Tracking planar structures with log-polar images

Alexandre Bernardino⋆, **José Santos–Victor**⋆⋆ **and Giulio Sandini**⋆⋆⋆ †

Instituto de Sistemas e Robótica
Instituto Superior Técnico
Lisboa, Portugal

Instituto de Sistemas e Robótica
Instituto Superior Técnico
Lisboa, Portugal

LIRA-Lab, DIST
University of Genoa
Genoa, Italy

Abstract. This work presents an algorithm to track planar surfaces by registering log-polar images. We start by motivating the choice of the planar model and the log-polar geometry. Tracking is important for many robotic applications such as navigation, ego–motion estimation, vehicle stabilization, mosaic creation. We describe the developed algorithm and present simulation results of a pan and tilt camera translating freely in 3D and observing a planar surface. The system keeps track of the position and orientation of a predefined template and controls the gaze direction towards it. Finally we describe future working directions that envision to relax the planarity constraint, taking advantage of the log-polar imaging geometry.

1 Introduction

Visual fixation systems have increasingly important applications in robotics. The attempt to use robots on dynamic environments raised the need for better sensors and algorithms to keep track of the changes occurring in the external environment. Ranging from underwater vehicles or airships that must dock or keep their positions relative to a static surface despite external disturbances [14], to land robots that must follow or avoid moving objects [10], visual fixation can play an important role.

In this paper we present a system capable of fixating a planar surface and keep track of its rigid motion along time. Planar surfaces can be found in many human made environments and represent good approximations for other kind of surfaces. For example, in aerial cartography or ocean floor exploration, the ground can be approximated by a plane if the camera is distant enough [8]. Therefore, planar surfaces provide an adequate and yet manageable model to work with. Notwithstanding, a future goal of this work is to be able to cope with more general (smooth) surfaces and relax the rigid planarity constraint.

Many researchers have proposed methods to track the relative motion of planar patches [11]. Many rely on optic flow techniques [2] and often use affine, scaled–euclidean or 2D–rigid models [9] that are approximations to the planar (projective) model. The use of these models can be justified when the images do not contain strong perspective effects. Instead, our method copes with the general projective model.

Optic flow methods are often based on the registration of image patches acquired in successive time instants [1]. Therefore they are likely to accumulate errors along time and present a significant drift from the expected solution. Our method is based on an initial template that is registered with the incoming images at every time step. Thus, in principle, this method is able to estimate the relative motion of the object with higher precision.

The log-polar image geometry was first motivated by its resemblance with the structure of the retina of some biological vision systems and by its data compression qualities [12]. When compared to the usual cartesian images, the log-polar images allow faster sampling rates on artificial vision systems without reducing the size of the field of view and the resolution on the central part of the retina (fovea) [13]. In the last years, however, it has been noticed that the log-polar geometry also provides important algorithmic benefits [5]. For instance in [3], it is shown that the use of log-polar images increases the size range of objects that can be tracked using a simple translation model. We expect that increasing the "order" of the transformation towards the planar model, these advantages can still be observed.

This paper is organized in the following way: Section 2 will briefly describe the general form of the *difference decomposition* method for image registration, where the usual cartesian representation of images is considered, for the sake of simplicity. Then, in Section 3 we summarize the log-polar image geometry and the method used to warp the images according to planar transformations. The conversion of the *difference decomposition* method to log-polar images and the structure of our particular algorithm are described in Section 4. Simulation results are presented in Section 5, and finally in Section 6 we draw some conclusions and envision future developments.

⋆ alex@isr.ist.utl.pt
⋆⋆ jasv@isr.ist.utl.pt
⋆⋆⋆ sandini@dist.unige.it

† This work was partially funded by projects ESPRIT-LTR 30185, NARVAL and VIRGO-TMR FMRX-CT96-0049.

