Radon-like Transforms of Log-polar Images for Affine Motion Estimation

V. Javier Traver *Filiberto Pla*

Dept. Llenguatges i Sistemes Informàtics
Edifici TI, Campus Riu Sec
Universitat Jaume I
E12080-Castellón, Spain

***Abstract* – The shift property of the Radon transform becomes useful for estimating translational motion in the image domain. But, how does a more general transform in the image domain affect the projection domain? This theoretical question was addressed in [1], and we now explore these results for their potential practical application for affine motion estimation using log-polar images, rather than uniformly sampled cartesian images, because of the advantages of foveal imaging in active vision tasks.**

I. INTRODUCTION

Motion estimation is an important problem in many areas in computer vision. Indeed, a great deal of effort has been devoted in the past [2], and it is still currently being carried out (e.g., [3]) to solve several different versions of this problem. Nowadays, active vision [4, 5] is an important research field where effective and efficient motion estimation techniques are needed. Active visual tracking (see, among many others, [6, 7, 8]), for instance, which is where our work focuses on, requires a moving target to be monitored, and its motion estimated and segmented from that of the active observer. By taking inspiration from biological visual systems, it has been shown that foveal image representations offer some advantages to solve active vision problems.

One particularly suitable space-variant (foveal) paradigm is the log-polar representation [9, 10, 11]. In log-polar sampled images, the spatial resolution is maximum in the center of the visual field (the fovea), and decreases exponentially with the radius of the image point. This geometric arrangement makes possible a good trade-off between resolution and image size, while preserving a high field of view. Thus, important data reductions are achieved in this way. As an immediate consequence of this, dramatic computational savings are attained, in terms of both memory and time requirements, when compared with conventional, uniformly sampled images.

For log-polar images to be really useful in an active vision scenario, the moving target has to be kept foveated (i.e., kept in the fovea, the highly resolved part of the image). To that end, the camera has to be moved to compensate the target's motion. In turn, this requires the motion of the target to be estimated. As said above, much work exists on motion estimation, but very little of it is actually devoted to space-variant images [12, 13]. To fully integrate active vision and space-variant imaging, this "gap" needs to be filled.

Some previous work on motion estimation in log-polar images exploited the shift property of image projections to estimate translational motions [14]. Given the interesting results obtained, it seemed natural to look for some extensions of the approach to be able to deal with motions more general than a translational model. In [1], Milanfar analyzed what affine motion in the image domain becomes in the projection domain. In this work, we make use of this theoretical result to study how the affine motion parameters can, in practice, be estimated. Thus, we propose a technique based on the definition of a set of projections in the log-polar plane. The solution of two overdetermined equation systems yields the estimated motion parameters.

The organization of the paper is as follows. First, a brief overview of the log-polar transform is given. Then, the affine motion estimation algorithm for log-polar images based on projections is introduced. Next, experimental results are presented. Finally, a discussion on the proposed method, its results, and further work is given.

II. LOG-POLAR MAPPING

The log-polar mapping used here defines the log-polar coordinates $(\xi,\eta) \triangleq \left(\log_a\left(\frac{\rho}{\rho_0}\right), q\cdot\theta\right)$, with (ρ,θ) being the usual polar coordinates. Because of the discretization, the continuous coordinates (ξ,η) become the discrete ones $(u,v) = (\lfloor\xi\rfloor, \lfloor\eta\rfloor)$, $0 \le u < R$, $0 \le v < S$, with R and S being the number of rings and sectors of the discrete log-polar image, and $q = \frac{S}{2\pi}$ sectors/radian. Having chosen R, ρ_0 (the radius of the innermost ring), and $\rho_{\max}$ (the radius of the visual field), the transformation parameter a is computed as $a = \exp(\ln(\frac{\rho_{\max}}{\rho_0})/R)$. Other log-polar models and further details on their computation can be found in [9]. Given the parameters of the cartesian and log-polar geometries, the log-polar transformation builds the map $\mathcal{L}$, where $\mathcal{L}(i,j)$ is the set of log-polar pixels (u,v) intersecting the cartesian pixel (i,j). From their biological motivation, *retinal* images are those in the usual format, while *cortical* images are those resulting from the log-polar mapping (i.e., the log-polar images themselves). An example of

Research supported in part by projects GV97-TI-05-27 from the *Conselleria d'Educació, Cultura i Ciència, Generalitat Valenciana*, CICYT TIC98-0677-C02-01 from the Spanish *Ministerio de Educación y Cultura*, and grant 0I005.09/1 from *Fundació Caixa-Casstelló Bancaixa*.
Corresponding author: V. Javier Traver, E-mail: vtraver@uji.es, URL: http://www.vision.uji.es.

