

# The Epipolar Geometry Toolbox: multiple view geometry and visual servoing for MATLAB

Gian Luca Mariottini and Domenico Prattichizzo  
Dipartimento di Ingegneria dell'Informazione, Università di Siena  
Via Roma 56, 53100 Siena, Italy  
Email: {gmariottini,prattichizzo}@dii.unisi.it

**Abstract**— The Epipolar Geometry Toolbox (EGT) was realized to provide a MATLAB user with an extensible framework for the creation and visualization of multi-camera scenarios and the manipulation of the visual information and the geometry between them. Functions provided, for both pin-hole and panoramic vision sensors, include camera placement and visualization, computation and estimation of epipolar geometry entities and many others. The compatibility of EGT with the Robotics Toolbox [7] allows to address general vision-based control issues. Two applications of EGT to visual servoing tasks are here provided. This article introduces the Toolbox in tutorial form. Examples are provided to show its capabilities. The complete toolbox, the detailed manual and demo examples are freely available on the EGT web site [21].

## I. INTRODUCTION

The Epipolar Geometry Toolbox (EGT) is a toolbox designed for MATLAB [29]. MATLAB is a software environment, available for a wide range of platforms, designed around linear algebra principles and graphical presentations also for large datasets. Its core functionalities are extended by the use of many additional toolboxes. Combined with interactive MATLAB environment and advanced graphical functions, EGT provides a wide set of functions to approach computer vision problems especially with multiple views.

The Epipolar Geometry Toolbox allows to design vision-based control systems for both pin-hole and central panoramic cameras. EGT is fully compatible with the well known Robotics Toolbox by Corke [7]. The increasing interest in robotic visual servoing for both 6DOF kinematic chains and mobile robots equipped with pin-hole or panoramic cameras fixed to the workspace or to the robot, motivated the development of EGT.

Several authors, such as [4], [9], [18], [20], [24], have proposed new visual servoing strategies based on the geometry relating multiple views acquired from different camera configurations, i.e. the Epipolar Geometry [14].

In these years we have observed the necessity to develop a software environment that could help researchers to rapidly create a multiple camera setup, use visual data and design new visual servoing algorithms. With EGT we provide a wide set of easy-to-use and completely customizable functions to design general multi-camera scenarios and manipulate the visual information between them.

Let us emphasize that EGT can also be successfully employed in many other contexts when single and multiple view geometry is involved as, for example, in visual odometry and structure from motion applications [23] [22]. For example in the first work an interesting “visual odometry” approach for robot SLAM is proposed in which the multiple view geometry is used to estimate the camera motion from pairs of images without requiring the knowledge of the observed scene.

EGT, as the Robotics Toolbox, is a simulation environment, but the EGT functions can be easily embedded by the user in Simulink models. In this way, thanks to the MATLAB Real-Time Workshop, the user can generate and execute stand-alone C code for many off-line and real-time applications.

A distinguishable remark of EGT is that it can be used to create and manipulate visual data provided by both pin-hole and panoramic cameras. Catadioptric cameras, due to their wide field of view, has been recently applied in visual servoing [32].

The second motivation lead to the development of EGT was the increasing distribution of “free” software in the latest years, on the basis of the Free Software Foundation [10] principles. In this way users are allowed, and also encouraged, to adapt and improve the program as dictated by their needs. Examples of programs that follow these principles include for instance the Robotics Toolbox [7], for the creation of simulations in robotics, and the Intel’s OpenCV C++ libraries for the implementation of computer vision algorithms, such as image processing and object recognition [1].

The third important motivation for EGT was the availability and increasing sophistication of MATLAB. EGT could have been written in other languages, such as C, C++ and this would have freed it from dependency on other software. However these low-level languages are not so conducive to rapid program development as MATLAB.

This tutorial assumes the reader has familiarity with MATLAB and presents the basic EGT functions, after short theory recalls, together with intuitive examples. In this tutorial we also present two applications of EGT to visual servoing.

Section 2 presents the basic vector notation in EGT, while in Section 3 the pin-hole and omnidirectional camera models together with EGT basic functions are presented. In Section 4 we present the setup for multiple camera geometry (Epipolar

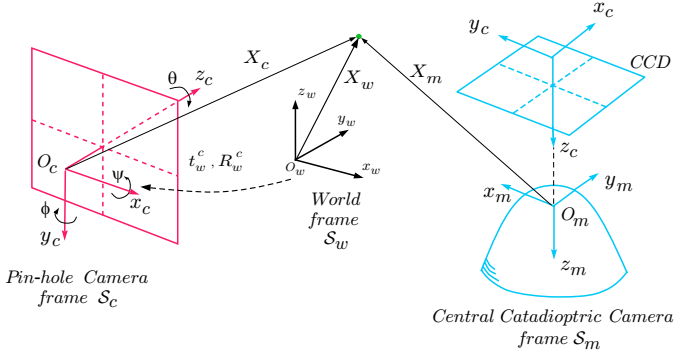


Fig. 1. Main reference frames notation and vector representation in EGT.

Geometry) while in Section 5 two applications of EGT to visual servoing are presented together with simulation results. In Section 6 we make a comparison between EGT and other software packages. EGT can be freely downloaded at [21], can be used under Windows and requires MATLAB 6.5 or upper. The detailed manual is provided in the EGT web site, with a large set of examples, figures and source code also for beginners.

## II. BASIC VECTOR NOTATION

We here present the basic vector notation adopted in Epipolar Geometry Toolbox. All scene points  $\mathbf{X}_w \in \mathbb{R}^3$  are expressed in the *world frame*  $\mathcal{S}_w = \langle O_w x_w y_w z_w \rangle$  (Fig.1). When referred to the *pin-hole camera frame*  $\mathcal{S}_c = \langle O_c x_c y_c z_c \rangle$  they will be indicated with  $\mathbf{X}_c$ . Moreover all scene points expressed w.r.t. a *central catadioptric camera frame*  $\mathcal{S}_m = \langle O_m x_m y_m z_m \rangle$  will be indicated with  $\mathbf{X}_m$ . For the reader convenience we briefly present the basic vector notation and transformation [25]. Refer to Fig. 1 and consider the  $3 \times 1$  vector  $\mathbf{X}_c \in \mathcal{S}_c$ . It can be expressed in  $\mathcal{S}_w$  as follows:

$$\mathbf{X}_w = \mathbf{R}_w^c \mathbf{X}_c + \mathbf{t}_w^c \quad (1)$$

where  $\mathbf{t}_w^c$  is the translational vector centered in  $\mathcal{S}_w$  and pointing toward the  $\mathcal{S}_c$  frame (Fig. 1). The matrix  $\mathbf{R}_w^c$  is the rotation necessary to align the world frame with the camera frame. For example we may choose  $\mathbf{R}_w^c = \mathbf{R}_{roll,pitch,yaw} = \mathbf{R}_{z,\theta} \mathbf{R}_{y,\phi} \mathbf{R}_{x,\psi}$ . The homogeneous notation aims to express (1) in linear form:

$$\tilde{\mathbf{X}}_w = \mathbf{H}_w^c \tilde{\mathbf{X}}_c$$

where  $\tilde{\mathbf{X}}_w = [\mathbf{X}_w^T \ 1]^T$ ,  $\tilde{\mathbf{X}}_c = [\mathbf{X}_c^T \ 1]^T$ . The  $4 \times 4$  matrix  $\mathbf{H}_w^c$  is referred to as *homogeneous transformation matrix*:

$$\mathbf{H}_w^c = \begin{bmatrix} \mathbf{R}_w^c & \mathbf{t}_w^c \\ \mathbf{0}^T & 1 \end{bmatrix}$$

