Semi-autonomous 3D tracking

Juan Andrade

Facultad de Ingeniería, Universidad de Cuenca, Cuenca, Ecuador

Corresponding author: juan.andrade@ucuenca.edu.ec

Fecha de recepción: 27 de junio 2012 - Fecha de aceptación: 30 de julio 2012

ABSTRACT

A 3D tracking system that works with a minimum of two cameras has been implemented. The proposed system consists of two main processes: a calibration process followed by a 3D tracking one. The calibration process is done only when the system is installed; but, should be repeated if camera parameters, either internal or external, are changed. Internal calibration was conducted based on the cameras' final locations; therefore, internal parameters include operating conditions. The adopted Wide Baseline Matching (WBM) scheme provides feature descriptors with high distinctiveness. Matching is achieved by using a voting algorithm based on a similarity transform and the robust Random Sample Consensus (RANSAC) statistical method that enforces the epipolar constraints. The implemented WBM procedure provides feature correspondences between the image planes of the two cameras used for the external calibration. The 3D tracking process corresponds to the normal operation of the system after the calibration process. The proposed 3D tracking scheme which combines 2D tracking data from each camera is based on a triangulation method and the determined internal and external camera calibration parameters.

Keywords: Tracking, epipolar, wide baseline matching, triangulation, camera calibration.

RESUMEN

Un sistema de seguimiento tridimensional (3D) que funciona con un mínimo de dos cámaras ha sido implementado. El sistema propuesto consiste de dos procesos principales; un proceso de calibración seguido de uno de seguimiento 3D. El proceso de calibración es ejecutado cuando el sistema es inicializado; pero, debe ser repetido si los parámetros de las cámaras, ya sea internos o externos, varían. La calibración interna se realiza en las ubicaciones finales de las cámaras por lo que los parámetros internos incluyen las condiciones de operación. El esquema de coincidencia de línea base amplia (WBM) adoptado proporciona descriptores con alta distinguibilidad, las coincidencias se determinan mediante el uso de un algoritmo de voto basado en la transformada de similaridad y RANSAC, que es un método estadístico robusto el cual se encarga de hacer cumplir las condiciones epipolares. Las correspondencias encontradas entre imágenes de dos cámaras mediante el procedimiento WBM son utilizadas para la calibración externa. El proceso de seguimiento 3D corresponde a la operación normal del sistema luego del proceso de calibración. El esquema de seguimiento 3D propuesto, el cual combina la información de seguimiento 2D de cada una de las cámaras, se basa en un método de triangulación que emplea los parámetros internos y externos de la calibración de las cámaras.

Palabras clave: Seguimiento, epipolar, coincidencias de línea base amplia, triangulación, calibración de cámara.

1. INTRODUCTION

Most of the times, video surveillance systems are formed by multiple sensors that can also have overlapping surveillance volumes among them. These common volumes can represent a waste of resources in a basic surveillance system, or the redundant information can be exploited to gain three-dimensional (3D) information from multiple views.

Of course, additional information about the optical system of the video sensors, as well as the relative positions and rotation of the sensors in the 3D space is required, information known as internal and external parameters (Hartley and Zisserman, 2003). Camera calibration consists in estimating the internal and external parameters of the cameras. This task becomes more difficult in outdoors systems where the use of specialized patterns for the camera calibration (Bouguet, 2004) is not a practical approach. In Lee *et al.* (2000) internal calibration is avoided by taking the internal parameters from the camera manufacturer's specifications, and the external calibration is solved by using homographies to align each camera view to a ground plane; a minimum of three cameras are required for this method. Collins *et al.* (2000) used a set of PTZ cameras of which the internal parameters are estimated by rotating and zooming the cameras, and the external parameters are defined using a previously measured set of landmarks. 3D tracking can also be achieved with one camera (Collins *et al.*, 2000); but in this case a 3D model of the terrain is required.

Three-D tracking of objects that are in the field of view on 2 cameras can be obtained using the 2D tracking information of the objects in each of the cameras. Therefore a 2D tracking system is necessary. The core of the 2D tracking system presented in this paper is a background subtraction model. The principle of the background subtraction method is to create a model of the scene's background, this background model is then used to determine the foreground as shown in Eq. (1.1):

$$|frame_i - background_i| > Th \tag{1.1}$$

where $frame_i$ and $background_i$ represent the frame and the background model at the instant i and Th is a global threshold.

Unfortunately, the image background is not stable. Therefore the background model needs to be able to cope with camera noise, illumination changes (fast and slow), new objects added or removed in the background (e.g. parked cars), repetitive motion (e.g. tree branches), etc. So, the simple approach of frame differentiation can only be used in very restricted environments, which is not the case of an outdoor environment. The simplest way to introduce memory in the background model is by using the average of the images across time as the background model. A running average as presented in Stauffer and Grimson (1999) can be used to reduce the memory requirements. In Lo and Velastin (2001) the background is updated by using a temporal median background update technique. A more developed statistical approach is presented in Wren *et al.* (1997), where a Gaussian probability density function (PDF) is used to model the process of every pixel. The PDF's parameters (mean and variance) are updated using the running average method presented in Stauffer and Grimson (1999).

Finally the 3D tracking is obtained by using the projection matrix, which incorporates the internal and external parameters of cameras, and the 2D tracking information from each camera. Detailed information of the process is provided in the following sections.

2. CAMERA CALIBRATION

Camera calibration is a required step for 3D tracking. A review of the camera calibration theory is given in Section 2.1 The implemented algorithms for internal and external calibration are presented in the Sections 2.2 and 2.3, respectively.

2.1. Camera calibration theory

A picture or a frame of a video sequence represents the projection of a 3D structure into a 2D image. This dimensional simplification, added to the deformation effects introduced by the camera lens, makes the reconstruction of the 3D world a non straightforward task. Faugeras *et al.* (1992) present a camera calibration method that does not require the use of a specific pattern; it is based on matching features in different views of the same scene. Therefore, this method is called camera auto- (self-) calibration. Nowadays, there are a wide variety of camera self-calibration schemes (Faugeras *et al.*, 1992; Hartley, 1994; Agapito *et al.*, 2001).

The most common camera model used in the computer vision community is the pinhole model (see Fig. 1). In this model the central projection of a 3D point $X = (X,Y,Z)^T$ into a point in the image plane $x = (x,y)^T$ is defined by the crossing of the line joining the camera center *C* and the 3D point *X* with the image plane. The image plane is the plane *Z* = *f*, where *f* is the focal length. Due to the dimension simplification from 3D to 2D, results are only determined up to a non-zero scale *w*. Therefore, in what follows, the sign = means equality up to an unknown scale factor *w*. Using the homogeneous notation, a *n*-dimensional point in Euclidean space, may be represented as a *n+1* vector in a projective space. For example, $\mathrm{X} = (X,Y,Z)^T$ in Euclidean space becomes $\mathrm{X} = (X,Y,Z,W)^T$.

Usually, the point X is normalized by *W* or $\mathrm{X} = \left(\frac{X}{W},\frac{Y}{W},\frac{Z}{W},1\right)^T = (X',Y',Z',1)$

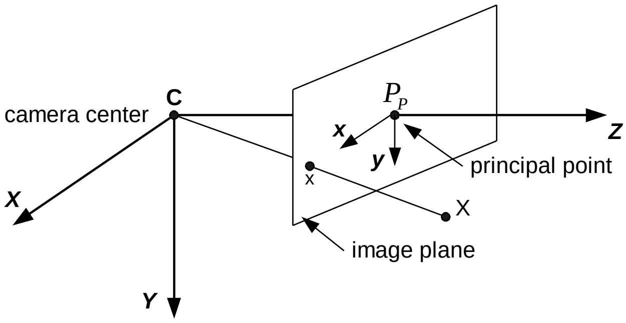

Figure 1. Pinhole model with the image plane in front of the camera center.

The camera projection matrix *P* encapsulates information about the camera's lens (internal parameters) and information about the relative camera's position and orientation (external parameters). The structure of the projection matrix is given by:

$$P = K[R\,|\,t] = KR[I\,|-C] \tag{2.1}$$

where *K* is a 3 x 3 camera calibration matrix, *R* is a 3 x 3 rotation matrix, *t* is a 3 x 1 translation vector, *C* is the 4 x 1 camera center in the 3D world coordinate system, and *I* is the 3 x 3 identity matrix.

2.2. Internal calibration

The most general expression for the camera calibration matrix *K* is given by:

$$K = \begin{bmatrix} \alpha_x & s & x_0 \\ 0 & \alpha_y & y_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.2}$$

where $\alpha_x = f\,m_x$, $\alpha_y = f\,m_y$, $x_0 = p_x m_x$, $y_0 = p_y m_y$, *s* the skew factor, *f* the focal length, and m_x and m_y are the number of pixels per unit length in each direction.

However, the *K* matrix for most modern cameras can be reduced, with acceptable loss of accuracy, to:

$$K = \begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix} \qquad (2.3)$$

where basically the focal length should be defined since the center of the image can be used as (p_x, p_y).

In order to define the *K* matrix, the method presented in Kim and Hong (2000) has been implemented. First, a set of overlapping images J_0, J_1, ..., J_N are acquired under a pure rotation of the camera. Then, for the images J_q, $q = 1, 2, ..., N$, 2D projective transformations H_1, H_2, ..., H_N are computed (Hartley and Zisserman, 2003) using point correspondences between images, where $J_0 = H_q J_q$ and H_q has the form:

$$H_q = K R_q K^{-1} \qquad (2.4)$$

A 2D projective transformation between a pair of matches X(x,y) ↔ X'(x',y') in two images is given by:

$$\mathrm{x}'_{\mathrm{i}} = H \, \mathrm{x}_i \qquad (2.5)$$

where *H* is a 3 x 3 matrix. *H* has 9 entries but, since it is defined only up to scale *H* has 8 degrees of freedom (Hartley and Zisserman, 2003); therefore 4 pairs of matches are necessary to determine *H* because each pair has 2 degrees of freedom corresponding to the *x* and *y* coordinates.

The matching process between images of the rotating camera uses corners that are detected using the Harris' method (Harris and Stephens, 1988; Mikolajczyk and Schmid, 2003). Then normalized cross correlation (Zitova and Flusser, 2003) is used to define putative matches that are refined using a voting algorithm based on a similarity transform (Lowe, 2001), and finally, Random Sample Consensus (RANSAC) (Fischler and Bolles, 1981) is used to define the projective transformation with the biggest support (highest number of inliers).

