Robust Egomotion Estimation from the Normal Flow using Search Subspaces*

César Silva José Santos-Victor

Abstract

We address the problem of egomotion estimation for a monocular observer moving under arbitrary translation and rotation, in an unknown environment. The method we propose is uniquely based on the spatio-temporal image derivatives, or the normal flow. We introduce a search paradigm which is based on geometric properties of the normal flow field, and consists in considering a family of search subspaces to estimate the egomotion parameters. Various algorithms are proposed within this framework. In order to decrease the noise sensitivity of the estimation methods, we use statistical tools, based on robust regression theory. Finally, we present and discuss a wide variety of experiments with synthetic and real images, for various kinds of camera motion.

Index Terms: motion analysis, egomotion, optical flow, normal flow.

1 Introduction

We address the problem of egomotion estimation for a monocular moving observer. The problem consists in determining the 3D motion parameters, by observing an image sequence over time. This is a real need for many robotic applications where an autonomous system must be able to estimate and/or control its motion parameters before any higher level tasks can be addressed (like a point to point moves or homing procedures).

The first step to estimate egomotion or structure from motion is the computation of displacement between consecutive frames. Usual approaches are based either on point correspondences [10]; on the estimation of the dense motion field, identified by the optical flow [6, 2]; or finally in the so-called direct methods [9, 3, 5]. The correspondence problem or the optical flow estimation are, in general, ill-posed [1], and it is usually necessary to introduce very restrictive assumptions about the observed scenes. These solutions often require extensive amounts of computation.

The direct methods are less demanding than the previous ones, and use the image brightness information directly to recover the motion parameters. The only image flow component

*The authors are with the Computer Vision Lab., VisLab, at the Instituto de Sistemas e Robótica; Instituto Superior Técnico, Av. Rovisco Pais, 1096 Lisboa Codex, Portugal. E-mail: {cesar,jasv}@isr.ist.utl.pt
This work was partially funded by projects PRAXIS/3/3.1/TPR/23/94, JNICT-PBIC/TPR/2550/95, EC ESPRIT-LTR Narval and a grant from the PRAXIS-XXI research program.

that can be estimated based on local measurements, is the motion along the image gradient direction, the normal flow [8]. Thus, direct methods are usually based on the normal flow or the spatio-temporal image derivatives [12]. Here we focus the approach developed by Fermüller and Aloimonos [3, 4, 5] who introduced a method based on a selection of image points that form global patterns in the image plane, using the orientation of global normal-flow vectors. This approach considers the complete egomotion estimation problem as a pattern recognition problem, introducing a set of geometric constraints that reveal a global structure hidden in normal-flow fields. Then, the use of global data favors the robustness of the algorithms.

The approach we follow is related to previous work [4, 13], and the method consists in searching the image for particular geometric properties of the normal flow, tightly connected to the egomotion parameters. Hence, rather than considering the whole set of image flow data, we use only the image sites, that have special geometric properties, which convey relevant information about the observer motion. As proposed by Heeger and Jepson [6], one can solve separately the rotational and translational motion parameters due to bilinear nature of the image motion equations. We also subdivide the problem, but in a different way. The difference is that we split the search domain in several geometric figures, and estimate sets of motion parameters for each one of them.

The salient aspects of the method proposed here, are the exclusive use of the spatio-temporal image derivatives; the introduction of a bounded search paradigm by using a subdivided search domain, based on geometric constraints of the flow field; the fact that it can cope with arbitrary linear and angular observer motion and the effort that it is made on robust estimation in order to decrease the noise sensitivity [11, 14].

2 Problem Statement

Given the camera linear velocity $\mathbf{t} = [U \ V \ W]^T$, angular velocity $\boldsymbol{\omega} = [\omega_1 \ \omega_2 \ \omega_3]^T$ and the scene depth at each point, Z, the motion field induced on the image plane, can be calculated at every pixel by the following well-known vector equation [6] :

$$\mathbf{v}(\mathbf{x}) = \rho(\mathbf{x})(\mathbf{x} - \boldsymbol{\Sigma}) + \mathbf{B}(\mathbf{x})\boldsymbol{\omega} \tag{1}$$

where $\mathbf{v}(\mathbf{x}) = [u(\mathbf{x}) \ v(\mathbf{x})]^T$ is the optical flow observed at a given image site, $\mathbf{x} = [x \ y]^T$; $\boldsymbol{\Sigma} = [\sigma \ \eta]^T = [fU/W, fV/W]^T$ is the Focus of Expansion (FOE) and corresponds to the projection of the observer linear velocity, $\mathbf{t}$, in the image plane. The function $\rho(\mathbf{x})$ is the inverse of the time-to-crash, given by $\rho(x, y) = W/Z(x, y)$. The rotation is multiplied by:

$$\mathbf{B}(\mathbf{x}) = \begin{bmatrix} \frac{xy}{f} & -(\frac{x^2}{f} + f) & y \\ (\frac{y^2}{f} + f) & -\frac{xy}{f} & -x \end{bmatrix} \tag{2}$$

