Eurographics Italian Chapter Conference (2007)
Raffaele De Amicis and Giuseppe Conti (Editors)

Marker-less real-time camera registration for Mixed Reality

A. Liverani and S. Grandi[1]

[1]DIEM - University of Bologna, 40136 Bologna, Italy

Abstract

A real-time and robust algorithm for 3D camera registration in a Mixed Reality (MR) environment is described in this paper. The most used technique for camera pose (position and orientation with respect to a fixed or moving object) is based on fiducial marker tracking. This method guarantees good results in real-time with a single camera, but needs several high contrast printed markers on external world in order to make possible the calculation of camera parameters and positioning. Thus real 3D geometric data are grabbed only through already known markers. The aim of this research is a real-time monocular camera tracking and registration through automatic image features extraction from video streaming. The first implementation of the method, several examples and confrontation with non interactive algorithm for SFM (Structure From Motion) have demonstrated that this meets the real-time response and sufficient precision needed by a Mixed Reality environment.

Categories and Subject Descriptors (according to ACM CCS): I.3.6 [Computer Graphics]: Methodology and Techniques

1. Introduction

Mixed Reality (MR) is a field of computer graphics research which deals with the combination of real world and computer generated data. With this technique, a real-time overlay of real environment and virtual objects is achieved. According with several definitions available in literature [Azuma97] and [Azuma01] the most important aspects of MR are:

- the combination of virtual objects in a real environment;
- the environment interactivity, elsewhere called real-time response;
- the alignment (registration) of real and virtual objects together.

More in detail, real-time means a natural reaction of the synthetic environment under the human requests. A standard Mixed Reality system may be summarized in a combination of software and hardware devices listed below (please see also Figure 1):

- a HMD (Head Mounted Display) with see-through optical option;
- a camera installed on the HMD in order to synchronize the human and synthetic visualization;
- a computer based graphics system running either software for rendering virtual objects or software for image processing of the video stream interactively coming from the camera.

The image analysis and processing provides the camera calibration and registration needed to synchronize real and virtual world (camera tracking).

Several researches have demonstrated the benefits of Mixed Reality application in several fields, like:

- Medical: doctors use MR as a visualization and training aid for surgery;
- Manufacturing: MR techniques are useful in the assembly, maintenance and repairing of complex machinery;
- Robot path planning in order to optimize the robot motion and improving its grabbing capabilities;
- Annotation and visualization: MR can be used to annotate objects and overlay real world with directions or public informations;
- Interior design and modeling for interactive and immersive simulation of the designed environment;
- Military training and also Entertainment.

1.1. Related work

A big deal of work has been done in the vision-based camera tracking targeted not only to Virtual Reality and Mixed Re-

