of camera calibration using the method of [42]. In both mentioned methods it is possible to extract the calibration matrix *K* and the lens distortion coefficients using several images of the same calibration rig of an *a priori* fixed pattern (e.g. chessboard, Fig. 3.3.1) in different configurations/orientations. Each camera as a varying amount of lens distortion and the calibration process allows us to compute the associated coefficients for successive correction [42].

3.3.1 Thin lens and pinhole camera model

A camera is composed by a set on lenses to direct the light. The reference model is the *thin lens* model [37] and it is a mathematical model defined by an axis – *optical axes* – and by a perpendicular plane to this axis – *focal plane* – with a circular aperture in the intersection between the plane and the axis.

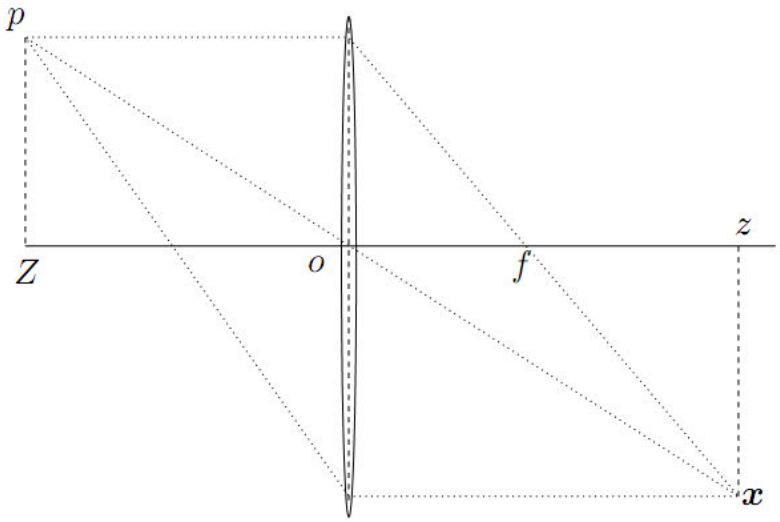

Figure 3.3.1: *Point p and its image point x in the thin lens model (*[37], *Fig.3.4, pp.49).*

The thin lens is characterized by two parameters: the *focal length f* and the *diameter d.* The thin lens has also two fundamental properties:

- Each ray parallel to the optical axis that enters through the circular aperture intersects the optical axis itself at a distance of *f* from the optical centre and that point is called the *focal point* of the lens (see Fig. 3.3.1).
- All rays through the optical centre are undeflected.

If we now consider a point *p* in the 3D space at a distance *Z* from the optical centre (see Fig. 3.3.1) we can construct two rays: the first one from the point *p* parallel to the optical axis and the second one from the point *p* through the optical centre *o*. The point *x* is the intersection of the two rays and *z* is the distance between the point *x* and the optical centre *o* and it is called the image point of *p*. Following [37] we can describe the fundamental equation for the *thin lens model* as follows:

$$\frac{1}{Z} + \frac{1}{z} = \frac{1}{f} \tag{3.16}$$

If we think now that the aperture of the thin lens becomes zero, all the rays are forced to pass through the optical centre *o* (see Fig. 3.3.2).

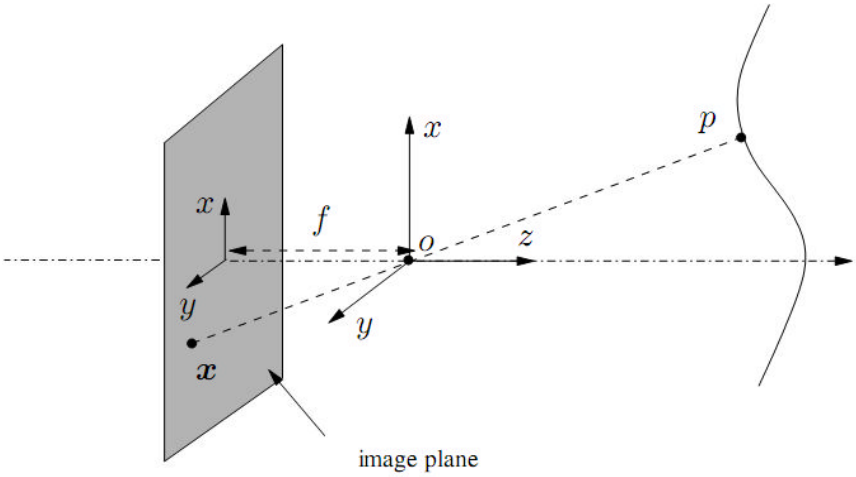

Figure 3.3.2: *Pinhole camera model (*[37],*Fig. 3.6,pp. 51).*

If we now consider a point *p* with coordinates $\boldsymbol{X} = \{X, Y, Z\}$ respect to the reference frame centred in the optical centre *o*, as reported in [37] we can recover the relation between the 3D coordinates of the point *p* in the 3D space with the image point *x* :

$$x = -f\frac{X}{Z}, \quad y = -f\frac{Y}{Z}, \tag{3.17}$$

where *f* is the *focal length*. It is important to notice that any other point on the line through *o* and *p* projects onto the same coordinates $\boldsymbol{x} = \{x, y\}$ and is this ambiguity that introduces the uncertainty in the depth of a feature point. The described model is called an ideal pinhole camera model and is an idealization of the thin lens model [37].

3.3.2 Optical Camera calibration

For the optical camera we use a chessboard of 9x7 squares of 28 millimetres (see Fig. 3.3.3). To calibrate the optical camera we used both the OpenCV algorithm and the Matlab calibration toolbox with similar results (see Table 3.3.1). As mentioned in introduction of Section 3.3, both implementations are based on Zhang camera calibration method [42].

The calibration data used in this work are the outputs from the Matlab camera calibration toolbox [41]. This choice is based on the possibility to compute not just the entries of the matrix *K* but also the associated errors, which is not possible using the OpenCV implementation [34]. The calculated errors are then used during the initialization process to compute the error related to the observation value within the extended Kalman filter.

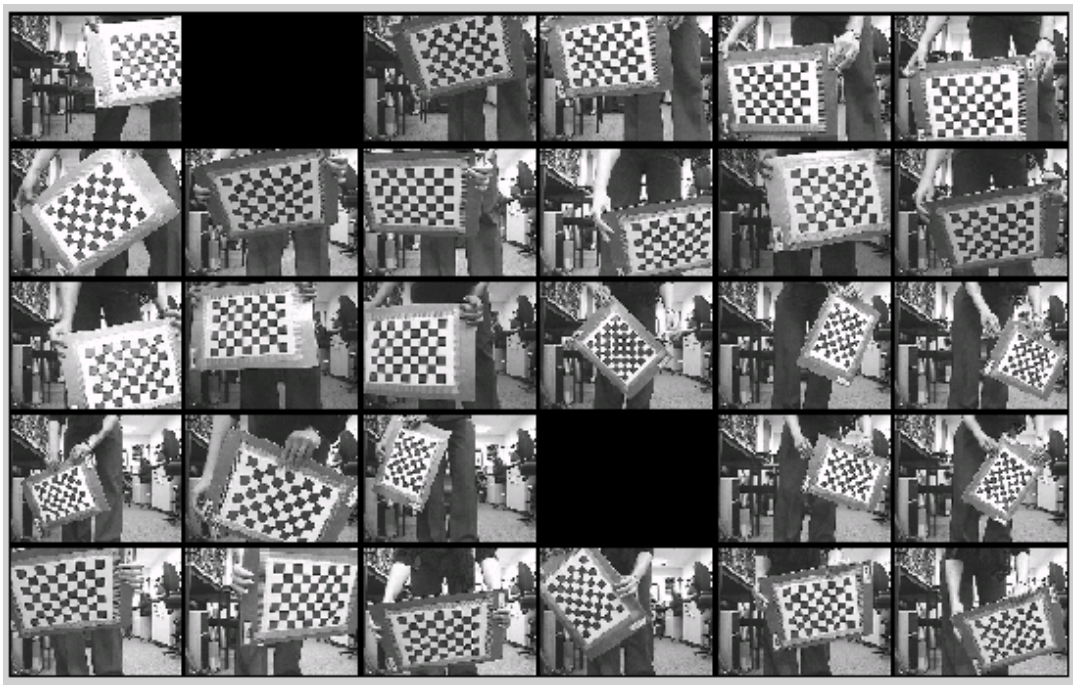

Figure 3.3.3: *Images used to calibrate the optical camera.*

As we can see from Fig. 3.3.3, the chessboard images are captured in varying orientations of the calibration rig and this set of images is used as input to the calibration process.

In Table 3.3.1 are shown the outputs of the calibration process for the optical camera. As we can see the entries of the calibration matrix *K* are compatible if we look at Matlab output errors. In both cases we used the same images (Fig. 3.3.3).

	OpenCV	*Matlab*
	Calibration matrix K	
$f_x = f \cdot s_x$	726.0	730.4 ± 8.5
$f_y = f \cdot s_y$	727.6	732.0 ± 8.6
c_x $[pixels]$	308.2	310 ± 5.4
c_y $[pixels]$	271.6	277 ± 5.9
	Lens distortion coefficients	
k_1	- 0.3210	-0.3189 ± 0.015
k_2	0.2177	0.1832 ± 0.052
k_3	0.0*	-0.0016 ± 0.0027

Table 3.3.1: *Outputs of the calibration process using OpenCV and Matlab implementations of the approach of* [42] .

Over a sequence of images with a known calibration target (here a 9x7 black/ white chessboard) and automatic corner detection routine is applied to detect the intersections of the checkers. In order to have a robust calibration, a set of images of the known calibration target has to be captured in different orientation and position. In each image locations of the corners are detected and the set of corners are correlated from one image to the other. This set of detected corners, together with the known geometry of the calibration target, is fed into the camera calibration technique of [42] in order to recover the set of camera parameters listed in Table 3.3.1.

 *OpenCV algorithm [34] does not estimate the k_3 lens distortion coefficient

3.3.3 Thermal Camera calibration

To calibrate the thermal camera we used a chessboard of the same dimension used for the optical camera but is made to be seen constructed to be visible within the thermal images. The thermal chessboard is composed of 9x7 squares of 28 millimetres and is made in steel plate using insulating tape to create the chessboard. In this way the two materials have a different temperature and the thermal camera can detect the different region (i.e. squares) of the calibration rig. To improve the visibility of the chessboard it can be useful to worm up the steel plate. In this way the squares not covered by the insulating tape tend to have a higher temperature increasing the gradient between them and the ones covered by the tape. The images used to calibrate the thermal camera are shown in Fig. 3.3.4.

Figure 3.3.4: *Images used to calibrate the thermal camera.*

The calibration is made using just the Matlab camera calibration toolbox [41]. Using the OpenCV camera calibration [34] is no robust and inconsistent results were found as the points in the chessboard were not initially identified within the algorithm. A reason for this it is possibly related to the reduce level of details of the thermal camera imagery compare to the optical. Using the Matlab camera calibration toolbox [41] allows it again to recover information about the errors of the calibration matrix entries and this is useful to initialize the error of a feature point in the image [31].

In Fig. 3.3.4 it is possible to see the images used to calibrate the thermal camera. As the reader can see from Table 3.3.2, the focal length of the thermal camera is larger than the focal length of the optical camera (see Table 3.3.1) and, as a consequence, the field of view of the two cameras is different.

	OpenCV	*Matlab*
	Calibration matrix K	
$f_x = f \cdot s_x$	---	1336.6 ± 135
$f_y = f \cdot s_y$	---	1588.3 ± 158
$c_x\ [pixels]$	---	350.6 ± 75
$c_y\ [pixels]$	---	286.7 ± 108
	Lens distortion coefficients	
k_1	---	-0.55 ± 0.22
k_2	---	1.90 ± 2.8
k_3	---	-0.0050 ± 0.014

Table 3.3.2: *Outputs of the calibration process using OpenCV and Matlab implementations of the approach of* [42] .

In Table 3.3.2 are shown the outputs for the calibration process related to the thermal camera. As we can see from the table above, the errors of the calibration matrix entries are larger compare with the same errors computed for the optical camera. This is could be related to the fact that the thermal images of the chessboard have fewer details than the optical images so when the Matlab camera calibration toolbox [41] extracts the corners there is a larger uncertainty within the corners location. To recover the camera parameters listed in Table 3.3.2 we use the same technique explained in Section 3.3.1.

3.3.4 Thermal to Optical camera transformation

After the calibration of the optical camera and the thermal camera to merge the feature points information it is necessary to recover the transformation between the two cameras. This transformation is the *planar homography* that project one plane (optical) to another (thermal). This type of *mapping* can be express in terms of matrix multiplication. As explained in [37], if we express a point $\boldsymbol{x}_T$ in the thermal image and a correspondent point $\boldsymbol{x}_O$ in the optical image (taken from the same scene) we can express the action of the homography as follows:

$$\boldsymbol{x}_O = sH\boldsymbol{x}_T, \tag{3.18}$$

where $\boldsymbol{x}_O = \{x_O, y_O, 1\}^T$ and $\boldsymbol{x}_T = \{x_T, y_T, 1\}^T$ are in homogeneous coordinates and the parameter s is an arbitrary scale factor.

(a)

(b)

Figure 3.3.5: *Images used to compute the homography that maps the (a) thermal images to the (b) optical images.*

In Fig. 3.3.5 are shown the two input images used to recover the homography. As the two cameras are in two different positions on the mobile robot (see Section 3.5) we need to know the transformation that maps the thermal feature points in the optical seen – used as a reference for the global coordinate system (Section 3.2).

To recover the homography H four correspondent points lying in a plane have to be selected in both images. For the examples of Fig. 3.3.5 the recovered homography transformation is:

$$H = \begin{bmatrix} 0.586002 & 0.081033 & 121.00 \\ -0.004422 & 0.504312 & 163.00 \\ -4.57e-05 & -0.000137 & 1.00 \end{bmatrix},$$

and Fig. 3.3.6. and Fig. 3.3.7 show the overlaid image of the two (optical and thermal) inputs seen of Fig. 3.3.5 before and after the application of the homography H. For each analyzed video it is necessary to recover the best homography matrix. Ideally the estimation of H can be included in the calibration process and the matrix H used for all the video sequences captured with a certain cameras configuration but this is not the real case. The best solution is to compute the best estimation of the homography matrix H within a video sequence and use it to analyze the video.

Figure 3.3.6: *Uncorrected overlap of the optical and thermal images before the computation of the homography.*

Looking at Fig. 3.3.7, the reader can see that the thermal image does not completely overlap the optical image, but the computed homography H is considered empirically a good result.

Figure 3.3.7: *Corrected overlap of the optical and thermal images after the computation of the homography.*

As introduced in this session the homography matrix *H* is used to relate the thermal features to the optical camera reference frame (i.e. the global reference frame). The matrix *H* estimated using the described method relates the pixel coordinates between the thermal imagery and the optical imagery as described in Equation 3.18. In our project we need to merge together the thermal landmarks and optical landmarks in one database used by the mobile robot for the navigation and localization. There are two possible solutions to achieve this:

- Use the 3D coordinates of the thermal landmarks in output of the initialization process (see Section 3.2) and find the transformation between the thermal camera reference frame and the optical camera reference frame.

- Transform the 2D calibrated coordinates and observations of thermal features from thermal images to optical images using the homography matrix *H* computed between the calibrated points (same points used to compute the homography matrix described in the first part of this paragraph).

The first solution is the ideal one because the errors introduced to transform points from one coordinate system to the other is reduced to the minimum. The problem in applying this solution is that given a point in the space (i.e. 3D point) expressed in the thermal camera reference frame, we cannot actually compute the transformation that maps the point from the thermal camera reference frame to the optical camera reference frame (to do that we need to know at least four 3D corresponded points between the cameras and from the calibration process or the image analysis we can have just recover the 2D coordinates of corresponded points). Due to this exclusion we have to implement the second approach. The disadvantage of the second solution is the amount of errors introduced during the transformation:

- An initial error is introduced just before the initialization process when transforming the 2D coordinates of a thermal feature from the thermal camera coordinate system to the optical one. The feature is then initialized as it was extracted from an optical imagery so no extra error components are introduced.

- A constant error is added for each feature observation. As an initialized feature is observed, the 2D coordinates of the matched point in pixels are being calibrated using the thermal camera intrinsic parameters and then past to the transformation matrix to express then respect to the optical camera coordinate system. For each observation an additional component of error is added into the process. As the 2D point of the observation is calibrated and transformed, the observation values are computed and no other error components are added as a result of the transformation.

To compute the transformation between a calibrated thermal point and a calibrated optical point we use the same four points used to compute the homography *H* described at the beginning of the section. Given the four points the first step is to correct the lens distortion using the coefficients computed during the calibration process (Section 3.3.1 and 3.3.2). After that, the four points in both views (i.e. four points in the thermal image and four points in the optical image) are calibrated using the correspondent calibration matrix as follow:

$$\tilde{\boldsymbol{x}}_T = K_T \boldsymbol{x}'_T \;\; and \;\; \tilde{\boldsymbol{x}}_O = K_O \boldsymbol{x}'_O, \tag{3.19}$$

where $\boldsymbol{x}'_T = \{x'_T, y'_T, 1\}^T$ is a point in the thermal image after the lens distortion correction (and $\boldsymbol{x}'_O = \{x'_O, y'_O, 1\}^T$ is the correspond corrected point in the optical image) and $\tilde{\boldsymbol{x}}_T = \{\tilde{x}_T, \tilde{y}_T, 1\}^T$ is the correspond calibrated point ($\tilde{\boldsymbol{x}}_O = \{\tilde{x}_O, \tilde{y}_O, 1\}^T$ is the correspond calibrated point in the optical image).

After correcting the lens distortion and calibrating the points in both images, $\boldsymbol{x'}_T$ and $\boldsymbol{x'}_O$ are used to compute the homography matrix H_c between the calibrated points. The Equation 3.18 is still representing the relation between the two points. The only difference is that during the mapping (i.e. transformation between a point in the thermal camera to a correspond point in the optical camera) the input has to be the calibrated coordinate of the thermal point and the calibrated correspond optical point is the output. Given a calibrated thermal point $\tilde{\boldsymbol{x}}_T$, the correspond optical point is estimated as follows:

$$\tilde{\boldsymbol{x}}_O = H_c \tilde{\boldsymbol{x}}_T. \tag{3.20}$$

After the described procedure the transformed calibrated thermal points are expressed in the optical camera reference frame and they are ready to be initialized as described in Section 3.2. The thermal features are still matched using the 2D pixel coordinates referred to the thermal camera reference frame but for each new observation of a feature or a landmark, the correspond observed point is transformed using Equation 3.20 and then the observation values (i.e. angles ϑ and φ) are computed as described in Section 3.2.

3.4 Artificial Test Environment and the Extended Kalman Filter

The EKF [38] is the key element of the SLAM problem merging all the information available from the robot sensors (e.g. GPS, encoders, etc.) and the input visual data (i.e. thermal features and optical features) in a robust iterative and statistical evaluation methodology (see Section 2.4).

During the implementation of the system, a test software and environment were implemented to analyze the performance of the EKF in a known environment. To perform this, an EKF example code is realized using a artificial test environment (see Fig. 3.4.1).

The EKF test code is based on the equations described in Section 2.4 and, as inputs, the code uses few features/landmarks placed in different position in front of the robot in the artificial test environment as shown in Fig. 3.4.1.

The motion of the robot it simulated as a translation motion along the x direction with a velocity of 0.2 m/s and the test code uses an integration interval time τ of 0.083s that corresponds to the integration interval time used is the main software. This interval is chosen based on the frames per second (fps) rate available from the video sequences (i.e. τ = 1/fps).

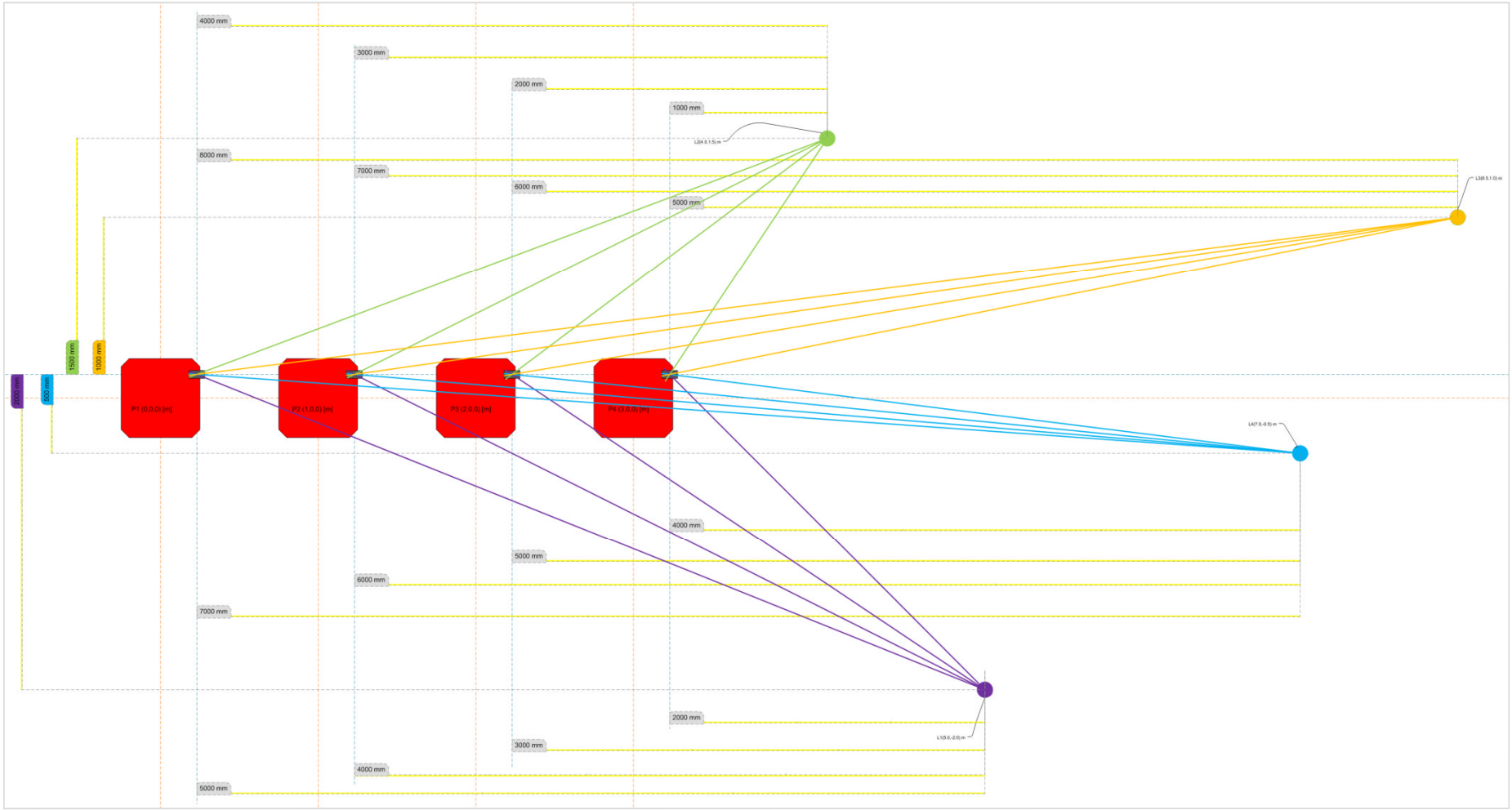

Figure 3.4.1: *Artificial test environment used to test the EKF implementation*

3.4.1 Artificial test environment initialization

The artificial test environment is used to analyze the efficiency of the implemented EKF code in relation with the different input components as positions, observations and errors of the robot and the landmarks.

Four artificial landmarks are used to create the test environment as illustrated in Fig. 3.4.1. From Fig 3.4.1 the reader can see an example of several robot positions (i.e. red squares) and the vectors used to represent the landmarks directions respect to the optical camera centre. For the 3 DOF case the landmarks lie on the same plane of the robot reference frame (i.e. $Z = 0$, see the *robot reference frame* definition in Section 3.2) whilst, in the 6 DOF, the landmarks are placed in different positions along the Z axis as shown in Table 3.4.1. All the coordinates are expressed respect to the robot reference frame.

	3 DOF [m]	**6 DOF** [m]
Landmark 1 (**L1**)	$\{5.00, -2.00, 0.00\}$	$\{5.00, -2.00, 1.00\}$
Landmark 2 (**L2**)	$\{4.00, 1.50, 0.00\}$	$\{4.00, 1.50, 0.50\}$
Landmark 3 (**L3**)	$\{8.50, 1.00, 0.00\}$	$\{8.50, 1.00, 2.00\}$
Landmark 4 (**L4**)	$\{7.00, -0.50, 0.00\}$	$\{7.00, -0.50, -0.15\}$

Table 3.4.1: *True coordinates value of the artificial landmarks.*

The aim of the test code for the EKF implementation is to verify the efficiency when noise is added to the inputs of the EKF, with particular regards to noise related to the true position of the landmarks (when first added to the state vector of the EKF) and to the robot position observations.

	3 DOF [m]	**6 DOF** [m]
Landmark 1 (**L1**)	$\{5.99, -2.40, 0.00\}$	$\{6.00, -2.40, 1.20\}$
Landmark 2 (**L2**)	$\{5.00, 1.95, 0.00\}$	$\{5.20, 1.95, 0.65\}$
Landmark 3 (**L3**)	$\{9.70, 1.20, 0.00\}$	$\{7.23, 0.85, 1.70\}$
Landmark 4 (**L4**)	$\{8.37, -0.67, 0.00\}$	$\{9.44, -0.67, -0.20\}$

Table 3.4.2: *Initial coordinates of the artificial landmarks with noise added.*

In this test software we apply two different techniques to generate the noise for landmarks and robot positions. During the feature initialization process (see Section 3.2) we observe that the main component of error is related to the 3D (or 2D) initial position of the feature/landmark rather than the direction values used to during the initialization and then in

the update process. Based on this observation, the aim of the test code - related to the landmarks - is to estimate the capability of the EKF to evaluate the true position of a landmark starting from a initial noisy coordinates affected by a positioning error (see Tab. 3.4.2 and Fig. 3.4.2) using measurement observations without added noise. The noise in the initial landmark coordinates is imposed empirically and it allows to maintain invariant the direction of the landmark. The corresponding initial error used to compute the covariance matrix for each landmark is chosen to be a value of ± 2 m along the *x* direction and ± 1 m along the *y* direction. These values are chosen empirically based on the outputs of the feature initialization process for a range of points with estimated initial coordinates of the same magnitude of the artificial landmarks selected. The direction observation values are not affected by any noise but, as requested from the EKF algorithm, a component of error have to be inserted to compute the associate covariance matrix. In this case we use an error of ± 2 deg for both angles that identify a landmark direction.