Internal calibration results

Once the positions of the cameras have been defined, videos under pure camera rotation (no translation) are taken with both cameras. Although pure rotation is assumed, this condition is violated most of the times since the precise location of the optical center is unknown and usually the rotation is carried out about a known fixed point that is close to the optical center. In order to compare the obtained results with different approaches, the software package Camera Calibration Toolbox for Matlab of Bouguet (2004) was used to define the camera calibration matrices for each camera used in in this study. Using Bouguet (2004) the results were:

$$K = \begin{bmatrix} 355{,}57 & 0 & 163{,}87 \\ 0 & 357{,}55 & 90{,}85 \\ 0 & 0 & 1 \end{bmatrix} \qquad (2.6)$$

$$K' = \begin{bmatrix} 347{,}20 & 0 & 156{,}29 \\ 0 & 348{,}67 & 93{,}75 \\ 0 & 0 & 1 \end{bmatrix} \qquad (2.7)$$

On the other hand, results of our implemented method were:

$$K = \begin{bmatrix} 347{,}47 & 0 & 160 \\ 0 & 347{,}47 & 120 \\ 0 & 0 & 1 \end{bmatrix} \qquad (2.8)$$

$$K' = \begin{bmatrix} 336{,}75 & 0 & 160 \\ 0 & 336{,}75 & 120 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.9}$$

where the positions of the principal points are assumed to be in the center of the image. The method presented in Bouguet (2004) requires the use of a special pattern and it is appropriate for laboratory environments rather than outdoor environments. Additionally, the computational reduction in the presented method is substantial.

2.3. External calibration

Generally, as shown in Fig. 2, the center of the camera does not necessary coincide with the world coordinate system (WCS). In this case Eq. (2.1) must be used for the camera projection matrix *P* since the camera coordinate system and the WCS are related by a 3D rotation and a 3D translation, as shown in Section 2.1.

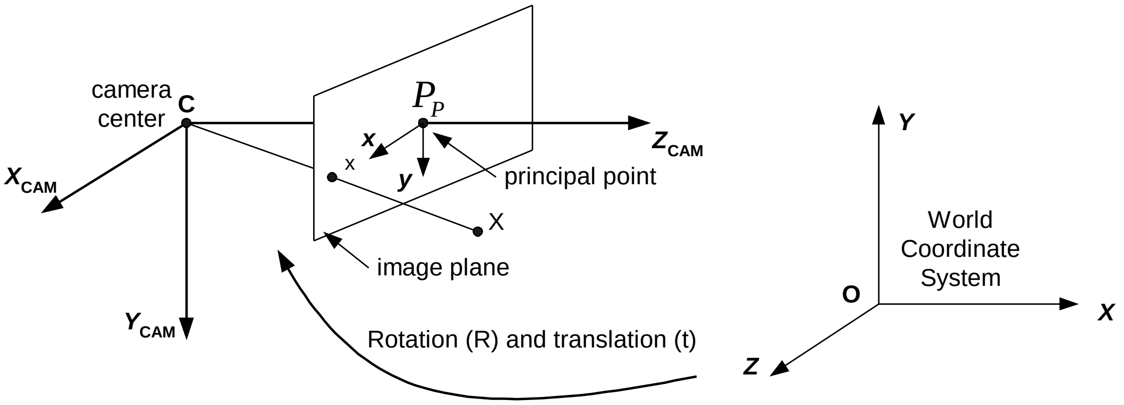

Figure 2. General situation where the camera coordinate system does not coincides with the global coordinate system.

The problem of camera calibration is to define the projection matrix *P* given a set of correspondences, $\mathrm{x}_i(x_i, y_i)^T \leftrightarrow \mathrm{X}_i(X_i, Y_i, Z_i)^T$ between 2D (image) and 3D (scene) points, respectively. From Eq. (2.1) the 3D to 2D projection is given by:

$$\mathrm{x} = P\,\mathrm{X} = K\,R\left[I \middle| -C\right]\mathrm{X} \tag{2.10}$$

Expanding Eq. (2.1) each correspondence provides a pair of equations, as follows:

$$x_i = \frac{p_{11}X_i + p_{12}Y_i + p_{13}Z_i + p_{14}}{p_{31}X_i + p_{32}Y_i + p_{33}Z_i + p_{34}} \tag{2.11}$$

$$y_i = \frac{p_{21}X_i + p_{22}Y_i + p_{23}Z_i + p_{24}}{p_{31}X_i + p_{32}Y_i + p_{33}Z_i + p_{34}} \tag{2.12}$$

where

$$P = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} \tag{2.13}$$

Equations (2.11) and (2.12) can be expressed as:

$$\begin{bmatrix} X & Y & Z & 1 & 0 & 0 & 0 & 0 & -x_i X_i & -x_i Y_i & -x_i Z_i & -x_i \\ 0 & 0 & 0 & 0 & X & Y & Z & 1 & -y_i X_i & -y_i Y_i & -y_i Z_i & -y_i \end{bmatrix} p = 0 \tag{2.14}$$

where *p* is a 12-element column vector that is obtained by unwrapping row-by-row the matrix *P* in Eq. (2.13).

With a minimum of 6 correspondences $\mathrm{x}_i \leftrightarrow \mathrm{X}_i$ a system of the form *Ap = 0* can be formed by stacking equations of the form of Eq. (2.14). Then, the resulting homogeneous system of equations can be solved using common methods such as the singular value decomposition (SVD) method which provides a linear solution that minimizes *Ap* under the constraint $|p| = 1$.

The *P* matrix in the Eqs. (2.1) and (2.13) can be expressed as:

$$P = [M \mid p_4] \tag{2.15}$$

where *M* is a 3 x 3 matrix formed with the first 3 columns of *P*, and p_4 is the last column of *P*.

From Eq. (2.1) it can be derived that the submatrix *M* corresponds to the product *M* = *KR*. *M* can be decomposed into *K* and *R* using the RQ-decomposition (Hartley and Zisserman, 2003). The translation vector *t* of *P* in Eq. (2.1) can be determined as:

$$t = K^{-1} p_4 \tag{2.16}$$

The camera position *C* with respect to the world coordinate system (WCS), which is the right null of *P* (i.e. *PC = 0*), can be found using:

$$C = -R^T t \tag{2.17}$$

Further, having defined the camera position *C* in the world coordinate system with Eq. (2.1) the translation vector *t* and the rotational matrix *R* can be derived, for which the MATLAB® implementation of the structure from the motion method for a single moving camera was used as presented in Qian and Chellappa (2004). In fact, two coordinate systems (CS) are used to model the motion of the camera. The first CS, called inertial world coordinate system (IWCS), is fixed to the initial position of the sensor, while the second CS moves attached to the sensor. Both CS have their origins on the centers of projection of the sensor, their X and Y planes parallel to the image planes and the Z plane normal to the image planes with their positive half parts pointing toward the observed scene.

The motion of the sensor at time k with respect to the inertial coordinate system is described using 5 parameters $(\psi_x, \psi_y, \psi_z, \alpha, \beta)$ where $\Psi = (\psi_x, \psi_y, \psi_z)^T$ are the rotation angles of the camera with respect to the IWCS, and *α* and *β* are the elevation and azimuth angles of the translation of the moving camera with respect to the IWCS.

A state space model with the state vector $\mathbf{X}_\mathrm{k}$ and the observation of the image features $\mathbf{Y}_\mathrm{k}$ at the stamp time k, can be written as:

$$\mathbf{X}_{\mathrm{k}+1} = \mathbf{X}_\mathrm{k} + n_x \tag{2.18}$$

$$\mathbf{Y}_\mathrm{k} = \mathrm{Proj}(\mathbf{X}_\mathrm{k}, S_\mathrm{k}) + n_y \tag{2.19}$$

where $\mathbf{X}_\mathrm{k} = (\psi_x, \psi_y, \psi_z, \alpha, \beta)^T$, n_x describes the time varying property of the state vector and $\mathrm{Proj}(\cdot)$ denotes the perspective projection which depends on the camera relative movement with respect to the inertial coordinate system ($\mathbf{X}_\mathrm{k}$) and features from the scene structure S_k.

The translation and rotation matrices can be obtained from the state vector using the Eqs. (2.20 and 2.21), or:

$$T(\alpha, \beta) = [\sin(\alpha)\cos(\beta), \sin(\alpha)\sin(\beta), \cos(\alpha)]^T \tag{2.20}$$

$$R = \begin{bmatrix} n_1^2 + \left(1-n_1^2\right)\eta & n_1 n_2 (1-\eta) + n_3\,\varsigma & n_1 n_3 (1-\eta) - n_2\,\varsigma \\ n_1 n_2 (1-\eta) - n_3\,\varsigma & n_2^2 + \left(1-n_2^2\right)\eta & n_2 n_3 (1-\eta) + n_1\,\varsigma \\ n_1 n_3 (1-\eta) - n_2\,\varsigma & n_2 n_3 (1-\eta) - n_1\,\varsigma & n_3^2 + \left(1-n_3^2\right)\eta \end{bmatrix} \tag{2.21}$$

where $n = (n_1, n_2, n_3)^T = \frac{\Psi}{|\Psi|}$, $\eta = \cos(|\Psi|)$ and $\varsigma = \sin(|\Psi|)$.

Using this state model, a sequential importance sampling (SIS) (Liu and Chen, 1998) can be applied to find an approximation of the posterior distribution of the motion parameters $P(\mathbf{X}_\mathrm{k} \mid \mathbf{Y}_\mathrm{k})$. Due to the total lack of previous information about the camera motion, samples for the motion parameters are taken randomly. Normally distributed samples $N(0,\sigma^2)$ are used for the rotation parameters ψ_x, ψ_y, ψ_z, and uniformly distributed samples within the dynamic range of each parameter are used for the translation parameters α, β.

For each matched feature obtained from the wide baseline matching (WBM) process, the likelihood $f\left(y_\mathrm{k}^{(i)} \mid \mathbf{X}_\mathrm{k}\right)$ under a set of motion parameters is evaluated using the epipolar distance from the feature position in camera 2 with respect to the epipolar line in the image plane in camera 2, which is a function of the motion parameters and the position of the same feature in the image plane of the first camera. The weight for a set of motion parameters uses all the feature matching from the WBM as given in Eq. (2.22):

$$f(\mathbf{Y}_\mathrm{k} \mid \mathbf{X}_\mathrm{k}) = \prod_{i=1}^{M} f\left(y_k^{(i)} \mid \mathbf{X}_\mathrm{k}\right) \tag{2.22}$$

where *M* is the number of matching features from the WBM algorithm. Only those motions sets that provide positive depths in the matching points will be selected for the resampling process. Given the projections x, x' of a 3D point X in the image plane of two cameras, *C* and *C'*, and the motion parameters between both cameras (*R*,*t*), the depth of X can be determined by minimizing the projection error in one of the image planes, and can be expressed as:

$$depth = \frac{(t_x r_z - t_z r_x)(t_z u - t_x) + (t_y r_z - t_z r_y)(t_z v - t_y)}{(t_x r_z - t_z r_x)(r_z u - r_x) + (t_y r_z - t_z r_y)(r_z v - r_y)} \tag{2.23}$$

where $(u,v,1)^T = K'^{-1}\mathrm{x}'$, $(t_x, t_y, t_z)^T = -t$ and $(r_x, r_y, r_z)^T = RK^{-1}\mathrm{x}$. The rotation matrix and translation vector (*R*,*t*), can be obtained from the state vector $\mathbf{X}$ using the Eqs. (2.20) and (2.21).

