Real-time Moving Obstacle Detection Using Optical Flow Models

Christophe Braillon, Cédric Pradalier, Jim Crowley, Christian Laugier

Real-time moving obstacle detection using optical flow models

Christophe Braillon[1], Cédric Pradalier[2], James L. Crowley[1] and Christian Laugier[1]

[1]Laboratoire GRAVIR
INRIA Rhône-Alpes
655 avenue de l'Europe
38334 Saint Ismier Cedex, France

[2]CSIRO ICT Centre
Autonomous Sytems Lab
1 Technology court
Pullenvale QLD 4069, Australia

Email: firstname.lastname@inrialpes.fr

Email: firstname.lastname@csiro.au

Abstract— In this paper, we propose a real-time method to detect obstacles using theoretical models of optical flow fields. The idea of our approach is to segment the image in two layers: the pixels which match our optical flow model and those that do not (i.e. the obstacles). In this paper, we focus our approach on a model of the motion of the ground plane. Regions of the visual field that violate this model indicate potential obstacles.

In the first part of this paper, we will describe the method we used to determine our model of the ground plane's motion. Then we will focus on the method to match both the model and the real optical flow field.

Experiments have been carried on the Cycab mobile robot in real-time on a standard PC laptop.

I. INTRODUCTION

This work takes place in the general context of mobile robots navigating in open and dynamic environments. Computer vision for ITS (Intelligent Transport Systems) is an active research area [1]. One of the key issues of ITS is the ability to avoid obstacles. This requires a method to perceive them.

In this article we address the problem of obstacle sensing through their motion (optical flow) in an image sequence. The perceived motion can be caused either by the obstacle itself or by the motion of the camera (which is the motion of the robot in the case of a camera fixed on it).

Many methods have been developed to find moving objects in an image sequence. Most of them use a fixed camera and use background subtraction (for example in [2] and [3]).

Others, based on optical flow have been inspired by biomimetic models [4]. For example, Franceschini and his collaborators have demonstrated models of optical flow based on insect retinas and have shown how such models may be used for local navigation. Duchon [5] proposed a reactive obstacle avoidance method based on insects' behaviour, and more recently Muratet and al. [6] implemented a model of the visual system of a fly, using a model helicopter. A survey of similar investigations is provided by Lee et al. ([7], as well as [8], [9]), who studied human perception of optical flow and their related behaviours. Other new approches by Hrabar et al. ([10], [11]) are based on optical flow and stereo camera to navigate urban environments with UAVs (Unmanned Aerial Vehicles) in a reactive way by doing a control oriented fusion.

These investigations cannot be easily integrated in complex systems, because they are purely reactive and do not provide the high level of information required for modern navigation techniques in unstructured environments.

Recently, in [12], a new approach to obstacle avoidance has been developed, based on ground detection by finding planes in images. The weak point of this method is that the robot must be in a static environment.

Model based approches using egomotion have been demonstrated in [13], [14]. The first one detects the ground plane by virtually rotating the camera and visually estimating the egomotion. The second one uses dense stereo and optical flow to find moving objects and robot ego-motion. These two methods have a large computational cost as several successive calculations (stereo, optical flow, egomotion, ...) are required.

In this paper, we demonstrate that by knowing the motion of the camera (in our case we used the odometric information), we can model the motion of the ground plane and determine the location of the obstacles.

In a first step we determine the ground plane's optical flow field using the odometric data. In the next step, we try to match this theoretical field with the actual motion field. The pixels which do not match the model are either obstacles (objects outside the ground plane) or objects in the ground plane that are moving.

One key point in this method is that we do not compute explicitly the optical flow of the image at any time. Optical flow computation is very expensive in terms of CPU time, is inaccurate and sensitive to noise. In general we can see in the survey led by Barron et al. ([15]) that the accuracy of optical flow computation is linked to the computational cost. We model the expected optical flow (which is easy and quick to compute) to get rid of the inherent noise and the time consuming optical flow step. As a consequence we are able to demonstrate robust and real-time obstacle detection.

