Visual-Based Obstacle Detection
A purposive approach using the normal flow

JOSÉ SANTOS-VICTOR* **and GIULIO SANDINI****

**Instituto Superior Técnico & Instituto de Sistemas e Robótica, Av. Rovisco Pais 1, 1096 Lisboa Codex - PORTUGAL*

***DIST & Lira-Lab, University of Genova, Via Opera Pia 13, 16145 Genova - ITALIA*

Abstract. *In this paper we present a reflex-type behaviour for visual obstacle detection for mobile robots. The main assumption that is made, is that the robot is moving on a ground plane. For the specific purpose of obstacle detection, it is then possible to design a simplified vision system that can successfully solve the problem in hand. Under this purposive approach, there is no need to perform any reconstruction of the environment and the system shows a robust performance and is running on simple hardware. The method uses the normal flow information as the input. There is no effort made in the sense of recovering both components of the optical flow.*

Key Words. Purposive Vision, Mobile Robots, Normal Flow, Affine Motion

1. INTRODUCTION

Traditional systems for robot navigation tend to perform some kind of map building to determine a safe trajectory avoiding all the obstacles. This approach is quite sensitive and computationally too demanding to be used in real-time for navigation or obstacle avoidance in mobile robotics, particularly when dealing with dynamic environments. Instead, within the framework of *purposive vision* [Aloimonos 90], we can use the specificity of this task, to propose simpler and more robust approaches for vision guided robotics.

The major hypothesis considered in the approach herein proposed, consists in assuming that the robot is moving on a flat ground plane. With this assumption, we propose a system for obstacle detection which is fast, robust and independent of a variety of motion or camera parameters.

A point to be noticed is the use of the normal flow field, computed over an image sequence acquired by a single camera, as the visual stimulus. There is no need, whatsoever, to use extra constraints for the flow field in order to overcome the *aperture problem*, as it is usually done in a variety of other methods.

Conceptually, we try to characterize the apparent motion of the ground plane globally, and detect violations to this coherent motion pattern, which correspond to points lying outside the ground plane. To determine the presence of the obstacles we use inverse projection techniques as suggested in [Mallot *et al.* 91]. However, instead of calibrating the extrinsic and intrinsic parameters of the camera, we use the image measurements directly, to estimate the projective transformation between the image plane and the ground plane. This transformation is used to inverse project the flow field onto the ground plane, highly simplifying the interpretation of the flow pattern.

The motion of the ground floor perceived in the image plane, can be fully described by a second order polynomial in the image coordinates [Subbarao and Waxman 86], which captures the motion of the ground plane as a whole. However, estimating the second-order coefficients of this polynomial leads to highly unstable algorithms [Negahdaripour and Lee 92]. Instead, we approximate the motion of the planar surface as an affine transformation and derive a robust parameter estimation procedure uniquely based on the normal flow information.

In an initialization stage, the affine parameters are used to estimate the projective transformation between the image plane and the ground plane (containing information about the ground plane orientation with respect to the image plane). This transformation is constant since the camera is rigidly attached to the robot. During the obstacle detection phase, the normal flow field is inverse projected onto the horizontal plane, and analysed.

For pure translational motion, the obstacle de-

tection analysis is very simple, as the points on the ground plane should present the same flow vectors, whereas points lying above or below the ground plane, will have respectively larger or smaller flow values. Contrasting with previous approaches [Carlsson and Eklundh 90, Enkelmann 90, Sandini and Tistarelli 90], the method we propose here, does not rely on the knowledge of the camera motion and, for pure translation, it is also independent of the camera intrinsic parameters. A geometric, intuitive explanation is given.

Different experiments were performed to test the various steps of the whole method. Finally, a real-time system was implemented on a robot, to detect obstacles lying on the ground plane.

In the following sections, we describe the problem of motion analysis of a planar patch. Then, the affine model approximation for the motion field is introduced, and we present the estimation procedure to recover the model parameters uniquely based on the normal flow. We then describe the inverse projection method, and show how to recover the needed image plane/ground plane transformation using the affine model parameters. Finally, a variety of tests are presented, and conclusions drawn.

2. PLANAR SURFACES IN MOTION

Many authors have addressed the problem of mission parameters estimation from global flow data, [Nelson and Aloimonos 88]. Also, our first concern is the characterization of the flow field perceived in the image plane. Particularly, in our approach, it is assumed that the robot is moving on a flat ground and the camera is facing the ground plane, hence observing a planar surface in motion. With this assumption, there is a globally valid parameterization for the corresponding optical flow field. These parameters can be robustly estimated based on the normal flow field, as it is shown later in this paper.

