Log-polar Mapping in Generalized Least-Squares Motion Estimation*

Raul Montoliu, V. Javier Traver, Filiberto Pla

Computer Vision Group
Universitat Jaume I
E12080-Castellón (Spain)
{montoliu|vtraver|pla}@uji.es

ABSTRACT
This work presents a study of how log-polar images can be combined with a generalized least squares-based technique in order to get an accurate and fast parametric motion estimation algorithm. The possibility of replacing the classical hierarchical algorithm—based on using a pyramid of multiple resolution—by an algorithm which gets a good approximation of the motion parameters using log-polar images, is also explored.

KEYWORDS
Motion estimation, Generalized least-squares, Log-polar mapping.

1 Introduction

Motion estimation is an important problem in computer vision, and much effort has been paid to solve it [7]. In spite of it, there are still challenging research open problems to make solutions faster, more robust and accurate, or more general.

Generalized least-squares methods [3] are an interesting approach to extend the applicability of least-squares techniques (e.g., to non-linear problems), and to get more accurate results [14]. The application of these framework to parametric motion estimation [4] has shown that very accurate motion estimates are possible. However, these approaches may suffer from two drawbacks: their high computational cost, and the possibility of getting trapped into local minima. Traditionally, hierarchical solutions have been proposed to deal with these problems [1].

Log-polar mapping [2] has attracted the attention of some researchers in the last years, because of its interesting characteristics in active vision [9, 6], and pattern recognition [12, 8]. However, little work exists on how log-polar images could be exploited in motion estimation algorithms [13], or how motion estimation could directly be applied to log-polar images [10].

The work presented in this paper addresses the use of log-polar mapping for motion estimation in several respects. On the one hand, a formulation is given to apply generalized least-squares methods directly on log-polar images. On the other hand, we explore how log-polar mapping could advantageously be combined with motion estimation on conventional, uniformly sampled (cartesian) images. In particular, we study the use of log-polar mapping as an alternative to Gaussian pyramids.

The rest of the paper is organized as follows. Section 2 gives an introduction to log-polar mapping and how log-polar images can be obtained from cartesian images. A brief introduction to generalized least-squares techniques, and their application to motion estimation on cartesian and log-polar images is presented in Section 3. Section 4 is devoted to explain the three different algorithms proposed in this paper, which are tested in experiments reported in Section 5. Finally, Section 6 summarizes the main conclusions of this work, and suggests what could be done as future work.

2 Log-polar mapping

The log-polar mapping used here defines the log-polar coordinates

\[(\xi, \eta) \triangleq \left( \log_\rho \left( \frac{\rho}{\rho_0} \right), q \cdot \theta \right), \tag{1} \]

with \((\rho, \theta)\) being the usual polar coordinates, defined from the cartesian coordinates \((x, y)\) as usual, i.e.,

\[(\rho, \theta) \triangleq \left( \sqrt{x^2 + y^2}, \arctan \frac{y}{x} \right). \]

The parameters of the transform are \(q, \rho_0\), and \(a\). Because of the discretization, the continuous coordinates \((\xi, \eta)\) become the discrete ones \((u, v) = ([\xi], [\eta])\). \(0 \leq u < R\). \(0 \leq v < S\), with \(R\) and \(S\) being the number of rings and sectors, respectively, of the log-polar image. Having chosen \(\rho_0\), \(\rho_1\) (the radius of the innermost ring), and \(\rho_{\text{max}}\) (the radius of the visual field), the transformation parameter \(a\) is computed as \(a = \exp(\ln(\rho_{\text{max}})/R)\). From the number of sectors \(S\), the angular resolution is found as \(q = \frac{\pi}{2S}\) sectors/radian. 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, our software-based log-polar transformation builds the map \( L \), where \( L(i,j) \) is the set of log-polar pixels \((u,v)\) intersecting the cartesian pixel \((i,j)\). The gray level at log-polar pixel \((u,v)\) is found by averaging the gray levels of all cartesian pixels \((i,j)\) intersecting \((u,v)\). The log-polar image resulting from applying the log-polar transformation \( L \) on a cartesian image \( I \) can be denoted as \( L(I) \). For visualization purposes, a log-polar image \( L \) can be mapped back to cartesian domain by the inverse transformation \( L^{-1}(L) \). An example is shown in Fig. 1.

![Log-polar mapping example](image)

**Figure 1.** Log-polar mapping: (a) a \( 10 \times 16 \) log-polar grid \( (L) \); (b) a \( 256 \times 256 \) cartesian image \( I \); (c) log-polar (cortical) image resulting from applying the log-polar mapping to image in (b), \( L(I) \); (d) inverse log-polar transform applied to image in (c), \( L^{-1}(L(I)) \).