The second technique used to add noise to the observation values regards the robot position observations. In this case we add a random Gaussian noise to each component of the observation vector which entries are the absolute position of the robot along the *x* and *y* axes and the orientation angle α around the *z* axis. The mean μ of the Gaussian model is based on the true position of the robot (predicted without noise based on the imposed translation velocity V_{tra}) and the deviation σ is based on the average error of the robot position estimated empirically.

3.4.2 System with 3 DOF

A first EKF code is being implemented to simulate the behaviour of the mobile robot with three degrees of freedom: 2D position in a plane (i.e. *x* and *y* coordinates) and the orientation (i.e. α angle around the *z* axis).

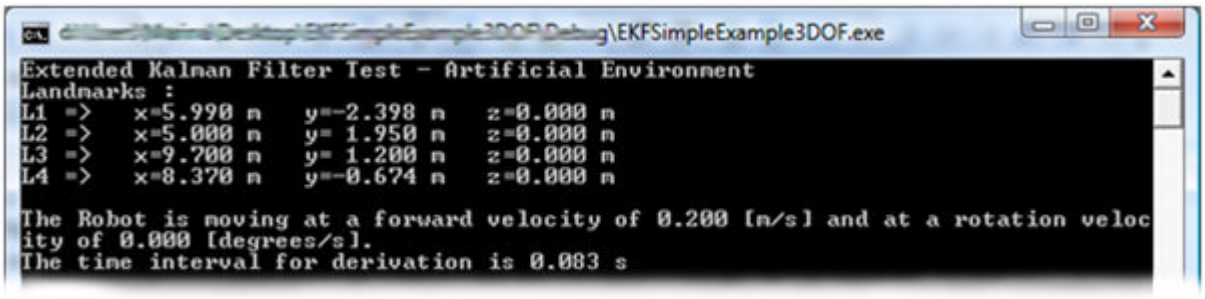

Figure 3.4.2: *Settings used to initialize the test code for the 3 DOF case.*

The robot starts at position $\{0,0,0\}^T$ and the initial noisy coordinates of the 4 landmarks (see Tab. 3.4.2) are added to the state vector. Fig. 3.4.2 shows the settings used to initialize the test code and it is also possible to see the coordinates of the artificial landmarks as previously mentioned and shown in Table 3.4.2.

			LANDMARKS			
		True robot position [m]	**L1** φ_1 [deg]	**L2** φ_2 [deg]	**L3** φ_3 [deg]	**L4** φ_4 [deg]
TIME	t ≅ 5 s	$\{1.0,0.0,0.0\}^T$	-63 ± 2	-63 ± 2	-82 ± 2	-85 ± 2
	t ≅ 10 s	$\{2.0,0.0,0.0\}^T$	-56 ± 2	-53 ± 2	-81 ± 2	-84 ± 2
	t ≅ 15 s	$\{3.0,0.0,0.0\}^T$	-45 ± 2	-34 ± 2	-80 ± 2	-83 ± 2

Table 3.4.3: *Estimated landmarks direction observations example for few integration time steps based on the true robot and landmarks positions (3 DOF case).*

Due to the fact that the test code simulates a 3 DOF mobile robot, some conventions about the observation vector has to be outlined. In the 3D world, a feature is represented by two angles that identify the feature direction: ϑ and φ (see Section 3.2). In the 3D case we can build a 2D environmental map so we assume, as previously outlined, that all the landmarks are in the same plane of the camera (i.e. Z=0, see the *robot reference frame* definition in Section 3.2). This observation brings to another necessary observation. Respect to the image reference frame (see definition in Section 3.2) the ϑ angle is estimated using the x and y point components (i.e. y and z coordinates respect to the robot reference frame) as follows:

$$\vartheta = \text{atan}\left(y/x\right),$$

and as all the points lie in the Z = 0 plane of the robot (i.e. Y = 0 respect to the image reference frame) the value of ϑ can be only 0 or $-\pi$ based on the sign of the x component. For this reason the direction angle ϑ can be consider not a useful observation value so for the update stage of the EKF for 3 DOF case we decided to use only the direction angle φ. The observations are used to update the EKF state vector and in this test code the landmark observation values are computed for each integration time step. Tab. 3.4.3 shows some examples of the observed direction value φ for each of the four artificial landmarks referring to three integration time steps. The values are computed using the true robot and landmark positions adding the Gaussian noise model described before.

Using the settings reported in Fig. 3.4.2 and the conventions about the observations exposed in the previous paragraph, we obtained a EKF test code for a 3 DOF mobile robot with a very high performance as shown in Tab. 3.4.4.

The obtained results are very accurate in terms of estimated position of the landmarks, with particular regards to the estimated value of Landmark 1 and 2 compare to the true values (i.e. L1 and L2 in Tab. 3.4.1). From Table 3.4.4 the reader can observe how the coordinates for all the landmarks gradually converging from the initial noisy value (see Tab. 3.4.2) to the true value (see Tab. 3.4.1). The convergence of the coordinates for the four test landmarks is not the same and the reason can be related to the error introduced into the initial value that is larger for L3 and L4 compare to L1 and L2. After 240 steps (around 20 seconds of navigation) the estimate position for L3 and L4 is still not close to the true one as for L1 and L2, but it remains compatible in terms of error interval with the true coordinates reported in Table 3.4.1. The results obtained for the implementation of the EKF [38] in a 3 DOF mobile robot are numerically shown in Tab 3.4.4 and graphically shown in Fig. 3.4.3

k	t [s]	$\{X_R, Y_R\}$ [m]	α_R [deg]	**L1** [m]	**L2** [m]	**L3** [m]	**L4** [m]
0	0	0.006±0.008 -0.0007±0.008	0.02 ±0.34	5.99±0.98 -2.39±0.46	5.01±0.98 1.88±0.43	9.70±1.00 1.14±0.36	8.37±1.00 -0.60±0.31
60	5	1.02±0.06 -0.01±0.05	0.05 ±0.65	5.58±0.78 -2.26±0.34	4.35±0.63 1.65±0.26	9.68±0.99 1.15±0.14	8.37±1.00 -0.61±0.09
120	10	2.02±0.08 0.002±0.07	0.02 ±0.78	5.19±0.36 -2.09±0.17	4.09±0.25 1.54±0.12	9.55±0.96 1.14±0.13	8.27±0.98 -0.61±0.09
180	15	3.06±0.10 0.01±0.08	-0.06 ±0.93	5.14±0.21 -2.06±0.11	4.07±0.15 1.53±0.07	9.22±0.89 1.11±0.13	7.91±0.87 -0.58±0.09
240	20	4.08±0.11 0.01±0.07	-0.11 ±1.04	5.11±0.16 -2.05±0.08	4.08±0.11 1.54±0.06	8.91±0.81 1.06±0.12	7.42±0.71 -0.54±0.08

Table 3.4.4: *Robot and landmarks positions estimated by the EKF for different time steps (mobile robot with 3 DOF).*

As we can observe from Tab. 3.4.4, the errors in the robot positions increases with the time and this is because we use the encoders as robot position sensor and the associate encoder errors increase with time (commonly known as drift) but using the EKF [38] helps to contain these errors merging the measurement observations available.

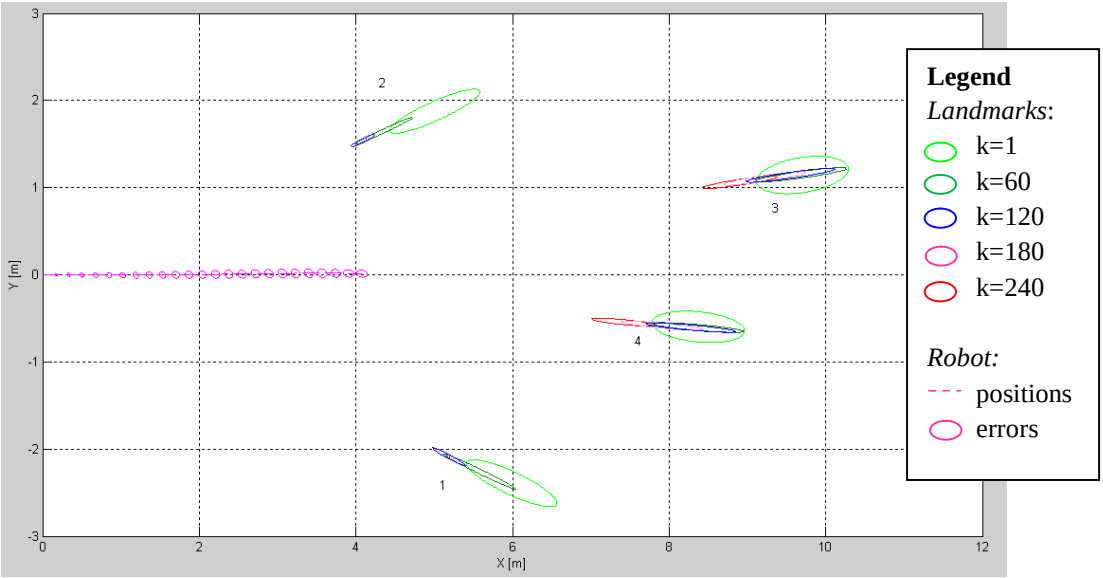

Figure 3.4.3: *2D map of the artificial test environment with robot and landmarks positions and errors.*

The results obtained for the implementation of the EKF [38] in a 3 DOF mobile robot are numerically shown in Tab 3.4.4 and graphically shown in Fig. 3.4.3. The last row is Table 3.4.4 represents the final values of robot and landmarks positions and we can see that are also the best estimation with regards to the landmarks positions if compare with the true position of Table 3.4.1. This result allows us to say that the implementation used is robust as it shows a convergence of the estimated value of the state vector (composed by the position/orientation of the robot and the landmarks positions) to the true value, with particular reference to the landmarks positions within the presence of noise (see Tab. 3.4.1 for true values and Tab 3.4.4 for the estimated values). After this implementation is then possible expand it for a 6 DOF mobile robot where the 3D coordinates of the landmarks are taken into account and a 3D map can be built. The EKF code example and results is discussed in the following section.

3.4.3 System with 6 DOF

After the development and test of the EKF for a 3 DOF mobile robot, the code is extended for the 6 DOF case. The six degrees of freedom are: 3D position in the space (i.e. x, y and z coordinates) and the orientation (i.e. γ angle around the x axis, β angle around the y axis and α angle around the z axis).

The robot starts at position $\{0,0,0,0,0,0\}^T$ and the 4 features are added to the state vector as in the 3 DOF case

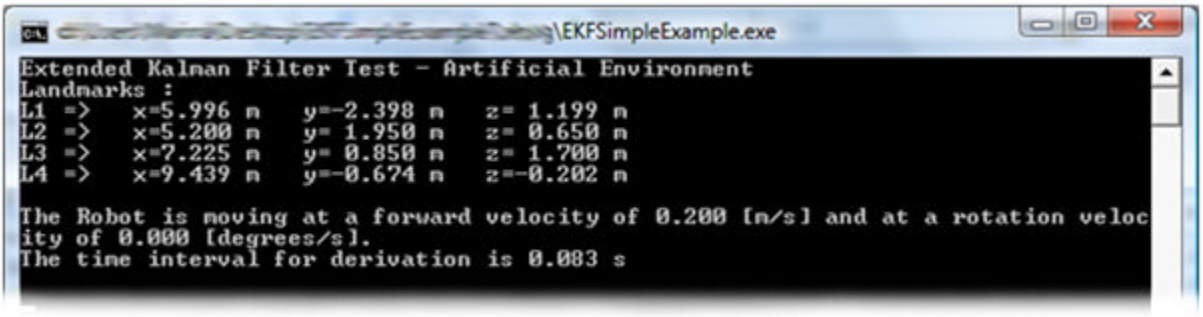

Figure 3.4.4: *Settings used to initialize the test code.*

In Fig. 3.4.4 are shown the settings used to initialize the test code and the initial 3D coordinates of the artificial landmarks.

As the 3 DOF case, the landmark observations are generated using the true position of the robot and the true position of the landmark (see Tab. 3.4.1) in the artificial test environment but this time both angles that represent the landmark direction are used. Again the initial position of a landmark is added to the EKF state vector with an error that affects the true position but not the direction (see values in Tab. 3.4.2). Some examples of the direction values for the four test landmarks are shown in Table 3.4.5 and they are estimated using the true position of the robot and the true position of the landmarks adding a Gaussian random noise.

			LANDMARKS			
		True robot position $\{X_R, Y_R, Z_R\}^T$ [m] $\{\gamma_R, \beta_R, \alpha_R\}^T$ [deg]	**L1** $\{\vartheta_1, \varphi_1\}$ [deg]	**L2** $\{\vartheta_2, \varphi_2\}$ [deg]	**L3** $\{\vartheta_3, \varphi_3\}$ [deg]	**L4** $\{\vartheta_4, \varphi_4\}$ [deg]
TIME	t ≅ 5 s	$\{1.0,0.0,0.0\}^T$ $\{0.0,0.0,0.0\}^T$	-26 ± 2 -61 ± 2	198 ± 2 -62 ± 2	243 ± 2 -73 ± 2	17 ± 2 -85 ± 2
	t ≅ 10 s	$\{2.0,0.0,0.0\}^T$ $\{0.0,0.0,0.0\}^T$	-27 ± 2 -53 ± 2	199 ± 2 -52 ± 2	243 ± 2 -71 ± 2	17 ± 2 -84 ± 2
	t ≅ 15 s	$\{3.0,0.0,0.0\}^T$ $\{0.0,0.0,0.0\}^T$	-26 ± 2 -42 ± 2	198 ± 2 -32 ± 2	244 ± 2 -68 ± 2	17 ± 2 -82 ± 2

Table 3.4.5: *Estimated landmarks direction observations example for few integration time steps based on the true robot and landmarks positions (6 DOF case).*

The results for the 6 DOF case are again very accurate in terms of estimation of the true landmark position and we obtained better result in terms of fast convergence to the true coordinates for the Landmark 1 and 2. After 240 steps (around 20 seconds of navigation) the

estimate positions for L3 and L4 are not the real one (reported in Tab. 3.4.1), but it remains compatible, in terms of error interval, with their coordinates.

k	t [s]	$\{X_R, Y_R, Z_R\}$ [m]	$\{\gamma_R, \beta_R, \alpha_R\}$ [deg]	**L1** [m]	**L2** [m]	**L3** [m]	**L4** [m]
0	0	0.000±0.008 0.000±0.008 0.000±0.014	0.00±0.53 0.00±0.53 -0.00±0.36	6.00±1.51 -2.40±0.61 1.20±0.32	5.20±1.58 1.95±0.60 0.64±0.22	7.22±1.78 0.85±0.24 1.70±0.46	9.44±1.98 -0.68±0.33 -0.20±0.13
60	5	1.01±0.06 -0.002±0.05 -0.003±0.050	0.07±1.04 -0.08±0.66 0.06±0.57	5.34±0.82 -2.15±0.36 1.08±0.19	4.24±0.66 1.60±0.28 0.54±0.11	7.50±1.52 0.87±0.19 1.75±0.38	9.39±1.96 -0.68±0.16 -0.20±0.10
120	10	2.01±0.08 0.008±0.07 -0.005±0.049	0.38±1.18 -0.22±0.66 0.13±0.66	5.12±0.33 -2.06±0.16 1.03±0.10	3.99±0.25 1.50±0.12 0.51±0.07	7.99±0.94 0.93±0.13 1.88±0.24	9.17±1.81 -0.67±0.15 -0.19±0.10
180	15	3.01±0.10 0.03±0.08 -0.004±0.050	0.61±1.28 -0.42±0.78 0.08±0.72	5.09±0.20 -2.05±0.10 1.02±0.08	3.98±0.15 1.49±0.07 0.51±0.06	8.19±0.60 0.96±0.09 1.93±0.16	8.26±1.33 -0.61±0.13 -0.17±0.09
240	20	4.01±0.11 0.02±0.07 0.004±0.052	0.47±1.28 -0.41±0.98 0.12±0.70	5.03±0.15 -2.02±0.08 1.01±0.07	4.01±0.12 1.51±0.06 0.51±0.06	8.34±0.45 0.98±0.07 1.97±0.12	7.54±0.80 -0.55±0.09 -0.15±0.08

Table 3.4.6: *Robot and landmarks positions estimated by the EKF for different time steps (mobile robot with 6 DOF).*

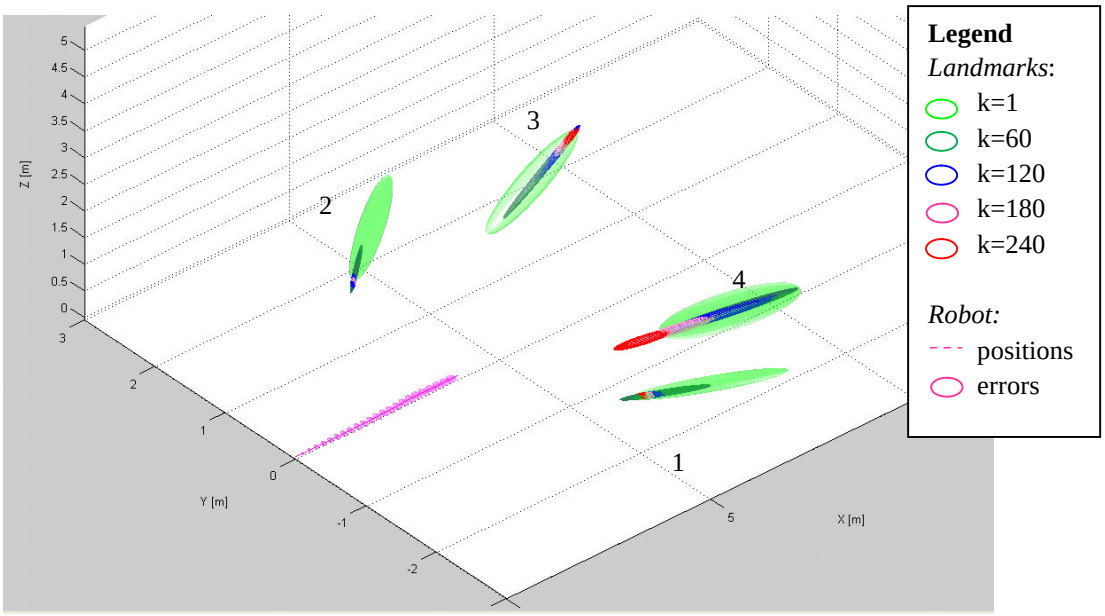

Figure 3.4.5: 3*D map of the artificial test environment with robot and landmarks positions and errors.*

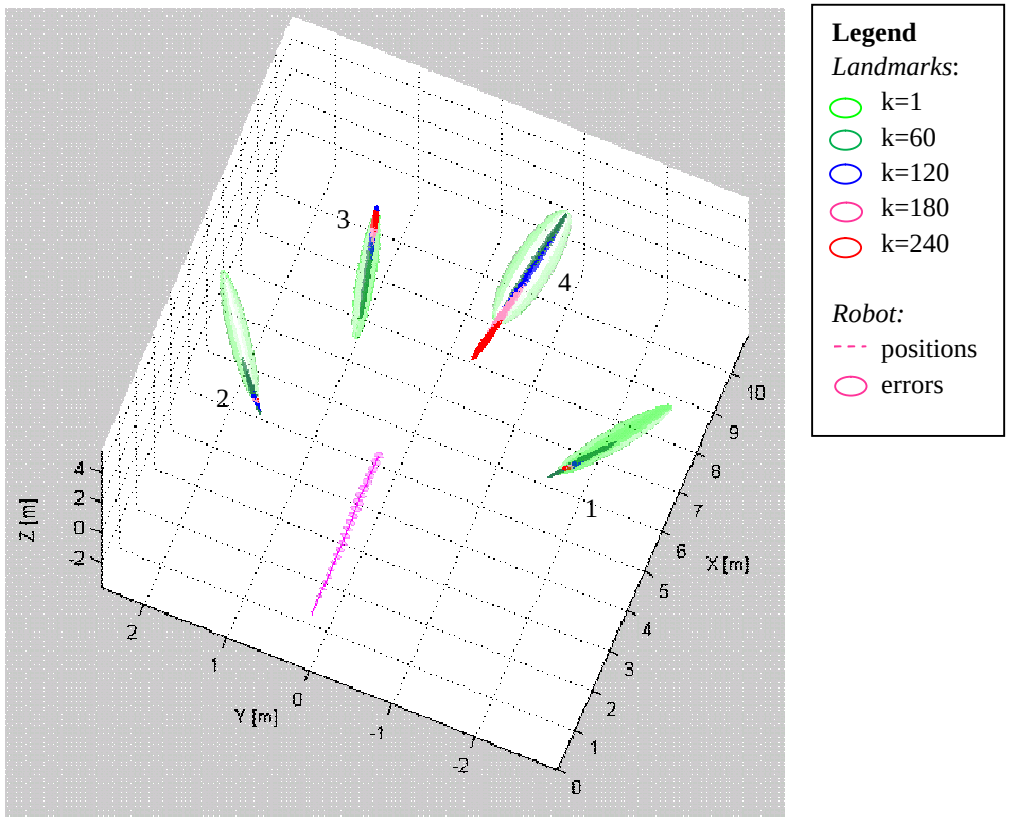

Figure 3.4.6: *Another view of the* 3*D map of the artificial test environment with robot and landmarks positions and errors*

The results obtained for the implementation of the EKF [38] in a 6 DOF mobile robot are numerically shown in Tab 3.4.6 and graphically shown in Fig. 3.4.5-3.4.6. From Tab. 3.4.6 we can observe that the error in the robot position still increase with the time and the reason is again related to the use of encoders as robot position sensor. The last row is Table 3.4.6 represents the final values of robot and landmarks positions and are also the best estimation with regards to the landmarks positions if compare with the true position of Table 3.4.1.

Comparing the results obtained for the 3 DOF (Tab. 3.4.4) and for the 6 DOF (Tab. 3.4.6) the reader can see that also if the initial error of the landmark positions are larger in the 6 DOF is larger, the EKF is able to manage the data giving similar final results for both cases.

After these EKF code examples it is now possible to introduce the implementation of the EKF to the wide system (see Fig. 3.1.1) and integrate all modulus together (see Section 3.1).

Chapter 4

Equipment and Environment

In this chapter we are going to describe the equipments used during the software evaluation and the type of environments analyzed. As described in Section 1.2 the SLAM problem is a very wide research topic and it can be expressed as follows:

"*The Simultaneous Localization And Mapping (SLAM) problem asks if it is possible for a mobile robot to be placed at an unknown location in an unknown environment and for the robot to incrementally build a consistent map of this while simultaneously determining its location within this map.*" [30].

The realization of a SLAM problem solution needs several sensors and tools. As mentioned in Section 2.5 we use encoders and GPS receiver as robot position sensors. The environment sensors used in this project are an optical and a thermal camera. In addition we use a laptop to manage all the sensors and the information from them mounted on the mobile robot. Indoor and outdoor environments are used to test the overall implementation.

In the follow sections the reader can find specifications about the different sensors, equipment used and information about the test environments.

4.1 Mobile Robot

The mobile robot is a Pioneer 3-AT produced by Mobile Robots Inc.. It has a sturdy aluminium body, balanced drive system, reversible DC motors, four wheel drive, high resolution motion encoders, sonar arrays and battery power, all managed by an onboard microcontroller and mobile-robot server software.

Figure 4.1.1: *Image of the Pioneer 3-AT in its standard configuration.*

The software includes the Advanced Robotics Interface for Applications (ARIA, developed by Mobile Robots Inc.) released under the GNU Public License. This software is used as an interface to the control aspect of the robot.

The robot is also equipped with a joystick connector that allows the user to manually control the mobile robot directly without the use of explicit control software. The robot is then equipped with an optical and a thermal camera, a GPS receiver and a laptop to manage all data. The final system is shown in Fig. 4.1.2a-b. More details about the sensors and the equipment are presented in the following section.

(a)

(b)

Figure 4.1.2: *Pioneer 3-AT in its final configuration used in the project.*

4.2 Optical and Thermal Cameras

The visual sensors consist in two cameras: a CCTV optical camera and a thermal camera. The optical camera is a Visionhitech VC57WD-24 and shown in Fig. 4.2.1. It is equipped with an manual zoom lens and focus. The reader can refer to Table 4.2.1 for camera specifications.

Figure 4.2.1: *Optical camera VC57WD-24 with zoom lens.*

Figure 4.2.2: *Thermal camera MIRICLE 110K.*

The thermal camera is a Thermoteknix MIRICLE 110K as shown in Fig. 4.2.2. Technical details about the thermal camera are reported in Table 4.2.1.

	VC57WD-24	MIRICLE 110K
Imager	1/3'' Sony Double Scan CCD	---
Pixels	640H x 480V	640H x 480V
Spectral response	---	7 – 14 μm
Response time	---	7 ms
Operation Temperature	-10°C to 50°C	-20°C ± 50°C
Humidity	Within 90% RH	5 – 95 % non condensing
Dimensions [mm] (Width x Height x Length)	62 x 58 x 140	42 x 40 x 40

Table 4.2.1: *Specifications for the optical and thermal cameras used in the project.*

4.3 GPS receiver

Another external sensor is the GPS used to measure the position of the robot. The GPS used is a GlobalSat BU–353 with a USB connector.

Figure 4.3.1: *GlobalSat BU–353 receiver*.

4.4 Data management

Data management is performed by a laptop installed on the mobile robot and connected to a driver interface via a RS–232 serial port. The laptop managers all the information from the robot sensors (using the serial port interface) and the GPS (connected by a USB port) controlling the motors of the robot, its velocity and trajectory. The cameras use a USB interface port for the optical camera and a FireWire interface port for the thermal camera.

Figure 4.4.1: *DELL™ Latitude™ E6400 ATG*.

The laptop is a DELL™ Latitude™ E6400 ATG running the Microsoft Windows XP Operating System and uses a Intel Core Duo CPU (P8700, 2.53 GHz) and 3.45 GB of RAM.

4.5 Indoor and Outdoor environments

The overall system is tested in both indoor and outdoor environments. The techniques used to solve the SLAM problem are developed with reference mainly to an outdoor environment but adjustment of some parameters, such as the minimum and maximum depth constraint of a feature (see Section 3.2), the software implementation can be easily used in an indoor scenario.

For the indoor environment the minimum depth used in the initialization process is 0.5 m for the optical camera and 1.5 m for the thermal camera (due to a larger depth of the field of view of the thermal camera with respect to the optical one). For the outdoor environment we similarly used a minimum depth value of 1 m for the optical camera and 2 m for the thermal camera.

