An Optimization Approach for Translational Motion Estimation in Log-Polar Domain*

V. Javier Traver and Filiberto Pla

Dept. Llenguatges i Sistemes Informàtics · Universitat Jaume I
Edifici TI · Campus Riu Sec · E12071 Castelló (Spain)
{vtraver|pla}@uji.es

Abstract. Log-polar imaging is an important topic in space-variant active vision, and facilitates some visual tasks. Translation estimation, though essential for active tracking, is more difficult in (log-)polar coordinates. We propose here a novel, conceptually simple, effective, and efficient method for translational motion estimation. It is based on a gradient-based minimization procedure. Experimental results with log-polar images using a software-based log-polar remmapper are presented.

Keywords: motion estimation, log-polar domain

1 Introduction

Recently, an increasing attention has being paid to the concept of *active* vision, and to the closely related topic of space-variant imaging. Space-variant images have an area of high-visual acuity at its center, and a decreasing resolution towards the periphery. This results in a trade-off between a wide field of view, a high resolution, and a small fast-to-process output. The log-polar geometry is by far the most often used space-variant model [4] because of its interesting properties in fields such as pattern recognition and active vision. In the latter case, it has been proven its usefulness for time-to-impact computation [9], for active docking in mobile robots [2], and for vergence control [3,5] to name but a few.

Despite its obvious advantages in some problems, log-polar space also complicates some other visual tasks. Estimating image-plane translation, for instance, becomes more difficult in log-polar domain than in cartesian coordinates. Many researchers use stereo configurations (e.g., [3], [5]), and a few of them address the problem of motion estimation using a single camera for tracking in the log-polar domain [8,1]. These approaches are computationally expensive or conceptually difficult (Okajima *et al.* use complex wavelets for motion estimation [8]), or have some limitations (Arhns and Neumann control only the pan angle of a pan-tilt

* This work has been partially supported by projects GV97-TI-05-27 from the *Conselleria d'Educació, Cultura i Ciència, Generalitat Valenciana*, and CICYT TIC98-0677-C02-01 from the Spanish *Ministerio de Educación y Cultura*.

W. Skarbek (Ed.): CAIP 2001, LNCS 2124, pp. 365–373, 2001.

head [1]). We propose a simple and effective method for translation estimation, which is based on a gradient-based minimization procedure. In the following sections we show the method and the experimental results obtained. The technique is intended to be used for active object pursuit with monocular log-polar images.

2 Optimization-Based Motion Estimation

The log-polar transform. The log-polar transform we use here [7] defines the log-polar coordinates as $(\xi, \eta) = \left(\log_a\left(\frac{\rho+\rho_0}{\rho_0}\right), \theta\right)$, where (ρ, θ) are the polar coordinates, defined as usual, and a and ρ_0 being parameters, which are found from the selected log-polar image resolution ($R \times S$), i.e., the number of rings (R) and sectors (S). The log-polar transformation of a cartesian image I will be denoted by $\mathcal{L}(I)$ or $\mathcal{I}$. An example of log-polar images can be seen in figure 1.

The role of the correlation measure. The correlation measure between two stereo log-polar images for varying values of the vergence angle has a profile with interesting properties, which are not present when correlation is computed between cartesian images [5,3]. On the one hand, a deep global minimum occurs at the correct vergence configuration. On the other hand —and more importantly— the correlation profile over the vergence angle range is unaffected by false local minima in the case of log-polar mapping while it behaves poorer in uniformly sampled cartesian images.

In the context of monocular fixation, Ahrns and Neumann present a similar idea [1] using log-polar images too. In this case, a gradient-descent control is proposed for controlling the pan degree of freedom of a camera mount. However, they do not show why a gradient-based search is appropriate. Moreover, at some point in their algorithm, they use the original cartesian images rather than the log-polar remapped ones. This has, at least, two inconveniences. On the one hand, it is not a biologically plausible option[1]. On the other hand, doing this may involve an additional computational cost which is opposed to the interesting data reduction advantage attributed to discrete log-polar images.