(a) (b) (c) (d)

Figure 1 - Log-polar mapping: (a) grid layout example (10×16), (b) original cartesian image (256×256), with grid (a) overlapped, (c) cortical image (64×128), (d) retinal image (256×256) obtained from (c) by the inverse mapping.

the log-polar mapping is shown in Fig. 1.

III. Radon-transform-based affine motion estimation

A. Notation

Image locations will be denoted by $\mathbf{p} = (x, y)$, and the *image* motion parameter vector by $\boldsymbol{\mu} = (\mu_1, \mu_2, \ldots, \mu_n)^T$. The *image* motion model can be defined by $\mathbf{f}(\mathbf{p}; \boldsymbol{\mu})$, which is a vector function indicating where location $\mathbf{p}$ in one frame moves to next frame, according to $\boldsymbol{\mu}$.

Let $I(\mathbf{p})$ be the grey-level value of image I at $\mathbf{p}$. The Radon transform $g(p; \phi) = \mathcal{R}_\phi[I]$ is the projection of I along an angle ϕ, i.e.:

$$g(p; \phi) = \iint I(x, y) \cdot \delta(p - x\cos(\phi) - y\sin(\phi))dxdy, \tag{1}$$

where p is just a scalar denoting a location in the projection domain, and $\delta(\cdot)$ is the known Dirac delta function. We also define $\mathbf{w} = (\cos(\phi), \sin(\phi))^T$ as the unit vector indicating the orientation of the projection.

The motion model in the projection domain will be denoted as $f(p; \boldsymbol{\mu})$ as a function of the *image* motion parameter vector $\boldsymbol{\mu}$, or as $f(p; \boldsymbol{\mu}_p)$ as a function of the *projection* motion parameter vector $\boldsymbol{\mu}_p$.

As can be noticed, boldface is used to denote vector quantities. As an example, $\mathbf{p}$, $\boldsymbol{\mu}$, or $\mathbf{f}(\cdot;\cdot)$ are boldfaced, while p, or $f(\cdot;\cdot)$ are *not*, because they are scalar variables or functions.

B. Model of affine motion in the projection domain

Affine motion models are widely used in the literature because it is a reasonably simple model (only 6 parameters) and still powerful enough to explain many 2D image projections of 3D world motion. The affine *image* motion model is defined as:

$$\mathbf{f}(\mathbf{p}; \boldsymbol{\mu}) = \mathbf{p}_0 + \mathbf{M} \cdot \mathbf{p}, \tag{2}$$

with $\mathbf{p}_0 = (u, v)$ the zero-order flow (constant vector denoting translational motion), and

$$\mathbf{M} = \begin{bmatrix} b & c \\ d & f \end{bmatrix}$$

the so-called *affine matrix*. Thus, our motion parameter vector is $\boldsymbol{\mu} = (b, c, d, f, u, v)^T$.

By using some properties of the Radon transform (the transform of derivatives, the derivatives of the transform, the linearity property, and the differential projected motion identity), as well as some approximations (representation of images using Hermite polynomials, and for sufficiently large p), the corresponding *projection* motion model can be written as [1]:

$$f(p; \boldsymbol{\mu}) \approx \mathbf{p}_0^T \mathbf{w} + (\mathbf{w}^T \mathbf{M} \mathbf{w}) \cdot p, \tag{3a}$$

which can be rewritten as:

$$\begin{aligned} f(p; \boldsymbol{\mu}) \approx\ & u\cos(\phi) + v\sin(\phi) \\ & + \left(\frac{b+f}{2} + \frac{b-f}{2}\cos(2\phi) + \frac{c+d}{2}\sin(2\phi) \right) \cdot p, \end{aligned} \tag{3b}$$

with ϕ the angle denoting the direction of the projection.