One of the greatest advantages of log-polar images is the selective data reduction: log-polar images achieve a significant data compression (compare the sizes of images (b) and (c) in Fig. 1), without reducing the field of view (notice that images (b) and (d) in this same figure have the same field of view), and with a high resolution at the point of interest (observe the radially decreasing resolution in Fig. 1(d)). Log-polar images exhibit many other advantages, like the well-known edge invariance [12], and the fact that centered rotations and scalings become simple translations in the log-polar domain. This feature is exploited in the motion model employed in this work.

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).

### 3 Generalized least squares-based motion estimation

#### 3.1 Theoretical formulation

The Generalized Least Squares (GLS) algorithm [3, 11] is based on minimizing an objective function \( O \) over a set \( R \) of \( r \) observation vectors, \( R = \{L_1, \ldots, L_r\} \),

\[
O = \sum_{L_i \in R} (F(\chi; L_i))^2,
\]

where \( \chi = (\chi^1, \ldots, \chi^p) \) is a vector of \( p \) motion parameters, and \( L_i \) is a vector of \( n \) observations, \( L_i = (L_i^1, \ldots, L_i^n), i = 1 \ldots r \).

The equation (2) is non-linear, but can be linearized using the Taylor expansion and neglecting the second and higher order derivatives. This implies that an iterative solution has to be found. At each iteration, the algorithm estimates \( \Delta \chi \), that improves the parameters as follows:

\[
\chi_{t+1} = \chi_t + \Delta \chi
\]

The increment \( \Delta \chi \) is calculated (see [3, 11]) using the following expressions:

\[
\Delta \chi = (A^T(BB^T)^{-1}A)^{-1} A^T(BB^T)^{-1} W
\]

\[
B = \begin{pmatrix}
B_1 & 0 & 0 & 0 \\
0 & B_2 & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & B_r
\end{pmatrix}_{(r \times (r \times n))}
\]

\[
A = \begin{pmatrix}
A_1 \\
A_2 \\
\vdots \\
A_r
\end{pmatrix}_{(r \times p)}
\]

\[
W = \begin{pmatrix}
w_1 \\
w_2 \\
\vdots \\
w_r
\end{pmatrix}_{(r \times 1)}
\]

\[
B_i = \left( \frac{\partial F_i(\chi; L_i)}{\partial L_i^1}, \frac{\partial F_i(\chi; L_i)}{\partial L_i^2}, \ldots, \frac{\partial F_i(\chi; L_i)}{\partial L_i^n} \right)_{(1 \times n)}
\]

\[
A_i = \left( \frac{\partial F_i(\chi; L_i)}{\partial \chi^1}, \frac{\partial F_i(\chi; L_i)}{\partial \chi^2}, \ldots, \frac{\partial F_i(\chi; L_i)}{\partial \chi^p} \right)_{(1 \times p)}
\]

\[
w_i = -F_i(\chi; L_i)
\]

#### 3.2 GLS for motion estimation

In motion estimation problems [5, 4] the objective function is based on the assumption that the grey level of all the pixels of a region remains constant between two images in a sequence. The motion parameter vector, \( \chi \), depends on the motion model being used. For each point \( i \), the vector
of observation \( L_i \) has three elements: column, row and grey level of the second image at these coordinates.

In this work, a similarity motion model is used, which is able to deal with translations, scalings, and rotations, and is defined with a 4-parameter motion vector \( \chi = (\delta_x, \delta_y, \alpha, \phi) \), where \( \delta_x \) and \( \delta_y \) are the horizontal and vertical shifts with respect to the center of the transformation, \( \alpha \) is the scale factor, and \( \phi \) is the rotation angle.

In the rest of this section, the application of the GLS algorithm in motion estimation in cartesian and log-polar images is presented.

### 3.2.1 Cartesian images

For cartesian images \( I_1, I_2 \), the next equation is used as objective function:

\[
O_C = \sum_{L_i \in \mathcal{L}} (F_i(\chi, L_i))^2 = \sum_{L_i \in \mathcal{L}} (I_1(x'_i, y'_i) - I_2(x_i, y_i))^2
\]

(5)

where \( I_1(x'_i, y'_i) \) is the grey level of the first image in the sequence at the transformed point \( (x'_i, y'_i) \), and \( I_2(x_i, y_i) \) is the grey level of the second image in the sequence at point \( (x_i, y_i) \). Here, \( L_i = (x_i, y_i, L_{(x_i, y_i)}) \). In this case, the affine motion model is used, which includes the similarity model, because it is more general than it: besides translations, rotations, and scaling, it can also deal with shear deformations. The affine motion model is defined with 6 parameters as follows:

\[
x'_i = a_i x_i + b_i y_i + c_1
\]

\[
y'_i = a_2 x_i + b_2 y_i + c_2
\]

