Estimation of Translation, Rotation and Scaling in Log-polar images Using Projections*

V. Javier Traver and Filiberto Pla
{vtraver|pla}@uji.es
Computer Vision Group · Universitat Jaume I
Campus Riu Sec · 12080 – Castellón, Spain

Abstract: *Efficient motion estimation algorithms are needed to achieve real-time performance in active vision applications. The work presented in this paper makes use of log-polar images, which allows an important selective data reduction. On the other hand, projections along different directions in the image plane are formally defined. By computing these projections on two consecutive frames of an image sequence, the estimation of translations, rotations, and scalings become the same simple problem of estimating the shift between two one-dimensional signals. This is possible by exploiting the known rotation and scale invariance properties of the log-polar mapping. A two-stage algorithm is proposed to deal with a combination of these basic motion components. Experimental results show the effectiveness of this approach, which can accurately estimate all four motion components, that can coexist at the same time, provided that only moderate translations are present.*

Keywords: Artificial vision, log-polar mapping, projections, motion estimation.

1 Introduction

Within the paradigm of active vision [4], the observer is allowed to dynamically interact with its environment. This possibility carries a number of advantages over the conventional passive vision. Closely related to the active vision is the concept of space-variant imaging which, unlike uniformly resolved images, proposes images with different resolutions across their spatial extent. Log-polar vision [2] is the most common approach for foveal vision: it offers a resolution which is maximum at the image center and decreases with the eccentricity. Used jointly, active and space-variant vision offer a powerful and efficient mechanism for active tracking of mobile targets [1, 3] which is a topic of interest in

*Research supported in part 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*.

useful real-world applications such as visual surveillance. One of the advantages brought by log-polar images is the drastic reduction of information (compare the sizes of images in figures 1(b) and 1(c)), which is of crucial importance to make active vision tasks exhibit real-time performance. In the context of tracking, motion estimation is needed to be able to move the camera to compensate for the relative motion between the moving object and the camera. Thus, for real-time tracking, efficient motion estimation algorithms are called for. Although a vast amount of literature exists on motion estimation in cartesian images, there are fewer works addressing motion estimation in the log-polar space.

Moments-based motion estimation algorithms [8] seem interesting because they are relatively simple and may lend themselves to real-time implementations. However, they usually assume that the projection onto the image plane of the moving object: (i) is perfectly segmented, and (ii) is completely contained in two consecutive images. Unfortunately, these assumptions are not very realistic and would be restrictive in our case. Projections, either in the spatial or the frequency domain, are also an attractive approach [6]. Our work is somehow inspired by these two kind of works. We use different sets of projections in the spatial domain in such a way that the disadvantages of moments are overcome.

While translation estimation is of key importance for active tracking [7], motion in the image plane can rarely be explained only with translation. Therefore, besides translation, we also consider rotation and scaling around the image center. Considering centered rotations and scalings makes sense in our problem because, in active tracking, the object is expected to be kept foveated most of the time. In this paper, we define four types of projections and show that they are appropriate for detecting four different motion components in log-polar images. We also explore the potential of these projections for estimating motion combining translation, rotation and scaling.

2 Log-polar mapping

A log-polar mapping commonly used in literature (e.g., [5]) defines the log-polar coordinates $(\xi,\eta) \triangleq \left(\log_a\left(\frac{\rho}{\rho_0}\right),\theta\right)$, with (ρ,θ) being the polar coordinates defined from the cartesian coordinates (x,y) as usual, i.e., $(\rho,\theta) \triangleq \left(\sqrt{x^2+y^2},\arctan\frac{y}{x}\right)$. Because of the discretization, the continuous coordinates (ξ,η) become the discrete ones $(u,v) = (\lfloor\xi\rfloor,\lfloor\theta\cdot q\rfloor)$, $0 \le u < R$, $0 \le v < S$, 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 [2].

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). An example of a log-polar transformation is shown in figure 1.

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

Figure 1. Log-polar mapping: (a) grid layout example (10×16), (b) original cartesian image 256×256, (c) cortical image 32×64, (d) retinic image 256×256 obtained by the inverse mapping.