II. OUR OPTICAL FLOW MODEL

By definition, an optical flow field is a vector field that describes the velocity of pixels in an image sequence. The

first step of our method is the modeling of the optical flow field for our camera.

This model is based on the classical pinhole camera model, that is to say, we neglect the distortion due to the lens. We also assume that there is no skew factor. We will see in the experimental results that these two assumptions are valid.

A. Notations

Figure 1 shows the position of the camera fixed on our Cycab robot, the coordinates frames and the notations we will use in all this paper.

The robot's position is given in the coordinate system of the world by (x, y) and its orientation is called θ. In the next parts we will call $\dot{x}, \dot{y}, \dot{\theta}$ the three derivatives of x, y, θ with respect to time.

We assume that at time t, we know the intensity image I_t, the linear velocity v_t and rotational velocity ω_t of the robot.
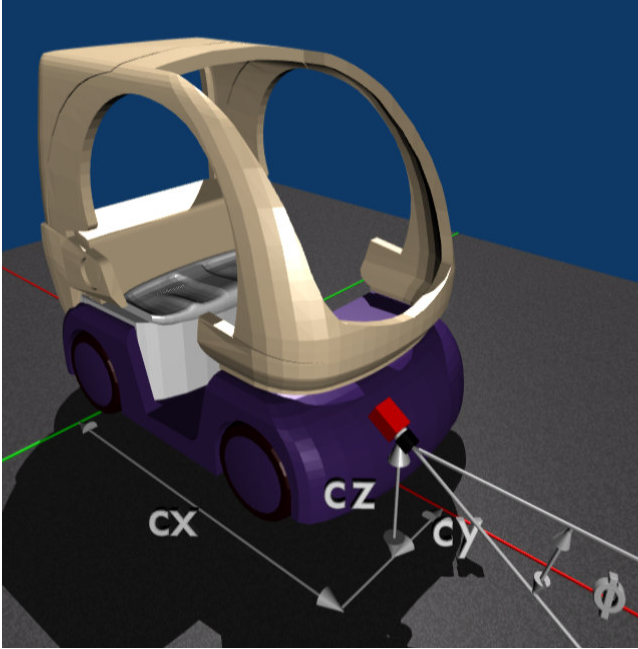

Fig. 1. Configuration of the camera on our mobile robot. The red line represents the abscissa axis and the green one the ordinate axis. The origin of the robot frame is the projection of the middle of the rear axis on the ground. We call (c_x, c_y, c_z) the position of the camera in this frame and ϕ its orientation (we suppose that only the tilt is not null).

We use the projective geometry formalisms to simplify the theoretical aspects of the motion modeling. The points of the world are noted $(X, Y, Z, 1)^T$ in the world frame and the pixel coordinates are $(u, v, w)^T$.

The camera's projection matrix (pinhole model) with reference to the camera frame is written:

$$\mathcal{P}_{Camera} = \begin{pmatrix} \alpha_u & 0 & u_0 & 0 \\ 0 & \alpha_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

B. Description of the model

The model we propose to study is the one of the ground. We assume that the ground is a plane located at $Z = 0$ coordinate. This means that the ground plane's points are written $(X, Y, 0, 1)^T$ Thus we can write the projection equation of the ground point on the image as:

$$\begin{pmatrix} u \\ v \\ w \end{pmatrix}_{Image} = \mathcal{P}_{World} \begin{pmatrix} X \\ Y \\ 0 \\ 1 \end{pmatrix}_{World} \tag{1}$$

In our case, we have a projection from one plane (the ground) to another (the image), which can be expressed as an homography:

$$\begin{pmatrix} u \\ v \\ w \end{pmatrix}_{Image} = \mathcal{H} \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix}_{Ground} \tag{2}$$

Actually, the matrix $\mathcal{H}$ is the matrix $\mathcal{P}$ in which we remove the third column. This point is important because we need an analytical expression of $\mathcal{H}$ which involves the position and orientation of the robot and camera.