Let us assume a rigid body motion model for the robot, with linear velocity $\mathbf{T} = [T_x \;\; T_y \;\; T_z]^T$ and angular velocity $\boldsymbol{\omega} = [\omega_x \;\; \omega_y \;\; \omega_z]^T$. The perspective projection (pinhole) of a 3D point, (X, Y, Z), into the image plane is given by :

$$\left\{ \begin{array}{rcl} x & = & f_x \frac{X}{Z} + c_x \\ y & = & f_y \frac{Y}{Z} + c_y \end{array} \right. \quad (1)$$

where, f_x, f_y denote the camera focal length expressed in pixels and c_x, c_y denote the image center coordinates, and x, y are given in pixels. The motion perceived in the image plane, is then given by the well known equations [Subbarao and Waxman 86, Sundareswaran 91] :

$$u(x,y) = \frac{xT_z - T_x}{Z(x,y)} + \omega_x xy - \omega_y(1+x^2) + \omega_z y \quad (2)$$

$$v(x,y) = \frac{yT_z - T_y}{Z(x,y)} + \omega_x(1+y^2) - \omega_y xy - \omega_z x \quad (3)$$

where u and v are the x and y components of the flow field.

As the camera is pointing at the ground plane, the observed surface is described by :

$$Z(X,Y) = Z_0 + \gamma_x X + \gamma_y Y \quad (4)$$

where γ_x, γ_y are the surface slopes along the horizontal and vertical directions (slant and tilt) and Z_0 is the distance measured along the optical axis. Using the perspective equations, we can describe the ground plane surface as a function of the image pixel coordinates, instead of the 3D coordinates :

$$Z(x,y) = \frac{Z_0}{1 - \gamma_x \frac{x}{f_x} - \gamma_y \frac{y}{f_y}} \quad (5)$$

Finally, using equation (5) together with equations (2,3), we obtain the quadratic equations describing the flow of a planar surface in motion [Subbarao and Waxman 86].

$$u(x,y) = u_0 + u_x x + u_y y + u_{xy} xy + u_{xx} x^2 \quad (6)$$

$$v(x,y) = v_0 + v_x x + v_y y + v_{xy} xy + v_{yy} y^2 \quad (7)$$

which is globally valid.

A straightforward approach consists in directly estimating the 8 parameters of the flow model. However, it has been shown analytically and experimentally in [Negahdaripour and Lee 92], that the estimates of the second-order parameters are often affected by noise which is higher, up to several orders of magnitude, than the first order parameters estimates, even for perfect planar motion. If the angle of view is small, and the depth range restricted [Koenderink and van Doorn 91], the affine model is a good approximation of the motion field. The modeling error (particularly at the image periphery) is still usually smaller than the estimation error of the full second order model. Therefore, to improve robustness, we use an affine motion model :

$$\begin{array}{rcl} u(x,y) & = & u_0 + u_x x + u_y y \\ v(x,y) & = & v_0 + v_x x + v_y y \end{array} \quad (8)$$

Affine motion parameters estimation using the normal flow

We have established a model that captures the optical flow of a planar surface in motion with

respect to a camera. The problem now is the estimation of the affine flow parameters. Particularly, we would like to constrain ourselves to the use of just normal flow information. These parameters are used to recover the ground plane orientation, needed for the obstacle detection mechanism. The same parameters can be used to control the docking manoeuvre of a mobile robot, as described in [Santos-Victor and Sandini 94].

The normal flow is the only flow component that can be reliably estimated [Fermüller 93, Horn and Shunck 81], due to the well known *aperture problem*, without having to assume further constraints on the motion field (such as smoothness). The optical flow constraint equation, assuming image brightness constancy over time [Horn and Shunck 81], is given by :

$$uI_x + vI_y = -I_t, \tag{9}$$

where I_x, I_y and I_t stand for the partial derivatives of the image with respect to x, y, and time. According to the affine model equations (8), the first term of equation (9) can be written as :

$$v_o I_y + v_x I_y x + v_y I_y y + u_o I_x + u_x I_x x + u_y I_x y = -I_t \tag{10}$$

Therefore, the temporal derivative over the planar patch can be expressed as a linear combination of the parameters to estimate:

$$\left[\begin{array}{cccccc} I_y & xI_y & yI_y & I_x & xI_x & yI_x \end{array}\right] \boldsymbol{\theta} = -I_t \tag{11}$$

where $\boldsymbol{\theta}$ is given by:

$$\boldsymbol{\theta} = [v_0 \;\; v_x \;\; v_y \;\; u_0 \;\; u_x \;\; u_y]^T \tag{12}$$