The motions sets that have shown good properties, i.e. those that provide positive depths and small epipolar distance (high likelihood) for the matching features from the WBM, are resampled maintaining a proper weighting in the samples. These samples can be used in a new iteration to improve the motion results. In this case, the motion parameters will be added noise in order to provide a refinement around each surviving set of motion parameters.

Once the posterior distribution of the motion parameters have been determined, peaks are found in the distributions of every parameter.

External calibration results

After the videos under pure rotation are taking for the internal calibration process, final positions of both cameras are defined and fixed, then a still picture from each camera is used to find feature correspondences in both pictures by applying the wide baseline matching process, named SIFT, which is described in Section 3.

A set of results for the structure from motion are presented in Fig. 3. Peaks values are determined as:

$$\mathbf{X} = (\psi_x, \psi_y, \psi_z, \alpha, \beta)^T = (0{,}0585,\ \ -0{,}0302,\ \ 0{,}0164,\ \ 1{,}5476,\ \ 4{,}5397)^T \tag{2.24}$$

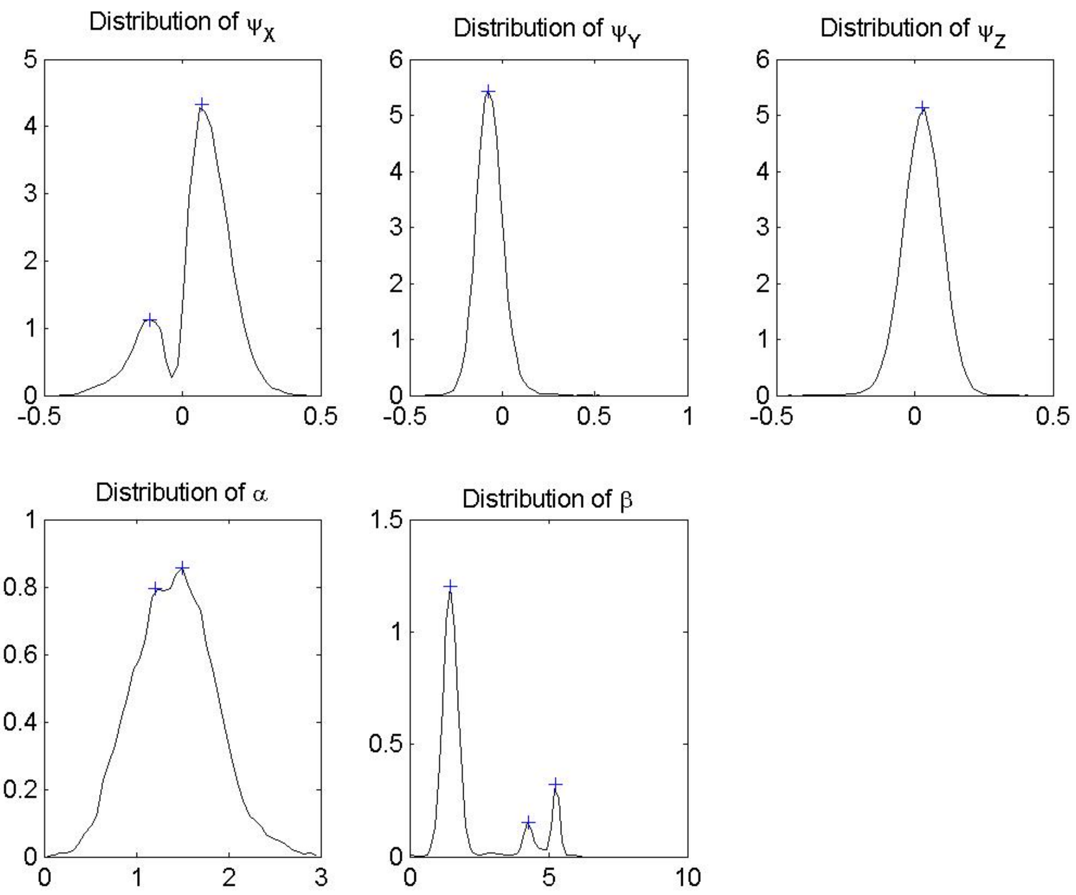

Figure 3. Posterior distribution of the sensor motion parameters.

The translation vector and rotation matrix obtained using the Eqs. (2.20) and (2.21) respectively are:

$$t = \begin{bmatrix} -0{,}1718 \\ -0{,}9849 \\ 0{,}0232 \end{bmatrix} \tag{2.25}$$

$$R = \begin{bmatrix} 0{,}9994 & 0{,}0155 & 0{,}0306 \\ -0{,}0172 & 0{,}9982 & 0{,}0582 \\ -0{,}0297 & -0{,}0587 & 0{,}9978 \end{bmatrix} \tag{2.26}$$

The projection matrices for each camera are obtained by replacing the Eqs. (2.25) and (2.26) in Eq. (2.1), or:

$$P = \begin{bmatrix} 347{,}47 & 0 & 160 & 0 \\ 0 & 347{,}47 & 120 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{2.27a}$$

$$P' = \begin{bmatrix} 331{,}8031 & -4{,}1866 & 169{,}9698 & -54{,}1469 \\ -9{,}3668 & 329{,}0803 & 139{,}3521 & -328{,}8658 \\ -0{,}0297 & -0{,}0587 & 0{,}9978 & 0{,}0232 \end{bmatrix} \tag{2.27b}$$

These matrices will be used in Section 4 to obtain the 3D tracking from the 2D tracking information of both cameras.

3. WIDE BASELINE MATCHING

Cameras providing surveillance of an area have generally wide relative positions. A main step in the proposed system is the wide baseline matching (WBM) of the views from both cameras. The problem of establishing reliable correspondences in images of the same scene but with different viewpoints, requires dealing with different scales, different illumination conditions and different geometric transformations between the images. Therefore, well known approaches used in template matching, such as the normalized cross correlation methods (Zitova and Flusser, 2003) can not be used.

One of the WBM methods is the Scale Invariant Feature Transform (SIFT) scheme (Lowe, 2004), which according to Mikolajczyk and Schmid (2003) results in highly distinctive feature descriptors. The adopted SIFT method defines putative matches between images of both cameras using an exhaustive search among feature descriptors in both images. In order to determine reliable matches from the aforementioned putative matches, a voting algorithm based on a similarity transform (Hartley and Zisserman, 2003) and a robust statistical method known as Random Sample Consensus, RANSAC (Fischler and Bolles, 1981; Hartley and Zisserman, 2003), is implemented in order to select the matches that satisfy the epipolar constraint.

3.1. Detection of invariant features

Invariant features, also known as keypoints, should be as invariant as possible to scale changes, geometric transformations (e.g. affine transform, Ma *et al*., 2004), and changes in illumination. To this end scale-space methods (Lindeberg, 1993) are widely used for finding scale-invariant features. Scale-space schemes provide a three-dimensional (3D) representation of an image in the form of *(x,y,r)* where *(x,y)* are the image spatial coordinates and *r* represents the scale. In Mikolajczyk and Schmid (2004 and 2011) a multi-scale version of the Harris' corner detector, known as Harris-Laplace, is presented. Harris-Laplace keypoints are first defined in the *(x,y)* image coordinates and, then, points that correspond to a local maxima through scales, are selected. A similar keypoint detector approach is used by Dufournaud *et al.* (2000) to match images with different resolutions (scales). Successful matches are reported in images of the same scene at different spatial resolutions but taken from the same viewpoint.

Detection of local maxima and minima throughout a scale-space domain that is created using a difference-of-Gaussian (DOG) is used in Lowe (2004). Only those keypoints that show a high contrast and curvature response (Harris and Stephens, 1988) are kept. Jiangjian and Shah (2003) use a novel edge-corner affine-invariant feature. First, edges are detected using the Canny edge detector (Canny, 1986) and, then, a Hough transform is used to find straight lines (and their slopes) through points over the previous detected edges. A point is considered as a corner candidate if more than one edge cross the point. Finally, Harris' corner operator (Harris and Stephens, 1988) is applied to measure the corner response. Results show that this method outperforms the approach presented in Mikolajczyk and Schmid (2011), but a comparison of computational requirements is not available. In Schmid *et al.* (2002) six interest point detectors (Harris and Stephens, 1988; Horaud *et al.*, 1990; Heitger *et al.*, 1992; Cottier, 1994; Förstner, 1994; Schmid *et al.*, 2002) are evaluated under the criteria of repeatability and information content. Repeatability, which shows the geometric stability of the detected keypoints under different transformations, is analyzed under changes in rotation, illumination, scale, viewpoint and camera noise.

3.2. Keypoint local descriptor

Once interest points have been localized, a signature has to be assigned to each keypoint in order to provide a unique identification. In Lowe (2004) a biological vision based descriptor is presented grounded on the response of complex neurons in the primary visual cortex. The method creates descriptors of 128 elements, defined using an array of weighted and oriented histograms around keypoints that are found in a scale-space created by means of DOGs. In Mikolajczyk and Schmid (2003) an evaluation of six different descriptors is presented. The evaluated descriptors include SIFT descriptors (Lowe, 2004), differential invariants (Koenderink and Van Doorn, 1987), moment invariants (Van Gool *et al.*, 1996), steerable filters (Freeman and Adelson, 1991), complex filters (Schaffalitzky and Zisserman, 2002) and cross correlation (Zitova and Flusser, 2003). Different interest point detectors were used as part of the evaluation, and it was concluded that the point detector does not influence the ranking of the descriptors. It was also found that the SIFT method of Lowe (2004) is the feature descriptor method that presents an overall better detection rate under image rotation, scale changes, affine transformations and illumination changes.