Analogously, a point  $\mathbf{X}_w$  can be expressed in the camera frame by the following transformation

$$\tilde{\mathbf{X}}_c = \begin{bmatrix} \mathbf{R}_w^c T & -\mathbf{R}_w^c T \mathbf{t}_w^c \\ \mathbf{0}^T & 1 \end{bmatrix} \tilde{\mathbf{X}}_w \quad (2)$$

Consider now the more general case in which two camera frames, referred to as *actual* and *desired*, are observing the same point  $\mathbf{X}_w$ . From (1)

$$\mathbf{X}_w = \mathbf{R}_w^d \mathbf{X}_d + \mathbf{t}_w^d \quad (3)$$

$$\mathbf{X}_w = \mathbf{R}_w^a \mathbf{X}_a + \mathbf{t}_w^a \quad (4)$$

Substituting (4) in (3) it follows

$$\mathbf{X}_d = \underbrace{\mathbf{R}_w^d T \mathbf{R}_w^a}_{\mathbf{R}_d^a} \mathbf{X}_a + \underbrace{\mathbf{R}_w^d T (\mathbf{t}_w^a - \mathbf{t}_w^d)}_{\mathbf{t}_d^a} \quad (5)$$

Equation (5) will be very useful in EGT for the analytical computation of epipolar geometry where it is necessary to know the relative displacement  $\mathbf{t}_d^a$  and orientation  $\mathbf{R}_d^a$  between the two camera frames.

## III. PIN-HOLE AND OMNIDIRECTIONAL CAMERA MODELS

EGT provides easy-to-use functions for the placement of pin-hole and central catadioptric (or omnidirectional) cameras. Their imaging model has been here implemented to allow users to manipulate the visual information. In this section the fundamentals of perspective and omnidirectional camera models are quickly reviewed. The reader is referred to [14], [17], [5] for a detailed treatment. According to the purposes of this tutorial, some basic EGT code examples are reported together with the theory.

### A. Perspective camera

Consider a pin-hole camera located at  $O_c$  as in Fig. 2. The full perspective model describes the relationship between a 3D point (in homogeneous coordinates)  $\tilde{\mathbf{X}}_w = [X \ Y \ Z \ 1]^T$  expressed in the world frame and its projection  $\tilde{\mathbf{m}} = [u \ v \ 1]^T$  onto the image plane according to

$$\tilde{\mathbf{m}} = \mathbf{K} \mathbf{I} \tilde{\mathbf{X}}_w$$

where  $\mathbf{K} \in \mathbb{R}^{3 \times 3}$  is the camera *intrinsic parameters* matrix given by:

$$\mathbf{K} = \begin{bmatrix} k_u f & \gamma & u_0 \\ 0 & k_v f & v_0 \\ 0 & 0 & 1 \end{bmatrix}.$$

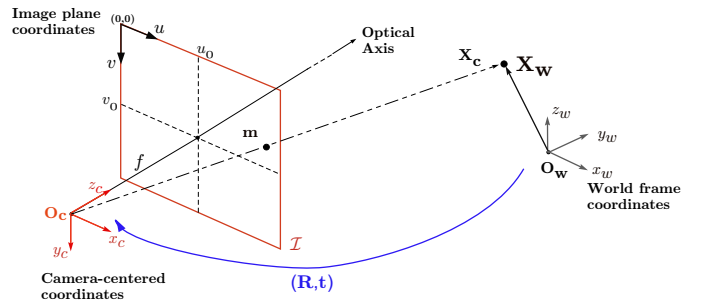


Fig. 2. The pin-hole camera model. The 3D point  $\mathbf{X}_w$  is projected onto  $\mathbf{m}$  through the optical center  $O_c$ . Note that  $\mathbf{m}$  is expressed in the image plane coordinates  $(u, v)$  (pixels).

Here  $(u_0, v_0)$  are the pixels coordinates, in the image frame, of the *principal point* (i.e. the intersection point between the image plane and the optical axis  $z_c$ ),  $k_u$  and  $k_v$  are the number of pixels per unit distance in image coordinates,  $f$  is the focal length (in meters) and  $\gamma$  is the orthogonality factor of the CCD image axes (*skew-factor*).

Matrix  $\mathbf{\Pi} = [\mathbf{R} | \mathbf{t}] \in \mathbb{R}^{3 \times 4}$  is the so-called *external parameters* camera matrix, that contains the rotation  $\mathbf{R}$  and the translation  $\mathbf{t}$  between the world and the camera frames. According to the commonly used notation, in the case of no camera rotation the optical axis  $z_c$  of pin-hole cameras is parallel to the  $y_w$  axis of  $\mathcal{S}_w$ . We then define:

$$\begin{aligned} \mathbf{R} &= (\mathbf{R}_{x, -\pi/2} \mathbf{R}_{rpy})^T \\ \mathbf{t} &= -(\mathbf{R}_{x, -\pi/2} \mathbf{R}_{rpy})^T \mathbf{t}_w^c \end{aligned}$$

In order to directly obtain the  $4 \times 4$  homogeneous matrix  $\mathbf{H}_c^w$  the function `f_Rt2H` is provided

```
>> H=f_Rt2H(R,t).
```

Note that with the use of `f_Rt2H` the position  $\mathbf{t}$  of a pin-hole camera is specified with respect to the *world frame* while the rotation  $\mathbf{R}$  is referred to the *pin-hole camera frame* axes. During the testing phase at the University of Siena this choice was appreciated from students of Robotics and Vision classes that addressed it as very intuitive.

*Example 1 (3D scene and pin-hole camera placement):*

Consider now a pin-hole camera rotated by  $\mathbf{R} = \mathbf{R}_{y, \pi/4} \in \mathbb{R}^{3 \times 3}$  and translated by  $\mathbf{t} = [-10, -5, 0]^T$ :

```
>> R=rotoy(pi/4);
>> t=[-10,-5,0]';
>> H=f_Rt2H(R,t);
```

In EGT the camera frame and the associated 3D camera can be visualized with functions `f_3Dframe(H)` and `f_3Dcamera(H)` respectively, where  $\mathbf{H}$  is the  $4 \times 4$  homogeneous transformation describing position and orientation of the camera with respect to  $\mathcal{S}_w$

```
>> f_3Dframe(H,1); %camera frame
>> hold on
>> f_3Dcamera(H); %3D pin-hole camera
>> axis equal, grid on, view(12,34)
>> title('3D setup - EGT Tutorial - Ex.1')
```

Plot of 3D view is reported in Fig. 3(a). All the functions have further options. See the EGT Manual [21] for details.

We can also place a set of  $N$  3D points  $\mathbf{X}_w^i = [X^i, Y^i, Z^i]$  (e.g. the rectangular panel vertexes) defined as

$$\mathbf{X}_w = \begin{bmatrix} X^1 & X^2 & \dots & X^N \\ Y^1 & Y^2 & \dots & Y^N \\ Z^1 & Z^2 & \dots & Z^N \end{bmatrix} \in \mathbb{R}^{3 \times N}$$

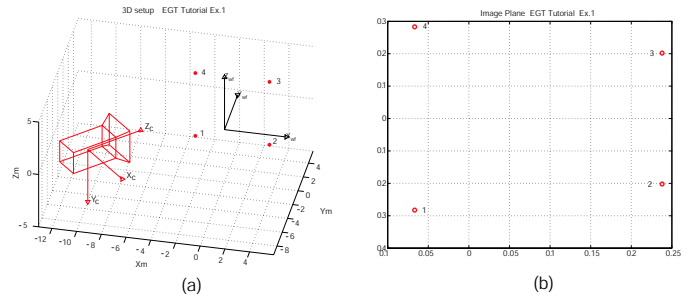


Fig. 3. Example 1. (a) A pin-hole camera is positioned in  $\mathbf{t} = [-10, -5, 0]$  in the 3D world frame and rotated by  $\pi/4$  around the  $y$ -axis. (b) The 3D scene points are projected onto the image plane. Note that in this case  $\mathbf{K} = \mathbf{I}$  for simplicity.