To estimate $\boldsymbol{\theta}$, it is sufficient to use 6 measurements of spatial-temporal image derivatives or, equivalently 6 measurements of the normal flow. The least squares estimate to this problem can be obtained by considering an over-determined system of equations. Let n be the number of measurements available, and define :

$$\mathcal{D} = [-I_{t1} \;\; -I_{t2} \;\; \ldots \;\; -I_{t_n}]^T \tag{13}$$

and let

$$\mathcal{M} = \left[\begin{array}{cccccc} I_{y1} & x_1 I_{y1} & y_1 I_{y1} & I_{x1} & x_1 I_{x1} & y_1 I_{x1} \\ I_{y2} & x_2 I_{y2} & y_2 I_{y2} & I_{x2} & x_2 I_{x2} & y_2 I_{x2} \\ & \vdots & & & \vdots & \\ I_{yn} & x_n I_{yn} & y_n I_{yn} & I_{xn} & x_n I_{xn} & y_n I_{xn} \end{array}\right] \tag{14}$$

The least squares estimate for this problem is given by the pseudo-inverse solution :

$$\hat{\boldsymbol{\theta}} = \left(\mathcal{M}^T \mathcal{M}\right)^{-1} \mathcal{M}^T \mathcal{D} \tag{15}$$

The direct application of the least squares estimation procedure is usually quite sensitive to outliers, thus leading to an ungraceful degradation of the estimates. To overcome this problem, we devised a recursive estimation procedure, to eliminate the effect of outliers. The algorithm works as follows :

1. Choose randomly a set of data points, $\{I_x,\ I_y,\ I_t\}$, to get an initial estimate, $\boldsymbol{\theta}_o$. Set $k = 1$.
2. Choose randomly a new set of data points, $\{I_x,\ I_y,\ I_t\}$, such that the modeling error in equation (11) is small, when evaluated with the current parameter estimate.
3. Estimate $\boldsymbol{\theta}_k$ based on the new data set. Set $k = k + 1$.
4. Return to step (2) until $\boldsymbol{\theta}$ remains unchanged or k exceeds a given number of iterations.

The rationale is that an approximate initial estimate, can be improved in subsequent steps, by only selecting the data coherent with the model. As a result, outliers are discarded.

A wide variety of tests were performed in order to evaluate the robustness of the estimation procedure. Figure 1 shows an image sequence acquired by a camera moving towards a slanted poster. Figure 2 shows an example of the optical flow field

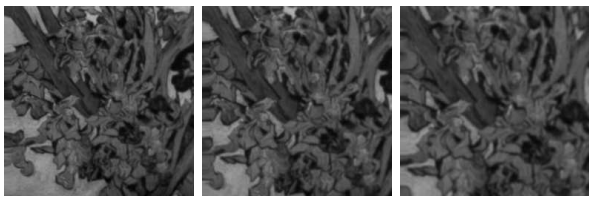

Figure 1 *Image sequence, acquired by moving a camera towards a slanted poster.*

computed based on this sequence. These data were used to estimate the affine model parameters, which were used to generate a "reconstructed" flow field shown, for comparison, in Fig. 2. As shown, the estimation procedure succeeds in discarding the outliers and the "reconstructed" optical flow is, at least from a qualitative point of view, consistent with the input data. Notice, for instance, the location of the Focus of Expansion in both cases.

Plane coefficients estimation - the intrinsic parameters

We have presented a procedure to estimate the affine motion model parameters. Once $\boldsymbol{\theta}$ has been

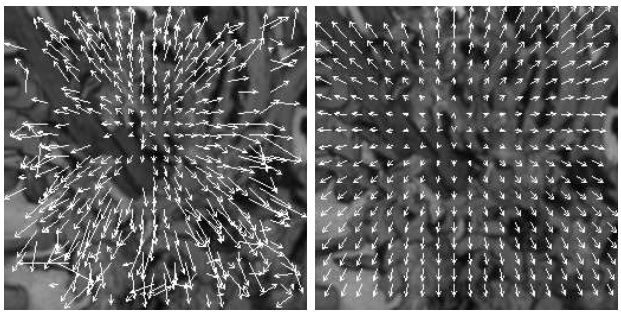

Figure 2 *Left: poster sequence optical flow. Right: reconstructed flow field based on the estimated parameters.*

estimated, one can determine γ_x and γ_y. However, these parameters can only be estimated up to a scale factor, which is the focal length expressed in pixel coordinates :

$$\begin{cases} \frac{\gamma_x}{f_x} = -\frac{v_x}{v_0} \\ \frac{\gamma_y}{f_y} = \begin{cases} -\frac{u_y}{u_0}, & \text{if } u_0 \neq 0 \\ \frac{u_x - v_y}{v_0}, & \text{otherwise} \end{cases} \end{cases} \quad (16)$$