In contrast, our contributions are as follows:

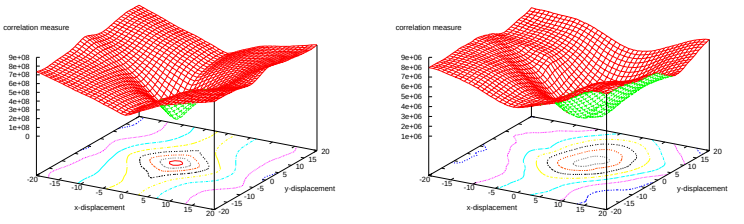
1. We first show that the idea used for vergence control can also be exploited in monocular tracking in log-polar space. The choice of a steepest-descent control is therefore justified by the shape of the resulting correlation surfaces.
2. We stick to the log-polar images as if they were directly grabbed from a hardware log-polar sensor. Therefore, after the log-polar mapping, no use at all is made of the cartesian images.
3. We propose an algorithm for translational motion estimation which lends itself for controlling both the pan and tilt angles of a camera mount.

[1] One may simulate by software the output of a retina-like sensor, but it is probably less acceptable to make use of the original cartesian images for purposes other than the log-polar mapping.

(a) A (b) B (c) $I_t = A_B(x_0, x_0)$ (d) $\mathcal{I}_t$ (e) $\mathcal{I}_{t+1}$

Fig. 1. A foveated target has undergone a retinal shift $(x_0, y_0) = (5, 5)$

In [5,3] the correlation index $\mathcal{C}$ for the vergence angle ψ in a certain range, gives rise to a 1-D function, $\mathcal{C}(\psi)$. Our approach extends this idea to 2-D function (the correlation surface), which is dependent on the translational components in x and y direction, x_0 and y_0, respectively, $\mathcal{C}(x_0, y_0)$. We can compute the value of a correlation measure $\mathcal{C}$ between one image and versions of this one shifted by (x_0, y_0). Figure 2 shows an example of the resulting surface $\mathcal{C}(x_0, y_0)$ computed in cartesian (figure 2(a)) and log-polar (figure 2(b)) domains for the same images. As can be seen in the figure, the surfaces shape is quite smooth. Both surfaces also exhibit a distinguishable minimum. In the case of log-polar domain, the location of such a minimum is very close to the actual translational motion undergone in image plane ($x_0 = 5, y_0 = 5$). In the case of cartesian images, however, this minimum is near $(0, 0)$, which are clearly wrong motion parameters. These surfaces have been obtained from the images shown in figure 1, where a small image patch (figure 1(b)) is pasted on the image in figure 1(a) at its center (figures 1(c) and 1(d)) and in a shifted position (figure 1(e)). This simulates the projection onto the image plane of a target motion.



(a) cartesian (b) log-polar

Fig. 2. Example of correlation surface in cartesian and log-polar spaces

As we have shown, the minimum of the correlation surface computed on log-polar images occurs at the correct displacement of the target. Additionally, in log-polar domain, when compared to cartesian case, the surface is somewhat

smoother and free of local minima and inflection points (which are present in the cartesian case, see figure 2(a)). This shows one of the most interesting properties of the log-polar geometry, which makes it advantageous over cartesian images for tracking purposes. The logarithmically sampled radial coordinate offers a built-in *focus-of-attention* mechanism, by means of which pixels at the center of the image are emphasized over outer pixels. As a result, information far from the fovea are much less distracting in log-polar case than in uniformly sampled images. This is so because, in uniformly resolved images, every pixel contributes the same to the correlation function, regardless of their position. In log-polar coordinates, however, Bernardino and Santos-Victor showed [3] that each pixel contribution is weighted by an amount inversely proportional to the squared distance of the pixel to the image centre. Thus, in the example shown in figure 1, the static background influences little over the shifted, but foveated, object.

The algorithm. To estimate the translation parameters (x_0, y_0), our approach consists of finding the location of the minimum of the correlation measure $\mathcal{C}(x'_0, y'_0)$. The shape of the correlation surfaces seen in section 2 suggests that a gradient-based search could be an adequate search algorithm. The estimation at iteration k, (x_0^k, y_0^k), is updated from the estimation at the previous iteration, (x_0^{k-1}, y_0^{k-1}), by using the gradient, $\nabla\mathcal{C}$, of the correlation measure, as the most promising direction to move, i.e.:

$$(x_0^k, y_0^k) = (x_0^{k-1}, y_0^{k-1}) - g(\nabla\mathcal{C}) \tag{1}$$

A common definition for $g(\cdot)$ is $g(\nabla\mathcal{C}) = \delta \cdot \frac{\nabla\mathcal{C}}{|\nabla\mathcal{C}|}$, i.e., consider the unit vector in the direction of the gradient and move a certain amount δ in that direction. The question of choosing the value for δ usually involves a trade-off. Then, an adaptive, rather than a fixed step, is called for. One possibility consists of moving "large" steps when far from the minimum, and "small" steps closer to it. This is the idea we use: initially $\delta = 1$ and whenever the minimum is surpassed the value of δ is halved. The search may be stopped using some criteria such as that δ is smaller than a given threshold $\delta_{\min}$, or that the search has reached a maximum number of iterations $k_{\max}$. The steps of the whole process are presented in algorithm 1. This is a basic formulation over which some variations are possible.

