Machine Vision and Applications Manuscript-Nr.
(will be inserted by hand later)

Uncalibrated Obstacle Detection using Normal Flow

José Santos-Victor[1] and Giulio Sandini[2]

[1] Instituto Superior Técnico, Instituto de Sistemas e Robótica
Av. Rovisco Pais, 1, 1096 Lisboa Codex - Portugal
e-mail: jasv@isr.ist.utl.pt

[2] DIST, Lira-Lab, University of Genova
Via Opera Pia, 13, 16145 Genova - Italia
e-mail:giulio@vision.dist.unige.it

Received June, 1994

Abstract. This paper addresses the problem of obstacle detection for mobile robots, using visual information provided by a single on-board camera, as the input.

The method proposed, assumes that the robot is moving on a planar pavement, and any point lying outside this plane is considered to be an obstacle. The problem of obstacle detection is addressed by exploiting the geometric arrangement between the robot, camera and scene. During an initialization stage, we estimate an inverse perspective transformation, that maps the image plane onto the horizontal plane. During normal operation, the normal flow is computed and inverse projected on the horizontal plane. The resultant flow pattern is much simpler, and fast tests can be used to detect obstacles.

A salient feature of the method proposed is the fact that only the normal flow information, or first order time and space image derivatives, is used, thus coping with the aperture problem. Another important issue is that, contrasting with other methods, the vehicle motion and camera intrinsic and extrinsic parameters are not required to be known or calibrated. Both translational and rotational motion can be dealt with.

Motion estimation results are presented, on synthetic and real image data. A real-time version implemented on a mobile robot, is described.

Key words: Mobile Robots – Robot Vision – Obstacle Detection – Normal Flow – Affine Motion

1 Introduction

The ability to detect and avoid obstacles is an essential capability for any mobile robot to operate in an unknown environment. In many existing systems, this goal has been accomplished to some extent by the use of ultrasound sensors. Even though the use of vision could provide much higher resolution when compared to sonars, some vision based solutions require much more computational power and the robustness achieved, in many systems, is still inadequate.

The problem addressed in this paper is a vision-based obstacle detection for an indoors mobile robot, equipped with a single camera. The system is robust and suitable for real time implementation.

In the earliest approaches, vision was used to reconstruct the 3D space in front of a vehicle and the reconstructed model would then be used to detect obstacles or the free areas in the scene. Often, this approach failed to be robust and required a unrealistic computing power.

However, this problem can be quite simplified by considering the sole purpose of detecting obstacles without, for instance, recovering their exact position or shape. This so-called purposive vision approach [1], that trades the generality of the reconstructive approaches by the specificity of the particular problem to be solved, has led to successful systems both for robot navigation [23, 24] and obstacle detection [3, 6, 22].

Most obstacle detection systems, however, required the vehicle motion and camera parameter to be known a priori, or the vehicle speed to remain constant, which are very restrictive hypotheses. Instead, the system we propose here, exploits the geometric properties of the vehicle-camera-scene arrangement, and does not depend on the knowledge of the camera parameters or vehicle motion.

The main idea is the analysis of the particular structure of the optical flow field, when a robot with a camera is moving over a ground plane. In Figure 1, Pc is the image plane of a camera moving forwards with pure translation, while facing a planar pavement. Even in this simple arrangement, the resulting flow pattern is complex. This is due to the perspective effects and makes the problem of motion analysis or obstacle detection difficult.

Suppose now, that the camera optical axis is vertical, such that the camera is pointing downwards and the image plane, Ph, is parallel to the ground plane. In this case, the flow pattern becomes much simpler as the distance to any point on the pavement is constant. All the

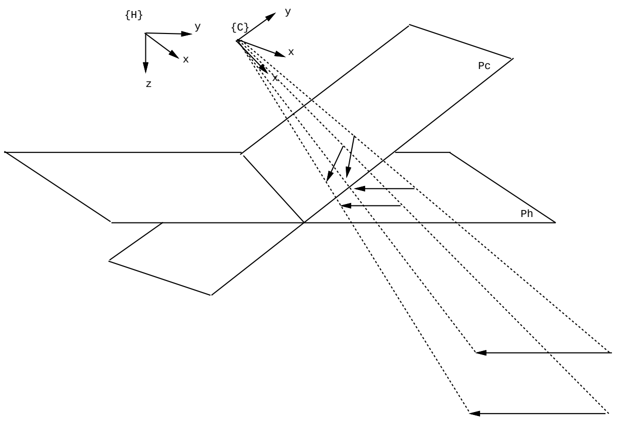

Fig. 1. Inverse perspective mapping. The coordinates systems (C) and (H) share the same origin even though in the picture they have been designed separately for simplification. While on (C) the motion of the ground floor is perceived as a complex vectorial pattern, in (H) all the vectors have equal length and orientation under translational motion.

vectors are equal, and points lying above or below the ground plane can be easily detected.

Orienting the optical axis vertically is not useful for practical reasons, but we can inverse project the optical flow in Pc onto the horizontal plane, Ph, as suggested in [17, 29], thus simplifying the original problem.

In [17, 29], the camera intrinsic parameters (focal length, image center, pixel dimension) and extrinsic (camera orientation relative to the ground plane) had to be measured. Instead, we use directly the image measurements to estimate the projective transformation between the image plane and the ground plane. The estimation procedure does not require the calibration of the camera parameters. The main constraints for the system to work are the following :

- The robot must be moving on a planar pavement.
- During the initialization phase, when the projective transformation is estimated, the vehicle motion must be a pure translation.

The problem of motion analysis has attracted a great deal of attention in the computer vision community [10, 12, 16, 18, 19, 20]. It is well understood that, by using local measurements alone, it is only possible to recover the component of the optical flow along the image gradient direction (the normal flow), while the other component remains unknown [12]. On the other hand, the normal flow, can be estimated robustly and conveys sufficient information for many tasks, if used properly [2, 13, 25]. An important aspect of the method proposed here, is the exclusive use of the normal flow information or, which is equivalent, first order time and space image derivatives.