Consequently, if the camera intrinsic parameters are known, the $\gamma_{x,y}$ coefficients can be obtained directly. However, even in the absence of the camera parameters, f_x and f_y, it is still possible to proceed with the method. If the mobile robot motion is assumed to be purely translational, the affine model parameters can be simplified to :

$$\begin{array}{llll} u_0 = -\frac{f_x T_x}{Z_0} & & v_0 = -\frac{f_y T_y}{Z_0} \\ u_x = \frac{T_z}{Z_0} + \frac{\gamma_x}{f_x}\frac{f x T_x}{Z_0} & & v_x = \frac{f_y T_y}{Z_0}\frac{\gamma_x}{f_x} \\ u_y = \frac{f_x T_x}{Z_0}\frac{\gamma_y}{f_y} & & v_y = \frac{T_z}{Z_0} + \frac{\gamma_y}{f_y}\frac{f_y T_y}{Z_0} \end{array} \quad (17)$$

These equations have been rewritten to emphasize the ambiguity between $\{f_x,\ f_y\}$ and $\{\gamma_x,\ \gamma_y,\ T_x,\ T_y\}$. If we scale up f_x or f_y by a given factor, ξ, divide the corresponding T_x or T_y by ξ and multiply the corresponding $\{\gamma_x,\ \gamma_y\}$ by ξ again, then all the flow parameters remain unchanged. Therefore, if a camera with different intrinsic parameters is used, it is possible to find a different orientation and velocity (suitably scaling $\gamma_x,\ \gamma_y,\ T_x,\ T_y$) such that the observed flow is exactly the same.

As we are not interested in the absolute camera speed or orientation, we can suppose that a *virtual* camera with unitary f_x and f_y, is being used. The simplified parameterization then follows :

$$\begin{array}{llll} u_0 = -\frac{T_x}{Z_0} & & v_0 = -\frac{T_y}{Z_0} \\ u_x = \frac{T_z}{Z_0} + \gamma_x\frac{T_x}{Z_0} & & v_x = \frac{T_y}{Z_0}\gamma_x \\ u_y = \frac{T_x}{Z_0}\gamma_y & & v_y = \frac{T_z}{Z_0} + \gamma_y\frac{T_y}{Z_0} \end{array} \quad (18)$$

where one should keep in mind that T_x, T_y, γ_x and γ_y do no longer convey information neither on the absolute orientation of the ground plane, nor on the absolute camera speed. They correspond to the speed and orientation, with respect to the ground floor, that a camera with unitary intrinsic parameters, f_x and f_y, should have in order to perceive the same flow field. Finally, γ_x and γ_y are simply determined by using equation (16) with unitary f_x, f_y.

In order to evaluate the performance of the estimation procedure to recover the plane orientation, we have generated a synthetic flow pattern corresponding to a robot moving forwards with a speed of 25cm/s equipped with a camera pointing down at 45^o : $\gamma_x = 0, \quad \gamma_y = 1$.

The camera intrinsic parameters are those of a standard CCD (6.4mm by 4.8mm) sensor with a 4.8 mm lens and 256×256 image resolution :

$$f_x = 192, \quad f_y = 256, \quad c_x = 128, \quad c_y = 128$$

The flow is computed according to these camera settings and motion and corrupted with zero-mean Gaussian noise (added independently to each component). Figure 3 shows the synthetic flow field in the absence of noise.

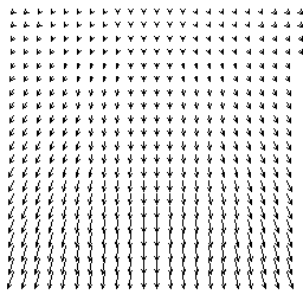

Figure 3 *Synthetic flow field corresponding to a robot moving forward on a flat ground, with a camera pointing down.*

Table 1 shows the slant and tilt angles estimates with the synthetic flow data with increasing noise levels. The estimates of the plane orientation are fairly robust and accurate even in the presence of significant levels of noise.

Noise σ	# points	slant (deg.)	tilt (deg.)
0.0	200	0.0	-45.0
1.0	200	-0.83	-46.97
1.5	200	-0.30	-42.79
2.0	200	-0.20	-46.63

Table 1 *Tilt and Slant estimates for different values of the noise standard deviation. The flow fields are shown in Fig. 5.*

3. INVERSE PERSPECTIVE FLOW TRANSFORMATION

The method we propose for ground plane obstacle detection, is based on the inverse perspective mapping of the flow field. The main idea is that by re-projecting the flow onto the horizontal plane, the analysis is much simplified. The resultant obstacle detection algorithm is very simple for pure translational motion and can also cope with rotation by fitting simple models to the transformed flow.