C. Our approach

Note that Eq. (3b) can yet be written as:

$$\begin{aligned} f(p; \boldsymbol{\mu}) \approx\ & \alpha_u \cdot u + \alpha_v \cdot v \\ & + (\beta_b \cdot b + \beta_c \cdot c + \beta_d \cdot d + \beta_f \cdot f) \cdot p, \end{aligned} \tag{3c}$$

with

$$\begin{array}{lllllll} \alpha_u & = & \cos(\phi), & & \alpha_v & = & \sin(\phi), \\ \beta_b & = & \frac{1}{2}\left(1 + \cos(2\phi)\right), & & \beta_c & = & \frac{1}{2}\sin(2\phi), \\ \beta_f & = & \frac{1}{2}\left(1 - \cos(2\phi)\right., & & \beta_d & = & \frac{1}{2}\sin(2\phi). \end{array} \tag{3d}$$

Further, it can be easily noticed that the form of Eq. (3c) is:

$$f(p; \boldsymbol{\mu}_p) = \alpha + \beta \cdot p, \tag{4}$$

with $\boldsymbol{\mu}_p = (\alpha, \beta)^T$ being the parameters of an affine transform in the 1D projection domain.

For a reason that will be clear below, we introduce the "dummy" parameter $h = c + d$, and then, define $\beta_h = \beta_c = \beta_d$.

For each particular projection k, with orientation ϕ_k, and this time using the coupled parameter h, Eqs. (3c) and (4) can be re-expressed, respectively, as:

$$\begin{aligned} f_k(p; \boldsymbol{\mu}) \approx\ & \alpha_u^{(k)} \cdot u + \alpha_v^{(k)} \cdot v \\ & + \left(\beta_b^{(k)} \cdot b + \beta_h^{(k)} \cdot h + \beta_f^{(k)} \cdot f \right) p, \end{aligned} \tag{5a}$$

and

$$f_k(p; \boldsymbol{\mu}_p^{(k)}) = \alpha_k + \beta_k \cdot p. \tag{5b}$$

By equating Eqs. (5a) and (5b), we have:

$$\alpha_k = \alpha_u^{(k)} \cdot u + \alpha_v^{(k)} \cdot v, \tag{6a}$$

$$\beta_k = \beta_b^{(k)} \cdot b + \beta_h^{(k)} \cdot h + \beta_f^{(k)} \cdot f. \tag{6b}$$

Eqs. (6a) and (6b), considered for N projections, can be written in matrix form as:

$$\mathbf{\Gamma}_\alpha \cdot \boldsymbol{\mu}_\mathbf{p} = \boldsymbol{\psi}_\alpha, \tag{7a}$$

$$\mathbf{\Gamma}_\beta \cdot \boldsymbol{\mu}_\mathbf{M} = \boldsymbol{\psi}_\beta, \tag{7b}$$

by introducing the *measurement* vectors

$$\boldsymbol{\psi}_\alpha = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_N \end{bmatrix}, \; \boldsymbol{\psi}_\beta = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_N \end{bmatrix}, \tag{8a}$$

the *unknowns* vectors

$$\boldsymbol{\mu}_\mathbf{p} = \begin{bmatrix} u \\ v \end{bmatrix}, \; \boldsymbol{\mu}_\mathbf{M} = \begin{bmatrix} b \\ h \\ f \end{bmatrix}, \tag{8b}$$

and the *coefficient* matrices:

$$\mathbf{\Gamma}_\alpha = \begin{bmatrix} \alpha_u^{(1)} & \alpha_v^{(1)} \\ \alpha_u^{(2)} & \alpha_v^{(2)} \\ \vdots & \vdots \\ \alpha_u^{(N)} & \alpha_v^{(N)} \end{bmatrix}, \mathbf{\Gamma}_\beta = \begin{bmatrix} \beta_b^{(1)} & \beta_h^{(1)} & \beta_f^{(1)} \\ \beta_b^{(2)} & \beta_h^{(2)} & \beta_f^{(2)} \\ \vdots & \vdots & \vdots \\ \beta_b^{(N)} & \beta_h^{(N)} & \beta_f^{(N)} \end{bmatrix}. \tag{8c}$$

The solution of the system (7a) would provide an estimate of the translation parameters $\boldsymbol{\mu}_\mathbf{p} = (u, v)^T$. Likewise, the matrix equation (7b) would yield the estimated affine parameters $\boldsymbol{\mu}_\mathbf{M} = (b, h, f)^T$. We will deal with overdetermined systems by using $N > 3$ projections.

We have considered the parameters c and d coupled into the parameter h because, otherwise, the corresponding matrix $\mathbf{\Gamma}_\beta$ would be singular, since $\beta_c^{(k)} = \beta_d^{(k)}, \; \forall\, k$.