Using the ground plane constraint, the motion of the ground floor perceived in the image plane can be described by a second order polynomial in the image coordinates. We approximate this model by an affine transform, which leads to improved robustness in the estimation procedure. The projective transformation is computed while moving the vehicle in a pure translation.

This projective transformation is constant in time, since the camera is rigidly attached to the robot. During normal operation, the normal flow field is inverse projected onto the horizontal plane, where obstacles are easily detected. The more salient features of the method we propose, are :

- Exclusive use of the normal flow information.
- Knowledge about the vehicle velocity, or camera parameters is not required.
- Apart from the initialization phase, the system can deal both with rotational and translational motion, of the robot.
- Appropriate for use in real-time.

In Section 2, we look at the problem of planar surfaces in motion and the resultant optical flow field. We will present some results on the estimation of the affine flow parameters, using the normal flow, and show how to recover information about the plane orientation. We also discuss the role of the camera intrinsic parameters and explain how, in some circumstances, the system does not require these parameters to be known.

In Section 3, we give all the details regarding the inverse projection process and show how the projective transformation can be defined, using the information of the plane orientation. This section also describes the procedure to detect points lying out of the ground plane, which are classified as obstacles.

In Section 4, we present results, both for the estimation procedure with synthetic and real image data, and for a real-time system, implemented on a robot. Finally, in Section 5 we emphasize the main aspects and contributions of this paper and draw some conclusions.

2 Planar surfaces in motion

Under the hypothesis that the ground floor is flat, the camera observes a planar surface in motion, as the robot moves in the workspace. The optical flow of a planar surface in motion will be analyzed in this section.

Assume that the camera moves with linear velocity $\mathbf{T} = [T_x \;\; T_y \;\; T_z]^T$, and angular velocity $\boldsymbol{\omega} = [\omega_x \;\; \omega_y \;\; \omega_z]^T$. Under perspective projection, the image flow is given by the well known equations [4, 5, 26, 27] :

$$u = f_x \left[\frac{\frac{x}{f_x} T_z - T_x}{Z(x,y)} + \omega_x \frac{xy}{f_x f_y} - \omega_y (1 + \frac{x^2}{f_x^2}) + \omega_z \frac{y}{f_y} \right]$$

$$v = f_y \left[\frac{\frac{y}{f_y} T_z - T_y}{Z(x,y)} + \omega_x (1 + \frac{y^2}{f_y^2}) - \omega_y \frac{xy}{f_x f_y} - \omega_z \frac{x}{f_x} \right] \quad (1)$$

where u and v are the x and y components of the flow field; f_x, f_y denote the camera focal length expressed in pixels (focal length divided by the horizontal and vertical pixel dimensions). It is assumed that pixel $(0,0)$ is at the image geometric center. The projection coordinates x, y are given in pixels. Using the ground plane constraint, we can express Z as :

$$Z(X,Y) = Z_0 + \gamma_x X + \gamma_y Y \quad (2)$$

where γ_x, γ_y quantify the plane slant and tilt slopes. By introducing the perspective equations, the ground plane surface can be described as a function of the image pixel coordinates :

$$Z(x, y) = \frac{Z_0}{1 - \gamma_x \frac{x}{f_x} - \gamma_y \frac{y}{f_y}} \tag{3}$$

Finally, using equation (3) together with equations (1), we obtain the quadratic equations that describe the optical flow of a planar surface in motion [21, 26], which are valid all over the image.

$$\begin{aligned} u(x, y) &= u_0 + u_x x + u_y y + u_{xy} xy + u_{xx} x^2 \\ v(x, y) &= v_0 + v_x x + v_y y + v_{xy} xy + v_{yy} y^2 \end{aligned} \tag{4}$$

It has been shown in [21], that the estimates of the second-order model parameters are often affected by noise which is up to several orders of magnitude higher than the first order parameters estimates, even in the case of perfect planar motion. However, if the angle of view is small and the depth of range limited, the second order parameters can be discarded and the optical flow field of the planar surface approximated by an affine transformation. The modeling error (higher at the image periphery) is still smaller than the estimation error of the full second order model. Then, to improve robustness, it is preferable to approximate the motion field by an affine model [14, 15, 21, 28] :

$$\begin{aligned} u(x, y) &= u_0 + u_x x + u_y y \\ v(x, y) &= v_0 + v_x x + v_y y \end{aligned} \tag{5}$$

The model parameters can be related to the camera motion and to the orientation of the ground plane relative to the image plane :

$$\begin{array}{ll} u_0 = -f_x \left[\frac{T_x}{Z_0} - \omega_y\right] & v_0 = f_y \left[-\frac{T_y}{Z_0} + \omega_x\right] \\ u_x = \frac{T_z + \gamma_x T_x}{Z_0} & v_x = \frac{f_y}{f_x} \left[\frac{T_y \gamma_x}{Z_0} - \omega_z\right] \\ u_y = \frac{f_x}{f_y} \left[\frac{T_x \gamma_y}{Z_0} + \omega_z\right] & v_y = \frac{T_z + \gamma_y T_y}{Z_0} \end{array} \tag{6}$$

The problem that remains to be solved, is the estimation of the plane slant and tilt slopes, which will determine the inverse projection operation.

2.1 Plane orientation estimation using the normal flow

The determination of the values of γ_x and γ_y is done in two steps. As a first step, the parameters for the affine model are estimated and used, in a second stage, to recover the plane orientation.

The affine motion parameters (equation (5)) can be directly estimated with 3 measurements of the optical flow $[u \;\; v]^T$, or a least squares approach can be used, if a large number of measurements is considered. However, using local measurements alone, only the component of the optical flow along the direction of the image gradient (normal flow) can be determined, while the other component remains unknown. This is the well known aperture problem [7, 8, 11].

To overcome this limitation, many authors use additional constraints on the flow field itself (such as smoothness), and have to use second or higher order image derivatives, which are not reliable. The method proposed here, relies exclusively on the normal flow field or, equivalently, on the first order time and space image derivatives.