The depth of the field of view of the optical camera is not as large as the thermal camera. Whilst the thermal camera can detect features of a minimum distance of around 1.5-2 m, the optical camera can actually detect features at a distance of 0.5 m. In the outdoor environment the feature are further away than in an indoor environment and this requires us to increase the minimum depth.

Figure 4.5.1: *Indoor environment examples used to test the software.*

The indoor environment used is a narrow corridor inside the Applied Mathematics & Computing (AMAC) group (Whittle Building, Cranfield University). Images taken from the video captured during the tests indoor are shown in Fig. 4.5.1 for the optical camera and the thermal camera.

Figure 4.5.2: *Outdoor environment examples used to test the software (optical camera).*

The outdoor environment refers to different parts of the Cranfield University campus. In Fig. 4.5.2 and Fig 4.5.3 are shown some examples of the outdoor environment used.

Figure 4.5.3: *Outdoor environment examples used to test the software (thermal camera).*

From the optical camera a wide range of features with different depths can be detected and this is thanks to the details in the images and the large field of view of the camera. From the thermal camera a variety of features can still be detected as people and part of building that give an amount of heat very obvious but in general there is a less feature density than in the optical images. However it has to be noticed that the density of the available optical features is dependent on the day light condition whereas the thermal features are largely constant and dependent on the thermal dynamic of the environment rather than time of the day.

Chapter 5

Results and Discussion

In this chapter we are going to show and discuss the results obtained. Some of the following examples are going to show the different results between the use of just either of the thermal and optical cameras or the combined use of them in the SLAM system.

The combined system uses the optical and thermal cameras simultaneously but each of them is treated as an independent sensor. During the navigation the optical camera extracts optical feature points and the feature initialization and update process of Section 3.2 is applied. At the same instance, the thermal camera extracts thermal feature points and transforms them from the thermal camera reference frame to the optical camera reference frame via the homography matrix described in Section 3.3.4. After this transformation the thermal feature points are initialized and updated, as for the optical feature points, using the technique described in Section 3.2. In the combined system during the feature initialization and updating process (Section 3.2) the feature point information from the optical and thermal cameras are treated separately. This means that the optical feature points and the thermal feature points are stored in two separate databases. However, as soon a feature point (optical or thermal) is marked as a landmark is store in a unique landmark database managed by the EKF describe in Section 2.4.

All the cases analyzed are presently processed *offline* and the *Zmax* for each new landmark (respect the first frame when they are seen the first time) is set to 3 m for the indoor environment and 5 m for the outdoor analysis in order to avoid unusable landmarks in the final map (for navigation purposes) decreasing the computational cost necessary to manage a map with a large amount of landmarks.

As outlined in Section 3.2, each new feature is initialized with a sum of Gaussians and this procedure is shown in Fig. 5.1 for five features, three thermal features and two optical features for an outdoor environment.

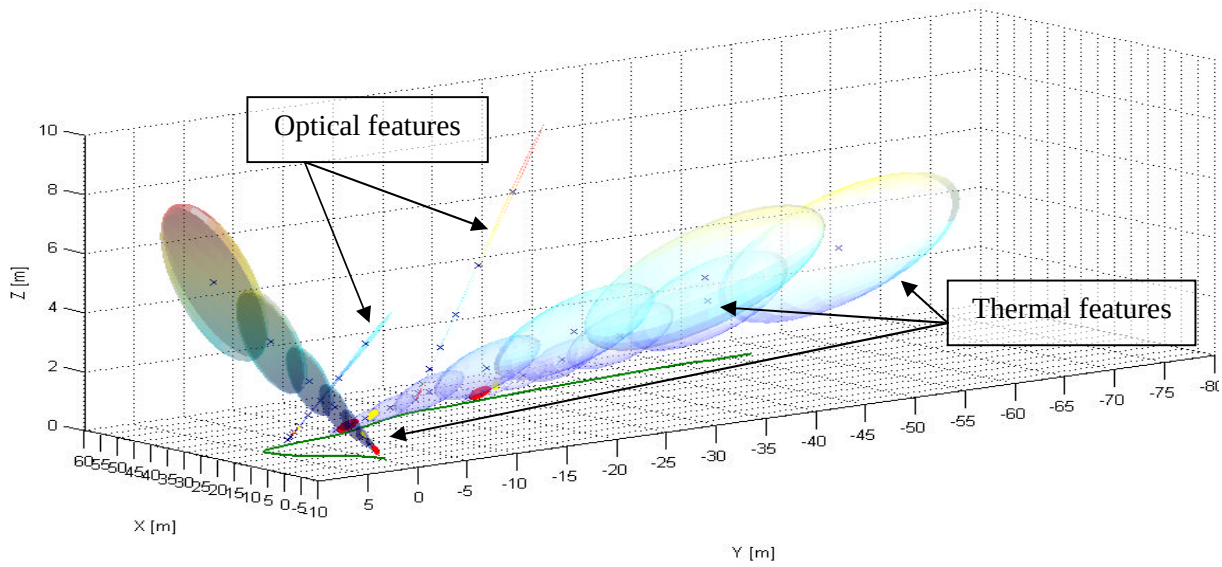

Figure 5.1: *Feature initialization – landmark selection and estimation related to the robot positions (green track).*

Each Gaussian represents a possible position of the feature in the space and it is represented by the mean (i.e. possible position of the point) and by a deviation (i.e. covariance matrix that represents the errors). In Fig. 5.1 each Gaussian is represented as an ellipsoid of error and the orientation is related to the first direction where the feature was seen the first time. After few observations it is possible to select the initial values for the associate landmark (i.e. red ellipses in Fig. 5.1) and successively update its position obtaining a better estimation (i.e. yellow ellipses in Fig. 5.1) using the EKF algorithm.

As the reader can observe from Fig. 5.1, the thermal features have larger error ellipses and this is related to the parameters used to initialize the feature points such as the minimum ray ρ_{min} and the transformation needed to convert the thermal point from the thermal camera image plane to the optical camera reference frame (i.e. homography matrix H, see Section 3.3).

5.1 Indoor Environment

The first environment tested is an indoor scenario. The main difference in the algorithm for an indoor scenario with respect to an outdoor environment is the initialization value of the minimum ray ρ_{min} used to compute the sum of Gaussians (see Section 3.2). Another difference between the two scenarios is the amount of features generally available with particular reference to thermal features.

For this example, the thermal and optical camera videos are first evaluated separately and then the information is merged together in the combined optical-thermal SLAM system in order to compare the performance of the single camera case with the system that uses both camera sensors.

Optical camera

In Fig. 5.1.1 is shown a typical output video frame when just the optical camera is used. The typical output video shows the current image frame and some information about the features, landmarks and state vector on the left.

Figure 5.1.1: *Example of the output when analyzing the optical video (indoor).*

In the current frame shown in the output video the feature/landmark that matches a point in the correspondent dataset are shown. The landmarks are marked in cyan colour whilst the features are visualized in magenta colour (see Fig. 5.1.1). On the left side of the output video are visualized the number of features stored in the *features database*, the number of landmarks in the *landmark database* and the dimension of the EKF state vector. The EKF

state vector contains the current robot position, the past *n* robot positions (not subject to update) and the *m* landmarks in the environmental map.

The results of the analysis are exported in Command Separated Values (CSV) format simplifying the post process in Matlab (i.e. plotting the 2D and 3D maps). Fig. 5.1.2 shows the 2D environmental map obtained for the indoor example.

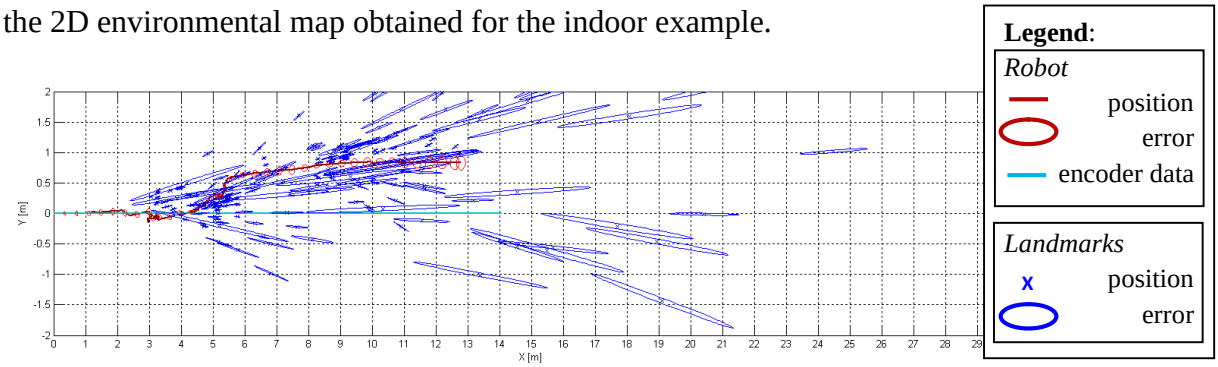

Figure 5.1.2: *2D environmental map for the optical video analysis (indoor).*

As shown in the 2D map (see Fig 5.1.2) a significant number of landmarks are detected but not all of them are correctly estimated. As consequence the robot position estimation is affected by a drift with respect to the ideal horizontal line (i.e. encoder data Fig. 5.1.2) that is actually the set track for the mobile robot. This drift can be better seen in Fig. 5.1.3 (i.e. 3D map) where the *z* coordinate of the optical camera is estimated to be above the $Z = 0$ plane. From Fig. 5.1.2 and Fig. 5.1.3 we can then observe that some landmark positions are estimated with a smaller positional error than others illustrated by the smaller ellipse in the resulting plots.

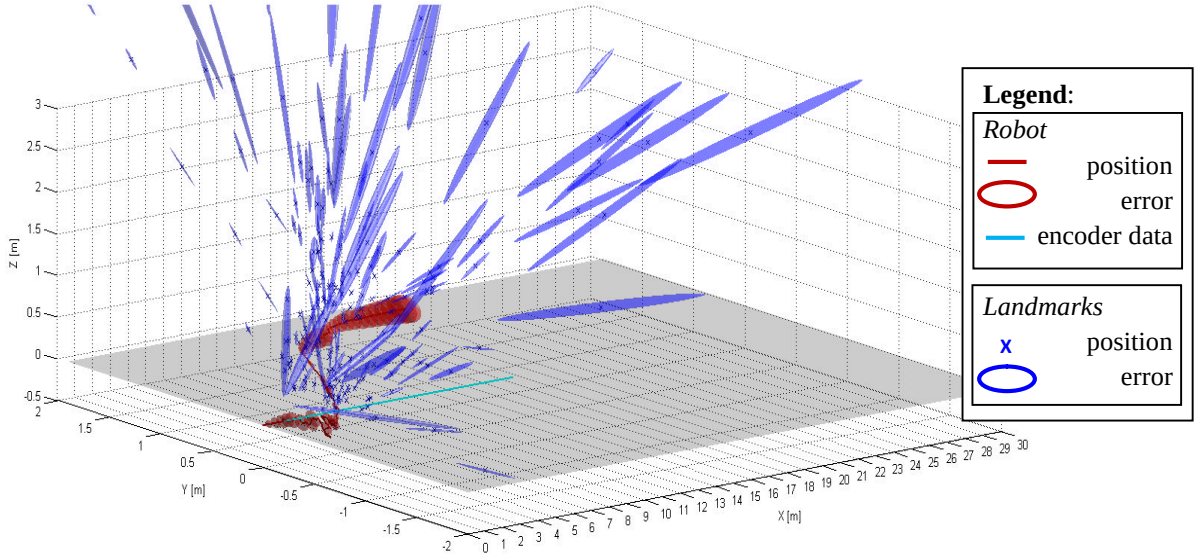

Figure 5.1.3: *3D environmental map for the optical video analysis (indoor).*

For the optical video analysis the system detects 192 landmarks with an average error for the position of 0.93 m, 0.13 m and 0.22 m along the *x*, *y* and *z* axis respectively.

Thermal camera

The same analysis is made for thermal camera imagery. The amount of details is less than in the optical images and the field of view is deeper as previously mentioned in Section 4.5 and as a consequence less features/landmarks can be detected. An example of the video output is shown in Fig. 5.1.4.

Figure 5.1.4: *Example of the output when analyzing the thermal video (indoor).*

In an indoor environment there is not the presence of a large amount of thermal features unless we are in a presence of lights, radiators, or sources of heat. In this indoor example for the thermal single case no landmarks are detected.

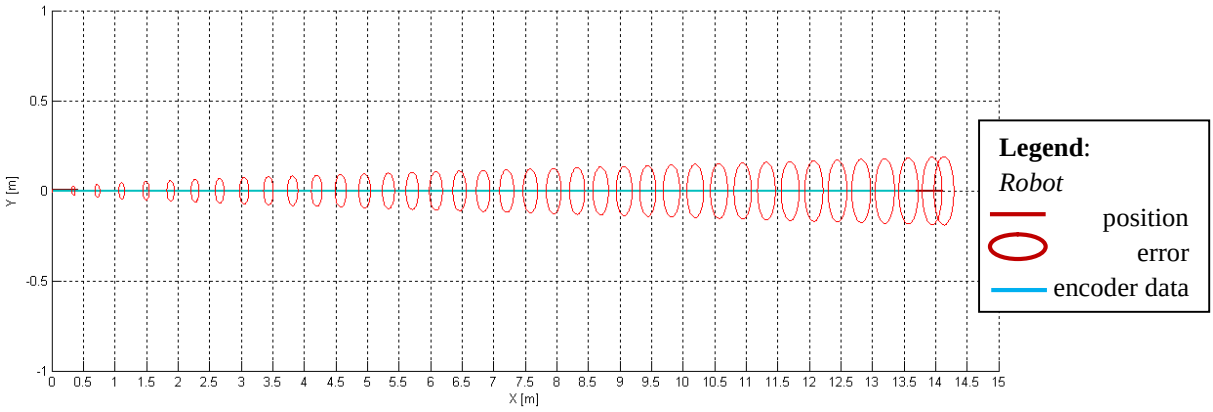

Figure 5.1.5: *2D environmental map for the only thermal video (indoor).*

Fig. 5.1.5 and Fig. 5.1.6 show the 2D and 3D maps for the thermal case and, as no thermal landmarks are added to the environmental map, the position of the robot is updated just using the information from the encoders.

Comparing the 2D map obtained from the thermal video with the one obtained from the optical video we can observe how the estimated robot positions are different and the main reason is related to some wrong initialization or matching of landmarks in the optical single case. For this example, as the robot route is quite short (i.e. the encoder drift starts to affect the position measurements just after a long navigation in terms of time) we can consider the estimated robot positions in the thermal single case a good estimation to use as a reference.

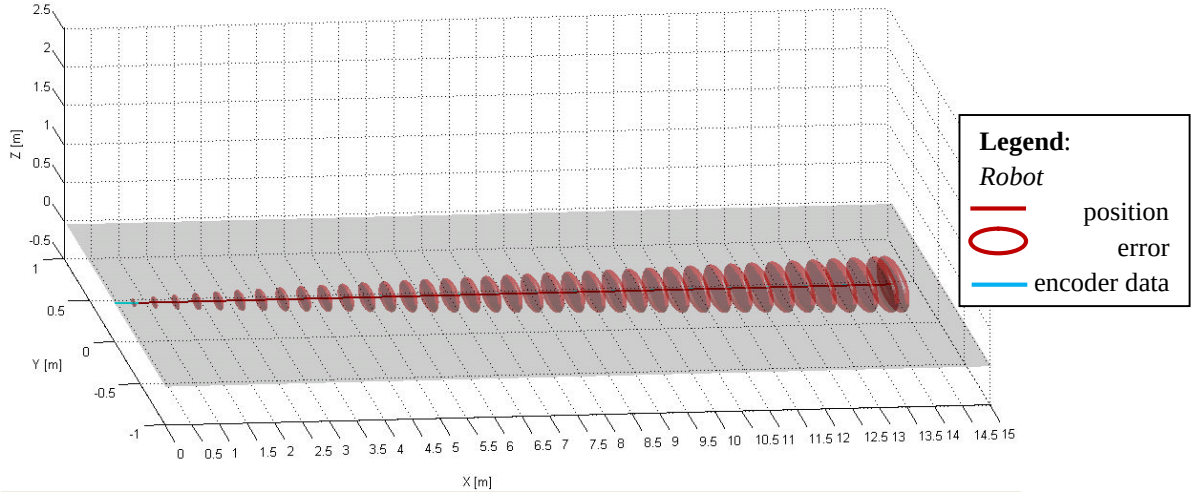

Figure 5.1.6: *3D environmental map for the thermal video analysis (indoor).*

Optical and Thermal cameras

The final stage of the analysis is to study both video together combining the information of the extracted features in a unique *landmark database* with reference to the optical camera reference frame. Fig. 5.1.7 shows a typical output video frame for this analysis.

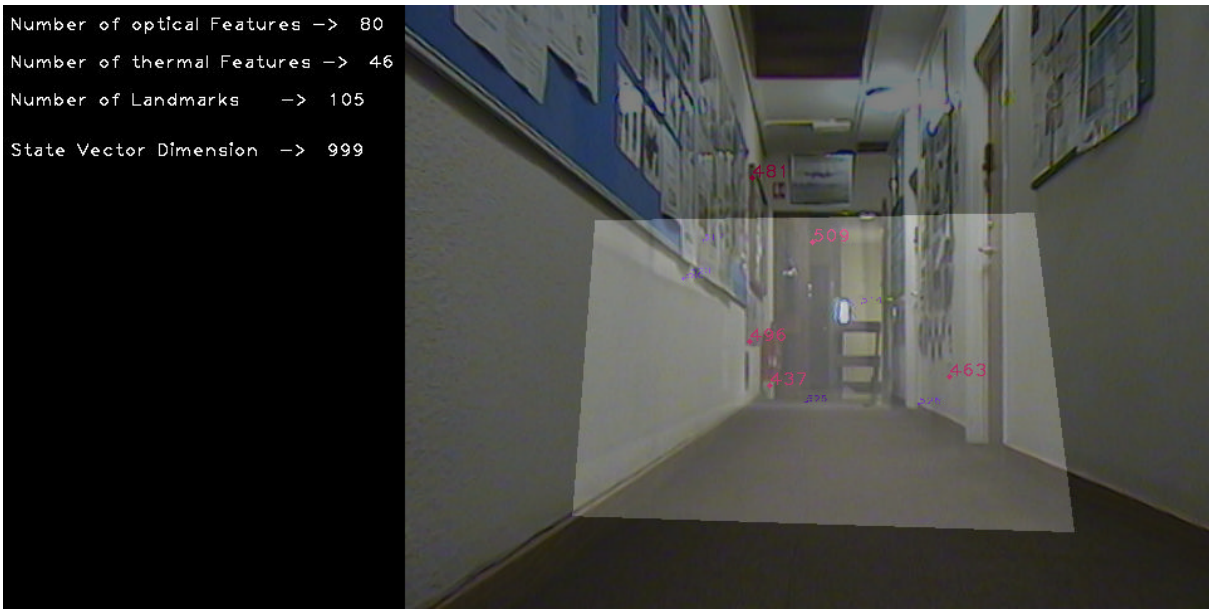

Figure 5.1.7: *Example of the output for the combined system (indoor).*

As in the single analysis, the visual output for the combined system shows the number of features (optical and thermal), the number of landmarks in the environmental map, the dimension of the EKF state vector on the left side of the window and the current combination of frames obtained (as described in Section 3.3) on the right side. Fig 5.1.8 shows the optical and thermal images correspondent to the video output of Fig. 5.1.7.

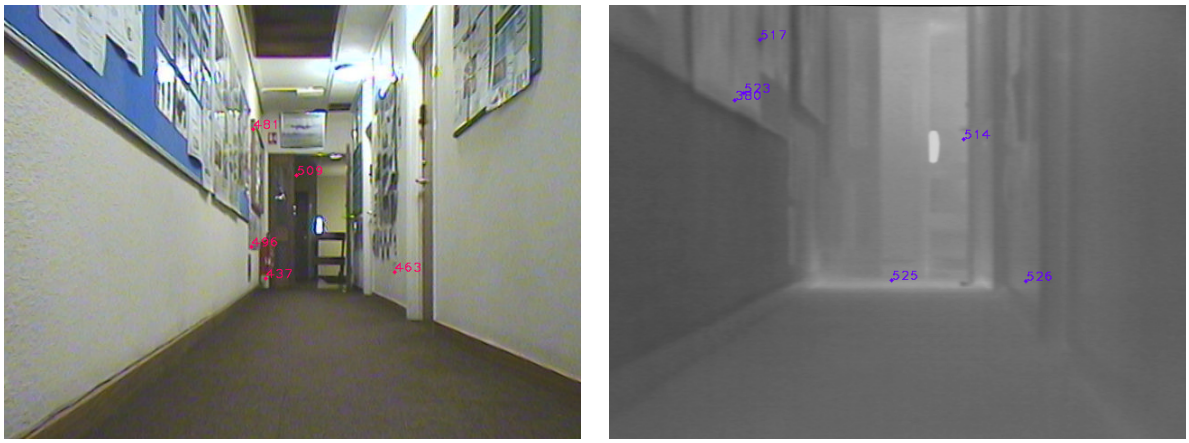

Figure 5.1.8: *Optical and Thermal video frame of Figure 5.1.7.*

The combination of the two video adds information to the environmental map as some thermal landmarks are detected. Another result is related to the robot position that is better estimated with respect to the optical single case (respect the same interval time steps). However, the field of view of the thermal camera is too narrow to allow the system to detect an amount of thermal landmarks similarly to the optical ones. The 2D map of the optical – thermal case is shown in Fig. 5.1.9.

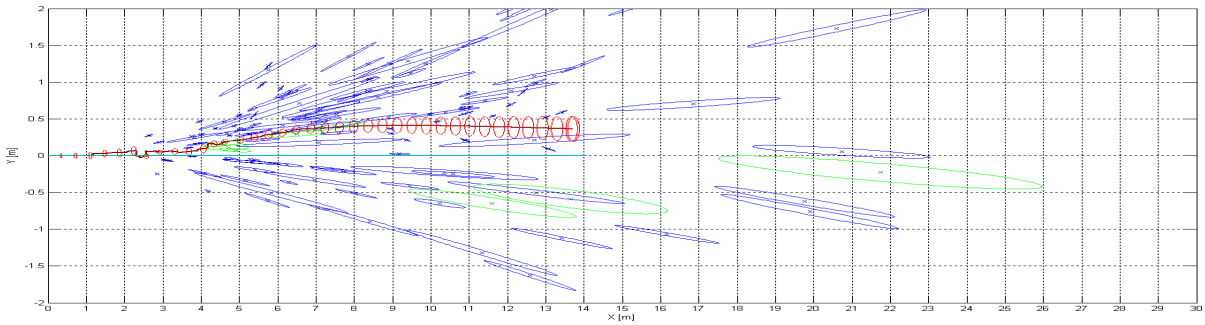

Figure 5.1.9: *2D environmental map for the combined system (indoor).*

Comparing the 2D (or 3D) map obtained from the single optical case with the one obtained from the combined system, we can tell that the number of optical landmarks extracted is similar to the optical single case. However a relevant difference can be observed if comparing the combined system with the thermal single case. In the analysis of the thermal only video

sequence no landmarks are detected whereas in the combined system a few thermal landmarks are added into the environmental map. The reason for this has to be attributed to the estimation of the EKF state vector (i.e. the robot position estimation).
As the feature/landmark coordinates are always related to the robot position, a small difference in its estimation can change the landmark detection and as a consequence can allow the system to detect more landmarks as in this analyzed case with reference to the thermal landmarks.

In Fig. 5.1.10 the 3D environmental map is represented and the plane $Z = 0$ is added to the plot. In this indoor example are detected 131 optical landmarks and 8 thermal landmarks. The overall amount of landmarks is smaller than the optical only case but combining the two video inputs together the system is able to detect thermal landmarks where it was not able before (i.e. in the thermal only case). This is because the feature-landmark position is strictly related to the robot position so a change in the robot position estimation can allow the combined system to detect new landmark with respect to the only camera case.

The average errors of the landmark position are 1.30 m, 0.18 m and 0.31 m along the *x*, *y* and *z* axis respectively. These errors are a bit larger than the average error for the optical only case, but the introduction of the thermal video and thermal landmarks improve the estimation of the robot position as we can see from Tab.5.1.1 – 5.1.4.

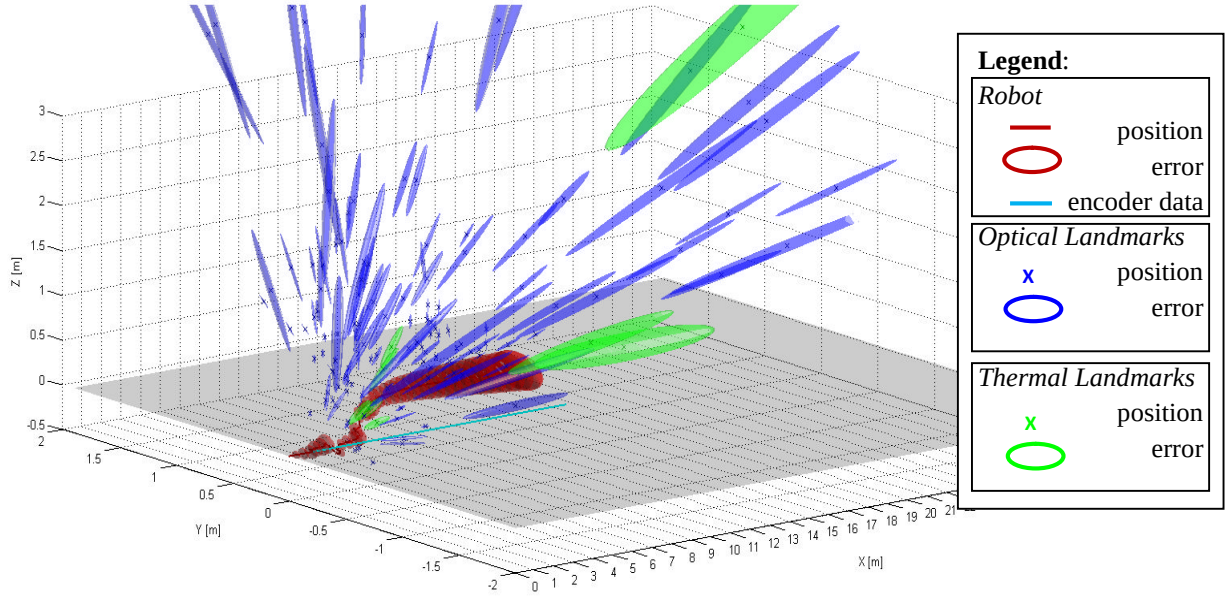

Figure 5.1.10: *3D environmental map for the combined system (indoor).*

In Table 5.1.1 – 5.1.4 are reported some numerical output obtained from both the single camera analysis cases and from the combined system. After 300 integration time steps (see

Tab. 5.1.1) the results obtained for the three analyzed cases in this indoor example are not completely compatible in terms of intervals of error. This incompatibility of the results is a consequence of incorrect optical landmarks position estimation as we can graphically see from Fig. 5.1.9, where several landmarks have a value of ± 1 metre along the *y* axis of the global reference frame. These values are clearly incorrect as the mobile robot is placed in the centre of a narrow corridor with a distance of ± 0.60 metre from the side walls.