3 Projections for motion estimation

3.1 Defining projections

We define four different projection functions, each along a different direction. An *angular* projection $\mathcal{A}$ over the log-polar image I is defined for each discrete angular log-polar coordinate v as

$$\mathcal{A}(v) \triangleq \sum_{u=0}^{R-1} I(u,v), \qquad v \in [0..S-1]. \tag{1}$$

Similarly, a *radial* projection $\mathcal{R}$ is defined for each discrete radial log-polar coordinate u as

$$\mathcal{R}(u) \triangleq \sum_{v=0}^{S-1} I(u,v), \qquad u \in [0..R-1]. \tag{2}$$

A *vertical* projection $\mathcal{V}$ at the discrete vertical coordinate i is defined as

$$\mathcal{V}(i) \triangleq \sum_{j=j_3}^{j_4} \sum_{(u,v)\in\mathcal{L}(i,j)} I(u,v), \qquad i \in [i_1..i_2]. \tag{3}$$

Likewise, a *horizontal* projection $\mathcal{H}$ at the discrete horizontal coordinate j is defined as

$$\mathcal{H}(j) \triangleq \sum_{i=i_3}^{i_4} \sum_{(u,v)\in\mathcal{L}(i,j)} I(u,v), \qquad j \in [j_1..j_2]. \tag{4}$$

Figure 2 illustrates the directions along which these projections are computed. Note how radial and angular directions in the cartesian domain are columns and rows in the log-polar image, and how vertical and horizontal directions are mapped to curves.

3.2 Using projections for motion estimation

Now we show how projections can be used to estimate the motion undergone by two consecutive log-polar images, I_1 and I_2. We add a subindex k to the projection functions to indicate that they are defined on image I_k, $k = 1, 2$.

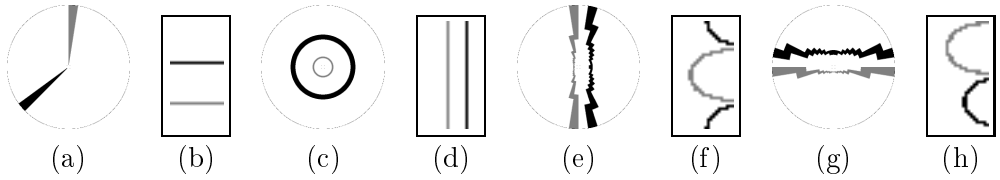


Figure 2. Examples of projection directions: (a,b) angular projections; (c,d) radial projections; (e,f) vertical projections; (g,h) horizontal projections, in retinic (a,c,e,g) and cortical (b,d,f,h) domains. For illustration purposes, the relative size of the cortical and retinic images is not the real one.

If a rotation ϕ around the image center has occurred between I_1 and I_2, the angular projections $\mathcal{A}_1$ and $\mathcal{A}_2$ are related as $\mathcal{A}_1(v) = \mathcal{A}_2(v - d_a)$, with $\phi = d_a/q$. Therefore, the angle of rotation can readily be computed from the translation d_a between the 1D signals $\mathcal{A}_1$ and $\mathcal{A}_2$. In turn, d_a can be estimated by some correlation-based measure.

After a change of scale about the image center by a factor α, the following relationship holds: $\mathcal{R}_1(u) = \mathcal{R}_2(u - d_r)$, with $\alpha = a^{d_r}$. Estimation of d_r can be performed exactly in the same way as d_a above.

One of the interesting features of the log-polar vision is the fact that rotations and scalings around the image center become mere translations in the log-polar domain. Note that by exploiting these properties, rotations and scalings are estimated very naturally here.