by the use of function `f_scenepnt(X)`

```
>> Xi=[-3, 3, 3, -3];
>> Yi=[ 3, 3, 3, 3];
>> Zi=[-3, -3, 3, 3];
>> Xw=[Xi; Yi; Zi];
>> f_scenepnt(Xw);
>> f_3Dwfenum(Xw); %enumerate points
```

The perspective projection  $\mathbf{m} = [u, v]^T$  of points  $\mathbf{X}_w$  is obtained with `f_perspproj(Xw, H, K)`:

```
>> [u,v]=f_perspproj(Xw,H,K);
>> plot(u,v,'rO')
```

Projection of scene points is represented in Fig. 3(b). □

Note that while the above example describes the placement of 3D points  $\mathbf{X}_w$ , EGT is also able to build scenes with more complex 3D objects returning surface points and normals (see function `f_3Dsurface` in [21]).

## B. Omnidirectional Camera Model

Omnidirectional cameras combine reflective surfaces (mirrors) and lenses. Several types of panoramic cameras can be obtained simply combining cameras (pin-hole or orthographic) and mirrors (hyperbolic, parabolic or elliptical) [5].

Panoramic cameras are classified according to the fact that they satisfy or not the single viewpoint constraint guaranteeing that the visual sensor only measures the light through a single point. Note that this constraint is required for the existence of epipolar geometry and for the generation of geometrically correct images [28] [12].

In [3], Baker *et al.* derive the entire class of catadioptric systems verifying the single viewpoint constraint. Among these EGT takes into account catadioptric systems consisting of pin-hole cameras coupled with hyperbolic mirrors, and orthographic cameras coupled with parabolic mirrors.

In [11] a unified projection model for central catadioptric camera systems has been proposed. In particular it was shown that all central panoramic cameras can be modelled by a

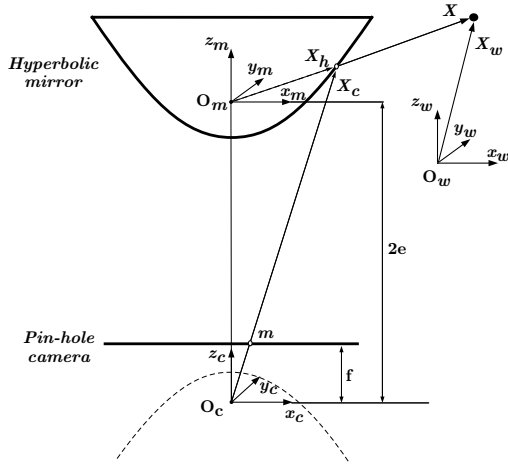


Fig. 4. The panoramic camera model (pin-hole camera and hyperbolic mirror). The 3D point  $\mathbf{X}_w$  is projected at  $\mathbf{m}$  through the optical center  $O_c$ , after being projected at  $\mathbf{X}_h$  through the mirror center  $O_m$ .

particular mapping on a sphere, followed by a projection from a point on the camera optical axis onto the image plane.

In order to keep in EGT a physically meaningful graphical representation we decided, without losing generality, to represent the central panoramic cameras not as spheres in the space but with the couple of a CCD camera with a parabolic or hyperbolic mirror (see for example Fig. 5).

In what follows the imaging model for a pin-hole camera with hyperbolic mirror is described.

Consider now the basic scheme in Fig. 4. Note that in this case all frames (for both the camera and the mirror) are aligned with the world frame. Three important reference frames are defined: (1) the world reference system centered at  $O_w$  whose vector is  $\mathbf{X}_w$ ; (2) the mirror coordinate system centered at the focus  $O_m$  whose vector is  $\mathbf{X} = [X, Y, Z]^T$ ; and (3) the camera coordinate system centered at  $O_c$  whose vector is  $\mathbf{X}_c$ .

Henceforth all equations will be expressed in the mirror reference frame if not stated otherwise.

Refer to Fig. 4 and let  $a$  and  $b$  be the hyperbolic mirror parameters

$$\frac{(z+e)^2}{a^2} - \frac{x^2+y^2}{b^2} = 1$$

with eccentricity  $e = \sqrt{a^2+b^2}$ , the transformation to obtain the projection  $\mathbf{u}$  in the pin-hole camera frame (see Fig. 4) is given by

$$\mathbf{m} = \mathbf{K} \frac{1}{2e} \left( \mathbf{R}_c^m \left( \lambda \mathbf{R}_w^m \left( \mathbf{X}_w - \mathbf{t}_w^m \right) \right) + \mathbf{t}_c^m \right) \quad (6)$$

where  $\lambda = \frac{b^2(-eZ \pm a\|\mathbf{X}\|)}{b^2Z^2 - a^2X^2 - a^2Y^2}$  is a nonlinear function of  $\mathbf{X}$ .  $\mathbf{K}$  is the internal calibration matrix of CCD camera looking at the mirror.  $\mathbf{t}_c^m$  is the mirror center expressed in the camera frame and corresponds to  $[0, 0, 2e]$ .  $\mathbf{R}_c^m$  is the matrix representing the rotation between camera and mirror frames. Analogously  $\mathbf{t}_w^m$  and  $\mathbf{R}_w^m$  represent the mirror configuration (rotation and orientation) with respect to the world frame.

In EGT a central catadioptric camera is defined by specifying the homogeneous transformation matrix between mirror and world frames

$$\mathbf{H}_w^m = \begin{bmatrix} \mathbf{R}_w^m & \mathbf{t}_w^m \\ \mathbf{0}^T & 1 \end{bmatrix}$$

*Example 2 (Panoramic camera placement):* In EGT a panoramic camera can be placed and visualized. Let us place the camera at  $\mathbf{t} = [-5, -5, 0]^T$  with orientation  $\mathbf{R} \equiv R_{z, \pi/4}$ . EGT provides a function to simply visualize the panoramic camera in the 3D world frame as in Fig. 5.

```
>> H=[rotoz(pi/4) , [-5,-5,0]';
>>      0 0 0 , 1 ];
>> f_3Dpanoramic(H);
```

Moreover, for assigned camera calibration matrix  $\mathbf{K}$ :

```
K=[ 10^3  0  320;
    0  10^3  240;
    0  0  1 ];
```

the projection of a 3D point  $\mathbf{X}_w = [0, 0, 4]^T$  in both the camera ( $\mathbf{m}$ ) and mirror ( $\mathbf{X}_h$ ) frames can be obtained from:

```
>> [m, Xh] = f_panproj(Xw, H, K);
>>
m =
4.1048e+002
2.4000e+002
1.0000e+000

Xh =
6.0317e-001
3.7881e-017
3.4120e-001
```

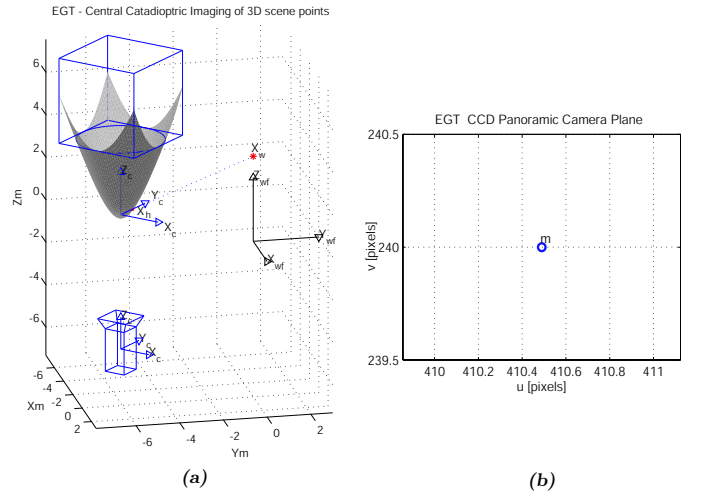


Fig. 5. Example 2. (a) A panoramic camera is positioned in  $[-5, -5, 0]^T$  in the 3D world frame and (b) the 3D point  $\mathbf{X}_w = [0, 0, 4]^T$  is projected to the pinhole camera after being projected in  $\mathbf{X}_h$ .

