INSTITUTO
SUPERIOR
TÉCNICO

UNIVERSIDADE TÉCNICA DE LISBOA
INSTITUTO SUPERIOR TÉCNICO

Target Tracking with Pan-Tilt-Zoom Cameras

Diogo Filipe Cabral de Sousa Leite

Dissertação para a obtenção do Grau de Mestre em
Engenharia Electrotécnica e de Computadores

Júri:

Presidente:	Doutor Carlos Filipe Gomes Bispo
Orientador:	Doutor José António da Cruz Pinto Gaspar
Co-orientador:	Doutor Alexandre José Malheiro Bernardino
Vogal:	Doutor Pedro Manuel Quintas Aguiar

Novembro de 2011

Contents

Agradecimentos

É com grande satisfação que escrevo esta secção por finalmente ter atingido os resultados esperados. No decorrer dos últimos sete meses passei por momentos positivos e negativos mas no final acabo com um sentimento de dever cumprido e com a certeza que me desenvolvi, tanto a nivel cientifico como pessoal. Gostaria de deixar um agradecimento ao Professor Doutor José Gaspar a quem, como aliás já esperava, só consigo apontar aspectos positivos. O apoio incansável que prestou, assim como a postura exigente mas marcadamente formativa, contribuiram decisivamente para o desenvolvimento dos meus conhecimentos. Gostaria também de agradecer ao Professor Doutor Alexandre Bernardino que, apesar do muito trabalho a que esteve sujeito durante estes meses, contribuiu com o seu apoio sempre que necessário. A nível pessoal gostaria ainda de agradecer a várias pessoas que contribuiram para este trabalho:

- Manuela Cabral e Licinio Leite: Mãe e Pai, ninguém melhor como vocês sabe exactamente o que eu penso e sinto. Seja para o bem, ou para o mal, estiveram e estão sempre presentes.

- Rodolfo Reis: Grande irmão, tal como os pais a tua presença é obrigatória. És um exemplo para mim em muitos aspectos e sem ti, provavelmente, não estaria a escrever esta tese.

- Tiago Almeida, Pedro Peixoto, João Peixoto, Fernando Lopes, Fernando Figueiredo, João Brua e Liliana Pires: Vocês são verdadeiros amigos. Há muito tempo que nos conhecemos, portanto, sabem bem o motivo do agradecimento.

- Fábio Barata, Carlos Isidoro e David Isidoro: Desde que iniciei o meu trajecto universitário, vocês sempre estiveram presentes. O vosso apoio cientifico e motivacional foi, sem dúvida, decisivo para o sucesso do meu percurso.

- Colegas da ZTE Portugal: Desde que entrei na empresa sempre me auxiliaram, tanto a nível profissional, como académico, como pessoal.

Resumo

Apesar das vantagens intrínsecas do uso de câmaras pan-tilt-zoom em sistemas de vigilância automática o seu uso é ainda escasso. A dificuldade de criar modelos de fundo para câmaras que se movem, e a dificuldade de manutenção de modelos geométricos de projecção ajustados a cada configuração de pose e de óptica, são razões determinantes para a sua menor utilização. A calibração geométrica é uma ferramenta útil para ultrapassar estas dificuldades. O desenvolvimento de modelos de fundo e de projecção para as câmaras pan-tilt-zoom torna possível criar simuladores e metodologias de vigilância semelhantes aos que tipicamente se utilizam para câmaras fixas.

Neste trabalho, é proposta uma metodologia para a auto-calibração de câmaras PTZ para toda a gama de zoom da câmara. Este método baseia-se na minimização de erros de re-projecção de features detectadas em imagens capturadas pela câmara para diferentes orientações e níveis de zoom. Os resultados obtidos, tanto com dados sintéticos, como com dados reais, demonstram que é possível calibrar uma câmara pan-tilt-zoom sobre toda a sua gama de zoom e para todo o seu field-of-view. Também neste trabalho, é proposto um simulador de cenários de teste, construídos unicamente com dados reais, com múltiplos eventos com trajectórias ground truth. A última contribuição deste trabalho é uma nova metodologia para vigilância automática que usa seguimento e previsão da trajectória de eventos para melhorar os resultados de detecção. Esta metodologia é comparada com métodos existentes, através de experiências conduzidas em cenários de teste, com múltiplos eventos, gerados com o simulador proposto. Os resultados obtidos revelam grande eficiência e potencial do nosso método na percepção da presença de eventos num cenário.

Palavras chave: Câmara pan-tilt-zoom, auto-calibração, distorção radial, detecção de eventos, métodos de vigilância automática, percepção de eventos.

Abstract

Although there are intrinsic advantages of using pan-tilt-zoom cameras their application in automatic surveillance systems is still scarce. The difficulty of creating background models for moving cameras and the difficulty of keeping fitted pose and optical geometrical projection models are key reasons for the limited use of pan-tilt-zoom cameras. Geometric calibration is a useful tool to overcome these difficulties. Once developed the background and projection models, it is possible to design system simulators and surveillance methodologies similarly to the ones commonly available for fixed cameras.

In this work we propose a method for PTZ camera auto-calibration over the camera's zoom range. This method is based on the minimization of re-projection errors of feature points detected in images captured by the camera at different orientations and zoom levels. Results obtained over both synthetic and real data show that a full zoom range, complete field of view, pan-tilt-zoom camera calibration is possible. Also in this work, a simulator capable of generating highly flexible, real data only, test scenarios with multiple events having ground truth motion is proposed. The final contribution of the present work is a new methodology for automatic surveillance control that resorts to tracking and prediction of targets' trajectories to enhance event presence awareness performance. This methodology is presented and compared with existing ones, through experiments conducted over real data testing scenarios with multiple events generated through our simulator. The results obtained reveal a great efficiency and potential of our proposed method in event presence awareness in a given scenario.

Keywords: Pan-tilt-zoom camera, auto-calibration, radial distortion, event detection, automatic surveillance methodologies, awareness of presence events.

List of Figures

List of Tables

Chapter 1

Introduction

Despite the high versatility and potential of pan-tilt-zoom cameras, their use in current surveillance systems is still much less frequent than the use of fixed cameras. The limited dissemination of pan-tilt-zoom cameras can be justified by the slightly higher costs of the hardware and the higher risk of failure due to the mechanical components. This is however just a partial justification, since the hardware costs and failures can decrease significantly with mass production. Other, more compelling, justifications arise from the operation of the cameras and the difficulty of developing surveillance methodologies. Aiming to have mostly high quality employment implies that installations with numerous pan-tilt-zoom cameras cannot involve many human operators. This motivates developing automatic control and surveillance methodologies in opposition to the currently utilized manual control. The difficulty of creating background models for moving cameras and the difficulty of keeping fitted pose and optical geometrical projection models are also key reasons for the limited availability of automatic surveillance methodologies provided by the industry. Geometric calibration is a useful tool to overcome these difficulties.

Pan-tilt-zoom cameras have the ability to acquire high-resolution imagery and allow tracking events over a wide range in the environment. The positioning, setting of pan and tilt angles, is nowadays very precise, being commonly less than half a degree error. In order to take advantage of the flexibility and precision of the cameras one needs calibration to construct vast and accurate background models. Due to typical mountings at high places, it is desirable that the cameras auto-calibrate. In addition one needs to design control and surveillance methodologies that optimize the surveillance performance. The optimization involves creating controlled, repeatable and multiple-event test scenarios, using for example simulation providing ground truth. In this work experiments are conducted to assess the capabilities of different methodologies to enable complete integration of pan-tilt-zoom cameras in automatic surveillance systems.

The main subjects addressed in this work are pan-tilt-zoom camera calibration, with estimation of intrinsic and radial distortion parameters over the camera's full zoom range, creation of, real data, test scenarios with multiple events and pan-tilt surveillance control methodologies. The surveillance control methodologies are tested and compared resorting to metrics originally used for fixed cameras [11].

1.1 Related Work

In terms of pan-tilt-zoom camera calibration, there are several documented methods requiring physical access to the camera, such as the method proposed by Bouguet [3], where intrinsic and radial distortion parameters are estimated by changing the orientations of a chess pattern placed in front of the camera. Past work on active camera calibration was essentially conducted for geometric calibration only. Hartley [7] presented an auto-calibration method for stationary cameras and later Agapito et al. [2] introduced a self-calibration method for rotating and zooming cameras. These methods answered the problem of geometric calibration with solutions that did not require non-linear optimizations to achieve reasonable intrinsic parameters estimation. However, these methods did not address the problem of radial distortion parameters estimation which is an effect that greatly affects PTZ cameras. In Sinha and Pollefeys [14], a method for active pan-tilt-zoom camera calibration, using non-linear optimization of re-projection errors, is presented. The approach is to estimate intrinsic and radial distortion coefficients based on imagery taken from a small range of the camera's FOV. The method enables good estimations, however, such results are achieved at high computational costs. In all methods presented there is no approach which achieves accurate estimation of both intrinsic and radial distortion parameters at low computational costs and with no restrictions on the camera's field-of-view used.

Surveillance and camera control is an active research topic as there is no clear consensus on the appropriate surveillance methodologies for pan-tilt-zoom cameras. The great capacities of these cameras can only be fully exploited with appropriate pan and tilt controllers. In this work several pan-tilt surveillance control methodologies are presented and tested. The difference in how image segmentation information is used, in control feedback, enable the separation of these methods into two major categories: Open loop and closed loop methods. Open loop methods disregard any kind of information retrieved from past detections and thus pan and tilt angles are generated independently from past results. In closed loop methods the information from previous detections is used as control feedback in order to generate appropriate pan and tilt angles to enable tracking of specific targets. A paradigmatic approach in closed loop methods is the image-based look-and-move, which requires odometry-based control of the pan and tilt rotation axis. This is widely considered in the literature [9, 15] and, as the camera odometry information is available, it is the base for the closed loop methods implemented.

1.2 Proposed Approach

Although pan-tilt-zoom cameras have been introduced long time ago in the market, full zoom active calibration with accurate estimation of intrinsic and radial distortion parameters methods are still scarce in literature. Existing solutions resort to computational expensive non-linear optimization techniques to achieve calibration and are very restrictive in the FOV range of the imagery needed for such procedure. The method we propose, reduces the computational burden by reducing the optimization steps needed and adds the possibility to use any kind of image sequence, in terms of camera FOV used, to perform calibration.

Surveillance with fully calibrated pan-tilt-zoom cameras involves not only video processing but also controlling the pan and tilt angles. Collecting real data in which to perform experiments, having at the same time ground truth, is a complicated and error prone task. Consequently, past experiments on surveillance control methods, were mainly conducted in synthetically generated scenarios [15]. In order to account for realistic color histograms, we propose generating synthetic scenarios, as in the previous works, but now using real data i.e. images acquired by a pan-tilt-zoom camera. Simulating multiple simultaneous events, e.g. multiple persons moving around, can be introduced in the scenario by segmenting the foreground of a previously acquired image sequence, and then pasting it multiple times on the currently being generated scene.

In literature, many surveillance control methodologies are presented, however, there is a lack of experiments on closed-loop methods that exploit human characteristics, such as memory of a particular event and estimation of its movement based on its previous knowledge. The Timed Lock and Random Search method, described in this work, achieves tracking of events similar to other more common closed-loop methods, such as the Lock and Random Search. However, its ability to release tracked events to search for more, while knowledge of the presence of the released ones is maintained through prediction of their trajectories, sets it apart.

1.3 Thesis Structure

In Chapter 1 the problem to be addressed in the thesis is presented, as well as a brief discussion on the state of the art. In Chapter 2 the pan-tilt-zoom scene representation is addressed with the description of the camera model, the cube based representation used to model the background and the pan-tilt-zoom camera auto-calibration method developed. The proposed technique is then tested and calibration results are provided. In Chapter 3 the methods used to model background uncertainties and perform single event detection are described and followed by the presentation of the results obtained. Still in this chapter, a description of a procedure to create a simulated surveillance scenario with multiple real active events is proposed. In Chapter 4 three documented surveillance methodologies and a new, proposed one, are presented, evaluated and compared and, finally, in Chapter 5 this work is concluded and the future work is stated.

The work described hereafter was partially published in [10].

Chapter 2

Automatic Calibration

In this chapter we describe the geometric model of a pan-tilt-zoom camera as well as the background representation. The kinematics involved in pan and tilt camera movements are briefly addressed. Our proposal for full-zoom-range auto-calibration of pan-tilt-zoom cameras is presented. The chapter is finalized with synthetic and real data experiments documenting the proposed calibration method.

2.1 Camera Model

The pin-hole camera model for the perspective pan-tilt-zoom camera consists of a mapping from 3D projective space to 2D projective space. This is represented by a 3x4 rank-3 perspective matrix, $\mathbf{P}$. The mapping from 3D to the image plane takes a point $\mathbf{X} = (X, Y, Z, 1)^T$ to a point $\mathbf{u} = \mathbf{PX}$ in homogeneous coordinates.

The matrix $\mathbf{P}$ may be decomposed in $\mathbf{P} = \mathbf{K}[\mathbf{R}\ \mathbf{t}]$, where $\mathbf{t}$ is a 3x1 vector that represents the camera location, $\mathbf{R}$ is a 3x3 rotation matrix, computed as in equation 2.14, that represents the orientation of the camera with respect to an absolute coordinate frame, as shown in figure 2.1, and $\mathbf{K}$ is a 3x3 upper triangular matrix called the calibration matrix.

The entries of matrix **K** are identified as the intrinsic parameters of the camera (in pixel). **K** may be written as

$$\mathbf{K} = \begin{bmatrix} k_u & s & u_0 \\ 0 & k_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.1}$$

where k_u and k_v are the magnifications in the respective u and v directions, u_0 and v_0 are the coordinates of the principal point of the camera and s is a skew parameter (in this work we assume $s = 0$).

The camera's pan and tilt movements are modeled by simple rotations about its projective center $\mathbf{O}$, which was chosen to be the world origin and thus $\mathbf{t} = (0, 0, 0)^T$. The goal, in a calibration method, is to estimate the unknown parameters of a model $\mathbf{K}^{p,t,z}$ that provide the intrinsic parameters for any

($pan = p$, $tilt = t$ and $zoom = z$) admissible PTZ configuration.

Most cameras deviate from the pin-hole model due to radial distortion. This effect decreases with increasing focal length. Due to radial distortion a 3D point, $\mathbf{X}$, is projected to a point $\delta\mathbf{x_d} = (\delta x_d, \delta y_d)^T$. This point is deviated from the point $\mathbf{x} = (x, y)^T$ according to the radial distortion function, D:

$$[x_d\ y_d]^T = D\left([x\ y]^T;\ k_1,\ k_2\right) = \mathbf{L}(r)[x\ y]^T = (1 + k_1 r^2 + k_2 r^4)[x\ y]^T \tag{2.2}$$

where $r = \sqrt{x^2 + y^2}$.

This radial distortion model corresponds to a simplified two coefficient version of the one proposed by Heikkila [8] where r is the radial distance (distance from point $\mathbf{x}$ to the center of distortion (x_c, y_c)), $\mathbf{L}(r)$ is a radially symmetric distortion factor and k_1 and k_2 are the two radial distortion coefficients considered. For every zoom level z, D^z is parameterized by $(x_c^z, y_c^z, k_1^z, k_2^z)$. In our model the principal point (u_0, v_0) is constrained to be the center of distortion and so, the radial distortion function is only parameterized by coefficients k_1^z and k_2^z.

In pan-tilt-zoom cameras there are two possible movements: the camera is either rotating or zooming. Both movements can be represented by homographies transforming a point $\mathbf{u}$ to a point $\mathbf{u'}$. Let these two points be the projections of a 3D point, $\mathbf{X}$, to images taken at different time instants from a camera that is either rotating or zooming. These two points are related to $\mathbf{X}$ by $\mathbf{u} = \mathbf{K}[\mathbf{R}\ \mathbf{t}]\mathbf{X}$ and $\mathbf{u'} = \mathbf{K'}[\mathbf{R'}\ \mathbf{t}]\mathbf{X}$, where $\mathbf{t} = (0, 0, 0)^T$, hence $\mathbf{u'} = \mathbf{K'R'R^{-1}K^{-1}u}$. In the model proposed the intrinsics remain the same for pure rotation at constant zoom ($\mathbf{K} = \mathbf{K'}$) so this equation may be simplified to $\mathbf{u'} = \mathbf{KR_rK^{-1}u}$, where $\mathbf{R_r} = \mathbf{R'R^{-1}}$ is the relative camera rotation about its projection center between two views, and $\mathbf{K}$ is the matrix of intrinsic parameters at a certain zoom level. If the camera is zooming but has fixed orientation and principal point, the equation is simplified $\mathbf{u'} = \mathbf{K'K^{-1}u}$. Concluding the two transformations, of point $\mathbf{u}$ to point $\mathbf{u'}$, are described by two homographies, $\mathbf{H}_{rot}$ and $\mathbf{H}_{zoom}$.

$$\mathbf{H}_{rot} = \mathbf{K}\mathbf{R}_r\mathbf{K}^{-1} \tag{2.3}$$

$$\mathbf{H}_{zoom} = \mathbf{K'}\mathbf{K}^{-1} \tag{2.4}$$

2.2 Cube Based Representation

There are several models to represent the background of a pan-tilt-zoom camera such as a cylinder, a plane, a sphere or a cube. We chose the cube based representation as it enables one to have a complete spherical field of view, 360^o x 360^o. Additionally this representation has limited memory requirements as the background can be completely represented using only six images. The cube shares the same coordinate frame of the camera at its homing position. That being z pointing forward, x pointing right and y pointing down (See Figure 2.1).