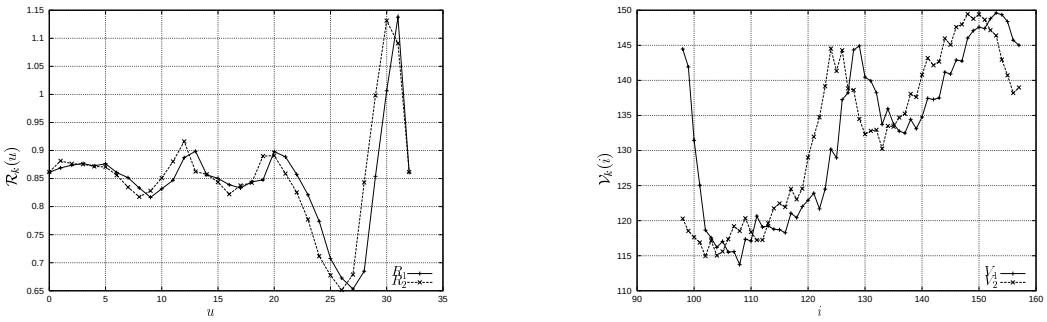
Translations in x and y directions (d_x and d_y) can be estimated by making use of the vertical and horizontal projections, respectively. Although these projections are not defined as straightforwardly in log-polar images as rotations and scalings are, the principle behind them is the same: projections are defined in the image plane in a direction orthogonal to the direction of the motion to be detected. Thus, d_x and d_y can be estimated by observing that $\mathcal{V}_1(i) = \mathcal{V}_2(i - d_x)$ and $\mathcal{H}_1(j) = \mathcal{H}_2(j - d_y)$.

The beauty of this approach lies in its simplicity and in the fact that different motion components are estimated by using the same approach in every case: estimating a translation between two one-dimensional signals.

Figure 3 depicts two examples of the typical 1D signals we are dealing with, as well as their translations. In practice, to deal with smaller numbers, all projections are multiplied by a scale factor β. Additionally, vertical (horizontal) projections are normalized, because projections at different horizontal (vertical) locations i (j) are computed over sets of pixels of different cardinality.

3.3 Projections-based algorithm

In general, a rotation (or scaling) will affect the vertical and horizontal projections, and translations will affect the angular (and radial) projections. Therefore, the independent component motion estimation may become unreliable. To tackle this problem, we propose a 2-stage algorithm: in a first stage, scalings and rotations are estimated (via projections).



(a) Radial projections (b) Vertical projections

Figure 3. Typical translations between two 1D projection functions: (a) radial projections before and after a scaling ($\alpha = 1.1$); (b) vertical projections before and after an horizontal displacement ($d_x = 4$)

These estimates are used to "undo" their effects on the image. Then, in the second stage, vertical and horizontal displacements are estimated. If the estimates of the first stage are good enough, it is expected that horizontal and vertical estimates at the second stage will also be quite accurate.

This approach was inspired by experimental results which revealed that rotations and scalings estimates were much unaffected by small translations. Additionally, "undoing" rotations and scalings can be done more effectively (and naturally) in log-polar domain than translations can.

4 Experimental results

As explained, each projection function is designed for a specific motion component. Therefore, a particular projection is affected by motion components other than the one the projection is suited to. The experiments we present here are aimed at exploring to what extent different motions can coexist and still be (accurately) estimated, independently one from each other. To perform the experiments, $N \times N = 256 \times 256$-sized cartesian image are transformed with known ground-truth motion parameters. Then, their 32×64 log-polar images, with $\rho_0 = 5$ and $\rho_{\max} = N/2 = 128$, are used as input to the motion estimation algorithm, which yields estimates of these motion parameters. This allows us to compare the true and estimated parameters, thus studying the estimation error. For a parameter p, its estimation is denoted with $\hat{p}$. For the vertical and horizontal projections, we used $i_1 = j_1 = i_3 = j_3 = N/2 - w$ and $i_2 = j_2 = i_4 = j_4 = N/2 + w$, with $w = 30$, to cover the image foveal region. For the 1D correlation, the correlation coefficient is used.