D. The projections in log-polar domain

Due to the geometric nature of the log-polar grid, the computation of image projections along a given direction is not immediate. By using the log-polar map $\mathcal{L}(i, j)$, we define and compute the projections support maps $\mathcal{P}_\phi$ for every needed projection orientation ϕ. The elements taking part in the definition of one of such maps are illustrated in Fig. 2. Hence, for a given orientation ϕ, $\mathcal{P}_\phi(p)$ is the set of log-polar pixels (u, v) lying on the straight line $\mathbf{r}_{\phi,p}$, whose equation in normal form can be given as:

$$\mathbf{r}_{\phi,p} \equiv p - x\cos(\phi) - y\sin(\phi) = 0.$$

More formally, for $-w_K \leq p \leq w_K$,

$$\mathcal{P}_\phi(p) = \left\{ (u, v) : (u, v) \in \mathcal{L}(x, y), (x, y) \in \mathbf{r}_{\phi,p}^{w_L} \right\}, \tag{9}$$

where $\mathbf{r}_{\phi,p}^{\ell}$ denotes the line segment formed by points on $\mathbf{r}_{\phi,p}$ whose distance d to the origin is $d \leq \ell$. In Fig. 3 some examples of the projections support maps are shown, where each p is coded as a grey-level value.

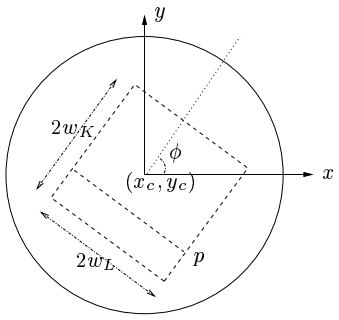

Figure 2 - Elements to define projections directions at orientation ϕ. A total of $2w_K$ lines are considered centered around the center of the log-polar mapping (x_c, y_c). One of these lines, p pixels away from the center, is drawn. The extension of each of these lines is $2w_L$.

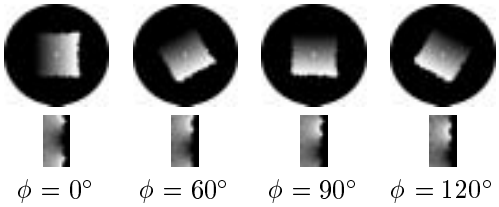

Figure 3 - Some examples of projections support maps in retinal plane (first row) and cortical plane (second row), at some projection orientations ϕ.

Having defined the support map $\mathcal{P}_{\phi_k}$ for the projections at orientation ϕ_k, the projection signal itself can be computed, by using a discrete version of the general equation (1), as:

$$g(p; \phi_k) = \sum_{(u,v) \in \mathcal{P}_{\phi_k}(p)} I(u, v), \tag{10}$$

i.e., the sum of the gray-level values of log-polar pixels along the straight line $\mathbf{r}_{\phi_k,p}$. Looking at Fig. 3, pixels with the same gray level would contribute to the same $g(p; \phi_k)$.

Fig. 4 shows an example of the (smoothed) 1D signals resulting from the projections at orientation $\phi = \pi/2$ rad. of two log-polar images, before and after an image scaling. It can be appreciated how the signals are similar at the center, and differ towards their ends, which is the expected theoretical effect of a scaling on the projection domain.

E. The algorithm

The details of the whole process are given in the algorithm in Fig. 5. For the same set Φ of orientations, the projection

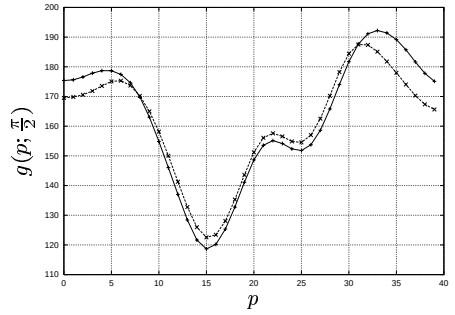

Figure 4 - Example of resulting projections $g(p; \frac{\pi}{2})$ signals before and after a uniform scaling in the image domain.