2.1 Egomotion estimation from the optical flow field

The motion field $\mathbf{v}$ is a linear function of $\boldsymbol{\omega}$ and bilinear in ρ and $\boldsymbol{\Sigma}$. With N sample points we can solve the bilinear equation (1), using the following system of equations:

$$\underbrace{\begin{bmatrix} \mathbf{v_1} \\ \vdots \\ \mathbf{v_N} \end{bmatrix}}_{\bar{\mathbf{v}}} = \underbrace{\begin{bmatrix} \mathbf{x_1} - \boldsymbol{\Sigma} & \dots & \mathbf{0} & \mathbf{B_1} \\ \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \dots & \mathbf{x_N} - \boldsymbol{\Sigma} & \mathbf{B_N} \end{bmatrix}}_{\mathbf{A}(\boldsymbol{\Sigma})} \underbrace{\begin{bmatrix} \rho_1 \\ \vdots \\ \rho_N \\ \boldsymbol{\omega} \end{bmatrix}}_{\mathbf{p}} \tag{3}$$

where $\bar{\mathbf{v}}$ is a $2N$ vector containing the optical flow of N points; $\mathbf{p}$ is a $N + 3$ vector containing the 3 unknown rotational parameters and the N unknown time-to-crash values. Finally, $\mathbf{A}(\boldsymbol{\Sigma})$ is a $2N$-by-$(N + 3)$ matrix which is completely computable if the FOE, $\boldsymbol{\Sigma}$, is known. The system of equations (3) can be solved for a known $\boldsymbol{\Sigma}$ provided that $N \geq 3$. In general, however, $\boldsymbol{\Sigma}$ is unknown, and the following cost function has to be minimized, jointly in $\boldsymbol{\Sigma}$ and $\mathbf{p}$[1]:

$$\min_{\hat{\boldsymbol{\Sigma}}} \left\{ \min_{\hat{\mathbf{p}}} \left\{ ||\bar{\mathbf{v}} - \mathbf{A}(\boldsymbol{\Sigma})\mathbf{p}||^2 \right\} \right\} \tag{4}$$

The main difficulty is that equation (1) depends on the *product* of $\boldsymbol{\Sigma}$ by the unknown 2-D function, $\rho(\mathbf{x})$, that depends on the depth for every pixel. To eliminate this dependency on $\rho(\mathbf{x})$, one can consider a unit vector $\mathbf{d}(\mathbf{x}, \boldsymbol{\Sigma})$, perpendicular to the direction $(\mathbf{x} - \boldsymbol{\Sigma})$ such that $\mathbf{d}(\mathbf{x}, \boldsymbol{\Sigma}) \cdot (\mathbf{x} - \boldsymbol{\Sigma}) = 0$. Notice that $\mathbf{d}(\mathbf{x}, \boldsymbol{\Sigma})$ is well determined if $\boldsymbol{\Sigma}$ is known. Then, taking the inner product of both sides of equation (1), the result does not depend on $\rho(\mathbf{x})$:

$$\mathbf{d}(\mathbf{x}, \boldsymbol{\Sigma}) \cdot \mathbf{v}(\mathbf{x}) = \mathbf{d}(\mathbf{x}, \boldsymbol{\Sigma}) \cdot [\mathbf{B}(\mathbf{x})\boldsymbol{\omega}] \tag{5}$$

Hence, the components of the optic flow perpendicular to lines going through the image pixel $\mathbf{x}$ and the FOE, do not depend on the camera translation, and depend linearly on the rotation parameters. At least three points are needed to compute $\boldsymbol{\omega}$ for a fixed $\boldsymbol{\Sigma}$.

[1] In this article we use the following notation: $\min_{\hat{\theta}} \{E(\theta)\}$ represents the minimum of the function $E(\theta)$, where the index $\hat{\theta}$ is the value of θ for which $E(\theta)$ is minimal.

Using this approach for egomotion estimation has the drawback that the minimization process involves searching the FOE in a given domain, which may be excessively time consuming; and the need to compute the motion field at a sufficiently large number of pixels.

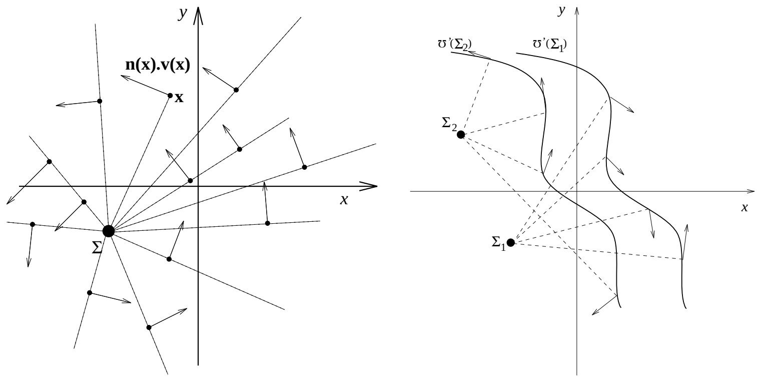
2.2 The aperture problem constraint

Due to the well known constraint of the *aperture problem* [9], we can only observe the projection of the optical flow, $\mathbf{v}$, on the direction of the image gradient, $\mathbf{n}$. At each image point, the normal flow, $v_n = \mathbf{n} \cdot \mathbf{v}$, provides a single constraint on the unknown components of the flow; and the estimation problem stated in equation (3) cannot be solved, since we only have N equations instead of $2N$.

To overcome this problem we can use the estimation approach based on equation (5), which requires a *single* equation per point that consists on the flow vectors *projected* along particular directions, $\mathbf{d}(\mathbf{x}, \mathbf{\Sigma})$. For each $\mathbf{\Sigma}$, if instead of using the flow all over the image, we select only the image sites $\boldsymbol{\mho}(\mathbf{\Sigma})$, where the normal flow and $\mathbf{d}(\mathbf{x}, \mathbf{\Sigma})$ are collinear, then the aperture problem is no longer a limitation. The cost function can be defined as follows:

$$\mathbf{E}(\mathbf{\Sigma}, \boldsymbol{\omega}) = \sum_{\mathbf{x} \in \boldsymbol{\mho}(\mathbf{\Sigma})} \left[v_n(\mathbf{x}) - \mathbf{n}^{\mathbf{t}}(\mathbf{x})(\mathbf{B}(\mathbf{x})\boldsymbol{\omega}) \right]^2, \quad \boldsymbol{\mho}(\mathbf{\Sigma}) = \{\forall \mathbf{x} : \mathbf{n}(\mathbf{x}) \cdot (\mathbf{x} - \mathbf{\Sigma}) = 0\} \qquad (6)$$