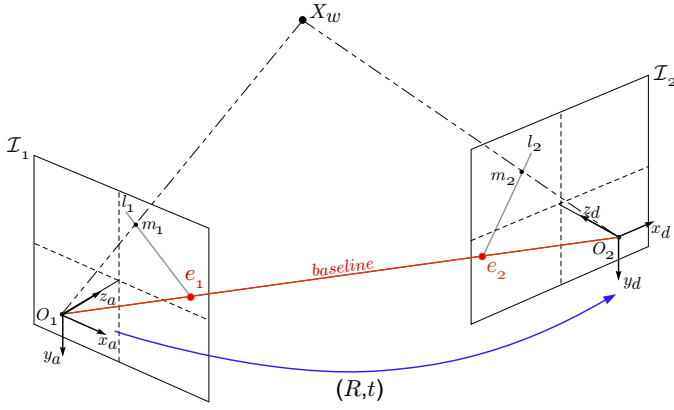


Fig. 6. Basic Epipolar Geometry entities for pinhole cameras.

1.0000e+000

Graphical results are reported in Fig. 5 and more accurately described in the *EGT Reference Manual*. More examples can be found in the directory demos.  $\square$

#### IV. EPIPOLAR GEOMETRY

In this section the EGT functions dealing with the Epipolar Geometry are presented.

##### A. Epipolar geometry for pin-hole cameras

Consider two perspective views of the same scene taken from distinct viewpoints  $O_1$  and  $O_2$ , as in Fig.6. Let  $\mathbf{m}_1$  and  $\mathbf{m}_2$  be the corresponding points in two views, i.e. the perspective projection through  $O_1$  and  $O_2$ , of the same point  $\mathbf{X}_w$ , in both image planes  $\mathcal{I}_1$  and  $\mathcal{I}_2$ , respectively. The epipolar geometry defines the geometric relationship between these corresponding points.

Associated with Fig.6 we define the following set of geometric entities [14], [17]:

##### Definition 1: Epipolar Geometry

- 1) The plane passing through the optical centers  $O_1$  and  $O_2$  and the scene point  $\mathbf{X}_w$ , is called an *epipolar plane*. Note that in a  $N$ -points scene, a pencil of planes exists (one for each scene point  $\mathbf{X}_w^i$ )
- 2) The projection  $\mathbf{e}_1$  ( $\mathbf{e}_2$ ) of one camera center  $O_1$  ( $O_2$ ) onto the image plane of the other camera frame  $\mathcal{I}_2$  ( $\mathcal{I}_1$ ) is called *epipole*. The epipole will be expressed in homogeneous coordinates

$$\tilde{\mathbf{e}}_1 = [e_{1x} \ e_{1y} \ 1]^T \quad \tilde{\mathbf{e}}_2 = [e_{2x} \ e_{2y} \ 1]^T$$

- 3) The intersection of the epipolar plane for  $\mathbf{X}_w$  with image plane  $\mathcal{I}_1$  ( $\mathcal{I}_2$ ), defines the *epipolar line*  $l_1$  ( $l_2$ ). Note that all epipolar lines pass through the epipole.

One of the main parameters of the epipolar geometry is the  $3 \times 3$  Fundamental Matrix  $\mathbf{F}$  that conveys most of the information about the relative position and orientation  $(\mathbf{t}, \mathbf{R})$  between the two views. Moreover, the fundamental matrix

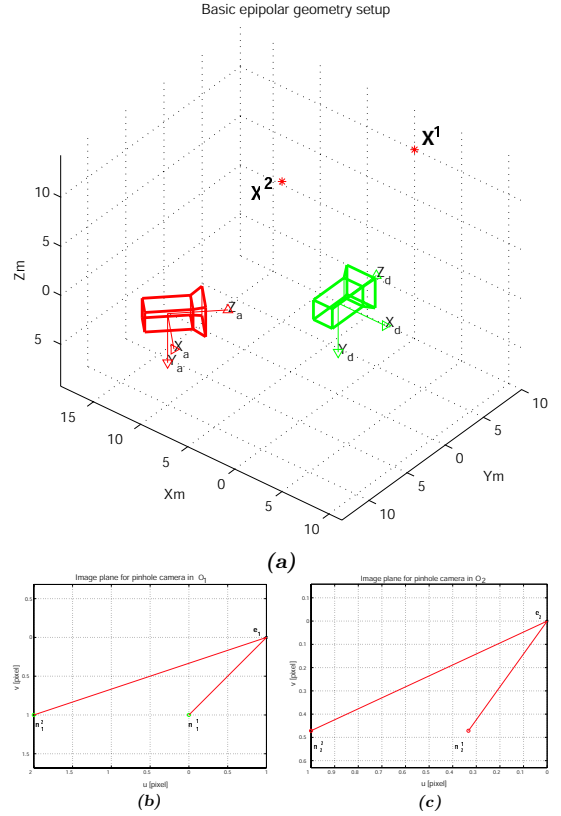


Fig. 7. Example 3. (a) 3D simulation setup for epipolar geometry, with both cameras looking at two scene points  $\mathbf{X}_w^1$  and  $\mathbf{X}_w^2$ ; (b)-(c) image planes of both pin-hole cameras with corresponding features and epipoles. Note that all epipolar lines meet at the epipole and pass through the corresponding points.

algebraically relates corresponding points in the two images through the *Epipolar Constraint*:

**Theorem 1: (Epipolar Constraint)** Let be given two views of the same 3D point  $\mathbf{X}_w$ , both characterized by their relative position and orientation  $(\mathbf{t}, \mathbf{R})$  and the internal calibration parameters  $\mathbf{K}_1$  and  $\mathbf{K}_2$ . Moreover, let  $\mathbf{m}_1$  and  $\mathbf{m}_2$  (homogeneous notation) be the two corresponding projections of  $\mathbf{X}_w$  in both image planes. The epipolar constraint is:

$$\mathbf{m}_2^T \mathbf{F} \mathbf{m}_1 = 0 \quad \text{where} \quad \mathbf{F} = \mathbf{K}_2^{-T} [\mathbf{t}]_{\times} \mathbf{R} \mathbf{K}_1^{-1}.$$

and  $[\mathbf{t}]_{\times}$  is the  $3 \times 3$  skew-symmetric matrix associated to  $\mathbf{t}$ .  $\square$

The epipolar geometry is reviewed extensively in [14], [17].

**Example 3 (Epipolar Geometry for pinhole cameras):** The computation of epipolar geometry with EGT is straightforward. Consider a two-views scene as in Fig. 7(a) where the first camera is placed at  $\mathbf{t}_1 = [-10, -10, 0]^T$  with orientation  $\mathbf{R}_1 = \mathbf{R}_{y, \pi/4}$ , while the second one is coincident with the world frame, i.e.  $\mathbf{t}_2 = [0, 0, 0]^T$  with orientation  $\mathbf{R}_2 = \mathbf{I}$ . Pin-hole cameras can be shown and their homogeneous transformation matrices  $\mathbf{H}_1$  and  $\mathbf{H}_2$  can be obtained with `f_Rt2H` as described in Sec. III-A. Two feature points in  $\mathbf{X}_w^1 = [0, 10, 10]^T$  and  $\mathbf{X}_w^2 = [-10, 5, 5]^T$  are observed and projected in  $\mathbf{m}_1^1, \mathbf{m}_1^2$  and  $\mathbf{m}_2^1, \mathbf{m}_2^2$  onto the

two image planes, respectively.