During the analysis we could observe how several features/landmarks were wrongly matched between successive frames.

k=300	**Combined Optical and Thermal**	**Optical only**	**Thermal only**
X [m]	3.24 ± 0.12	3.31 ± 0.11	3.82 ± 0.14
Y [m]	0.18 ± 0.08	-0.05 ± 0.06	0.00 ± 0.14
Z [m]	0.13 ± 0.10	-0.04 ± 0.08	0.00 ± 0.17
γ [deg]	-3.55 ± 1.80	-1.13 ± 1.18	0.00 ± 3.49
β [deg]	-5.50 ± 1.31	-3.03 ± 0.70	-0.04 ± 3.49
α [deg]	2.99 ± 1.26	4.31 ± 0.64	-0.01 ± 1.75

Table 5.1.1: *EKF state vector for the final system, the optical single case and the thermal single case respectively after k = 300 frames (i.e.25 seconds) – (indoor).*

From Tab. 5.1.2 we can observe a larger incompatibility in terms of error interval among the optical, thermal and combined cases. As the robot is moving forward in a limited area and for a small amount of time the encoder drift of the thermal single analysis can be ignored, permitting the use of the results obtained as a ground truth reference (no landmarks are detected during the navigation so the mobile robot position is updated just using the encoder data). Comparing the results for the optical only case of Tab. 5.1.2 with the thermal only case (used in this example as a ground truth reference) we can observe how the *Y* and *Z* robot position values for the optical case are clearly wrong. However, the combined system results of Tab. 5.1.2 are compatible in terms of interval error with the thermal single analysis showing a better robot position estimation compare to the optical single case.

k=600	**Combined Optical and Thermal**	**Optical only**	**Thermal only**
X [m]	7.00 ± 0.18	5.94 ± 0.11	7.58 ± 0.19
Y [m]	0.16 ± 0.12	0.63 ± 0.08	0.00 ± 0.21
Z [m]	0.26 ± 0.14	0.42 ± 0.08	0.00 ± 0.25
γ [deg]	-3.87 ± 2.46	-1.09 ± 2.02	0.00 ± 4.92
β [deg]	-9.02 ± 1.70	-5.87 ± 1.48	-0.06 ± 4.92
α [deg]	0.31 ± 1.37	6.45 ± 1.55	-0.01 ± 2.45

Table 5.1.2: *EKF state vector for the final system, the optical single case and the thermal single case respectively after k = 600 frames (i.e.50 seconds) – (indoor).*

After 900 integration time steps (see Tab 5.1.3) the optical only case results are not completely compatible in terms of error intervals with the other two cases whilst the combined system is compatible in terms of error intervals with the thermal single analysis among all the entries (i.e. robot position and orientation).

k=900	**Combined Optical and Thermal**	**Optical only**	**Thermal only**
X [m]	10.84 ± 0.23	9.83 ± 0.20	11.32 ± 0.24
Y [m]	0.15 ± 0.22	0.84 ± 0.13	0.00 ± 0.27
Z [m]	0.26 ± 0.22	0.52 ± 0.14	0.00 ± 0.30
γ [deg]	-4.12 ± 4.18	-3.48 ± 3.84	0.00 ± 6.01
β [deg]	-9.24 ± 3.71	-8.55 ± 3.24	-0.05 ± 6.01
α [deg]	0.21 ± 2.94	1.86 ± 2.80	-0.01 ± 2.99

Table 5.1.3: *EKF state vector for the final system, the optical single case and the thermal single case respectively after k = 900 frames (i.e.75 seconds) – (indoor).*

In Tab. 5.1.4 is shown the final result obtained in this example for an indoor environment. In the final time step the combined system shows a better robot position estimation compared to the optical single analysis and is still comparable in terms of error intervals with the thermal single case whilst the optical single case shows a robot position estimation which is not completely comparable. The reason can again be attributed to the erroneous initialized position for some landmarks as we could observe in Fig. 5.1.9 and this erroneous landmark positions appear to add incorrect information (during the update stage) into the EKF algorithm. This behaviour is related to the non-uniqueness of the environment where the

system is tested (e.g. blue board along the left wall, see Fig. 5.1.11) and also to the small scenario where the system is tested as an indoor environment.

k=1115	**Combined Optical and Thermal**	**Optical only**	**Thermal only**
X [m]	13.58 ± 0.26	12.65 ± 0.24	14.01 ± 0.27
Y [m]	0.14 ± 0.29	0.83 ± 0.19	0.00 ± 0.32
Z [m]	0.26 ± 0.26	0.53 ±0.20	0.00 ± 0.33
γ [deg]	-4.19 ± 5.10	-3.70 ± 4.69	0.00 ± 6.69
β [deg]	-9.30 ± 4.73	-8.39 ± 3.02	-0.05 ± 6.69
α [deg]	0.06 ± 3.31	0.61 ± 2.96	0.00 ± 3.32

Table 5.1.4: *Final value of the EKF state vector for the final system, the optical single case and the thermal single case respectively (k = 1115 frames, i.e. about 93 seconds) – (indoor).*

In this first example we analyzed an automatic drive of the robot of the mobile robot and the obtained results for the combined system can be considered successfully in terms of number of optical and thermal landmarks extracted with reference to the single cases. Adding the thermal features information to the optical ones allows the system to extract few thermal landmarks where the thermal single analysis could not. As the reader can see from Fig. 5.1.9 and Fig. 5.1.10 also if thermal landmarks are detected, the number of them is not close to the number of optical landmarks extracted and the corresponding positioning error remains larger also after few successive observations. The low number of thermal landmarks is related to the nature of the environment itself: an indoor environment generally has fewer thermal features compared to an outdoor scenario. Finally, the correlated error of a thermal landmark position is by definition initially larger than an optical landmark (as exposed in the introduction of this section) so if the landmark is not seen for a reasonable amount of time, the error cannot be strongly decreased and this makes the landmarks not particularly useful for navigation purpose.

Overall, if comparing the optical and thermal single cases with the combined system we can say the use of the thermal camera adds a positive contribution into the estimation of thermal landmarks and as a consequence permits a better estimation of the robot position over time.

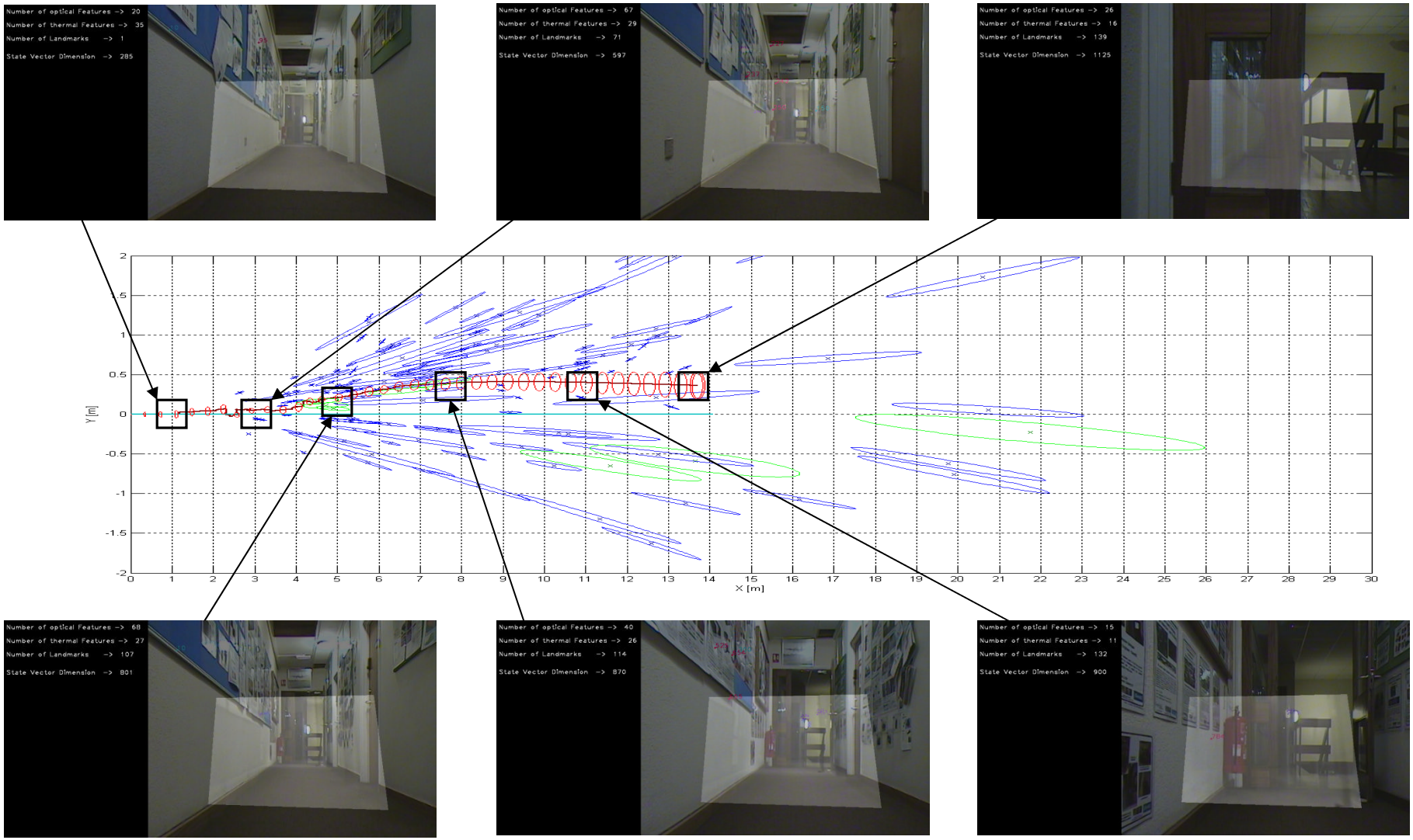

Figure 5.1.11: *2D environmental map with captured frames along the track (indoor).*

5.2 Outdoor Environment

Example 1

The main environment where we aim to use the mobile robot is an outdoor environment so we test the system over several parts of the *Cranfield University* campus. The first outdoor example refers to a non automatic navigation. This means that the robot is moved using the joystick and uses the information from the encoders to recover the translation and rotation velocity to use in the prediction stage of the EKF. The obtained results for this first outdoor example are shown in the successive paragraphs.

Optical camera

In Fig. 5.2.1 is shown a typical output video frame for the optical and is the same described for the indoor analysis.

Figure 5.2.1: *Example of the output when analyzing the optical video (example 1).*

The output of the analysis is saved in Command Separated Values (CSV) format simplifying the post process in Matlab (plotting the 2D and 3D map with robot and landmarks positions). Fig. 5.2.2 shows the 2D environmental map obtained for this first example.

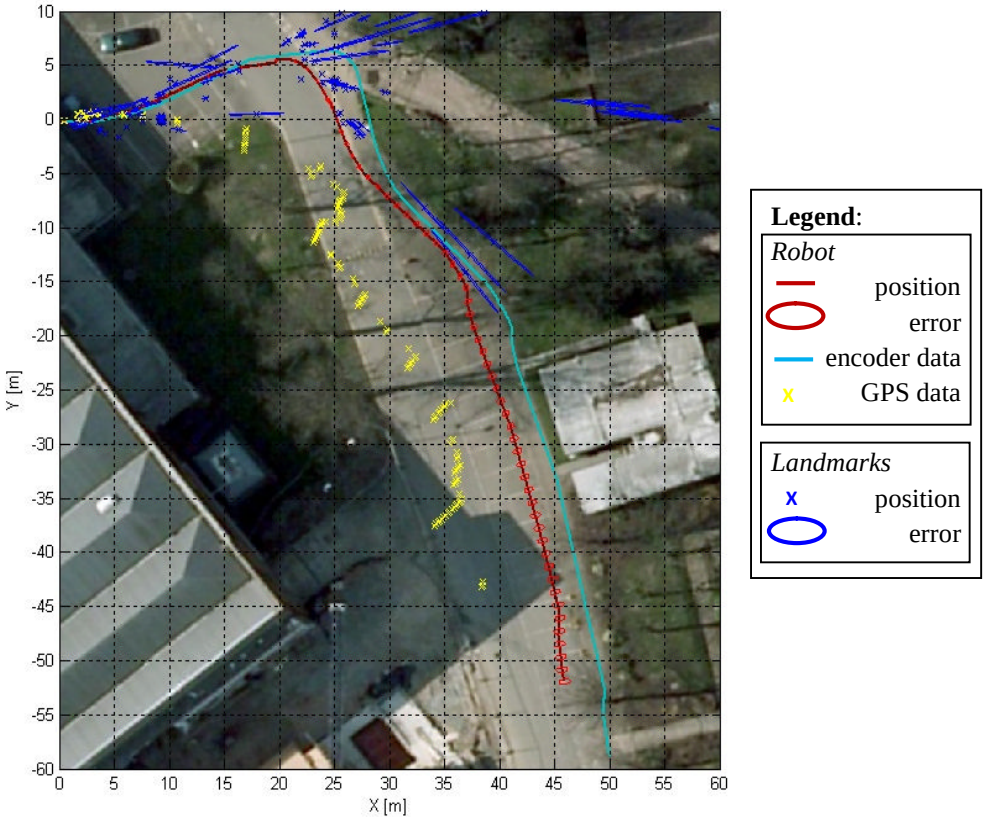

Figure 5.2.2: *2D environmental map for the optical video analysis (example 1).*

For all the outdoor examples a real map is added from Google Maps [44] (the scale matches the dimension of the graph) into the 2D environmental map and the result is shown in Fig. 5.2.2. As we can see from Fig. 5.2.2 (and in Fig. 5.2.3 for the 3D map) the position of the robot recovered from the GPS data is far from the information extracted from the encoders and we expected that (see Paragraph 2.5.2). The optical landmarks detected in this single video analysis are 129 with average errors of 1.46 m, 0.60 m and 0.43 m along the *x*, *y* and *z* axis respectively.

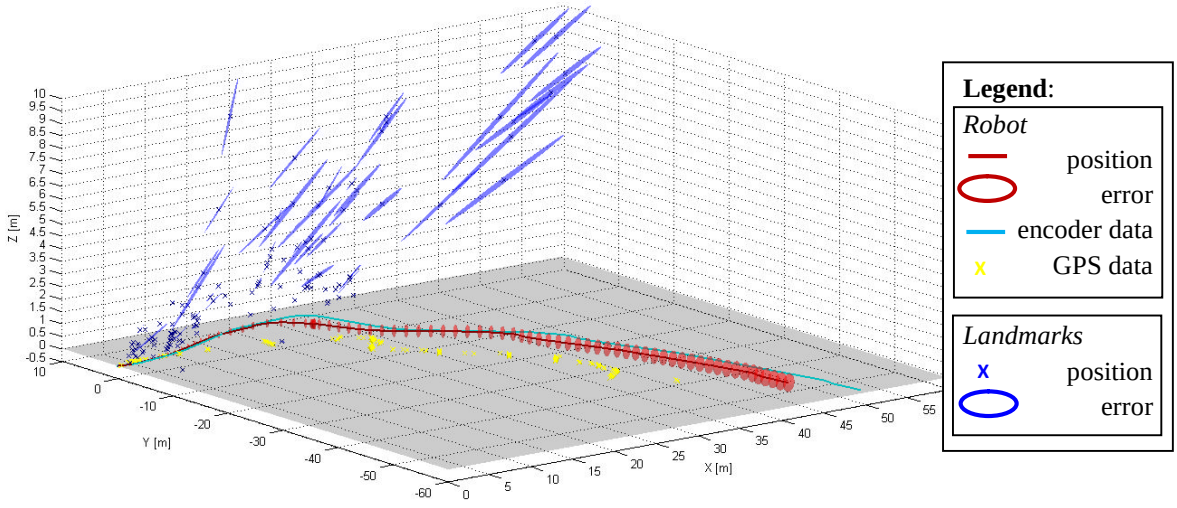

Figure 5.2.3: *3D environmental map for the optical video analysis (example 1).*

Thermal camera

The same analysis is made for the thermal camera. Fig. 5.2.4 shows a typical output video frame for the thermal camera and contains the same scene as the optical camera case.

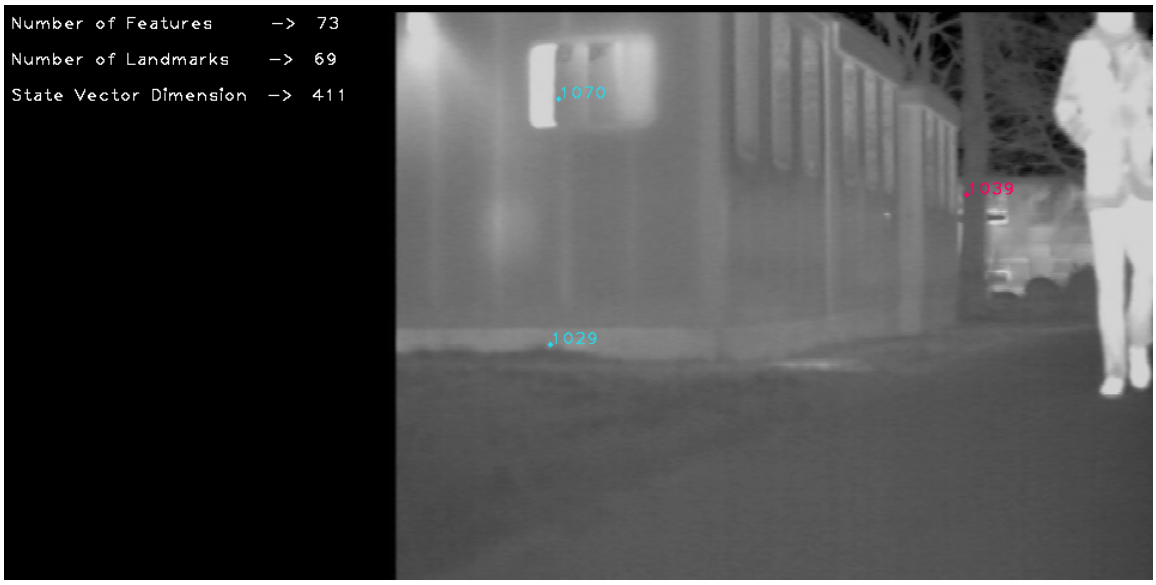

Figure 5.2.4: *Example of the output when analyzing the thermal video (example 1).*

The output of the analysis is saved in CSV format and post processed in Matlab. The resulted 2D environmental map for the thermal case is shown in Fig. 5.2.5.

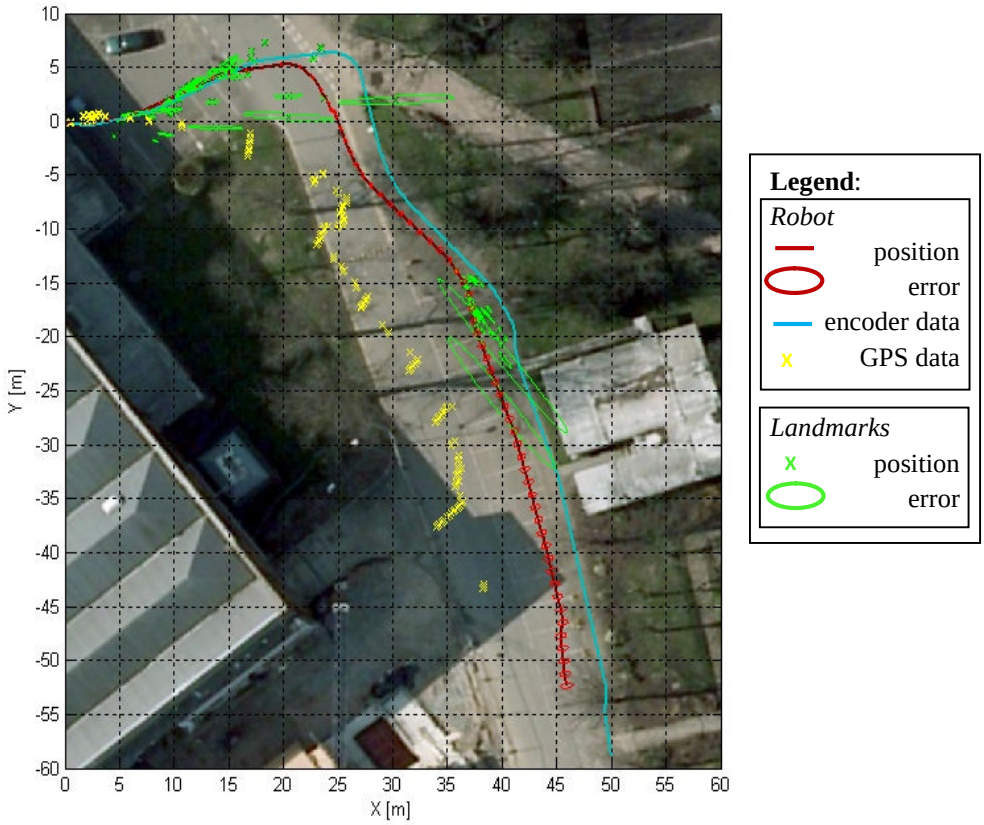

Figure 5.2.5: *2D environmental map for the thermal video analysis (example 1).*

Comparing the 2D map obtained from the thermal video (see Fig. 5.2.5) with the one obtained from the optical analysis (see Fig. 5.2.2) the reader can easily observe the different in terms of landmarks detected. The thermal images do not have as much details as the optical images and also the field of view is narrower (see Fig.5.2.8) and this causes the thermal camera to detect distant feature points that are often not very stable (i.e. they are not being detected for several consecutive frame) during the navigation.

As the minimum ray ρ_{min} used to initialize new thermal feature points is larger than the one used for optical features as the field of view of the thermal camera is narrower of the one of the optical camera, the ellipses of error for the thermal landmarks result are larger than for the optical landmarks error ellipses (see Fig. 5.2.5 and Fig. 5.2.6). For the thermal analysis the landmarks detected are 122 with average errors of 0.94 m, 0.68 m and 0.31 m along the *x*, *y* and *z* axis respectively.

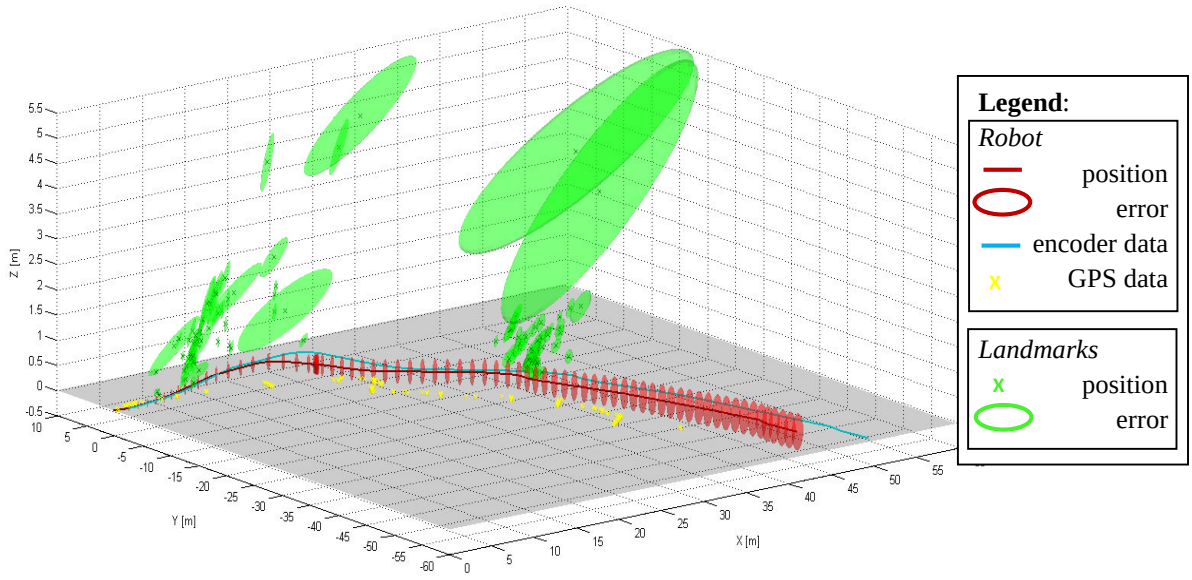

Figure 5.2.6: *3D environmental map for the thermal video analysis (example 1).*

Optical and Thermal cameras

The final stage of the analysis is to study both video together combining the information of the extracted features in a unique *landmark database* with reference to the optical camera reference frame.

As the reader can observe in Fig. 5.2.7 the transformation applied from the thermal image to the optical image is subject to error. The transformation matrix used to join the thermal and optical imagery together is computed as described in Section 3.3.4. Fig. 5.2.7 shows a typical output video frame for this analysis.

Figure 5.2.7: *Example of the output when analyzing the optical and thermal video together (example 1).*

As the single analysis case, the visual output for the combined system shows the number of features (optical and thermal), the number of landmarks, the dimension of the EKF state vector and the current combination of frames obtained as described in Section 3.3. Fig 5.1.8 shows the optical and thermal images correspondent to the video output of Fig. 5.2.7.

Figure 5.2.8: *Optical and Thermal video frame of Figure 5.2.7.*

Finally in Fig. 5.2.9 is shown the output 2D environmental map obtained for the combined system analysing the optical and thermal sensor video inputs at the same time.