2 The *difference decomposition* registration method

The *difference decomposition* method was proposed in [7], as a registration method for planar patches in a sequence of images. Given a reference image patch I_0 and a target image patch I_1, the registration problem is that of computing the transformation $(x', y') = \omega(x, y)$ that maps the reference image patch onto the specified region of the target image, such that $I_1(x, y) = I_0(\omega(x, y))$. Usually, these transformations, are parameterized as a function of a vector $\mathbf{q}$. The planar transformation depends on 8 parameters, $\mathbf{q} = [q_1 \ldots q_8]^T$:

$$x'(x, y; \mathbf{q}) = \frac{q_1 x + q_2 y + q_3}{q_7 x + q_8 y + 1}, \quad y'(x, y; \mathbf{q}) = \frac{q_4 x + q_5 y + q_6}{q_7 x + q_8 y + 1} \tag{1}$$

We assume an initial estimate $\mathbf{q}_0$ for the transformation between the two images such that we only need to estimate a residual transformation $\Delta\mathbf{q}$. With $\mathbf{q}_0$, we can warp image I_1 using the inverse transform ω^{-1} and obtain image I_1'. A *difference template* $B(\Delta\mathbf{q})$ is defined as:

$$B(\Delta\mathbf{q}) = I_0(\Delta\mathbf{q}) - I_0 \tag{2}$$

where $I_0(\Delta\mathbf{q})$ is the reference image warped by the transformation expressed in vector $\Delta\mathbf{q}$. The main assumption of the *difference decomposition* method is that for small $\Delta\mathbf{q}$ the difference template $B(\Delta\mathbf{q})$ can be expressed by linear combination of a set of *sample templates* computed *a priori* $B_i = B(\Delta\mathbf{q}_i)$:

$$B(\Delta\mathbf{q}) \approx \sum_{i=1}^{m} k_i B_i \quad \text{for} \quad \Delta\mathbf{q} = \sum_{i=1}^{m} k_i \Delta\mathbf{q}_i \tag{3}$$

The set of *sample vectors* $\{\Delta\mathbf{q}_i\}$ should form a basis for the considered vector space ($\Re^8$ in our case), and can be chosen among likely deformations that occur in a particular problem. For instance, if we have a camera approaching a surface perpendicularly, we should include sample vectors that express image expansion.
To register the two image regions, one can minimize an appropriate error function, such as the sum-of-squared-differences ($L2$-error criterion). Considering images as column vectors the $L2$ error function is given by :

$$e(\mathbf{k}) = \frac{1}{2} \parallel I_1' - \bar{I}_0(\mathbf{k}) \parallel^2 = \frac{1}{2} \parallel D - \bar{B}(\mathbf{k}) \parallel^2 \tag{4}$$

where D is the difference image $I_1' - I_0$, $\mathbf{k} = [k_1 \ldots k_m]^T$ and $\bar{B}(\mathbf{k}) = B(\sum_{i=1}^{m} k_i \Delta\mathbf{q}_i)$. Introducing the *difference decomposition* assumption (3) the error function takes the form:

$$e(\mathbf{k}) = \frac{1}{2} \parallel D - \mathbf{B}\mathbf{k} \parallel^2 \tag{5}$$

where $\mathbf{B}$ is the matrix $[B_1 \ldots B_m]$. Minimizing this cost function we can determine the vector $\mathbf{k}$:

$$min_k \parallel D - \mathbf{B}\mathbf{k} \parallel^2 \quad \Rightarrow \quad \mathbf{B}^T D - \mathbf{B}^T \mathbf{B}\mathbf{k} = 0 \tag{6}$$

Since the sample templates $\mathbf{B}$ can be computed *a priori* (once the reference template is known), most of the computation can be done off–line. One on-line iteration of the algorithm only requires matrix multiplications and one image warping. Thus, it is appropriate for real–time implementations. After determining $\mathbf{k}$, the solution for $\Delta\mathbf{q}$ can be calculated using the linear combination in eq. (3). Using the updated estimates of $\Delta\mathbf{q}$, the algorithm can proceed iteratively until a stopping criteria is met, and use different resolutions of an image pyramid architecture.
The advantages of this method over standard local gradient descent methods is the ability to customize the set of sample vectors according to the kind and range of expected image deformations. Also with the increase of computational power, we can easily add new sample vectors to improve the estimation results. As we will describe in Section 4, the algorithm can be customized to estimate the larger and more constrained motions in the first iterations and the finer and more generic transformations in the last iterations, which greatly improves its robustness.