Assuming image brightness constancy over time [12], the optical flow constraint equation is given by :

$$uI_x + vI_y = -I_t, \tag{7}$$

where I_x, I_y and I_t stand for the partial derivatives of the image with respect to x, y, and t. Combining the optical flow equation with the affine model equations (5), we have :

$$\left[\, I_y \;\; xI_y \;\; yI_y \;\; I_x \;\; xI_x \;\; yI_x \,\right] \boldsymbol{\theta} = -I_t \tag{8}$$

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

$$\boldsymbol{\theta} = [v_0 \; v_x \; v_y \; u_0 \; u_x \; u_y]^T \tag{9}$$

Therefore, the first order time and space image derivatives, over the planar patch, can be expressed as a linear function of the parameters to estimate. To estimate $\boldsymbol{\theta}$, it is sufficient to use 6 measurements of spatial-temporal image derivatives or, which is equivalent, 6 measurements of the normal flow.

By considering a large number of data points, we can solve an over-determined system, in least squares. The straight forward application of the least squares estimation procedure is quite sensitive to outliers, which usually leads to an ungraceful degradation of the estimates, in the presence of noise. To overcome this problem, we use a recursive estimation procedure, aiming at eliminating the effect of outliers. The method works as follows :

1. Choose randomly a set of data points $[I_x, \, I_y, \, I_t]^T$ to get an initial estimate of $\boldsymbol{\theta_0}$. Set $k = 1$.
2. Choose randomly a new set of data points $[I_x, I_y, I_t]^T$ such that the residue in equation (8), is small.
3. Estimate $\boldsymbol{\theta_k}$ based on the new data set. Set $k = k+1$.
4. Proceed to step (2) until $\boldsymbol{\theta}$ remains unchanged or k exceeds a given number of iterations.

The rationale behind this method is that by getting an approximate initial estimate, one can successfully improve $\boldsymbol{\theta}$ by selecting the data points that are most coherent with the model. Outliers will be discarded in the point selection mechanism, and the estimate improved in subsequent steps. Other alternative robust estimation procedures, like non linear least squares, can be found in [9].

2.2 Plane coefficients estimation - the intrinsic parameters

To this point we have presented a procedure to estimate the affine motion model parameters. Once $\boldsymbol{\theta}$ has been determined, the values of γ_x and γ_y can be estimated by referring to the equation set (6). However, if we assume that the motion of the robot, and the camera focal length are unknown, then γ_x and γ_y cannot be determined.

We will assume that, during the initialization phase, the robot is undergoing pure translational motion, which allows us to determine the values of the slant and tilt parameters scaled by f_x and f_y :

$$\begin{cases} \frac{\gamma_x}{f_x} = -\frac{v_x}{v_0} \\ \frac{\gamma_y}{f_y} = \begin{cases} -\frac{u_y}{u_0}, & \text{if } u_0 \neq 0 \\ \frac{u_x - v_y}{v_0}, & \text{otherwise} \end{cases} \end{cases} \tag{10}$$

Note that the case where $v_0 = 0$ is not considered, because if the robot is moving forwards ($T_y \neq 0$), v_0 will be different from zero.

By restricting the robot motion to be purely translational, we are able to estimate the plane slant and tilt parameters scaled by f_x and f_y, respectively. In order to determine the values of γ_x and γ_y, we would need to know the camera intrinsic parameters. However, we will show that it is sufficient to know the values of γ_x and γ_y up to a scale factor, as there is no need to recover the *absolute* orientation of the ground plane relative to the camera. Let us consider again the equations of the affine model parameters, slightly rearranged to highlight the role of the intrinsic parameters :

$$\begin{array}{ll} u_0 = -\frac{f_x T_x}{Z_0} + f_x \omega_y & v_0 = -\frac{f_y T_y}{Z_0} + f_y \omega_x \\ u_x = \frac{T_z}{Z_0} + \frac{\gamma_x}{f_x} \frac{f x T_x}{Z_0} & v_x = \frac{f_y T_y}{Z_0} \frac{\gamma_x}{f_x} - \frac{f_y}{f_x} \omega_z \\ u_y = \frac{f_x T_x}{Z_0} \frac{\gamma_y}{f_y} + \frac{f_x}{f_y} \omega_z & v_y = \frac{T_z}{Z_0} + \frac{\gamma_y}{f_y} \frac{f_y T_y}{Z_0} \end{array} \tag{11}$$

Admit, for a while, that the angular velocity is zero. In these circumstances, if we scale f_x or f_y up by a given factor, divide the corresponding T_x or T_y by the same factor and multiply the corresponding $\{\gamma_x, \gamma_y\}$ by the same factor again, all the flow parameters remain unchanged. With pure translation, if a camera with different intrinsic parameters is used, it is possible to find an orientation and velocity (given by suitably scaling $\{\gamma_x, \gamma_y, T_x, T_y\}$) that produces exactly the same flow field. As we are not interested in the absolute camera speed, nor on the absolute camera orientation, we can suppose that a *virtual* camera with unitary f_x and f_y, is being used.

If we consider the angular velocity term, the situation is somewhat different, as ω_z appears multiplied by different factors, in different equations. However, if we multiply f_x and f_y by *the same* factor (so that the ratio between f_x and f_y remains constant), and divide ω_y and ω_x by this factor, the equations remain unchanged as well. Hence, we only require the pixel aspect ratio to be known a priori, which is far less restrictive than knowing the individual values of f_y and f_x.

Therefore, there are three different situations to be considered. If the intrinsic parameter are known, the γ_x and γ_y parameters can be determined unambiguously. If only the aspect ratio is known, then the parameters can be estimated up to an unknown scale factor, which is enough for the method to be applied. Finally, if nothing is known about the intrinsic parameters, we can still use the method, but the robot motion must be constrained to pure translation.

3 Inverse Perspective Flow Transformation

This section shows how to perform the inverse perspective mapping of the flow field onto the horizontal plane, and how the obstacle detection algorithm becomes very simple.

Similarly to what has been proposed in [17, 29], if it is possible to inverse project the flow field perceived in the image plane, onto the horizontal plane, then the camera translation becomes parallel to the ground floor, and the rotation is solely around the vertical axis. The resultant flow pattern is much simpler, as illustrated in Figure 1.