The homography matrix allows us to estimate the position of a ground point in the image, but in this study, we need to compute the optical flow for a given pixel. Therefore, we need to differentiate equation (2) so that we can infer the relation between pixel coordinates and the related optical flow vector.

$$\begin{pmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{pmatrix}_{Image} = \dot{\mathcal{H}} \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix}_{Image} + \mathcal{H} \begin{pmatrix} \dot{X} \\ \dot{Y} \\ 0 \end{pmatrix}_{Image} \tag{3}$$

Since we are working on the ground pixels, the relation can be simplified because they are stationary in the world coordinate system, thus $\dot{X} = 0$ and $\dot{Y} = 0$. The new relation is:

$$\begin{pmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{pmatrix}_{Image} = \dot{\mathcal{H}} \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix}_{Image} \tag{4}$$

From equation (2) and (4) we can infer the following relation between the homogeneous coordinates of a pixel of the ground and its coordinates derivatives:

$$\begin{pmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{pmatrix}_{Image} = \dot{\mathcal{H}}\mathcal{H}^{-1} \begin{pmatrix} u \\ v \\ w \end{pmatrix}_{Image} \tag{5}$$

Using equation (5) we can express the optical flow vector $\vec{f}(u, v, w)$ by:

$$\begin{aligned} \vec{f}(u, v, w) &= \left(\left(\frac{\dot{u}}{w} \right), \left(\frac{\dot{v}}{w} \right) \right) \\ &= \left(\frac{\dot{u}w + u\dot{w}}{w^2}, \frac{\dot{v}w + v\dot{w}}{w^2} \right) \end{aligned}$$

As we on know the u and v coordinates of the pixels we need to simplify the previous equation to be able to compute the optical flow vectors. We use the following transformations:

$$\begin{pmatrix} u' \\ v' \end{pmatrix} = \frac{1}{w} \begin{pmatrix} u \\ v \end{pmatrix} \text{ and } \begin{pmatrix} \dot{u}' \\ \dot{v}' \\ \dot{w}' \end{pmatrix} = \frac{1}{w} \begin{pmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{pmatrix}$$

and finally, by combining with equation (5), we obtain:

$$\begin{pmatrix} \dot{u}' \\ \dot{v}' \\ \dot{w}' \end{pmatrix}_{Image} = \dot{\mathcal{H}}\mathcal{H}^{-1} \begin{pmatrix} u' \\ v' \\ 1 \end{pmatrix}_{Image} \quad (6)$$

In equation (6) (u', v') are the euclidian coordinates of a pixel in the image (whereas (u, v, w) are its projective coordinates).

Now we can obtain the optical flow vector $\vec{f}$ for the pixel (u', v') by the formula:

$$\vec{f}(u', v') = \begin{pmatrix} \dot{u}' - u'\dot{w}' \\ \dot{v}' - v'\dot{w}' \end{pmatrix} \quad (7)$$

Finally from equations (6) and (7) we can express the theoretical optical flow vector for each pixel in the image (with the assumption that each pixel is in the ground plane).

C. Evaluation of the homography matrix

As we only focus on a differential problem, we can assume that the robot has the configuration $(x = 0, y = 0, \theta = 0)$ all the time. In this case, the velocities are: $\left(\dot{x} = v_t, \dot{y} = 0, \dot{\theta} = \omega_t \right)$, where v_t and ω_t are the linear and angular velocities of the robot, respectively.

Using the notations introduced in section II-A, we can express the projection matrix and the homography matrix as well:

$$\mathcal{H} = \begin{pmatrix} h_{1,1} & h_{1,2} & h_{1,3} \\ h_{2,1} & h_{2,2} & h_{2,3} \\ h_{3,1} & h_{3,2} & h_{3,3} \end{pmatrix}$$

with:

$$\begin{aligned} h_{1,1} &= u_0 \cos\phi \\ h_{1,2} &= -\alpha_u \\ h_{1,3} &= \alpha_u + u_0 \left(-\cos\phi + c_z \sin\phi \right) \\ h_{2,1} &= -\alpha_u \sin\phi + v_0 \cos\phi \\ h_{2,2} &= 0 \\ h_{2,3} &= \alpha_v \left(\sin\phi + c_z \cos\phi \right) + v_0 \left(-\cos\phi + c_z \sin\phi \right) \\ h_{3,1} &= \cos\phi \\ h_{3,2} &= 0 \\ h_{3,3} &= -\cos\phi + c_z \sin\phi \end{aligned}$$