3.3. Feature matching

An exhaustive search can be used to find the nearest neighbors in high dimensional spaces (e.g. 128 in Lowe, 2004). Of course, the complexity of the feature matching method depends on the level of distinctiveness of the used feature descriptors. Some methods use the minimum Euclidian distance among descriptors to find possible matches (Lowe, 2004), while others report the use of the Mahalanobis distance (Baumberg, 2000). The epipolar geometry is normally used as a last step in almost all matching methods (Pritchett and Zisserman, 1998; Baumberg, 2000; Jiangjian and Shah, 2003; Lowe, 2004).

3.4. Scheme for Wide Baseline Matching

The SIFT feature descriptor method (Lowe, 2004) has been adopted as part of the implemented WBM scheme since it outperforms other methods (Mikolajczyk and Schmid, 2003). Although SIFT is an empirically based method, its descriptors have good invariant properties with respect to image rotation, scale, illumination changes and affine transformations. Keypoints can be defined using any invariant point detector. Lowe (2004) proposed a method that seeks absolute maxima through a scale–space map of the image. The main steps of the SIFT method are described in the following.

3.4.1. Scale-space feature detection

When creating a multi-scale representation of an image, one has to create a family of images at different scales, where high frequency information, or fine-scale details, are continuously suppressed or diffused. This procedure is called scale-space smoothing or blurring. The Gaussian kernel has been shown to be one kernel (Lindeberg, 1993) that can be used to create a scale-space representation of a signal in any dimension. Additionally, a Gaussian kernel is self-reproducing as shown below:

$$G_{\sigma_1} \otimes \left(G_{\sigma_2} \otimes f\right) = \left(G_{\sigma_1} \otimes G_{\sigma_2}\right) \otimes f = G_\sigma \otimes f \qquad (3.1)$$

where G_σ is a Gaussian filter with a standard deviation of σ, f is the filtered signal, $\sigma^2 = \sigma_1^2 + \sigma_2^2$ and $\otimes$ represents the convolution operation. Therefore, incremental smoothing can be easily implemented.

A pyramid representation of an image is a stack of successive blurred and subsampled versions of the original image. The pyramid is called oversampled when not all the blurred levels are followed by a subsampled action. Usually, the subsampling action has to reduce the number of pixels by a factor 2^N, where N is the dimension of the signal. In this study N = 2. The image size decreases exponentially in pyramids, which implies a reduction in computations at each subsequent stage of the pyramid.

Keypoints in SIFT are found by searching for absolute maxima in a Laplacian pyramid built as a difference of low pass filters (DOLP) (Crowley and Stern, 1982). When Gaussian low pass filters are used in the DOLP, the DOLP receives the name of Difference of Gaussians (DOG).

Let $L(x,y,\sigma)$ be the resulting Gaussian image obtained after convolving the image $I(x,y)$ with a finite Gaussian mask with a standard deviation *σ*, $G_\sigma = G(x,y,\sigma)$, then $L(x,y,\sigma)$ can be expressed as:

$$L(x,y,\sigma) = G(x,y,\sigma) \otimes I(x,y) \tag{3.2}$$

where:

$$G(x,y,\sigma) = \frac{1}{\left(\sqrt{2\pi\sigma}\right)^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{3.3}$$

then, the DOG image, $DOG(x,y,\sigma)$, can be expressed as the subtraction of two of the aforementioned Gaussian images *L*, as shown in Eq. (3.4):

$$DOG(x,y,\sigma_1) = L(x,y,\sigma_2) - L(x,y,\sigma_1) \tag{3.4}$$

where $\sigma_2 > \sigma_1$.

In order to avoid discarding the high frequency information of the image due to the blurring applied before the keypoint detection, the image is first upsampled by a factor of two using linear interpolation. In Lowe (2004) an oversampled Laplacian pyramid is used that has the following parameters:

- ✓ Number of octaves. This is the number of times that the initial *σ* will be doubled through the scale-space; downsampling is applied every time the standard deviation is duplicated.
- ✓ Number of intervals. Represents the number of levels in which the octaves will be divided. This number also represents the number of searches for maxima that will be carried out in groups of 3 neighboring DOGs.

The number of intervals, *s*, defines the step *k* that the standard deviation of the Gaussian filters will take, by the relation:

$$k = 2^{1/s} \tag{3.5}$$

Then, the standard deviation of the Gaussian filters that are needed to obtain the Gaussian images by convolving directly with the original image $I(x,y)$ are given by:

$$\sigma_n = k\sigma_{n-1} \qquad n > 1 \tag{3.6}$$

But, using the self-reproducing characteristic of the Gaussian kernel as stated in Eq. (3.1), the standard deviation of the Gaussian filters can be found as:

$$\sigma_n = \sigma_{n-1}\sqrt{k^2 - 1} \tag{3.7}$$

Since the SIFT method finds keypoints by detecting local maxima and minima in a DOG and its two immediate neighbors (previous and next), *s + 2* number of DOG levels are needed per octave. The DOG in which the keypoint was detected will be referred as the “keypoint DOG image”. Let $L(x,y,k^i\sigma)$ and $L(x,y,k^{i+1}\sigma)$ be the 2 Gaussian images that were used to create the DOG in which the keypoint was detected, the Gaussian image $L(x,y,k^i\sigma)$ with the smaller standard deviation, $\left(k^{i\sigma} < k^{i+1}\sigma\right)$ will be denoted as the “keypoint Gaussian image”.

Although the number of keypoints increases with *s*, an optimum value that maximizes the repeatability is reached using 3 levels per octave (*s = 3*). An example of the scale-space of Gaussians for three octaves and three intervals per octave is shown in Fig. 4 and the corresponding DOG images are shown in Fig. 5.

Once keypoints have been detected the following two refinement processes are carried out.

a) Low contrast elimination. Detected keypoints that have low contrast are discarded since they are unstable.

b) Edge responses elimination. Keypoints that are unstable with respect to small amounts of noise can be detected along edges since the DOG has a strong response along edges. These keypoints can be eliminated by comparing their responses in all directions. This can be done using the Hessian matrix (Harris and Stephens, 1988), given by:

$$H = \begin{bmatrix} D_{XX} & D_{XY} \\ D_{XY} & D_{YY} \end{bmatrix} \tag{3.8}$$

where the derivatives D_{XX}, D_{YY} and D_{XY} are obtained by taking a difference of neighboring samples in the DOG image of the keypoint, and finally, keypoints are selected as valid if they satisfy the following:

$$\frac{\mathrm{Tr}(H)^2}{\mathrm{Det}(H)} < \frac{(r+1)}{r} \tag{3.9}$$

where Tr(*H*) and Det(*H*) represent the trace and determinant function of the *H* matrix, respectively. A value of *r = 10* is suggested.

An orientation, and local descriptor will be assigned to the keypoints that satisfy Eq. (3.9).

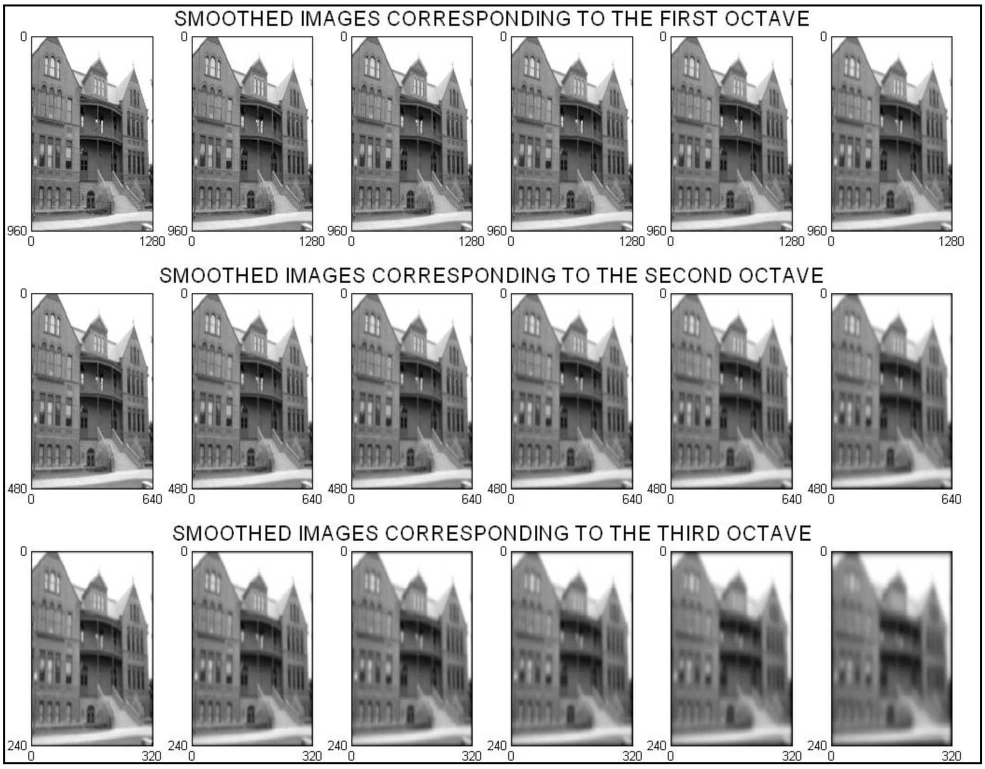
SMOOTHED IMAGES CORRESPONDING TO THE FIRST OCTAVE
0
960
1280
SMOOTHED IMAGES CORRESPONDING TO THE SECOND OCTAVE
0
480
640
SMOOTHED IMAGES CORRESPONDING TO THE THIRD OCTAVE
0
240
320

Figure 4. Scale-space of Gaussians, three octaves and three intervals per octave has been used.