Input: Two log-polar images I_1 and I_2, and a set of N orientations $\Phi = \{\phi_1, \phi_2, \dots \phi_N\}$
Output: The estimated (reduced) affine motion vector $(b, h, f, u, v)^T$

{Off-line stage (to be computed only once)}
for $k \in \{1, 2, \dots, N\}$ **do**
  Compute the projections support maps $\mathcal{P}_{\phi_k}$ (Eq. (9))
  Compute coefficients $\alpha_u^{(k)}, \alpha_v^{(k)}, \beta_b^{(k)}, \beta_h^{(k)}$, and $\beta_d^{(k)}$ (Eq. (3d))
  Build matrices $\mathbf{\Gamma}_\alpha, \mathbf{\Gamma}_\beta$ (Eqs. in (8c))
end for
{On-line stage (to be computed once for each image pair)}
for $k \in \{1, 2, \dots, N\}$ **do**
  Compute the projections $g_i(p; \phi_k)$ of $I_i,\ i \in \{1, 2\}$ (Eq. (10))
  Estimate the 1D affine motion parameters $\boldsymbol{\mu}_p^{(k)}$ between $g_1(p; \phi_k)$ and $g_2(p; \phi_k)$
end for
Build vectors $\boldsymbol{\psi}_\alpha, \boldsymbol{\psi}_\beta$ (Eq. (8a))
Solve overdetermined systems (Eqs. (7a) and (7b)) for $\boldsymbol{\mu}_{\mathbf{M}}$ and $\boldsymbol{\mu}_{\mathbf{P}}$
$(b, h, f)^T \leftarrow \boldsymbol{\mu}_{\mathbf{M}}$
$(u, v)^T \leftarrow \boldsymbol{\mu}_{\mathbf{P}}$

Figure 5 - Algorithm for affine motion estimation using Radon-like transforms in log-polar images.

support maps $\mathcal{P}_{\phi_k}$, $\phi_k \in \Phi$, as well as the coefficients in matrices in Eqs. (8c) need only to be computed once and for all, in an off-line stage, which saves on-line time. For a given image pair, the on-line process consists of computing the projections for each orientation in Φ. Then, using the projection signals, the two parameters α_k, β_k of the projection motion vector $\boldsymbol{\mu}_p^{(k)}$ are estimated. This estimation can be carried out with any possible technique. Here, we simply used an exhaustive correlation-based search, by using a set of possible values for (α_k, β_k), in a range of expected variation of motions. Although this approach may not be the most efficient one, our aim here was mainly to study the feasibility of using projections for affine motion estimation. Finally, the overdetermined systems (Eqs. (7a) and (7b)) are solved, by using the QR decomposition provided in the MatClass library [15].

IV. Experimental results

Set-up and settings. To simulate motion, we apply a known affine transformation $\boldsymbol{\mu}^*$ to an original 256×256 cartesian image. The log-polar transformations of these both images (i.e., before and after the transformation) are used as input to the approach described in Sect. III and the estimated motion parameter vector $\boldsymbol{\mu}$ is computed. We used 32×64 log-polar images, with $\rho_0 = 5$ and $\rho_{\max} = 128$. The set of projections used was $\Phi = \{0°, 45°, 60°, 90°, 120°, 165°\}$, and $w_K = w_L = 50$. Although the algorithm has been tested on several images, the experiments reported in this paper correspond to the test image shown in Fig. 6.

Test 1 (translation+scaling). In a first experiment, up to 4-pixel large translations were combined with scalings factors of up to 20% big. The true parameters are made to vary so that the deformation rates (translation magnitude and scaling factor) grow with the experiment number. The estimation results are shown in Fig. 7, one graph per parameter. In each graph, the ground-truth parameter and its estimation are plotted. The goodness of the results can be appreciated by observing how close to each other the curves

(a) cartesian images (b) retinal images

Figure 6 - Test image before and after a shear transformation.

corresponding to true and estimated parameters are (e.g., see the plots of the horizontal translation u). Even though reasonable good estimates are obtained, it is also clear that, in general, accuracy is not always very high (e.g., in the plots of b, in the experiments 0–15, when b is very small, the algorithm is not so sensitive so as to estimate those small image variations). Notice that we are dealing with anisotropic scalings, as $b^* \neq f^*$. This would not be possible with a simpler motion model such as a similarity motion model, which consider translation, rotation and (isotropic) scaling. Obviously, as the deformation in the image increases, the accuracy in the results is affected (e.g., note how the estimation of the coupled paramter h is worse in the last experiments, when both the scaling and the translation are big). It might be possible to refine estimates by iterative improvement.