We have also:

$$\dot{\mathcal{H}} = \begin{pmatrix} \dot{h}_{1,1} & \dot{h}_{1,2} & \dot{h}_{1,3} \\ \dot{h}_{2,1} & \dot{h}_{2,2} & \dot{h}_{2,3} \\ \dot{h}_{3,1} & \dot{h}_{3,2} & \dot{h}_{3,3} \end{pmatrix}$$

with:

$$\begin{aligned} \dot{h}_{1,1} &= \alpha_u \omega_t \\ \dot{h}_{1,2} &= u_0 \omega_t \cos\phi \\ \dot{h}_{1,3} &= -u_0 v_t \cos\phi \\ \dot{h}_{2,1} &= 0 \\ \dot{h}_{2,2} &= \left(-\alpha_v \sin\phi + v_0 \cos\phi \right) \omega_t \\ \dot{h}_{2,3} &= \left(\alpha_v \sin\phi - v_0 \cos\phi \right) v_t \\ \dot{h}_{3,1} &= 0 \\ \dot{h}_{3,2} &= \omega_t \cos\phi \\ \dot{h}_{3,3} &= -v_t \cos\phi \end{aligned}$$

Figure 2 shows the result of our model for a camera at position $c_x = 1.74$ m, $c_y = 0$ m, $c_z = 0.83$ m and $\phi = 0$ rad. The model is valid only below the horizon line whose equation is: $y = v_0 - \alpha_v \tan\phi$. Therefore there is no flow vector above the horizon line.
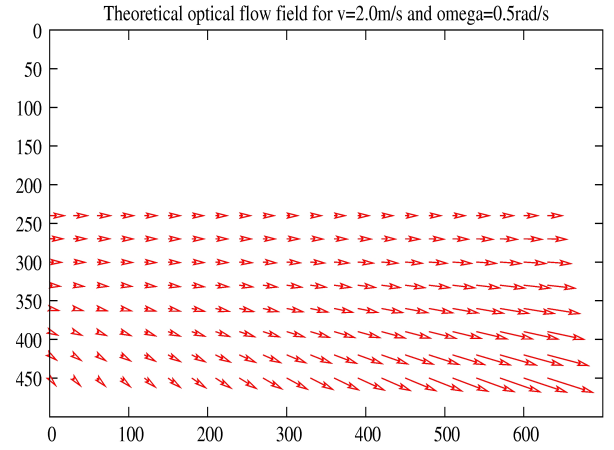

Fig. 2. Example of theoretical optical flow field for a moving robot with a velocity of 2m.s-1 and a rotation speed of 0.5rad.s^{-1}

III. GROUND MOTION SEGMENTATION

In part II-B, we have seen how to compute the theoretical optical flow vector for a given pixel in the image. This means that if the pixel corresponds to a point on the ground, it will have the same optical flow vector as the model. Therefore to know if a pixel does not belong to the ground, we will compare its actual displacement with the theoretical one.

One way to do that would be to compute the optical flow vector for each pixel and compare it to the theoretical vector. We did not use this method for three reasons:

1) The optical flow computation for all the pixels we want to test would be very expensive in CPU-time
2) The computed optical flow field is in general not accurate and very noisy (if we used methods which are real-time) especially on poor-textured surfaces like a road.
3) The comparison function is not easy to define. The simple euclidian norm is not sufficient (the points far from the camera will always be seen as ground points because of their slow motion)

To solve the problem we use a generative method. For each pixel we calculate the theoretical optical flow vector and determine if it is a possible displacement. To do that, we measure the similarity of a pixel $(u', v')^T$ in an image I_t with the pixel $(u', v')^T - \Delta t \vec{f}(u', v')$ in image $I_{t-\Delta t}$.