3.4.2. Keypoint orientation

The keypoint orientations are found by determining the prominent orientations of the weighted histogram of the gradient orientations. To provide scale-invariance, the gradient orientation, as well as its magnitude, are calculated at each sample position using neighboring sample differences in the Gaussian image of the keypoint using the Eqs. (3.10) and (3.11), respectively:

$$m(x,y) = \sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2} \tag{3.10}$$

$$\theta(x,y) = \mathrm{arctan}((L(x,y+1) - L(x,y-1))/(L(x+1,y) - L(x-1,y))) \tag{3.11}$$

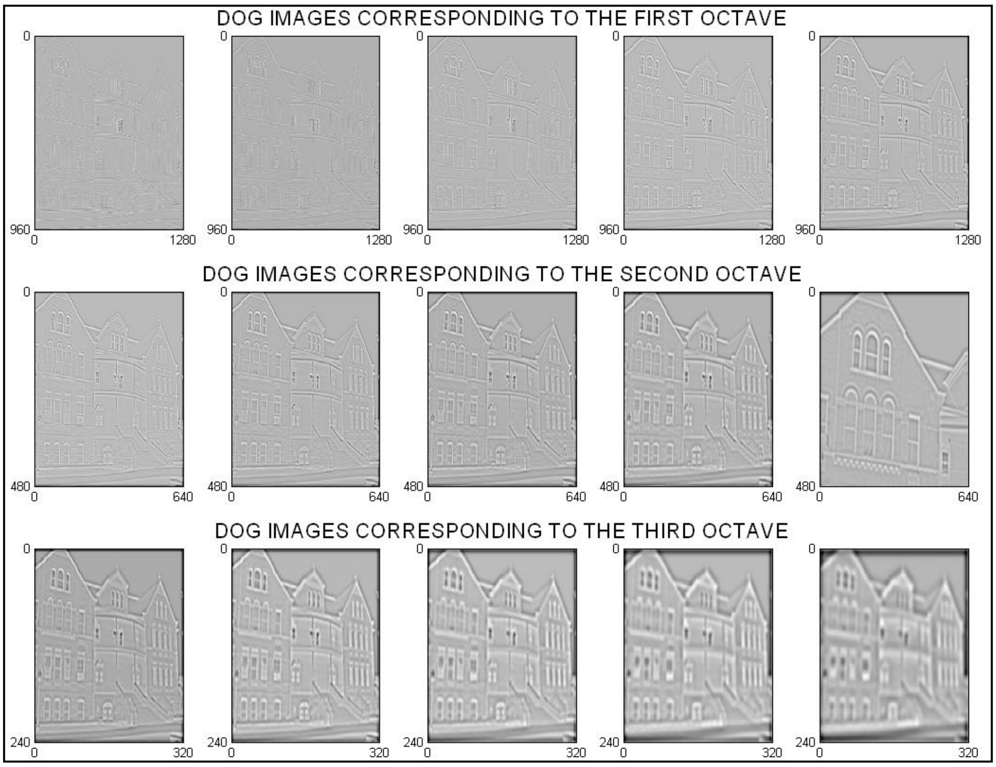
DOG IMAGES CORRESPONDING TO THE FIRST OCTAVE
0
960
0
1280
DOG IMAGES CORRESPONDING TO THE SECOND OCTAVE
0
480
0
640
DOG IMAGES CORRESPONDING TO THE THIRD OCTAVE
0
240
0
320

Figure 5. Difference of Gaussians acquired from subtracting contiguous levels of Gaussians shown in Fig. 3.

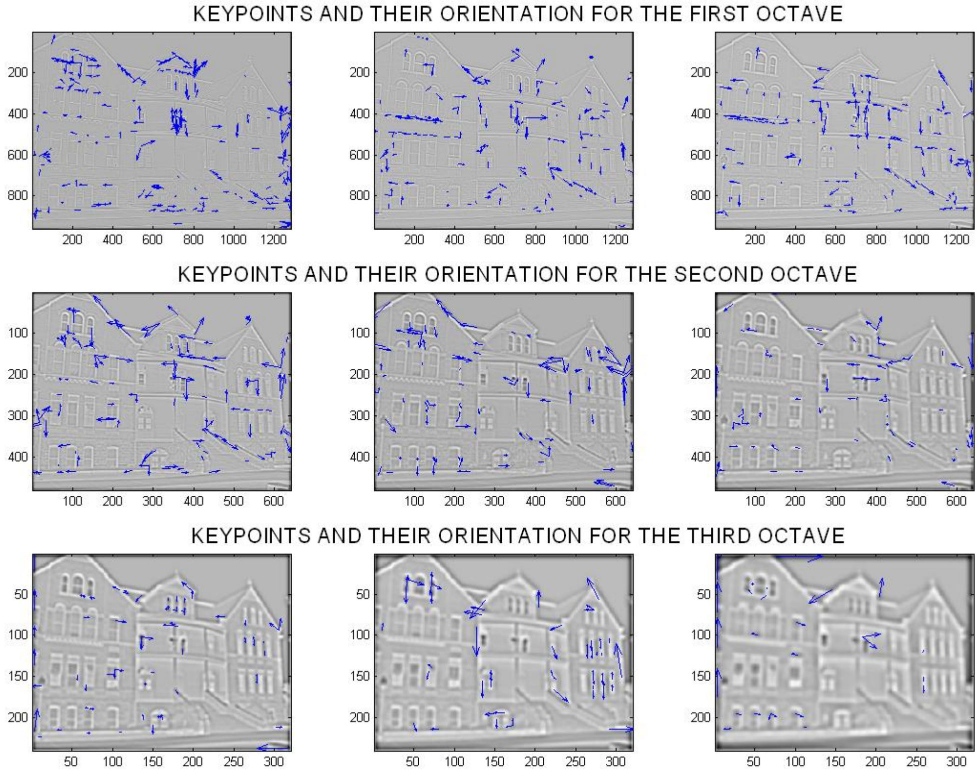
KEYPOINTS AND THEIR ORIENTATION FOR THE FIRST OCTAVE
200
400
600
800
200 400 600 800 1000 1200
KEYPOINTS AND THEIR ORIENTATION FOR THE SECOND OCTAVE
100
200
300
400
100 200 300 400 500 600
KEYPOINTS AND THEIR ORIENTATION FOR THE THIRD OCTAVE
50
100
150
200
50 100 150 200 250 300

Figure 6. Keypoints detected on the scale-space of Fig. 3. Arrows' orientations indicate the orientation of the keypoints and their sizes are proportional to the magnitude of the gradient.

Since a local descriptor is desired for each keypoint, a circular Gaussian mask centered at the considered keypoint is used as a weighting function in order to weight the computed gradient orientations. In this way, more emphasis is given to pixels that are close to the keypoint (Lowe, 2004). Since a local descriptor is desired for each keypoint, a circular Gaussian mask centered at the considered keypoint is used as a weighting function in order to weight the computed gradient orientations. In this way, more emphasis is given to pixels that are close to the keypoint. (Lowe, 2004) suggests that a standard deviation equals to 1,5 times the standard deviation of the keypoint Gaussian image of the keypoint be used for the applied circular Gaussian mask in order to find the gradient orientation. Then, a weighted histogram of gradient orientations with 36 bins is created, which corresponds to a bin spacing of 10°. The maximum orientation of the created histogram is assigned as the orientation of the keypoint. Figure 6 shows the positions of the detected keypoints and their orientations illustrated by the directions of the shown arrows, for the scale-space example of Fig. 4.

3.4.3. Local image descriptor

The last step in SIFT (Lowe, 2004) consists of assigning to each keypoint, a local feature descriptor based on the information around the keypoint. For this purpose an array of weighted histograms of gradient orientations of the keypoint Gaussian image is used. Lowe (2004) established experimentally that a window of 16 x 16 samples, used to create an array of 4 x 4 histograms, provides the highest rate of keypoints that match correctly in a database of 40000 keypoints. Histograms of 8 bins, i.e. bins of 45° are employed; therefore, keypoints descriptors of 4 x 4 x 8 = 128 elements are assigned.

3.4.4. Feature matching

Although SIFT descriptors have good distinctiveness properties, i.e. they have information that allows to establish similarities among them, accurate matching can not be achieved by just seeking for descriptors with smallest Euclidean distances in both images. Therefore, the keypoint matching implementation presented in this paper consists of the following steps.

a) Selection of Regions of Interest (ROI). In order to improve the performance of the matching ROIs may be selected in one of the images to be matched. ROIs shall not be selected by only considering common content in both images, but also by avoiding the consideration of arrays of identical features, e.g. rows of identical windows, since wrong matches can be introduced. Once the ROI has been selected, Euclidian distances are computed and a group of tentative matches, consisting of the closest N matches, is assigned to every keypoint within the selected ROIs. Experimentally, we concluded that a value of N = 3 is recommended since bigger groups produce mismatches and also represent higher computation requirements.

b) Similarity voting. According to Lowe (2001) the use of a voting algorithm based on a similarity transform, performs better with more complex 3D objects, than a one based on an affine transform (Lowe, 1990). A similarity transform consists of an isotropic scaling and a translation as follows:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} p \cdot \cos\theta & -p \cdot \sin\theta \\ p \cdot \sin\theta & p \cdot \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix} \tag{3.12}$$

where p is the isotropic scaling factor, θ is a rotation angle and t_x,t_y represent the translation in the x and y directions, respectively.

Equation (3.12) can be rewritten as:

$$\begin{bmatrix} x & -y & 1 & 0 \\ y & x & 0 & 1 \end{bmatrix} \begin{bmatrix} m \\ n \\ t_x \\ t_y \end{bmatrix} = \begin{bmatrix} x' \\ y' \end{bmatrix} \tag{3.13}$$

where $m = p\ cos\theta$ and $n = p\ sin\theta$.

The voting algorithm consists of solving Eq. (3.13) for every tentative match.

c) Robust matching using epipolar constraint. After these procedures some mismatches can still be seen. These mismatches can be reduced by using the statistical method Random Sample Consensus (RANSAC) (Fischler and Bolles, 1981; Hartley and Zisserman, 2003) to select those matches that satisfy more precisely the constraint of the epipolar geometry. Using RANSAC, the fundamental matrix is computed using a set of randomly selected pairs. In the present work, the normalized 8 points algorithm was adopted (Hartley, 1997) in which sets of eight pairs of matches are used. Then, for each computed F the number of supporting pairs are determined as those that satisfy, within some margin of tolerance, the relation:

$$\mathrm{x}'F\mathrm{x} = 0 \tag{3.14}$$

where x and x' are the tentative matched points, and F is the fundamental matrix.

The F matrix that gets the largest number of supporting pairs is selected. The supporting pairs of the selected F matrix are called inliers and represent the final matching pairs. The matching features will be used to determine the relative 3D camera movement (R,t).

Wide baseline matching results

Tentative matches after the similarity voting are presented in Fig. 7. There are many points that are not present in both images. Therefore, the RANSAC algorithm is applied to enforce epipolar constrains in both images. Final results of feature matching are presented in Fig. 8. Feature matching results for the example presented in Section 2 are shown in Fig. 9.