The representation is achieved in two steps: first it is necessary to compute a back-projection for

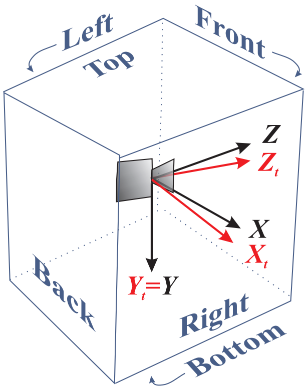


Figure 2.1: Cube based representation for the background of the pan-tilt-zoom camera and coordinate frame adopted.

each image point and find the correct cube face, and finally, the back-projection rays obtained must be projected to the right cube face ([5, 4]).

2.2.1 Back-Projection and Identification of the Correct Cube Face

With the knowledge of the camera's intrinsic parameters and the pan and tilt angle orientations used when acquiring a specific image, it is possible to back-project each 2D image point to a 3D world point:

$$[X\ Y\ Z]^T = (\mathbf{KR})^{-1}\mathbf{u} \tag{2.5}$$

The world point, $[X\ Y\ Z]^T$, can then be scaled to touch the lateral faces of a cube having edges with lengths $2L$ as is represented in equation 2.6. With the information of the 3D world point and the size of the cube faces, it is possible to define a latitude angle, dependent on longitude, known as critical latitude ($\varphi_c(\theta)$). This angle is computed resorting equation 2.7.

$$[X_c\ Y_c\ Z_c]^T = [X\ Y\ Z]^T L/\max(|X|,|Z|) \tag{2.6}$$

$$\varphi_c(\theta) = \arctan(\max(|X|,|Z|)/\sqrt{X^2+Z^2}) \tag{2.7}$$

Given the computed critical latitude, $\varphi_c(\theta)$, and by converting world coordinates, $[X\ Y\ Z]^T$, to spherical coordinates longitude, $\theta = \arctan(X/Z)$, and latitude, $\varphi = \arctan(-Y/\sqrt{X^2+Z^2})$, it is possible to determine the correct cube face for each point. The rules that determine such correspondence are shown in table 2.1.

Condition.	Cube Face
$\varphi \geq \varphi_c(\theta)$	Top
$\varphi \leq -\varphi_c(\theta)$	Bottom
$\lvert\varphi\rvert < \varphi_c(\theta) \wedge \lvert\theta\rvert \leq 45^o$	Front
$\lvert\varphi\rvert < \varphi_c(\theta) \wedge \lvert\theta\rvert \geq 135^o$	Rear
$\lvert\varphi\rvert < \varphi_c(\theta) \wedge 45^o < \theta < 135^o$	Right
$\lvert\varphi\rvert < \varphi_c(\theta) \wedge -135^o < \theta < -45^o$	left

Table 2.1: Rules for the matching 3D directions to cube faces.

2.2.2 Projection to Cube Face

The final step consists in the mapping of each image point to its corresponding cube face, as determined in section 2.2.1. This mapping consists in the projection of the back-projection ray previously computed to the respective cube surface, through application of the projection matrix, $\mathbf{P_{WF}}$:

$$\mathbf{P_{WF}} = \mathbf{K_F} \begin{bmatrix} \mathbf{R_{WF}} & 0_{3x1} \end{bmatrix} \tag{2.8}$$

where $\mathbf{K_F}$ is an intrinsic matrix characterizing the resolution of the cube faces and $\mathbf{R_{WF}}$ are rotational matrices defining optical axis orthogonal to each cube face.

More precisely, if one considers each cube face having $N \times N$ pixels then:

$$\mathbf{K_F} = \begin{bmatrix} (N+1)/2 & 0 & (N-1)/2 \\ 0 & (N+1)/2 & (N-1)/2 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.9}$$

which represents a perspective camera with a $90^o \times 90^o$ field of view and an image coordinate system such that the top-left pixel coordinate is $(1, 1)$. The rotation matrices $\mathbf{R_{WF}}$ in essence rotate the 3D points closest to each of the faces, of the cube, towards the front face.

In more detail, $\mathbf{R_{WF}}$ is $I_{3\times3}$, $Rot_Y(180^o)$, $Rot_Y(-90^o)$, $Rot_Y(+90^o)$, $Rot_X(-90^o)$ or $Rot_X(+90^o)$ for the front, back, left, right, top or bottom cube faces, respectively. $Rot_Y Rot_X$ denote the following matrices:

$$Rot_X = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(\beta) & -sin(\beta) \\ 0 & sin(\beta) & cos(\beta) \end{bmatrix} \tag{2.10}$$

$$Rot_Y = \begin{bmatrix} cos(\alpha) & 0 & sin(\alpha) \\ 0 & 1 & 0 \\ -sin(\alpha) & 0 & cos(\alpha) \end{bmatrix} \tag{2.11}$$

In summary, an image point $\mathbf{u}$ is mapped to a point on a cube face $\mathbf{u_F}$ as:

$$\mathbf{u_F} \sim \mathbf{K_F R_{WF} R^{-1} K^{-1} u} \tag{2.12}$$

where $\sim$ denotes equality up to a scale factor.

2.3 Pan-Tilt Direct and Inverse Kinematics

The rotations performed by the camera, pan and tilt, perform transformations mapping 3D world points, $\mathbf{^wX} = [^wX\ ^wY\ ^wZ\ 1]$ to the camera's reference frame, $\mathbf{^cX} = [^cX\ ^cY\ ^cZ\ 1]$. This is a simple linear transformation performed solely by the rotation matrix, $\mathbf{R}$, only if there are no translations between the reference frames (See equation 2.13). The rotation matrix, $\mathbf{R}$, is thus a linear transformation mapping $\mathbf{^wX}$ to $\mathbf{^cX}$. This matrix is a function of pan (α) and tilt (β) as shown in equations 2.14 and 2.15.

$$\mathbf{^cX} = \mathbf{T}.\mathbf{^wX} = \mathbf{R}.\mathbf{^wX} \tag{2.13}$$

$$\mathbf{R} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(\beta) & -sin(\beta) \\ 0 & sin(\beta) & cos(\beta) \end{bmatrix} \times \begin{bmatrix} cos(\alpha) & 0 & sin(\alpha) \\ 0 & 1 & 0 \\ -sin(\alpha) & 0 & cos(\alpha) \end{bmatrix} = \tag{2.14}$$

$$= \begin{bmatrix} cos(\alpha) & 0 & sin(\alpha) \\ sin(\beta)sin(\alpha) & cos(\beta) & -sin(\beta)cos(\alpha) \\ -sin(\alpha)cos(\beta) & sin(\beta) & cos(\beta)cos(\alpha) \end{bmatrix}. \tag{2.15}$$

Given the defined coordinate systems, world reference frame and camera reference frame, the centering of an object in an image is simply stated as the regulation of the pan and tilt angles so that the centroid of that object is moved to the image center. Back-projecting this centroid point (equation 2.5) gives a valid 3D point ($^c\mathbf{M} = [^cM_x\ ^cM_y\ ^cM_z]^T$) that can then by moved to the origin by applying the right pan and tilt angles (equation 2.16).

$$^c\mathbf{M} = \mathbf{R}(\alpha + \delta\alpha, \beta + \delta\beta).[0\ 0\ ||^c\mathbf{M}||]^T \tag{2.16}$$

where $\delta\alpha$ and $\delta\beta$ represent the pan and tilt increments that need to be found. This problem has a single solution if $\delta\alpha\ \epsilon\ [0^o, 360^o]$ and $\delta\beta\ \epsilon\ [-90^o, 90^o]$. This process can be further simplified by assuming that the initial pan and tilt angles are null. Given this, their increments can be easily computed through equations 2.17 and 2.18.

$$\delta\alpha = -\arctan(^cM_x/^cMz) \tag{2.17}$$

$$\delta\beta = -\arctan(^cM_y/\sqrt{^cMx^2 +^c Mz^2}) \tag{2.18}$$

2.4 Calibration Methodology

As stated in section 1.1, there are several documented methods for pan-tilt-zoom auto-calibration. In particular, there are two main methodologies: in the first methodology pan-tilt-zoom auto-calibration is done assuming that the camera is well represented by a pin-hole model (Agapito et al. [2]). It enables the estimation of intrinsic parameters for different zoom levels but does not perform radial distortion parameters estimation. In the second methodology the camera is represented by a pin-hole model combined with radial distortion (Sinha and Pollefeys [14]), hence implying non-linear numerical optimization. This method relies on the detection and matching of feature points, from different captured images, with subsequent projection of points to a plane and minimization of re-projection errors. The use of a plane limits the field-of-view due to problems associated with the projection of points tangent to such a representation. This limitation implies the use of few images for calibration greatly reducing feature information essential to the estimation process. This problem associated with an extrinsic parameter estimation results in the use of four optimization steps to fully calibrate the camera over its entire zoom range, thus making it a highly computational expensive method. In this thesis, the approach was to use non-linear optimization of re-projection errors [14]. In our computationally less expensive, field of view robust, approach, the necessary optimization steps are reduced to two: the intrinsics and radial distortion coefficients are first iteratively estimated at minimum zoom level and then computed for an increasing zoom sequence. The reduction of the associated computational burden is achieved by not estimating extrinsic parameters, as the camera odometry is accurate, and by using a sphere as a back-projection model thus allowing the use of more feature information.

2.4.1 Calibration at minimum zoom level

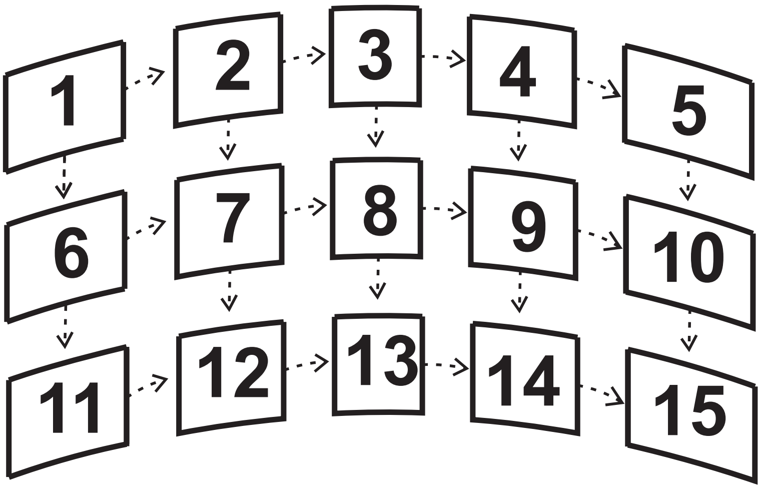


Figure 2.2: Spherical grid of images used for minimum zoom camera calibration.

The first step of our method is computing the intrinsics and radial distortion coefficients at minimum zoom level (z_0). To achieve this, images are captured at various pan-tilt location describing a spherical grid with constant pan-tilt steps (Figure 2.2). The acquired images have intersecting fields of view. The intersections are described in a graph whose nodes denote the images and the links contain the locations of corresponding feature points. Feature points in every image are detected and matched for all possible image pair combinations through SIFT algorithm. This is followed by a match filtering that excludes all matches not forming closed loops. Homographies between every adjacent horizontal images, $\mathbf{H_i}$ and between every adjacent vertical images, $\mathbf{V_i}$ are then robustly computed using RANSAC-based homography estimation and nonlinear minimization [6] (Chapter 3, p. 108). One of the images, $\mathbf{I_r}$ is chosen to be the reference. Homographies $\mathbf{T_i}$ mapping points from the other images $\mathbf{I_i}$ to $\mathbf{I_r}$ are computed through composition of the previous obtained $\mathbf{H_i}$ and $\mathbf{V_i}$ homographies. Since residual errors accumulate in the composed homographies it is not possible to build an accurate mosaic at this point. To achieve global image alignment and an accurate geometric calibration of the camera a non-linear optimization of the re-projection errors of all image feature points is applied using Levenberg-Marquardt algorithm [16]. The cost function (See Eq. 2.19) used to solve this global minimization problem is initialized with the matrix of intrinsic parameters $\mathbf{K^{z_0}}$, obtained using Agapito's method [1], and null radial distortion coefficients ($(k_1, k_2) = (0, 0)$).

$$(\mathbf{K^{z_0}}^*, \mathbf{D^{z_0}}^*) = \arg \min_{\mathbf{K^{z_0}}, \mathbf{D^{z_0}}} \sum_{j=1}^{m} \sum_{i=1}^{n} ||\mathbf{u_i^j} - \hbar(\mathbf{K}^{z_0}\mathbf{D}^{z_0}(R_i X^j;\ k_1,\ k_2))||^2 \tag{2.19}$$

where $\mathbf{D}$ is a homogenization of D, defined in 2.2, i.e. it adds a unitary third coordinate thus transforming a 2-vector to a 3-vector output, $\hbar$ is a dehomogenization function, i.e. $\hbar([a\ b\ c]^T) = [a/c\ b/c]^T$, $\mathbf{u_i^j}$ are the observed features for the given image $\mathbf{I}_i$, $\mathbf{R}_i$ are the rotation matrices, computed through direct kinematics (See section 2.3, equation 2.15) given the camera´s odometry (i.e. pan and tilt angles), for the respective images, m and n are the feature-count and image-count and $\mathbf{X}^j$ is a global feature list. This global list is computed by correctly labeling every 3D world point in order to have a unique description (See Section 2.4.3).

2.4.2 Zoom sequence calibration

To perform a zoom range calibration the camera is fixed at a certain orientation and images for progressive zoom levels are acquired in the form of a graph as shown in figure 2.4. The computation steps that follow are similar to the ones applied in the minimum zoom calibration method (See Section 2.4.1) with the exception of the homographies computation. In fact, there is no need to compute initial intrinsic parameters as in the first zoom step one has the estimate obtained as described in the previous section. In other words homographies are unnecessary. In zoom sequence calibration, parameters in $\mathbf{K}^{z_i}$ and $\mathbf{D}^{z_i}$ are iteratively estimated for every zoom level i with the starting estimates being the parameters found for zoom level $i - 1$, $\mathbf{K}^{z_{i-1}}$ and $\mathbf{D}^{z_{i-1}}$. This second step of the calibration method is initialized with the estimates for the parameters obtained as result of the first step calibration method, $\mathbf{K}^{z_0}$ and

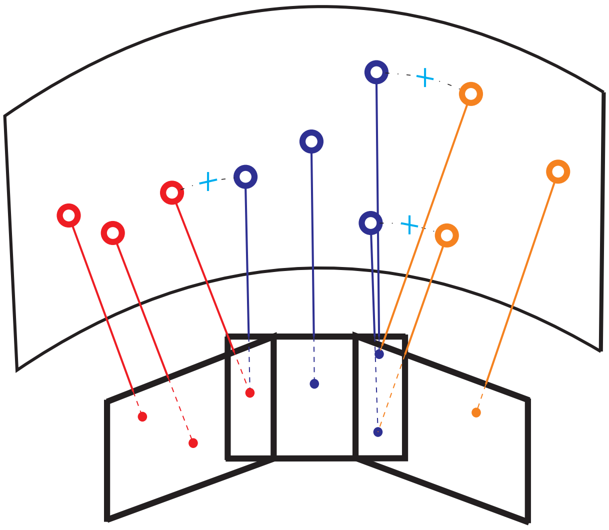


Figure 2.3: Back-projection of points found in three overlapping images. Red, blue and orange points correspond to images leftmost, center and rightmost, respectively. Some points are seen in more than one image thus forming clusters in the unit sphere, which are represented by their median points, shown as cyan crosses.