Test 2 (translation+shear). The same amount of translation as in Test 1 was combined with a shear of up to 0.2. Shear is another motion component which can be represented in an affine motion model, but not in simpler motion models. Estimation results are plotted in Fig. 8 (to save space, only plots of three out of the five parameters are shown). Upper row corresponds to the estimation results using the same set of projections as in Test 1, while lower row shows the results using a reduced set $\Phi' = \{0°, 45°, 90°\}$ of only three projections, rather than six. Results are very similar to those in the first test: reasonably good, although not very fine, estimates of the motion parameters are obtained for up to large translations and shear rates, and errors increase with larger image deformations. Shear is one of the motion components which may

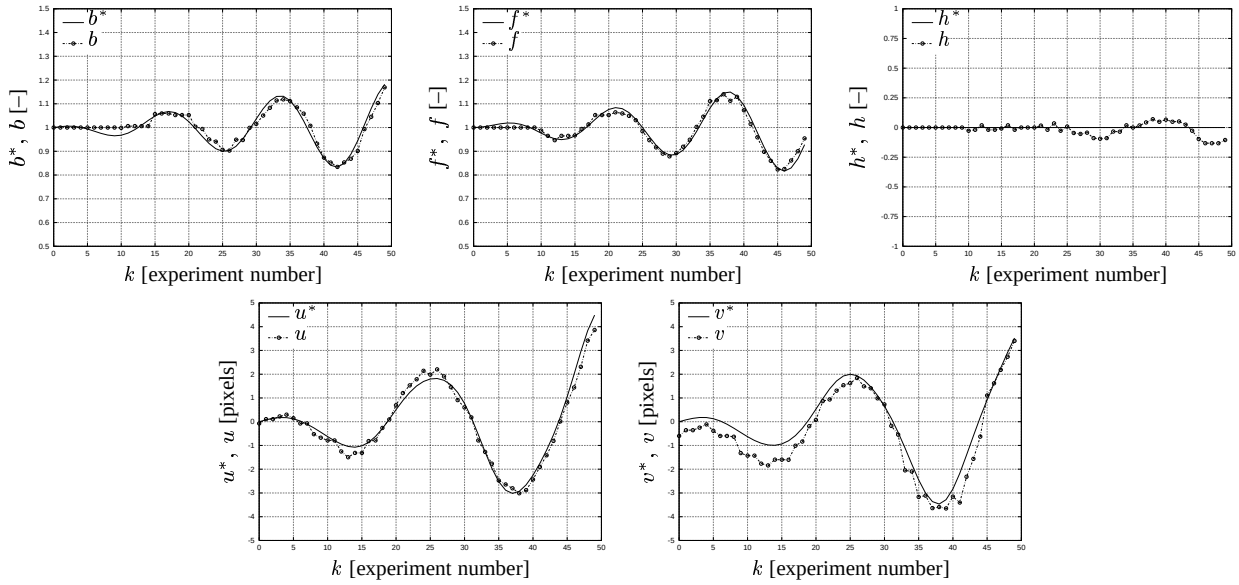

Figure 7 - True and estimated motion parameters (motion included a combination of translation and scaling).

be coded in the parameters c and d. Therefore, even though good estimates for h are also obtained, little knowledge can be gained from h alone. This problem seems not to have an easy solution. The use of some *a priori* information could provide some extra information about the actual motion being undergone.

Finally, it is worth comparing the results using the two sets of projections Φ and Φ'. As can be appreciated, using more projections leads, in general, to better results, because the method becomes more robust against errors in the estimation of the 1D affine motion vectors $\boldsymbol{\mu}_p$ (compare both estimates of h, where this improvement is more evident). Obviously, many more projections do not always imply much better accuracy, while it is always true that the more projections are used, the more costly the process. We found that a set of $N = 6$ projections was a good trade-off.

Test 3 (rotation). Rotations of increasing angles up to 20° were applied, and the estimates are shown in Fig. 9 (again, plots of two of the parameters are not shown). As can be seen, in this case results are not so encouraging. This is due to the very nature of the approximate model of the effect of image motion on the projection domain [1]: no term $c - d$ appears in Eq. (3b). Unfortunately, this makes the approach inappropriate if curl is one of the expected motion components.

V. Conclusions

An algorithm to estimate affine motion in the image domain by using an approximate model of the effect of such a motion in the projected domain has been presented. The approach is quite effective since it can successfully deal with very large deformations, a feature which is not always possible in many motion estimation algorithms, like many optic-flow-based ones. Our method is also reasonably fast, due to two main reasons: (1) the use of very small log-polar images, and (2) the fact of working in the projection domain, rather than in one high dimensional search space.