Figure 1a exemplifies a set $\boldsymbol{\mho}(\mathbf{\Sigma})$ for a given $\mathbf{\Sigma}$. It corresponds to the set of normal flow vectors that are perpendicular to a line joining the image point and the FOE. The minimization problem



(a) (b)

Figure 1: (a) $\mho(\Sigma)$ for a given FOE localization. (b) Example of a search subspace.

consists of searching the minimum of $\mathbf{E}(\mathbf{\Sigma}, \boldsymbol{\omega})$ for a large array of possible $\mathbf{\Sigma}$ values. In order to reduce the complexity of the corresponding minimization algorithm, we propose to consider the nature of the sets $\boldsymbol{\mho}(\mathbf{\Sigma})$ and define a useful set of search subspaces.

2.3 Defining a set of search subspaces

For a set of pixels $\mathbf{x} \in \mho(\mathbf{\Sigma})$, the normal flow vectors that are perpendicular to a line joining $\mathbf{x}$ and the FOE, are solely dependent on the camera rotation according to equation (5), and can be used to determine $\boldsymbol{\omega}$. If $\mathbf{\Sigma}$ is the correct location of the FOE, then the estimation residuals should be small. This principle can be used to search the FOE as illustrated in Figure 1b. For a given location $\mathbf{\Sigma_1}$, we define a subspace in the image (in this case, a curve), and the subspace points, $\mho'(\mathbf{\Sigma_1})$, where the normal flow vectors are perpendicular to the line going through the curve points and $\mathbf{\Sigma_1}$. Then we can estimate the likelihood of $\mathbf{\Sigma_1}$ being the FOE and the search process can continue by using other locations $\mathbf{\Sigma_i}$ and other curves, defining the corresponding subsets $\mho'(\mathbf{\Sigma_i})$ (see Figure 1b)[2].

In this work we estimate the egomotion parameters based on the general framework of defining various possible search subspaces. The direction of the "interesting" normal flow vectors must be selected during the search process itself. The possibility of establishing these vectors a priori dramatically reduces the problem complexity, and will be discussed in the next section.

3 Egomotion Estimation

In general, the estimation methods presented before involve computationally demanding search algorithms. In this section we develop a low complexity estimator by subdividing the search domain, according to the geometric constraints of the normal flow.

3.1 Geometric properties of the radial and circular normal flow

The *radial normal flow* is the set of normal flow vectors with radial directions, $\mathbf{n_r}$. Consider the subset of the radial normal flow vectors that verify also: $\mathbf{n_r} \cdot (\mathbf{x} - \mathbf{\Sigma}) = 0$.

Unlike other pixels, in these image sites, the radial normal flow is also perpendicular to lines going through the FOE and depends solely on the camera rotation (see equation (5)). To express these image points, we can use polar coordinates ($x = r\cos\psi$, $y = r\sin\psi$):

$$(\cos\psi, \sin\psi) \cdot (r\cos\psi - \sigma, r\sin\psi - \eta) = 0 \quad \Leftrightarrow \quad r = (\sigma, \eta) \cdot (\cos\psi, \sin\psi) \tag{7}$$

[2]Some subspaces are inconclusive about the exact location of a given $\mathbf{\Sigma}$: this is the case of a straight line containing $\mathbf{\Sigma}$. However, finding such line can be useful to reduce the search space of the subsequent algorithms.

which describes the Γ-**circle** having the image center and the FOE, (σ, η) as diametrical opposite points, as shown in Figure 2a. The rotational component, v_r, of the radial normal flow can be written in polar coordinates:

$$v_r = \left(\frac{r^2}{f} + f\right)(\omega_1, \omega_2) \cdot (\sin\psi, -\cos\psi) \tag{8}$$

which exhibits a sinusoidal behavior in ψ for a fixed radius, r, and vanishes along a line defined by the image origin and the Axis of Rotation (AOR), $\left(f\frac{\omega_1}{\omega_3}, f\frac{\omega_2}{\omega_3}\right)$, shown in Figure 2a.

Similarly we can analyze the *circular normal flow* defined as the set of normal flow vectors with a direction $\mathbf{n_c}$, perpendicular to radial lines. We then consider the subset of the *circular normal flow* field, that verifies: $\mathbf{n_c} \cdot (\mathbf{x} - \mathbf{\Sigma}) = 0$.

For all these points, the circular normal flow is also perpendicular to lines going through the FOE, hence depending exclusively on the camera rotation (see Figure 2b). We can identify these special image sites, by using polar coordinates again:

$$(\sigma, \eta) \cdot (\sin\psi, -\cos\psi) = 0 \tag{9}$$

which defines the Ψ-**line** going through the FOE, and the image center. The rotational component of the circular normal flow vectors is given by:

$$v_r = f(\omega_1, \omega_2) \cdot (\cos\psi, \sin\psi) - \omega_3 r \tag{10}$$

This component, v_r, is affine in r along all the lines containing the image origin (Figure 2b). Considering $\omega_3 \neq 0$, v_r vanishes when

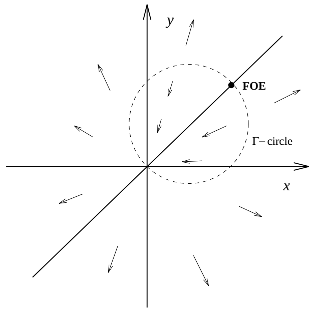
$$r = \left(f\frac{\omega_1}{\omega_3}, f\frac{\omega_2}{\omega_3}\right) \cdot (\cos\psi, \sin\psi), \tag{11}$$

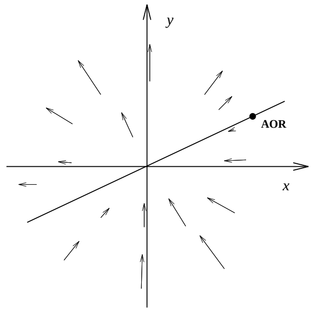
which represents a circle having the AOR and the image origin as diametrical opposite points.