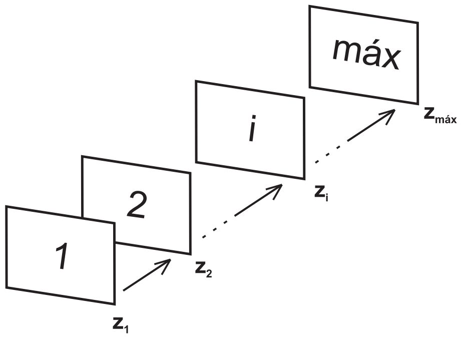


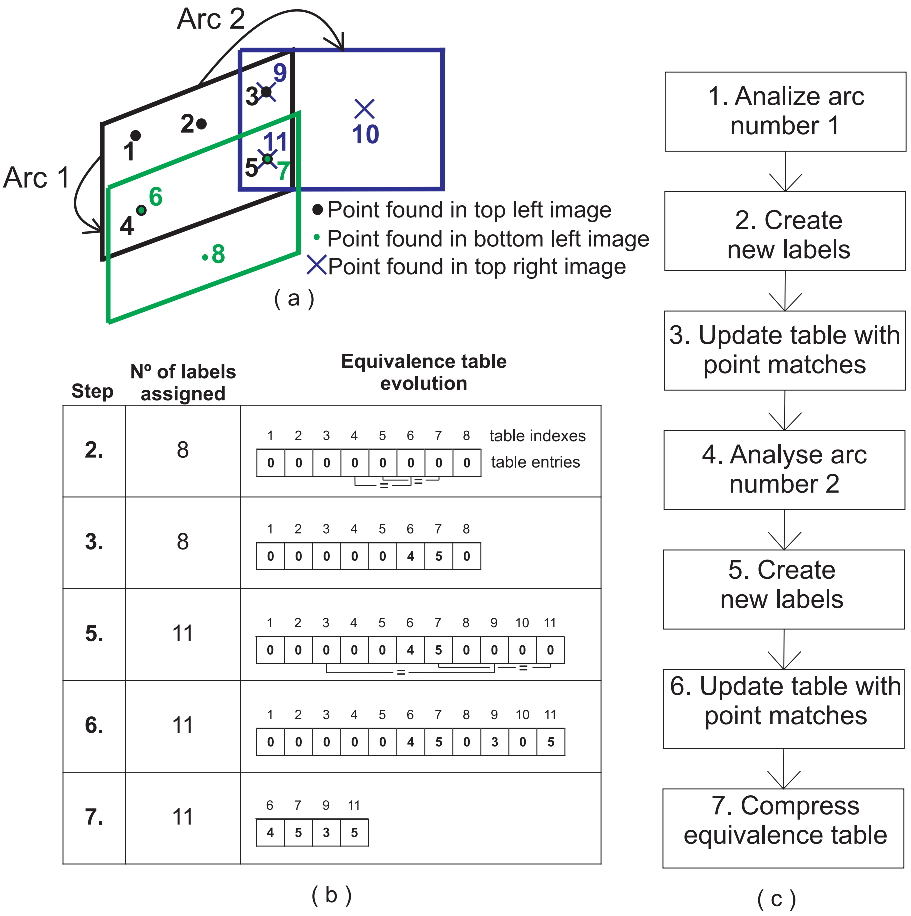
Figure 2.4: Image graph created to perform zoom sequence calibration.

$\mathbf{D}^{z_0}$. These are the only parameters kept fixed through the optimization process, this is to avoid radial distortion functions being compensated in subsequent zoom levels. The cost function (See Eq. 2.20) for this problem is evaluated for every zoom level z_i:

$$(\mathbf{K^{z_i}}^*, \mathbf{D^{z_i}}^*) = \arg \min_{\mathbf{K^{z_i}}, \mathbf{D^{z_i}}} \sum_{j=1}^{m} \sum_{k=i-1}^{i} ||\mathbf{u_k^j} - \hbar(\mathbf{K}^{z_k}\mathbf{D}^{z_k}(X^j;\ k_1,\ k_2))||^2 \tag{2.20}$$

To evaluate the uncertainties associated to the radial distortion coefficients, k_1 and k_2, the method is applied to several zoom sequences acquired at different camera orientations. At some zoom level the estimates of these parameters becomes very incoherent. At these particular zoom levels and for the subsequent ones the respective coefficient is clamped to zero.

2.4.3 Unique labeling of 3D points


Arc 2
Arc 1
1
2
3
9
4
5
6
7
8
10
11
● Point found in top left image
• Point found in bottom left image
✕ Point found in top right image
(a)
(b)
1. Analize arc number 1
2. Create new labels
3. Update table with point matches
4. Analyse arc number 2
5. Create new labels
6. Update table with point matches
7. Compress equivalence table
(c)

Step	Nº of labels assigned	Equivalence table evolution
2.	8	1 2 3 4 5 6 7 8 table indexes 0 0 0 0 0 0 0 0 table entries = =
3.	8	1 2 3 4 5 6 7 8 0 0 0 0 0 4 5 0
5.	11	1 2 3 4 5 6 7 8 9 10 11 0 0 0 0 0 4 5 0 0 0 0 = =
6.	11	1 2 3 4 5 6 7 8 9 10 11 0 0 0 0 0 4 5 0 3 0 5
7.	11	6 7 9 11 4 5 3 5

Figure 2.5: Example of unique labeling for eleven points distributed through three images. (a) Display of points and theirs correspondences, (b) Equivalence labels table evolution, (c) Steps associated to the equivalence table evolution.

The problem of unique labeling of 3D points appears in the context of the proposed calibration methodology as it is necessary to have a global list of all feature points detected. Corresponding image points back-project to different 3D points in case the parameters of the projection model are not precisely known. Choosing the means of back-projected points, and re-projecting them over the images, one obtains 2D mismatches whose minimization (equation 2.20) serves as an indicator of the precision of the calibration parameters.

Consequently one needs to label uniquely 3D back-projected points such that 3D means can be computed. This implies the creation of an equivalence labels table with the size of all feature points detected. The fields of this table are filled according to the point correspondences found. To achieve this, the algorithm loops all links inter images and with the information retrieved from them (start image, end image, features in each of the two images and possible correspondences between them) proceeds based on four possible outcomes determined by a table look-up evaluation: (1) The table entries for both start and end nodes have not yet been filled, (2) only the table entries for the start node have been filled, (3) only the table entries corresponding to the end node have been filled and (4) both start and end node entries have already been filled.

1. *Both start and end nodes empty*

   Here new labels are assigned to all feature points in both images and the equivalence labels table is filled according to the correspondences between the two sets of points. This information is retrieved from the corresponding arc. In filling the equivalence table the priority is given to the points with a lower label number, so points indexed higher are labeled with the correspondent lower labeled points (See Figure 2.5).

2. *Empty only on end node*

   First, new labels are assigned to points found in the end node. Then the equivalence table is filled according to the correspondences found with the start image: entries of points with no matches are filled with zeros meaning these points are labeled with their own index in the table and points matched in the two images get the lowest label possible and so the entries of the higher indexed points are filled with the index of their matching point.

3. *Empty only on start node*

   If only the start node entries are empty then new labels are assigned to the respective points and the equivalence table is filled accordingly (See Subsection 2.4.3 2. Empty only on end node).

4. *Both start and end nodes filled*

   If points in both images are already labeled and the correspondent equivalent table entries filled there is an evaluation to verify if each table entry of all matched points is the correct one. If that is the case, then no further action is taken regarding the entries of that particular point. If the table entries do not represent the matching of a given point between the two images then the table is modified to contemplate this point correspondence.

The equivalence table is characterized by its indexes and their respective entries. The indexes are the initial labels for every point found in every analyzed image. Their fields give information relative to the correspondences between them. When the process of table filling is finished its entries contain either zeros or multiple positive integers. The zero entry indexes are the only labels needed to create the unique list of 3D feature points and so these constitute the final labels. The indexes of non-zero entries correspond to labels that can be discarded and substituted by the ones in the corresponding fields. These are contained in the final labels list and represent the feature point correspondences. The final step is a table compression where only the non-zero entries remain because they are the only ones that give relevant information as the missing ones do not need to be modified.

2.5 Results

In this section we present the experiments conducted with the presented calibration method. These experiments contemplated both synthetic and real data.

2.5.1 Synthetic Data

First the method was tested with synthetic data. For this purpose a Canon VB-C10 PTZ camera was simulated and four thousand 3D points were created and randomly placed in from of it at various random distances (See Figure 2.6).

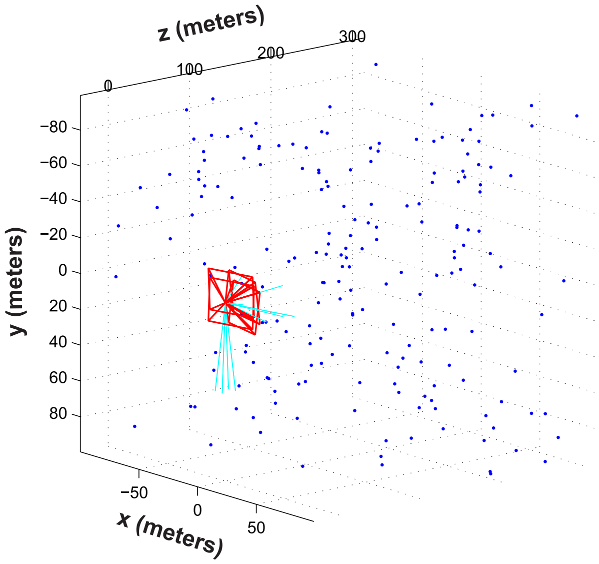
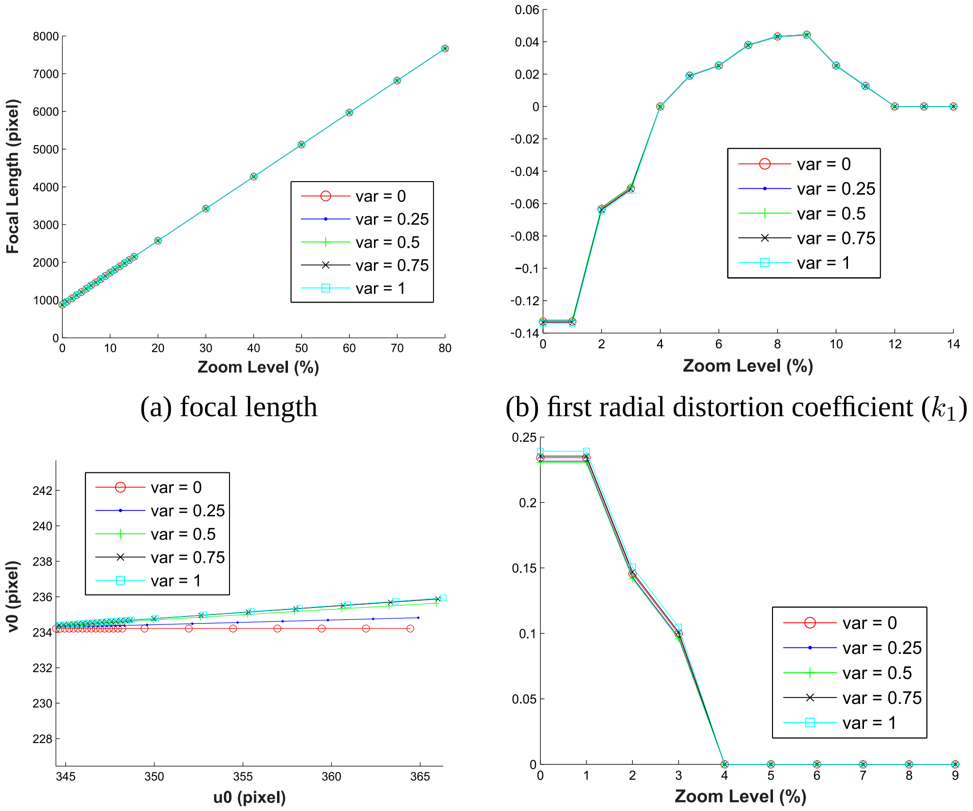


Figure 2.6: Display of synthetic camera and 3D points generated.

For this camera both the intrinsic and radial distortion parameters were known, for multiple zoom levels, and so, these values constituted the ground truth used to compare with estimations obtained through our method. The experiments were conducted in two calibration steps as described in section 2.4. The results obtained consisted of intrinsic and radial distortion parameters estimated for an increasing zoom sequence (Figure 2.7).



(a) focal length

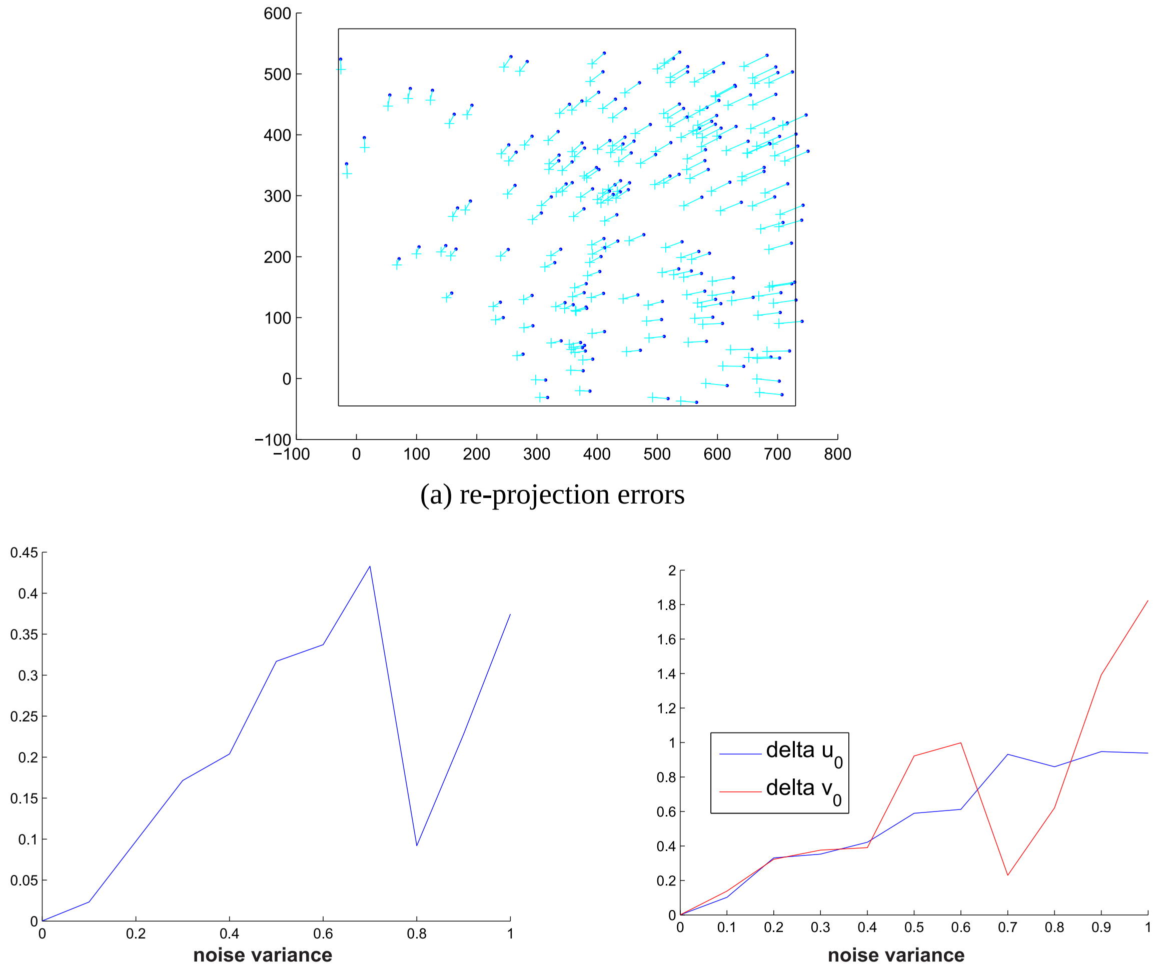
(b) first radial distortion coefficient (k_1)

(c) principal point

(d) second radial distortion coefficient (k_2)

Figure 2.7: Synthetic camera's intrinsic and radial distortion parameter estimations for multiple noise levels.

The experiments were conducted using random Gaussian noise, with zero mean and variable variance. The noise was applied in the re-projection of back-projected points i.e. in the image plane, resulting in random re-projection errors. These errors are greater in the beginning of the calibration method as can be seen in figure 2.8 (a), where the crosses represent the original points projected to the image plane from 3D projective space and the blue dots are their correspondent re-projections. The re-projection errors for the first image plane are displayed right before the application of our method. These re-projection errors are then iteratively minimized by our non-linear optimization method but, due to the applied noise, the parameters are subject to estimation uncertainties as is represented in figures 2.8 (b) and (c).



(a) re-projection errors

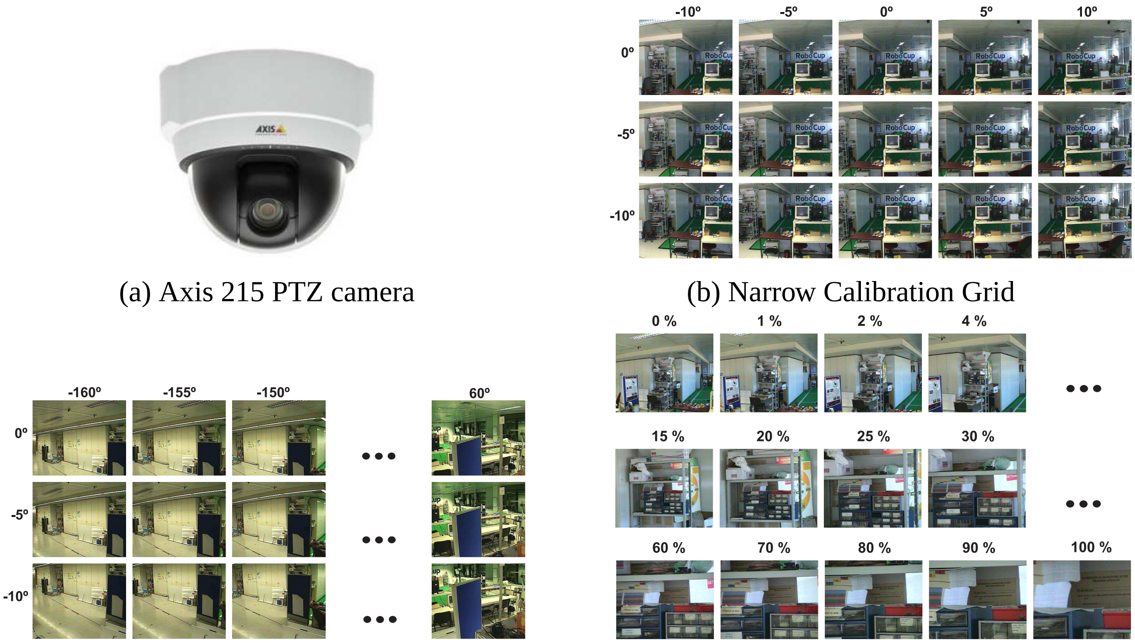
(b) focal length uncertainties

(c) principal point uncertainties

Figure 2.8: Display of re-projection errors in the first image plane (a) and the uncertainties associated to the estimations of the intrinsic parameters, (b) and (c). In image (a) the cyan crosses represent original points, projected to the image plane from 3D projective space, and blue dots correspond to their matching re-projections.