The proposed algorithm does not yield very fine estimations of the motion parameters, but quite good rough estimates are possible. This makes the method suitable for applications where fast motions are expected. Also, it could be useful as a first step for other motion estimation algorithms to refine the solution. In general, the more projections are used, the better results are obtained. Interestingly, the number of projections could theoretically be increased according to the availability of computational resources. In practice, however, there is a limit in the accuracy that can be achieved. A reasonable cardinality for a practical set of projections would be 5–10. On the other hand, if one is only interested in translation estimation, only system (7a) has to be solved, not (7b).

As we are using an affine motion model, deformations like anisotropic scalings and shear can be dealt with, which is not the case with simpler motion models such as a similarity model. However, not all the six affine motion parameters could be fully estimated, because two of them can only be estimated as a single coupled parameter. Unfortunately, this limits the practical usefulness of the approach. Finding an appropriate way to decouple this parameter would definitely improve the method.

Summing up, the motion estimation algorithm introduced in this paper relies on a sound theoretical approach, and exhibits a good behavior in some useful practical problems. Nevertheless, it also has some limitations: only a reduced set of affine motions can be considered, and some *a priori* knowledge might be needed to disambiguate the coupled parameter for a correct interpretation of the actual underlying motion.

As for future work, finding similar approximations of the effect of the image motion in the cartesian and, mainly, directly in the log-polar domains on a 1D projected domain remains a challenging, interesting research topic.

References

[1] Peyman Milanfar, "A model of the effect of image motion in the Radon transform domain", *IEEE Trans. on Image Processing*, vol. 8,

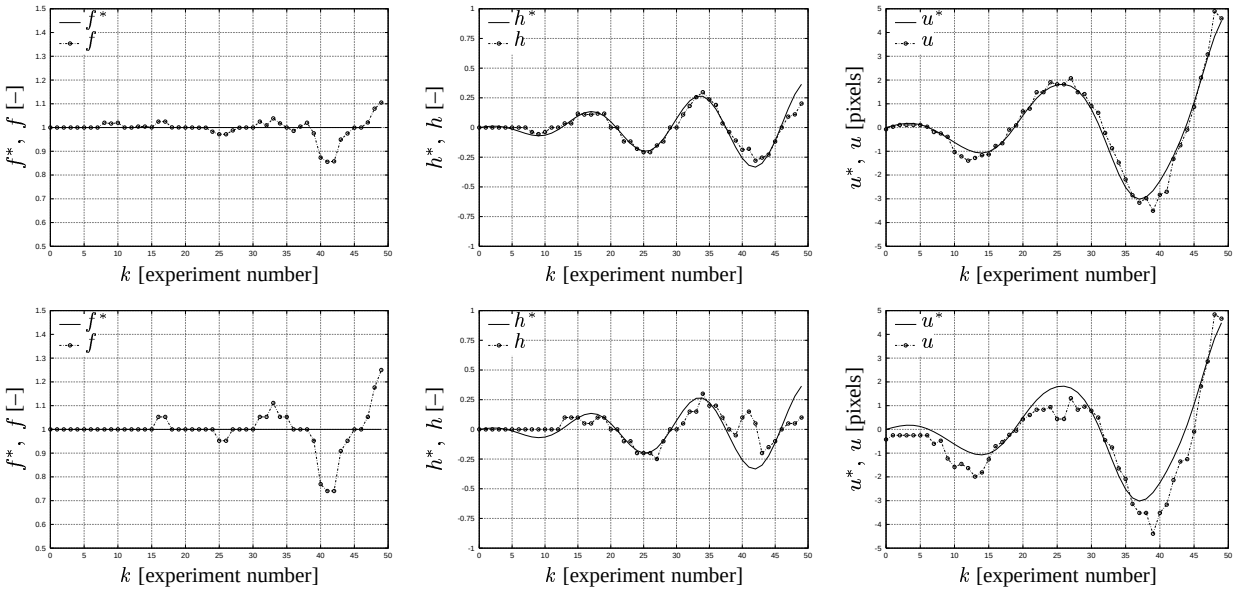

Figure 8 - True and estimated motion parameters (motion included a combination of translation and shear). Upper and lower rows correspond to results using two different sets of projections (see text for details).