In EGT the computation of epipoles and fundamental matrix is obtained through function

```
>> [e1,e2,F]=f_epipole(H1,H2,K1,K2);
```

while the epipolar lines are retrieved with:

```
>> [l1,l2]=f_epipline(m1,m2,F);
```

Note that  $\mathbf{l}_1$  ( $\mathbf{l}_2$ ) is a vector in  $\mathbb{R}^3$  in homogeneous coordinates. In (Fig. 7(b)) both image planes are represented together with feature points, epipoles and the corresponding epipolar lines. In this case, for the sake of simplicity, it is assumed that  $\mathbf{K} = \mathbf{I}$ .

### B. Epipolar Geometry for panoramic cameras

As in the pinhole cameras, the epipolar geometry is here defined when a pair of panoramic views is available. Note however that in this case epipolar lines become epipolar conics. The reader is referred to [11], [27] for an exhaustive treatment.

Let now consider two panoramic cameras (with equal hyperbolic mirror  $\mathbf{Q}$ ) with foci  $O_{m1}$  and  $O_{m2}$  respectively, as represented in Fig. 8. Moreover, let be  $\mathbf{R}$  and  $\mathbf{t}$  be the rotation and translation between the two mirror frames, respectively. Let  $\mathbf{X}_{h1}$  and  $\mathbf{X}_{h2}$  represent the mirror projections of the scene point  $\mathbf{X}$  onto both hyperbolic mirrors.

The coplanarity of  $\mathbf{X}_{h1}$ ,  $\mathbf{X}_{h2}$  and  $\mathbf{X}$  can be expressed as:

$$\mathbf{X}_{h2}^T \mathbf{E} \mathbf{X}_{h1} = 0 \quad \text{where} \quad \mathbf{E} = [\mathbf{t}]_{\times} \mathbf{R} \quad (7)$$

Equivalently to the pin-hole case, the  $3 \times 3$  matrix  $\mathbf{E}$  is referred to as the *essential matrix* [15].

Vectors  $\mathbf{X}_{h1}$ ,  $\mathbf{X}_{h2}$  and  $\mathbf{t}$  define the *epipolar plane* with normal vector  $\mathbf{n}_{\pi}$ , that intersects both mirror quadrics  $\mathbf{Q}$  in the two *epipolar conics*  $\mathbf{C}_1$  and  $\mathbf{C}_2$  [26]. For each 3D point  $\mathbf{X}^i$  a pair of epipolar conics  $\mathbf{C}_1^i$  and  $\mathbf{C}_2^i$  exist for the two views setup.

All the epipolar conics pass through two pairs of epipoles,  $\mathbf{e}_1$  and  $\mathbf{e}'_1$  for one mirror and  $\mathbf{e}_2$  and  $\mathbf{e}'_2$  for the second one, which are defined as the intersections of the two mirrors  $\mathbf{Q}$

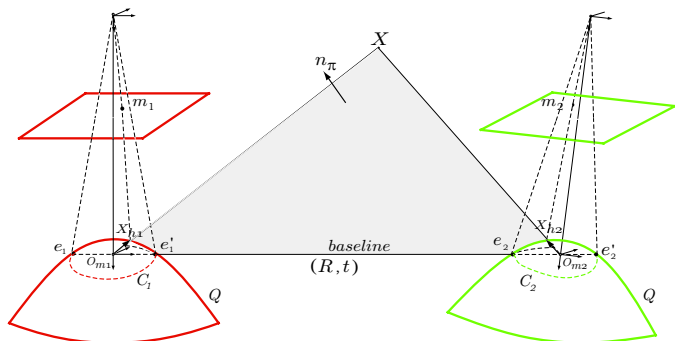


Fig. 8. Epipolar geometry for panoramic camera sensors.

with the baseline  $\overline{O_{m1}O_{m2}}$  (see Fig.8).

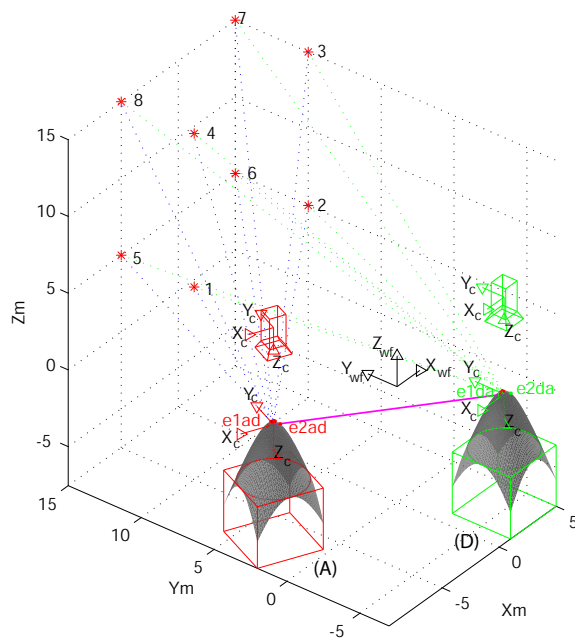


Fig. 9. Example 4: Central catadioptric cameras: epipolar geometry computation and visualization in EGT.

*Example 4 (Epipolar geometry for panoramic cameras):* Similar camera and scene configurations as for the Example 3 have been set. Now we placed 8 feature points in the space. By using the EGT function:

```
>> [e1c,e1pc,e2c,e2pc] = f_panepipoles(H1,H2);
```

the CCD coordinates of the epipoles are easily computed and plotted.

The CCD projections of all epipolar conics  $\mathbf{C}_1^i$  and  $\mathbf{C}_2^i$  where  $i = 1, \dots, 8$ , corresponding to all 3D points in  $\mathbf{X}_w$  are computed and visualized by

```
>> [C1m,C2m] = f_epipconics(Xw,H1,H2);
```

□

Fig. 9 presents the application of EGT to the computation of epipolar geometry between central catadioptric cameras (A) and (D). Note that also the epipoles onto the mirror surface are visualized.

Epipolar conics in both CCD image plane, together with corresponding feature points, are shown in Fig. 10(A)-(B). The complete MATLAB source code is available at the EGT web-site [21].

### C. Estimation of Epipolar Geometry

Up to now the EGT functions we presented for the epipolar geometry computation were based on the knowledge of the relative position and orientation between the two cameras. However many algorithms exist in the literature to estimate

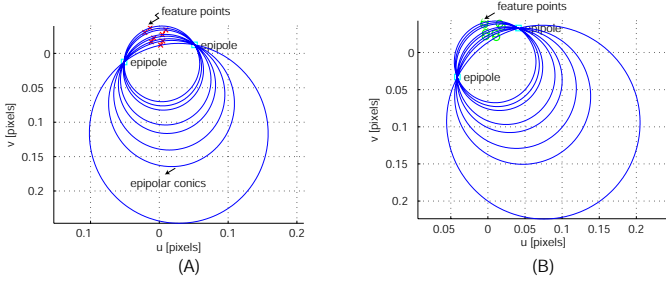


Fig. 10. Example 4: Epipolar conics plot in camera planes (A) and (B). All epipolar conics, in each image plane, intersect at the epipoles.