2.5.2 Real Data Calibration at Minimum Zoom

To test the calibration method in a real environment setting we used an Axis 215 PTZ camera (See Figure 2.9 (a)). The camera movement was limited according to its specifications. To perform the minimum zoom calibration method a grid of 15 images ranging from -10^o to 10^o pan, and from -10^o to 0^o tilt, was acquired (See Figure 2.9 (b)). To test the robustness of the method an additional wider grid ranging from -160^o to 60^o pan, and from -10^o to 0^o tilt, was acquired (See Figure 2.9 (c)). To assess the quality of the estimations Bouguet's method [3] was applied to obtain ground truth values for the intrinsic and radial distortion parameters. For both sets of images all the information regarding each arc in the respective image graph was computed (SIFT feature detection and matching) and the calibration method was applied.



(a) Axis 215 PTZ camera

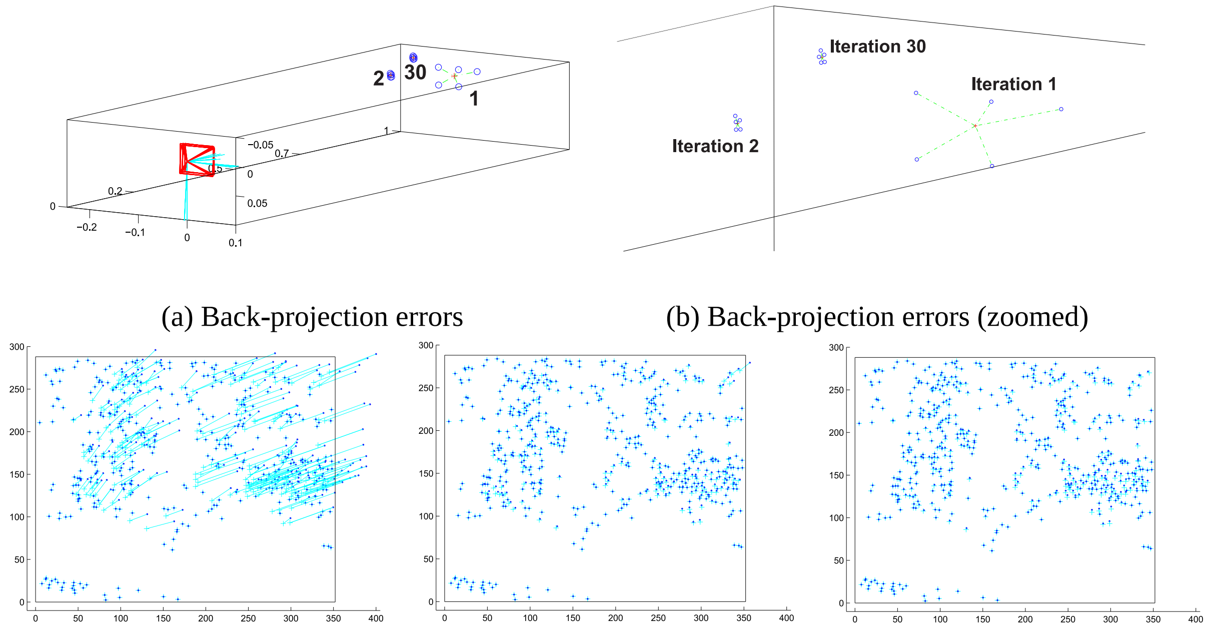
(b) Narrow Calibration Grid

(c) Wide Calibration Grid

(d) Zoom Sequence

Figure 2.9: Camera used in real setting (a) and images acquired to calibrate it for minimum zoom level, (b) and (c), and for its zoom range (d). The sequences of images used for calibration are better displayed, in larger figures, in Appendix A.

As described in section 2.4, our calibration method consists in back-projecting the feature points detected in every image to a unit sphere centered in the camera's position. Then every cluster of back-projected points (feature points common to several images planes) is represented by it's median point, as represented in figure 2.3, which is then re-projected to the images contributing to the respective cluster. This re-projected point deviates from its original point in each image, thus originating an error. This re-projection error is due to incorrect intrinsic and radial distortion parameters, applied in back-projection and re-projection. Our method iteratively minimizes the re-projection errors in every image producing better estimates of these parameters. The evolution of our method is graphically represented in the back-projection of a point for the first, the third, and the last iteration (See figures 2.10 (a) and (b)) and in the re-projection errors of all points, in the first image plane, for the same calibration method iterations (See figures 2.10 (c), (d) and (e)). Having obtained better estimates for the intrinsic and radial distortion parameters, errors tend to decrease and so, with each iteration back-projections errors decrease. In fact, and as figure 2.10 (b) shows, the cluster of points back-projected tends to close in on its median point, underlying the existence of an intersection point, in the unit sphere, for all back-projection rays produced by matching points originated in different image planes.



(a) Back-projection errors

(b) Back-projection errors (zoomed)

(c) Re-projection errors (iteration 1)

(d) Re-projection errors (iteration 2)

(e) Re-projection errors (iteration 30)

Figure 2.10: Display of the back-projection (a) and (b), and re-projection (c), (d) and (e) errors for three iterations of our calibration method.

The estimations obtained with our method are presented in table 2.2, for both the narrower and the wide field of view calibration grids. Additionally, this table contains the ground truth values computed using the calibration method proposed by Bouguet [3] and the intrinsics computed through Agapito's method [1].

With the obtained results it is possible to construct precisely a cube based representation of the scenario. For this purpose 1220 images of the environment where acquired and filtered to only include the ones taken with camera orientations ranging from -170^o to 170^o pan and -70^o to 0^o tilt. With these angle restrictions uncontrollable camera movements are avoided, such as auto-flip and e-flip, while maintaining full description of the surrounding scenario. Mosaics where created with intrinsic and radial distortion parameters estimated by Agapito's method (See figure 2.11) and with the same parameters

	G. T.	A. M.	N. G.	W. G.
k_u	383.46	742.55	384.12	386.27
k_v	420.10	548.80	409.61	419.37
u_0	178.18	142.62	195.99	203.64
v_0	149.76	159.04	155.75	151.36
k_1	-0.3093	0	-0.2774	-0.2876
k_2	0.1643	0	0.1151	0.0952

Table 2.2: Calibration results for real data experiments. Ground truth values (G.T.), Agapito's method estimations (A.M.) and the estimations from our calibration method for both the narrower (N.G.) and the wider (W.G.) FOV grids of images.

estimated through our method using the wider FOV calibration grid of images (See figure 2.12).

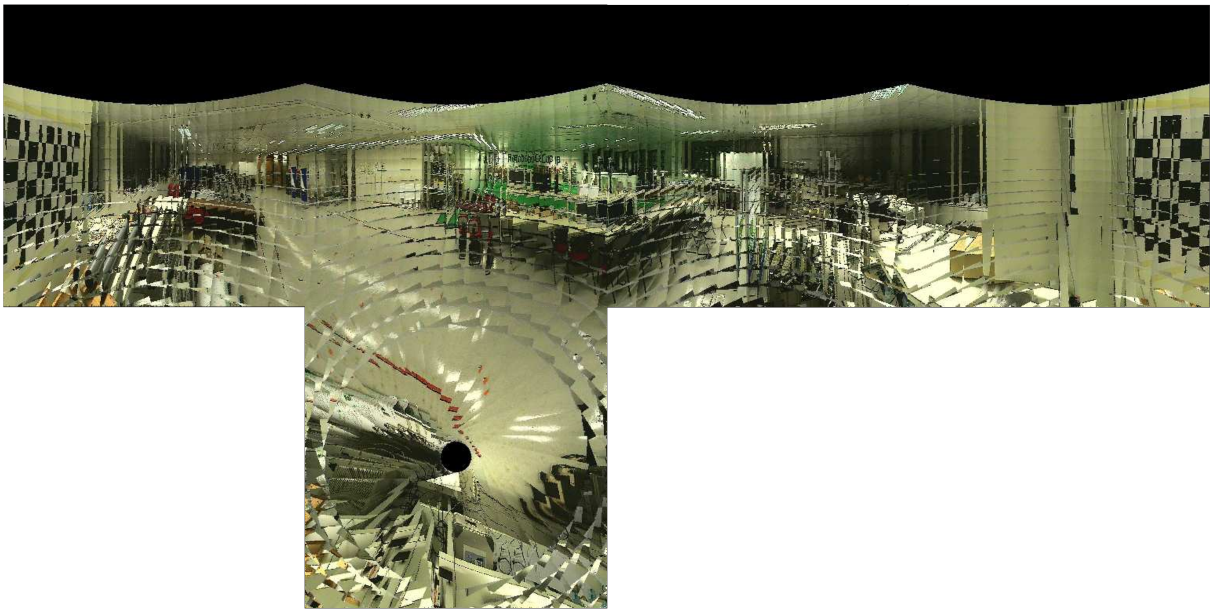


Figure 2.11: Cube mosaic created using Agapito's estimated parameters.

2.5.3 Real Data Calibration for Increasing Zoom Sequence

To achieve calibration for increasing zoom steps, 25 images were acquired, from a fixed camera orientation, ranging from minimum zoom level $z_0 = 0\%$ to maximum zoom level $z_{max} = 100\%$. These images were acquired with smaller steps for the initial zoom levels to minimize the strong effects of radial distortion. These steps went progressively bigger for higher zoom levels. The results obtained for the intrinsic and radial distortion parameters for increasing zoom steps are displayed in figures 2.13 (a), (b), (c) and (d).

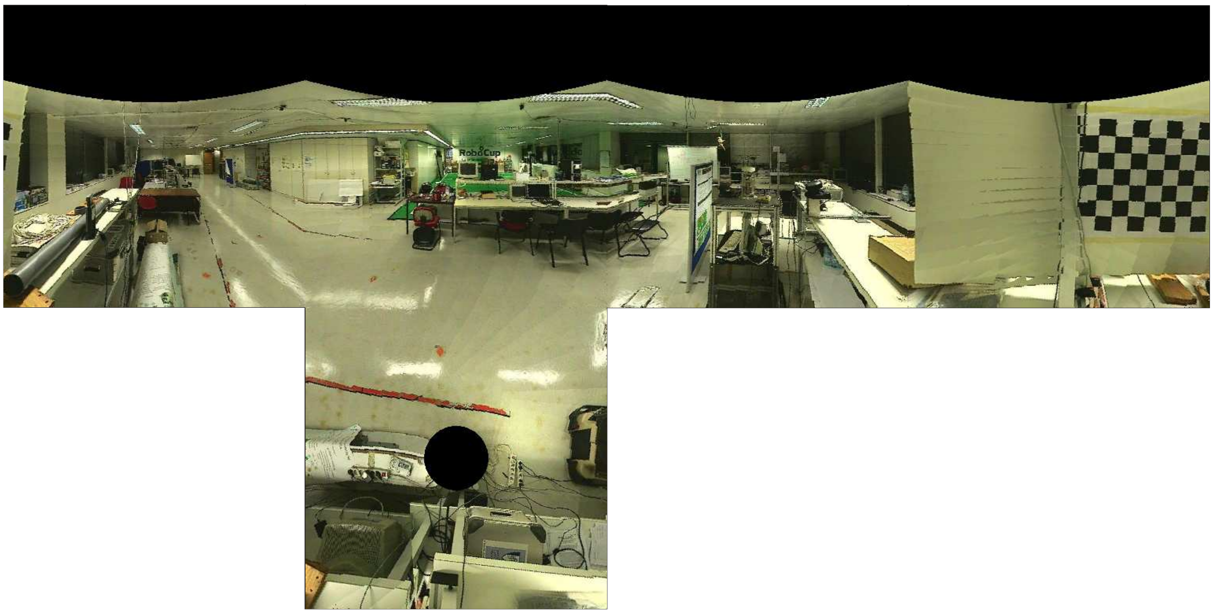
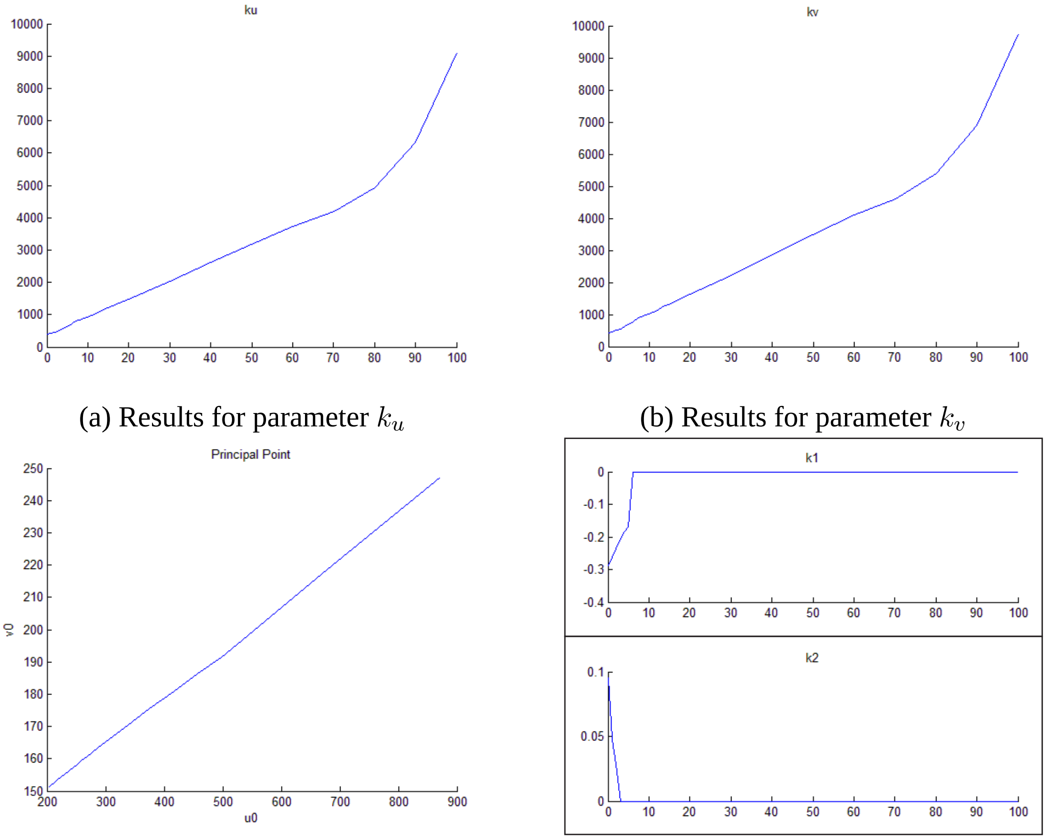


Figure 2.12: Cube mosaic created using our calibration method’s estimated parameters.



(a) Results for parameter k_u

(b) Results for parameter k_v

(c) Principal point estimation

(d) k_1 and k_2 estimates

Figure 2.13: Intrinsic (a), (b) and (c) and radial distortion (d) parameters estimations with our calibration method for real data, for an increasing zoom sequence.

Chapter 3

Surveillance Scene Simulator

In this chapter the background uncertainty model is presented followed by the techniques applied to perform event detection. Based on the event detection results achieved, a method to create simulated surveillance scenarios, using real data both for the background and for generating events, is presented. With this method one can generate multiple copies of events detected in an image sequence and project them at multiple locations of the background representation.

3.1 Background Uncertainty Model

The creation of a panoramic background representation comprises the superposition of various images, acquired from different pan and tilt angle orientations. This implies that a single 3D object captured at different pan and tilt poses, despite having the same radiance, is seen as a varying irradiance due to the vignetting. The estimation and correction of the vignetting effect is not performed in this work thus implying a simple background representation. Although, in the following the background model is presented accounting the vignetting effect and its implications on such a representation. A background model is usually represented by the mean value and variance of the irradiance at each background location M, respectively $\mu_{B(M)}$. Without vignetting correction the "gray level" value of a background location will change as the camera rotation changes. The values of the background thus depend not only on image noise but also on the changes due to vignetting in the imaged pixel $V(m)$, which can now be considered a random variable with mean, $\mu_{V(m)}$, and a variance, $\sigma^2_{V(m)}$:

$$B(M) = L(M)V(m) + \eta \tag{3.1}$$

where η is a zero mean noise process and $L(M)$ denotes the radiance that is expected to be observed at the background pixel M.