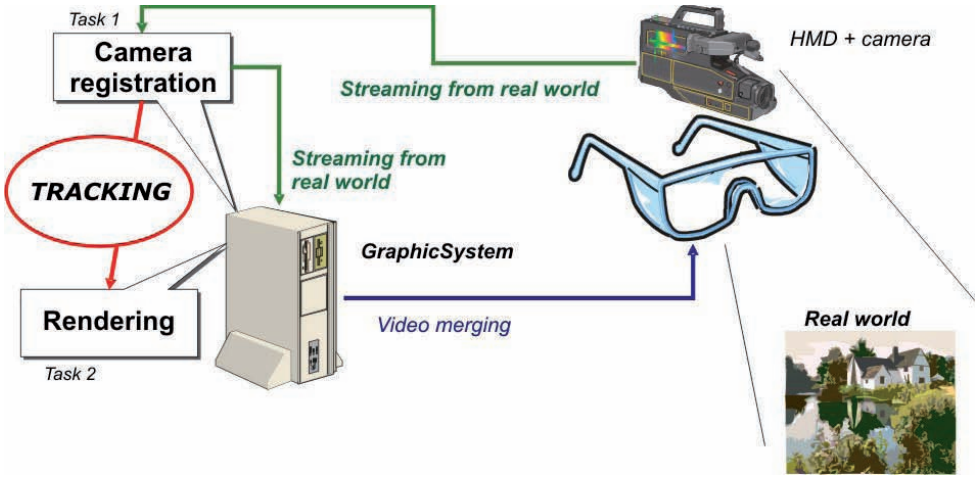

Figure 1: *A video based Mixed Reality system.*

ality environment, but also to robotic computer vision. One of the best classification activity in the VR and MR field is due to Milgram [Milgram94], who focused the attention on MR visual displays. Azuma [Azuma97] published one of the most complete surveys on Augmented Reality (AR) techniques, describing the characteristics of the primary systems with their applications. The same author completed the previous article adding recent developments and further applications ([Azuma01]). In these papers it is clear that the camera tracking appears as a critical aspect in all AR and MR systems, greatly influencing the final performances. Many authors have developed algorithms for camera calibration and registration, like Uenohara [Uenohara95], whose method performed object registration by a combination of template matching, feature detection and pose calculation of objects from feature positions in the image. A similar method was proposed by Grimson [Grimson95], who computed camera and object positions separately and applyed this technique in medicine. However, most of the methods actually used for camera tracking are based on fiducial markers (usually, squares of known size with high contrast patterns in their centers) located in the 3D environment, as it is described in [Kato99]. Work closely related to this approach is also described in [Koller97], where a method for accurately tracking the 3D motion of a camera in a known 3D environment is proposed, by estimating the camera location in a dynamical way with automatic feature selection. Kutulakos [Kutulakos96], without using any metric information about the calibration parameters of the camera, used artificial markers for tracking the camera motion, implementing a video-based AR system where the user had to interactively select at least four no-coplanar points to obtain the desired values. Trying to avoid the usage of markers, the Structure From Motion (SFM) methodology recovers the 3D geometry from a couple of two consequent images. Broida [Broida90] developed a method based on SFM techniques using recursive algorithms to estimate the object position in the scene following several 2D features on the image sequence. Also [Shariat90] and [Roach80] developed algorithms for camera position and orientation calculation starting from two shifted images. Bolles and Baker [Bolles85], instead, analyzed a dense sequence of images (shot at high frequency), that simulate a stereo vision with a single camera.

1.2. Objective

This paper reports the implementation and test of a software dedicated to camera registration and extraction of real features coordinates for 3D tracking. This way virtual and real worlds can be aligned and syncronized for a correct MR session. The 3D tracking is based on two different activities: the camera calibration and a real-time SFM (Structure From Motion) algorithm, as described in the following sections.

2. Camera calibration stage

The camera calibration is normally an offline activity targeted to characterize virtual camera parameters in order to make a correct mathematical model of a real camera. This model is based on the calibration matrix $\mathbf{K}$, which is 3×3 upper triangular and has the following form:

$$\mathbf{K} = \begin{pmatrix} \alpha_x & s & x_0 \\ 0 & \alpha_y & y_0 \\ 0 & 0 & 1 \end{pmatrix}$$

where:

- α_x and α_y are the focal lengths along the x-axes and y-axes;

- $\mathbf{P_0} = [x_0, y_0]$ is the principal point, which is the projection center on the image plane ;
- $s = \tan\theta$, where θ is the angle between the x and y axes of the image, is the skew coefficient .

The external parameters refer to position and orientation of the real MR camera, simulated as a pin-hole camera, with respect to the world reference system. The six parameters are computed starting from the **t** vector and the **R** matrix:

- three parameters for the translation vector $\mathbf{t} = [t_x, t_y, t_z]$;
- three parameters for the matrix rotation

$$\mathbf{R} = [C1\ \ C2\ \ C3]$$

with:

$$C1 = \begin{pmatrix} Cos(\beta)\,Cos(\gamma) \\ -Cos(\beta)\,Sin(\gamma) \\ Sin(\beta) \end{pmatrix}$$

$$C2 = \begin{pmatrix} Sin(\alpha)\,Sin(\beta)\,Cos(\gamma) + Cos(\alpha)\,Sin(\gamma) \\ -Sin(\alpha)\,Sin(\beta)\,Sin(\gamma) + Cos(\alpha)\,Cos(\gamma) \\ -Sin(\alpha)\,Cos(\beta) \end{pmatrix}$$

$$C3 = \begin{pmatrix} -Cos(\alpha)\,Sin(\beta)\,Cos(\gamma) + Sin(\alpha)\,Sin(\gamma) \\ Cos(\alpha)\,Sin(\beta)\,Sin(\gamma) + Sin(\alpha)\,Cos(\gamma) \\ Cos(\alpha)\,Cos(\beta) \end{pmatrix}$$

In the calibration stage a printed marker is projected and acquired by the MR internal camera. So we can provide at least n corresponding points $(\mathbf{x}_i, \mathbf{X}_i), i = 1,..,n$, where $\mathbf{X}_i$ is a point on the scene and $\mathbf{x}_i$ is its image on the projection plane. The calibration algorithm for the camera is made by two fundamental steps:

1. compute the matrix $\mathbf{P} = \mathbf{K}[\mathbf{R}|\mathbf{t}]$ such that $\mathbf{x}_i = \mathbf{P}\mathbf{X}_i$
2. compute the QR decomposition of $\mathbf{P}$ to obtain $\mathbf{K}, \mathbf{R}, \mathbf{t}$.

From each relation $\mathbf{x}_i = \mathbf{P}\mathbf{X}_i$ there are two equations:

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

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

Multiplying, these equations become:

$$x_i(p_{31}X_i + p_{32}Y_i + p_{33}Z_i + p_{34}) = p_{11}X_i + p_{12}Y_i + p_{13}Z_i + p_{14}$$

$$y_i(p_{31}X_i + p_{32}Y_i + p_{33}Z_i + p_{34}) = p_{21}X_i + p_{22}Y_i + p_{23}Z_i + p_{24}$$

Rewriting the previous equations, where the unknown variables are the entries of the matrix **P**, the system becomes:

$$\begin{pmatrix} X & Y & Z & 1 & 0 & 0 & 0 & 0 & -xX & -xY & -xZ & -x \\ 0 & 0 & 0 & 0 & X & Y & Z & 1 & -yX & -yY & -yZ & -y \end{pmatrix} \mathbf{p} = 0 \quad (1)$$

where

$$\mathbf{p} = (p_{11}, p_{12}, p_{13}, p_{14}, p_{21}, p_{22}, p_{23}, p_{24}, p_{31}, p_{32}, p_{33}, p_{34})^T$$