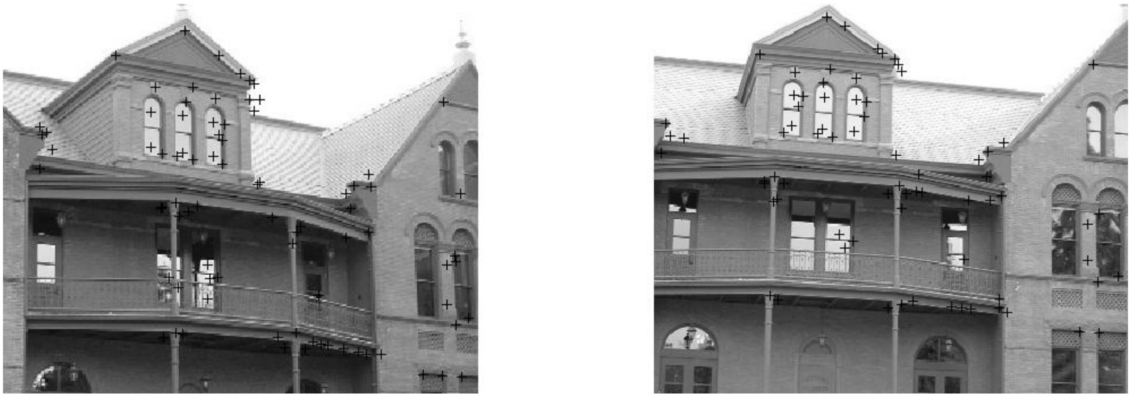

Figure 7. Tentative matches after similarity voting.

4. 2D AND 3D TRACKING

Three-D tracking of objects that are in the field of view on 2 cameras can be obtained using the 2D tracking information of the objects in each of the cameras. Therefore, a 2D tracking system is necessary. The core of the 2D implemented tracking system is a background subtraction model. The 2D tracking information, as well as the projective matrices of both cameras is fed into a triangulation algorithm (Hartley and Sturm, 1995) to recover the 3D information of the track.

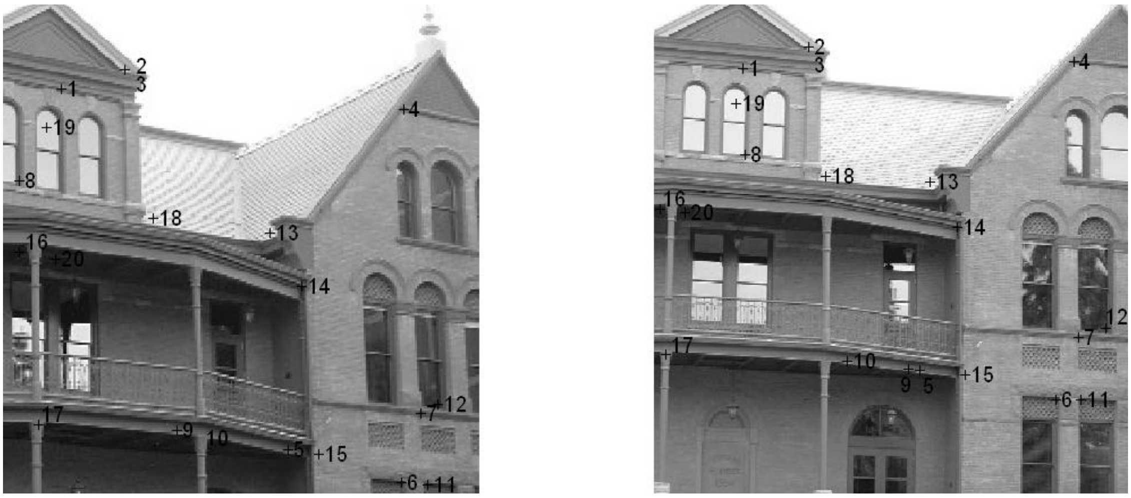

Figure 8. Final matching points using RANSAC to enforce the epipolar constrains in both images.

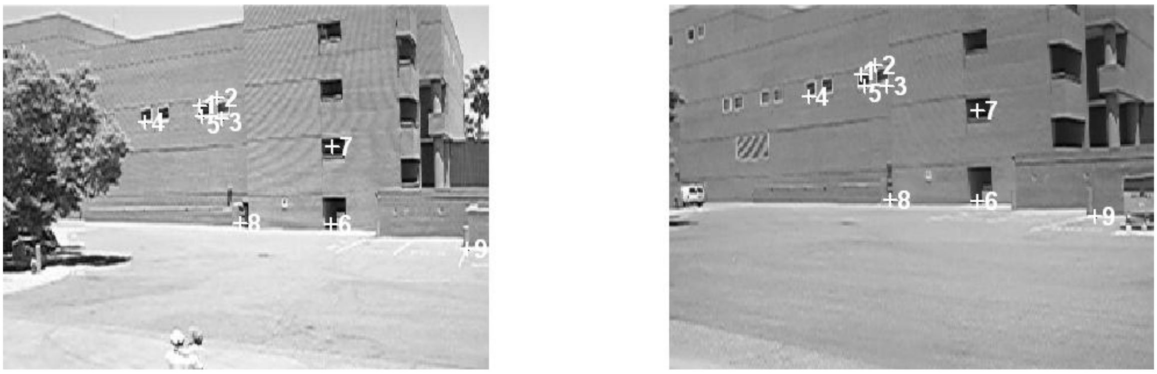

Figure 9. Final matching points applying RANSAC to enforce the epipolar constrains in both images of the example presented in Section 2.

4.1. 2D tracking

The principle of background subtraction methods is to create a model of the scene's background. This background model is then used to determine the foreground as shown in Eq. (4.1), see also Eq. (1.1):

$$\left| frame_i - background_i \right| > Th \qquad (4.1)$$

where $frame_i$ and $background_i$ represent the frame and the background model at the instant *i*, and *Th* is a global threshold.

Different background subtraction methods present different approaches to model the background scene. As stated in the Introduction of this paper, the image background is not stable. Therefore, the background model needs to be able to cope with camera noise, illumination changes (fast and slow), new objects added or removed in the background (e.g. parked cars), and repetitive motion (e.g. tree branches) for example. So, the simple approach of frame differentiation can only be used in very restricted environments, which is not the case of an outdoor environment.

The background subtraction used in this work is close to the method presented in Elgammal *et al.* (2002). A set of *N* previous frames are kept in memory, $I_1, I_2, \ldots, I_N$. Given a new frame, I_t at time $t = N + 1$, the probability of each new pixel can be estimated as follows:

$$\Pr(I_t) = \frac{1}{N}\sum_{i=1}^{N}\prod_{j=1}^{3}\frac{1}{\sqrt{2\pi\Sigma_j}}e^{-\frac{(I_{tj}-I_{ij})^2}{2\Sigma_j^2}} \tag{4.2}$$

where j corresponds to the indices of color planes and $\sum_j$ is the standard deviation per pixel and per image plane. Then, a pixel is considered to belong to the foreground if the following condition is satisfied:

$$\Pr(x_t) < Th \tag{4.3}$$

where the threshold is global for the whole image.

Since a normal distribution $N(\mu, \sigma^2)$ is assumed for every pixel in the image I_i, the distribution of the difference of neighbor images, $(I_i - I_{i-1})$, is also normal given by $N(0, 2\sigma^2)$. The standard deviation can be calculated as:

$$\sigma = \frac{m}{0.68\sqrt{2}} \tag{4.4}$$

where m is the median of $|I_i - I_{i-1}|$ for every image I_i in memory.

Post-processing is applied to the detected foreground information since the resulting foreground presents false detections. Morphological erosion with a square mask is applied to eliminate false detections; but, since the erosion also eliminates correct foreground, pixel connectivity is done between closest neighbors.

In order to segment different objects in the scene, labeling is applied, and the mass center of the detected blobs, formed by a number of pixels higher than a threshold, is extracted. Finally, the position of the detected objects are ordered based on the Euclidean distance with respect to the position of the objects detected in the previous frame. The resulting 2D track information from both cameras is further used in the triangulation procedure.

2D tracking results

Results of the 2D tracking is presented in Fig. 10. The undesirable effect of the trees and shadows can be appreciated in Fig. 10(b). Spurious foreground dots are eliminated by using erosion, the foreground image after been eroded by a 2 x 2 mask is shown in Fig. 10(c). Since erosion also carves part of the useful foreground, blobs of foreground close to each other with a distance of 15 and 8 pixels along the vertical and horizontal directions, respectively, are connected. The mass center of the blobs found in Fig. 10(d) are used as the 2D tracking of the object in the image plane of each camera.

4.2. 3D tracking

The position of a 3D point X can be determined by triangulation given its projections in two cameras x and x'. But, from the 2D tracking one does not have any clue about the correspondent object in the other camera (when more than one moving object are presents in the field of view); therefore, a cross correspondence ought to be established among objects in both cameras previously to apply triangulation.

The fundamental matrix obtained from the wide baseline matching in Section 3 does not provide acceptable results since it is determined using points that are not in the plane where people is being tracked (building features). Therefore, the fundamental matrix was enriched by including correspondences of the people being tracked.

2D homographies are used in order to find cross correspondences in the 2D tracking information of both cameras. Using a set of n random selected points, homographies are computed for all possible combinations among the objects in both cameras applying the normalized direct linear transformation (NDLT); the inliers of H, defined as those pairs that satisfy

$$\mathrm{x}' = H\,\mathrm{x} \tag{4.5}$$

within some tolerance, are determined for every *H*. The homography that gets more support, i.e. more inliers pairs is selected and its inliers are included in the enrichment of the fundamental matrix. In Fig. 11 the result of the cross correspondence among 2D tracks in 2 cameras is presented.

The fundamental matrix computed using the feature matching from the WBM scheme and the cross correspondences among 2D tracks has shown good results when used to establish cross correspondences. Figure 12 shows the cross object correspondence established in two cameras.