Taking expected values we get:

$$\begin{cases} \mu_{B(M)} = L(M)\, \mu_{V(m)} \\ \sigma^2_{B(M)} = L^2(m)\, \sigma^2_{V(m)} + \sigma^2_\eta \end{cases} \tag{3.2}$$

where σ^2_η is the noise variance. A correction of the vignetting would allow the decrease of the variance at the superposition.

Considering that the processes of vignetting and vignetting-correction can be characterized by a mean gain, $\mu_{V_c(m)V(m)}$, and a variance of gains, $\sigma^2_{V_c(m)V(m)}$, then we have that the background mean and variance are

$\mu_{B(M)} = L(M)\, \mu_{V_c(m)V(m)}$ and $\sigma^2_{B(M)} = L^2(m)\, \sigma^2_{V_c(m)V(m)} + \sigma^2_\eta$.

As previously stated, the scope of this work did not include vignetting correction, so one has, $V_c(m) = 1$, meaning that the vignetting directly effects on the image, $\mu_{V_c(m)V(m)} = \mu_{V(m)}$ and $\sigma^2_{V_c(m)V(m)} = \sigma^2_{V(m)}$.

3.2 Single Event Detection

The identification of objects appearing in the camera´s field-of-view is of crucial importance in almost every surveillance systems. A common approach to this problem is to perform background subtraction. This is a class of techniques used for segmenting objects of interest from a database set model. There are several approaches used to perform background subtraction such as mixture of gaussians and kernel density estimation (KDE). However, as seen by Piccardi [12], these methods require a lot o memory to perform well. So given the restrictive memory limitations a simple gaussian model was chosen for the task. With this method only three backgrounds, one that accumulates the sum of all the pixels values, $\sum_{i=1}^{N_{uv}} I_{iuv}$, one that stores the sum of the square of the pixels values, $\sum_{i=1}^{N_{uv}} I^2_{iuv}$, and the other that saves the number off times a pixel is seen, N_{uv}, need to be stored in memory.

Event detection is done by comparing the currently captured image, I_{uv} with the corresponding image from the background database, B_{uv}. This comparison is performed by computing the log likelihood function (Eq. 3.3).

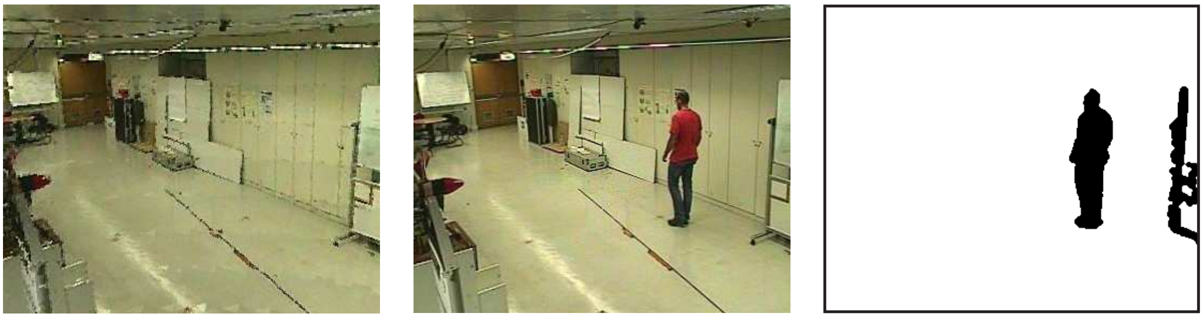
$$L_{uv} = \frac{-0.5(I_{uv} - B_{uv})^2}{\Sigma^2_{uv}} - 0.5\ln(\Sigma^2_{uv}) - 0.5\ln(2\pi) \tag{3.3}$$

where Σ^2_{uv} denotes the background variance.

A first classification of a pixel (u, v) is performed by assessing the log likelihood function values, L_{uv}. A pixel is first considered part of the foreground if the log likelihood, L_{uv} is greater than a threshold in at least two RGB components of the image, I_{uv}. Then pixels forming small objects are excluded as only objects with more than 500 pixels are considered part of the foreground. This later classification is performed to attenuate differences created by the vignetting especially in very noisy, low resolution, background panoramas. The final step is then conducted by applying a mean filter to the foreground image.

As the vignetting effect is not corrected, several objects are classified as foreground objects without actually being active events. To distinguish the real single active event, from the inactive ones, the centroid of each foreground object is computed for every captured image, I_{uv}, and only the object with the minimum centroid variation compared to the previous selected active object (i.e. from the previous image) is selected.

Experiments were carried out to detect a single active event in an image sequence. These experiments enabled the gathering of important data, namely the locations of the foreground pixels from the active event and their corresponding RGB values. The background, the operational and the decision images are respectively presented in figures 3.1 (a), (b) and (c), for one iteration of an image sequence.



(a) Background image (b) Operational image (c) Decision image

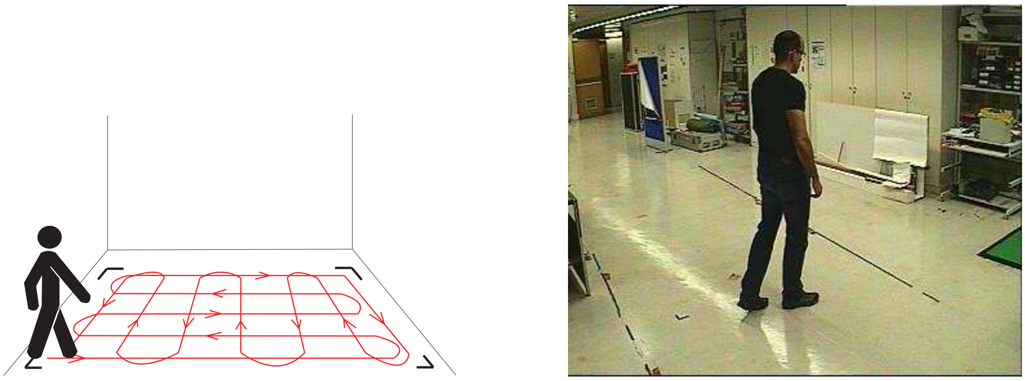
Figure 3.1: Result of the single event detection experiments conducted for one iteration of an active event image sequence.

Figure 3.1(c) shows the detection of a person and of an object that moved between the background acquisition and the detection time. The detected object will be later tagged not relevant since it stays static for a long time, therefore being possible to be reintegrated with the background.

3.3 Simulated Surveillance Scenario

A major difficulty in testing control techniques for surveillance systems, over real data, is the acquisition of image testing sequences. These acquisitions can often be very challenging as difficulties arise when uncontrollable events occur, such as lightning condition changes, human targets not performing in the way they were expected and even limitations presented by the test set, such as very small areas where actual tracking is possible. So, in the following, a method to generate simulated environments, with multiple targets, using only real data is presented. With this method one is able to generate multiple targets in a world representation (e.g. cube based representation) with a minimum of two dataset image acquisitions. One image sequence describes a moving target and the other image sequence describes the background of the entire testing scenario. Figure 3.2(b), shows the moving target, a person.

The feet of the person are used to characterize the person's location and repositioning using 3D rigid transformations.



(a) Target's trajectory pattern

(b) Example of acquired image from target sequence

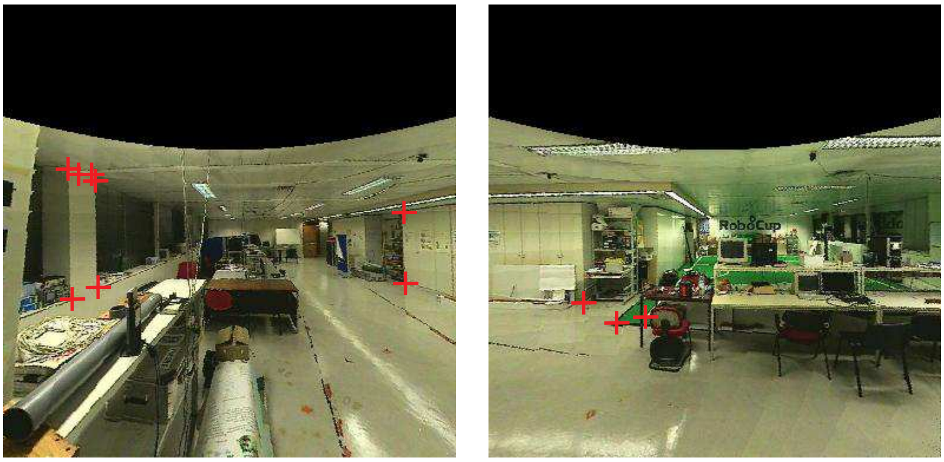
Figure 3.2: Target's trajectory pattern and example of image acquired from the target's moving sequence.

3.3.1 Reconstruction of Target's 3D Trajectory

The background representation and the image sequence with a moving target, allow doing background subtraction and single event detection. The background subtraction complemented with segmentation produce a collection of 2D image points describing a single event along all the images in the sequence. Multiple instances of subsequences of the original sequence can then be projected to the background representation creating a dynamic scenario with several copies of a single person at multiple locations. To achieve this, one needs to be able to characterize the target's 3D trajectory in a world reference frame and map it to the camera frame.

The first step consists therefore in estimating the rigid transformation between the world reference frame and the camera (at home position) frame. More specifically, one must identify 3D world points well distributed in the scenario, so as to allow a good characterization of the scenario in terms of 3D world coordinates. Additionally, at least three points must be taken from the floor plane of the setting. All these points will be completely characterized by their 3D world coordinates (${}^{\mathbf{w}}\mathbf{X}$), their 2D image coordinates and the pan and tilt orientations from which the respective image was acquired. In figure 3.3 (a), the left side face of the cube based representation of this scenario is presented along with the collection of points whose 3D world coordinates have been measured. Additionally, in figure 3.3 (b), the front face of the same background representation is presented with three floor points, whose 3D coordinates have also been measured. These three points are characterized by their 3D world coordinates, their 2D image coordinates and the respective pan and tilt angles.

Assuming that the origins of the world and camera frame are coincident, the mapping of 3D world points from the world reference frame to 3D points in the camera's reference frame is just a simple rotation (${}^{\mathbf{c}}\mathbf{R}_{\mathbf{w}}$):



(a) Points in left cube face (b) Points in front cube face

Figure 3.3: Points gathered from real scenario to calibrate left cube face and to describe the floor plane.

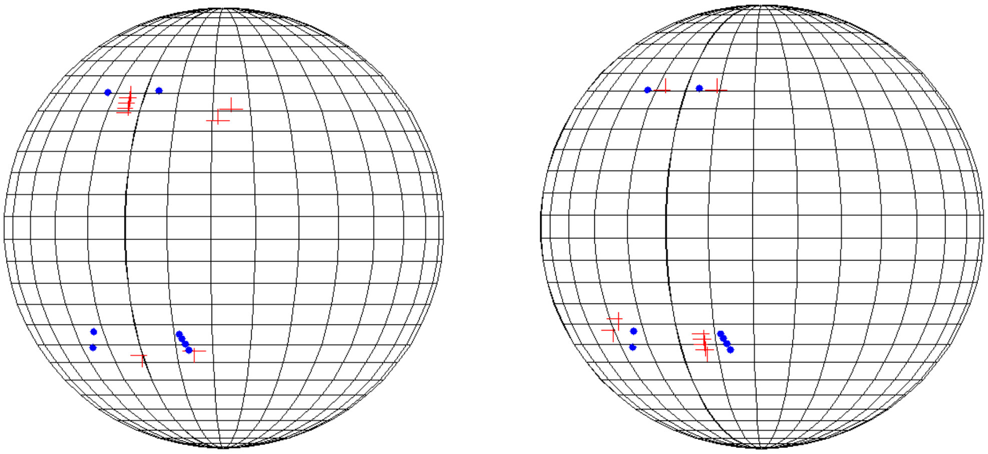
$$ {}^{\mathbf{c}}\mathbf{M} = {}^{\mathbf{c}}\mathbf{R}_{\mathbf{w}} . {}^{\mathbf{w}}\mathbf{M} \tag{3.4} $$

This rotation, ${}^{\mathbf{c}}\mathbf{R}_{\mathbf{w}}$, is estimated using the 3D world coordinates of points shown in figure 3.3 (a) and the corresponding image points back-projected to a unit sphere centered at the camera's location. This estimation is performed by a simple linear transformation i.e. rotation, that best fits points in world coordinate frame, normalized relative to the camera's location, to their corresponding imaged points in the camera's reference frame. In figure 3.4 (a), the 3D world points (${}^{\mathbf{w}}\mathbf{M}$) and the 3D camera points (${}^{\mathbf{c}}\mathbf{M}$) are shown in the unit sphere centered at the camera's world location, before the application of the linear transformation, ${}^{\mathbf{c}}\mathbf{R}_{\mathbf{w}}$. In figure 3.4 (b) the same points are shown but with the application of the estimated transformation (${}^{\mathbf{c}}\mathbf{R}_{\mathbf{w}}$) to the 3D world points, as in equation 3.4.

Using the target's 2D trajectory, extracted from the image sequence, is now possible to obtain the corresponding 3D trajectory in the camera's reference frame. To perform this computation, it is necessary to have the 2D image point coordinates of the mean point ($\mathbf{m}$) of the target's feet. Each of these points can be back-projected to the floor plane described in the camera's coordinate frame:

$$ {}^{\mathbf{c}}\mathbf{M}_{\mathbf{x}} = b(m) \tag{3.5} $$

where $b(m)$ represents the back-projection of the 2D image points. Given the ${}^{\mathbf{c}}\mathbf{R}_{\mathbf{w}}$ matrix, computed earlier, and the 3D world coordinates of the three points from the scenario floor, one is able to map the three points to the camera's reference frame thus achieving the floor plane pavement description. The target's 3D trajectory is then reconstructed by intersecting each back-projection ray with the floor plane defined by ${}^{\mathbf{c}}\mathbf{M}_{\mathbf{1}}$, ${}^{\mathbf{c}}\mathbf{M}_{\mathbf{2}}$ and ${}^{\mathbf{c}}\mathbf{M}_{\mathbf{3}}$. The 3D intersection point is obtained by solving equation: $\alpha {}^{\mathbf{c}}\mathbf{M}_{\mathbf{x}} = \beta({}^{\mathbf{c}}\mathbf{M}_{\mathbf{2}} - {}^{\mathbf{c}}\mathbf{M}_{\mathbf{1}}) + \gamma({}^{\mathbf{c}}\mathbf{M}_{\mathbf{3}} - {}^{\mathbf{c}}\mathbf{M}_{\mathbf{1}}) + {}^{\mathbf{c}}\mathbf{M}_{\mathbf{1}}$, meaning that the desired point is given by the correct scaling of its corresponding back-projection ray ($\alpha {}^{\mathbf{c}}\mathbf{M}_{\mathbf{x}}$).



(a) 3D scenario points before ${}^{c}R_{w}$ is applied (b) 3D scenario points after ${}^{c}R_{w}$ is applied

Figure 3.4: Representation of corresponding 3D points in world and camera's coordinates. The 3D world points are represented as crosses and the corresponding points in the camera´s reference frame are shown as blue dots.

A reconstructed target's 3D trajectory is shown in figure 3.5.

3.3.2 Multi-Event Scenario Creation

The construction of a scenario with multiple active events, at different locations in the background, has several problems associated. In fact, the construction of such a scenario, specially if the events represent real people, must emulate a real setting with real distinct active events, so some details must be taken into account, such as: projecting events only on areas that could in reality be occupied by a person, the image points representing the person must be projected in such a way the person does not get distorted, and finally, the flow created by the projection of an active event during several consecutive time instants must be coherent with the flow of a person moving (i.e. the generated person can not appear to float in the air).