Thus we define a new image J_t of similarity using the following formula:

$$J_t(p) = Sim_{I_t, I_{t-\Delta t}}\left(p, p - \Delta t \vec{f}(p)\right) \qquad (8)$$

Where $p = (u', v')^T$ is a pixel in the image and $Sim_{A,B}(p, q)$ is a similarity between the neighbourhood of pixel p in image A and the neighbourhood of pixel q in image B. We will assume that the intensity of the points will not change between images A and image B. We can see on figure 3 an example of similarity measure between two consecutive images.
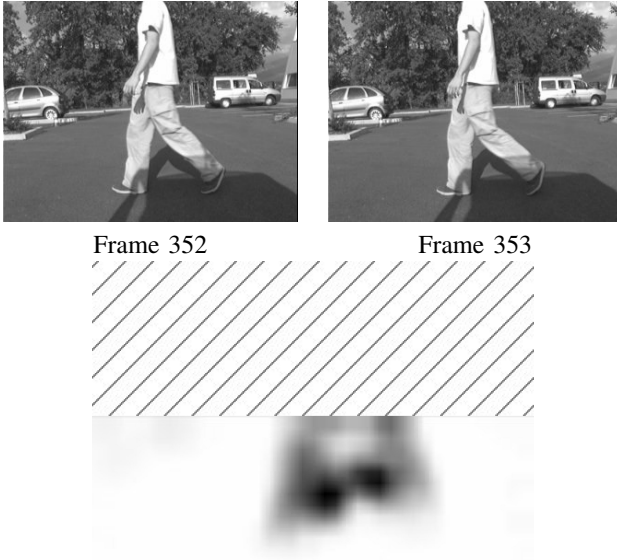

Fig. 3. Similarity measurement: on the top left, image $I_{t-\Delta t}$, on the top right, image I_t. On the bottom the result of a similarity measure. The hatched region corresponds to the area where the theoretical flow cannot be computed (these pixels will never belong to the ground)

We tried several similarity measures such as: SAD (Sum of Absolute Differences), SSD (Sum of Squares Differences), ZSAD (Zero mean Sum of Absolute Differences), ZSSD (Zero mean Sum of Squared Differences), ZNCC (Zero mean Normalized Cross Correlation). The best result were given by the SAD and SSD measures. Actually ZSAD, ZSSD and ZNCC are zero-mean and ZNCC is normalized, this leads in general to incorrect associations, because they can associate a dark region and a bright one. In our case, the lighting condition between two images are the same, that is why the best results were given by SAD and SSD. Figure 4 shows the results of the different similarity measures.
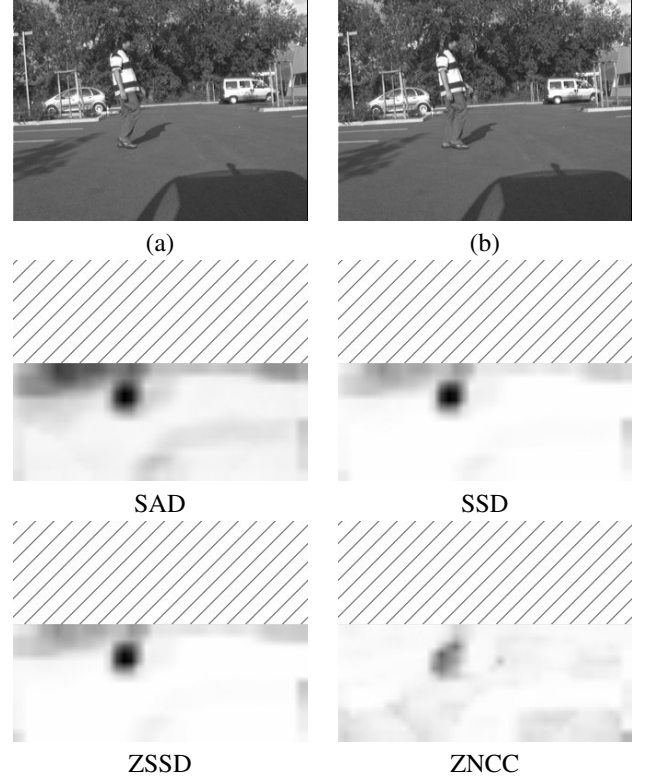

Fig. 4. Examples of similarity measures between image (a) and (b)

We can see that the SSD measure is more selective than SAD. This is due to the fact that SSD emphasize big differences because of the square. As a result, we have more details in SAD but also more noise. Therefore we will use the SSD measure in all our experiments.