The computational cost of the algorithm is $\mathcal{O}(K)$, where K denotes the actual number of iterations. In turn, K depends on the motion magnitude, and on the particular convergence criteria. In our case, the main factor is the image displacement. Therefore, as we expect only small translations of the target onto the image plane, the efficiency of the algorithm is guaranteed. Regarding the initial guess, (x_0^0, y_0^0), in the absence of any other information, it is sensible to use $(0, 0)$. However, during an active tracking, it may be expected the target to move following some dynamics. Therefore, if the target motion can be predicted, we can use the expected future target position as the initial guess for the algorithm. This would help speed up the motion estimation process.

Algorithmus 1 Gradient-descent-based translation estimation of log-polar images

Input: Two log-polar images $\mathcal{I}$ and $\mathcal{I}'$
Output: The estimated translation vector (x_0, y_0)

$(x_0^0, y_0^0) \leftarrow (0,0)$ { *initial guess* }
$k \leftarrow 0$ { *iteration number* }
$\delta \leftarrow 1$
while $(\delta > \delta_{\min}) \wedge (k < k_{\max})$ **do**
  $k \leftarrow k + 1$
  $(x_0^k, y_0^k) \leftarrow (x_0^{k-1}, y_0^{k-1}) - g(\nabla \mathcal{C})$ { *estimation update rule* }
  if minimum surpassed **then**
    $\delta \leftarrow \delta / 2$
  end if
end while
$(x_0, y_0) \leftarrow (x_0^k, y_0^k)$

Computing the correlation gradient. To evaluate the equation 1 we need a way to compute the gradient. In the case of the well-known SSD (sum of squared differences) correlation measure [6], the gradient $\nabla \mathcal{C} = (\mathcal{C}_{x_0}, \mathcal{C}_{y_0})$ becomes:

$$\begin{cases} \mathcal{C}_{x_0} = \dfrac{\partial \mathcal{C}}{\partial x_0} = 2 \displaystyle\sum_{(\xi,\eta)\in\mathcal{D}} \Big\{ \big(I'(\xi',\eta') - I(\xi,\eta)\big) \cdot I'_{x_0}(\xi',\eta') \Big\} \\ \mathcal{C}_{y_0} = \dfrac{\partial \mathcal{C}}{\partial y_0} = 2 \displaystyle\sum_{(\xi,\eta)\in\mathcal{D}} \Big\{ \big(I'(\xi',\eta') - I(\xi,\eta)\big) \cdot I'_{y_0}(\xi',\eta') \Big\} \end{cases} \tag{2}$$

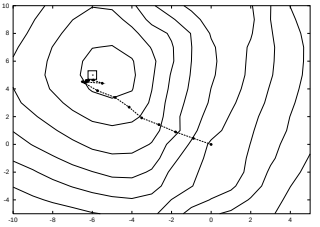
with $\mathcal{D}$ being a certain set of image pixels (usually, the entire image), and where $I'_{x_0} = I'_{x_0}(\xi',\eta')$ and $I'_{y_0} = I'_{y_0}(\xi',\eta')$ are:

$$\begin{cases} I'_{x_0} = I'_{\xi'} \cdot \xi'_{x_0} + I'_{\eta'} \cdot \eta'_{x_0} \\ I'_{y_0} = I'_{\xi'} \cdot \xi'_{y_0} + I'_{\eta'} \cdot \eta'_{y_0} \end{cases} \tag{3}$$

with $\xi'_{x_0} = \xi_x$, $\xi'_{y_0} = \xi_y$, $\eta'_{x_0} = \eta_x$, and $\eta'_{y_0} = \eta_y$. The common notation for partial derivatives, $f_z = \partial f / \partial z$, is used. On the other hand, $(\xi',\eta') = (\xi + \xi_0, \eta + \eta_0)$, where the increments in log-polar coordinates, ξ_0 and η_0, due to a translational cartesian displacement (x_0, y_0) depend on the position in the log-polar space, and can be approximated as:

$$\begin{cases} \xi_0 \approx \xi_x \cdot x_0 + \xi_y \cdot y_0 \\ \eta_0 \approx \eta_x \cdot x_0 + \eta_y \cdot y_0 \end{cases} \tag{4}$$