The Equation System 1 for $i = 1,...,n$ has $2n$ equations in 12 unknown variables of the form $\mathbf{Lp} = 0$, for an appropriate matrix **L**. Using the Singular Value Decomposition and the bundle adjustment process ([Golub70]), an approximate solution for the value of **P** may be found. The first 3×3 submatrix of **P**, called **M**, is the product of **K** and **R**

$$\mathbf{M} = \mathbf{KR}$$

and, exploiting the QR decomposition of **M**, the rotation matrix and the calibration matrix will be calculated. Finally the translation vector **t** is computed as

$$\mathbf{t} = \mathbf{K}^{-1}(p_{14}, p_{24}, p_{34})^T$$

3. Structure From Motion (SFM)

The Structure From Motion indicates a general problem of recovering 3D geometry from 2D geometry. This reconstruction method is based on the bundle adjustment process of minimizing the distances between estimated 3D structure projections and actual image measurements. Once the 2D projection of a point in the real scene has been found, its position in 3D can be assumed somewhere along the ray connecting the camera optical center and the corresponding spot in the image plane. Tracking its projections along multiple images and using triangulation allows the localization of the 3D point. This part of the algorithm is related to epipolar geometry and will be described in the following section. If extraction and correspondence can be performed for a sufficient number of points and lines and over images acquired from different directions, then the camera position and orientation may be calculated. Finally also the 3D coordinates (in world reference system) of the real geometric features location can be deduced.

3.1. Epipolar geometry

Any two perspective views are related by the epipolar geometry, which allows to determine one camera position with respect to the other. Epipolar geometry consists on applying projective geometry techniques in Computer Vision. Given two images $\mathbf{I_j}$ and $\mathbf{I_{j+1}}$, taken from the same camera in different positions, the fundamental 3×3 matrix **F** establishes the relation between an epipolar line on $\mathbf{I_{j+1}}$ and the corresponding point on $\mathbf{I_j}$.

The epipolar plane is the plane which contains a point **P** and the two centers of projection of the camera in the two different positions. The epipolar line is defined as the intersection between the epipolar plane and the image plane. According to [Luong97], the fundamental matrix has the following expression, up to a scale factor:

$$\mathbf{F} \cong \mathbf{K}^{-T}\mathbf{NRK}^{-1} \quad (2)$$

where **R** and **t** are the relative rotation and translation between the two camera positions and **N** is the skew matrix

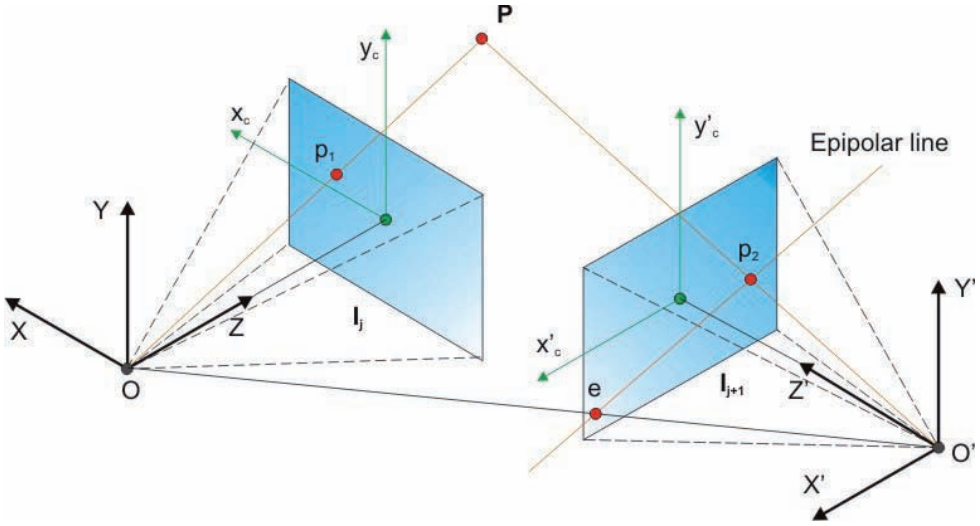

Figure 2: *Relation between the point* $\mathbf{p_1}$ *on* $\mathbf{I_j}$ *and the corresponding epipolar line* l'_1 *on* $\mathbf{I_{j+1}}$.

associated to the vector $\mathbf{t}$:

$$\mathbf{N} = \begin{pmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{pmatrix}$$

Suppose that n pairs of corresponding points in the two images are known. For each pair of corresponding points $(\mathbf{x}, \mathbf{x}')$ in the two images, the following epipolar equation is valid:

$$\mathbf{x}'^T \mathbf{F}\mathbf{x} = 0$$

The epipoles $\mathbf{e}$ and $\mathbf{e}'$, respectively of $\mathbf{I_j}$ and $\mathbf{I_{j+1}}$, are the eigenvectors of $\mathbf{F}$ corresponding to the eigenvalue zero:

$$\mathbf{F}\mathbf{e} = 0 \qquad \mathbf{F}\mathbf{e}' = 0$$

Using the epipolar equation applied to all the n corresponding points, an homogeneous system whose solution is the matrix $\mathbf{F}$, always up to a scale factor, is obtained. This system can be rewritten in the form

$$\mathbf{B}\mathbf{X} = 0$$

For an appropriate matrix $\mathbf{B}$, it is usually an overdetermined system. Through the Singular Value Decomposition [Golub70], the system can be solved and $\mathbf{F}$ can be determined. We remark that this matrix depends only on the rotation and translation of the camera between the two positions and we assume that the calibration matrix $\mathbf{K}$ is unchanging under camera motions.