Comparing equations (7) with (9) and equations (8) with (11), we see that the geometrical properties found for the radial and circular normal flow vectors are dual, as shown in Figure 2. In summary, the translational component of the radial/circular normal flow vanishes on the Γ-circle/Ψ-line, respectively; the rotational component is sinusoidal in ψ or affine in r.

3.2 Constrained search algorithms

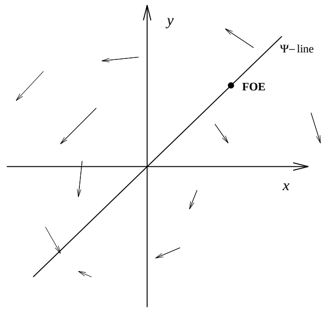
Several subspaces can be defined based on the geometric properties described in Section 3.1. We describe subspaces on cartesian coordinates by using an equation of a parameter k: $F(\mathbf{x}; k) = 0$. The following examples show the main subspaces used in this paper, as illustrated in Figure 3.

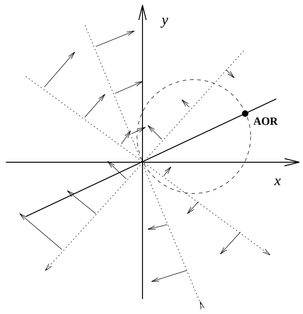




Translational component of the *radial normal flow* Rotational component of the *radial normal flow*

(a)





Translational component of the *circular normal flow* Rotational component of the *circular normal flow*

(b)

Figure 2: Geometric properties found for (a) radial and (b) circular normal flow vectors.

- **Parallel lines** (Figure 3a): $F_0(\mathbf{x}; k) = A_0 x + B_0 y + k$, where $\|(A_0, B_0)\| = 1$.
- **Radial lines** (Figure 3b): $F_r(\mathbf{x}; k) = x \sin k + y \cos k$
- **Tangential circles**[3] (Figure 3c): $F_c(\mathbf{x}; k) = x^2 + y^2 - k(x \cos \Psi + y \sin \Psi)$.

For each subspace, there is an appropriate algorithm to compute a set of motion parameters, that suggests and adequate search strategy.

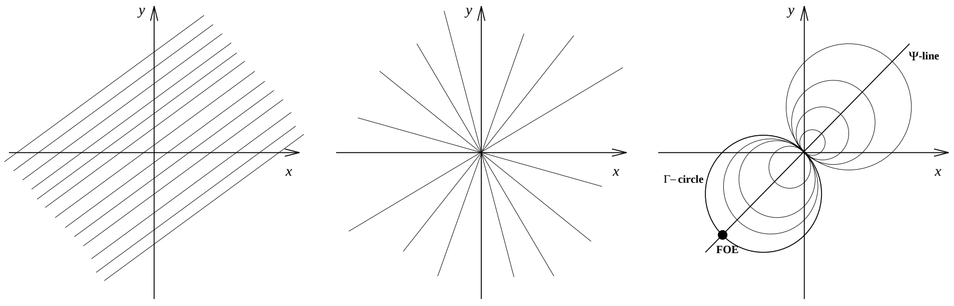
3.2.1 The Ψ-line algorithm

We can compute the *circular normal flow*, for each of the radial lines with orientation ψ, defined by $F_r(\mathbf{x}; \psi) = x \sin \psi + y \cos \psi$:

$$U_\psi(r) \;=\; \mathbf{n_c} \cdot \mathbf{v} = -\rho(x,y)(\sigma,\eta) \cdot (-\sin\psi, \cos\psi) + f(\omega_1 \cos\psi + \omega_2 \sin\psi) - \omega_3 r \qquad (12)$$

where r and ψ denote the polar coordinates of the point (x, y). The circular normal flow, U_ψ, is composed by an affine term in r, and a term that, in general, is non-linear in r. However, this

[3]The tangential circles depend on a constraint on $\mathbf{\Sigma}$, defined by: $(\sigma, \eta) \cdot (-\sin \Psi, \cos \Psi) = 0$, defining a set of circles containing the image center and having diameter k, along the Ψ-line defined in Section 3.1.



(a) (b) (c)

Figure 3: a) Parallel lines described by $F_0(x;k)$. b) Radial lines described by $F_r(x;k)$. c) Tangential circles described by $F_c(x;k)$.

non-linear component vanishes both when ψ corresponds to the FOE direction (i.e. $\psi = \Psi$), or when the FOE is located at the image origin. The rationale of the Ψ-line algorithm is to search a radial line ψ^k, containing the image origin, such that $U_\psi(r)$ is affine in r, as follows:

$$\min_{\hat{\psi}^k} \left\{ \min_{\hat{p},\hat{\omega}_3} \left\{ \boldsymbol{\mathcal{M}} \left(v_n - (p - \omega_3 r) | \psi^k \right) \right\} \right\} \tag{13}$$

where $p = f(\omega_1 \cos\psi + \omega_2 \sin\psi)$; $\boldsymbol{\mathcal{M}}(R|D)$ expresses a measurement of the residuals R on a search domain D, according to some metric (the mean of squares, the median of squares, etc). The search domain is the set of radial lines described by $F_r(\mathbf{x};k) = x\sin k + y\cos k = 0$. For each line ψ^k, at least three points are needed to estimate p and ω_3, but more points are used to increase robustness. The direction corresponding to the FOE minimizes $\boldsymbol{\mathcal{M}}$, when approximating $U_\psi(r)$ to an affine function of r. Once the Ψ-line is detected, the slope of $U_\Psi(r)$ corresponds to $-\omega_3$, while the value for $r = 0$, is a linear combination of ω_1 and ω_2, expressing the projection of (ω_1, ω_2) on the Ψ-line. Once the Ψ-line has been determined, the problem that remains to be solved is that of estimating the individual values of ω_1, ω_2, and the FOE, as only its direction is known.