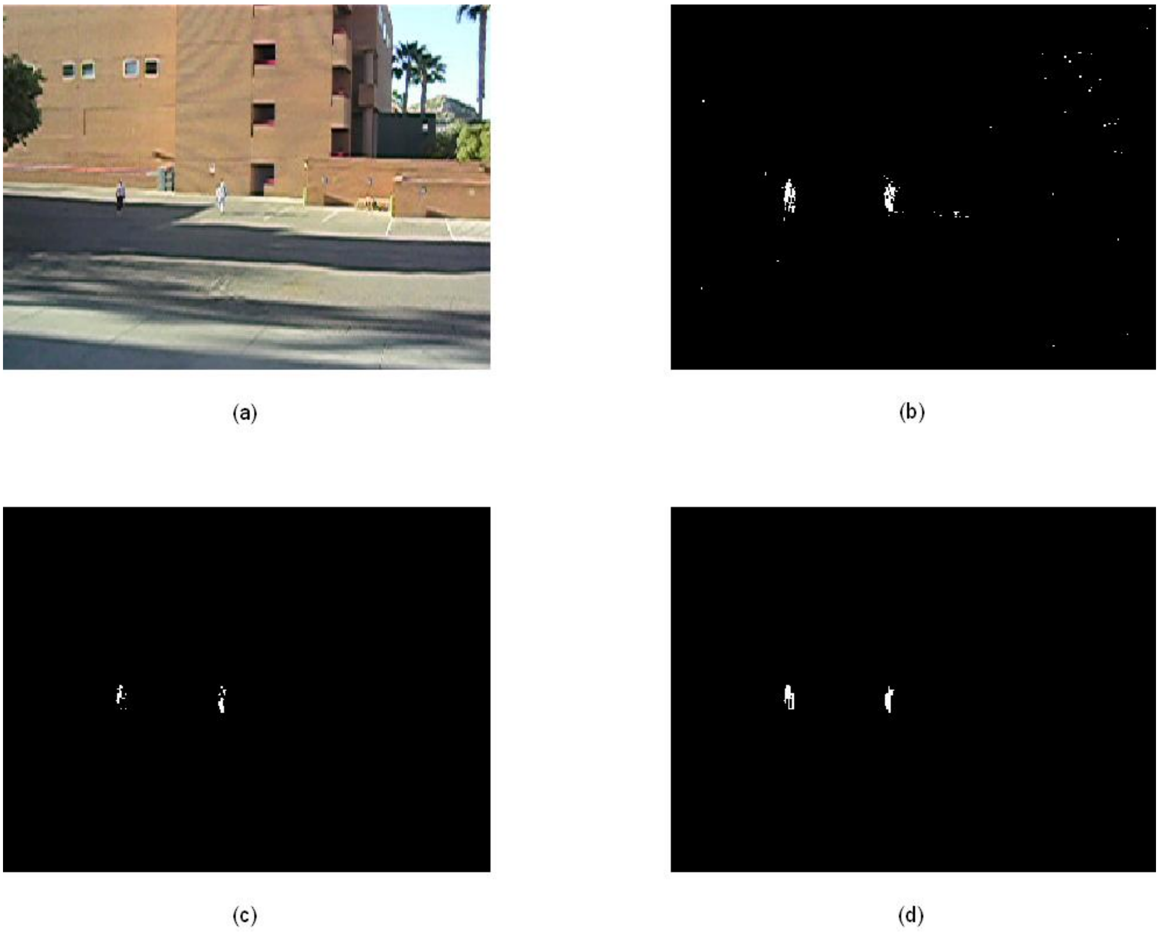

Figure 10. 2D tracking results: (a) original frame; (b) detected foreground after thresholding the difference between the image in (a) from the background model, effects of trees and shadows are present; (c) foreground after erosion; and (d) foreground after interconnecting blobs that are close enough.

The 2D tracking information after getting an appropriate cross correspondence using the improved fundamental matrix, combined with the information obtained from the internal and external camera calibration procedures is used to determine the 3D tracks. Given the projective matrices *P* and *P'* which were computed in Section 3, and the information about the 2D tracking from each camera, a triangulation method was used to define the 3D position of the objects (Hartley *et al.*, 1992; Hartley and Sturm, 1995). The information provided by the 2D tracking after establishing cross correspondences does not provide matches between the 2 camera image planes that satisfy $\mathrm{x}F\mathrm{x}'=0$. Therefore, for every pair of correspondent points in the 2 camera images $\mathrm{x}, \leftrightarrow \mathrm{x}'$, a new pair of points $\hat{\mathrm{x}}, \leftrightarrow \hat{\mathrm{x}}'$ is found, (see Fig. 13). The points $\hat{\mathrm{x}}$ and $\hat{\mathrm{x}}'$ are the pair of points that minimize the following:

$$d(\mathrm{x},\hat{\mathrm{x}})^2 + d(\mathrm{x}',\hat{\mathrm{x}}')^2 \tag{4.6}$$

under the epipolar constraint given by:

$$\mathrm{x}\, F\, \mathrm{x}' = 0 \tag{4.7}$$

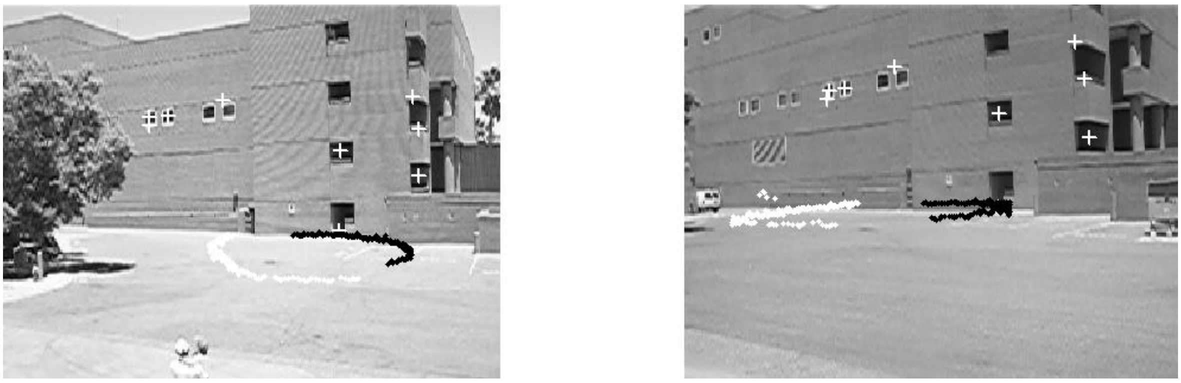

Figure 11. Matching features used to compute the fundamental matrix, ‘+’ represent the matching features obtained from the WBM scheme. ‘•’ represent the 2D tracking cross correspondences obtained by using 2D homographies.

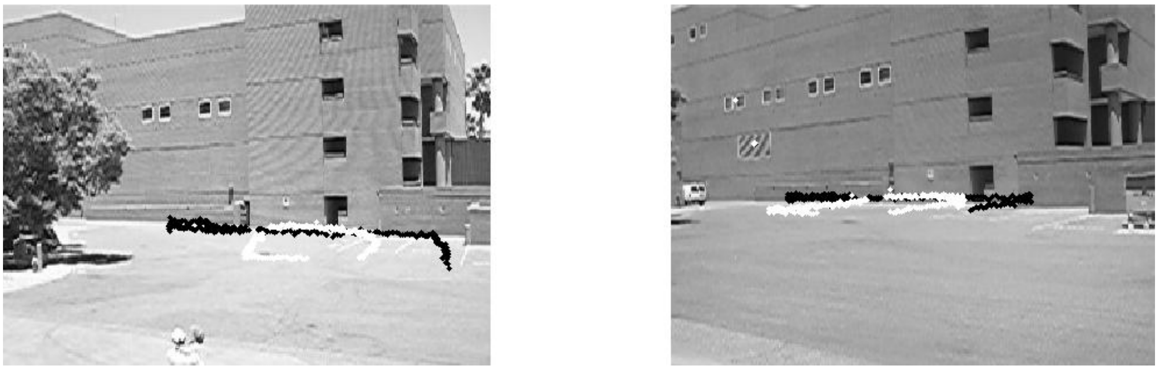

Figure 12. Cross correspondence established between moving objects by means of the computer fundamental matrix including information of the 2D cross correspondences.

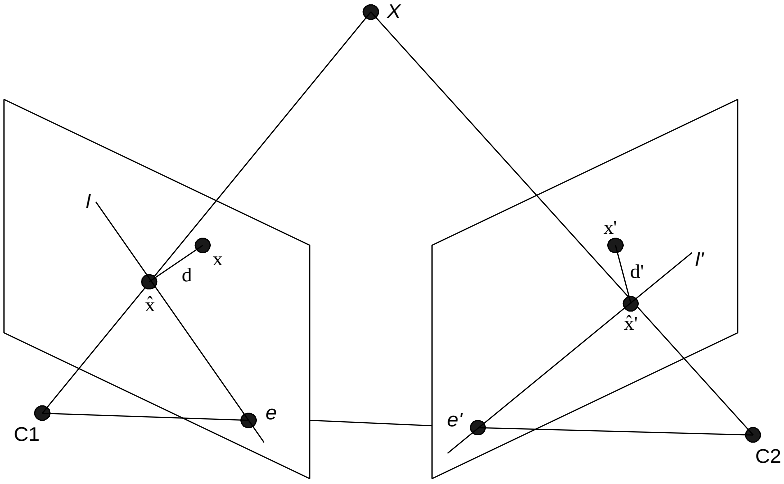

Figure 13. A pair of approximate correspondences $\mathrm{x}, \leftrightarrow \mathrm{x}'$ and the closest pair $\hat{\mathrm{x}}, \leftrightarrow \hat{\mathrm{x}}'$ that satisfy the epipolar constraints.

In Eq. (4.6) *d(.,.)* represents the Euclidean distance. Note that $d\left(x,\hat{x}\right)$ and $d\left(x',\hat{x}'\right)$ in Eq. (4.6) are the distances from the detected points in the 2D tracking to the epipolar lines *l* and *l'*, respectively. The minimization problem is reduced to finding and evaluating the roots of a polynomial of 6th degree. A linear triangulation method is used with the new resulting pair of correspondences.

Every point in an image plane can be mapped to a line that contains the camera's center of projection and the point given by:

$$X_R = P^+ x \tag{4.8}$$

where X_R is known as the reprojection of the point in the image plane *x*, and P^+ is the pseudo-inverse of the projection matrix *P* and is given by:

$$P^+ = P^T\left(P\,P^T\right)^{-1} \tag{4.9}$$

Given the center of projection for both cameras, *C* and *C'*, and the points $\hat{\mathrm{x}}$ and $\hat{\mathrm{x}}'$ in the image plane of each camera, two three-dimensional lines are defined, and finally the middle point of the smallest segment that joints both lines is considered as the 3D position of the object with projections x and x' in the image plane of camera *C* and *C'*, respectively.

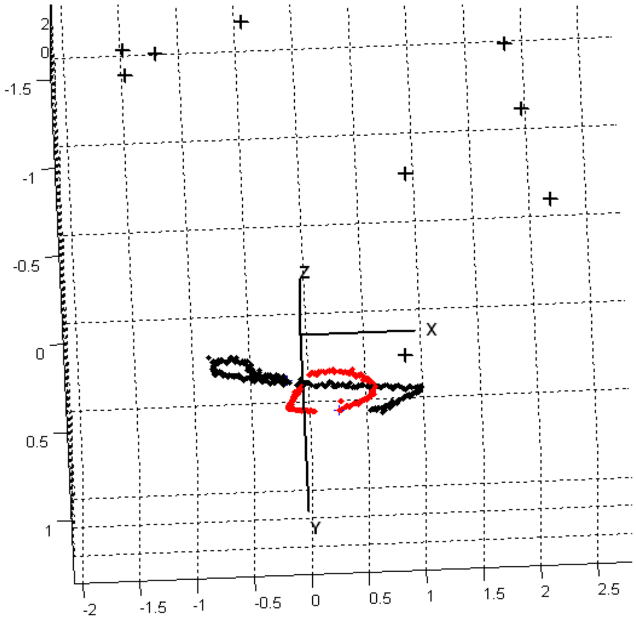

Figure 14. Superior view of the 3D tracking results. The depth of the WBM features as well as the depth of the 3D tracks agree with the experimental setup.