In all the subsequent analysis, we will include the intrinsic camera parameters, which can be discarded in some conditions, according to the discussion in Section 2.2. Let

$$\begin{array}{l} C = \{X_c,\ Y_c,\ Z_c\ \} \\ H = \{X_h,\ Y_h,\ Z_h\ \} \end{array} \tag{12}$$

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

$$^{\mathbf{H}}\mathbf{P} = {}^{\mathbf{H}}\mathbf{R}_{\mathbf{C}}\ {}^{\mathbf{C}}\mathbf{P} \tag{13}$$

where $\mathbf{P}$ is a point in the 3D space measured in the coordinate frame specified by the upper index. The relative orientation results from a rotation of ψ around the camera x axis, and a rotation of ϕ around the camera y axis (which corresponds to a camera pointing down in front of a mobile robot). The rotation matrix, $^{\mathbf{H}}\mathbf{R}_{\mathbf{C}}$, will then have the following structure (in most practical systems, ϕ is small and can even be neglected) :

$$^{\mathbf{H}}\mathbf{R}_{\mathbf{C}} = \begin{bmatrix} c_\phi & -s_\phi s_\psi & -s_\phi c_\psi \\ 0 & c_\psi & -s_\psi \\ s_\phi & c_\phi s_\psi & c_\phi c_\psi \end{bmatrix} \tag{14}$$

where $s_\psi, c_\psi, s_\phi, c_\phi$ stand for $\sin\psi,\ \cos\psi,\ \sin\phi,\ \cos\phi$, respectively.

The perspective projection of a point in a plane β is defined as :

$$\mathbf{P}\ = \mathcal{P}_\beta(\ \mathbf{P}) \tag{15}$$

$$\begin{bmatrix} s\ x'_\beta \\ s\ y'_\beta \\ s \end{bmatrix} = \begin{bmatrix} f_x X_\beta \\ f_y Y_\beta \\ Z_\beta \end{bmatrix} \tag{16}$$

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

$$^{\mathbf{C}}\tilde{\mathbf{P}} = [\lambda \frac{x'_c}{f_x} \quad \lambda \frac{y'_c}{f_y} \quad \lambda]^T \tag{17}$$

which describes a beam passing through the projection center and the projection point in the image plane. As any 3D point expressed in the camera coordinate system, $\tilde{\mathbf{P}}$ can also be expressed in the frame attached to the horizontal plane :

$$^{\mathbf{H}}\tilde{\mathbf{P}} = {}^{\mathbf{H}}\mathbf{R}_{\mathbf{C}} \, {}^{\mathbf{C}}\tilde{\mathbf{P}} \tag{18}$$

Finally, this point can be projected into the horizontal plane, P_h, combining equations (14) and (15) :

$$\begin{bmatrix} x'_H \\ y'_H \end{bmatrix} = {}^H\mathcal{P}_C(x'_c, y'_c) \tag{19}$$

where ${}^H\mathcal{P}_C(x'_c, y'_c)$ denotes the operator, projecting the image plane onto the horizontal plane. This plane-to-plane projective transformation can be written in homogeneous coordinates as :

$$\begin{bmatrix} \mu\, x'_H \\ \mu\, y'_H \\ \mu \end{bmatrix} = \begin{bmatrix} c_\phi & -s_\phi s_\psi & -s_\phi c_\psi \\ 0 & c_\psi & -s_\psi \\ s_\phi & c_\phi s_\psi & c_\phi c_\psi \end{bmatrix} \begin{bmatrix} x'_c/f_x \\ y'_c/f_y \\ 1 \end{bmatrix} \tag{20}$$

This equation expresses how to determine the inverse projection in the horizontal plane, of the point projected in the image pixel (x'_c, y'_c). To obtain the inverse projection of a flow vector, the time derivatives of (x'_H, y'_H) must be used :

$$\begin{bmatrix} u'_H(x'_c, y'_c) \\ v'_H(x'_c, y'_c) \end{bmatrix} = \frac{\partial {}^H\mathcal{P}_C(x'_c, y'_c)}{\partial x'_c}\, u + \frac{\partial {}^H\mathcal{P}_C(x'_c, y'_c)}{\partial y'_c}\, v \tag{21}$$

which can be written in homogeneous coordinates as :

$$\begin{bmatrix} \rho\, u'_H(x'_c, y'_c) \\ \rho\, v'_H(x'_c, y'_c) \\ \rho \end{bmatrix} = \begin{bmatrix} f_x\ [(c_\psi + s_\psi\ \frac{y'_c}{f_y})\frac{u}{f_x} - s_\psi\ \frac{x'_c}{f_x}\frac{v}{f_y}] \\ f_y\ [s_\phi(s_\psi - c_\psi\ \frac{y'_c}{f_y})\frac{u}{f_x} + (c_\psi s_\phi\ \frac{x'_c}{f_x} + c_\phi)\frac{v}{f_y}] \\ (s_\phi\ \frac{x'_c}{f_x} + c_\phi s_\psi\ \frac{y'_c}{f_y} + c_\phi c_\psi)^2 \end{bmatrix} \tag{22}$$

The only problem left is the estimation of ψ and ϕ using the affine model and the γ_x, γ_y parameters discussed in the previous sections.

3.1 Estimating the Slant and Tilt parameters

During the initialization phase we must determine the inverse projection operator, preferably without explicitly calibrating the system. The process simply consists in determining γ_x, γ_y as explained in Section 2.2, taking into consideration what information is available, regarding f_x and f_y.

As any point in the camera coordinate system can be expressed in the horizontal plane coordinate system, according to equation (13), we can examine the z component of such term:

$$Z_H = s_\phi X + c_\phi s_\psi Y + c_\phi c_\psi Z \tag{23}$$

However, in terms of the camera coordinates, Z is a function of X and Y according to the ground plane constraint, given in equation (2). Combining these two equations, and knowing that in the coordinate system of the horizontal plane, all the points on the ground floor have a constant depth, Z_H, we get:

$$\begin{aligned} \psi \ &= -\arctan \gamma_y \\ \phi \ &= -\arctan(\gamma_x c_\psi) \\ Z_H &= c_\psi c_\phi\ Z_0 \end{aligned} \tag{24}$$

To summarize, once γ_x and γ_y have been estimated, as explained in Section 2.1, and 2.2, the optical flow can be inverse projected onto the horizontal plane. The values of ψ and ϕ are estimated only once during the initialization phase and remain constant, provided that the camera is in a fixed position relative to the robot.