3.2.2 The Φ-line algorithm

Consider a set of parallel lines ϕ^k, described by the equation $F_0(\mathbf{x};k) = 0$. Let the Φ-line be defined as the special one, containing the FOE. Select the points where the normal flow is perpendicular to the lines ϕ^k. For the Φ-line, these flow vectors depend exclusively on the camera rotation. For each selected observations, minimize and solve the expression

$$\min_{\hat{\boldsymbol{\omega}}} \left\{ \boldsymbol{\mathcal{M}} \left(v_n(\mathbf{x_i}) - \mathbf{n^t}(\mathbf{B}\boldsymbol{\omega}) | \phi^k \right) \right\} \tag{14}$$

The Φ-line corresponds to a minimum measurement $\boldsymbol{\mathcal{M}}$ of the residuals. If the direction of the ϕ lines is not collinear with the Ψ-line, we can introduce in this expression the known value of ω_3 and the linear constraint in ω_1 and ω_2, established in the Ψ-line algorithm. The intersection of the Ψ-line and the Φ-line (if not collinear) defines the position of the FOE, and the estimated rotation parameters complete the solution to the problem.

3.2.3 The Γ-circle algorithm

Suppose that the *circular* normal flow has been computed to apply the Ψ-line algorithm. In order to solve the remaining parameters, we can use the *radial* normal flow, $V_\psi(r) = \mathbf{n_r} \cdot \mathbf{v}$. In Section 3.1, we have pointed out that the translational component of V_ψ vanishes for every point on the Γ-circle. Given the known FOE-direction Ψ, we can define the set of circles Γ^k containing the image center and having diameter k, along the Ψ-line, described by the equation $F_c(\mathbf{x}; k) = 0$. If we compute $V_\psi(r)$ for the circle Γ^k and divide by $\frac{r^2}{f} + f$, we have:

$$\forall (x,y) : (x,y) \in \Gamma^k, \;\; T^k(\psi) = \frac{V_\psi(r)}{\frac{r^2}{f} + f} = \rho(x,y)\frac{r - (\sigma,\eta)\cdot(\cos\psi, \sin\psi)}{\frac{r^2}{f}+f} + \omega_1 \sin\psi - \omega_2 \cos\psi, \tag{15}$$

where r is dependent on k and ψ, to ensure that $T^k(\psi)$ is computed on the circle Γ^k. Equation (15) shows that if $\hat{\Gamma}^k$ corresponds to the Γ-circle, then $T^k(\psi)$ becomes sinusoidal in ψ. Based on this property, the algorithm consists in searching a circle Γ^k, such that $T^k(\psi)$ is sinusoidal in ψ. To estimate the sinusoid parameters, we need two points, but more data are needed to find the Γ-circle. When approximating $T^k(\psi)$ to a sinusoidal function, the Γ-circle minimizes the following criterion:

$$\min_{\hat{\omega}_1, \hat{\omega}_2} \left\{ \boldsymbol{\mathcal{M}} \left(\frac{v_n}{\frac{r^2}{f} + f} - (\omega_1 \sin\psi - \omega_2 \cos\psi) | \Gamma^k \right) \right\} \tag{16}$$

In this algorithm we only use the radial normal flow, solving the remaining egomotion parameters, namely the FOE and the individual values of ω_1 and ω_2.

3.2.4 The histogram based algorithm

After using the Ψ-line algorithm, the direction of the FOE, the value of ω_3, and a linear constraint in ω_1 and ω_2, are known. The remaining parameters can be solved from an image point verifying:

$$\begin{cases} \mathbf{n}(\mathbf{x_i}) \cdot (\mathbf{x_i} - \mathbf{\Sigma}) = 0 \\ \mathbf{x_i} \notin \Psi - line \end{cases} \tag{17}$$

For all these points, the normal flow is perpendicular to a line going through the pixel $\mathbf{x_i}$ and the FOE, and therefore is dependent only on the camera rotation:

$$v_n(\mathbf{x_i}) = \mathbf{n}(\mathbf{x_i}) \cdot (\mathbf{B}(\mathbf{x_i})\boldsymbol{\omega}) \tag{18}$$

Then, for each point in the image, we can estimate values for ω_{1_i} and ω_{2_i} using equation (18). As only the image pixels fulfilling the constraint (17) will verify equation (18), we can expect the estimates of ω_{1_i} and ω_{2_i} to be spread over a large range of values, with a concentration of coherent estimates around the correct values of ω_1 and ω_2. We estimate all possible values of $(\omega_{1_i}, \omega_{2_i})$, determine their histograms and select the most frequent values (similar to Hough transform). For the pixels corresponding to the most frequent $(\omega_{1_i}, \omega_{2_i})$, the lines perpendicular to $\mathbf{n}(\mathbf{x_i})$ should all intersect the Ψ-line at the FOE.

3.3 Parallel and Sequential algorithms

The various algorithms described here can be organized in a parallel or sequential structure, that can be generalized for other search algorithms applied for different search domains.