By taking the partial derivatives of ξ and η as defined above, with respect to x and y, we get ξ_x, ξ_y, η_x, and η_y, as follows:



iso-correlation contour lines and descent path

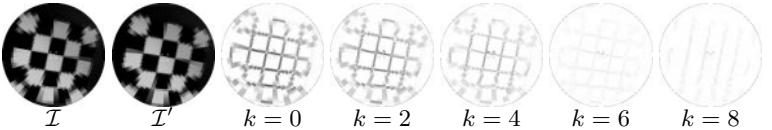


Fig. 3. The input images, $\mathcal{I}$ and $\mathcal{I}'$, to the algorithm ($I'(x,y) = I(x-x_0, y-y_0)$); the iso-correlation contour lines, and the descent followed by the algorithm (the square represents the true motion parameters, $(x_0 = -6, y_0 = 5)$); and the absolute difference $|\mathcal{I}_k - \mathcal{I}'|$ at selected iterations k.

$$\begin{cases} \xi_x = \dfrac{\partial \xi}{\partial x} = \dfrac{\cos\theta}{(\rho+\rho_0)\ln a} \\ \\ \xi_y = \dfrac{\partial \xi}{\partial y} = \dfrac{\sin\theta}{(\rho+\rho_0)\ln a} \end{cases} \qquad \begin{cases} \eta_x = \dfrac{\partial \eta}{\partial x} = -\dfrac{\sin\theta}{\rho} \\ \\ \eta_y = \dfrac{\partial \eta}{\partial y} = \dfrac{\cos\theta}{\rho} \end{cases} \tag{5}$$

3 Experimental Results

Qualitative analysis. Figure 3 illustrates graphically the workings of the technique. Let $\mathcal{I}_k = \mathcal{L}(I_k)$ be the image at iteration k of the process (see algorithm 1), i.e., $I_k(x,y) = I(x - x_0^k, y - y_0^k)$. As the descent proceeds, $\mathcal{I}_k$ is more and more similar to $\mathcal{I}'$. This can be appreciated by observing that the difference image $|\mathcal{I}_k - \mathcal{I}'|$ becomes whiter and whiter (the whiter a pixel is, the less the gray level difference between the two images). Notice that $\mathcal{I}_k$ is computed here for illustrating purposes, but it is not computed during the actual process.

Quantitative analysis. For each image I_i in our image test set, a set of displacements $\{(x_0, y_0)\}$ were applied. Each of these translations correspond to a translation magnitude m_j (cartesian pixels) in a set M_t and to a translation orientation o_k (direction) in a set O_t. Let n_m and n_o be the size of sets M_t and O_t, respectively. The elements $m_j \in M_t$ are selected to range from small to large motions. Then, the elements $o_k \in O_t,\ k \in \{0, 1, \ldots, n_o - 1\}$ are equally

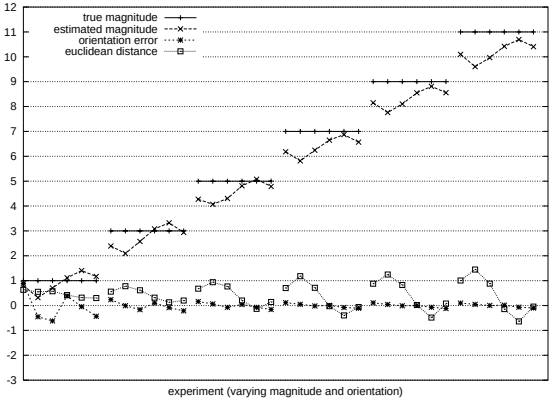


Fig. 4. True and estimated motion parameters, and error measures between them

spaced and selected to cover the span of possible orientations from 0 to 2π radians, i.e., $o_k = 2\pi k/n_o$ rad. For each combination of magnitude and direction of motion, the translated version of I_i, $I_i^{j,k}$, is computed, and their log-polar transformations, $\mathcal{I}_i$ and $\mathcal{I}_i^{j,k}$, are input to the algorithm. The output of the algorithm (the estimated translation) is compared to the known motion parameters.