Vertical and horizontal translations. First of all, we test translations both in x and y directions simultaneously. A set of displacements $d_x, d_y \in \mathcal{S}_{d_x} = \mathcal{S}_{d_y}$ (see table I) was used to generate pairs of log-polar images. This set of values is chosen to test some small displacements and a few of larger ones. Figure 4 shows the errors estimating each

Table I. Sets of values of motion parameters used in the experiments

Set Name	Set Contents
$\mathcal{S}_{d_x}, \mathcal{S}_{d_y}$	$\{-17, -12, -8, -5, -3, -2, -1, 0, 1, 2, 3, 5, 8, 12, 17\}$
$\mathcal{S}_\phi$	$\{-90, -60, -45, -30, -15, -10, -5, 0, 5, 10, 15, 30, 45, 60, 90\}$
$\mathcal{S}_\alpha$	$\{0.3, 0.6, 0.7, 0.8, 0.9, 0.95, 1, 1.1, 1.15, 1.3, 1.5, 1.8, 2, 2.5, 3\}$

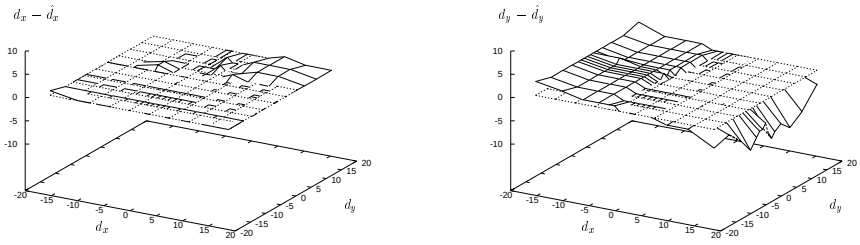
of the $|\mathcal{S}_{d_x}| \cdot |\mathcal{S}_{d_y}| = 225$ motions, for two different images. For reference purposes, the zero-error plane is plotted. In one image, very small errors can be observed (figure 4(a), on the left) for all the values of d_x. In this same image, estimation of the d_y component is accurate enough for small displacements, but grow as $|d_x|$ does. The maximum error that occurred in $|d_y - \hat{d_y}|$ (figure 4(a), on the right) was of 9 pixels. This is a typical result of many motion estimation algorithms, which make good estimations of small motions and worse estimations of larger motions. Note that error does not increase with d_y, but only with d_x. This is probably due the the particular image contents, which affect the projection signals. In a second image, results (figure 4(b)) are very interesting: $|d_x - \hat{d_x}| = 0$ for every d_x tested, and $|d_y - \hat{d_y}|$ is 0 in most cases, and 1 only in 10 out of the 225 tests. Additionally, all of these 10 cases occurred just in large displacements ($d_y = -17$). Although not clear from the plots, errors are always in integer units because of the way we estimate the translations between the projection functions.

Rotations and scalings. Next, we test rotations and scaling simultaneously. The experiments are carried out with the set of rotations $\phi \in \mathcal{S}_\phi$ (in degrees), and the set of scale factors $\alpha \in \mathcal{S}_\alpha$, as presented in table I. Figure 5 plots the results for two images. In the first case, the average error was $0.15°$, with a standard deviation of $27°$. In figure 5(a), it can be seen that only in a few out of 225 cases, the error $|\phi - \hat{\phi}|$ is big ($180°$). As for the estimation of the scaling factor α, the error is very small, with a maximum estimation error of 6%. It is important to note that both rotations and scalings tested included high values, and even in these cases the estimation is very accurate in most cases. In the second image, the results are better (figure 5), with a maximum $|\phi - \hat{\phi}| \approx 28°$, an average of $0.75°$, and a standard deviation of $3.3°$. Similar results to first image are obtained for the estimation of α.