3 Log-polar image warper

The system described in this paper is based on a log-polar space variant image sampling. One advantage of this kind of sampling is data reduction. This is obtained by reducing the resolution at the image periphery, thus high resolution in the fovea improves the performance of the tracking process. We map 128x128 cartesian images to 64x32 log-polar images and achieve 8 times increase in efficiency (both storage and speed).
The log-polar transformation is a conformal mapping from the points on the *cartesian* plane (x, y) to points in the *log-polar* plane (ξ, η) [12]. The mapping is described by :

$$\xi(x, y) = \log(\sqrt{x^2 + y^2}) \quad \eta(x, y) = \arctan\frac{y}{x} \tag{7}$$

For tracking, the main advantage of the log-polar geometry is that objects occupying the central high resolution part of the visual field become dominant over coarsely sampled background elements in the image periphery. This embeds an implicit focus of attention in the center of the visual field where the target is expected to be most of the time.
In order to use the *difference decomposition* method we need a fast way to warp the log-polar images according to a generic planar transformation. We use the *Intel Image Processing Library* [6] to perform the warping. This library contains optimized code with MMX instructions and performs linear or bicubic interpolation to avoid aliasing in the generated images. The library function requires, as input, a warping field computed by:

$$\omega_\xi(\xi,\eta) = \xi(x',y') \quad \omega_\eta(\xi,\eta) = \eta(x',y') \tag{8}$$

where x' and y' are given by Eq. (1). An example of a transformation including translation and rotation is shown in Fig. 1. Notice that the singularity in the center of the original image does not appear in the warped image because it is interpolated. Each image warping takes about 10 msec. on a Pentium 400 Mhz computer, including the computation of the warping maps $\omega_\xi(\xi,\eta)$ and $\omega_\eta(\xi,\eta)$.

Fig. 1.: The original log-polar (on the left) is warped according to a transformation that includes translation and rotation (second from the left). The corresponding retinal images are also shown (right).

4 Algorithm implementation

To implement an algorithm based on the *difference decomposition* method, some design choices must be taken, such as: (i) the number and distribution of sample vectors $\Delta\mathbf{q}_i$; (ii) the iterative (or not) structure of the algorithm, i.e. the number of iterations and/or the stopping criteria; (iii) the number of scales of a possible image pyramid architecture and (iv) the choice of the reference template.
Since we are interested in a real-time implementation of the algorithm, it is more important to have a fixed computation time than a very precise tracking, therefore we fix the number of iterations of the algorithm to 3. Also we opted not to use a image pyramid and perform all the computations in only one scale.
Regarding the choice of the sample vectors, and after many experiments we chose to create three sets: one with sample translation vectors, one with sample affine vectors and one with sample planar vectors. Each set is composed by 48 vectors with non–uniform distributions, sampling more densely small deformations but still considering large deformations. For instance, the sample translation set is composed by vectors that translate template by the amounts $(x,y) \in \{-6,-3,-1,1,3,6\} \times \{-6,-3,-1,1,3,6\}$. The idea is to have good precision when the deformations are small but still be able to detect large deformations. The three iterations are organized in the following way:

- The first iteration uses the **sample translation vectors**. Since in the beginning a large deformation is likely to exist, it is more robust to estimate more constrained transformations. This iteration is intended to center the template with the current image and leave to the next iterations the estimation of the remaining deformations.
- The second iteration uses the **sample affine vectors**. This iteration should estimate most of the rotation, scaling and shear present in the transformation.
- The last iteration uses the **sample generic planar vectors**. It should estimate the remaining deformations and make the final fine adjustment to the template. Since this set spans 8 degrees of freedom, it should be used only with small deformations, otherwise it is likely to produce erroneous results.