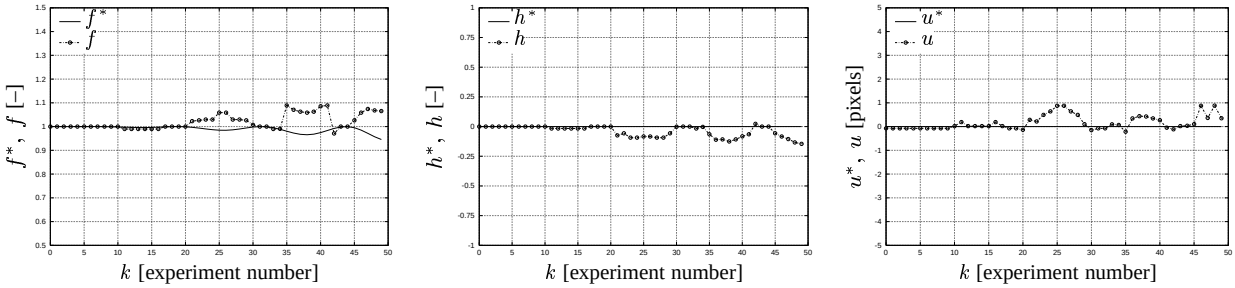

Figure 9 - True and estimated motion parameters (motion was only rotational).

no. 9, pp. 1276–1281, Sept. 1999.

[2] Christopher Stiller and Janusz Konraad, "Estimating motion in image sequences: A tutorial on modeling and computation of 2D motion", *IEEE Signal Processing Magazine*, pp. 70–91, July 1999.

[3] R. Montoliu and F. Pla, "Multiple parametric motion model estimation and segmentation", in *IEEE Intl. Conf. on Image Processing (ICIP)*, Thessaloniki, Greece, Oct. 2001, vol. II, pp. 933–936.

[4] "Promising directions in active vision", Technical Report CS 91-27, University of Chicago, Nov. 1991, Written by the attendees of the NSF Active Vision Workshop.

[5] Joel Hynoski and H. R. Wu, "Active vision — a survey of the field and research directions", Tech. Rep. 95-04, Faculty of Computing and Information Technology. Dept. of Robotics and Digital Technology. Monash University, May 1995.

[6] Don Murray and Anup Basu, "Motion tracking with an active camera", *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 16, no. 5, pp. 449–459, 1994.

[7] Han Wang, Choon Seng Chua, and Ching Tong Sim, "Real-time object tracking from corners", *Robotica*, vol. 16, pp. 109–116, 1998.

[8] R. Herpers, K. Derpanis, W. J. MacLean, G. Verghese, M. Jenkin, E. Milios, A. Jepson, and J. K. Tsotsos, "SAVI: An actively controlled teleconferencing system", *Image and Vision Computing*, vol. 19, pp. 793–804, 2001.

[9] Marc Bolduc and Martin D. Levine, "A review of biologically motivated space-variant data reduction models for robotic vision", *Computer Vision and Image Understanding (CVIU)*, vol. 69, no. 2, pp. 170–184, Feb. 1998.

[10] F. Pardo, J. A. Boluda, J. J. Pérez, B. Dierickx, and D. Scheffer, "Design issues on CMOS space-variant image sensors", in *SPIE Conf. on Advanced Focal Plane Arrays and Electronic Cameras (AFPAEC)*, Berlin, Germany, Oct. 1996.

[11] Richard S. Wallace, Ping-Wen Ong, Benjamin B. Bederson, and Eric L. Schwartz, "Space-variant image processing", Tech. Rep. 589-R256, Vision Applications, Inc., Nov. 1991.

[12] Hilary Tunley and David Young, "Dynamic fixation of a moving surface using log polar sampling", in *British Machine Vision Conference*, Univ. York, York, Sept. 1994, vol. 2, pp. 579–588, BMVA Press.

[13] Alexandre Bernardino, José Santos-Victor, and Giulio Sandini, "Tracking planar structures with log-polar images", in *Symp. on Intelligent Robotic Systems*, Reading, UK, July 2000, (Also as VisLab TR 06/2000).

[14] V. J. Traver and F. Pla, "Estimation of translation, rotation, and scaling in log-polar images using projections", in *Conf. de la Asoc. Española para la Inteligencia Artificial (CAEPIA)*, Gijón, Spain, Nov. 2001, vol. I, pp. 645–654.

[15] C. R. Birchenhall, "A draft guide to MatClass: A matrix class for C++", 1993, Dept. of Econometrics & Social Statistics, University of Manchester.