Similarly to what has been proposed in [Mallot *et al.* 91], the rationale is that if it is possible to inverse project the flow field perceived on the image plane, (π_C), onto the horizontal plane, (π_H), then the camera translation becomes parallel to the ground floor and the rotation is solely around the vertical axis, which greatly simplifies the flow pattern as illustrated in Fig. 4. The coordinate systems (C) and (H) share the same origin even though they have been drawn separately for the sake of clarity.

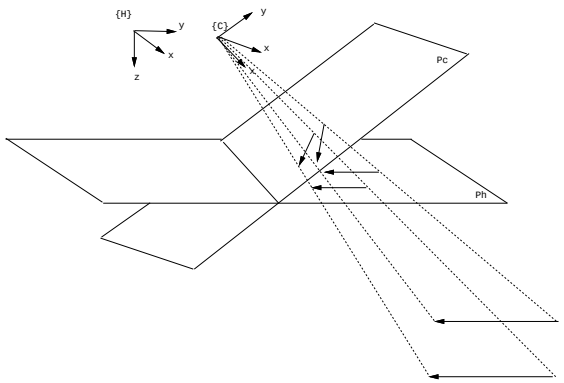

Figure 4 *Inverse perspective mapping. While on (C) the motion of the ground floor is perceived as a complex vectorial pattern, in (H) all the vectors have equal length and orientation under translational motion.*

In all the subsequent analysis, we will include the camera intrinsic parameters, which can be considered unitary and discarded, in the case of translational motion, as discussed previously. Let

$$C = \{X_c,\ Y_c,\ Z_c\ \}$$
$$H = \{X_h,\ Y_h,\ Z_h\ \} \tag{19}$$

be the coordinate frames associated to the camera and to the horizontal plane. As both frames share a common origin, the coordinate transformation relating both systems is just a rotation term, ${}^{H}\mathbf{R}_{C}$:

$${}^{H}\mathbf{P} = {}^{H}\mathbf{R}_{C}\ {}^{C}\mathbf{P} \tag{20}$$

where ${}^{C}\mathbf{P}$ is a point in the 3D space expressed in the camera coordinates, and ${}^{H}\mathbf{R}_{C}$ results from rotating a tilt angle, ψ, around the camera x axis and a pan angle, ϕ, around the camera y axis (corresponding to a camera pointing down in front of a mobile robot). The rotation matrix will have the following structure (in most practical systems the pan angle, ψ, is small and can even be neglected) :

$${}^{H}\mathbf{R}_{C} = \begin{bmatrix} \cos\phi & -\sin\phi\sin\psi & -\sin\phi\cos\psi \\ 0 & \cos\psi & -\sin\psi \\ \sin\phi & \cos\phi\sin\psi & \cos\phi\cos\psi \end{bmatrix} \tag{21}$$

Let the perspective projection of a point in a plane β, be defined :

$${}^{\beta}\mathbf{P}' = \mathcal{P}_{\beta}({}^{\beta}\mathbf{P}) \tag{22}$$

$$\begin{bmatrix} s\ x'_{\beta} \\ s\ y'_{\beta} \\ s \end{bmatrix} = \begin{bmatrix} f_x X_{\beta} \\ f_y Y_{\beta} \\ Z_{\beta} \end{bmatrix} \tag{23}$$

where $\mathcal{P}_{\beta}$ denotes the projection operator, and x', y' are image points expressed in pixel coordinates. The set of points in the 3D space that project on a given image pixel (x'_c, y'_c) is given by :

$${}^{C}\tilde{\mathbf{P}} = [\lambda\frac{x'_c}{f_x} \quad \lambda\frac{y'_c}{f_y} \quad \lambda]^T \tag{24}$$

which describes a beam passing through the projection center and the projection point in the image plane. We can now express $\tilde{P}$ in the frame attached to the horizontal plane :

$${}^{H}\tilde{\mathbf{P}} = {}^{H}\mathbf{R}_{C}\ {}^{C}\tilde{\mathbf{P}} \tag{25}$$

Finally, this point can be projected into the horizontal plane, π_H, combining equations (21) to (25) :

$$\begin{bmatrix} x'_H \\ \\ y'_H \end{bmatrix} = {}^{H}\mathcal{P}_{C}(x'_c, y'_c) \tag{26}$$

where ${}^{H}\mathcal{P}_{C}(x'_c, y'_c)$ denotes the operator projecting from the image plane to the horizontal plane. This plane-to-plane projective transformation can be rewritten as :

$$\begin{bmatrix} s\ x'_H \\ s\ y'_H \\ s \end{bmatrix} = {}^{H}\mathbf{R}_{C} \begin{bmatrix} x'_c/f_x \\ y'_c/f_y \\ 1 \end{bmatrix} \tag{27}$$