3.2. Camera registration

Starting from the first camera registration data, calculated in the previous section thanks to Equation 2, the software has to extract interactively the camera position and orientation during the real video-camera movement. The Dornaika's method [Dornaika01] is appropriate because it uses the results of Equation 2 and performs an efficient solution of the following equation:

$$\mathbf{K}^T\mathbf{F}\mathbf{K} = \mathbf{N}\mathbf{R}$$

Due to the orthogonality of matrix $\mathbf{R}$, that is $\mathbf{R}\mathbf{R}^T = \mathbf{I}$, the previous inequality can be rewritten:

$$\mathbf{K}^T\mathbf{F}\mathbf{K}\mathbf{R}^T = \mathbf{N} \tag{3}$$

Calling the matrix $\mathbf{K}^T\mathbf{F}\mathbf{K}\mathbf{R}^T = \mathbf{A} = \{a_{ij}\}, 1 \leq i, j \leq 3$, the equality $\mathbf{A} = \mathbf{N}$ can be imposed, obtaining some conditions for the entries of matrix $\mathbf{A}$:

$$a_{11} = a_{22} = a_{33} = a_{12} + a_{21} = a_{13} + a_{31} = a_{23} + a_{32} = 0 \tag{4}$$

By representing the rotation matrix $\mathbf{R}$ through its associated unit quaternion $\mathbf{q} = (q_0, q_x, q_y, q_z)^T$, the matrix $\mathbf{R}$ can be rewritten as:

$$\mathbf{R} = \begin{pmatrix} q_0^2 + q_x^2 - q_y^2 - q_z^2 & 2(q_x q_y - q_0 q_z) & 2(q_x q_z + q_0 q_y) \\ 2(q_x q_y + q_0 q_z) & q_0^2 - q_x^2 + q_y^2 - q_z^2 & 2(q_z q_y - q_0 q_x) \\ 2(q_x q_z - q_0 q_y) & 2(q_z q_y + q_0 q_x) & q_0^2 - q_x^2 - q_y^2 + q_z^2 \end{pmatrix}$$

Calling $\mathbf{v} = (a_{11}, a_{22}, a_{33}, a_{12} + a_{21}, a_{13} + a_{31}, a_{23} + a_{32})^T$, the constraints in Equation 4, associated with one single pair of images (i. e. one motion of the camera), can be expressed now in a more compact way:

$$\mathbf{v} = 0$$

Consider now n fundamental matrices and define the error function f, that will be minimized over the unknown variables, as:

$$f(\alpha_x, \alpha_y, x_0, y_0, \mathbf{q}_1, \ldots, \mathbf{q}_n) = \sum_{i=1}^{n} ||v_i||^2 + \lambda(1 - ||q_i||^2)^2 \tag{5}$$

where $\alpha_x, \alpha_y, x_0, y_0$ are the internal parameters of the camera, λ is a positive real number and $s = 0$, that is the image axes are orthogonal. As it is stated in [Dornaika01], the value of λ is set to 10^3. This is a nonlinear least squares constrained minimization problem and it can be solved by applying standard nonlinear optimization techniques [Fletcher]. If the calibration matrix $\mathbf{K}$ is considered as an unknown ([Abdullah02]), Problem 5 may be solved starting from initial approximated values of $\mathbf{K}$, $\mathbf{N}$ and $\mathbf{R}$ and then achieving the self-calibration of the camera. However, in Section 2 the unchanging value of matrix $\mathbf{K}$ has been already computed, so the calibration problem becomes a registration problem. Starting from the value of $\mathbf{K}$ computed in Section 2, the error function f has only $\mathbf{q}_1, ..., \mathbf{q}_n$ and the entries of matrix $\mathbf{F}$ as unknowns and the nonlinear least squares constrained minimization problem can be solved as showed in [Dornaika01].