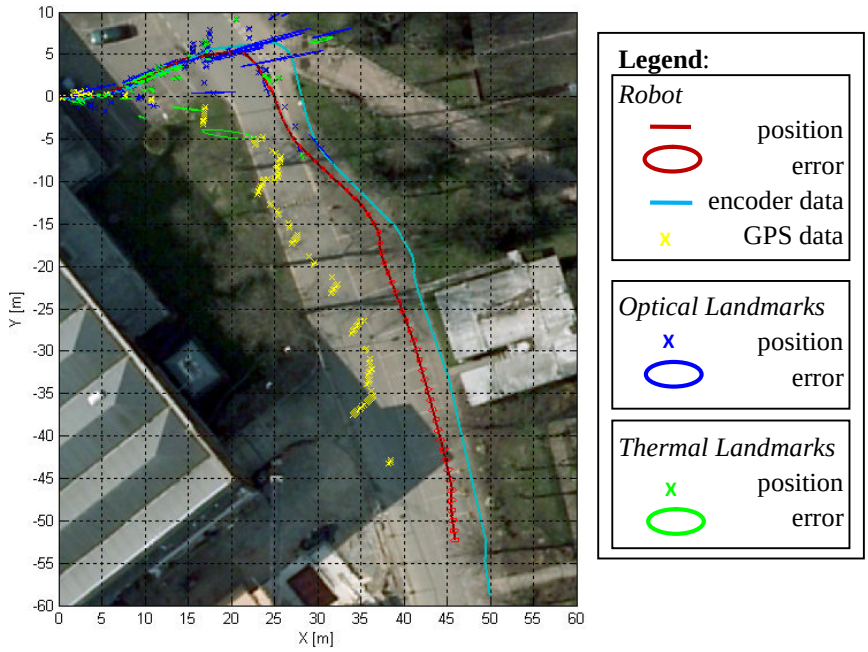

Figure 5.2.9: *2D environmental map for the combined system (example 1).*

Comparing the 2D (or 3D) environmental maps in the thermal and optical single cases with the one obtained from the combined system, we can tell that the number of optical landmarks extracted is similar whilst the number of thermal landmarks is smaller in the combined system (are 122 for the thermal video single case and 67 for the combined system). The reason of that has to be associated with the estimation of the EKF state vector (i.e. the robot position estimation). As the feature initialization and the landmark position update are strictly related to the robot position where the observation of the feature/landmark is taken, a change in the robot position estimation introduces a different behaviour with respect to the feature initialization and landmark estimation processes. In the combined system the optical landmarks detected are 104 and the thermal landmarks are 67 with average errors of 1.06 m, 0.43 m and 0.26 m along the *x*, *y* and *z* axis respectively.

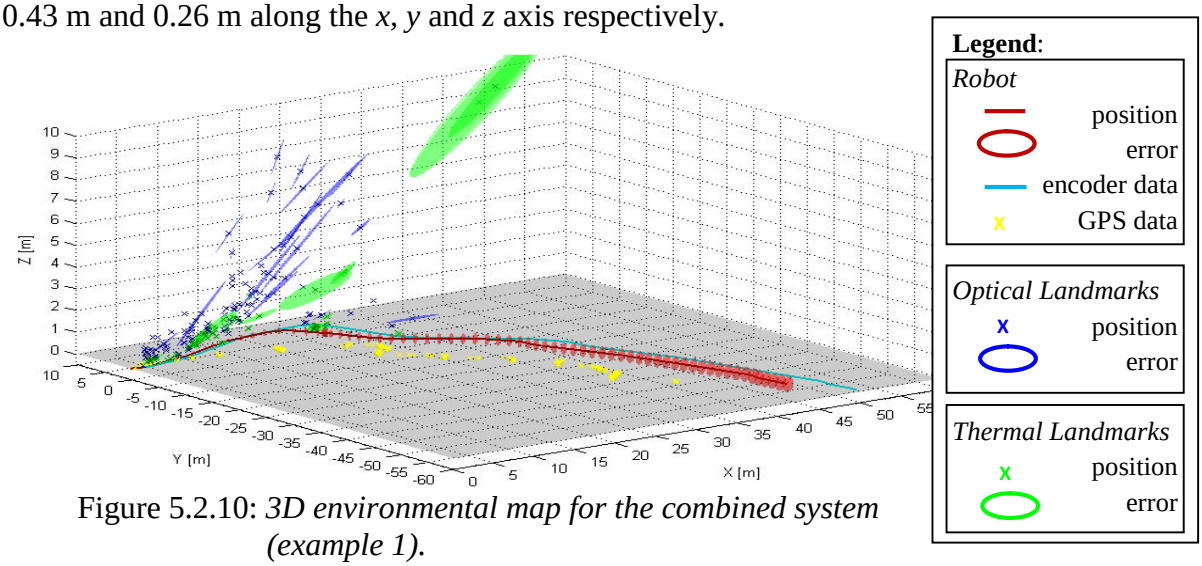

Figure 5.2.10: *3D environmental map for the combined system (example 1).*

Despite the decrease of the number of thermal landmarks detected in the combined system with respect to the thermal only case analysis, the ellipses of error for the thermal landmarks in the combined system are actually smaller than the ones recovered in the thermal only case. In Fig. 5.2.10 the 3D environmental map is represented and the plane $Z = 0$ is added to the plot.

In Tab. 5.2.1 and following are reported some numerical output obtained from the single camera analysis and from the combined system. After 600 integration time steps (Tab. 5.2.1) the values of the EKF state vector are compatible in terms of error intervals for all the three analyzed cases.

k=600	**Combined Optical and Thermal**	**Optical only**	**Thermal only**
X [m]	24.04 ± 0.21	24.07 ± 0.21	24.06 ± 0.21
Y [m]	1.76 ± 0.28	2.47 ± 0.23	1.72 ± 0.27
Z [m]	0.05 ± 0.24	0.04 ± 0.25	0.00 ± 0.25
γ [deg]	1.51 ± 4.91	-7.93 ± 3.90	1.26 ± 4.92
β [deg]	-5.56 ± 4.88	-8.82 ± 3.25	-5.07 ± 4.92
α [deg]	-60.52 ± 2.41	-57.07 ± 2.29	-60.80 ± 2.41

Table 5.2.1: *EKF state vector for the final system, the optical single case and the thermal single case respectively after k = 600 frames (i.e.50 seconds) – (example 1).*

From Tab 5.2.1 we can also observe that the best compatibility is obtained between the combined system and the thermal camera analysis in terms of error intervals. As the landmarks detected and observed give information to update the current position of the robot, in these cases the number of landmarks detected at this stage is almost the same and this permits a similar estimation of the EKF state vector.

k=1200	**Combined Optical and Thermal**	**Optical only**	**Thermal only**
X [m]	34.28 ± 0.46	34.25 ± 0.45	34.28 ± 0.46
Y [m]	-11.90 ± 0.41	-11.41 ± 0.40	-11.96 ± 0.40
Z [m]	0.06 ± 0.34	0.04 ± 0.35	0.02 ± 0.35
γ [deg]	0.94 ± 6.93	-13.28 ±6.19	0.80 ± 6.94
β [deg]	-8.53 ± 6.91	-13.23 ± 5.86	-7.74 ± 6.94
α [deg]	-45.52 ± 3.31	-45.70 ± 3.31	-45.73 ± 3.31

Table 5.2.2: *EKF state vector for the final system, the optical single case and the thermal single case respectively after k = 1200 frames (i.e.100 seconds) – (example 1).*

After 1200 integration time steps (see Tab 5.2.2) the robot position estimations are still compatible in terms of error interval with particular regards to the case of the combined system and the thermal camera analysis. As previously mentioned, also a small difference in the robot position changes the detection of landmarks and the reader can actually see that for each outlined time step (from Tab. 5.2.1 to Tab. 5.2.4) the EKF state vector for the optical camera single case are slightly different from the other two cases also if it remains compatible.

k=1800	**Combined Optical and Thermal**	**Optical only**	**Thermal only**
X [m]	42.74 ± 0.77	42.63 ± 0.77	42.71 ± 0.78
Y [m]	-34.56 ± 0.43	-34.20 ± 0.43	-34.62 ± 0.43
Z [m]	0.05 ± 0.42	0.07 ± 0.42	0.01 ± 0.43
γ [deg]	-1.34 ± 8.48	-17.16 ± 7.88	-1.07 ± 8.49
β [deg]	-11.09 ± 8.47	-16.49 ±7.63	-9.86 ± 8.49
α [deg]	-75.30 ± 3.87	-75.37 ± 3.87	-75.55 ± 3.87

Table 5.2.3: *EKF state vector for the final system, the optical single case and the thermal single case respectively after k = 1800 frames (i.e.150 seconds) – (example 1).*

In Tab. 5.2.4 is shown the final result obtained in this first example. The output is again compatible in terms of interval error among the cases.

k=2248	**Combined Optical and Thermal**	**Optical only**	**Thermal only**
X [m]	45.96 ± 0.96	45.90 ± 0.96	45.92 ± 0.96
Y [m]	-52.28 ± 0.43	-51.95 ± 0.43	-52.34 ± 0.43
Z [m]	0.05 ± 0.47	0.07 ±0.47	0.01 ± 0.48
γ [deg]	-3.65 ± 9.48	-20.18 ± 8.94	-2.91 ± 9.49
β [deg]	-12.82 ± 9.46	-18.59 ± 8.73	-11.14 ± 9.49
α [deg]	-81.49 ± 4.20	-81.41 ±4.20	-81.76 ± 4.20

Table 5.2.4: *Final value of the EKF state vector for the final system, the optical single case and the thermal single case respectively (k = 2248 frames, i.e. about 187 seconds) – (example 1).*

In this example we cannot use one of the cases as a ground truth reference as the robot travels for a long distance and in all the case landmarks are detected. However, if we consider

that the mobile robot is moving in a surface that can be considered plane, the results obtained for the combined system and the thermal single analysis seem to better estimate the robot position with particular regards to the robot orientation along the *x* and *y* axes where the values should be around zero.

In this first outdoor example we analyzed a non automatic navigation of the mobile robot and the obtained results are compatible with what we expected in terms of robot and landmarks position estimation. A final observation can be made looking at the numerical results reported in tables from 5.2.1 to 5.2.4. In this example, the single case for the optical camera detects a larger number of landmarks than in the thermal case but the information about the orientation of the robot seems to diverge with particular regards to the angle around the *x* axis. We can also observe that combining the information of the optical and thermal cameras the robot position estimation in the combined system seems to be corrected by the thermal landmarks observations as the only difference between the optical single case and the combined system is the addition of thermal landmarks (the reference frame used during the analysis is exactly the same).

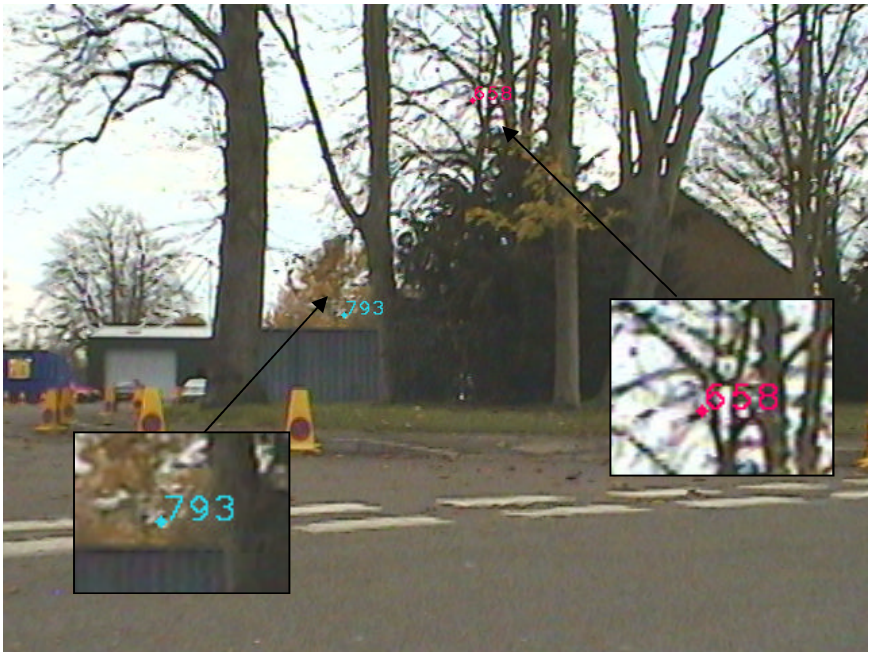

Figure 5.2.11: *Example of an inadequate feature detection and landmark initialization.*

The final aspect to highlight in this example is that in this video sequence a significant number of features are detected in the trees (see Fig. 5.2.11) and sometimes they are selected as landmark, but they are not generally unique and can easily cause wrong a feature matching. As consequence this type of feature is inadequate for the navigation and we could

observe that sometimes they are initialized with a wrong position and this introduces successively wrong observations used to update the EKF.

In the combined system, despite the problem with the presence of erroneous matched landmarks, the thermal camera information increases the performance of the system improving the estimation of the robot position and decreasing the average errors related to the landmark positions.

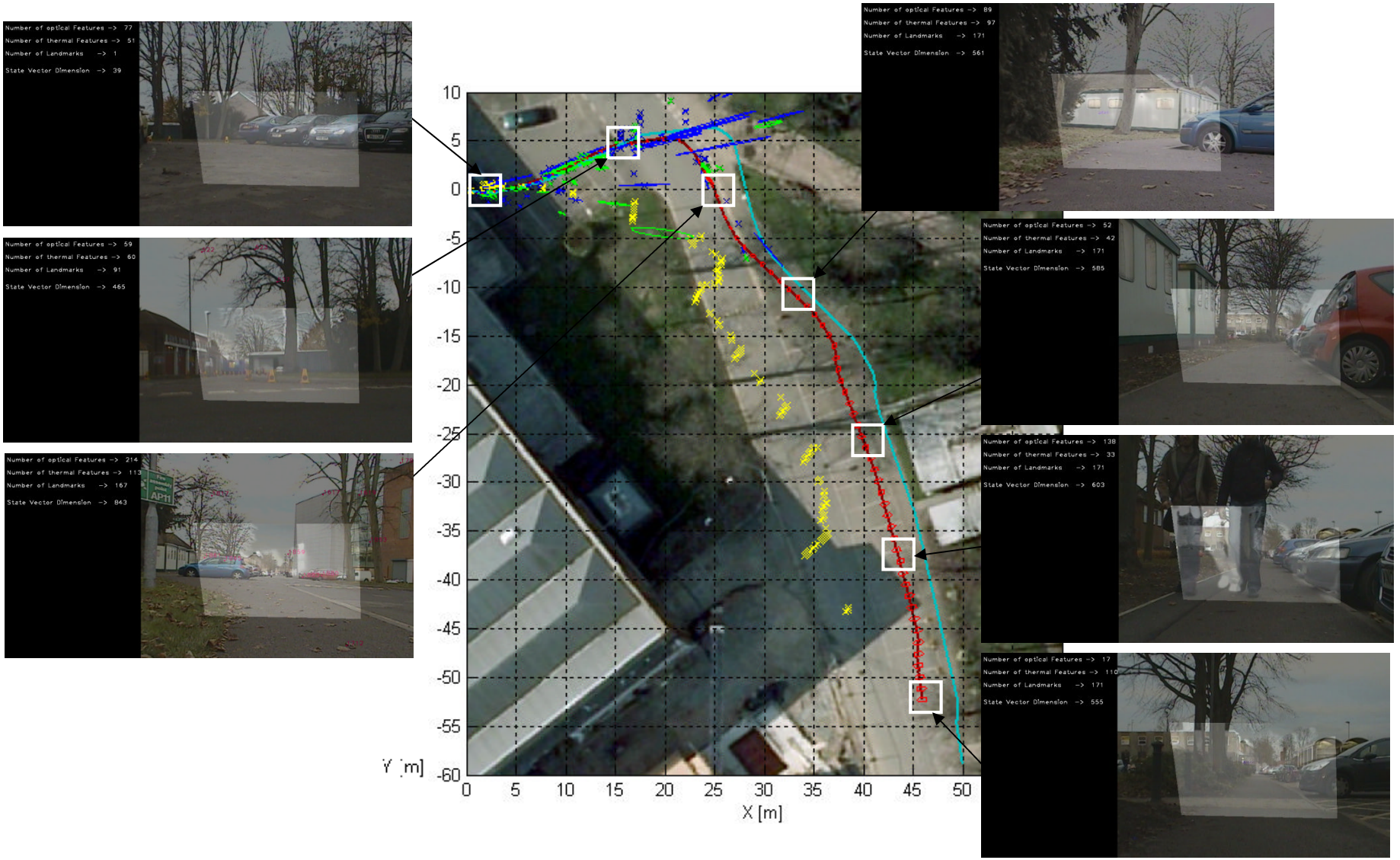

Figure 5.2.12: *2D environmental map with captured frames along the track (example 1).*

Example 2

The second outdoor example refers to a part of the Cranfield University campus where there is less presence of trees so this should assist in detecting more useful landmarks to use for the SLAM purpose. This example refers to an automatic drive of the robot. The translation velocity of the robot is set to 0.2 m/s with a null rotation velocity (expect for a small part of the route). In this analysis the information from the encoders is purely used to update the EKF state vector. As the analyzed video is part of a larger video sequence, the first robot position considered is $\{14.64\,,-0.65\,,0.00\}$ metres with a correspondent orientation of $\{0.00\,,0.00\,,-5.80\}$ degrees. The analysis is done *offline* using the implemented techniques described in the previous chapters.

Optical camera

Fig. 5.2.13 shows a typical output video frame for the optical camera for this second example. The video output shows the current frame (with features and landmarks matched), the number of features stored in the *features database*, the number of landmarks in the *landmark database* and the dimension of the EKF state vector.

Figure 5.2.13: *Example of the output when analyzing the optical video (example 2).*

The output of the analysis CSV format is again used for the post process in Matlab (plotting the 2D and 3D map with robot and landmarks positions). Fig. 5.2.14 shows the 2D environmental map for this case.

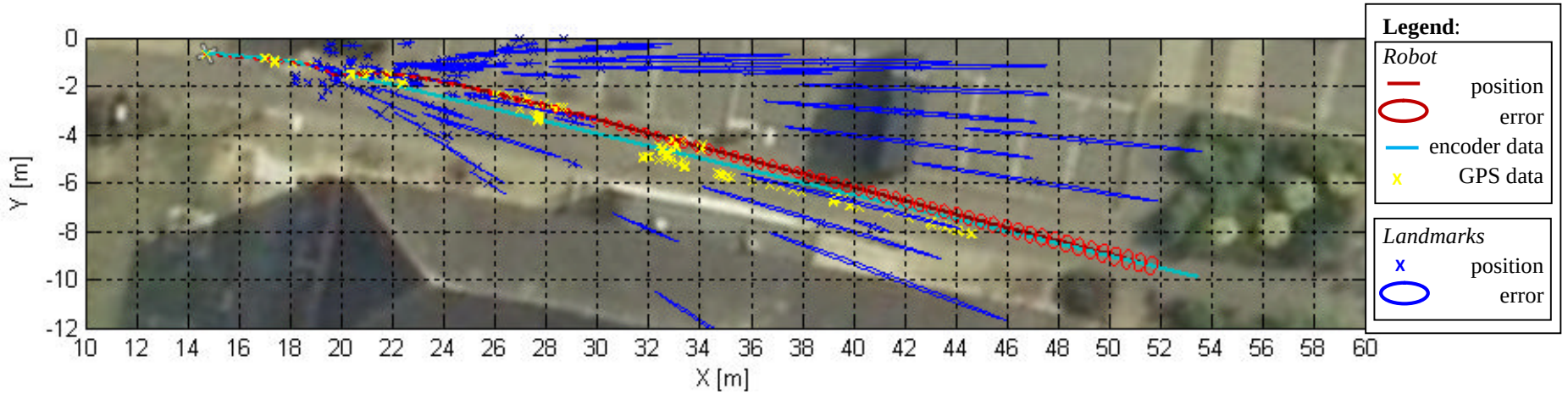

Figure 5.2.14: *2D environmental map for the optical video analysis (example 2).*

In the 2D environmental map the real map is added trying to match between the scales of the 2D environmental map and the map from extracted from [44]. From Fig. 5.2.14, the position of the robot recovered from the GPS data is better estimated respect to the *example 1*. The GPS data are very sensitive to different factors and respect to the case before in this part of the campus there are less trees and this probably is the main difference that provides a better signal for the GPS receiver. Despite the better estimated robot position from the GPS receiver, it does not have a relevant influence during the update of the EKF as the correspondent error is very large in magnitude (see Section 2.5.2). The landmarks detected in the optical single case for this second example are 135 with average errors of 2.10 m, 0.40 m and 0.21 m along the *x*, *y* and *z* axis respectively.

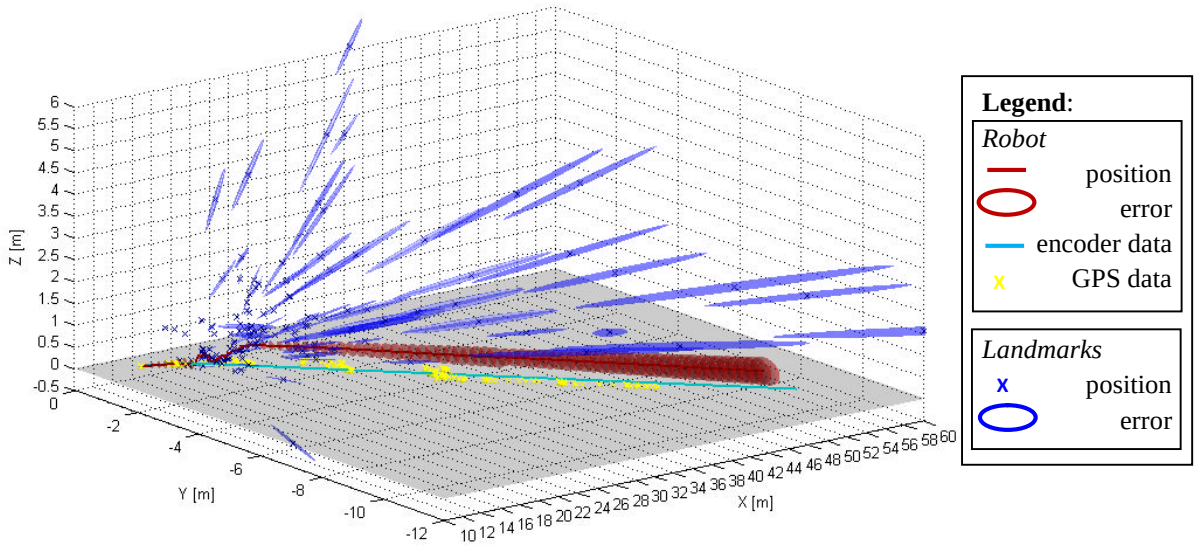

Figure 5.2.15: *3D environmental map for the optical video analysis (example 2).*

The 3D map is shown in Fig. 5.2.15 and we can see a wide range of landmark positioning errors (i.e. ellipses of error in Fig. 5.2.14 and 5.2.15). From the 2D and the 3D maps we can observe several different magnitudes of the landmark ellipses of error and the reason are related to the initial error is based on the distance of the feature/landmark from the camera and to the amount of available observations for the same landmark are available.

Thermal camera

Same analysis is done for thermal camera. Fig. 5.2.16 shows a typical output video frame for the thermal camera and contains the same type of information shown in the optical camera case.

Figure 5.2.16: *Example of the output when analyzing the thermal video (example 2).*

The output of the analysis is saved in CSV format and post processed in Matlab. The resulted 2D environmental map for the thermal case is shown in Fig. 5.2.17.

Comparing the 2D map obtained from the thermal video (see Fig. 5.2.17) with the one obtained from the optical video (see Fig. 5.2.14) the reader can easily observe the difference in terms of landmarks detected as we could expect. The reason of that is again related to the fact that the thermal images do not have as much detail as the optical images and also the field of view is narrow. For this second outdoor example the extracted thermal landmarks appear not very useful for navigation purpose as the positioning error remains quite large. The single thermal analysis recovers for this second analyzed case just 7 thermal landmarks with average errors of 8.31 m, 1.30 m and 0.97 m along the *x*, *y* and *z* axis respectively.

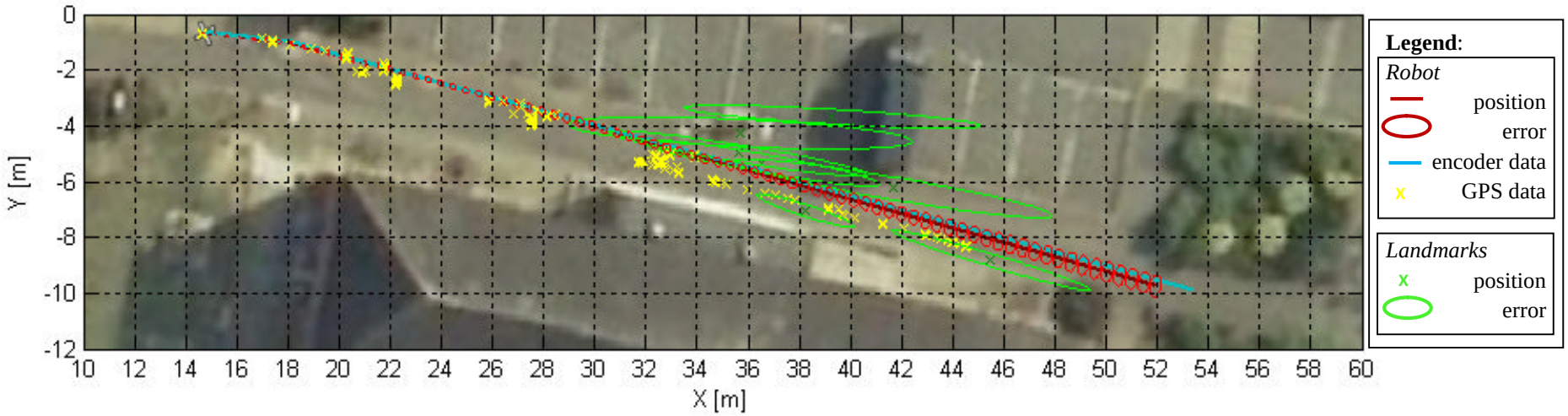

Figure 5.2.17: *2D environmental map for the thermal video analysis (example 2).*

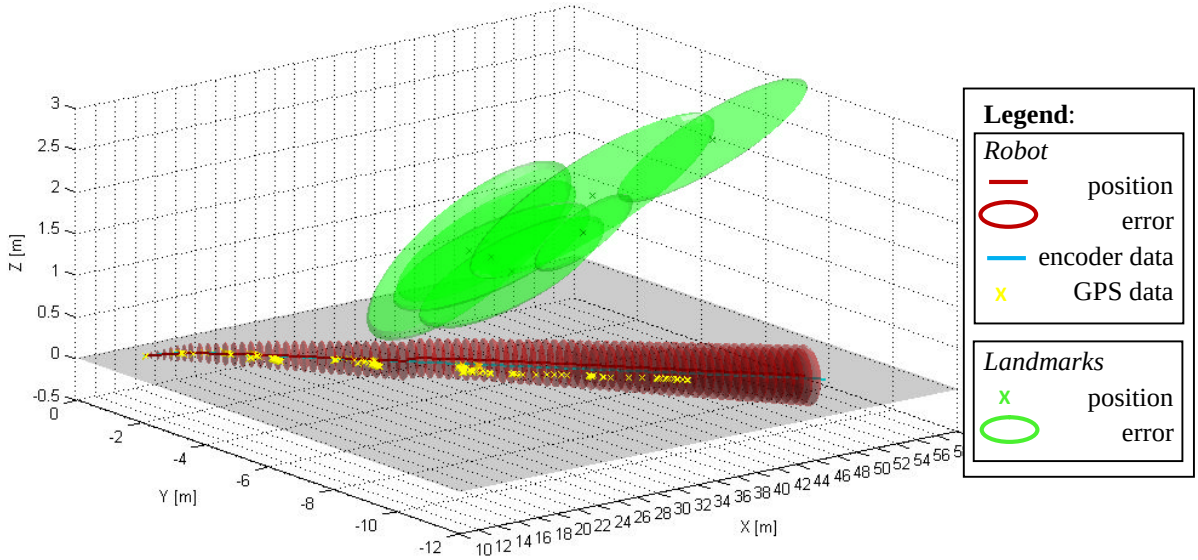

Figure 5.2.18: *3D environmental map for the only thermal video (example 2).*

Optical and Thermal cameras

As final stage of the analysis we study both video together combining the information of the extracted features in a unique *landmark database* using as reference the optical camera reference frame. Fig. 5.2.19 shows a typical output video frame for this analysis.