3D tracking results

The 3D tracking of the moving objects shown in Fig. 11 as well as the 3D position of the feature matching obtained from the WBM scheme shown in Fig. 12 are shown in the Figs. 14 and 15. In the experiment the distance between cameras was approximately 7,2 meters and the distance from the base of camera *C* to the base of the building in which most of the WBM matching features were detected was approximately 44 meters. Then, considering that the scale factor for the 3D results corresponds to the distance between the center of projection of both cameras, it was concluded that the depth (Z coordinate) of the WBM features agree with the reality. Additionally, the shape of the 3D tracks agrees with the objects' movement and finally it can be seen that the tracks are contained in a plane (the floor).

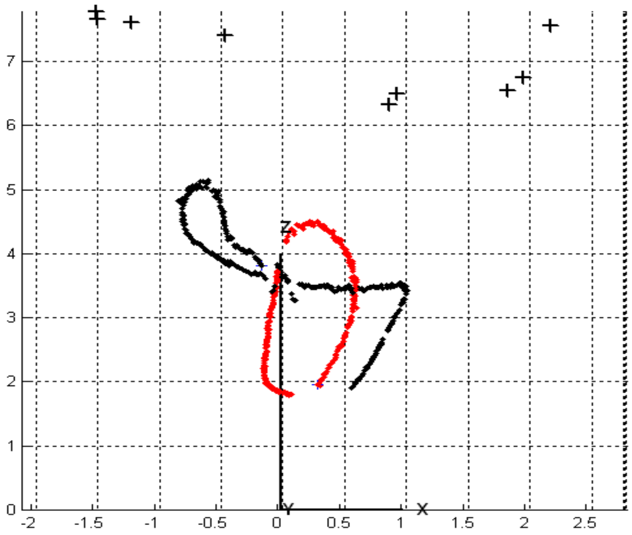

Figure 15. Frontal view of the scene where the moving objects where detected, WBM matching features are represented with crosses. The center of coordinate system is on the center of the projection of the camera.

5. CONCLUSIONS

This work presents contributions in the area of video surveillance and monitoring (VSAM) technology. A 3D tracking systems that exploits redundant information in a multiple-camera system, has been implemented. A semi-autonomous 3D tracking system was implemented. The system requires a minimum of 2 video cameras with a common volume under surveillance, which is a very common situation in VSAM systems. The proposed system can operate even when the separation between video sensors is wide. For this purpose a wide baseline matching scheme is used in the camera calibration process, without the need for pre-measured landmarks.

Further research shall be carried out in order to improve the distinctiveness of the SIFT descriptors. In Ke and Sukthankar (2004) an improvement to the SIFT descriptors based on Principal Component Analysis (PCA), instead of the smoothed weighted histograms of SIFT, is presented. Another improvement to the robustness and distinctiveness of SIFT using PCA is described in Mikolajczyk and Schmid (2003). The method is called Gradient-Location Orientation Histogram (GLOH). Additionally, a 2D tracking system, that is able to run in real-time and handle properly small camera movements (due to wind in outdoor cameras), is required to reduce the high computation and memory requirements of most existing methods for background subtraction. Finally, research is needed to reduce the bandwidth requirements of video streams when transmitted, either wired or wireless, to a central processing unit. This will consists in developing a low-power and low-complex 2D tracking system, that can be incorporated to the video sensors. In this way, only 2D positions need to be transmitted to the central processing unit.

ACKNOWLEDGEMENTS

The author is very grateful to Dr. Gang Qian for making available the MATLAB® implementation of the structure from motion method, as presented in Qian and Chellappa (2004).

REFERENCES

Agapito, L., E. Hayman, L. Reid, 2001. Self-Calibration of Rotating and Zooming Cameras. *Int. J. Comput. Vision*, 45(2), 1-23.

Baumberg, A., 2000. Reliable feature matching across widely separated views. *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, p. 774-781.

Bouguet, J.Y. (n.d.). Camera calibration toolbox for Matlab. Retrieved the 15th of October 2004 from *http://www.vision.caltech.edu/bouguetj/calib_doc/.*

Canny, J., 1986. A computational approach to edge detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, p. 679-698.

Collins, R., A. Lipton, T. Kanade, H. Fujiyoshi, D. Duggins, T. Tsin, D. Tolliver, N. Enomoto, O. Hasegawa, P. Burt, L. Wixson, 2000. A System for Video Surveillance and Monitoring: VSAM. *Robotics Institute, Carnegie Mellon University*, Pittsburgh, Final Report, 69 pp.

Cottier, J.C., 1994. Extraction et appariements robustes des points d'intérèt de deux images non étalonnées. Internal Report, *LIFIA-IMAG-INRIA*, Rhône-Alpes, Grenoble, France.

Crowley, J., R. Stern, 1982. Fast computation of the difference of low-pass transform. *Robotics Institute, Carnegie-Mellon University*, Pittsburgh, PA, USA, 42 pp.

Dufournaud, Y., D. Schmid, R. Horaud, 2000. Matching images with different resolutions. *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, p. 612-618.

Elgammal, A., R. Duraiswami, D. Harwood, L.S. Davis, 2002. Background and foreground modeling using nonparametric kernel density estimation for visual surveillance. *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, p. 1151-1163.

Faugeras, O., Q. Loung, S. Maybank, 1992. Camera self-calibration: Theory and experiments. *Proc. 2nd European Conf. on Computer Vision*, pp. 321-334.

Fischler, A., R. Bolles, 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 381-395.

Freeman, W., E. Adelson, 1991. The design and use of steerable filters. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, p. 891-906.

Harris, C., M. Stephens, 1988. A combined corner detector. *4th Alvey Vision Conf.*, p. 147-151.

Hartley, R., 1994. Self-calibration from multiple views with a rotating camera. *Proc. 3rd European Conf. on Computer Vision*, p. 471-478.

Hartley, R., 1997. In defense of the eight-point algorithm. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, p. 580-593.

Hartley, R., P. Sturm, 1995. Triangulation. *6th Int. Conf. on Computer Analysis of Images and Patterns*, p. 190-197.

Hartley, R., A. Zisserman, 2003. Multiple View Geometry in Computer Vision. *Cambridge University Press*, Cambridge, UK, 655 pp.

Hartley, R., R. Gupta, T. Chang, 1992. Stereo from uncalibrated images. *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, p. 761-764.

Heitger, F., L. Rosenthaler, R. Von Der Heydt, E. Peterhans, O. Kuebler, 1992. imulation of neural contour mechanism: From simple to end-stopped cells. *Vision Res.*, 963-981.

Horaud, R., T. Skordas, F. Veillon, 1990. Finding geometric and relational structures in an image. *Proc. 1st European Conf. on Computer Vision*, p. 374-384.

Jiangjian, X., M. Shah, 2003. Two frame wide baseline matching. *9th IEEE Int. Conf. on Computer Vision*, p. 603-609.

Ke, Y., R. Sukthankar, 2004. PCA-SIFT: a more distinctive representation for local image descriptors. *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, p. 506-513.

Kim, H., K.S. Hong, 2000. A practical self-calibration method of rotating and zooming cameras. *15th Internat. Conf. on Pattern Recognition*, p. 354-357.

Koenderink, J., A. Van Doorn, 1987. Representation of local geometry in the visual system. *Biol. Cybern.*, 367-375.

Lee, L., R. Romano, G. Stein, 2000. Monitoring activities from multiple video streams: Establishing a common coordinate Frame. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, p. 758-767.

Lindeberg, T., 1993. Scale-Space Theory in Computer Vision. *Kluwer Academic Publishers*, 423 pp.

Liu, J.S., R. Chen, 1998. Sequential monte carlo methods for dynamic systems. *J. Am. Statistic Assoc.*, 1032-1044.

Lo, B., S. Velastin, 2001. Automatic congestion detection system for underground platforms. *Intern. Symp. on Intelligent Multimedia, Video and Speech Processing*, p. 158-161.

Lowe, D., 1990. Object recognition from local scale-invariant features. *Proc. Int. Conf. on Computer Vision*, p. 1150.

Lowe, D., 2001. Local feature view clustering for 3D object recognition. *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, p. 682-688.

Lowe, D., 2004. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 91-110.

Ma, Y., S. Soatto, J. Kosecka, S. Shankar, 2004. *An invitation to 3-D vision from images to geometric models.* Springer-Verlag, NY, USA, 526 pp.

Mikolajczyk, K., S. Schmid, 2003. A performance evaluation of local descriptors. *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, p. 257-263.

Mikolajczyk, K., S. Schmid, 2004. Scale & affine invariant interest point detectors. *Int. J. Comput. Vision*, 63-86.

Mikolajczyk, K., S. Schmid, 2011. Indexing based on scale invariant interest point detector. *Proc. 8th IEEE Int. Conf. on Computer Vision*, p. 525-531.

Pritchett, P., A. Zisserman, 1998. Wide baseline stereo matching. *Int. Conf. on Computer Vision*, p. 754-760.

Qian, G., R. Chellappa, 2004. Structure from motion using sequential Monte Carlo methods. *Int. J. Comput. Vision*, 5-31.

Schaffalitzky, F., A. Zisserman, 2002. Multi-view matching for unordered image sets. *Proc. European Conf. on Computer Vision*, p. 414-431.

Schmid, C., R. Mohr, C. Bauckhage, 2002. Evaluation of interest point detectors. *Int. J. Comput. Vision*, 37(2), 151-172.

Stauffer, C., W. Grimson, 1999. Adaptive background mixture of models for real-time tracking. *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, p. 246-252.

Van Gool, L., T. Moons, D. Ungureanu, 1996. Affine / photometric invariants for planar intensity patterns. *Proc. European Conf. on Computer Vision*, p. 642-651.

Wren, C., A. Azarbayejani, T. Darrell, P. Pentland, 1997. Pfinder: Real-time tracking of the human body. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 6 pp.

Zitova, B., J. Flusser, 2003. Image registration methods: a survey. *Image Vision Comput.*, 977-1000.