Figure 4 plots the estimation errors for image *Barche* (one of the test images). The tested translations are computed from the particular set $M_t = \{1, 3, 5, 7, 9, 11\}$ and $n_o = 6$. The horizontal axis of the plot represents each experiment with a given combination of motion magnitude and direction. In the vertical axis, different (ground-truth and estimated) parameters and error measures are represented. From this graphic, several observations can be made. First of all, the estimated motion magnitude differs little from the true motion magnitude. The difference is less than a pixel for the smaller displacements, and a little bigger for larger ones. Secondly, the orientation error becomes smaller with increasing magnitude. Big angular errors (in radians) occur only when translations are small. Lastly, the Euclidean distance between the true and estimated (x_0, y_0), reveals how small the overall motion estimation error is. The error is less than a pixel for small-medium displacements (up to 5 pixels), and about one pixel and a half for big displacements (more than 5 pixels).

Table 1 gives some statistical results of the Euclidean distance as an error measure. The average error ranges from less than one pixel to ≈ 2.5 pixels in an extreme case, depending on the image tested. It is important to notice that the mean errors in table 1 are somewhat biased by the big estimation errors resulting in the case of large motion displacements. Thus, the median is a better global measure. Maximum errors can be large; but these errors occur when translations are also large. In these cases, the global minimum in the correlation surfaces is far from the initial guess (x_0^0, y_0^0). The shape of the correlation surface far from the global minimum may make difficult for a gradient-descent search to succeed.

The overall results are quite interesting in the context of an active tracking task, where only small displacements are expected within the fovea.

Table 1. Some statistics about the Euclidean distance as an error measure

Image	Mean	Std. Dev.	Median	Minimum	Maximum
Barche	1.33	0.62	1.24	0.27	2.70
Lena	2.46	1.62	2.07	0.30	7.11
Grid	1.00	0.52	0.95	0.25	2.49
Oranges	1.13	0.99	0.77	0.21	4.54
Lab	0.90	0.38	0.96	0.04	1.84
Rubic	1.10	1.35	0.79	0.14	6.50
Face	1.60	0.87	1.56	0.23	3.76
Corridor	1.07	0.42	1.00	0.43	2.33
Scream	0.98	0.69	0.85	0.13	3.32

4 Conclusions

We have shown that the correlation surface computed between two log-polar images (in which a foveated target has moved) over a range of displacements along the x and y axes, presents a minimum at the actual displacement undergone by that target. This interesting property does not hold in case of cartesian images, which demonstrates the superiority of log-polar domain over cartesian geometry for monocular object tracking.

We have developed a gradient-descent-based algorithm for searching the global minimum of the correlation function, as a means to estimate a translational motion. Experimental results of this approach demonstrate its feasibility. An analysis of the estimation errors reveals that the method works best with small-medium displacements, which is the case in foveated active tracking.

References

1. Ahrns, I., Neumann, H.: Real-time monocular fixation control using the log-polar transformation and a confidence-based similarity measure. In: Jain, A.K., Venkatesh, S., Lovell, B.C. (eds.): Intl. Conf. on Pattern Recognition (ICPR). Brisbane, Australia (Aug. 1998) 310–315
2. Barnes, N., Sandini, G.: Direction control for an active docking behavior based on the rotational component of log-polar optic flow. In: Tsai, W.-H., Lee, H.-J. (eds): European Conf. on Computer Vision, vol. 2. Dublin, Ireland (Jun 2000) 167–181
3. Bernardino, A., Santos-Victor, J.: Visual behaviors for binocular tracking. Robotics and Autonomous Systems **25** (1998) 137–146
4. Bolduc, M., Levine, M.D.: A review of biologically motivated space-variant data reduction models for robotic vision. Computer Vision and Image Understanding (CVIU) **69**(2) (Feb 1998) 170–184

5. Capurro, C., Panerai, F., Sandini, G.: Dynamic vergence using log-polar images. Intl. Journal of Computer Vision **24**(1) (1997) 79–94
6. Brown, L.G.: A survey of image registration techniques. ACM Computing Surveys **24**(4) (Dec 1992) 325–376
7. Jurie, F.: A new log-polar mapping for space variant imaging. Application to face detection and tracking. Pattern Recognition **32** (1999) 865–875
8. Okajima, N., Nitta, H., Mitsuhashi, W.: Motion estimation and target tracking in the log-polar geometry. 17th Sensor Symposium. Kawasaki, Japan (May 2000) `chihara3.aist-nara.ac.jp/gakkai/sensor17`
9. Tistarelli, M., Sandini, G.: On the advantages of polar and log-polar mapping for direct estimation of time-to-impact from optical flow. IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI) **15** (1993) 401–410