Figure 5.2.19: *Example of the output when analyzing the optical and thermal video together (example 2).*

As the reader can observe in Fig. 5.2.19, the transformation applied from the thermal image to the optical image (i.e. homography matrix *H*, see Section 3.3.4) is still subjected to error also if the homography matrix *H* is chosen after computing it among several video

frames. Fig 5.2.20 shows the optical and thermal images correspondent to the video output of Fig. 5.2.19.

Figure 5.2.20: *Optical and Thermal video frame of Figure 5.2.19.*

Finally in Fig. 5.2.21 is shown the output 2D environmental map obtained for the combined system analysing the two video at the same time.

In the combined system the amount of optical landmarks is similar to the optical single case but we can observe an increase in the amount of thermal landmarks detected (7 thermal landmarks are detected in the thermal single case, 36 in the combined system). Again some thermal landmarks still have a large error but others (i.e. the ones at the beginning of the robot route) have an error comparable, in magnitude, with the one estimated for the optical landmarks.

From Fig. 5.2.19 we can see how the environment that the mobile robot is exploring is characterized by the presence of cars on the left side and of building on the right side and in front. The combination of the optical and thermal video allows the system to detect 96 optical landmarks and 36 thermal landmarks with average errors of 3.06 m, 0.34 m and 0.31 m along the *x*, *y* and *z* axis respectively.

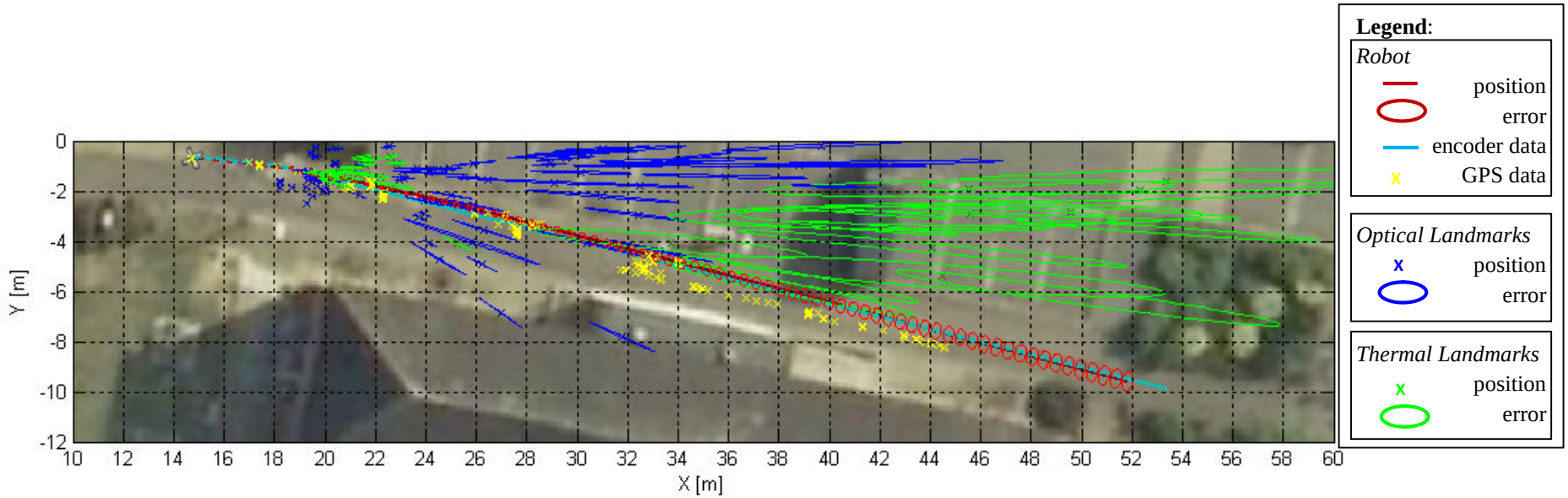

Figure 5.2.21: *2D environmental map for the combined system (example 2).*

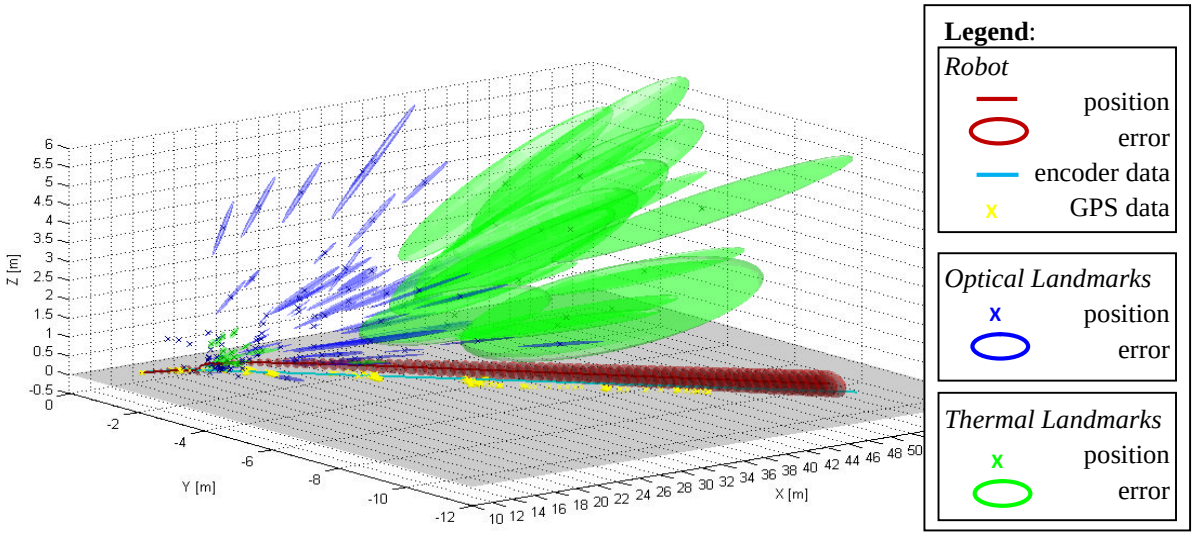

Figure 5.2.22: *3D environmental map for the combined system (example 2).*

During the analysis of this case, we could observe that during the navigation of the robot some people entering in the field of view of the camera but this did not affect the result of the analysis. This observation permits to outline an important aspect: also if the software is developed to cope with a static environment is also able to work with the presence of dynamic objects in the scene as shown in Fig. 5.2.23. This property is related to a threshold added before use the observation vector in the update stage of the EKF (see Section 2.4).

Figure 5.2.23: *Example of people within the field of view of the camera (example 2).*

Before the update stage, the observation of a certain landmark is pruned if the difference between the observed direction and the one computed using the information of the landmark from the map is above a given threshold. The application of this extra threshold has being empirically validated during several analyses adding the property to the system to be able to cope with dynamic object in the field of view of the cameras.

k=600	**Combined Optical and Thermal**	**Optical only**	**Thermal only**
X [m]	24.21 ± 0.18	23.81 ± 0.16	24.47 ± 0.19
Y [m]	-2.30 ± 0.16	-1.81 ± 0.11	-2.59 ± 0.19
Z [m]	0.22 ± 0.17	0.34 ± 0.11	0.01 ± 0.24
γ [deg]	0.31 ± 3.49	-2.02 ± 2.66	-0.64 ± 4.64
β [deg]	-1.40 ± 2.24	-1.81 ± 1.92	-0.90 ± 2.75
α [deg]	-12.16 ± 2.02	-10.29 ± 1.96	-13.88 ± 2.21

Table 5.2.5: *EKF state vector for the final system, the optical single case and the thermal single case respectively after k = 600 frames (i.e.50 seconds) – (example 2).*

In Tab. 5.2.5 and 5.2.6 are reported the numerical output obtained from the single camera analysis and from the combined system after 600 integration time steps and 2310 integration time steps (i.e. final integration time step). From Tab 5.2.5 we can observe how the estimated values for the robot position and orientation in the three cases are compatible in terms of error interval with particular regards to the combined system and the thermal single case. As for the first outdoor example, we cannot compare the result with a ground truth reference but we can assume that the robot orientation along the *x* and *y* axes and the *Z* robot position should be around the zero value (as the surface is plane for hypothesis). Also for this second example it seems that adding the information from the thermal camera permits to have a more correct estimation of the robot position and orientation (e.g. the *z* coordinate and the γ angle around the *x* axis).

In Tab. 5.2.6 is shown the final estimated position of the mobile robot and the values are still compatible in terms of interval error among cases. Observing the value of the Z robot position for the three cases the better estimation (based on the hypothesis to have a plane surface) is from the thermal single analysis but we can also observe a best estimation in the combined system compared to the optical single case. This permits us to say that the thermal camera introduced useful information during the robot position estimation done by the EKF

permitting the combined system to obtain a better estimation of the robot position compare to the optical single analysis.

The reader can observe how the errors for the γ and β angles are larger than the error in α. The explanation is related to the observation vector used to update the EKF state vector which contains the encoder data, GPS information and direction of the observed landmarks. Encoder and GPS data are strictly related to the position of the robot (i.e. X and Y coordinates) and to the α angle whilst the landmarks observations are related to all the components of the robot position and orientation. The encoder information is used at each single integration step and the GPS one time per second, whilst the landmarks observations do not always contributed to the update stage of the EKF (as we do not observe a landmark for each single frame) and this allows the system to contain the errors for the X, Y and α components as we can observe comparing the values in Tab. 5.2.5 and 5.2.6.

k=2310	**Combined Optical and Thermal**	**Optical only**	**Thermal only**
X [m]	51.82 ± 0.39	51.61 ± 0.39	51.91 ± 0.40
Y [m]	-9.59 ± 0.66	-9.41 ± 0.65	-9.73 ± 0.65
Z [m]	0.22 ± 0.45	0.37 ± 0.40	0.06 ± 0.47
γ [deg]	-0.55 ± 8.96	-3.85 ± 8.04	-1.60 ± 9.06
β [deg]	-2.73 ± 8.56	-4.25 ±7.81	-5.32 ±7.84
α [deg]	-14.42 ± 4.59	-14.67 ± 4.59	-14.40 ± 4.59

Table 5.2.6: *Final value of the EKF state vector for the final system, the optical single case and the thermal single case respectively (k = 2310 frames, i.e. 192 seconds) – (example 2).*

In this example we could also observe how the combined system detects more thermal landmarks compared to the thermal single case and this is another positive aspect to highlight: also if the two cameras work initially in an independent way (i.e. during the feature initialization and updating process), merging the landmark information together increases the performance of the system also in terms of the amount of the thermal landmarks detected.

The overall system increased the amount of thermal landmarks detected and improved the robot position estimation for all of the integration time steps, provided the thermal information does not decrease the average error in landmark position.

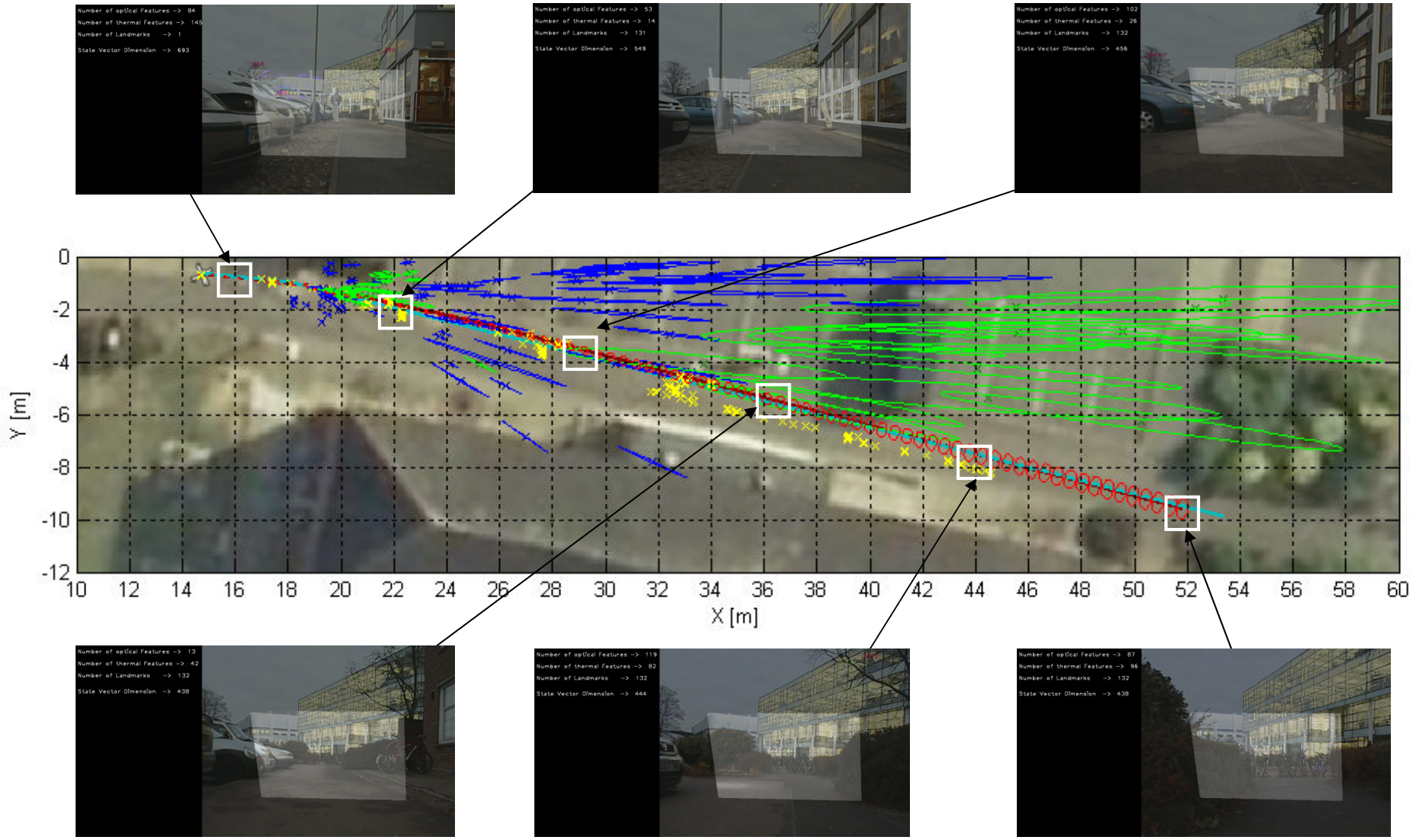

Figure 5.1.24: *2D environmental map with captured frames along the track (example 2).*

Example 3

This example refers to an automatic drive of the robot in which the translation velocity of the robot is set to 0.2 m/s and the rotation velocity remains null during the navigation. Also for this case, the information from the encoders is purely use to update the EKF state vector.

Optical camera

Fig. 5.2.25 shows a typical output video frame for the optical camera for this second example.

Figure 5.2.25: *Example of the output when analyzing the optical video (example 3).*

The result 2D and 3D environmental maps are shown in Fig. 5.2.26 and Fig 5.2.27 in which also the robot position is shown with the corresponding positioning error.

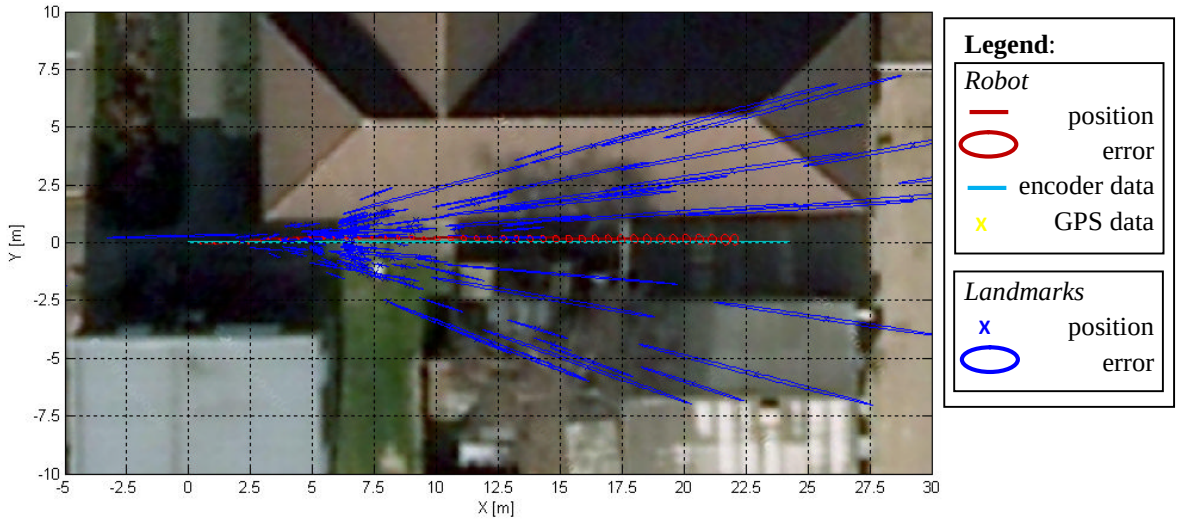

Figure 5.2.26: *2D environmental map for the optical video analysis (example 3).*

From Fig 5.2.25 we can see the type of environment where the surroundings are closer to the mobile robot compared to the first two outdoor examples. This choice was made to analyze different behaviour of the system related to the distance of the environmental features respect to the mobile robot. Based on the chosen environment we have to outline the fact that the available GPS data are actually unusable in this example. The reason is probably related to the presence of buildings and trees closed to the mobile robot as we can see from Fig. 5.2.25. This does not mean that we do not use the GPS data but they all refer to the first position of the robot during the exploration. Again, based on the large GPS error this seems not to have a large impact in the final result.

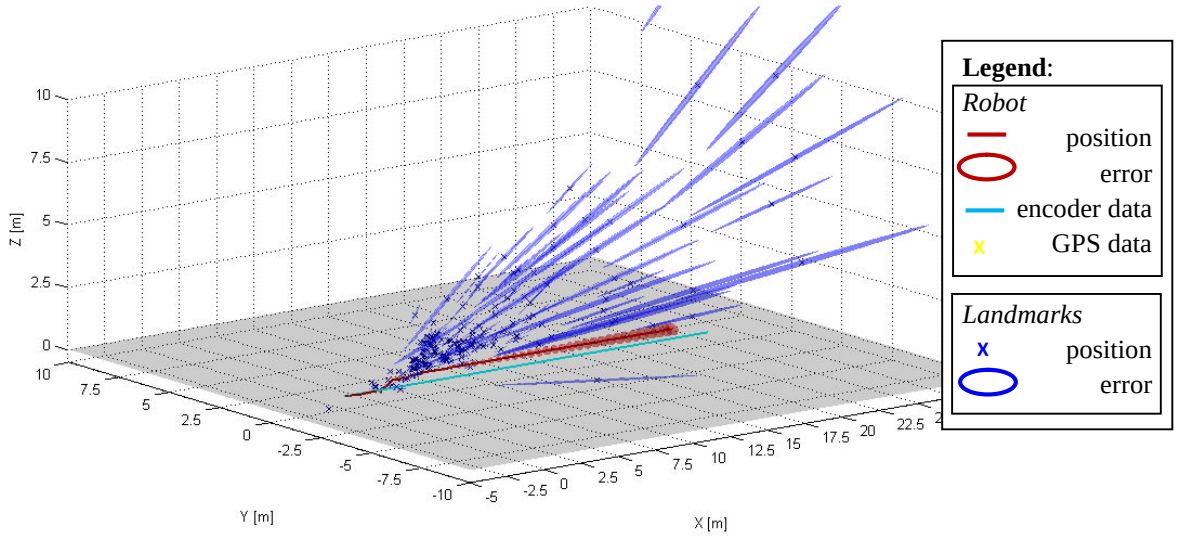

Figure 5.2.27: *3D environmental map for the optical video analysis (example 3).*

From the 2D and the 3D environmental maps (see Fig. 5.2.26 and Fig. 5.2.27) we can observe that a large amount of optical landmarks are detected an added to the environmental maps. The optical landmarks detected in this example are indeed 187 with average errors of 1.66 m, 0.34 m and 0.43 m along the *x*, *y* and *z* axis respectively. As for the other analyzed cases, there is a wide range of ellipses of error of the estimated landmarks positions. This is still related to the first position of a landmark: larger is the initialized depth of the feature/landmark and larger is going to be the errors related to its initial position.

Thermal camera

The same analysis is made for thermal camera but in this case no thermal landmarks are detected. Running the software without imposing the condition of *Zmax* of 5 m it allows the code to detect few landmarks but, as mentioned at the beginning of this chapter, there is not a valid reason to remove this condition.

Optical and Thermal cameras

As final stage of the analysis we study the optical and thermal videos together merging the information of the extracted features in the environmental map. Fig. 5.2.28 shows a typical output video frame for this analysis where the optical and thermal frames are merged together in the same image.

Figure 5.2.28: *Example of the output when analyzing the optical and thermal video together (example 3).*

Fig 5.2.29 shows the optical and thermal images corresponding to the video output of Fig. 5.2.28.

Figure 5.2.29: *Optical and Thermal video frame of Figure 5.2.28.*

The 2D environmental map for this outdoor example is shown in Fig. 5.2.30 and the corresponding 3D map is shown in Fig. 5.2.31.

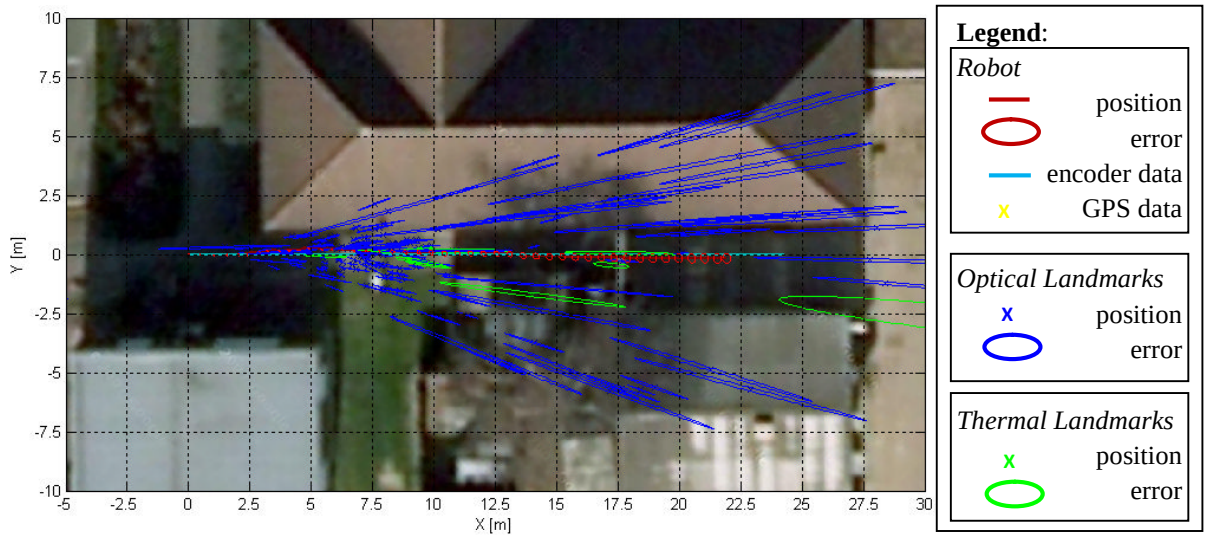

Figure 5.2.30: *2D environmental map for the combined system (example 3).*

As in the previous examples, in the combined system the performance in terms of landmarks extraction increases, but in this case we pass from a total absence of thermal landmarks detected (in the thermal single analysis) to an extraction of thermal landmarks when analysing the optical and thermal videos together. The reason of this behaviour is again related to the estimation of the robot position and orientation done by the EKF algorithm. A small difference in the estimation can allows the combined system to detect landmarks where the single case could not. Indeed, the thermal landmarks detected for the combined system are 13 and the optical landmarks are 208 with average errors of 1.66 m, 0.29 m and 0.40 m along the *x*, *y* and *z* axis respectively.