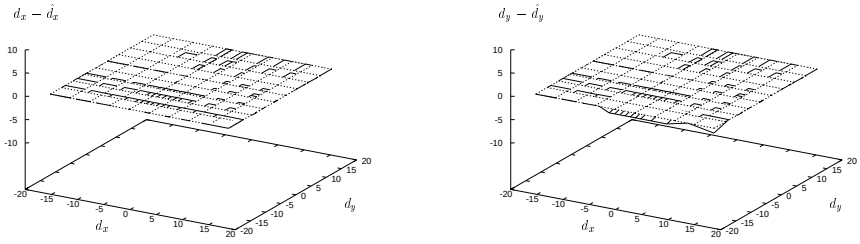
Mixing translations, rotations and scalings. In the experiments above, we have assumed that only translations were present (when estimating translations), or only scalings and rotations were present (when estimating rotations and scalings). Now, we address the question of estimating the four parameters (d_x, d_y, ϕ, and α) when a combination of the four motion components is present. Thus, the 2-stage algorithm is needed now.

In figure 6 we include the estimation errors of the four motion parameters corresponding to varying d_x and ϕ, using the same set of values for these parameters used in previous experiments[1]. The results show that when both d_x and ϕ are large, errors are generally

[1]The combination of values for d_x and ϕ are shown on the horizontal axis. The order in which these tests are carried out (ϕ's in an inner loop for every d_x in the outer loop), allows us to associate values



(a) Image Boat



(b) Image Corridor

Figure 4. Errors in translation estimates. Results for two different images are presented. Errors $d_x - \hat{d}_x$ are plotted on the left, and errors $d_y - \hat{d}_y$ are plotted on the right.

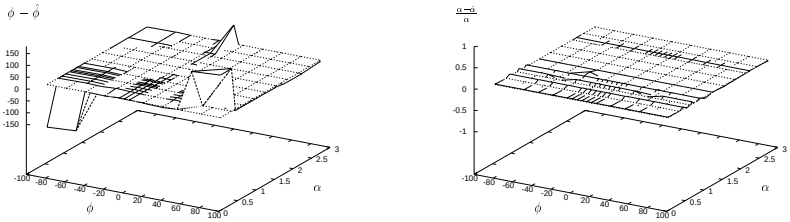
high. On the other hand, for small d_x, errors are much smaller, even for large values of ϕ. Again, errors are "discretized" because of the lack of sub-pixel accuracy in our estimation of the shift between two projection signals.

Finally, to test the average behavior of the algorithm in active vision scenarios, we perform $Q = 500$ experiments by randomly varying the true motion parameters uniformly in $\{d_x\} \times \{d_y\} \times \{\phi\} \times \{\alpha\} \in [-3, 3] \times [-3, 3] \times [-45, 45] \times [0.7, 1.3] \subset \mathbf{R}^4$. Table II summarizes the results obtained for three different images. The results are similar in the three cases. As can be seen, very good results are obtained in the estimation of ϕ and α; for instance, the average estimation error of the rotation angle is as small as 3°. As for the estimation error in translations, the average error is always smaller than 1 pixel, and the median is even better. The maximum errors reported in the table are really outliers.

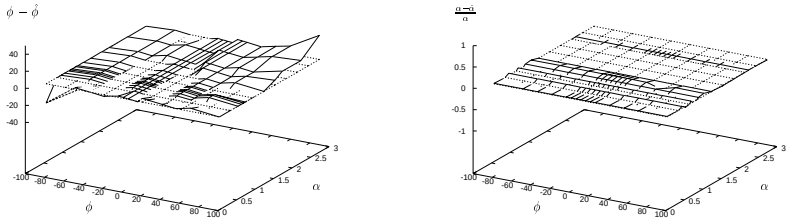
5 Conclusions

A uniform approach for the estimation of vertical and horizontal translations, rotations, and scalings has been presented. Projections along different directions are defined for each motion component. Estimating each of the motion components becomes a simple

for d_x and ϕ to each experiment number.