A parallel method to determine the egomotion parameters consists in a collection of Φ-line algorithms, using parallel lines $F_0(\mathbf{x}; k) = 0$, with different orientations. The resulting set of Φ-lines $\left\{\hat{\phi}_0, \hat{\phi}_1, \ldots, \hat{\phi}_n\right\}$ intersect at the FOE[4]. Due to its parallelism and subdivided nature, the complexity of the algorithm is extremely low.

We can transform the parallel structure in a sequential structure, if some information provided by the first search algorithm is used in the next one. Thus, we propose a sequential algorithm with two steps. In the *first step*, the Ψ-line algorithm is used to determine the FOE direction, ω_3 and one constraint on ω_1 and ω_2. The *second step* uses the results provided by the Ψ-line algorithm and computes the remaining motion parameters. We consider three different possible approaches: (a) the Φ-*line algorithm* to find a second line (the Φ-line) containing the FOE, by exploring a direction not collinear to the Ψ-line; (b) the Γ-*circle algorithm*; (c) and the *histogram based algorithm* applied to all image points.

[4] To increase the robustness of the algorithm one can produce an accumulation matrix, like in Hough transform techniques. Each line is added to this matrix, and the maximum corresponds to the FOE location.

4 Results

The algorithms we proposed in this paper, were tested in a series of experiments of increasing difficulty, using both synthetic and real image data (Figure 4). In the synthetic images, the true location of the FOE is determined using the known motion parameters. In the real images, the true FOE is measured in the image by using known relevant points of the scene. The exact location of the FOE for each sequence is represented by a square mark in Figure 4.

In Sequence 1, with synthetic images, the camera moves along the Z axis and rotates around the optical axis at 1^o/frame. In Sequence 2, the camera rotates around the Y axis at $\omega_2 = -0.00785$rad/frame, while translating. In Sequence 3, with real images, the camera undergoes a translational motion in a real cluttered scene. The true rotation parameters are approximately zero. Finally, in Sequence 4, the camera undergoes a translational motion and rotates around the Y axis, with $\omega_2 = -0.005$ rad/frame.

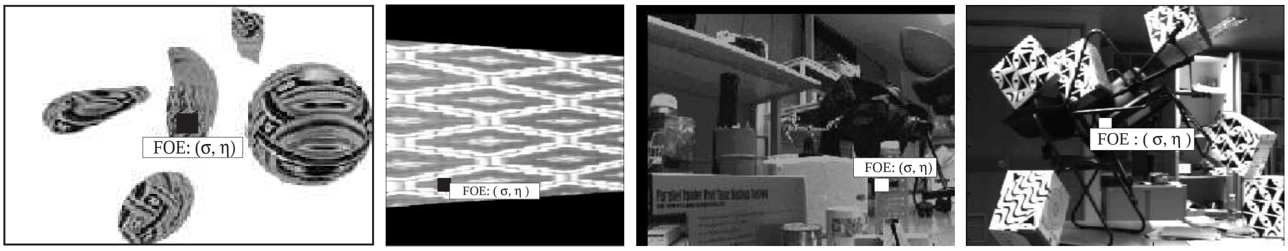
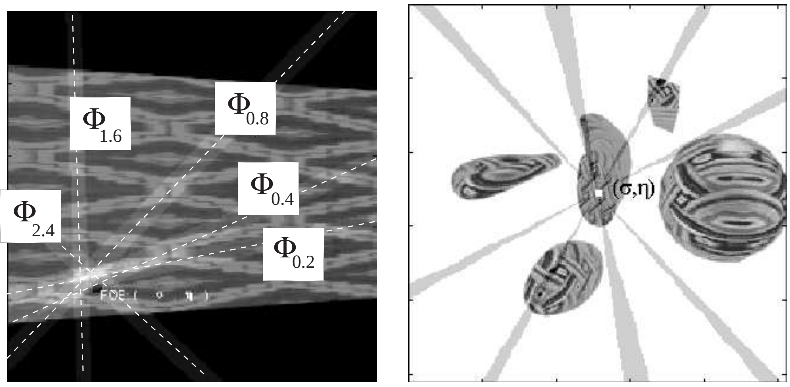


Figure 4: FOE location for the experimental Sequences 1, 2, 3 and 4.

We applied the parallel and sequential egomotion estimation algorithms. In the parallel approach, we used the least squares estimator to estimate the rotational values from the corresponding observations. In the sequential approach, we used a known robust estimator: the *least median of squares* estimator [11]. In the latter case the estimator is designed for a simple bidimensional estimation problem. See [14] for more details about the estimation procedure.

4.1 The parallel algorithm

We applied a parallel algorithm to Sequence 2, using a set of five Φ-line algorithms, corresponding to five different families of parallel lines with different orientations. Figure 5a shows the Φ lines found, superimposed with the original image, the index representing the orientation of the lines. The estimated FOE is close to the real location. By choosing the rotational parameter



(a) (b)

Figure 5: (a) Parallel algorithm applied to Sequence 2. (b) Ψ-line algorithm using Sequence 1.

estimates with smallest variance, we have $\hat{\boldsymbol{\omega}} = [-0.000324 \ -0.00915 \ -0.000632]^T$ rad/frame. As expected, ω_1 and ω_3 are very small, and ω_2 was recovered with an error of 15%.