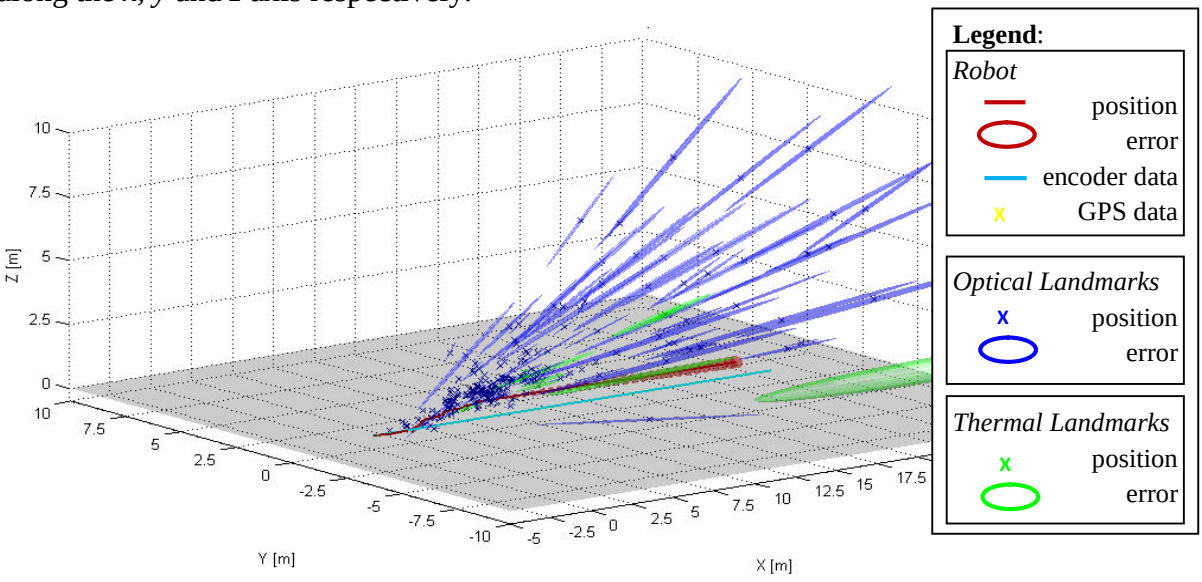

Figure 5.2.31: *3D environmental map for the combined system (example 3).*

In Tab. 5.2.7 and 5.2.8 are reported the numerical output obtained from the single camera analysis and from the combined system after 600 integration time steps and 1312 (i.e. final

integration time step). In this example, as for the indoor example of Section 5.1, we can use the thermal only case as a ground truth reference as no thermal landmarks are detected (i.e. just the encoder data are used to update the robot position by the EKF). In this example we also refer to an automatic navigation and, as the mobile robot does not travel for a long period of time, the encoder drift can be ignored.

k=600	**Combined Optical and Thermal**	**Optical only**	**Thermal only**
X [m]	9.35 ± 0.15	9.51 ± 0.15	10.53 ± 0.19
Y [m]	0.09 ± 0.14	0.18 ± 0.14	0.00 ± 0.22
Z [m]	0.61 ± 0.14	0.33 ±0.16	0.00 ± 0.25
γ [deg]	5.54 ± 3.0	0.67 ± 2.65	0.00 ± 4.92
β [deg]	-10.62 ± 2.67	-9.07 ± 1.90	-0.94 ± 4.92
α [deg]	-1.09 ± 2.34	0.89 ± 1.37	-0.05 ± 2.45

Table 5.2.7: *EKF state vector for the final system, the optical single case and the thermal single case respectively after k = 600 frames (i.e.50 seconds) – (example 3).*

From Tab 5.2.7 we can observe how the values for the robot position are not completely compatible in terms of error intervals among the three analyzed cases. Observing the first results shown in Tab. 5.2.7 we can see how the *Y* robot position of the combined system is better estimated compared to the optical only case value using the thermal single analysis as reference. However, the *Z* robot position value and the γ and β angles are better estimated in the optical single case. These results are persistent also at the end of the robot navigation as we can observe from Tab. 5.2.8.

k=1312	**Combined Optical and Thermal**	**Optical only**	**Thermal only**
X [m]	21.92 ± 0.27	22.04 ± 0.28	22.77 ± 0.29
Y [m]	-0.17 ± 0.36	0.14 ± 0.39	-0.01 ± 0.40
Z [m]	0.62 ± 0.30	0.32 ± 0.31	0.00 ± 0.36
γ [deg]	7.68 ± 5.80	-0.11 ± 5.93	0.01 ± 7.25
β [deg]	-11.70 ± 4.40	-10.54 ± 5.66	-2.17 ±7.25
α [deg]	-0.63 ± 3.50	-0.02 ± 3.57	-0.09 ± 3.57

Table 5.2.8: *Final value of the EKF state vector for the final system, the optical single case and the thermal single case respectively (k = 1312 frames, i.e. 109 seconds) – (example 3).*

The Z position of the robot for the combined system compared to the optical and thermal single cases is not estimated correctly and this is due to a wrong landmarks matching or initialization. This error in the matching/initialization can be caused by the non-uniqueness of the environment such as the bricks wall on the left of the view and bushes along almost all the mobile robot route (see Fig. 5.2.27 and Fid. 5.2.31). In this example we can say that the thermal landmarks do not add useful information to use during the robot position estimation as the thermal imagery does not have much details as the previous example and this seems to cause an error in the matching/initialization as mentioned. Although the magnitude of the *Z* position is not correctly estimated in the combined system with respect to the thermal single analysis, the value stays quite invariant until the end of the process as we can observe in Tab. 5.2.8 in which the final robot position estimations are shown. This result can be related to an initial error and not to an error during the entire robot navigation. If this was the case we should expected and increasing value of the *Z* robot position whilst it stays quite invariant as just outlined.

The final errors associated with the robot position estimation in the combined system result almost the same of the ones estimated from the optical single case, but the thermal camera again adds information during the navigation permitting the combined system to add thermal landmarks and also more optical landmarks into the environmental map with respect to the optical video single analysis.

In this outdoor example the combined system does not show a better robot position estimation as for the previous examples compared to the corresponding optical single case, but we could still observe an increment of the performance of the combined system compared to the optical and thermal single cases in terms of amount of landmarks detected: in the combined system thermal landmarks are detected and added to the 3D environmental map whilst no landmarks are detected in the thermal single case.

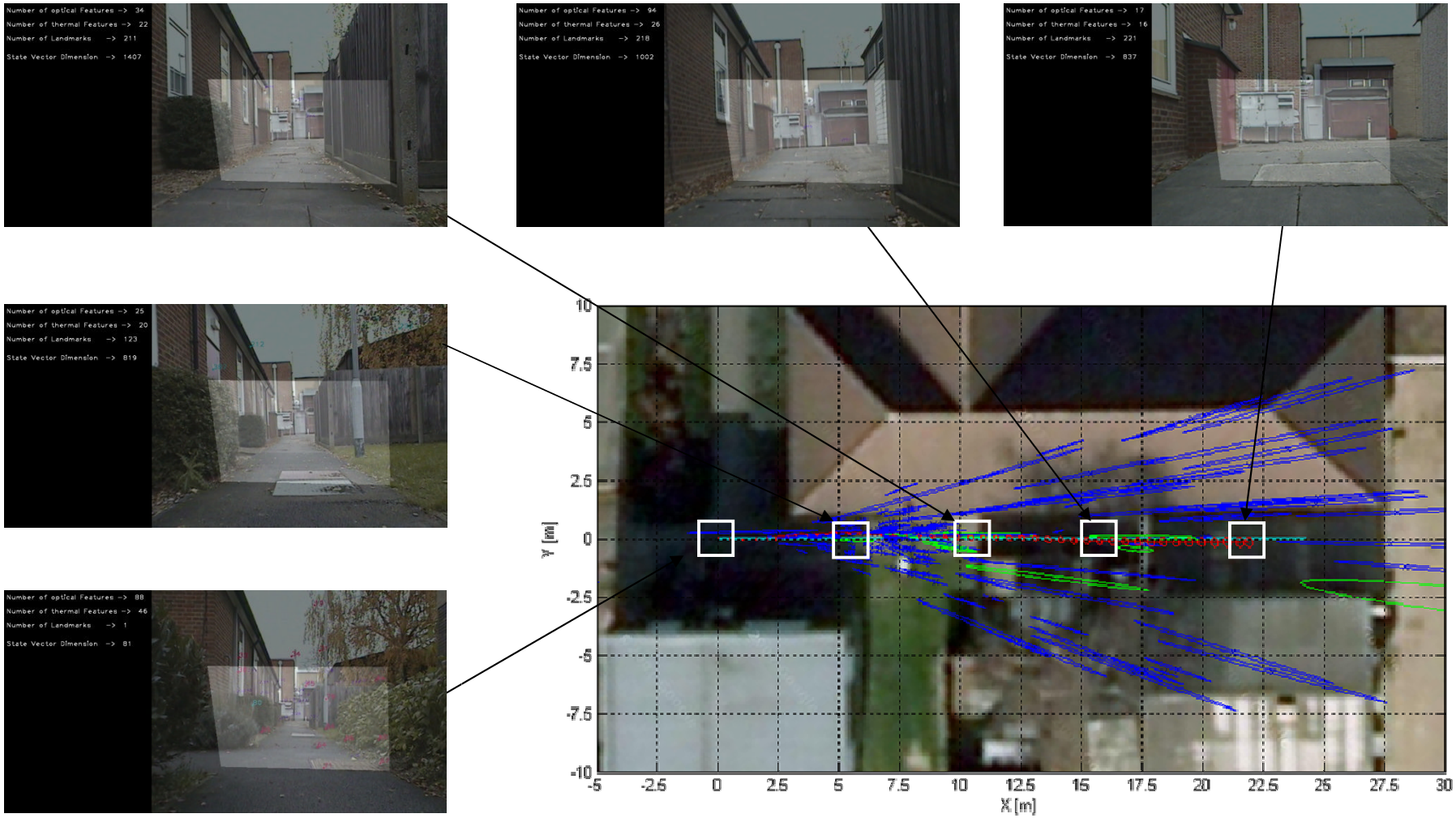

Figure 5.2.32: *2D environmental map with captured frames along the track (example 3).*

Example 4

In this fourth outdoor analyzed case the joystick connected to the mobile robot is used for navigation so is referred to a non-automatic navigation. As the second example, the video used for this case is extracted from a larger video sequence so the first robot position refers to $\{65.21\ ,-38.23\ ,0.00\ \}$ metres and an orientation of $\{0.00\ ,0.00\ ,-31.81\}$ degrees.

Optical and Thermal cameras

An image of the output for this example is shown in Fig. 5.2.33. In this example the thermal camera has a different orientation respect to the previous ones: the orientation allows the camera to capture images on the left side of the mobile robot whilst before was oriented parallel to the optical camera capturing the scene in the centre of the optical camera. Fig 5.2.34 shows the optical and thermal images correspondent to the video output of Fig. 5.2.33.

Figure 5.2.33: *Example of the output when analyzing the optical and thermal video together (example 4).*

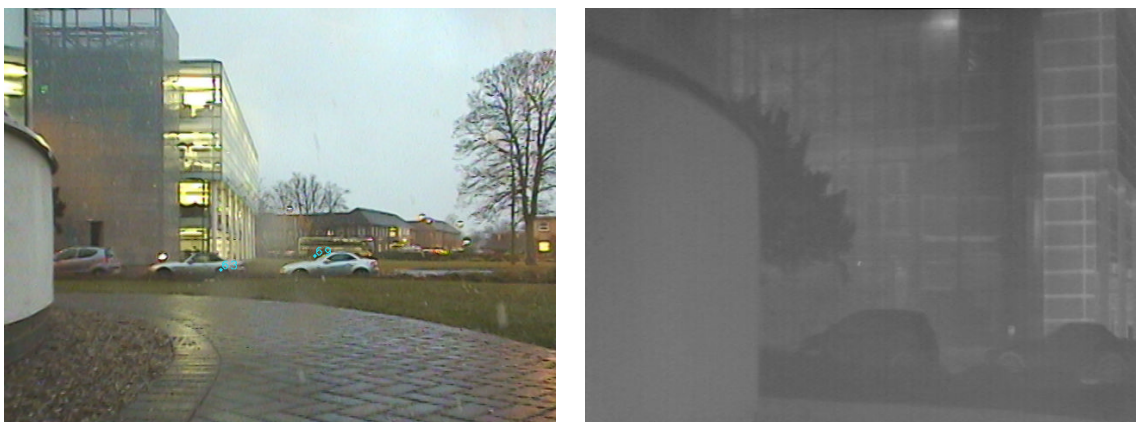

Figure 5.2.34: *Optical and Thermal video frame of Figure 5.2.33.*

Fig. 5.2.35 shows the output 2D environmental map obtained for the combined system analysing the two video at the same time and Fig. 5.2.36 shows the correspondent 3D environmental map.

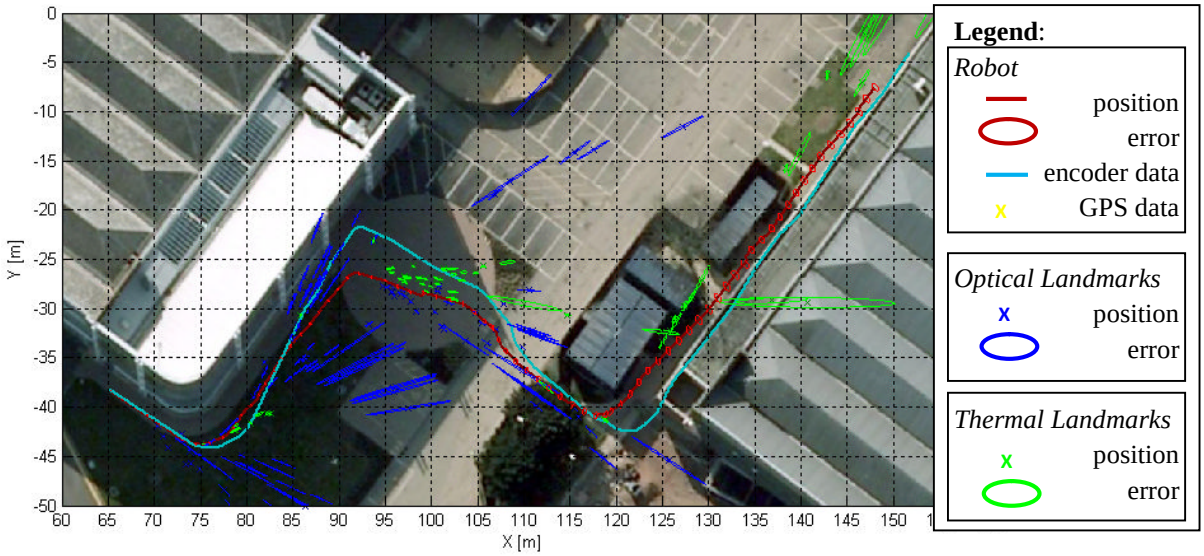

Figure 5.2.35: *2D environmental map for the combined system (example 4).*

In this example we can see a prevalence of optical landmarks at the beginning of the navigation whilst a detection of thermal landmarks seems to dominate the last part of the mobile robot route. This is related to a higher presence of optical features in the first part respect to the end of the route whilst for the thermal images the amount of thermal feature points seems to have an opposite trend. From Fig. 5.2.33 and Fig. 5.2.37 we can see how the environment that the mobile robot is exploring is again characterize by the presence of cars, buildings and also by the presence of people at the end of the route. In this example 141 optical landmarks and 72 thermal landmarks are extracted with an average errors of 1.82 m, 1.49 m and 0.58 m along the *x*, *y* and *z* axis respectively.

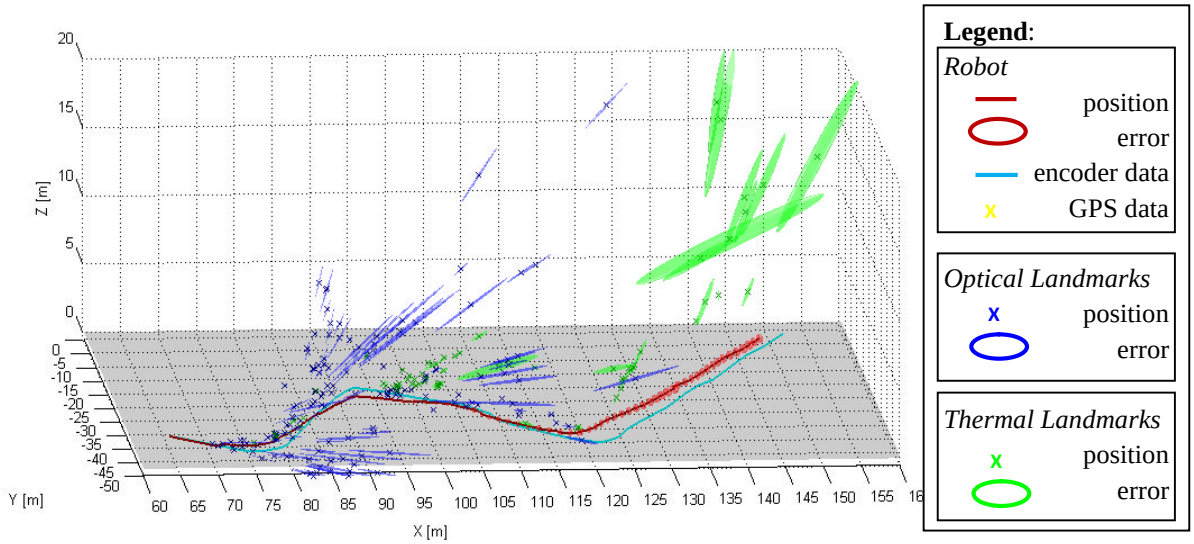

Figure 5.2.36: *3D environmental map for the combined system (example 4).*

In Tab. 5.2.9 are reported the numerical output obtained for the combined system for three different integration time steps: 600, 1400 and 2310. The robot position and orientation errors increase with the time as we expected and this is related to the use of the encoders as sensor to recover the displacement of the mobile robot.

	Combined Optical and Thermal		
	k = 600	*k = 1400*	*k = 2310*
X [m]	87.83 ± 0.16	118.52 ± 0.45	148.00 ± 0.60
Y [m]	-30.38 ± 0.17	-40.81 ± 0.59	-7.52 ± 0.63
Z [m]	0.34 ± 0.16	0.35 ± 0.32	0.35 ± 0.44
γ [deg]	10.67 ± 1.99	11.21 ± 5.77	10.99 ± 4.28
β [deg]	-9.37 ± 1.64	-7.87 ± 5.00	-11.13 ± 6.68
α [deg]	49.89 ± 1.19	1.31 ± 3.48	51.09 ± 2.98

Table 5.2.9: *EKF state vector for the combined system for different integration time steps (example 4).*

For this case, different optical and thermal landmarks are detected during the navigation and a wide range of error ellipses are present in the map. This variety of value for the landmark position errors is again related to the initial coordinates used for a new landmark: further away is a feature/landmark and larger is going to be the initialized correspondent error. We can finally observe how the main thermal landmarks detected are situated in the left side of the robot route and this is as a consequence of the chosen orientation for the thermal camera. As the thermal camera has a smaller and deeper field of view respect the optical camera, having two thermal cameras looking one to each side of the mobile robot could be an idea to increase the amount of information extracted from thermal images for navigation purpose.

In this example just the combined system was tested as it was not possible analyze the thermal video as a single case. The changing on the orientation of the thermal camera cannot be recovered precisely so is not possible to find the necessary transformation that related the centre of mass of the mobile robot (and the correspondent reference frame) to the thermal camera centre and coordinate system. The combined system still benefits from adding the thermal information as several thermal landmarks are added into the environmental map and they result useful for the navigation purpose.

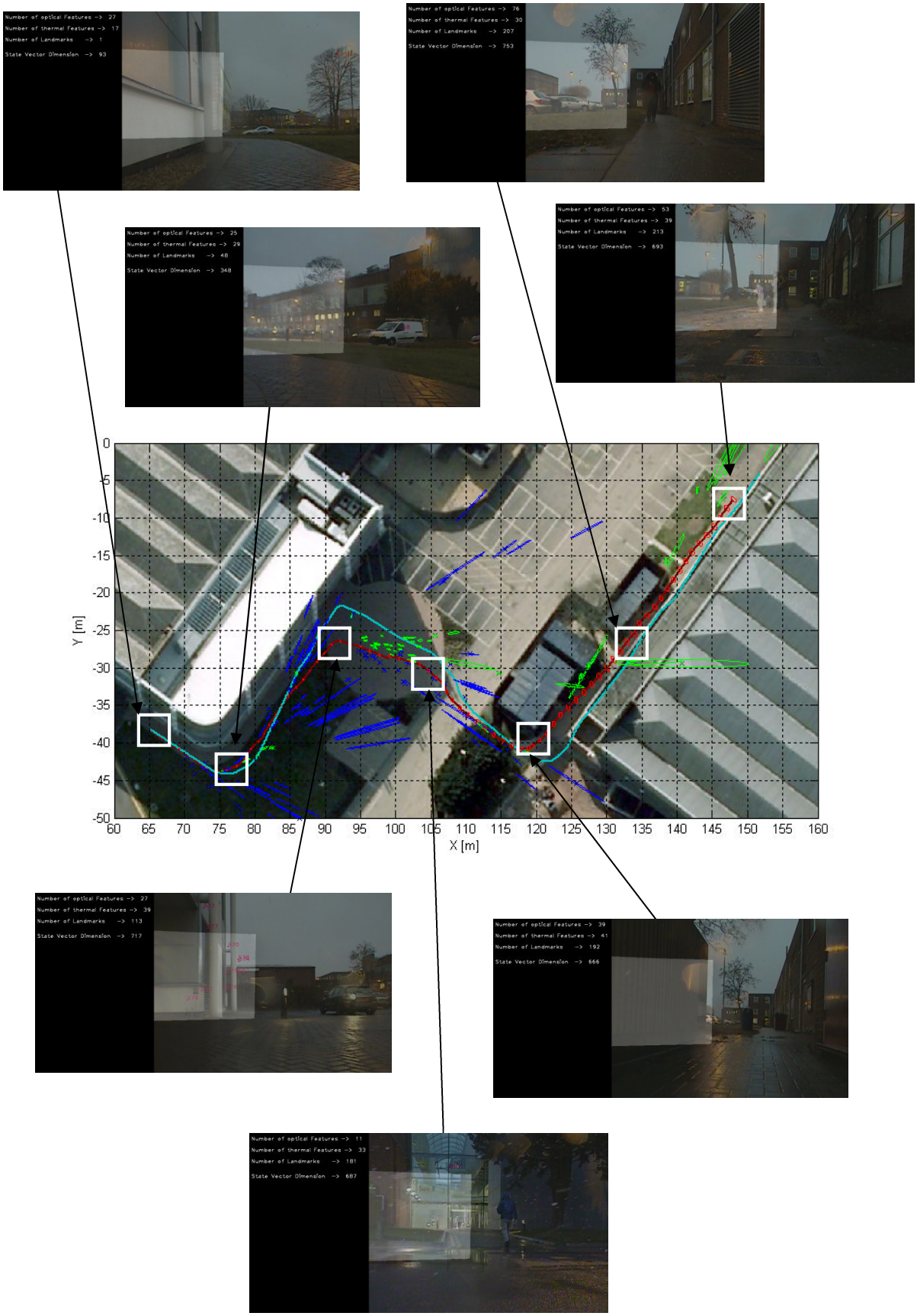

Figure 5.2.37: *2D environmental map with captured frames along the track (example 4).*

Example 5

The last example refers to a car park of the Cranfield University campus where the robot is moved using the connected joystick. As the second and fourth examples, the video used for this case is extracted from a larger video sequence so the first robot position refers to $\{14.45\ ,-139.51\ ,0.00\ \}$ metres and an orientation of $\{0.00\ ,0.00\ ,66.01\}$ degrees.

Optical and Thermal cameras

In Fig. 5.2.38 is shown an example of the output for this analyzed case and as in the previous example the thermal camera is oriented to look at the left side of the mobile robot.

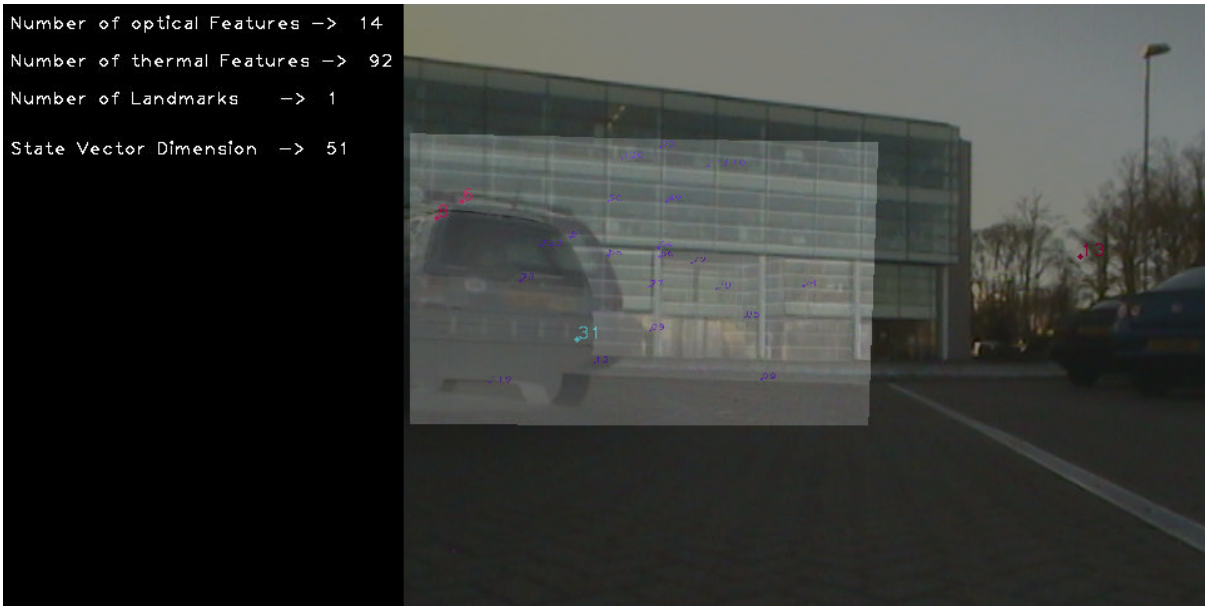

Figure 5.2.38: *Example of the output when analyzing the optical and thermal video together (example 5).*

Fig 5.2.39 shows the optical and thermal images correspondent to the video output of Fig. 5.2.38.

Figure 5.2.39: *Optical and Thermal video frame of Figure 5.2.38.*

Finally in Fig. 5.2.40 is shown the output 2D environmental map obtained for the combined system analysing the optical and the thermal imagery merging the information together.