To obtain the inverse projection of a flow vector, we have to calculate the time derivatives of (x'_H, y'_H) :

$$\begin{bmatrix} u_H(x'_c, y'_c) \\ \\ v_H(x'_c, y'_c) \end{bmatrix} = \frac{\partial {}^H p_C(x'_c, y'_c)}{\partial x'_c} u + \frac{\partial {}^H p_C(x'_c, y'_c)}{\partial y'_c} v \tag{28}$$

which can be written in homogeneous coordinates as :

$$\begin{bmatrix} s\, u_H(x'_c, y'_c)/f_x \\ s\, v_H(x'_c, y'_c)/f_y \\ s \end{bmatrix} = \begin{bmatrix} (\cos\psi + \sin\psi \frac{y'_c}{f_y})\frac{u}{f_x} - \sin\psi \frac{x'_c}{f_x}\frac{v}{f_y} \\ \\ \sin\phi(\sin\psi - \cos\psi \frac{y'_c}{f_y})\frac{u}{f_x} + (\cos\psi \sin\phi \frac{x'_c}{f_x} + \cos\phi)\frac{v}{f_y} \\ \\ (\sin\phi \frac{x'_c}{f_x} + \cos\phi \sin\psi \frac{y'_c}{f_y} + \cos\phi \cos\psi)^2 \end{bmatrix} \tag{29}$$

Remains to be considered the estimation of ψ and ϕ using the affine model parameters.

Recovering the Slant and Tilt parameters

During the initialization phase, we must determine the rotation matrix, preferably without explicitly calibrating the system. The process simply consists in determining γ_x, γ_y as explained in the previous sections and assuming that both f_x and f_y are unitary (for the case of pure translation).

Since any point in the camera coordinate system can be expressed, according to equation (20), in the horizontal plane coordinate system, the Z component is given by :

$$Z_H = \sin\phi X + \cos\phi \sin\psi Y + \cos\phi \cos\psi Z \tag{30}$$

However, in camera coordinates, Z is a function of X and Y according to the ground plane constraint, given in equation (4). Combining these two equations, and knowing that, in the coordinate system of the horizontal plane, all the points in the ground floor have a constant depth, Z_H, we get:

$$\begin{array}{rcl} \psi & = & -\arctan\gamma_y \\ \phi & = & -\arctan(\gamma_x \cos\psi) \\ Z_H & = & \cos\psi \cos\phi\ Z_0 \end{array} \tag{31}$$

To summarize, once γ_x and γ_y have been estimated, the inverse perspective transformation is applied, to inverse project the optical flow onto the horizontal plane. The values of ψ and ϕ are estimated only once during an initialization phase and remain constant provided that the camera is in a fixed position relative to the robot.

Obstacle Detection

In this section we will analyse the obstacle detection mechanism. Using the horizontal plane coordinate frame, the linear velocity is constrained to be parallel to the "*new image*" plane (hence $T_z = 0$), the angular velocity component is solely around the vertical axis and the distance to the ground floor is constant. Therefore, the optical flow equations are given by :

$$\begin{aligned} u_H(x_H, y_H) &= \left(-\frac{T_x}{Z_H} + \omega_z y_H\right) \\ v_H(x_H, y_H) &= \left(-\frac{T_y}{Z_H} - \omega_z x_H\right) \end{aligned} \tag{32}$$

In many situations, it is reasonable to assume that the rotation component is neglectable when compared to the translational component. In other cases, we can avoid computing the flow during fast rotations, which correspond to a saccadic suppression mechanism observed in many biological vision systems.

For pure translational motion, the transformed flow vectors are constant for every point on the ground plane and a simple test can be performed to check if there are any obstacles above or below the ground plane. The detection mechanism simply relies on the fact that the optical flow should be globally constant no matter what the motion direction and speed might be. The normal flow vectors can be projected onto the direction of motion, which is constant all over the image.

$$\begin{aligned} u_H(x, y) &= -\frac{T_x}{Z_H} \\ v_H(x, y) &= -\frac{T_y}{Z_H} \end{aligned} \tag{33}$$

In the case of general motion, the main difficulty consists in separating the rotation component from the purely translational component [Hummel and Sundareswaran 93].

In our simplified reference frame, we can simply apply an estimation procedure to the inverse projected flow in order to recover the translational (constant all over the image) and rotational components (which depend on the x or y coordinates). The same kind of estimation techniques, used to estimate the affine flow parameters, can again be used to separate the rotation and translation components of motion. Once this has been done, the translational component can be checked, as in the case of purely translational motion.

4. RESULTS

We present now some results of the proposed method. Initial tests used the synthetic data, while the final tests were based on a mobile vehicle.