ID	Estimation Algorithm
1	<i>Unconstrained linear estimation</i> [16]
2	<i>The normalized 8-point algorithm</i> [13]
3	<i>Algorithm based on Geometric Distance (Sampson)</i> [14]

TABLE I

THE EPIPOLAR GEOMETRY ESTIMATION ALGORITHMS IMPLEMENTED IN EGT (WITH BIBLIOGRAPHY).

the epipolar geometry from the knowledge of corresponding feature points  $\mathbf{m}_a^i$  and  $\mathbf{m}_d^i$ . A very important feature of EGT is that some of the most important algorithms for the epipolar geometry estimation have been embedded in the toolbox code. In particular we implemented the linear method [16], the well known Hartley's normalized 8-points algorithm [13] and the iterative algorithm based on geometric distance (Sampson) [14] as summarized in Table I. A complete comparison and review of these methods has been reported in [14]. The epipolar geometry estimation is one of the most relevant features of EGT. The main reason is that users can run simulations where the fundamental matrix is estimated from noisy feature points.

The basic EGT function to estimate the fundamental matrix  $F$  is

```
>> F=f_Festim(U1,U2,algorithm);
```

where  $U1$  and  $U2$  are two  $(2 \times N)$  matrices containing all  $N$  corresponding features

$$U_1 = \begin{bmatrix} x_1^1 & x_1^2 & \dots & x_1^N \\ y_1^1 & y_1^2 & \dots & y_1^N \end{bmatrix}$$

The integer algorithm identifies the type of estimation algorithm to be used between those available according to Table I.

It is worth noting that these functions allows EGT to estimate the epipolar geometry also in real experiments other than simulations.

## V. VISUAL SERVOING

When used together with a robotics toolbox, such as the Robotics Toolbox (RT) by Corke [7], EGT proves to be an efficient tool to implement visual servoing techniques based also on multiple view geometry.

The visual servoing controllers proposed by Rives in [24] (for pin-hole cameras) and that proposed in [19] (for central panoramic cameras) have been considered as tutorial examples to show the main EGT features in a visual servoing context.

### A. Visual Servoing with Pin-hole Cameras for 6DOF Robots

In what follows the algorithm proposed in [24] is briefly recalled for the reader convenience. The main goal of this visual servoing is to drive a 6DOF robot arm (simulated with RT), equipped with a CCD camera (simulated with EGT), from a starting configuration toward a desired one using only image data provided during the robot motion (Fig. 11). The basic visual servoing idea consists in decoupling the camera/end-effector rotations and translations by the use of the hybrid vision-based task function approach [9]. The servoing task is performed minimizing an error function  $\varphi$  which is divided in both a *priority task*  $\varphi_1$ , that acts rotating the actual camera until the desired orientation is get, and in a *secondary task*  $\varphi_2$  (to be minimized under the constraint  $\varphi_1 = 0$ ) that has been chosen in [24] to be the camera distance to the target position.

The analytical expression of the error function is given by

$$\varphi = W^+ \varphi_1 + \beta (\mathbb{I}_6 - W^+ W) \frac{\partial \varphi_2}{\partial X} \quad (8)$$

where  $\beta \in \mathbb{R}^+$ , while  $W^+$  is the pseudo-inverse of the matrix  $W \in \mathbb{R}^{m \times n}$ :  $Range(W^T) = Range(J_1^T)$  where  $J_1$  is the Jacobian matrix of task function  $J_1 = \frac{\partial \varphi_1}{\partial X}$ .

The *priority task* accomplishes to rotate the camera/end-effector using a well-known epipolar geometry property stating that when both the initial and the desired cameras gain the same orientation, then all the distances between feature points  $\mathbf{m}_i$  and corresponding epipolar lines  $l_i^j$  are zero.

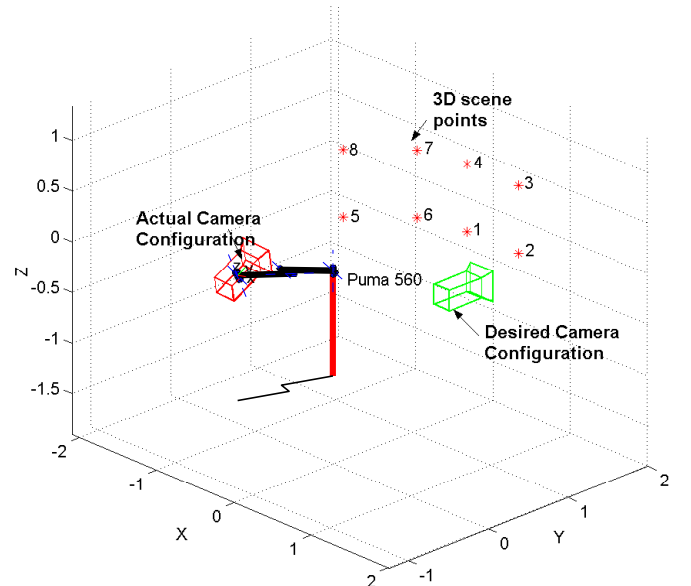


Fig. 11. Simulation setup for EGT application to visual servoing. EGT is compatible also with Robotics Toolbox.

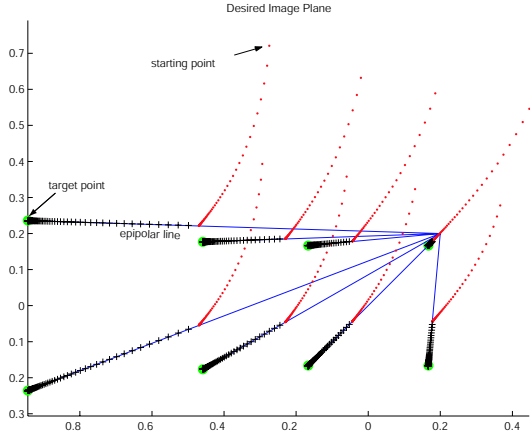


Fig. 12. Migration of feature points from initial position toward desired one. Feature points, from a starting position, reach the epipolar lines at the end of the first task (priority task) and then move toward desired ones (crossed points), during the second task.

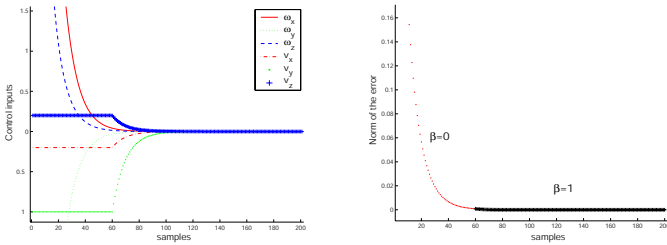


Fig. 13. (a) Control inputs to PUMA 560; (b) Norm of the error function  $e$  when  $\beta = 0$  and when  $\beta = 1$ .

Fig. 12 shows the simulation results of the algorithm previously proposed for a scene with 8 points. The dotted points in Fig. 12 shows, in the image plane ( $\mathbf{K} = \mathbf{I}$ ), the migration of the features to the epipolar lines during the rotation (first task), while crossed points correspond to the camera translation (second task).

In Fig. 13 the control inputs ( $\mathbf{v}_c, \omega_c$ ) and the plot of the norm of the error function  $e$  are reported.

The MATLAB code of this simulation (SIM\_Rives00.m) is included in the directory demo\_pinhole of EGT [21].

### B. Visual Servoing with Panoramic Cameras for Mobile Robot

While visual servoing with pin-hole cameras has been deeply investigated in the literature, less results exist for omnidirectional cameras. Interesting contributions for mobile robots equipped with panoramic cameras can be found in [6], [8], [31].

We here briefly recall the application of the Epipolar Geometry Toolbox to the design of a visual servoing technique for holonomic mobile robots equipped with a central panoramic camera presented in [19]. The proposed strategy does not assume any knowledge about the 3D observed scene geometry and position with respect to the camera/robot.