For more details and experiments on the choice of the sample vectors, check [4].
The last point relates to the choice of the reference template. For our problem is sufficient to start with the image observed in the initial position. However, when we transform this reference image, part of the transformed image has no information (see Fig.1). Thus, to compute the difference image $D = I_0 - I_1'$ and the sample templates B_i (see section 2), we represent images without the 4 outer rings (see Fig. 2). This means that if the template is not translated more than 20 pixels from the center of the image, then this *border effect* does not affect the computations, otherwise we say that the template is outside the *safety region* and the performance of the algorithm can be degraded.

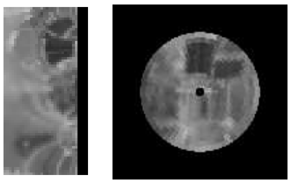

Fig. 2.: One of the translation sample templates (left). The corresponding retinal image

5 Results

The experiments presented in this paper are obtained by simulating the motion of a camera observing a static planar surface. The texture mapped on the surface and the image observed by the camera in the initial position are presented in Fig. 3. This last image is selected as the template to be registered with the observed images at each time step.

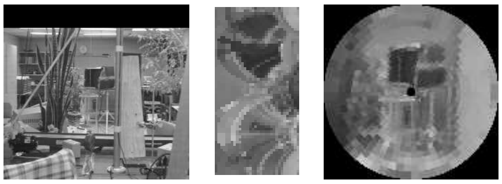

Fig. 3.: Left: image used as texture map to simulate a planar surface. Center: image observed by the log-polar camera in the initial position. Right: the corresponding retinal image.

We present three sets of experiments. In the first one (performance evaluation) we simulate camera motions that produce, in the retinal plane, increasing image translations, rotations and scalings. This experiment shows the range limitations of the algorithm, i.e., the maximal image deformations allowed between two consecutive images. In the second set of experiments (open–loop tracking), the camera translates in 3D and registers the template with the acquired images, keeping track of its position. Since the template progressively leaves the visual field, the displacement is limited. Instead, in the third set of experiments (closed–loop tracking), we also control the pan and tilt angles of the camera in order to gaze towards the center of the template, allowing bigger camera displacements.

5.1 Performance evaluation

In these experiments we evaluate the algorithm ability to track translations, rotations and scalings of the template. The algorithm is applied to increasingly bigger deformations and at each iteration the estimated and real transformations are compared. The comparison criteria is given by the L2–norm of the vector that contains the displacements of the corners of the polygonal window that contains the template. The results are presented in Fig.4.

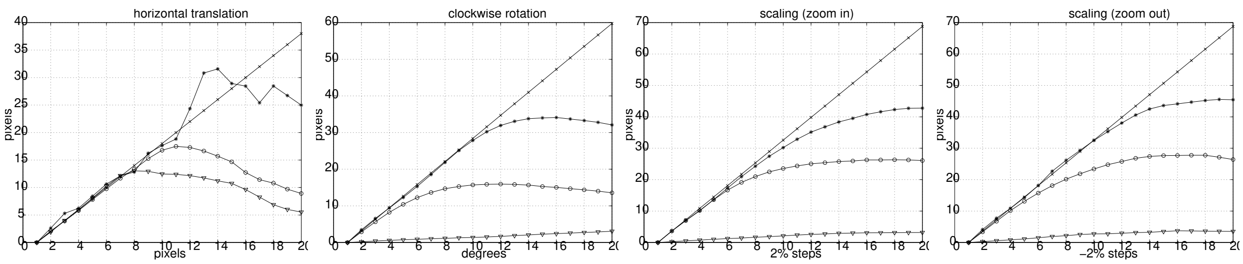

Fig. 4.: Performance of the algorithm for translation, rotation, zoom–in and zoom–out transformations. × – ground truth. ∇ – first iteration (translation sample vectors). ∘ – second iteration (affine sample vectors). ∗ – third iteration (planar sample vectors)

With the images and transformations used, the algorithm is able to estimate with precision (error less than half pixel per window corner) about 10 pixels translation, 11 degrees rotation, 18% zoom–in or 24% zoom–out. It is interesting to note that the sample translation vectors do most of the work when the deformation consists solely of translation but are almost useless when the deformation does not include translations. However, for the problems we are interested with, translation is a dominant deformation and is essential to include translation sample vectors in the sample set.