3.2 Obstacle Detection

In the new coordinate frame, attached to the horizontal plane, the translation is constrained to be parallel to the "*new image*" plane (hence $T_z = 0$), the rotational component is solely around the Z axis and the distance to the ground floor is constant. Therefore, the optical flow is given by :

$$\begin{aligned} u_h(x_H, y_H) &= -\frac{T_x}{Z_H} + \omega_z y_H \\ v_h(x_H, y_H) &= -\frac{T_y}{Z_H} - \omega_z x_H \end{aligned} \tag{25}$$

There are now a number of important situations which should be considered in the analysis of this flow pattern.

1. When the angular velocity is zero, for a robot moving in piecewise linear trajectories, the optical flow vectors are constant for every point lying on the ground plane. A simple test can be performed to check if there are any obstacles above or below the ground floor. This detection mechanism relies solely on the fact that the optical flow should be globally constant, no matter what the motion direction and velocity might be. At this point the normal flow vectors can be projected onto the direction of motion, which is constant all over the image.
2. A second relevant situation arises when the robot linear velocity has only a forward component ($T_x = 0$). In this case, typical of many mobile platforms, u_h can be used to determine ω_z, and used to separate the translation component of v_h. As a result, the remaining flow should become constant for every point lying on the ground floor.

3. If none of the previous cases applies, the main difficulty is how to separate the rotation component from the purely translational component [2, 13, 25]. In our simplified reference frame, one could 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 depends on the x or y coordinates). The same kind of estimation techniques that were 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 analyzed, as in the former two cases.

4 Results

In this section we present some results obtained using the method proposed here. Initially, we will illustrate the robustness of the estimation procedure, using synthetic and real image data. Finally, results obtained with a real robotic application, are presented.

4.1 The estimation procedure

We have generated a flow pattern corresponding to a robot moving forward at a speed of 25 cm/s, with a camera pointing down at 45^o, yielding the following parameters : $\gamma_x = 0, \quad \gamma_y = 1$.

The camera intrinsic parameters assume a standard CCD (6.4mm by 4.8mm), a 4.8mm lens and 256×256 size images, leading to : $f_x = 192, \quad f_y = 256, \quad c_x = 128, \quad c_y = 128$.

The optical flow field was synthesized according to this camera settings and motion, and zero-mean Gaussian noise was added independently to each component. Figure 2 shows the synthetic flow field in the absence of noise.

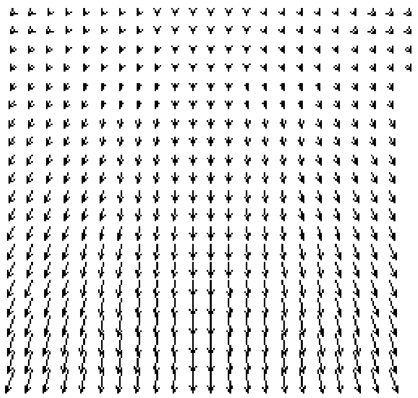

Fig. 2. Flow field generated for the simulation study, which corresponds to a robot, with a camera pointing down, moving on a flat ground.

We have applied the motion estimation algorithm with various levels of noise, and estimated the values of the plane tilt and slant angles. The results obtained are shown on Table 1. The method used to estimate the plane orientation angles, has performed robustly and accurate estimates were obtained, even with significant noise levels.

Std	# points	slant	tilt
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 noise standard deviation. The results show the robustness of the estimation procedure.The flow fields with different noise levels are shown in Figure 5.

Additional tests were made using a real image sequence acquired by a camera moving towards a slanted poster. The image sequence is shown in Figure 3.

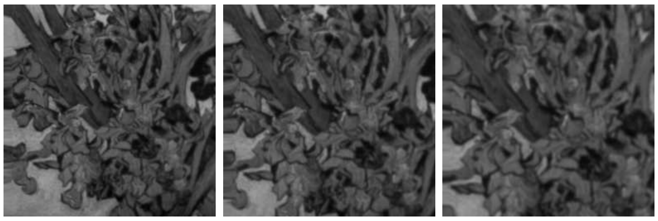

Fig. 3. Image sequence used for the estimation tests, acquired by moving a camera towards a slanted poster.

Figure 4 shows a sample of the optical flow field computed with this image sequence. For presentation purposes, we deliberately show the 2D flow field, so that the outlier vectors can be easily detected. There are various large vectors pointing in erroneous directions.

The affine motion parameters, estimated as explained in Section 2.1, were then used to reconstruct the flow field, which is shown for comparison in Figure 4. The "reconstructed" optical flow is, from a qualitative point of view, consistent with the input data, preserving important features like the Focus of Expansion. The estimation procedure is robust to the presence of outliers, as expected from the results obtained with synthetic data.

4.2 Obstacle Detection

Using the synthetic flow field, described in Section 4.1, with increasing levels of noise, we have estimated the flow parameters, and performed the inverse projection procedure assuming that the intrinsic parameters are unknown. The results are shown in Figure 5.

From the figure, we can see that even for severely corrupted data, the flow becomes relatively constant after the inverse projection, the degradation being more noticeable in the image areas where the flow amplitude is too small, and the signal to noise ratio is poor.