The SSD image we obtain is normalized by its maximum value so that the each pixel has a value between 0 and 1. We have then applied a threshold of 0.7 to segment the obstacles.

IV. EXPERIMENTAL RESULTS

A. System description

In our experiments, we use a Cycab mobile robot, which provides odometric measurements and a digital video stream (a color camera is fixed on the front of the robot). We need to have synchronized odometry data and video stream (otherwise the detection will be incorrect). The robot is driven manually in a carpark where pedestrians and cars are moving. To give an overview of our experimental results, we will use a video where two pedestrians are crossing in front of the robot.

B. Results and comments

Figures 5 and 6 show the video sequence on which we superimposed the contour of the detected objects and the SSD image for the corresponding frames, respectively. The horizontal line in the image corresponds to the horizon. Obviously all the pixels above this line do not belong to the ground plane. We can also see that even far objects are detected (for example on frame 135 on figure 5).

We have simulated a CPU overload in our experiments by running useless processes when doing our experiments. It has shown that even with gaps in the video sequence (some frames were lost because of other processes slowing the computer down) our method is able to detect the obstacles correctly . This is an important characteristic since the robots' embedded computers have a high CPU load (especially when using computer vision) and frame loss may occur.
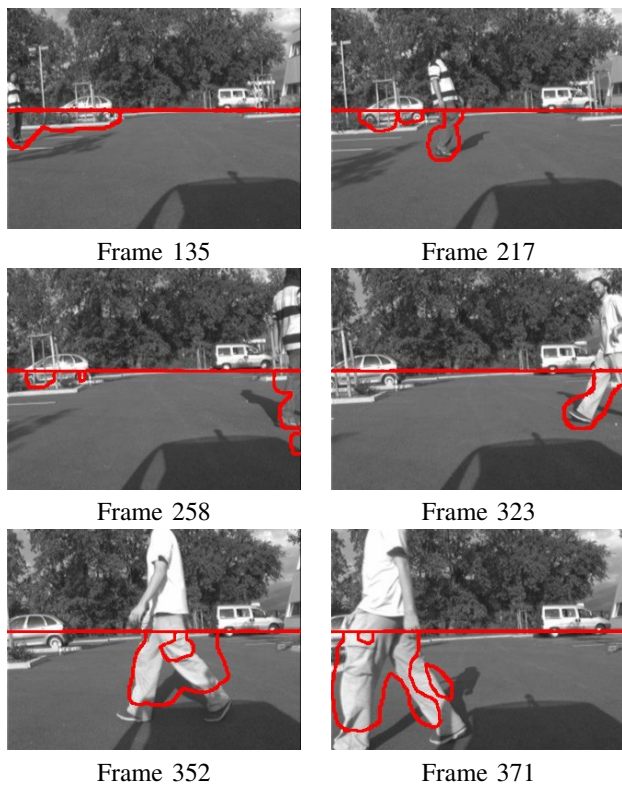

Frame 135   Frame 217

Frame 258   Frame 323

Frame 352   Frame 371

Fig. 5. Six frames of the video sequence. The red curves are the contours of the obstacles (see figure 6).

C. Discussion

We have not led a quantitative survey on the impact of pan and roll of the camera, its position error and flatness of the ground, but when watching the video, we can see that the roll is not null (we can easily see that on figure 7) and it does not seem to disturb the detection. Furthermore the position of the camera has been roughly estimated. Thus we can guess that the impact of such errors is not significant, but we will study the effect of these parameters in a future work.