5.2 Open–loop tracking

In this set of experiments the camera translates in 3D and the template is tracked in a covert fashion, i.e., the camera motion is not controlled to direct gaze towards the template but the system knows at any time instant the transformation that relates the original and current pose of the template. The camera performs an elliptic trajectory, illustrated in Fig. 5. The camera trajectory produces at most 20 pixel translation of the template in the image plane.

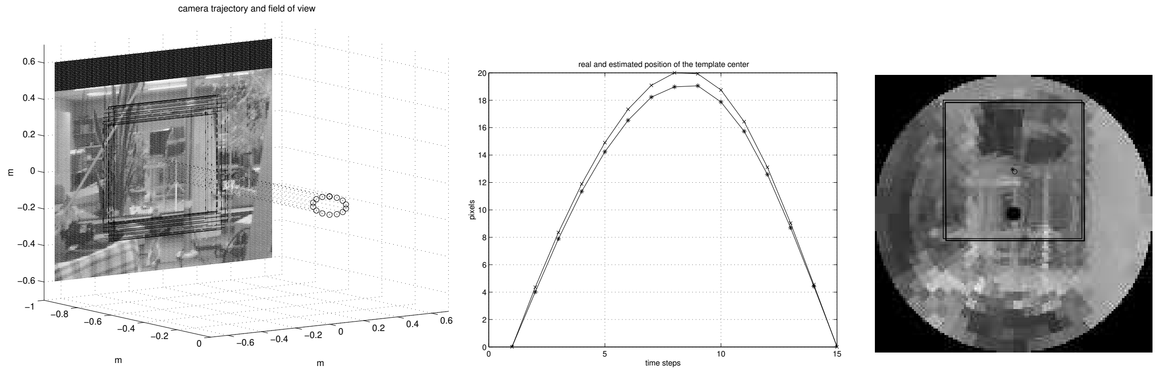

Fig. 5.: Open–loop tracking experiment. Left: The camera performs an elliptic trajectory in 3D without changing its orientation. The rectangles on the planar surface represent the field of view at each time step. Center: Real ($\times$) and estimated ($*$) displacements of the template's center at each time step. Right: situation with bigger error (frame 8).

This is a requirement to keep the template completely inside the *safety region* (see Section 4). The real and estimated displacements of the template's center are also shown in Fig. 5. As can be observed, the tracking error is at most 1 pixel, at the limit of the *safety region*.

5.3 Closed-loop tracking

In this set of experiments we are able to control the camera's pan and tilt degrees of freedom and we use this ability to keep the template centered in the image. We assume that the velocity of each joint follows a simple first order dynamic model, with a time constant of 200 msec, while the sampling frequency is assumed to be 10Hz (100 msec). These parameters define a relatively slow dynamics. The reference velocity input to the pan and tilt motors is proportional to the vertical and horizontal position of the estimated template's center, respectively. Again, the camera performs an elliptic trajectory, 4 times larger than before, as shown in Fig. 6. The camera moves during the first 30 time steps and remains still during the remaining time. By actively tracking the template it remains almost always inside the *safety region* and bigger displacements for the camera are allowed. The real and estimated displacements of the template's center are also shown in Fig. 6. It is interesting to note that between frames 24 and 28 the template is out of the *safety region*, but the algorithm is still able to estimate its position without much error.

6 Conclusions and future work

This paper presented a system capable of tracking a surface undergoing planar deformations. The method is based on the registration of a given initial template using the *difference decomposition* algorithm, adapted to log-polar images. Our results show that registration is performed with very good precision. Although for active tracking purposes, precision may not be a strong requirement, the method has the advantage of not accumulating errors along time, in opposition to usual optic flow algorithms. Also, precise registration can be used to obtain precise pose or ego-motion measurements, useful in auto–localization applications.