These problems are addressed by following the procedure shown in figure 3.6. The first step is to convert the 2D image data coordinates of the event to 3D coordinates. This is done by back-projecting the set of 2D points, obtained through single event detection, to a plane tangent to a cylinder orthogonal to the ground plane, with radius equal to the distance between the target and the camera, and centered at the camera's position. The cylinder radius is computed resorting to the corresponding event's 3D trajectory points computed in section 3.3.1. As each collection of 2D points, describing an event at a specific time instant, has a 3D trajectory point associated, it is possible to determine the cylinder radius by simply computing the distance from this 3D trajectory point to the projection of the camera's position in the floor plane. By projecting the 2D image points characterizing an event to the plane tangent to this cylinder, the generated person will always appear undistorted in the created scenario. Finally some transformations need to be done, to the 3D points obtained, in order to generate the event in the desired

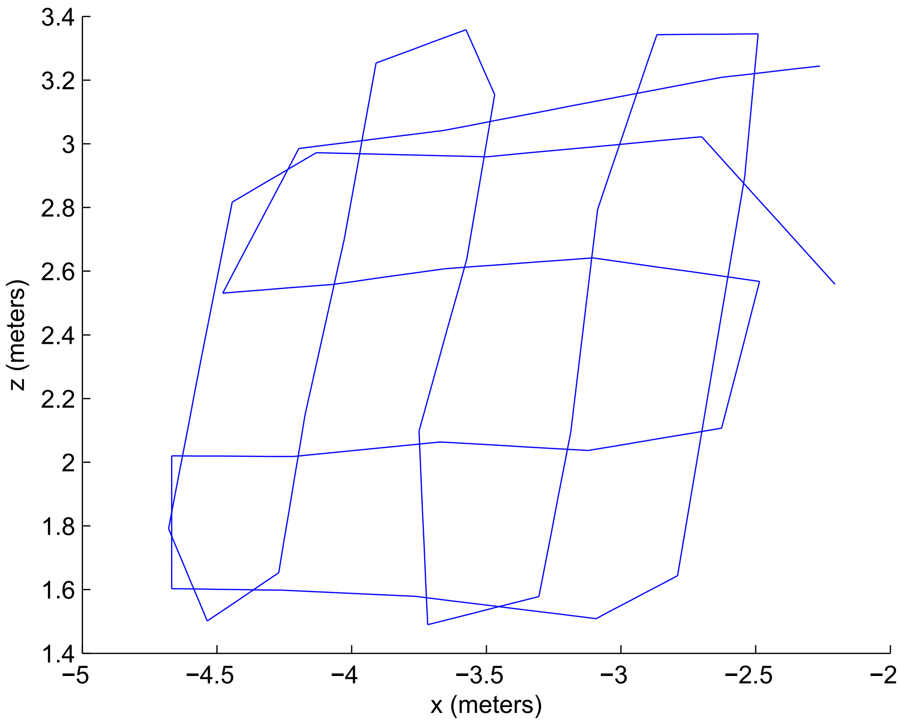


Figure 3.5: Reconstructed target's 3D trajectory from an image sequence with a person moving in a pattern similar to the one presented in figure 3.2 (a).

location in the cube based background representation. This is achieved by applying a rotation, around the y axis ($\Delta\theta$), and a translation along the z axis (Δd), to the 3D points projected in the place.

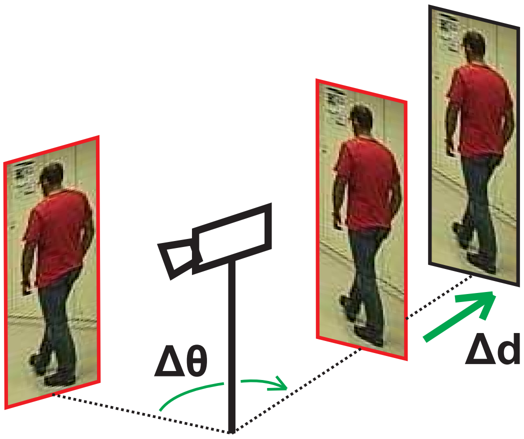


Figure 3.6: Transformations produced to generate events in different locations of the cube based background representation.

In the experiments conducted, a scenario was generated resorting only to one active event image sequence lasting 140 seconds, from which subsets were projected in the scenario thus creating an active simulated scenario with duration measurable in time units ($1\ t.u. = 0.60\ seconds$). In figure 3.7, a panorama obtained from time unit 8 of a, 115 time unit (69 second), action scenario is presented. In this scenario a total of 26 targets are displayed throughout its duration, from which a maximum of five appear in the same time instant. These events are projected from -20^o to 90^o pan, relative to the camera's home location, and as far as 2.5 meters from the camera relative to their original position i.e. projection with no transformations.

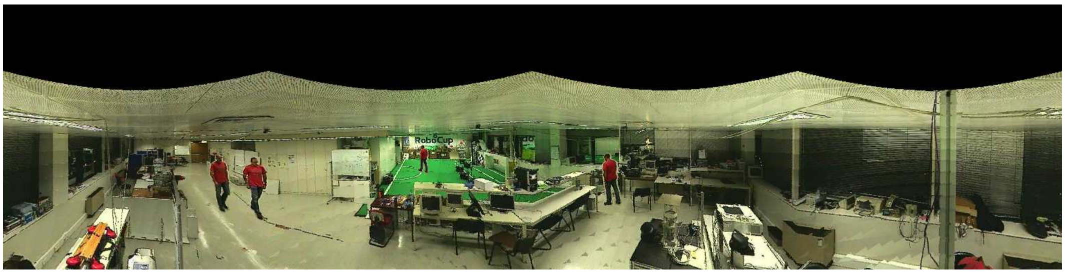


Figure 3.7: Panorama at iteration 8 of the created simulated scenario.

Chapter 4

Surveillance Methodologies

Pan-tilt-zoom cameras provide the flexibility of selecting the desired field-of-view. This flexibility is lost if one can not successfully design effective pan and tilt controllers whose application directly impacts the surveillance performance. In this chapter several surveillance control methodologies are presented and tested. The experiments conducted produce results that are evaluated resorting to known, documented metrics [11].

There are several methodologies used to control a pan-tilt-zoom camera. In this work four control methods are implemented: Random Search (RaS), Rotation Search (RoS), Lock and Random Search (LRS) and Timed Lock and Random Search (TLRS). The first two are open loop algorithms and the later two are closed loop algorithms, i.e. the camera´s orientation control depends on past segmentation results. The Random Search method is implemented by assigning random orientations to the pan-tilt camera. These angles are generated from a uniform distribution between well defined pan and tilt limits and so, this method, requires a camera with high operational speeds as it can jump anywhere at any time. In the Rotation Search, the tilt angle is kept constant while the camera is rotated by changing the pan angle with a constant step. This method systematically searches the scene by a rotation process similar to a RADAR. Both these two open loop control methods have a limited performance as they do not try to keep a detected target in the camera's FOV. The Lock and Random Search is a closed loop method as it uses segmentation information to try to keep the detected target in the camera's FOV. In this algorithm the camera randomly searches for targets, operating similar to the Random Search, and when it detects one event it locks it and tracks it by commanding the camera to an orientation that keeps the target close to the center of the image. This is achieved by computing the centroid of the detected target and using this, 2D coordinate, information to determine new pan and tilt angle orientations that can move it to the center of the next image. This is achieved through inverse kinematics, as described in section 2.3. In TRLS, targets are tracked using the same mechanisms introduced for the LRS. However, there is a limited tracking of a target in the sense that the tracking is suspended by the camera at given instants in time. This is done in order for the camera to search for new targets using the RaS method. While the new search takes place the suspended trackings are predicted resorting to Kalman filtering.

4.1 Open-Loop Methods

From the four methods tested, two correspond to open loop methods, Random Search and Rotation Search. They are classified as open loop methods as the pan and tilt angles are determined disregarding any information relative to previous detection phases. This means the camera moves without any influence from the segmentation results.

4.1.1 Random Search (RaS)

The Random Search algorithm performs random control on the pan-tilt camera (Figure 4.1). This means the pan and tilt angles are generated from a uniform distribution and thus, the camera moves randomly trying to find events. This makes it the simplest method possible to control a pan-tilt camera. If one event is found in iteration i the information regarding this detection is not used in the next iteration, $i + 1$. In fact, in iteration $i + 1$ the pan and tilt angles are generated independently from the ones that produced the detection in the previous iteration. A random uniform distribution was used to generate the angles, to guarantee all orientations have equal probability of being chosen.

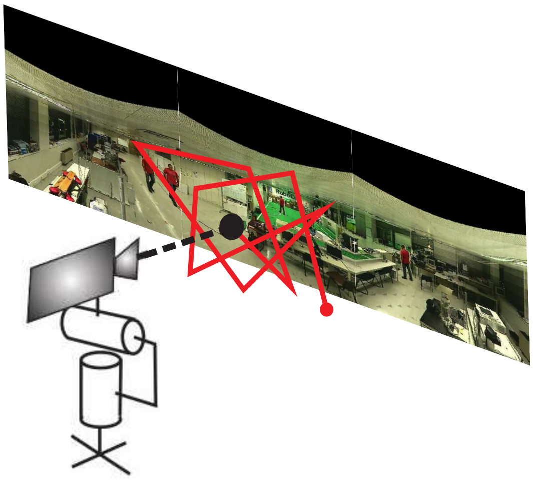


Figure 4.1: Random Search (RaS) detection method. The pan and tilt angles are randomly generated from a uniform distribution and are completely uncorrelated with previous information about the targets.

4.1.2 Rotation Search (RoS)

In the Rotation Search, as in the Random Search (Section 4.1.1), a detection in iteration i does not influence the control for iteration $i + 1$. The main difference between these two open loop control methods is that the Rotation Search uses a pattern control motion. This motion emulates the movement of an active radar as the tilt angle is kept constant and the pan angle is iteratively increased at a constant step thus creating the patterned rotation control motion (Figure 4.2). The mindset behind this method is that the camera can be installed in a way that events occur in a well defined pan range thus discarding vertical motion.

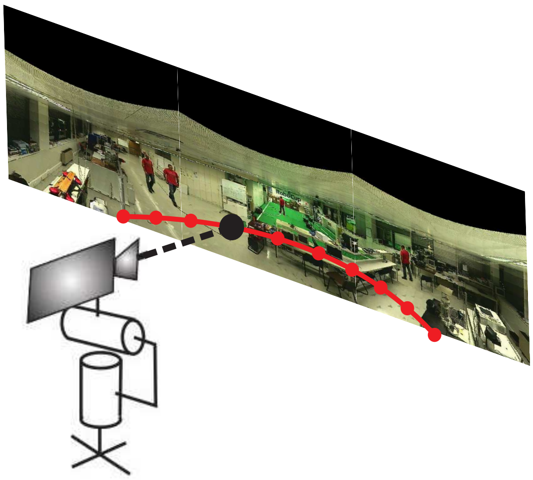


Figure 4.2: Rotation Search (RoS) detection method. The tilt angle is kept constant as the pan angle is continuously increased producing a rotation motion of the camera.

4.2 Closed-Loop Methods

Unlike open loop control methods, closed loop control methods resort to detection information from previous iterations to control the camera's pan and tilt angles. This control consists in computing the correct pan and tilt angles so that the event detected in iteration i is centered, as best as possible, in the image acquired at iteration $i+1$. In fact, this centering is not perfect as the target moves between iterations and thus, the best one can expect is an approximation to the desired outcome. Two approaches commonly used to perform this object centering are image-based look-and-move and image-based virtual-servo [9]. In this work we have a pan-tilt camera with internal odometry control-loop and in addition we have a complete description of the camera's intrinsic parameters as full calibration is done before calibration. Concluding, we use the image-based look-and-move method.

4.2.1 Lock and Random Search (LRS)

In the Lock and Random Search method when a target is found in a frame, the pan and tilt angles are updated so the object is centered in the next frame. The method is based on the principle that if an object whose centroid, $[u\ v]^T$, is detected at time instant t, then there is a high probability that the same object will appear at a neighborhood of that position, $[u + \epsilon_u\ v + \epsilon_v]^T$, at time instant $t + \delta t$, provided δt is small enough. This method thus comprises two types of control as described next.

1. If no event is found in the current frame, object not locked, the pan and tilt angles are generated

randomly, similarly to the method described in section 4.1.1. So it acts as an open loop control method, more precisely RaS, until some detection is made and so, this comprises the searching phase of the method.

2. If an event is found in the current frame, the object is locked, i.e. the next pan and tilt angles will be computed so the detected event is moved to the center of the next frame. This is achieved by computing the centroid of the object and using its back-projection to determine the necessary pan and tilt angle increments necessary, using equations 2.17 and 2.18 from section 2.3. The process here described consists of the tracking phase of this surveillance control methodology.

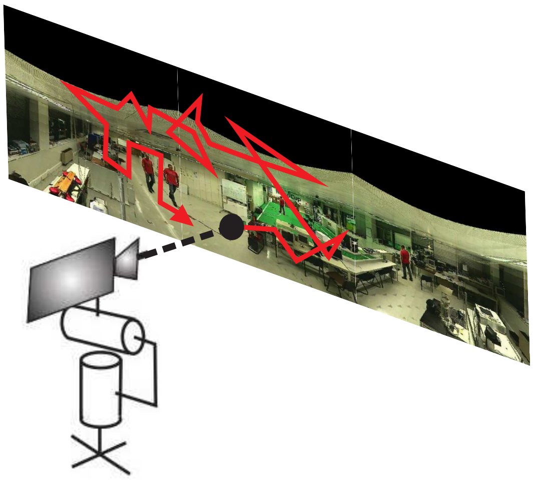


Figure 4.3: Lock and Random Search (LRS) and Timed Lock and Random Search (TLRS) methods. The pan and tilt angles are constantly updated to enable the tracking of a target but, as soon as the tracking is lost, the angles are generated randomly.

However situations can arise that create conflicts in the two processes described. More precisely, (i) if the method is searching for targets and at some point in time detects more than one, and (ii) if while tracking a specific object another one appears in the current FOV. The first issue is addressed by choosing the object with the largest area, i.e. greater number of pixels. This situation only arises when none of the objects is being tracked as if one is currently being tracked and another, whose area could be larger, appears in the camera's FOV, the conflicting issue is, in fact, the second one identified. This is resolved by maintaining the current object locked. This way the camera is aware, and accounts, for the other targets in its current FOV, but never loses track on the original object until it leaves the scenario.

4.2.2 Timed Lock and Random Search (TLRS)

The mindset behind the Timed Lock and Random Search method is that it is not necessary for an object to be in the camera's FOV for the camera to be aware of its presence. More precisely, if some active event is detected by the camera during a specific time interval, it is possible to *learn* its movement and so, all its future actions can be estimated allowing the camera to search for other events while keeping the knowledge of the original one. The TLRS has some similarities to the LRS method as, in fact, it uses the method of centering a detected object in the frame to perform tracking and randomly generated pan and tilt angles to perform a search for new targets (Figure 4.3). Despite the similarities, there are major differences in how these control tools are used. In TLRS when a target is detected it is only tracked for some few time units. After that, the camera releases the tracking and starts to randomly search for new events. While doing this new search, new trackings can be started, and the positions of the already stated trackings are estimated for the subsequent time instants. This estimation is based on the Kalman Filter prediction process as described later in this section. When the predictions reach a certain threshold variance, the controller switches back to tracking the original objects. When the switching back, two situation can arise: the target is still in the scenario and thus the camera tracks it for some more (e.g. two) time units or, the target is not there anymore. This last situation may be associated to false detection errors as the target may have disappeared from the scenario while predictions of its state were still being done.

The prediction of multiple targets locations is done through implementation of Kalman filtering. In this work, it is applied by the TLRS method whenever track on any given target is released by the camera, making it necessary to predict its next movements.

The Kalman Filter comprises two different phases: predict and update. In this work, it is applied based on the last two known positions of a target to estimate its future trajectory. This prediction process is performed until a measure of the quality of the estimation, based on the covariance, reaches a certain threshold. When the prediction process is initialized there are two real measurements available: the last known, normalized, positions of the target in 3D coordinates of the camera frame, $\mathbf{M_{-1}} = [X_{-1}\ Y_{-1}\ Z_{-1}]^T$ and $\mathbf{M_o} = [X_o\ Y_o\ Z_o]^T$ and thus it is possible to compute the displacement between them, $d\mathbf{M_o} = \mathbf{M_o} - \mathbf{M_{-1}}$.

$$\hat{\mathbf{M}}_i = \hat{\mathbf{M}}_{i-1} + t.d\tilde{\mathbf{M}}_{i-1} \tag{4.1}$$

where $\hat{\mathbf{M}}_i$ is the current state estimate, $\hat{\mathbf{M}}_{i-1}$ is the last state estimation (when $i = 1$ we set $\hat{\mathbf{M}}_{i-1}$ equal to the first observation), t is the discrete time interval between states (in our case t is always equal to one) and $d\tilde{\mathbf{M}}_i$ is the normal distribution of the displacement ($d\tilde{\mathbf{M}}_i \sim \mathcal{N}(d\mathbf{M_o}, \sigma_o)$), i.e. speed, of the specific target, between time instants $i-1$ and i.

Each predicting step can thus be described as $\hat{\mathbf{M}}_i = \mathbf{F}.\hat{\mathbf{M}}_{i-1} + \mathbf{G}.d\tilde{\mathbf{M}}_{i-1}$, where $\mathbf{F}$ is a 3x3 identity matrix and $\mathbf{G} = [1\ 1\ 1]^T$.

The quality of this estimation is determined by computing the covariance between the current state estimate and the last observed state $\Sigma(\mathbf{M_o}, \hat{\mathbf{M}}_i) = \begin{bmatrix} \sigma^2 & \sigma.\hat{\sigma} \\ \hat{\sigma}.\sigma & \hat{\sigma}^2 \end{bmatrix}$.

In fact, the covariance matrix is a generalization of the concept of variance and so it is possible to retrieve the variance of the estimate, $\hat{\sigma}^2$, which is the real measure of its quality. It is when $\hat{\sigma}^2$ exceeds a certain threshold, T, that the estimation process is terminated, culminating with the camera trying to track the target again. This threshold is defined according to the three-sigma rule [13], i.e. $T = 3\sigma_o$, meaning an estimate is considered bad if its variance exceeds three times the value of the standard deviation of the normal distribution of the displacement, i.e. $\hat{\sigma}^2 > 3\sigma_o$

4.3 Performance Metrics

A common metric used to assess the quality of surveillance control modalities is the well known percentage of Correct Detection ($\%CD$)) in a sequence of N images:

$$\%CD = 100 \times \frac{\sum_{i=1}^{N} CD(I_i)}{\sum_{i=1}^{N} CT(I_i)} \tag{4.2}$$

where $CD(I_i)$ denotes the number of correct detections in image I_i and $CT(I_i)$ is the ground truth number of objects in the image.