The system has been tested in real time, on a mobile robot. A camera with a 8mm lens has been installed on a TRC Labmate mobile platform. The camera was placed in the front part of the robot facing the ground

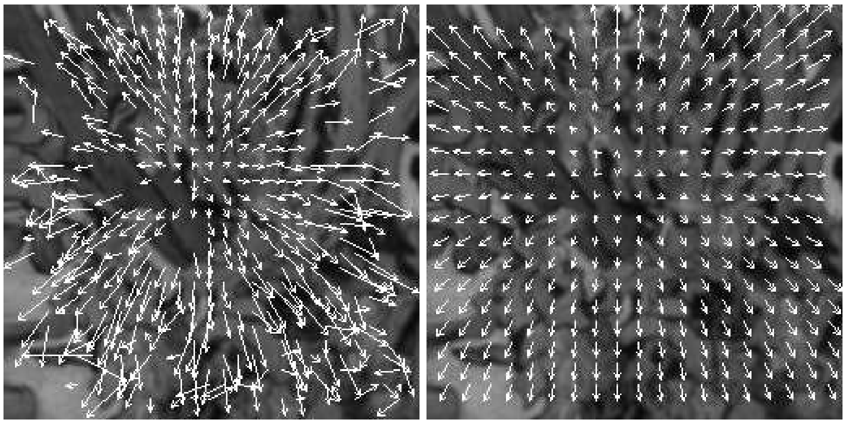

Fig. 4. The left image shows the optical flow estimated for the poster sequence. The right image shows the reconstructed flow field based on the affine model parameters.

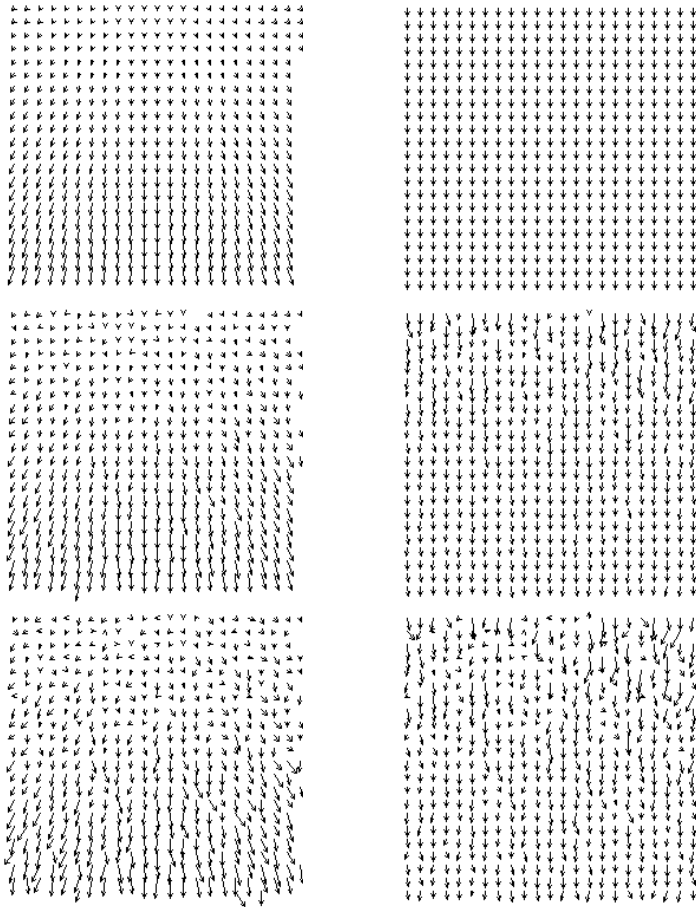

Fig. 5. The left column shows the synthetic flow fields corrupted by noise with 0.0, 1.0 and 2.0 pixels of standard deviation. On the right column we show the flow inverse projected onto the horizontal plane which becomes constant, and can be used for obstacle detection.

plane with an angle of about 65 degrees, with no calibration. The robot speed was set to 10 cm/s. The images resolution is 128×128 pixels and, to reduce the processing time, the computations are performed on a central window of 80×80 pixels. The running frequency of the vision loop is around 1Hz.

Figure 6 shows an image of the normal flow of the ground plane, measured during the initialization phase where the robot is performing a pure translation. The corresponding time and space image derivatives were used to estimate the affine model parameters and the inverse perspective transformation. This procedure does not require the knowledge of the camera intrinsic parameters or robot velocity. The figure also shows the corresponding inverse projected flow, which is approximately constant.

To test the obstacle detection capabilities of the system, we have performed several experiments, with differ-

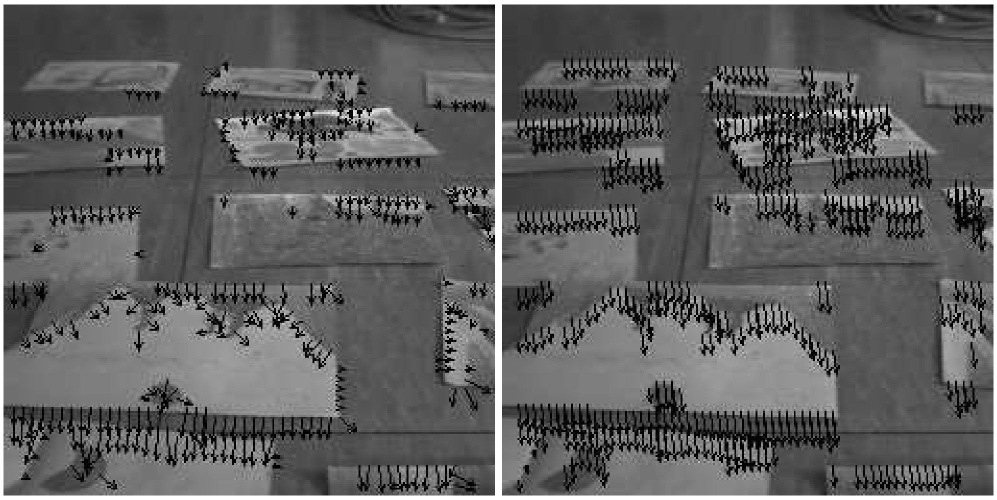

Fig. 6. The left image shows a sample of the ground plane normal flow field measured during the robot motion. The right image shows the resulting inverse projected flow.