3.3. Matrix accumulation

The camera position and orientation at time τ_j are given by the solution of Equation 3 for the couple of images captured at τ_{j-1} and τ_j, that means frames $\mathbf{I}_{j-1}$ and $\mathbf{I}_j$. These transformation matrices are calculated with respect to a coordinate system aligned with camera in $\mathbf{I}_{j-1}$. However the current camera registration is resulting by the combination of each rotation matrix $\mathbf{R}_j$ and translation vector $\mathbf{t}_j$ from frame $\mathbf{I}_0$ up to frame $\mathbf{I}_N$. So a cumulative vector for translation can be rewritten as:

$$\mathbf{t}_N = \mathbf{t}_0 + ... + \mathbf{t}_{N-1}$$

and a cumulative matrix for rotation as:

$$\mathbf{R}_N = \mathbf{R}_0 \cdot ... \cdot \mathbf{R}_{N-1}$$

4. Matching of the corresponding points

4.1. Visual feature extraction and correlation between images

The method described in Section 3.1 stands on the assumption of knowing n corresponding points $(\mathbf{x}_i, \mathbf{x'}_i)$, for $i = 1, ..., n$, respectively in the two images $\mathbf{I_{j-1}}$ and $\mathbf{I_j}$. In the real world, corresponding points are visual features which have to be extracted and calculated in real-time by a very efficient set of computations. Two points belonging to two different images $\mathbf{I_{j-1}}$ and $\mathbf{I_j}$ are conjugated if they are the projection of the same 3D point on the scene. The vector corresponding to the difference between a pair of conjugated points, when the two images are superimposed, is called disparity vector. These conjugated points are characterized by a particular light intensity and they are found using the corner method ([Fang82] and [Mokhtarian98]). According to this method, a point on the image is classified thanks to its directional variation:

- it is a plane point if there is no variation;
- it is an edge point if there is a variation along one direction;
- it is a corner point if the variations are along all directions.

Let $\mathbf{P}(u, v)$ be a point on the image, let $\mathbf{W}$ be a neighborhood of $\mathbf{P}$ and $\mathbf{d} \in \mathbf{W}$. The following function can be defined:

$$\mathbf{E}_h(\mathbf{P}) = \sum_{\mathbf{d} \in \mathbf{W}} [\mathbf{I}(\mathbf{P}+\mathbf{d}) - \mathbf{I}(\mathbf{P}+\mathbf{d}+\mathbf{h})]^2 \tag{6}$$

This function computes the variation of brightness between two points of $\mathbf{W}$ for a displacement $\mathbf{h}$, where $\mathbf{h}$ is the maximum limit of the computable disparity. Using truncated Taylor's series, $\mathbf{E}_h(\mathbf{P})$ has the following form:

$$\mathbf{E}_h(\mathbf{P}) = \sum_{\mathbf{d} \in \mathbf{W}} [\nabla \mathbf{I}(\mathbf{P}+\mathbf{d})^T \mathbf{h}]^2 =$$

$$= \sum_{\mathbf{d} \in \mathbf{W}} \mathbf{h}^T (\nabla \mathbf{I}(\mathbf{P}+\mathbf{d}))(\nabla \mathbf{I}(\mathbf{P}+\mathbf{d}))^T \mathbf{h} =$$

$$= \sum_{\mathbf{d} \in \mathbf{W}} \mathbf{h}^T \begin{pmatrix} \mathbf{I}_u^2 & \mathbf{I}_u \mathbf{I}_v \\ \mathbf{I}_u \mathbf{I}_v & \mathbf{I}_v^2 \end{pmatrix} \mathbf{h}$$

where $\nabla \mathbf{I}(\mathbf{P}+\mathbf{d}) = [\mathbf{I}_u \mathbf{I}_v]^T$. It is convenient to introduce a gaussian weight function $w(*)$ on $\mathbf{W}$ defined as:

$$w(\mathbf{d}) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-||\mathbf{d}||^2}{2\sigma^2}}$$

The Taylor's series can be now rewritten as:

$$\mathbf{E}_h(\mathbf{P}) = \mathbf{h}^T \begin{pmatrix} \sum_{\mathbf{d} \in \mathbf{W}} \mathbf{I}_u^2 \, w(\mathbf{d}) & \sum_{\mathbf{d} \in \mathbf{W}} \mathbf{I}_u \mathbf{I}_v \, w(\mathbf{d}) \\ \sum_{\mathbf{d} \in \mathbf{W}} \mathbf{I}_u \mathbf{I}_v \, w(\mathbf{d}) & \sum_{\mathbf{d} \in \mathbf{W}} \mathbf{I}_v^2 \, w(\mathbf{d}) \end{pmatrix} \mathbf{h} \tag{7}$$

Defining the 2×2 matrix $\mathbf{C}$, called the pseudo-hessian, as

$$\mathbf{C} = \begin{pmatrix} \sum_{\mathbf{d} \in \mathbf{W}} \mathbf{I}_u^2 \, w(\mathbf{d}) & \sum_{\mathbf{d} \in \mathbf{W}} \mathbf{I}_u \mathbf{I}_v \, w(\mathbf{d}) \\ \sum_{\mathbf{d} \in \mathbf{W}} \mathbf{I}_u \mathbf{I}_v \, w(\mathbf{d}) & \sum_{\mathbf{d} \in \mathbf{W}} \mathbf{I}_v^2 \, w(\mathbf{d}) \end{pmatrix} \tag{8}$$

Equation 7 can be rewritten in the following form:

$$\mathbf{E}_h(\mathbf{P}) = \mathbf{h}^T \mathbf{C} \, \mathbf{h}$$

$\mathbf{C}$ is symmetric and positive-defined, so it is diagonalizable and its eigenvalues are non-negative. In order to classify the point $\mathbf{P}$, the eigenvalues λ_1, λ_2 of $\mathbf{C}$ (for unitary displacements $\mathbf{h}$) have to be computed :

- if $\lambda_1 \approx \lambda_2 \approx 0$, $\mathbf{P}$ is a plane point;
- if $\lambda_1 \approx 0$ and $\lambda_2 > 0$, $\mathbf{P}$ is an edge point;
- if $\lambda_1 > 0$ and $\lambda_2 > 0$, $\mathbf{P}$ is a corner point.