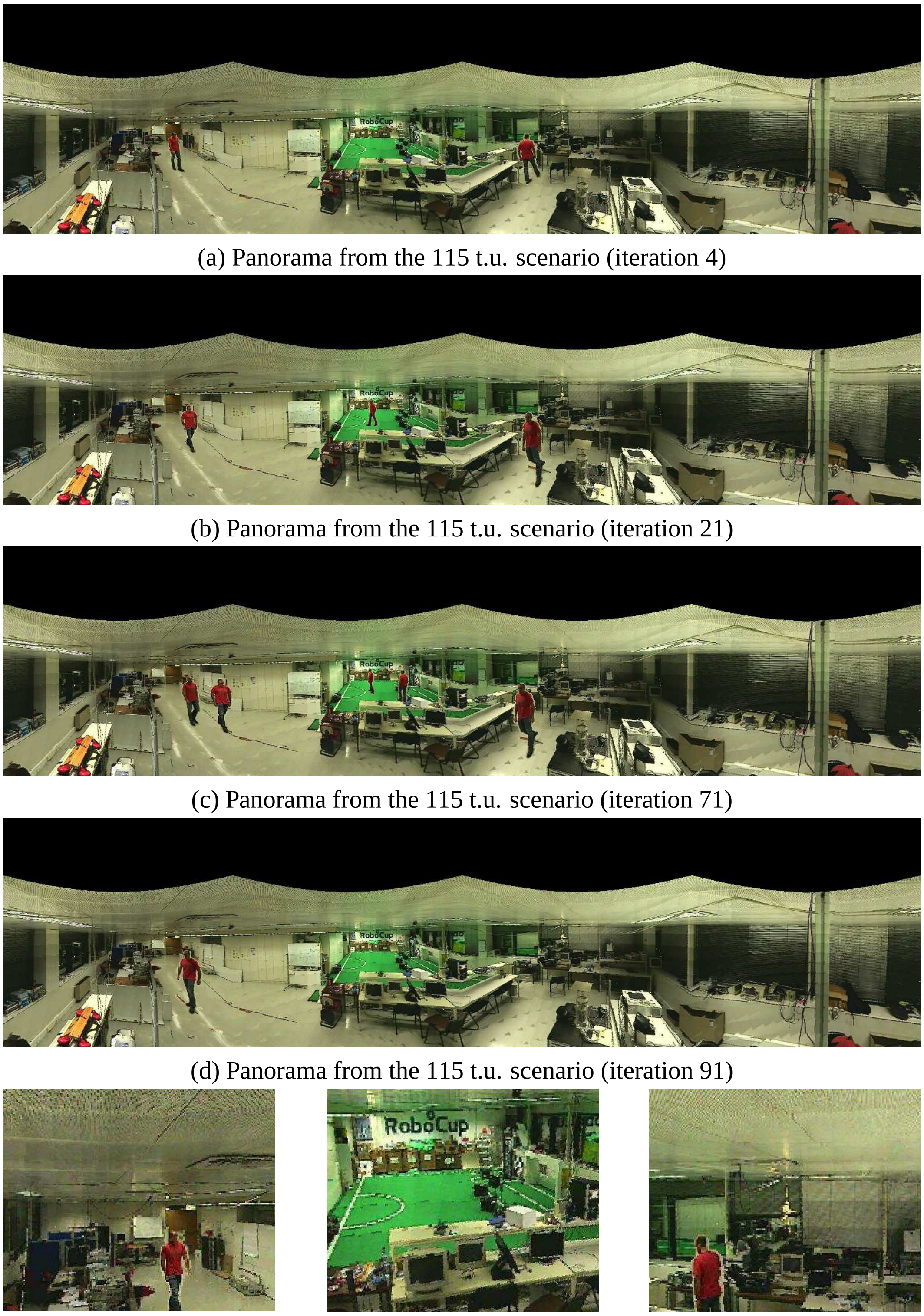
This metric does not provide however the notion of awareness [15]. In fact it does not take into account that there may be several objects, in the complete pan-tilt camera's FOV, that might not be in the field of view of image I_i. So to better describe surveillance methods tested we adopted the percentage of Events Found ($\%EF$) metric:

$$\%EF = 100 \times \frac{\sum_{i=1}^{N} CD(I_i)}{\sum_{i=1}^{N} CT(I_i) + \sum_{i=1}^{N} CT(\bar{I}_i)} \tag{4.3}$$

This metric is similar to $\%CD$ (Eq. 4.2) except in an extra factor added to its denominator. This factor accounts for objects in the pan-tilt camera's complete FOV that are not detected in image I_i. With this statistical characterization it is possible to determine how effective a method is in its awarness of the number moving objects in the complete surveyed scene.

4.4 Experiments and Results

As described in section 3.3 simulated surveillance scenarios were created using real acquired data. For the purpose of testing the four surveillance methods presented two such scenarios were generated. The first one encompassed a total of 13 different active events showcased in a dynamic scene lasting 86 time units, and a second, where 26 events appear in an action scenario lasting 115 time units. Examples of panoramas generated in one of the two scenarios, as well as images acquired from them, by the camera, are presented in figure 4.4. The following two experiences are presented in order to assess the method's tracking and event awareness capabilities.



(a) Panorama from the 115 t.u. scenario (iteration 4)

(b) Panorama from the 115 t.u. scenario (iteration 21)

(c) Panorama from the 115 t.u. scenario (iteration 71)

(d) Panorama from the 115 t.u. scenario (iteration 91)

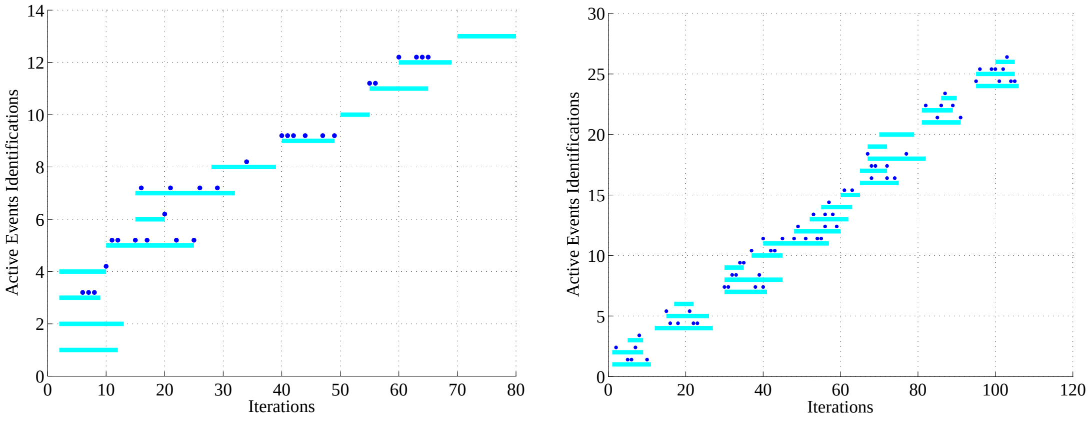
(e) Examples of images acquired with horizontal field-of-view equal to 51.6^o

Figure 4.4: Examples of panoramas obtained from a surveillance simulated scenario and images acquired from those panoramas with a camera with a horizontal field-of-view of 51.6^o.

4.4.1 Experiment 1

In experiment 1, the four methods are applied to the two simulated scenarios while the camera's horizontal field-of-view is kept constant at 51.6^o. Results are shown, for each method, to assess the detections made versus the real presence of a given event. To represent this, every target has a numerical tag and its real presence (i.e. ground truth) is displayed in a cyan line from the time instant it enters the scene to the time it leaves it. Every detection made is shown as a blue point in the respective time instant, and associated to the correspondent event identification tag. After presenting these results the four methods are compared resorting to the percentage of Events Found ($\%EF$), computed as in section 4.3.

Figure 4.5 shows that the Random Search method is acceptable for target detection but poor to track a specified target. The information it retrieves for a detected event is minimal, consisting often in its location just for one single specific time instant.



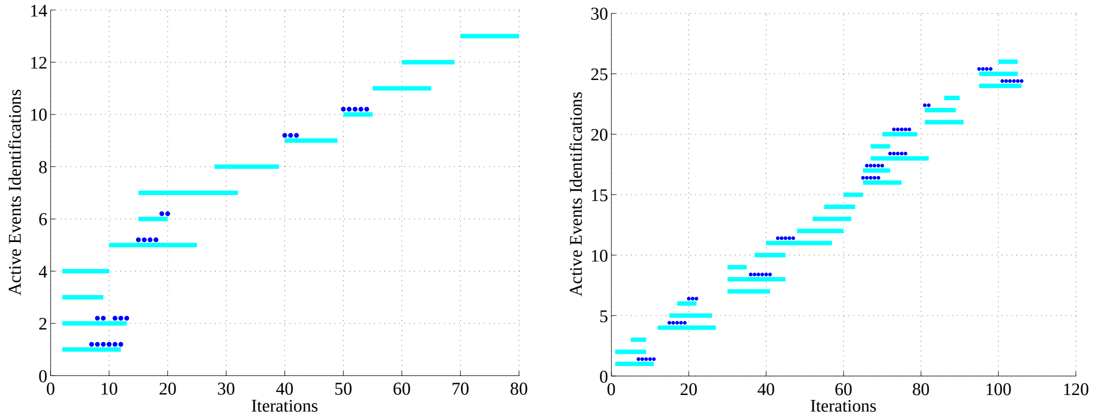
(a) Results for the 86 t.u. scene

(b) Results for the 115 t.u. scene

Figure 4.5: Detection results obtained with the Random Search method for the two generated scenes. Cyan lines indicate the time that a target is visible in the complete field of view of the pan-tilt camera, the blue dots indicate targets detected by the camera.

In Rotation Search (Figure 4.6), because the camera is controlled in a well defined pattern, it is possible to identify one target for multiple, consecutive, time instants but no real tracking is actually being performed. The RoS method becomes appropriate in a scenario where there is a well defined range for events to appear far from the camera.

The Lock and Search Search method is the first closed-loop control method tested and its main strength is its capacity of following a specific target throughout the scenario, acquiring often all its information, i.e. describing its entire trajectory. This feature is well shown in figures 4.7 where several targets are tracked from the instant they appear to the instant they leave the scenario. The fact that one is able to lock a specific target and follow it wherever it goes in the camera's FOV is a strong feature if the surveillance system requires this type of tracking. For instance, in some applications, such as surveillance systems in banks, it may be preferable to always keep track of certain objects perceived as

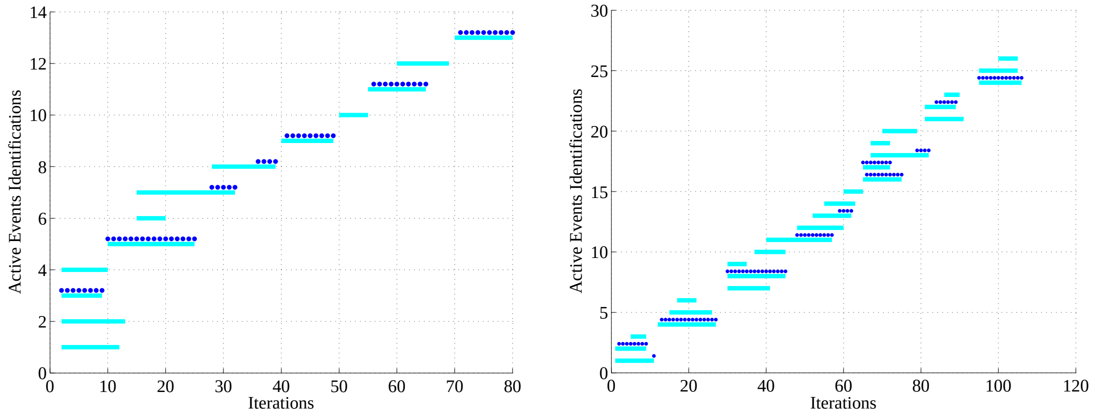


(a) Results for the 86 t.u. scene

(b) Results for the 115 t.u. scene

Figure 4.6: Detection results obtained with the Rotation Search method for the two generated scenes.

dangerous or suspicious. This method can thus efficiently track an event, however, as there is an absolute lock on a target, others appearing in different areas of the scenario are completely disregarded and their presence becomes unknown to the surveillance system.



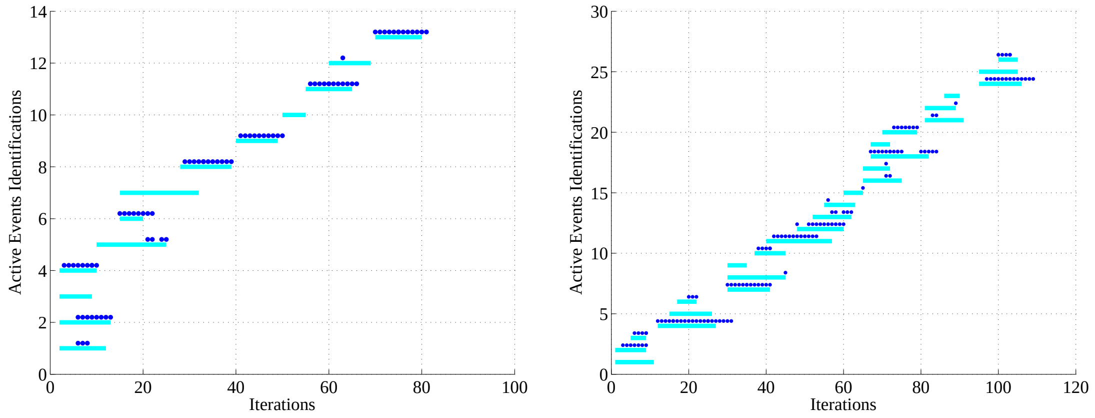
(a) Results for the 86 t.u. scene

(b) Results for the 115 t.u. scene

Figure 4.7: Detection results obtained with the Lock and Random Search method for the two generated scenes.

In the TLRS method a target is tracked, with its trajectory being well characterized in most of its presence in the scenario. However, and in contrast to the Lock and Random Search there is no permanent lock on a target as there is a trade off between tracking and event presence awareness. So, and as shown in figure 4.8, the method achieves a good compromise between these two concepts. Nonetheless, there is a downside to this method as there may be false detections. In fact, and as was earlier explained in section 4.2.2, when the control leaves one target to search for other (unknown) ones it assumes the target it left continues to move in the scenario by continuously predicting its position through application of Kalman filter. At some point the camera moves back to the last target's predicted position to check for it.

However, there is always the possibility that in the time interval the surveillance system was performing the predictions the target had already disappeared from the scene, thus reporting false detections. This situation is well shown in figure 4.8 (b), where the fourth event on the scene is perceived, by the camera, to be present for more 4 time units that it really is, before the control moves the camera back to its last predicted position and discovers it's already gone. TLRS is a method that offers enhanced event presence awareness and, although, there is a downside, as false detections may occur, this method is adequate for various surveillance applications. For instance, in the military surveillance applications, it may be preferable to have some false detections rather than miss-detections, as it is mandatory to have a very good event presence awareness.



(a) Results for the 86 t.u. scene

(b) Results for the 115 t.u. scene

Figure 4.8: Detection results obtained with the Sentinel Search method for the two generated scenes.

In Table 4.1 the percentage of events found ($\%EF$) results obtained for each method in the two simulated scenarios tested are presented. As expected the open loop control methods (RaS and RoS) clearly lose in event presence awareness to the closed loop methods (LRS and TLRS). These results show that the rotation search has the worst results from all the methods. In fact, the best it achieves is to match the detection efficiency of the RaS in the 115 t.u. scenario, as in the other it loses to it by performing approximately 3% less detections. The RoS is prompted to achieve better results if the targets moves along with the rotation of the camera at a patterned movement. As long as the targets move in random positions in the scenario, describing random trajectories, the RaS method achieves better results from the two open loop methods tested.

The two closed loop methods are clearly more efficient in event presence awareness relatively to the open loop methods. This greater efficiency is largely due to the capability of these two methods of using past event information to perform new detections. Between the two, the TLRS achieves the best results in the experiments conducted. The LRS tracks an event and never loses it until it is out of the scenario, as for the TLRS, the lock on an event is released at some point so that new ones may be detected. Although the event is released, in the TLRS method, the event presence awareness is kept by predicting its movement (See section 4.2.2). This capability of maintaining the awareness of an event although it is

EF (%)	86 t.u. scene	115 t.u. scene
RaS	20.89	22.83
RoS	17.19	22.83
LRS	52.43	37.41
TLRS	57.81	45.78

Table 4.1: Percentage of Events Found ($\%EF$) computed for the Random Search, Rotation Search, Lock and Random Search and Sentinel Search methods. Two action scenarios where used and the camera's horizontal FOV was kept at 51.6^o.