The use of log–polar images reduces the computation time and memory storage needs. Additionally, the log-polar geometry implements an implicit focus of attention in the image center. Although not evident from the results shown

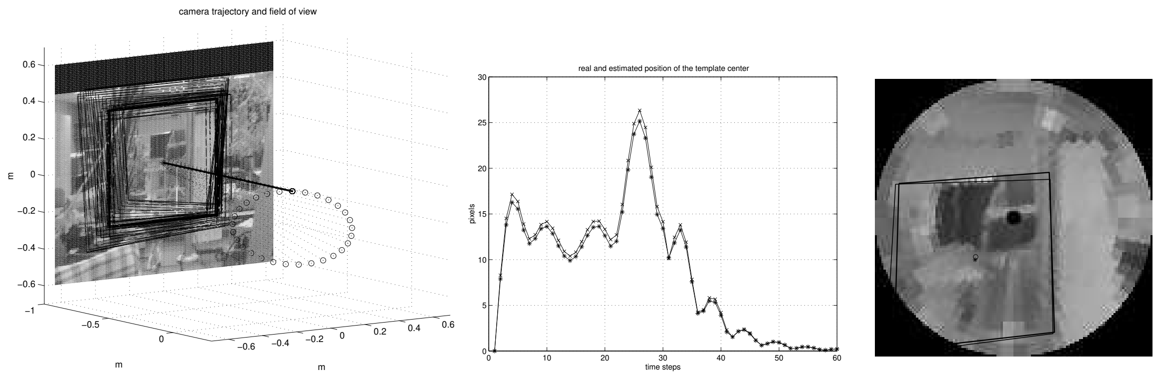

Fig. 6.: Closed–loop tracking experiment. Left: The camera performs an elliptic trajectory in 3D without controlling its orientation towards the template's center. The rectangles on the planar surface represent the field of view at each time step. Center: Real ($\times$) and estimated ($*$) displacements of the template's center at each time step. Right: situation with bigger error (frame 26).

in this paper, the log–polar geometry reduces the effect of image artifacts not in the center of the image. Therefore, once tracking a planar object, even if the background does not follow the planar model, motion estimation should perform better with log–polar than cartesian images. These topics will be addressed in future work.

References

1. J. Barron, D. Fleet, and S. Beauchemin. Performance of optical flow techniques. *IJCV*, 12(1):43–78, February 1994.
2. J. Bergen, P. Anandan, K. Hanna, and R. Hingorani. Hierarchical model-based motion estimation. In *Proc. ECCV*, pages 237–252, Santa Margherita Ligure, Italy, May 1992.
3. A. Bernardino and J. Santos-Victor. Binocular visual tracking: Integration of perception and control. *IEEE Transactions on Robotics and Automation*, 15(6), December 1999.
4. Alexandre Bernardino. Personal web page. http://www.isr.ist.utl.pt/ alex.
5. C. Capurro, F. Panerai, and G. Sandini. Dynamic vergence using log-polar images. *IJCV*, 24(1):79–94, August 1997.
6. Intel Corporation. Intel image processing library. http://developer.intel.com/vtune/perflibst/IPL.
7. M. Gleicher. Projective registration with difference decomposition. In *Proc. CVPR'97*, pages 331–337, June 1997.
8. N. Gracias and J. Santos-Victor. Trajectory reconstruction using mosaic registration. In *Proc. SIRS'99*, Coimbra, Portugal, July 1999.
9. C. Morimoto and R. Chellappa. Fast electronic digital image stabilization. In *Proc. ICPR*, Vienna, Austria, August 1996.
10. G. Salgian and D. Ballard. Visual routines for vehicle control. In D. Kriegman, G. Hager, and S. Morse, editors, *The Confluence of Vision and Control*. Springer Verlag, 1998.
11. J. Santos-Victor and G. Sandini. Visual behaviors for docking. *CVIU*, 67(3), September 1997.
12. E. Schwartz. Spatial mapping in the primate sensory projection : Analytic structure and relevance to perception. *Biological Cybernetics*, 25:181–194, 1977.
13. C. Weiman. Log-polar vision for mobile robot navigation. In *Proc. of Electronic Imaging Conference*, pages 382–385, Boston, USA, November 1990.
14. H. Zhang and J. Ostrowski. Visual servoing with dynamics: Control of an unmanned blimp. In *Proc. of ICRA'99*, Detroit, USA, May 1999.

This article was processed using the TEX macro package with SIRS2000 style