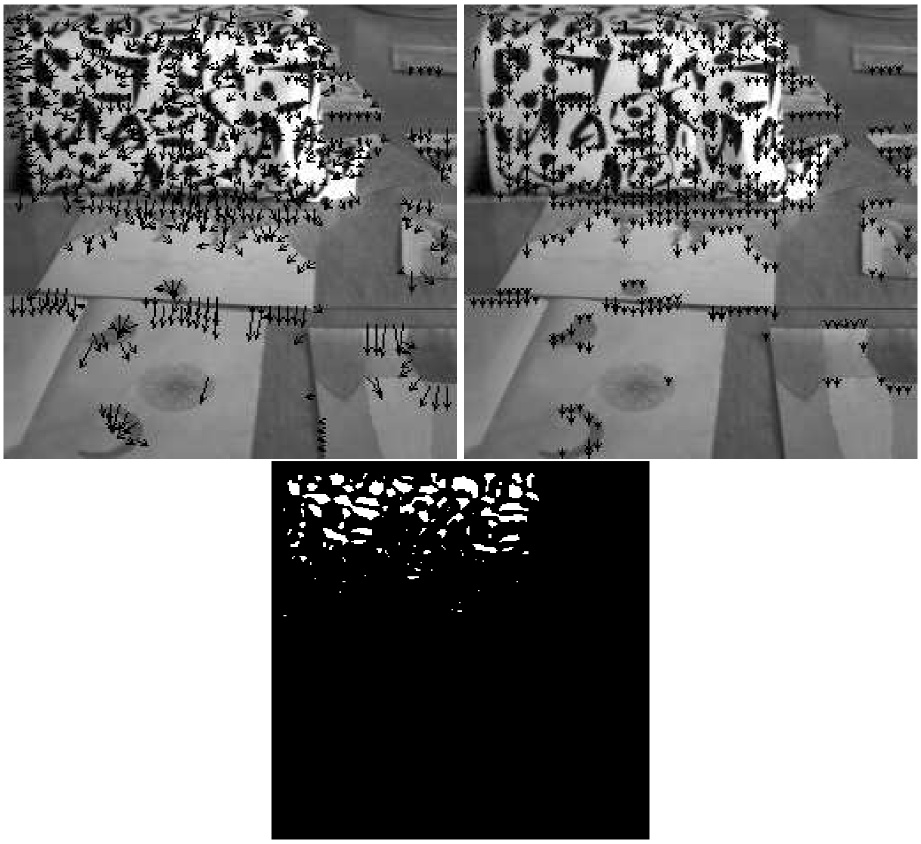

Fig. 7. The left image shows a sample of the ground plane normal flow field, during the robot motion. The right image shows the resulting inverse projected flow, where the obstacle can be easily detected, as shown in the bottom image.

ent obstacles. Figure 7 shows an example of the normal flow field measured during the robot motion, with an obstacle in the field of view. These flow fields are then inverse projected on the horizontal plane, as shown in the middle image of the same figure. The rightmost image of Figure 7 shows the image regions where points lying outside the ground plane, have been found. When the robot is undergoing pure linear motion, we have simply projected the inverse mapped flow in the y direction and subtracted the median flow. As the robot we used has a single forward component of the linear velocity, we have also estimated the rotation component and canceled this term from the overall inverse projected flow and then, the same detection process was applied. In both cases, the similar results were obtained.

In all the tests performed, the system performed robustly detecting various obstacles and stopping the robot accordingly. The resolution of the overall system determines the minimum obstacle size that can be detected, and strongly depends on the image resolution. If more computational power is available, the image resolution can be increased, the same happening with the global system resolution. Another important remark is that all these results were obtained without using any information about the camera intrinsic parameters or motion.

5 Conclusions

This paper describes a vision based 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 to be an obstacle.

The method exploits the geometric structure of the camera-robot-pavement arrangement to detect obstacles lying outside the ground floor. The approach is based on the inverse projection of the flow vector field onto the ground plane, where the analysis of the flow pattern is much simplified.

As opposed to other systems, an important feature is that the knowledge of the vehicle motion is not required and under certain circumstances, the approach is independent of the camera intrinsic parameters. For pure translational motion, it is not necessary to know the camera focal length and the flow vectors become constant all over the image, with the obstacles having a larger flow than points lying on the pavement. If the robot linear velocity has a single forward component, as in many mobile platforms, it is sufficient to know the pixel aspect ratio, and the system can deal with angular as well as linear motion.

In the initialization stage, the projective transformation between the image and horizontal planes is determined. A salient feature is that the estimation procedure is exclusively based on first order time and space image derivatives. The optical flow of the ground floor is approximated by an affine model. The affine model parameters, are estimated in a robust procedure and used to determine the inverse projection operator. No explicit calibration of the camera intrinsic or extrinsic parameters is needed.

Several tests were presented to illustrate the robustness of the estimation procedure using both synthetic and real image data. The system is suitable for implementation in real time, and results obtained with a mobile platform were described.

Acknowledgement. The research described in this paper has been supported by the Special Projects on Robotics of the Italian National Council of Research and by the ESPRIT project VAP-II. A fellowship from a bilateral collaboration between Consiglio Nazionale delle Ricerche (CNR) and Junta Nacional de Investigação Científica e Tecnológica (JNICT) is gratefully acknowledged by José Santos-Victor.