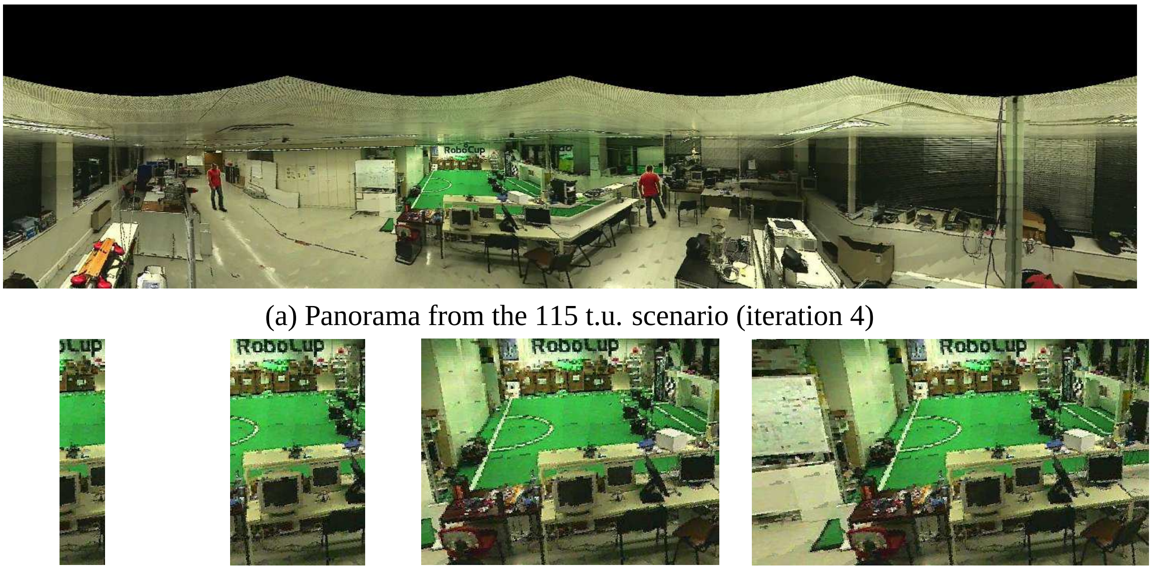
out of sight of the camera gives the TLRS method a clear advantage in event presence awareness as seen in table 4.1. The LRS can, given the right conditions, achieve results closer to the TLRS. This is made possible when, during the experiment's time interval, there is a large concentration of events entering the scenario in relatively close positions. By having an event locked while others enter the scenario close to it, the LRS is capable to have them all in the camera's current FOV, while the TLRS will be jumping to other positions to search for more targets.

4.4.2 Experiment 2

In experiment 2 the four surveillance methods were applied to the 115 t.u. simulated scenario for different camera's horizontal field of views. More precisely, ten experiments were made for each method ranging from a horizontal FOV of 10^o to one of 100^o. In figure 4.9 a scenario panorama is presented along with images acquired, in a specific camera orientation, for four of the different horizontal field of views tested.

To assess the event presence awareness of each method, for increasing horizontal field of views, the percentage of events found ($\%EF$) was computed. These computations allow the description of each detection method in terms of their event presence awareness for varying horizontal field of views. The results obtained are shown in figure 4.10.

The results obtained (figure 4.10) confirm the closed loop control methodology as the most efficient in terms of event presence awareness. In fact, the closed loop methods achieve far better results as they are always ahead RaS and RoS by approximately 10%. The difference is greater when the field-of-view takes intermidiate values ($hFOV \epsilon [40^o, 70^o]$). Although RaS and RoS clearly lose, there is no real indication of which, of these two, achieves best results, they achieve very similar $\%EF$ values. As stated, the closed loop methods achieve far better results than both the RaS and RoS. However, between LRS and TLRS, and contrary to the open loop methodologies, there is a clear separation in event presence awareness efficiency. When the FOV is very narrow, in the order of 10^o to 30^o, they achieve similar results as there is no real advantage taken by the TLRS in searching the scenario for more targets, because the probability of finding events by randomly searching the scene, with a narrow FOV, becomes very small. As soon as the FOV starts to increase, the TLRS improves, becoming clearly ahead of LRS in terms of event presence awareness. This situation is clear in the transition from a field of view of 40^o to one of



(a) Panorama from the 115 t.u. scenario (iteration 4)

(b) hFOV = 10º (c) hFOV = 30º (d) hFOV = 70º (e) hFOV = 100º

Figure 4.9: Example of a panorama from the simulated scenario used in the experiments and three images acquired from the same camera orientation for four different horizontal field of views.

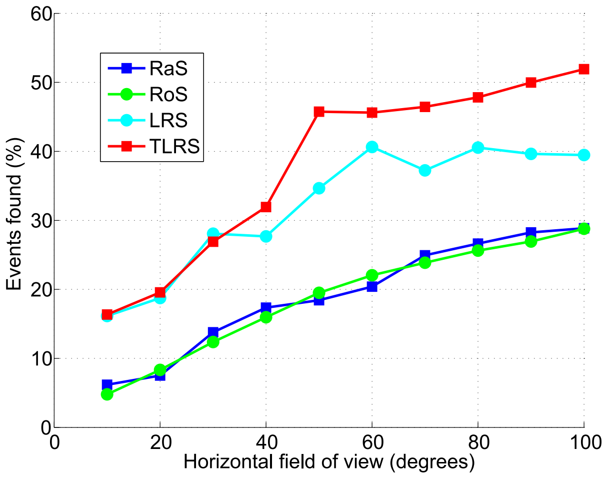


Figure 4.10: $\%EF$ for RaS, RoS, LRS and TLRS methods, for increasing horizontal FOV conditions.

50^o. The difference achieved, for subsequent FOVs, reflects the actions of the predictor. More precisely, as the field of view increases, also increases the probability that, in TLSR, novel events are found right after target tracking is suspended. This situation creates a scenario in which the camera is aware of more targets, without actually having them in its FOV, as multiple events are being just predicted. The potential of TLRS is thus very high, more so, if one realizes the LRS tends to saturate around 40%, while Timed Lock and Random Search continues to increase past 50%.

Chapter 5

Conclusion and Future Work

The work described in this thesis consists of three main subjects: pan-tilt-zoom camera auto-calibration, creation of real data simulated scenarios with multiple active events, and surveillance control methodologies using target tracking. The first subject, PTZ camera auto-calibration, encompassed estimating intrinsic and radial distortion parameters over the camera's full zoom range. This task culminated in the generation of a cube based representation of a $360^o \times 180^o$ scenario captured by the camera. These results allowed then to approach the next subjects namely creating real-data based test-scenarios and designing intelligent PTZ camera controllers capable of tracking active events and, simultaneously, achieving good event presence awareness results.

A well known documented method had already focused in geometric auto-calibration of PTZ cameras [1]. However, it aimed solely to estimate intrinsic parameters, not including radial distortion, and thus, its application produced considerable errors. Another, well known approach to auto-calibrate these cameras, addressing estimation of both intrinsic and radial distortion parameters, was considered [14]. This method only uses a small fraction of a pan-tilt-zoom camera's field of view and so our approach was to create a method that, while based on the same principles allows using complete FOV of the PTZ camera.

The task of designing a surveillance control methodology for a PTZ camera was initially blocked by the difficulty in defining a real data test setting scenario on which to experiment and compare several such approaches. Related work suggested using synthesized abstract scenarios [15] as, at the same time, two major difficulties were avoided: complex real data acquisitions became unnecessary as well as camera geometric calibration. In this work, however, the approach was to use only real data, so a methodology, to create a real-data based simulated scenario with multiple events, was proposed. The proposed methodology produced highly flexible simulated scenarios with multiple events, through simple data acquisitions, and, by supporting it on a cube based representation, created the conditions to manipulate camera characteristics as desired.

With the establishment of a base setting on where to test surveillance methodologies, the focus of this work turned to the creation of a PTZ camera controller that, while encompassing target tracking, explored human characteristics to achieve an efficient automatic surveillance system. In [15], several control methodologies were already addressed, however, their application on our testing scenarios was found to be inefficient as some were too simple and others too rigid in their concept. In this work a surveillance methodology that explored memory of past events was proposed. This approach achieved significantly better results than the documented methods tested.

Future work on the problems addressed will focus mainly on the design and implementation of new automatic surveillance methodologies, for PTZ cameras, with efficient application on real world scenarios. The mindset is to adapt known concepts, largely adopted by humans in their day-to-day life, and enhance them resorting to the vast memory resources currently available.

Appendix A

Image Sequences for Camera Calibration

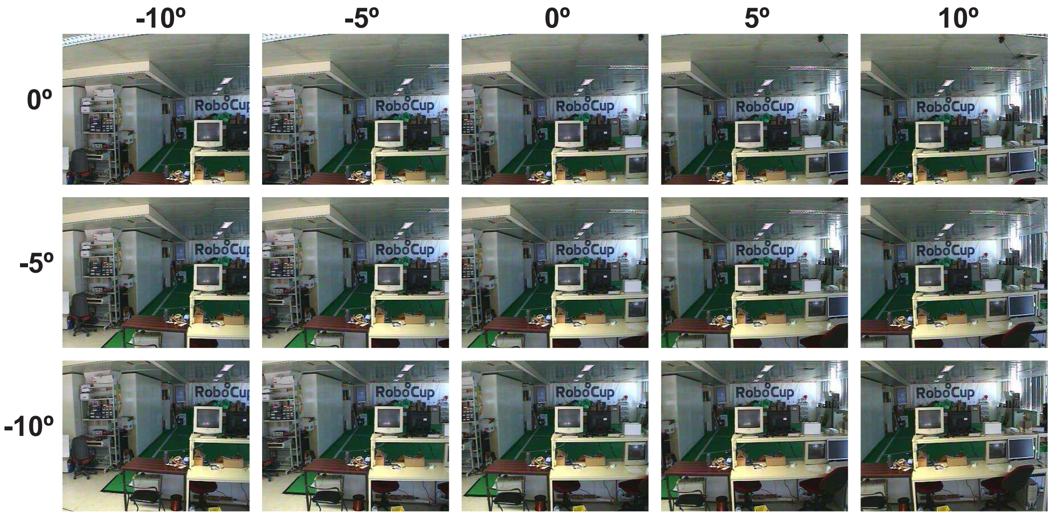


Figure A.1: narrow field-of-view calibration grid ($pan \ \epsilon \ [-10^o, 10^o]$ and $tilt \ \epsilon \ [-10^o, 10^o]$) used for minimum zoom calibration with real data.

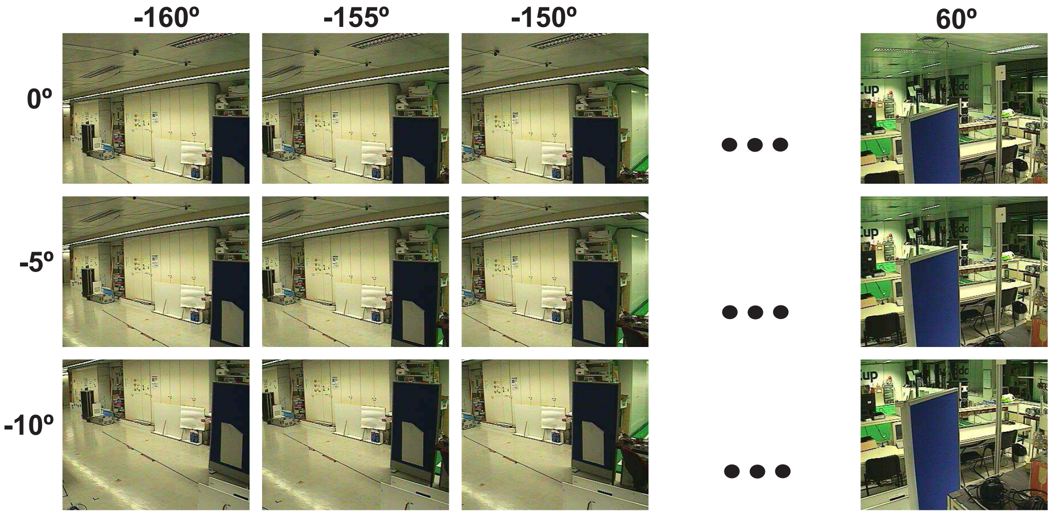


Figure A.2: wide field-of-view calibration grid ($pan \in [-160^o, 60^o]$ and $tilt \in [-10^o, 10^o]$) used for minimum zoom calibration with real data

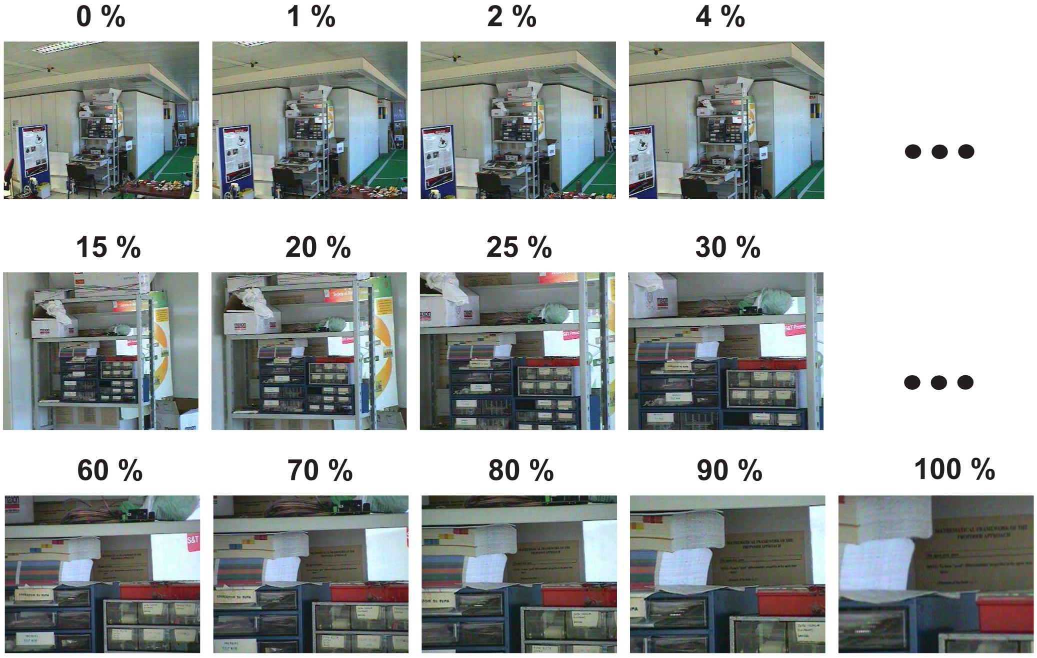


Figure A.3: calibration grid of images used for real data full zoom range camera calibration.

Bibliography

[1] L. Agapito, R. Hartley, and E. Hayman. Linear calibration of a rotating and zooming camera. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pages 15–21, 1999.

[2] L. De Agapito, E. Hayman, and I. Reid. Self-calibration of a rotating camera with varying intrinsic parameters. In *Proc 9th British Machine Vision Conf, Southampton*, pages 105–114, 1998.

[3] Jean-Yves Bouguet. Camera calibration toolbox for matlab. `http://www.vision.caltech.edu/bouguetj`.

[4] Ricardo Galego, Alexandre Bernardino, and José Gaspar. Surveillance with pan-tilt cameras: Background modeling. In *RECPAD 2010. 16^{th} Portuguese Conference on Pattern Recognition*, Oct. 2010.

[5] Ricardo Galego, Alexandre Bernardino, and José Gaspar. Vignetting correction for pan-tilt surveillance cameras. In *VISAPP 2011. International Conference on Computer Vision Theory and Applications*, Mar. 2011.

[6] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[7] Richard I. Hartley. Self-calibration of stationary cameras. *International Journal of Computer Science*, 22(1):5–23, February 1997.

[8] Janne Heikkila and Olli Silven. A four-step camera calibration procedure with implicit image correction. In *1997 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'97)*, 1997.

[9] S. Hutchinson, G. Hager, and P. Corke. A tutorial on visual servo control. *IEEE Transactions on Robotics and Automation*, 12(5):651–670, 1996.

[10] Diogo Leite, Alexandre Bernardino, and José Gaspar. Auto-calibration of pan-tilt-zoom cameras: Estimating intrinsic and radial distortion parameters. In *17th edition of the Portuguese Conference on Pattern Recognition (RecPad 2011)*, 2011.

[11] Jacinto C. Nascimento and Jorge S. Marques. Performance evaluation of object detection algorithms for video surveillance. *IEEE Transactions on Multimedia*, 8(4):761–774, 2006.

[12] M. Piccardi. Background subtraction techniques: a review. In *Systems, Man and Cybernetics, 2004 IEEE International Conference on*, volume 4, pages 3099 – 3104 vol.4, Oct. 2004.

[13] Friedrich Pukelsheim. The three sigma rule. *The American Statistician*, 48(2):88–91, May 1994.

[14] Sudipta N. Sinha and Marc Pollefeys. Pan-tilt-zoom camera calibration and high-resolution mosaic generation. *Comput. Vis. Image Underst.*, 103(3):170–183, 2006.

[15] D. Vicente, J. Nascimento, and J. Gaspar. Assessing control modalities designed for pan-tilt surveillance cameras. In *In 15th Portuguese Conference on Pattern Recognition (RecPad 2009)*, October 2009.

[16] Eric W. Weisstein. Levengerg-marquardt method. `http://mathworld.wolfram.com/Levenberg-MarquardtMethod.html`.