The synthetic flow fields, used for testing, were corrupted with increasing levels of noise, as shown in Fig. 5. The intrinsic parameters are assumed to be unknown.

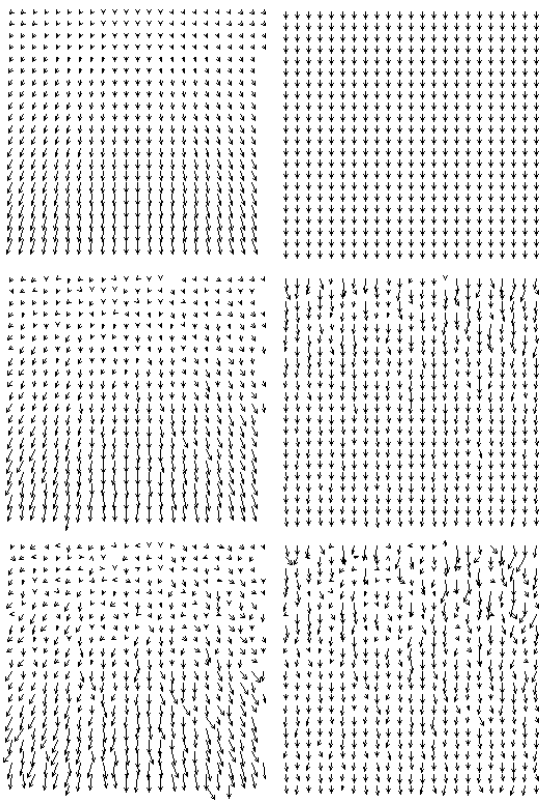

Figure 5 *Left : synthetic flow fields corrupted by noise with* $0.0, 1.0$ *and* 2.0 *pixels/frame of standard deviation. Right : inverse projected flow fields.*

The left column shows the synthetic flow, degraded with noise of different intensities. The right column shows the corresponding inverse projected flow.

We can see that even for severely corrupted data, the flow becomes relatively constant after the inverse projection. The degradation is more noticeable in the image areas corresponding to the far part of the visual field, where the flow amplitude is small and the signal to noise ratio is poor. The method is considerably more accurate at short range, instead.

The system has been tested in real time on a mobile robot, moving around in a room and stopping whenever an obstacle was detected. The experimental setup is composed of a camera with a 8mm lens, installed on a TRC Labmate mobile platform. The camera was placed in the front part of the robot facing the ground plane with an angle of about 65 degrees.

In all the experiments, the robot speed was set to 10cm/s. The images resolution is 128×128 pixels and a central window of 80×80 pixels is used for the normal flow computation and the obstacle detection process. This resolution is a tradeoff between the necessary detail for the flow computation and analysis, and the computing speed. The system is running at a frequency of about $1 H_z$ on a VDS Eidobrain image processing workstation.

Figure 6 shows the normal flow field measured while the robot is moving, in the absence of obstacles. Notice how the flow magnitude increases from the top to the bottom. This flow is used in the initialization stage to estimate the ground plane parameters. The figure shows also the result of inverse projecting the flow field. The resulting flow is approximately constant over the whole image.

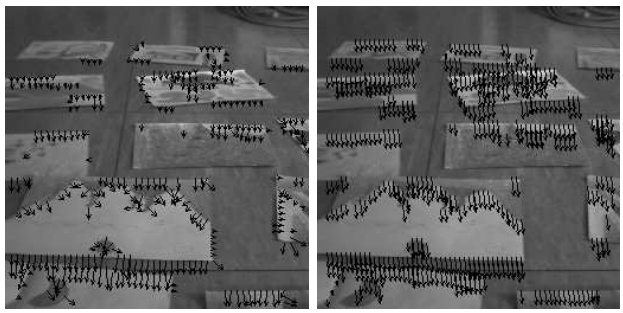

Figure 6 *Left: ground plane normal flow. Right: inverse projected flow field.*

Several tests were performed with different obstacles placed on the ground floor. An example is shown in Fig. 7. On the left we have the normal flow observed during the trajectory, with an obstacle in the visual field. On the right side we show the result of the inverse projection mechanism, where the obstacle flow becomes clearly larger than the ground flow.