(6)

Therefore, the parameter vector is \( \chi = (a_1, b_1, c_1, a_2, b_2, c_2) \). The components of \( \chi \) can be expressed in terms of \( (\delta_x, \delta_y, \alpha, \phi) \) as follows:

\[
\phi = \arctan \frac{b_1}{a_1}, \quad \alpha = \frac{a_1}{\cos(\phi)}
\]

\[
d_x = c_1 - (M_x(1 - a_1) - M_y b_1)
\]

\[
d_y = c_2 - (M_y(1 - a_2) - M_x b_2)
\]

(7)

where \( M_x \) and \( M_y \) are half the number of columns and rows of the image, respectively.

### 3.2.2 Log-polar images

For log-polar images \( I_1, I_2 \), the next equation is used as objective function:

\[
O_{LP} = \sum_{L_i \in \mathcal{L}} (F_i(\chi, L_i))^2 = \sum_{L_i \in \mathcal{L}} (I_1(x'_i, y'_i) - I_2(u_i, v_i))^2
\]

(8)

with

\[
u' = u + \frac{\cos(v/q)}{\rho \alpha^u \ln a} c_1 + \frac{\sin(v/q)}{\rho \alpha^u \ln a} c_2 + u_0
\]

\[
u' = v - \frac{q \sin(v/q)}{\rho \alpha^u} c_1 + \frac{q \cos(v/q)}{\rho \alpha^u} c_2 + v_0
\]

(9)

where \( \rho, q, \) and \( \rho_0 \) are the parameters used to generate the log-polar image (Section 2). \( q \) and \( \rho \) represent the column and the row numbers in the cortical image. In this case, \( \chi = (c_1, c_2, u_0, v_0) \). The motion parameter \( \chi = (c_1, c_2, u_0, v_0) \) relates to parameters \( (\delta_x, \delta_y, \alpha, \phi) \) as follows:

\[
d_x = c_1, \quad d_y = c_2
\]

\[
\alpha = \alpha^{\rho_0}, \quad \phi = \frac{v_0}{q}
\]

(10)

### 4 Proposed algorithms

Three motion estimation algorithms, based on GLS, are proposed in this work. We call them H-GLS-C, GLS-LP, and LP-GLS-C:

- **H-GLS-C** (Hierarchical GLS on Cartesian images). It is based on estimating the motion parameters using a pyramid of images at multiple resolutions. The algorithm starts by applying GLS (Section 3) at the coarsest resolution, and the parameters estimated at level \( l \) in the pyramid are used as initial parameters for level \( l + 1 \). The estimated parameters correspond to the original image (i.e., at the highest resolution).

- **GLS-LP** (GLS on log-polar images). Motion parameters are estimated directly in the cortical (log-polar) images, which are computed from the original cartesian images using the log-polar mapping, as described in Section 2.

- **LP-GLS-C** (GLS on cartesian images, using log-polar images for initialization). In this case, the algorithm GLS-LP is used to initialize the motion estimation algorithm for cartesian images. Therefore, it is an alternative to using a hierarchy of cartesian images, as in H-GLS-C, which try to exploit the properties of log-polar images to help GLS converge.

### 5 Experimental work

#### 5.1 Experiment set-up

**Image transformations.** A series of image transforms has been carried out, from the well-known Lena image (see Fig. 1), in order to test the proposed algorithms. In the first four series, only one parameter is varied, while the other three are set to a fixed value. In series 5 and 6, all the parameters are changed, but translation is made to be the
dominant motion component in the fifth series, while rotation and scaling are dominant in the sixth series. In the last series, all parameters are varied, combining small and big translations with small and big scalings and rotations. This information is summarized in Table 1.

Table 1. Values used to generate the series.

<table>
<thead>
<tr>
<th>No.</th>
<th>(c_1)</th>
<th>(c_2)</th>
<th>(\alpha)</th>
<th>(\phi)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>([-30, -30])</td>
<td>0.0</td>
<td>1.0</td>
<td>0.0</td>
</tr>
<tr>
<td>2</td>
<td>0.0</td>
<td>([-30, -30])</td>
<td>1.0</td>
<td>0.0</td>
</tr>
<tr>
<td>3</td>
<td>0.0</td>
<td>0.0</td>
<td>([0.7, -1.3])</td>
<td>0.0</td>
</tr>
<tr>
<td>4</td>
<td>0.0</td>
<td>0.0</td>
<td>1.0</td>
<td>([-30, -30])</td>
</tr>
<tr>
<td>5</td>
<td>([-30, -30])</td>
<td>([-30, -30])</td>
<td>0.9.1</td>
<td>-2.0</td>
</tr>
<tr>
<td>6</td>
<td>2.5</td>
<td>2.5</td>
<td>([0.7, -1.3])</td>
<td>([-30, -30])</td>
</tr>
<tr>
<td>7</td>
<td>([0, -10])</td>
<td>([0, -10])</td>
<td>([0.7, -1.3])</td>
<td>([-30, -30])</td>
</tr>
</tbody>
</table>