Consider a holonomic mobile robot and let  $q = (x, y, \theta)^T$  be its configuration vector with respect to the world frame

$\langle Oxyz \rangle_w$  (Fig. 14). Let  $v_x, v_y$  be the translational velocities and  $\omega$  the rotational velocity, respectively

$$\begin{aligned} \dot{x} &= v_x \\ \dot{y} &= v_y \\ \dot{\theta} &= \omega \end{aligned} \quad (9)$$

Suppose that a fixed catadioptric camera is mounted on the mobile robot such that its optical axis is perpendicular to the  $xy$ -plane. Without losing generality, suppose that a target panoramic image, referred to as *desired*, has been previously acquired in the desired configuration  $q_d = (0, 0, 0)^T$ . Moreover, one more panoramic view, the *actual* one, is available at each time instant from the camera-robot in the actual position.

As shown in the initial setup of Fig. 15 (upper-left), the mobile robot disparity between the actual and the desired poses is characterized by rotation  $R \in SO(2)$  and translation  $t \in \mathbb{R}^2$ . The control law will be able to drive the robot disparity between the actual and the desired configuration to zero only using visual information. In particular, we will regulate separately the rotational disparity (*first step*) and the translational displacement (*second step*) (upper-center and upper-right in Fig. 15, respectively).

In the *first step* the rotational disparity is compensated exploiting the auto-epipolar condition [19]. This novel property is based on the concept of bi-osculating conics (or bi-conics, for short), computed on the image plane from observed image corresponding features. It has been shown [19] that when all bi-conics intersect in only two points then the rotational disparity between the actual and the desired view is compensated. Fig. 15 (bottom-left) shows the image plane bi-conics when the two cameras are not in auto-epipolar configuration. On the other hand Fig. 15 (bottom-center) shows the image plane bi-conics when the cameras are in auto-epipolar configuration and all bi-conics intersect in only two points.

Once the rotational disparity is compensated, then the *second step* is executed. It consists of a translational motion in order to approach the desired position (Fig. 15 top-right). This translational control law can be designed on the image plane constraining all current features to lie on the epipolar conics and also the distance between current and desired features  $s(t)$

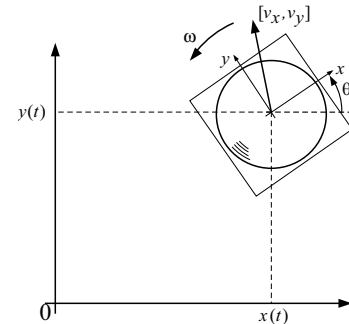


Fig. 14. The holonomic mobile robot with a fixed catadioptric camera moving on a plane.

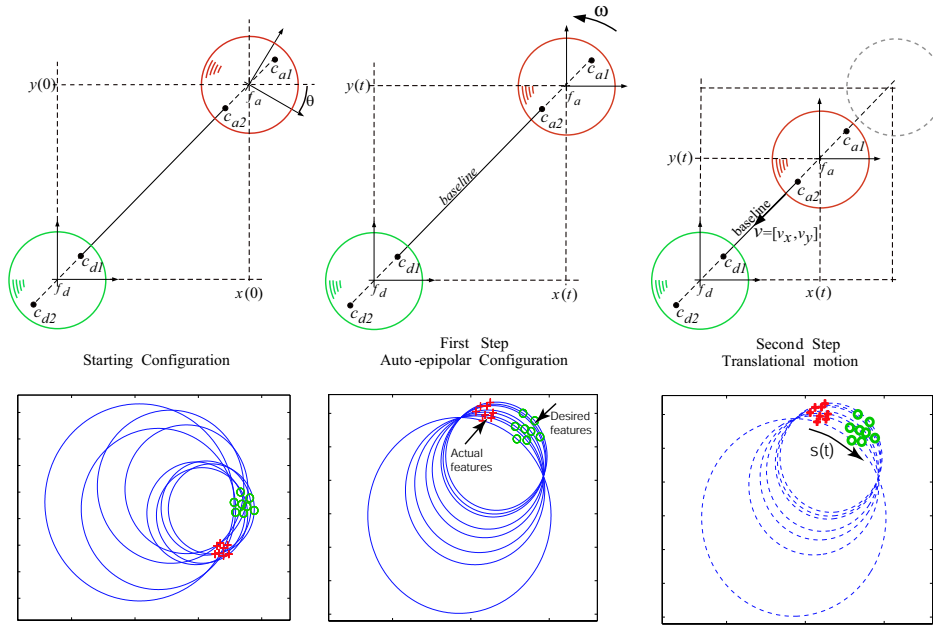


Fig. 15. The visual servoing control law for panoramic cameras drives the holonomic camera-robot from the starting configuration centered at  $f_a$  (no common intersection between bi-conics exists), through an intermediate configuration when both views have the same orientation (bi-conic common intersection), towards the position  $f_d$  (until  $s(t) = 0$ ). Bottom figures have been obtained with EGT.

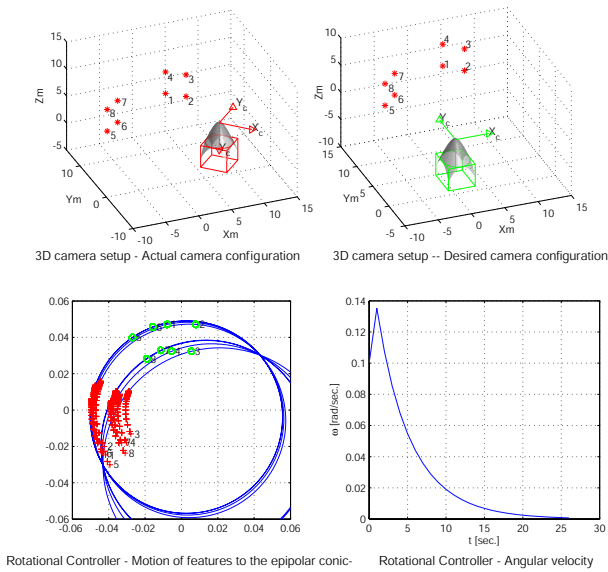


Fig. 16. Simulation results with EGT for the first step. The robot, placed in the actual configuration, rotates until the same orientation of the desired configuration is obtained.

to decrease as the robot approach the goal (Fig. 15 *bottom-right*).

Fig. 16 gathers some simulation results for the first step. The left and the right top windows show, respectively, the actual and desired camera locations, together with scene points. On the left bottom window of Fig. 16 we can observe the feature points (crosses) moving from initial position towards

the epipolar conics. In the right bottom Fig. 16 it is shown the angular velocity  $\omega$  provided by the controller that decreases to zero. Fig. 17 shows simulation results for the second step. In the left-top window the translational velocities  $v_x(t)$  and  $v_y(t)$  are reported. In the right-top window the distance  $s(t)$  between all feature points goes to zero, as reported in the bottom center window, where crossed points (actual positions) go to the desired one (circles).

The MATLAB code for the auto-epipolar property is the `ex5_autoepipolar.m` and is included in the directory `demo_panoramic` of EGT.

## VI. COMPARISON WITH OTHER PACKAGES

To the best of our knowledge no other software packages for the MATLAB environment exist that deal with the creation of multi-camera scenarios for both pin-hole and panoramic cameras.

An interesting MATLAB package is the Structure and Motion Toolkit by Torr [30]. This toolkit is specifically designed for features detection and matching, robust estimation of the fundamental matrix, self-calibration, and recovery of the projection matrices. The main difference with EGT is that the work of Torr is specifically oriented to pin-hole cameras and is not specifically designed to run simulations.