(a) Image Boat

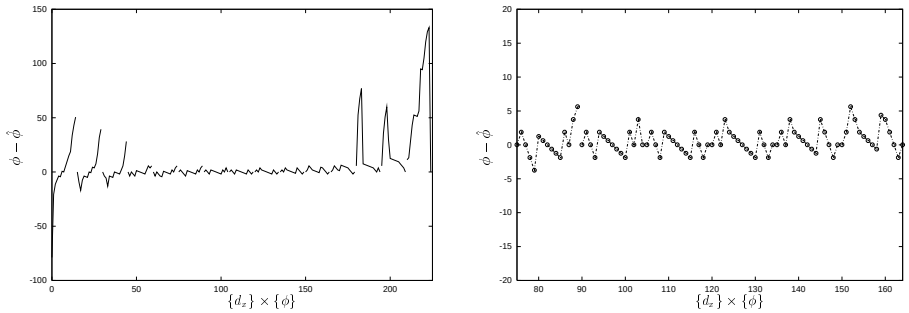


(b) Image Corridor

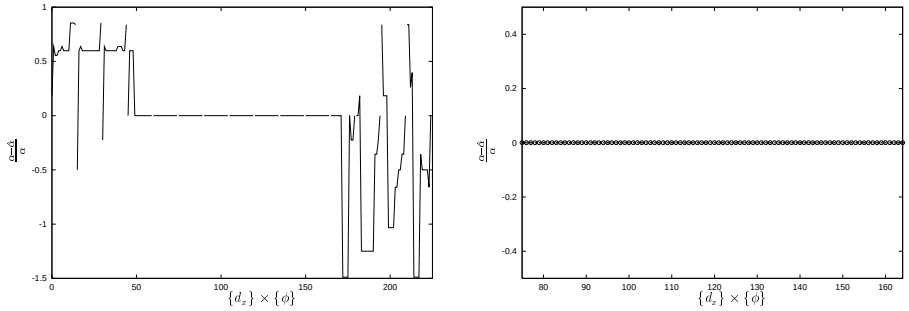
Figure 5. Errors in rotation and scaling estimates. Results for two different images are presented. Errors $\phi - \hat{\phi}$ are plotted on the left, and errors $\frac{\alpha - \hat{\alpha}}{\alpha}$ are plotted on the right. Rotation estimation errors are in degrees; scaling estimation errors are dimensionless relative units.

Table II. Statistics on motion parameters estimation errors. The error functions are $e_{d_x} = |d_x - \hat{d}_x|$, $e_{d_y} = |d_y - \hat{d}_y|$, $e_\phi = |\phi - \hat{\phi}|$, and $e_\alpha = |\frac{\alpha - \hat{\alpha}}{\alpha}|$.

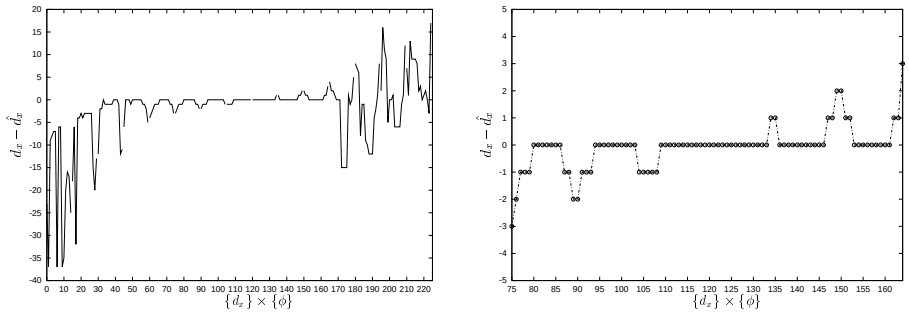
Image	Parameter	Mean	Std. Dev.	Median	Minimum	Maximum
Boat	d_x	0.83	0.74	0.65	0.00	5.00
	d_y	0.84	0.96	0.61	0.01	15.89
	ϕ	3.00	2.81	2.06	0.00	15.0
	α	0.044	0.311	0.027	0.0001	6.967
Corridor	d_x	0.79	0.68	0.62	0.00	4.00
	d_y	0.88	1.15	0.60	0.01	15.52
	ϕ	2.00	1.43	1.65	0.00	6.61
	α	0.033	0.042	0.027	0.000	0.609
Rubik	d_x	0.95	0.74	0.79	0.00	4.00
	d_y	0.76	0.62	0.59	0.01	3.65
	ϕ	2.26	1.71	1.90	0.00	8.96
	α	0.029	0.018	0.027	0.000	0.098