The result of the detection process is shown in Fig. 8. The white areas correspond to obstacle points, while the dark areas represent free space. The obstacle has been detected, particularly at the object top regions, as a result of the sensitivity constraints imposed by the image resolution. The object contours are reasonably well defined and an acceptable segmentation could be achieved. However, as the main goal is to provide a simple, fast and robust reflex-type behaviour for obstacle de-

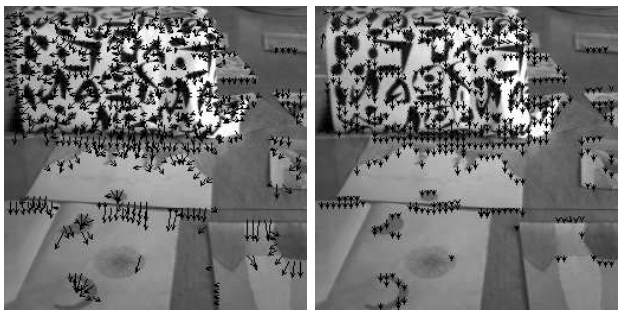

Figure 7 *Left: normal flow field. Right: resulting inverse projected flow.*

Figure 8 *Points detected as obstacles above the ground plane.*

tection, a precise reconstruction of the obstacle position is not pursued. In all the tests performed, the system behaved robustly, detecting various obstacles and safely stopping the robot. Several simple strategies can be used in order to circumvent the obstacles detected. The robot can either rotate a given angle away from the image side where the object is located, or we can determine the amount of rotation needed to remove the obstacle from the robot visual field.

5. CONCLUSIONS

We have described a method for fast obstacle detection for mobile robots. The basic assumption is that the robot is moving on a ground floor and any object not lying on this plane is considered an obstacle. The method is based on the inverse projection of the normal flow field onto the ground plane. For pure translational motion, the flow vectors become constant all over the image with the obstacles having a larger flow than every other point lying on the pavement.

The flow induced by the motion of the ground plane, is described by an affine model. The model parameter estimation procedure relies exclusively on the first order space and time derivatives, or equivalently, on the normal flow. Based on the affine model parameters, the tilt and slant angles of the image plane with respect to the ground are estimated and used for the inverse projection operation. No explicit calibration of the camera intrinsic or extrinsic parameters is needed.

In the absence of rotational motion, the method is independent of the camera intrinsic parameters. It is not necessary to know the robot motion, as the detection strategy is based on the constancy of the flow vectors. The robustness of the approach was illustrated by several tests, in real time, with a mobile robot.

REFERENCES

[Aloimonos 90] Aloimonos Y., Purposive and qualitative active vision, in: *Proc. of the 10th. IEEE ICPR*, Atlantic City, NJ - USA (June 1990).

[Carlsson and Eklundh 90] Carlsson S. and Eklundh J., Obstacle detection using model based prediction and motion parallax, in: *Proc. of the 1st. ECCV*, Antibes - France (April 1990).

[Enkelmann 90] Enkelmann W., Obstacle detection by evaluation of optical flow field from image sequences, in: *Proc. of the 1st. ECCV*, Antibes (France) (Springer Verlag, 1990) 134–138.

[Fermüller 93] Fermüller C., Navigational preliminaries, in: Aloimonos Y. (Ed.), *Active Perception* (Lawrence Erlbaum Associates, 1993).

[Horn and Shunck 81] Horn B. and Shunck B., Determining optical flow, *Artificial Intelligence* **17** (1981) 185–203.

[Hummel and Sundareswaran 93] Hummel R. and Sundareswaran V., Motion parameter estimation from global flow field data, *IEEE Trans. on PAMI* **15** (5) (1993) 459–476.

[Koenderink and van Doorn 91] Koenderink J. and van Doorn J., Affine structure from motion, *Journal of the Optical Society of America* **8** (2) (1991) 377–385.

[Mallot *et al.* 91] Mallot H., Bulthoff H., Little J. and Bohrer S., Inverse perspective mapping simplifies opticla flow computation and obstacle detection., *Biological Cybernetics* **64** (1991) 177–185.

[Negahdaripour and Lee 92] Negahdaripour S. and Lee S., Motion recovery from images sequences using only first order optical flow information., *Intl. Journal of Computer Vision* **9** (3) (1992) 163–184.

[Nelson and Aloimonos 88] Nelson R. and Aloimonos J., Finding motion parameters from spherical motion fields (or the advantage of having eyes in the back of your head), *Biological Cybernetics* **58** (1988).

[Sandini and Tistarelli 90] Sandini G. and Tistarelli M., Robust obstacle detection using optical flow, in: *Proc. of the IEEE Intl. Workshop on Robust Computer Vision*, Seattle, (WA) (October 1990).

[Santos-Victor and Sandini 94] Santos-Victor J. and Sandini G., Visual behaviors for docking., *submitted to CVGIP : IU* (1994).

[Subbarao and Waxman 86] Subbarao M. and Waxman A., Closed form solutions to image flow equations for planar surfaces in motion, *CVGIP* **36** (1986) 208–228.

[Sundareswaran 91] Sundareswaran V., Egomotion from global flow field data, in: *Proc. of the IEEE Workshop on Visual Motion*, Princeton, New Jersey (October 1991).