For the practical computation of the eigenvalues, we refer to [Harris88]. Let $\lambda_1 > \lambda_2$; according to this method, $\mathbf{P}$ is a corner if:

$$\lambda_1 > 0 \qquad \frac{\lambda_2}{\lambda_1} \to 1$$

After having defined corner points on both images, the problem is to find all the pairs of conjugated points. The

most used likelihood function is the Normalized Cross-Correlation function NCC ([Klimenko01])

$$\mathbf{F}_{u,v}(x,y) = \frac{\sum_{i=-n}^{n}\sum_{k=-n}^{n} g_{j-1}\, g_j}{\sqrt{\sum_{i=-n}^{n}\sum_{k=-n}^{n} g_{j-1}^2 \;\; \sum_{i=-n}^{n}\sum_{k=-n}^{n} g_j^2}} \tag{9}$$

$$g_{j-1} = \mathbf{F_{j-1}}(x+k, y+i) - \mu_{j-1}$$

$$g_j = \mathbf{F_j}(x+k+u, y+i+v) - \mu_j$$

where $\mathbf{F_{j-1}}$ and $\mathbf{F_j}$ are now the two brightness functions on the two images, (x,y) is a point on the first image $\mathbf{I_{j-1}}$, (u,v) is a point on the second image $\mathbf{I_j}$ and μ_{j-1}, μ_j are the mean values of the neighborhood of (x,y) and (u,v) respectively. If $\mathbf{F_{u,v}}(\mathbf{x},\mathbf{y})$ is near 1, the two considered points are conjugated. In order to maximize the correlation between two images and find a point on $\mathbf{I_j}$ conjugated to the point (x,y) on $\mathbf{I_{j-1}}$, the maximum of $\mathbf{F}$ has to be found. Repeating these steps for all the corner points previously found, all the pairs of corresponding points may be calculated. Now that n pairs of conjugated points are known, the method described in Section 3.1 can be applyed to find the rotation matrix $\mathbf{R}$ and the translation vector $\mathbf{t}$.

4.2. NCC approximation

Consider now the Normalized Cross-Correlation function previously defined in Equation 9. A direct calculation of this value for a $M \times M$ window and a $N \times N$ feature requires $N^2(M-N+1)^2$ additions and $N^2(M-N+1)^2$ multiplications for the numerator and more than $3N^2(M-N+1)^2$ operations for the denominator ([Lewis95]). Denoting with $num(\mathbf{F}_{u,v})$ and $den(\mathbf{F}_{u,v})$ the numerator and the denominator of Equation 9 respectively, the idea introduced in this section is to compute separately these two quantities, approximating $den(\mathbf{F}_{u,v})$ in order to reduce its computational cost. Equation $num(\mathbf{F}_{u,v})$ is a convolution of the image with the reversed feature and it can be computed via a Fast Fourier Transform (FFT) algorithm; in this case the complexity can be reduced and if both N and M are large, N approaches to M, $12M^2 \log_2 M$ multiplications an $18M^2 \log_2 M$ addictions are needed. The quantity called $den(\mathbf{F}_{u,v})$ can be efficiently computed from tables containing the integral of the image $s(x,y)$ and image square $s^2(x,y)$ over the search area ([Lewis95]):

$$s(x,y) = \mathbf{F_{j-1}}(x,y) + s(x-1,y) + s(x,y-1) - s(x-1,y-1) \tag{10}$$

$$s^2(x,y) = \mathbf{F^2_{j-1}}(x,y) + s^2(x-1,y) + s^2(x,y-1) - s^2(x-1,y-1) \tag{11}$$

From Equations 10 and 11 the values of $\mathbf{F_{j-1}}(x,y)$ and $\mathbf{F^2_{j-1}}(x,y)$ can be computed and inserted in Equation 9; doing the same steps for the quantities $\mathbf{F_j}(x,y)$ and $\mathbf{F^2_j}(x,y)$ allows to determine the value of $den(\mathbf{F}_{u,v})$ expanding it into an expression involving only the image sum and sum squared. The construction of the tables requires approximately $3M^2$ operations. Thanks to these computational techniques, the global computational cost of the algorithm may be controlled and reduced.