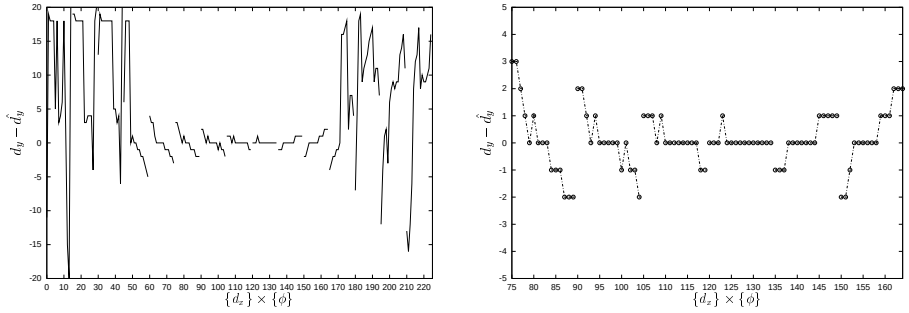
(a) Error in ϕ



(b) Error in α



(c) Error in d_x



(d) Error in d_y

Figure 6. Errors in rotation, scaling and translations in image Corridor for transformations involving displacements in x and rotations. Errors for the experiments in all the ranges in d_x and ϕ are shown on the left column. The center of this range (which corresponds to small d_x), is zoomed in and shown on the right column.

common problem: estimation of the translation between two one-dimensional signals. This is possible by exploiting the features of rotation and scale invariance that log-polar images possess.

Experiments reveal that even very large rotations and scalings can coexist and still be accurately estimated (in the absence of translational motion). Similarly, large displacements (both vertical and horizontal) can simultaneously be estimated (assuming no rotation nor change of scale).

When all four motions components occur together, empirical evidence show that the estimation of rotation and scaling can tolerate moderate translations. This key observation allows us to "undo" the effects of these motions for a more reliable estimation of the translational motion components. For larger displacements, however, this simple method would generally fail. This is probably less restrictive than it sounds, because the algorithm is intended to be used in active tracking contexts, where an effective fixation strategy allows to keep object shift small. It is important to bear in mind that the computational advantages gained with the small 32×64 log-polar images imply some loss in precision beyond the fovea area.

References

[1] I. Ahrns and H. Neumann. A view-based approach for real-time fixation using log-polar mapping. In C. Freksa, editor, *Proc. in Artificial Intelligence*, pages 89–96, June 1998.

[2] M. Bolduc and M. D. Levine. A review of biologically motivated space-variant data reduction models for robotic vision. *Computer Vision and Image Understanding (CVIU)*, 69(2):170–184, Feb. 1998.

[3] C. Capurro, F. Panerai, and G. Sandini. Dynamic vergence using log-polar images. *Intl. Journal of Computer Vision*, 24(1):79–94, 1997.

[4] J. Hynoski and H. R. Wu. Active vision — a survey of the field and research directions. Technical Report 95-04, Faculty of Computing and Information Technology. Dept. of Robotics and Digital Technology. Monash University, May 1995.

[5] V. Krüger. Optical flow estimation in the complex logarithmic plane. Master's thesis, Lehrstuhl für Kognitive Systeme, Institutus für Informatik und Praktische Mathematik, 1995.

[6] L. Lucchese. A frequency domain technique based on energy radial projections for robust estimation of global 2D affine transformations. *Computer Vision and Image Understanding (CVIU)*, 81:72–116, 2001.

[7] V. J. Traver and F. Pla. An optimization approach for translational motion estimation in log-polar plane. In *Intl. Conf. on Computer Analysis of Images and Patterns (CAIP)*, Warsaw, Poland, Sept. 2001. Accepted.

[8] Y. Zhang, C. Wen, and Y. Zhang. Simultaneously recovering affine motion and defocus blur using moments. In *Intl. Conf. on Pattern Recognition (ICPR)*, volume 3, pages 881–884, Barcelona, Spain, Sept. 2000.