4.2 First step of the sequential approach: the Ψ-line algorithm

We have applied the Ψ-line algorithm to all image sequences. In Sequence 1, as the FOE is located at the image origin, the Ψ-line algorithm is sufficient to calculate all the motion parameters, as all the lines going through the image origin verify the constraints of the Ψ-line algorithm (Figure 5b). Additionally, having multiple Ψ-lines with different directions, all the rotation parameters can be estimated. The estimated values of ω_1, ω_2 are negligible ($\approx 10^{-4}$ rad/frame); and we get $\omega_3 = 1.02^o/frame$, which represents a 2% error. In the other sequences, only two constraints on the rotational motion are recovered. The true FOE of each sequence is

Figure 6: Ψ-line found by the first step of the algorithm, using sequences 2, 3 and 4.

shown in Figure 6, together with the corresponding Ψ-line found. Table 1 shows the estimated parameters. Notice that the rotation estimates are very close to the true values.

Sequence	Ψ-line angle	$\omega_1 \cos\Psi + \omega_2 \sin\Psi$	ω_3
2	0.817 rad	-0.005717	-0.001496
3	-0.707 rad	-0.000135	0.000125
4	0.166 rad	0.00000	-0.000011

Table 1: Results of the Ψ-line algorithm. All rotational values are in rad/frame.

4.3 Second step of the sequential algorithm

We can now apply the three algorithms that perform the second part of the estimation process. Table 2 shows the estimates of (ω_1, ω_2). The direction of the FOE (the Ψ-line) and ω_3 were

Seq.	True values	Φ-line alg.	Γ-circle alg.	Hist. based alg.
2	$(0, -0.0079)$	$(0.00059, -0.0066)$	$(-0.00079, -0.00886)$	$(0.00029, -0.0072)$
3	$(0, 0)$	$(-0.00013, -0.00020)$	$(0.00000, -0.00000)$	$(0.00026, 0.00047)$
4	$(0, -0.0050)$	$(0.00000, -0.0055)$	$(0.00002, -0.0054)$	$(0.00039, -0.0048)$

Table 2: Estimates of (ω_1, ω_2) using the second part of the algorithm (in rad/frame).

already computed in the Ψ-line algorithm. By applying the Φ-line algorithm, the FOE location is given by the intersection of the Ψ-line and the Φ-line. Similarly, the intersection of the Γ-circle found by our algorithm and the Ψ-line determines the location of the FOE. The Ψ-lines and Γ-circles are shown in Figure 7, and intersect close the true location of the FOE.

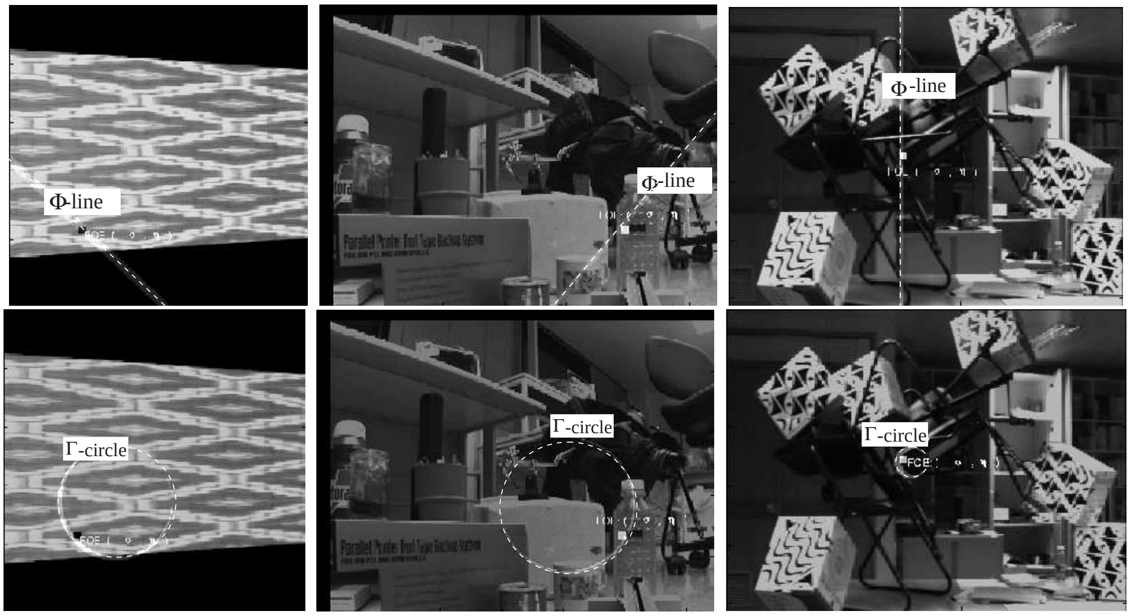
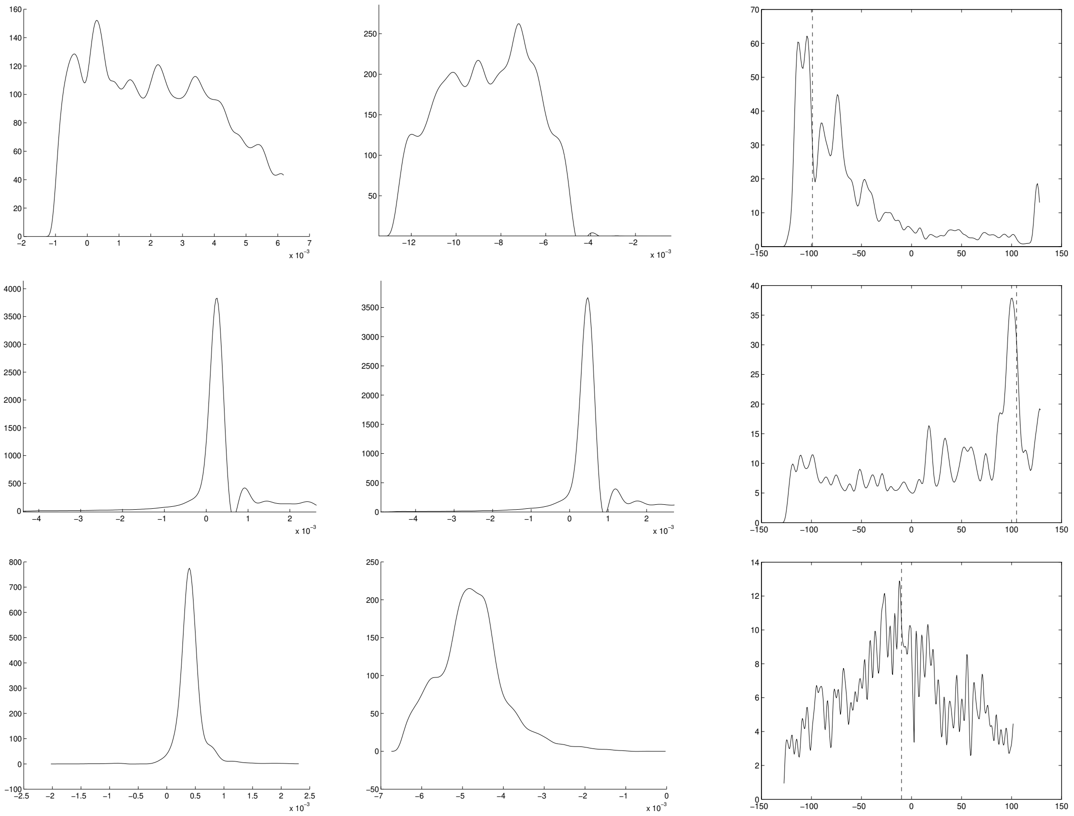