References

1. Y. Aloimonos. Purposive and qualitative active vision. In *Proc. of the 10th. IEEE Int. Conference on Pattern Recognition*, Atlantic City, NJ - USA, June 1990.
2. Y. Aloimonos and Z. Duvic. Estimating the heading direction using normal flow. *International Journal of Computer Vision*, 13(1):33–56, 1994.
3. S. Carlsson and J. Eklundh. Obstacle detection using model based prediction and motion parallax. In *Proc. of the 1st. European Conference on Computer Vision*, Antibes - France, April 1990.
4. R. Cipolla and A. Blake. Surface orientation and time to contact from image divergence and deformation. In *Proc. of the 2nd. European Conference on Computer Vision*, Santa Margherita, Italy, May 1992.
5. R. Cipolla, Y. Okamoto, and Y. Kuno. Robust structure from motion using motion parallax. In *Proc. of the 4th. International Conference on Computer Vision*, Berlin, Germany, May 1993.
6. W. Enkelmann. Obstacle detection by evaluation of optical flow field from image sequences. In *Proc. of the 1st. European Conference on Computer Vision*, pages 134–138, Antibes (France), 1990. Springer Verlag.
7. C. Fermüller. Global 3d motion estimation. In *Proc. of the IEEE CVPR*, New York, USA, June 1993.
8. C. Fermüller. Navigational preliminaries. In Y. Aloimonos, editor, *Active Perception*. Lawrence Erlbaum, 1993.
9. A. Gelb. *Applied Optimal Estimation*. MIT Press, 1974.
10. F. Girosi, A. Verri, and V. Torre. Constraints for the computation of optical flow. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 116–124, 1989.
11. B. Horn. *Robot Vision*. MIT Press, 1986.
12. B.K.P. Horn and B. Shunck. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.
13. B.K.P. Horn and E.J. Weldon. Direct methods for recovering motion. *International Journal of Computer Vision*, 2(1):51–76, 1988.
14. M. Irani, B. Rousso, and S. Peleg. Recovery of ego-motion using image stabilization. In *Proc. of the IEEE CVPR*, Seattle, USA, 1994.
15. J. Koenderink and J. van Doorn. Affine structure from motion. *Jounal of the Optical Society of America*, 8(2):377–385, 1991.
16. J.J. Little and A. Verri. Analysis of differential and matching methods for optical flow. In *Proc. of the IEEE International Conference on Robotics and Automation*, pages 173–180, 1989.
17. H. Mallot, H. Bulthoff, J. Little, and S. Bohrer. Inverse perspective mapping simplifies opticla flow computation and obstacle detection. *Biological Cybernetics*, 64:177–185, 1991.
18. H. Nagel. Displacement vectors derived from second-order intensity variations in image sequence. *CVGIP*, 21:85–117, 1983.
19. H. Nagel and W. Enkelmann. An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences. *IEEE Transactions on PAMI*, 8:565–593, 1986.
20. H. H. Nagel. On the estimation of optical flow: Relations between different approaches and some new results. *Artificial Intelligence*, 33:299–323, 1987.
21. S. Negahdaripour and S. Lee. Motion recovery from images sequences using only first order optical flow information. *International Journal of computer Vision*, 9(3):163–184, 1992.
22. G. Sandini and M. Tistarelli. Robust obstacle detection using optical flow. In *Proc. of the IEEE Intl. Workshop on Robust Computer Vision*, pages 396–411, Seattle, (WA), October 1990.
23. J. Santos-Victor, G. Sandini, F. Curotto, and S. Garibaldi. Divergent stereo for robot navigation: Learning from bees. In *IEEE International Conference on Computer Vision and Pattern Recognition.*, 1993.

24. J. Santos-Victor, G. Sandini, F. Curotto, and S. Garibaldi. Divergent stereo in autonomous navigation : From bees to robots. *International Journal of computer Vision*, 14(2):159–177, 1995.
25. D. Sinclair, A. Blake, and D. Murray. Robust estimation of egomotion from normal flow. *International Journal of Computer Vision*, 13(1):57–70, 1994.
26. M. Subbarao and A. Waxman. Closed form solutions to image flow equations for planar surfaces in motion. *Computer Vision Graphics and Image Processing*, 36:208–228, 1986.
27. V. Sundareswaran. Egomotion from global flow field data. In *Proc. of the IEEE Workshop on Visual Motion*, Princeton, New Jersey, October 1991.
28. J. Wang and E. Adelson. Layered representation for motion analysis. In *Proc. of the IEEE CVPR*, New York, USA, 1993.
29. T. Zielke, K. Storjohann, H. Mallot, and W. Seelen. Adapting computer vision systems to visual environment: Topographic mapping. In *Proc. of the 1st. European Conference on Computer Vision*, Antibes, France, April 1990.

José Santos-Victor was born in Lisbon, Portugal, on September 1965. He obtained the Licenciatura degree in Electrical and Computer Engineering from Instituto Superior Técnico in 1988, the MSc degree in 1991 and the Ph.D in 1995 from the same institution, specialising in Active Computer Vision and its applications to robotics. He has been a lecturer at the Instituto Superior Técnico, in the areas of Computer Vision and Robotics, Systems Theory, Control and Robotics, since 1988, and is currently an Assistant Professor and a researcher of the Institute of Systems and Robotics (ISR). He was a visiting researcher at Università degli Studi di Genova in 1992 and 1994 where we worked in various visual behaviours for mobile robots. He has been involved in various national and international research projects in the area of computer vision, robotics and control; and participates in a Human Capital Mobility network, SMART, in the areas of vision and robotics. His research interests are in the areas of Computer and Robot Vision, Robotics, Intelligent Control Systems, particularly in the relationship between visual perception and the control of action, namely in (land and underwater) mobile robots. He has published various conference and journal papers in the areas of computer vision and robotics.

Giulio Sandini was born on September 7, 1950 in Correggio, Italy. In 1976 he received a degree in Electrical Engineering from the University of Genoa, Italy. From 1976 to 1978 he has worked on models of the visual system and on electrophysiology of the cat visual cortex at the Laboratorio di Neurofisiologia del CNR with a Fellowship from the Scuola Normale Superiore in Pisa. In 1978 and 1979 he was a Visiting Scientist at the Harvard Medical School in Boston, developing a system for topographic analysis of brain electrical activity. Since 1980 he has been at Department of Communication, Computer and Systems Science of the University of Genoa, where he became associate professor in 1986. Currently teaches the course of: "Natural and Artificial Intelligent Systems". The main research interests are in Computer Vision and Robotics particularly in the areas of early vision and vision based control. He has also been very active in exploiting the neurobiological principles of early vision to postulate and implement artificial visual sensing at the "retinal" level of computer vision. Giulio Sandini has been among the founders of the "Laboratory for Integrated Advanced Robotics" (LIRA-Lab) at DIST and of the "Interuniversitary Center of Agricultural and Environmental Robotics".

This article was processed by the author using the LaTeX style file *cljour2* from Springer-Verlag.