The model we described in II considers that the ground points are not moving which is a reasonable assumption. When there is a fixed shadow on the ground, no obstacles are detected and the observed optical flow is the same as the models'. When the object that casts the shadow starts moving, the optical flow on the shadows' edge changes and does not match the model

Frame 135   Frame 217
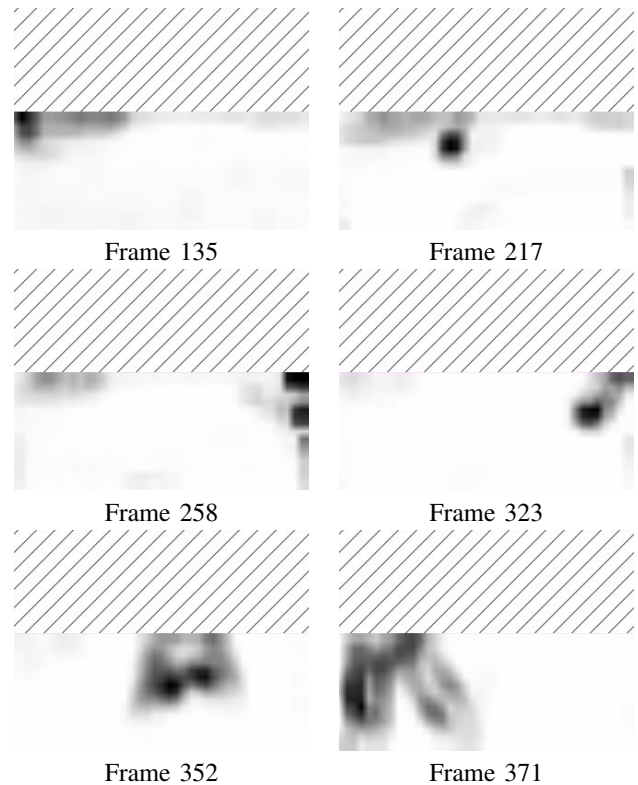
Frame 258   Frame 323

Frame 352   Frame 371

Fig. 6. SSD images corresponding to figure 5. The grey scale corresponds to the SSD value. White means that the SSD is low and black that it is high. The hatched part corresponds to an area of the image which cannot be the ground (the algorithm mark these pixel as unusable)

anymore. Therefore in our experiments we have seen false positives on the edge of moving shadows (see figure 7) This is due to our assumption that the point of the ground have a constant illumination (which is false is the case of a moving shadow). This problem can be removed by fusing with other techniques which are not shadow-sensitive (e.g. stereo)

In paragraph IV-B, we have described the method we used to find the obstacles in the similarity image. If we have a combination of fast and slow motions, the slow motion is not as well detected as the fast motion, because the fast motion takes priority (due to the normalization we performed). However the slow motion is still visible in the similarity image (see frame 352 in figure 6). This is due to both the scale applied to the similarity image (to be between 0 and 1) and the fixed threshold.

V. FUTURE WORK

In future work, we will explore new models of optical flow, which will focus on the impact of the ground flatness (or curvature) on the efficiency of our method. Improvements can be made with regards to calibrating the whole system online. We are currently researching this aspect in the context of a CSIRO-INRIA collaboration, and we expect to submit these results in future articles.
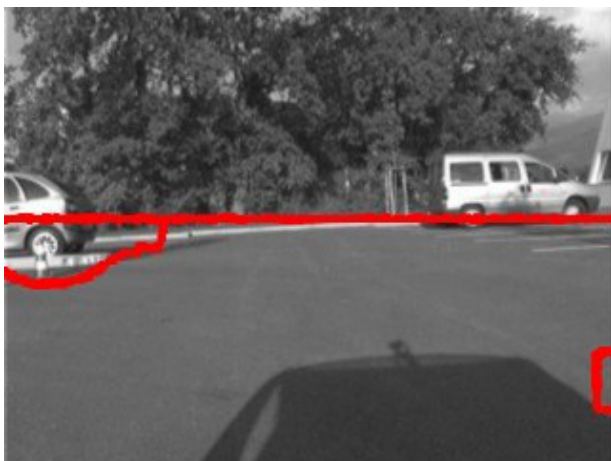
Fig. 7.   On frame 401 we can see the detection of a shadow as an obstacle
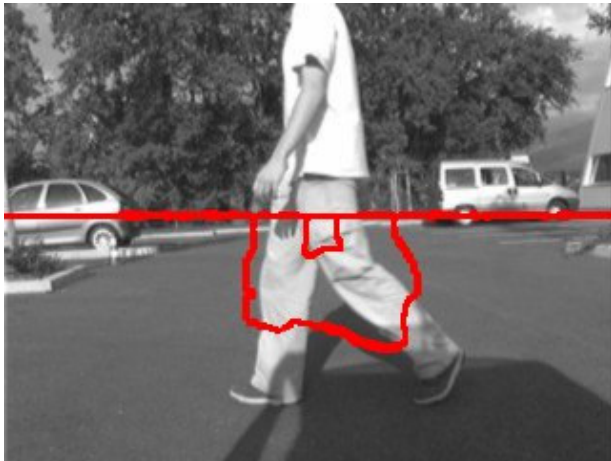
Frame 353