Figure 7: Results of the Φ-line (top) and Γ-circle (bottom) algorithms using sequences 2, 3, 4.

The histogram based algorithm generates a set of histograms for $(\hat{\omega_1}, \hat{\omega_2})$, computed over the



Histogram of ω_1 *Histogram of* ω_2 *Histogram of FOE in* Ψ*-line*

Figure 8: Results for the histogram based algorithm using sequences 2, 3, 4.

image. Figure 8 shows that the maximum of each histogram is well defined yielding the rotation estimates shown in Table 2. The directions of the normal flow vectors, that correspond to the maxima of the rotational histograms, intersect the Ψ-line in several points. The number of these intersections has a maximum close to the true FOE, shown in Figure 8 by a dashed line.

5 Conclusions

We have addressed the problem of egomotion estimation for a monocular observer, moving under arbitrary translation and rotation. Our approach depends solely on spatio-temporal image derivatives, and is based on the subdivision of the search domain in various subspaces, which depend on specific geometric constraints of the normal flow field. A search algorithm of low complexity is applied along a coherent family of subspaces. The resulting estimates can be used in other search algorithms, defining a global structure which can be parallel or sequential.

To decrease the sensitivity to measurement noise, we use a direct approach (computing just the normal flow) and apply robust estimators on bidimensional estimation problems. Besides, the robustness is improved if more subspaces are explored, but this choice depends on the required time of computation vs. the required accuracy of the estimates.

Future research will address the following aspects: (a) development of more sophisticated estimation methods to increase the system robustness and performance; (b) pathological or degenerate cases [7]; (c) exceptional cases of increasing difficulty, such as when the FOE is outside the image plane, or when the translational vector is parallel to the image plane; (d) by considering an active observer, one can use further constraints in the egomotion estimation process, and the algorithms proposed can be used in a closed loop navigation system.

References

[1] Y. Aloimonos, I. Weiss, and A. Banddophaday. Active vision. *Int. Journal of Computer Vision*, 1(4):333–356, January 1988.

[2] K. Daniilidis and I. Thomas. Decoupling the 3d motion space by fixation. In *ECCV96*, Cambridge, UK, April 1996.

[3] C. Fermüller. Passive navigation as a pattern recognition problem. *Int. Journal of Computer Vision*, 14(2):147–158, March 1995.

[4] C. Fermüller. Qualitative egomotion. *IJCV*, 15(1/2):7–29, June 1995.

[5] C. Fermüller and Y. Aloimonos. Direct perception of three-dimensional motion from patterns of visual motion. *Science*, 270:1973–1976, December 1995.

[6] D. Heeger and A. Jepson. Subspace methods for recovering rigid motion i: Algorithm and implementation. *Int. Journal of Computer Vision*, 7(2):95–117, January 1992.

[7] B.K.P. Horn. Motion fields are hardly ever ambiguous. *International Journal of computer Vision*, 1(3):259–274, 1987.

[8] B.K.P. Horn and B. Shunck. Determining optical flow. *Art. Intelligence*, 17:185–203, 1981.

[9] B.K.P. Horn and E.J. Weldon. Direct methods for recovering motion. *International Journal of computer Vision*, 2(1):51–76, 1988.

[10] F. Lustman O. Faugeras and G. Toscani. Motion and structure from motion from point and line matches. In *Proc. 1st Intern. Conf. Comput. Vision*, London, June 1987.

[11] P.J. Rousseeuw and A.M. Leroy. *Robust Regression & Outlier Detection.* John Wiley & Sons, Inc, 1987.

[12] J. Santos-Victor, G. Sandini, F. Curotto, and S. Garibaldi. Divergent stereo in autonomous navigation : From bees to robots. *Int. Journal of Computer Vision*, 14(2):159–177, 1995.

[13] C. Silva and J. Santos-Victor. Direct egomotion estimation. In *Proc. of the 13th Int. Conference on Pattern Recognition*, Vienna,Austria, August 1996.

[14] C. Silva and J. Santos-Victor. Robust egomotion estimation from the normal flow using search subspaces. Technical Report 6/96, ISR/Inst. Sup. Tecnico - VisLab, 1996.