4.3. 3D Reconstruction of the collimated points

At this stage, the camera location in frame $\mathbf{I_j}$ is notorious ($\mathbf{K}$ and $\mathbf{F}$ matrices are fixed upon real camera movements) and also 3D coordinates of most visual features (points) can be calculated, with respect to the world reference system, exploiting an inverse projection computation. Supposing that the world coordinate system coincides with the system of the first camera TC1, the projection matrix $\mathbf{P}_{j-1}$ which transforms an image point $\mathbf{x}_i$ belonging to the first image in a 3D point $\mathbf{X}_i$ has the following form:

$$\mathbf{P}_{j-1} = \mathbf{K}[\mathbf{I}|\mathbf{0}]$$

and the relation is

$$\mathbf{x}_i = \mathbf{P}_{j-1}\mathbf{X}_i \tag{12}$$

Analogously, the projection matrix $\mathbf{P}_j$ transforms an image point $\mathbf{x'}_i$ belonging to the second image into a 3D point $\mathbf{X}_i$:

$$\mathbf{P}_j = \mathbf{K}[\mathbf{R}|\mathbf{t}]$$

and the relation now is

$$\mathbf{x'}_i = \mathbf{P}_j\mathbf{X}_i \tag{13}$$

Knowing $\mathbf{P}_{j-1}$ and $\mathbf{P}_j$, the coordinates of the point $\mathbf{X}$ corresponding to the two projection points considered may be determined. Define now

$$\mathbf{P}_{j-1} = [\mathbf{c}_1 \;\; \mathbf{c}_2 \;\; \mathbf{c}_3]^T \qquad \mathbf{P}_j = [\mathbf{c'}_1 \;\; \mathbf{c'}_2 \;\; \mathbf{c'}_3]^T$$

Equation 12 can be rewritten in the following form:

$$\omega\,[x_i \;\; y_i \;\; 1]^T = [\mathbf{c}_1 \;\; \mathbf{c}_2 \;\; \mathbf{c}_3]^T\mathbf{X}_i \tag{14}$$

where ω stands for the homogeneous coordinate, and Equation 13 becomes:

$$\omega[x'_i \;\; y'_i \;\; 1]^T = [\mathbf{c'}_1 \;\; \mathbf{c'}_2 \;\; \mathbf{c'}_3]^T\mathbf{X}_i \tag{15}$$

Developing Equation 14, a system of three equations is obtained:

$$\begin{cases} \omega\, x_i = \mathbf{c}_1\, \mathbf{X}_i \\ \omega\, y_i = \mathbf{c}_2\, \mathbf{X}_i \\ \omega = \mathbf{c}_3\, \mathbf{X}_i \end{cases}$$

and it can be reduced to

$$\begin{cases} \mathbf{c}_3\, \mathbf{X}_i\, x_i = \mathbf{c}_1\, \mathbf{X}_i \\ \mathbf{c}_3\, \mathbf{X}_i\, y_i = \mathbf{c}_2\, \mathbf{X}_i \end{cases} \Rightarrow$$

$$\begin{cases} \mathbf{c}_3\,\mathbf{X}_i\,x_i - \mathbf{c}_1\,\mathbf{X}_i = 0 \\ \mathbf{c}_3\,\mathbf{X}_i\,y_i - \mathbf{c}_2\,\mathbf{X}_i = 0 \end{cases} \Rightarrow$$

$$\begin{bmatrix} \mathbf{c}_3\,x_i - \mathbf{c}_1 \\ \mathbf{c}_3\,y_i - \mathbf{c}_2 \end{bmatrix} \mathbf{X}_i = 0 \tag{16}$$

In the same way, starting from Equation 15 the final system is:

$$\begin{bmatrix} \mathbf{c'}_3\,x'_i - \mathbf{c'}_1 \\ \mathbf{c'}_3\,y'_i - \mathbf{c'}_2 \end{bmatrix} \mathbf{X}_i = 0 \tag{17}$$

Combining together Relations 16 and 17, the triangulation relation is:

$$\begin{bmatrix} \mathbf{c}_3\,x_i - \mathbf{c}_1 \\ \mathbf{c}_3\,y_i - \mathbf{c}_2 \\ \mathbf{c'}_3\,x'_i - \mathbf{c'}_1 \\ \mathbf{c'}_3\,y'_i - \mathbf{c'}_2 \end{bmatrix} \mathbf{X}_i = 0 \tag{18}$$

which appears as a linear homogeneous system and it can be solved using the Least Squares Method ([Abdi03]). For each pair of corresponding points on the two images, a 3D point $\mathbf{X}$ expressed in the world reference system can be determined. The unknown scale factor in Equation 2 is usually determined by using a known length on the scene and by observing its variation.

5. Implementation

5.1. Software Environment

In order to test the overall performances of the previous described algorithms, a C++ code has been developed and compiled both in standalone executable, and in Dynamic Link Library (DLL) form. The second choice has been introduced with the objective of replacing the marker based camera registration routine in ARToolKit [Kato99]. This approach was useful to evaluate the performances of camera registration by itself, keeping the same rendering effort.

5.2. Performance

In the marker based approach the images are processed one by one and the computational effort doesn't change significantly. The CPU load greatly depends upon the image resolution and camera frame rate. In this case, where the number of tracked image features changes along the sequence, the computing cost is variable. By forcing a limited set of features, the balance between a good recognition and a fast registration has been obtained. The average frame rate has been on 20 fps (frame per second) and the environment has never run at less than 12 fps with good camera tracking and 3D object reconstruction.