Regarding the visual servoing, it is worthwhile to mention the interesting Visual Servoing Toolbox (VST) by Cervera *et al.* [2] providing a set of Simulink blocks for simulation of vision-controlled systems. The main difference is that the VST implements some well known visual servoing control laws but does not consider neither multiple view geometry

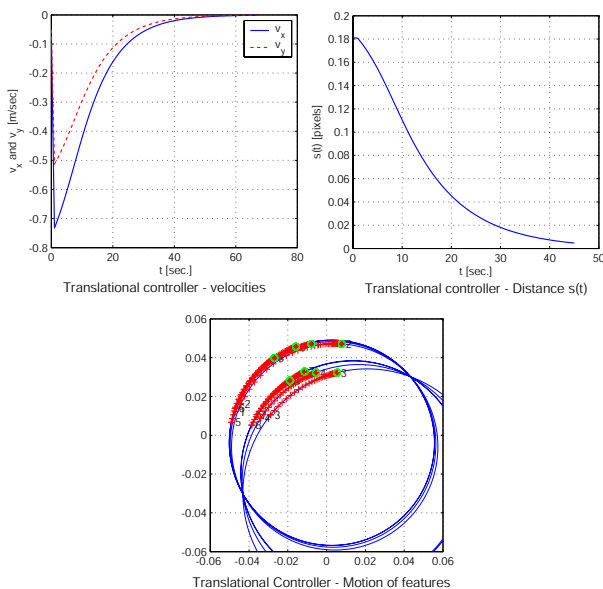


Fig. 17. Simulation results with EGT for the second step. The translational controller guarantees the motion of the robot along the baseline (the feature slide along the bi-conic). Then it stops when the distance  $s(t)$  between corresponding features is equal to zero i.e. the robot is in the desired pose.

nor panoramic cameras that are the distinguishing features of EGT.

## VII. CONCLUSIONS

The Epipolar Geometry Toolbox for MATLAB is a software package targeted to research and education in Computer Vision and Robotics Visual Servoing. It provides the user a wide set of functions for design multi-camera systems for both pin-hole and panoramic cameras. Several epipolar geometry estimation algorithms have been implemented. EGT is freely available and can be downloaded at the EGT web site together with a detailed manual and code examples.

## VIII. ACKNOWLEDGMENTS

Authors want to thank Eleonora Alunno for her help to develop EGT functions during her Master Thesis and to Jacopo Piazzi for his useful comments. EGT has been widely tested by the students of the Robotics and Vision classes and laboratories at the University of Siena who contributed to the testing phase of Epipolar Geometry Toolbox. Special thanks go to Paola Garosi, Annalisa Bardelli and Fabio Morbidi that helped us with useful comments on the toolbox and on the paper.

## REFERENCES

- [1] Intel's OpenCV. <http://sourceforge.net/projects/opencvlibrary/>.
- [2] *The Visual Servoing Toolbox*, 2003. <http://vstoolbox.sourceforge.net/>.
- [3] S. Baker and S.K. Nayar. Catadioptric image formation. In *Proc. of DARPA Image Understanding Workshop*, pages 1431–1437, 1997.
- [4] R. Basri, E. Rivlin, and I. Shimshoni. Visual homing: Surfing on the epipoles. In *International Conference on Computer Vision*, pages 863–869, 1998.
- [5] R. Benosman and S.B. Kang. *Panoramic Vision: Sensors, Theory and Applications*. Springer Verlag, New York, 2001.

- [6] D. Burschka and G. Hager. Vision-based control of mobile robots. In *2001 IEEE International Conference on Robotics and Automation*, pages 1707–1713, 2001.
- [7] P.I. Corke. A robotics toolbox for MATLAB. *IEEE Robotics and Automation Magazine*, 3:24–32, 1 1996.
- [8] N. Cowan, O. Shakernia, R. Vidal, and S. Sastry. Vision-based follow the leader. In *2003 IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 2, pages 1796–1801, 2003.
- [9] B. Espiau, F. Chaumette, and P. Rives. A new approach to visual servoing in robotics. In *IEEE Transactions on Robotics and Automation*, volume 8(3), 313–326 1992.
- [10] Free Software Foundation. The free software definition. <http://www.fsf.org/philosophy/free-sw.html>.
- [11] C. Geyer and K. Daniilidis. A unifying theory for central panoramic systems. In *6th European Conference on Computer Vision*, pages 445–462, 2000.
- [12] C. Geyer and K. Daniilidis. Mirrors in motion: Epipolar geometry and motion estimation. In *9th IEEE International Conference on Computer Vision*, volume 2, pages 766–773, 2003.
- [13] R. Hartley. In defence of the 8-point algorithm. In *IEEE International Conference on Computer Vision*, pages 1064–1070, 1995.
- [14] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge University Press, September 2000.
- [15] H.C. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293:133–135, 1981.
- [16] Q.T. Luong and O.D. Faugeras. The fundamental matrix: theory, algorithms and stability analysis. *International Journal of Computer Vision*, 17(1):43–76, 1996.
- [17] Y. Ma, S. Soatto, J. Kořecká, and S. Shankar Sastry. *An invitation to 3-D Vision, From Images to Geometric Models*. Springer Verlag, New York, 2003.
- [18] E. Malis and F. Chaumette. 2 1/2 d visual servoing with respect to unknown objects through a new estimation scheme of camera displacement. *International Journal of Computer Vision*, 37(1):79–97, 2000.
- [19] G.L. Mariottini, E. Alunno, J. Piazzi, and D. Prattichizzo. Epipole-based visual servoing with central catadioptric camera. In *2005 IEEE International Conference on Robotics and Automation*, 2005. (in press).
- [20] G.L. Mariottini, G. Oriolo, and D. Prattichizzo. Epipole-based visual servoing for nonholonomic mobile robots. In *2004 IEEE International Conference Robotics and Automation*, pages 497–503, 2004.
- [21] G.L. Mariottini and D. Prattichizzo. *The Epipolar Geometry Toolbox for Matlab*. University of Siena, <http://egt.dii.unisi.it>.
- [22] D. Nister. An efficient solution to the five-point relative pose problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(6):756–770, 2004.
- [23] D. Nister, O. Naroditsky, and J. Bergen. Visual odometry. In *2004 IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 652–659, 2004.
- [24] P. Rives. Visual servoing based on epipolar geometry. In *International Conference on Intelligent Robots and Systems*, volume 1, pages 602–607, 2000.
- [25] M.W. Spong and M. Vidyasagar. *Robot Dynamics and Control*. John Wiley & Sons, New York, 1989.
- [26] T. Svoboda. *Central Panoramic Cameras Design, Geometry, Egomotion*. Phd thesis, Center for Machine Perception, Czech Technical University, September 1999.
- [27] T. Svoboda and T. Pajdla. Epipolar geometry for central catadioptric cameras. *International Journal of Computer Vision*, 49(1):23–37, 2002.
- [28] T. Svoboda, T. Pajdla, and V. Hlaváč. Motion estimation using central panoramic cameras. In *IEEE Conference on Intelligent Vehicles*, pages 335–340, 1998.
- [29] Inc. The MathWorks. Matlab. [www.mathworks.com](http://www.mathworks.com).
- [30] Philip Torr. *The Structure and Motion Toolkit for MATLAB*. <http://www.cms.brookes.ac.uk/~philliptorr/>.
- [31] K. Usher, P. Ridley, and P. Corke. Visual servoing for a car-like vehicle - An application of omnidirectional vision. In *2003 IEEE International Conference on Robotics and Automation*, pages 4288–4293, 2003.
- [32] R. Vidal, O. Shakernia, and S. Sastry. Omnidirectional vision-based formation control. In *Annual Allerton Conference on Communication, Control and Computing*, pages 1637–1646, 2002.