Frame 424

Fig. 8.   On frame 353, the far obstacles are not correctly detected because the magnitude of the optical flow vector is significantly smaller than that of the pedestrian moving close to the camera. On frame 424, the far obstacles are detected because there is no fast-moving object

VI. CONCLUSION

We proposed in this paper a new way of detecting obstacles in a mobile robot environment by separating them from the ground floor in an image sequence. The originality in this method is that we detect obstacle based on the motion in the image sequences. Firstly, we extract a model of the environment and then we separate the obstacles from the ground by trying to fit the theoretical optical flow model to the observed video stream. Tests show a computation time of 25 to 30 milliseconds, which corresponds to a frame rate between 30 and 40 Hz, on a standard 2.0 GHz laptop PC.

VII. ACKNOWLEDGMENT

REFERENCES

[1] E. Dicksmanns, "The development of machine vision for road vehicles in the last decade," 2002.
[2] C. Stauffer and W. Grimson, "Adaptative background mixture models for real-time tracking," January 1998.
[3] R. Collins, A. Lipton, H. Fujiyoshi, and T. Kanade, "Algorithms for cooperative multisensor surveillance," *Proceedings of the IEEE*, vol. 89, no. 10, pp. 1456 – 1477, October 2001.
[4] N. Franceschini, J. M. Pichon, and C. Blanes, "From insect vision to robot vision," in *Philosophical Transactions of the Royal Society of London*, 1992, pp. 283–294.
[5] A. Duchon, "Maze navigation using optical flow," in *International Conference on Simulation of Adaptative Behaviour*, 1996.
[6] L. Muratet, S. Doncieux, and J. Meyer, "A biomimetic reactive navigation system using the optical flow for a rotary-wing UAV in urban environment," in *International Conference on Robotics and Automation*, New Orleans, april 2004.
[7] D. Lee, "A theory of visual control of braking base on information about time-to-collision," *Perception*, vol. 5, pp. 437–459, 1976.
[8] ——, "Plummeting gannets: a paradigm of ecological optics," *Nature*, vol. 293, september 1981.
[9] D. Lee, D. Young, and D. Rewt, "How do somersaulters land on their feet?" in *Journal of Experimental Psychology*, vol. 18, 1992.
[10] S. Hrabar, P. Corke, G. Sukhatme, K. Usher, and J. Roberts, "Combined optic-flow and stereo-based navigation of urban canyons for a UAV," 2005.
[11] S. Hrabar and G. Sukhatme, "A comparison of two camera configurations for optic-flow based navigation of a UAV through urban canyons," pp. 2673–2680, sep 2004.
[12] K. Young-Geun and K. Hakil, "Layered ground floor detection fo vision-based mobile robot navigation," in *International Conference on Robotics and Automation*, New Orleans, april 2004, pp. 13–18.
[13] Q. Ke and T. Kanade, "Transforming camera geometry to a virtual downward-looking camera: robust ego-motion estimation and ground layer detection," 2003.
[14] A. Talukder and L. Matthies, "Real-time detection of moving objects from moving vehicle using dense stereo and optical flow," October 2004.
[15] J. Barron, D. Fleet, and S. Beauchemin, "Performance of optical flow techniques," in *IJCV*, vol. 12, no. 1, 1994, pp. 43–77.