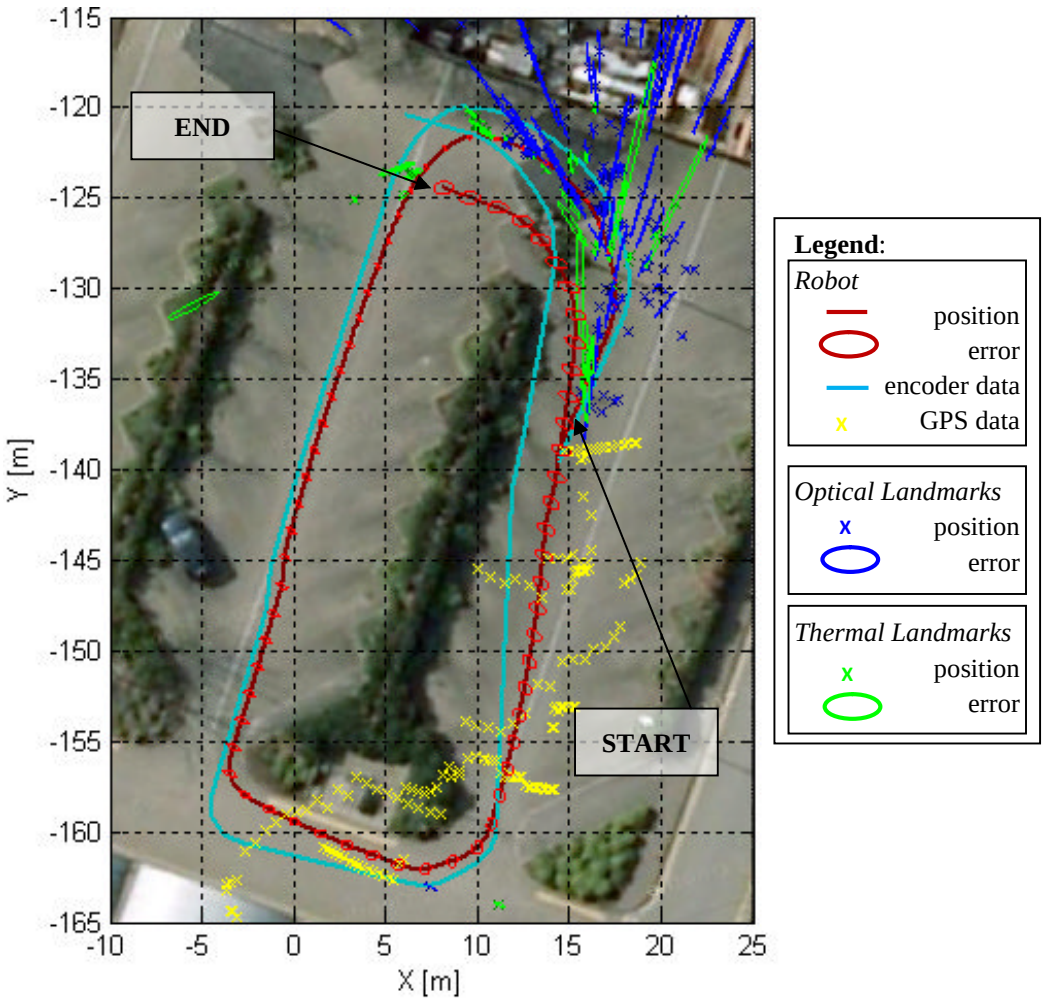

Figure 5.2.40: *2D environmental map for the combined system (example 5).*

As we can see from the 2D environmental map shown in Fig. 5.2.40, the mobile robot positions recovered from the GPS data using the procedure described in Section 2.5.2 are very far from the encoder data, but this seems to do not highly affect the estimation of the robot position by the EKF algorithm. Another observation to be made is related to the amount of optical and thermal landmarks extracted. There is a consistent amount of landmarks extracted during the first part of the robot navigation but then just few landmarks are added to the environmental map during the all navigation. Observing the frames shown in Fig. 5.2.42, we can see that the cars at the beginning of the navigation are close to the mobile robot respect to the second part of the route (i.e. up-left frame in Fig. 5.2.42) and probably the larger distance from the environment does not permit the system to recover landmarks.

The 3D environmental map is show in Fig. 5.2.41 where the landmarks position and relative errors are shown respect to the robot reference frame. In this last analyzed case, the system extract 157 optical landmarks and 11 thermal landmarks are extracted with an average errors of 0.89 m, 1.78 m and 039 m along the *x*, *y* and *z* axis respectively.

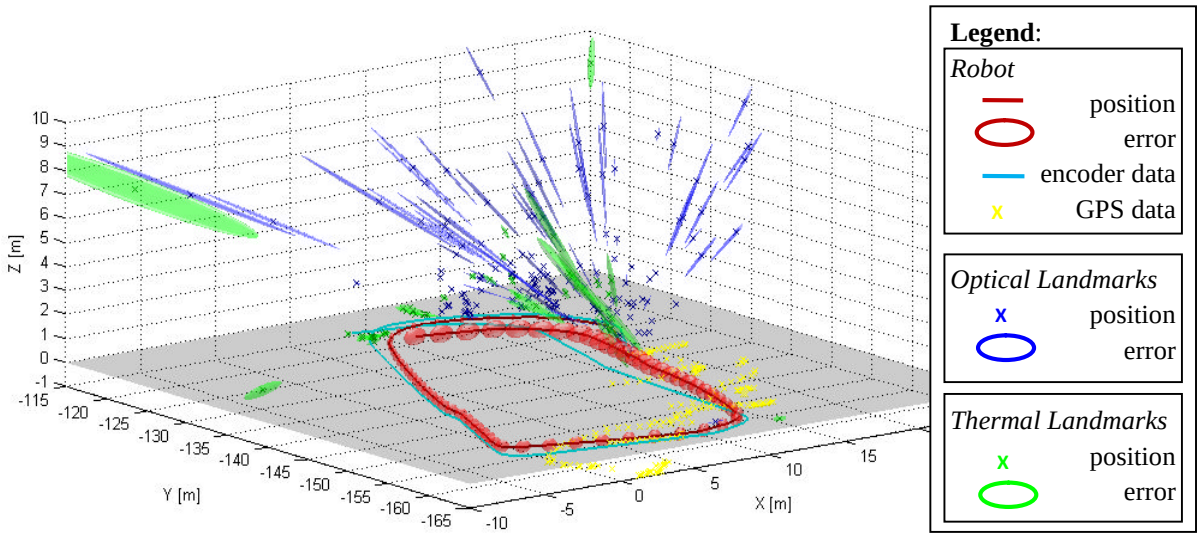

Figure 5.2.41: *3D environmental map for the combined system (example 5).*

In Tab. 5.2.10 are reported the numerical output obtained for three different integration time steps. From the estimated robot positions shown in Tab. 5.2.10 we can observe how again the errors increase with the time.

	Combined Optical and Thermal		
	k = 600	*k = 1400*	*k = 2300*
X [m]	5.15 ± 0.25	5.19 ± 0.59	8.13 ± 0.98
Y [m]	-127.30 ± 0.26	-161.64 ± 0.46	-124.43 ± 0.58
Z [m]	0.01 ± 0.24	0.03 ± 0.37	0.04 ± 0.48
γ [deg]	6.92 ± 4.28	6.66 ± 7.09	8.06 ± 9.31
β [deg]	1.26 ± 4.51	1.99 ± 7.23	2.74 ± 9.41
α [deg]	247.10 ± 2.41	336.33 ± 3.49	153.97 ± 4.15

Table 5.2.10: *EKF state vector for the combined system for different integration time steps (example 5).*

This case is being chosen as a future work proposed is the loop closing technique [31] (see Section 3.1) so this is a useful example that shows how the encoders drift behaves. The robot track close to the end of the navigation route does not exactly follow the initial track of the mobile robot so we could expect different robot potion estimations. When the robot turns on the left (see Fig. 5.2.40) the ending track is almost the same of the start track but from the 2D

environmental map of Fig. 5.2.40 we can see a large difference between the initial and final robot positions and this is caused by the encoder drift.

For this last tested example just the combined system was analyzed as again it was not possible analyze the thermal video as a single case because of the non parallel orientation of the thermal camera with respect to the optical camera. The combined system still benefits from adding the thermal information as several thermal landmarks are added into the environmental map also if the scenario does not have a large presence of features/landmark close to the mobile robot. We can finally say that the thermal camera information results useful for the navigation purpose as adds information into the environmental map and generally increases the overall robot position estimation.

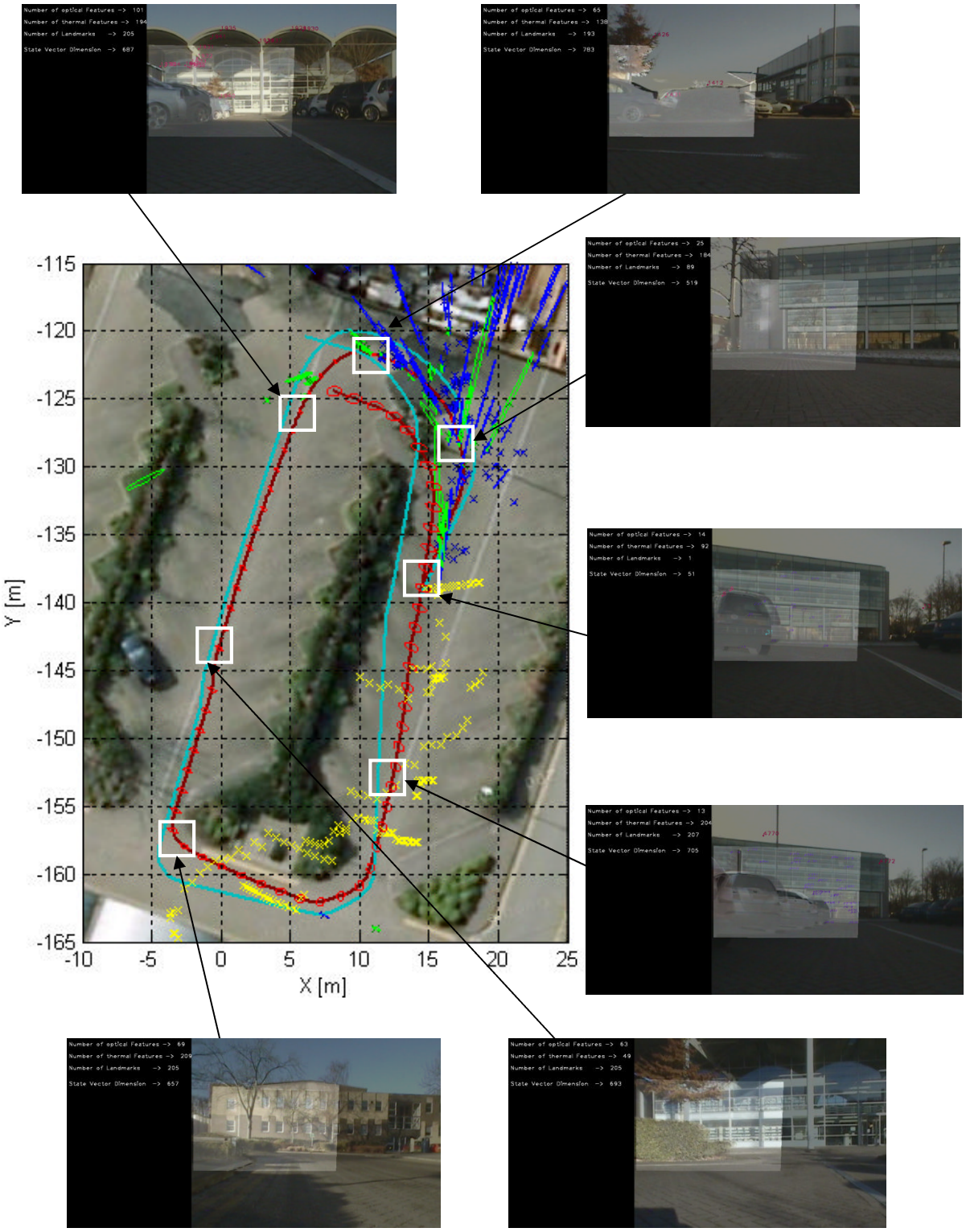

Figure 5.2.42: *2D environmental map with captured frames along the track (example 5).*

5.3 Summary

Observing the results shown for the indoor environment (i.e. Section 5.1) and for the outdoor environment (i.e. Section 5.2) we can say that in each analyzed case the thermal camera information adds useful information within the SLAM problem solving with regards to the robot position, robot estimated error, amount of detected landmarks and landmarks positioning errors.

For the indoor environment the average errors for the estimated robot position (see Tab. 5.3.1) and landmark positions (see Tab. 5.3.2) increase in the combined analysis, but the use of the thermal camera allows the system to detect thermal landmarks whereas in the single case of the thermal camera was not possible and as a consequence permits a better estimation of the mobile robot position within the environmental map.

The best results are obtained during the navigation in an outdoor environment. As we can see from Tab. 5.2.1 the robot positioning errors for the combined system have similar values with respect to the optical case. However, as we observed during the analysis of Section 5.2, in several cases the estimated value for the robot position improve because of the introduction of thermal landmarks within the environmental map.

In Tab. 5.3.2 we can observe the amount of landmarks extracted and the average landmark positioning errors (i.e. arithmetic mean errors) for the indoor and outdoor examples analyzed in Section 5.2. For each example we show the results for the optical (OP) single analysis, the thermal (TH) single analysis and the combined system (OP+TH). Firstly we compare the optical landmark properties (i.e. amount of landmarks extracted and the arithmetic mean positioning errors) for the optical single analysis – where available – with the corresponding properties of the optical landmarks extracted in the combined system. Secondly the results for the thermal landmarks are given. Finally, a summary of the landmark properties for the combined system is provided to permit a comparison with those obtained for the optical single analysis.

From Tab. 5.3.2, focusing on the outdoor examples, we can estimate the overall average increment or decrement of the landmark positioning errors along the *x*, *y* and *z* axes for the combined system with respect to the optical single analysis. Computing the arithmetic mean among the first three outdoor examples – those with results for the combined system and the optical single analysis – we can estimate an increment of 6.1% for the landmark positioning error along the *x* axis, a decrement of 19.3% along the *y* axis and an increment of 0.4% along

the *z* axis. Taking the arithmetic mean of these three percentages gives an overall decrease of the landmark positioning error of 4.3%.

Finally, the other important result to outline is that the amount of landmarks detected during the navigation increases of a 16.2% when using the combined system compare to the optical single case analysis.

	INDOOR	OUTDOOR ENVIRONMENTS		
Optical landmarks		*Example 1*	*Example 2*	*Example 3*
x [m]	0.24	0.96	0.39	0.28
y [m]	0.19	0.43	0.65	0.39
z [m]	0.20	0.47	0.40	0.31
γ [deg]	4.69	8.94	8.04	5.93
β [deg]	3.02	8.73	7.81	5.66
α [deg]	2.96	4.20	4.59	3.57
Thermal landmarks				
x [m]	0.27	0.96	0.40	0.29
y [m]	0.32	0.43	0.65	0.40
z [m]	0.33	0.48	0.47	0.36
γ [deg]	6.69	9.49	9.06	7.25
β [deg]	6.69	9.49	7.84	7.25
α [deg]	3.32	4.20	4.59	3.57
Optical and thermal				
x [m]	0.26	0.96	0.39	0.27
y [m]	0.29	0.43	0.66	0.36
z [m]	0.26	0.47	0.45	0.30
γ [deg]	5.10	9.48	8.96	5.80
β [deg]	4.73	9.46	8.56	4.40
α [deg]	3.31	4.20	4.59	3.50

Table 5.3.1: *Errors for the final robot position in the cases analyzed in Section 5.1 and 5.2.*

	INDOOR ENVIRNMENT			OUTDOOR ENVIRONMENTS										
				Example 1			*Example 2*			*Example 3*			*Example 4*	*Example 5*
System configuration*	OP	TH	OP+TH	OP	TH	OP+TH	OP	TH	OP+TH	OP	TH	OP+TH	OP+TH	OP+TH
Optical landmarks														
Number	192	---	131	129	---	104	135	---	96	187	---	208	141	157
X average error [m]	0.93	---	1.22	1.46	---	0.99	2.10	---	2.11	1.66	---	1.50	2.00	0.94
Y average error [m]	0.13	---	0.18	0.60	---	0.39	0.40	---	0.27	0.34	---	0.28	1.47	1.93
Z average error [m]	0.22	---	0.30	0.43	---	0.28	0.21	---	0.18	0.43	---	0.39	0.51	0.40
Thermal landmarks														
Number	---	0	8	---	122	67	---	7	36	---	0	13	72	11
X average error [m]	---	---	2.52	---	0.94	1.16	---	8.31	5.60	---	---	4.21	1.46	0.74
Y average error [m]	---	---	0.17	---	0.68	0.47	---	1.30	0.50	---	---	0.47	1.52	1.31
Z average error [m]	---	---	0.39	---	0.31	0.25	---	0.97	0.66	---	---	0.58	0.72	0.36
Optical and thermal landmarks														
Number (compare to OP)	---	---	139 (-27.6%)	---	---	171 (+32.6%)	---	---	132 (-2.2%)	---	---	221 (+18.2%)	213	168
X average error [m] (compare to OP)	---	---	1.30 (+39.8%)	---	---	1.06 (-27.4%)	---	---	3.06 (+45.7%)	---	---	1.66 (+0.0%)	1.82	0.89
Y average error [m] (compare to OP)	---	---	0.18 (+38.5%)	---	---	0.43 (-28.3%)	---	---	0.34 (-15.0%)	---	---	0.29 (-14.7%)	1.49	1.78
Z average error [m] (compare to OP)	---	---	0.31 (+40.9%)	---	---	0.26 (-39.5%)	---	---	0.31 (+47.6%)	---	---	0.40 (-7.0%)	0.58	0.39

Table 5.3.2: *Average errors for the detected landmarks in the cases analyzed in Section 5.1 and 5.2 (*OP = Optical single analysis, TH = Thermal single analysis, OP+TH = combined system).*

Chapter 6

Conclusions

In this thesis we have developed a solution for the SLAM problem using an optical and a thermal visual sensor. This work uses a previous work as a reference where a monocular SLAM solution was implemented using an optical camera [31]. The implementation of this technique is being carried out with success and permits the introduction of the novel aspect of this work: the use of the secondary thermal sensor to complement the existing optical sensor in the SLAM navigation task. The application of this type of camera is not been done in other SLAM solutions and in this thesis we proved that is a useful sensor for extracting information from the environment for navigation purpose. This work extents the state of the art with respect to the monocular single sensor SLAM approach of [18] [19] [31].

The evaluation of the presented system undertaken in the Chapter 5 has confirmed that the information added by the thermal camera improves the performance of the monocular SLAM approach developed in the previous work such as the increase of the number of detected landmarks, the decrease of the average error related to the landmark positions and finally improving the mobile robot position estimation throughout the time integration steps. The performance of the system is confirmed for different type of environments and in varying lighting conditions and finally has demonstrated the ability to cope with the presence of moving objects within the scene.

6.1 Future works

Future develop of this work regards the integration of the optical flow technique [34] as a tool to measure the displacement of the mobile robot between successive frames. The optical flow techniques should allow to add less noisy information about the robot position with respect to the sensors used in this project (i.e. encoders and GPS).

A useful technique is also the loop closing method [31] that consists in reuse the environmental information available from a previous visit of a area to increase the precision in the robot position estimation as outlined in the last example of Section 5.2.

The use of two thermal cameras could also be considered as introduce during the presentation of the fourth example in Section 5.2. Using two thermal cameras looking to each side of the mobile robot generate a larger field of view for the thermal sensor and could add more useful thermal information for the navigation of the mobile robot.

Finally testing the code and improving it to work in real time could allow the system to use the thermal camera data also for surveillance purpose adding extra modules such as a tracking technique [16].

References

[1] Gaszczak, A., Breckon, T. and Han, J. (2011), "Real-time people and vehicle detection from UAV imagery", Vol. 7878, SPIE The International Society for Optical Engineering, .

[2] Bay, H., Tuytelaars, T. and Gool, L. V. (2006), "Surf: Speeded up robust features ", *In ECCV* pp. 404.

[3] Matas, J., Chum, O., Urban, M. and Pajdla, T. (2004), "Robust wide-baseline stereo from maximally stable extremal regions", *Image and Vision Computing*, vol. 22, no. 10, pp. 761-767.

[4] iRobot Corporation 2010, *iRobot®, Robots that make a difference*, available at: http://www.iroboteurope.co.uk (accessed 2010, 05/02).

[5] The ABB Group, *ABB: Power and productivity for a better world*, available at: http://www.abb.com/product/us/9AAC129474.aspx (accessed 04/13).

[6] Google Maps, *Street View: Explore the world at street level* , available at: http://www.google.com/intl/en_us/help/maps/streetview/ (accessed 04/15).

[7] Greene, K., *Stanford's New Driverless Car.*, available at: http://www.technologyreview.com/read_article.aspx?id=18908 (accessed 04/15).

[8] iRobot Corporation 2010, *Government & Industrial*, available at: http://www.irobot.com/sp.cfm?pageid=109 (accessed 04/16).

[9] Zimmer, U. R. , *Autonomous underwater vehicles: a collection of groups and project*, available at: http://www.transit-port.net/Lists/AUVs.Org.html (accessed 05/02).

[10] NURC 2007, *NURC – Partnering for Maritime Innovation*, available at: http://www.nurc.nato.int/ (accessed 04/15).

[11] Desouza, G. N. and Kak, A. C. (2002), "Vision for mobile robot navigation: a survey ", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 2, pp. 237-267.

[12] Magnabosco, M. and Zoppellari, S. (2009), *Mappatura con griglie di occupazione per robot mobili* (unpublished Ingegneria Aerospaziale, corso di Robotica thesis), Università di Padova, Padova.

[13] Nüchter, A. (ed.) (2009), *3D Robotic mapping the simultaneous localization and mapping problem with six degrees of freedom*, Springer Tracts in Advanced Robotics.

[14] Last, J. C. and Main, R. (2009), *Techniques Used In Autonomous Vehicle Systems: A Survey* University of Northern Iowa, .

[15] Dissanayake, M. W. M. G., Newman, P., Clark, S., Durrant-Whyte, H. F. and Csorba, M. (2001), "A solution to the simultaneous localization and map building (SLAM) problem ", *IEEE Transactions on Robotics and Automation*, vol. 17, no. 3, pp. 229-241.

[16] Wang, C. C. and Thorpe, C. (2002), "Simultaneous localization and mapping with detection and tracking ", *IEEE International Conference on Robotics and Automation, 2002. Proceedings. ICRA '02.* Vol. 3, pp. 2918.

[17] Choi, J., Ahn, S. and Chung, W. K. (2005), "Robust sonar feature detection for the SLAM of mobile robot ", *EEE/RSJ International Conference on Intelligent Robots and Systems, 2005. (IROS 2005),* pp. 3415.

[18] Williams, B., Klein, G. and Reid, I. (2007), "Real-Time SLAM Relocalisation ", *IEEE International Conference on Computer Vision,* vol. 0, pp. 1-8.

[19] Zhang, Z., Huang, Y., Li, c. and Kang, Y. (2008), "Monocular vision simultaneous localization and mapping using SURF ", *7th World Congress on Intelligent Control and Automation, 2008. WCICA 2008,* pp. 1651.

[20] Paz, L. M., Pinies, P., Tardos, J. D. and Neira, J. (2008), "Large-Scale 6-DOF SLAM With Stereo-in-Hand ", *IEEE Transactions on Robotics,* vol. 24, no. 5, pp. 946-957.

[21] Sturgess, P., Alahari, K., Ladický, L. and Torr, P. H. S. (September 2009), "Combining Appearance and Structure from Motion Features for Road Scene Understanding ", *British Machine Vision Conference* London, .

[22] Muhammad, N., Fofi, D. and Ainouz, S. (2009), "Current state of the art of vision based SLAM ", *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series* Vol. 7251, feb, .

[23] Choset, H. and Nagatani, K. (2001), "Topological simultaneous localization and mapping (SLAM): toward exact localization without explicit localization ", *IEEE Transactions on Robotics and Automation,* vol. 17, no. 2, pp. 125-137.

[24] Sola, J., Monin, A., Devy, M. and Vidal-Calleja, T. (2008), "Fusing Monocular Information in Multicamera SLAM ", *IEEE Transactions on Robotics,* vol. 24, no. 5, pp. 958-968.

[25] Irani, M. and Anandan, P. (2000), "About Direct Methods", in Triggs, B., Zisserman, A. and Szeliski, R. (eds.) *Vision Algorithms: Theory and Practice,* Springer Berlin / Heidelberg, , pp. 267-277.

[26] Solomon, C. J. and Breckon, T. P. (2010), *Fundamentals of Digital Image Processing: A Practical Approach with Examples in Matlab,* Wiley-Blackwell.

[27] Harris, C. and Stephens, M. (1988), "A combined corner and edge detector", Manchester (UK), pp. 147-151.

[28] Liu, F. and Philomin, V. (September 2009), "Disparity Estimation in Stereo Sequences using Scene Flow", *British Machine Vision Conference,* London, .

[29] Lowe, D. G. (1999), "Object Recognition from Local Scale-Invariant Features", *Proceedings of the International Conference on Computer Vision,* Vol. 2, pp. 1150-1157.

[30] Durrant-Whyte, H. and Bailey, T. (2006), "Simultaneous Localisation and Mapping (SLAM): Part I The Essential Algorithms ", *IEEE Robotics and Automation Magazine,* vol. 2, pp. 2006.

[31] Lemaire, T., Berger, C., Jung, I. and Lacroix, S. (2007), "Vision-Based SLAM: Stereo and Monocular Approaches", *International Journal of Computer Vision*, vol. 74, no. 3, pp. 343-364.

[32] Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T. and VanGool, L. (2005), "A Comparison of Affine Region Detectors", *International Journal of Computer Vision*, vol. 65, no. 1/2, pp. 43-72.

[33] Evans, C. , *OpenSURF - Open Source SURF feature extraction library*, available at: http://code.google.com/p/opensurf1/ (accessed 03/09).

[34] Bradski, G. and Kaehler, A. (September 2008), *Learning OpenCV. Computer Vision with the OpenCV Library*, First ed, O'Reilly Media, Sebastopol,CA.

[35] Fischler, M. A. and Bolles, R. C. (June 1981), "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography ", *Communications of the Association for Computing Machinery*, vol. 24, no. 6, pp. 381-395.

[36] Mount, M. and Arya, S. , *ANN: A Library for Approximate Nearest Neighbor Searching. Version 1.1.2*, available at: http://www.cs.umd.edu/~mount/ANN/ (accessed 04/29).

[37] Ma, Y., Soatto, S., Kosecka, J. and Sastry, S. S. (2003), *An Invitation to 3-D Vision: From Images to Geometric Models*, SpringerVerlag.

[38] Schmidt, S. (1966), "Applications of state-space methods to navigation problems", *Advances in Control Systems*, vol. 3, pp. 293-340.

[39] Nebot, E. (May 2005), *Navigation System Design* (unpublished Centre of Excellence for Autonomous thesis), University of Sydney,Australia, .

[40] *NMEA library*, available at: http://nmea.sourceforge.net/ (accessed 06/08).

[41] Bouguet, J. Y., *Camera Calibration Toolbox for Matlab*, available at: http://www.vision.caltech.edu/bouguetj/calib_doc/ (accessed 03/15).

[42] Zhang, Z. (2000), "A flexible new technique for camera calibration ", *IEEE Transactions on on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330-1334.

[43] Lee, Y., Kwin, T. and Song, J. (October 2007), "SLAM of a mobile robot using thinning-based topological information", *International Journal of Control, Automation and Systems*, vol. 5, no. 5, pp. 577-583.

[44] Google Maps, *Explore the world using interactive maps*, available at: http://www.google.co.uk/help/maps/tour/ (accessed 12/02).