6. Conclusions

A real-time camera registration is a time consuming activity that exploits image processing routines and greatly influences the MR environment performances. Thus an interactive and marker-less software is considered an hard job on a standard PC. In this paper a robust and very efficient algorithm for this purpose has been developed and implemented. The method exposed, based on automatic image features extraction from video streaming with monocular vision, calculates camera position and orientation with respect to the world reference system at interactive frequency. The algorithm has been implemented in C++ programming language and compared with non interactive algorithm for Structure From Motion, obtaining a real-time response and similar precision in 3D objects coordinates recovery.

References

[Abdi03] H. ABDI: Least Squares. *Encyclopedia for research methods for the social sciences*, 2003, pp. 792-795.

[Abdullah02] J. ABDULLAH AND K. MARTINEZ: Camera Self-Calibration for the ARToolKit. *Proceedings of First International Augmented Reality Toolkit Workshop*, 2002, pp. 84-88.

[Azuma97] R.T., AZUMA: A survey of Augmented Reality. *Presence: Teleoperators and virtual environments*, 1997, vol. 6, pp. 355-385.

[Azuma01] R.T. AZUMA, Y. BAILLOT, R. BEHRINGER, S. FEINER, S. JULIER AND B. MACINTYRE: Recent advances in Augmented Reality. *Computer Graphics and Applications, IEEE Computer*, 2001, vol. 21, pp. 34-47.

[Behringer98] R. BEHRINGER, G. KLINKER AND D. MIZELL: Augmented Reality: Placing Artificial Objects in Real Scene. *IWAR, San Francisco*, 1998.

[Bolles85] R.C. BOLLES AND H.H. BAKER: Epipolar-plane image analysis: a technique for analyzing motion sequences. *IEEE 3rd Workshop on Computer Vision: Representation and Control*, 1985, pp. 168-178.

[Broida90] T.J. BROIDA, S. CHANDRASHEKHAR AND R. CHELLAPPA: Recursive 3D motion estimation from a monocular image sequence. *IEEE Trans. Aerospace and Electronic Systems*, 1990, vol.26, pp. 639-656.

[Dornaika01] F. DORNAIKA AND R. CHUNG: An algebraic approach to camera self calibration. *Computer Vision and Image Understanding*, 2001, pp. 195-215.

[Fang82] J.Q. FANG AND T.S. HUANG: A Corner-Finding Algorithm for Image Analysis and Registration . *Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition*, 1982, pp. 46-49.

[Fletcher] R. FLETCHER: Practical methods for optimization. *Wiley, New York*, 1990.

[Golub70] G.H. GOLUB AND C. REINSCH: Singular value decomposition and least squares solutions. *Springer, Berlin*, 1970.

[Grimson95] W. GRIMSON, T. LOZANO-PEREZ, W. WELLS, G. ETTINGER AND S. WHITE: An automatic registration method for frameless stereotaxy, image, guided surgery and enhanced reality visualization. *IEEE Conference on Computer Vision and Pattern Recognition*, 1995, pp. 430-436.

[Harris88] C. HARRIS AND M. STEPHEN: A combined corner and edge detector. *Proceedings of the 4th Alvey Vision Conference*, 1988, pp. 189-192.

[Kato99] H. KATO AND M. BILLINGHURST: Marker Tracking and HMD Calibration for a Video-Based Augmented Reality Conferencing System. *Proceedings of the 2nd IEEE and ACM International Workshop on Augmented Reality*, 1999.

[Klimenko01] S. KLIMENKO, G. MITSELMAKHER AND A. SAZONOV: A cross-correlation technique in wavelet domain for detection of stochastic gravitational waves. *Publication*, 2001.

[Koller97] D. KOLLER: Real-Time Vision-Based Camera Tracking for Augmented Reality Applications. *Proceedings of the Symposium on Virtual Reality Software and Techology*, 1997.

[Kutulakos96] K.J. KUTULAKOS AND J. VALLINO: Affine Objects Representation for Calibration-Free Augmented Reality. *Virtual Reality Ann. Int'l Symposium (VRAIS '96)*, 1996, pp. 430-436.

[Lewis95] J.P. LEWIS: Fast Normalized Cross-Correlation. *Publication*, 1995.

[Luong97] Q.T. LUONG AND O. FAUGERAS: Self-calibration of a moving camera from point correspondances and fundamental matrices. *International Journal of Computer Vision*, 1997, pp. 261-289.

[Milgram94] P. MILGRAM: A Tsaxonomy of Mixed Reality Visual Displays. *IEEE Transactions on Information Systems*, 1994, vol. 12.

[Mokhtarian98] F. MOKHTARIAN AND R. SUOMELA: Robust Image Corner Detection Through Curvature Scale Space. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1998, vol. 20, pp. 1376-1381.

[Roach80] J.W. ROACH AND J.K. AGGARWAL: Determining the movement of objects from a sequence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1980, vol. 6, pp. 554-562.

[Shariat90] H. SHARIAT AND K.E. PRICE: Motion estimation with more than two frames. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1990, pp. 417-434.

[Uenohara95] M. UENOHARA AND T. KANADE: Vision-Based Object Registration for Real-Time Image Overlay. *Computers in Biology and Medicine*, 1995, pp. 249-260.