Parameter setting. The size of used cartesian images is \(256 \times 256\). When using pyramids, a total of 4 levels has been used. For the log-polar transform, we set \(R \times S = 32 \times 64\), \(\rho_0 = 5\) and \(\rho_{\text{max}} = 128\).

5.2 Results

In the following paragraphs, motion estimates obtained with the different proposed algorithms when applying image transformations with known motion parameters are reported. Superscripts \(C\), \(L\), and \(M\) are added to the estimated motion parameters to indicate which algorithm was used, either H-GLS-C, GLS-LP, or LP-GLS-C, respectively.

Varying only one parameter at a time (Series of experiments 1–4). In Fig. 2, motion estimates obtained with H-GLS-C are compared to those obtained with GLS-LP. As can be observed, when translations are small-medium, both algorithms yield good estimates, but the performance of GLS-LP gracefully degrades as translations increase. This result is linked to the geometry and non-uniform resolution of log-polar images. However, and as one could expect, log-polar images deals much better with large scalings and rotations.

Varying all parameters, with translation being dominant (Series 5). Plots in Fig. 3 show estimates when the dominant motion is translation. The three proposed algorithms perform well under small-medium translations, but GLS-LP and LP-GLS-C behave poorly when translations are large. Big translations affect not only the estimation of the translational parameters, but also the scaling factor and the rotation angle.

Varying all parameters, with rotation and scaling being dominant (Series 6). Results when rotation and scaling are the dominant motion are depicted in Fig. 4. In this case, H-GLS-C gives erroneous estimates under strong rotations or scalings. The hierarchical coarse-to-fine processing does not provide any help in this case, and H-GLS-C seems
to get trapped into local minima. On the contrary, GLS-LP gives quite good results in all the deformation range. Although for big rotations/scalings the accuracy is not so high, estimates are still reasonable good. As for LP-GLS-C, interestingly, as GLS-LP is used to initialize the GLS algorithm, much better estimates result, because GLS-LP provides a good starting point for GLS to converge to a good global optimum, thus refining the solution provided by GLS-LP.

Varying all parameters (Series 7). Similar observations can be made in results presented in Fig. 5, which indicates that LP-GLS-C seems an interesting algorithm, and suggests that log-polar images can be favorably be combined with cartesian images in some image processing algorithms, and motion estimation in particular.

Computation times. Even though GLS on log-polar images may not yield as accurate estimates as GLS on cartesian images, the use of log-polar images provides a substantial speed-up. The use of a hierarchy of images at different resolutions is usually of help to make the algorithm faster. However, when the coarser levels do not guide the search for finer levels, the overall time can be increased. The use of log-polar images as an initialization stage offers a better average behavior than the pyramid, because it exhibits a more stable behavior than the best and worst cases with pyramids.

6 Conclusions

A Generalized Least-Squares motion estimation algorithm, previously developed for cartesian images, has been formulated to log-polar images. The results have been compared to those of the original formulation, and reveal that the GLS framework can successfully be applied to log-polar images. An advantage of using log-polar images is an increase in the speed of the algorithm (due to the selective data reduction of log-polar images). However, when using log-polar images, motion estimation accuracy degrades as the magnitude of translations increase, whereas a hierarchical GLS applied to cartesian images behaves well in all the deformation range.

Nevertheless, when rotation and scaling are the dominant motion, better results are obtained with log-polar images. As a result, it has been proposed to use motion estimates obtained in this way to initialize the GLS algorithm applied on cartesian images. Interestingly, this outperforms the results that can be achieved with a traditional pyramid of cartesian images.

In this work, a similarity motion model has been used. It would be interesting to study if the results obtained here generalize to more complex motion models. Although computation times are smaller when GLS is applied to log-polar images, further work is needed to make implementation faster so that real-time is feasible. This would allow application of GLS in active vision tasks, such as target tracking. In this sense, it deserves attention the study of the performance of the GLS-based algorithms when only part of the image (corresponding to a foveated target) undergoes a transformation.

Figure 4. Motion estimates when varying all parameters, rotation and scaling being dominant.
Figure 5. Motion estimates when varying all parameters.

References


