Multi-directional geometric enhancement of optical flow approach for moving objects recognition

Stefan Dumitru, Danut Bucur and Vladimir Balan

Abstract. A variant of the Lucas-Kanade tracking procedure jointly used with the infinitesimal interactive affine motion approximation of displacement, are applied to subsequent camera-acquired images. The outputs are subject to ICA, followed by a variance-based decision criterion, in order to outline the moving object (assumed to have a closed non self-intersecting contour). The output determines the segmentation of the image into moving obstacles and dominant plane.The originality of the approach resides in using a unitary circular Gateaux-derivative treatment of all neighboring directions in the process of preliminary motion-evaluation based on pixel shift decomposition, which allows better detection of significantly mobile clusters of pixels. In order to highlight our results, a comparison was made between the proposed algorithm and some others from literature. The algorithm can be applied to a large class of images and is intended to be an interactive tool in obstacle-detection for single-camera endowed moving robots.

M.S.C. 2010: 42A20, 42A32.
Key words: optical flow; dominant plane detection; robot vision; Independent Component Analysis; Gateaux derivatives.

1 Introduction

Using video sensors for mobile robots navigation offers clear advantages besides other sensors from the same class. Still, the video cameras are cheaper than other sensors, such as: laser range-finders, sonar and other radar-like sensors, which are used in similar domains [17],[2],[5],[16],[11].

Moreover, artificial vision also offers flexibility of use, which other sensors hardly achieve, since they provide only one fixed method for data collecting. Despite the fact that in some environments or different tasks, other sensors may provide more accurate data, the artificial vision gives information about the texture, color of surface, environment structure and motion, details which are relevant to robot navigation [3].

BSG Proceedings, Vol. 19, 2012, pp. 45-57.

As all other methods, the artificial vision has its own disadvantages. Its performances decrease in some environmental conditions such as low light situations when the captured image is noisy and may lead to wrong results due to errors induced by the noise. Also, applying artificial vision in real tasks is restricted by the required processing resources (in order to process in short time high amount of data collected from the environment) and by the environment structure (situations when the geometry of the environment hampers the object detection).

Optical flow is a method to estimate the motion based on local derivatives in a sequence of images. In other words, in image plane (2D), it specifies how much a pixel has been moved in a sequence of pictures. This displacement of pixels causes temporal variations of image intensity.

Computing the optical flow follows two steps:

1. computing spatial-temporal derivatives;
2. computing velocity by means of spatial-temporal derivatives.

The algorithms developed in order to compute the optical flow are based on methods created by Lucas and Kanade [10] and Horn and Schunck [6], in 1981.

Starting from these methods, Idaku Ishii & al [8] designed a real time motion detection system using a high-speed camera at 1000 fps. Further, J. Li & al [9] used these methods in surveillance applications and video analysis.

Within this developed framework, Jean-Yves Bouguet described a pyramidal method of the optical flow [1], setting the correspondence between the pixel position in the current image, and the position of the same pixel in the next image. The method managed to detect the obstacles that appear on the ground plane, assumed as dominant. Numerous enhancements of this algorithm were proposed, e.g. [13],[12],[14].

One way of achieving fair object recognition was the usage of Independent Component Analysis (ICA) - a statistical technique which separates signals into independent components. In our current research, we used the FastICA algorithm - an optimized version of the original ICA algorithm, [7] - in order to extract the independent components from an optical-flow image sequence, components which will be used to identify moving objects located within the dominant plane [7]. This is possible, assuming that the planar surface corresponds to the major part of the image. ICA is robust against the noises and errors that may occur due to optical flow computing, the errors produced by the dominant plane extraction from the image sequence, and, last but not least, the noise induced by video camera.

The purpose of our proposed method is to improve the optical flow algorithm by a new approach of spatial-temporal derivatives, using slant Gateaux directional derivatives.

2 Proposed algorithm

The proposed method (fig. 1) captures images from the robot's environment using a monocular video system, builds the optical flow using Lucas-Kanade algorithm, which, alongside with the affine motion form the inputs for ICA module. Having two inputs, ICA provides two outputs which are analyzed using a decision criterion, after which the obstacles are detected.

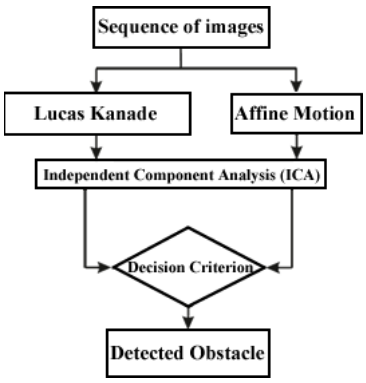

Figure 1: Obstacle detection proposed algorithm.

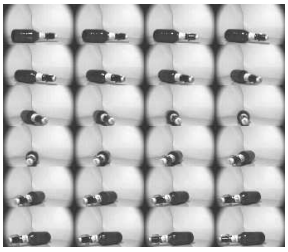

Figure 2: Samples of images captured by the robot camera.

2.1 The image sequence

The image sequence was captured using a high speed camera, at a resolution of 640 x 480 pixels and having the capturing speed of 1000 fps. The images were taken under normal lighting condition. Samples of the captured images are presented in figure 2. We assume that robot continuously moves on a planar surface, which ensures that obtained data are reliable. Having a high speed camera on the robot, the displacement speed does not play an important role in implementing the method.

The advantage of using a high speed camera is that it allows to catch even the smallest motion in the robot's view range, providing accurate information from the environment. Using a normal video camera requires the robot to move with a certain speed in order to avoid the blur effect that occurs in the transition frames in the captured image sequence. The used 1000 fps speed allowed us to obtain the most accurate detection of obstacles that may occur in the workspace of an autonomous robot.

2.2 The optical flow

The optical flow represents the motion of the environment related to the robot. Based on the apparent growing of closer objects and shrinking of more distant ones, landmarks are created which are used in robotics in the orientation system of autonomous robots.

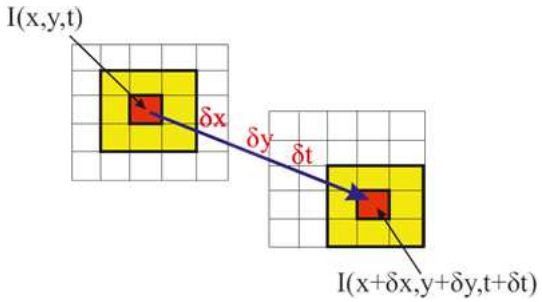

Figure 3: Pixel motion from frame t to frame $t+1$.

In our proposed method we use a 2D image sequence formed under perspective projection based on the relative motion of the camera and the objects that it captures. Motion in the picture causes temporal variation of pixel intensity. It is assumed that these changes are determined only by motion. Let $I(x, y, t)$ be the intensity value at pixel location (x, y) from an $n \times n$ neighborhood at some moment t. Taking into consideration the fact that $I(x, y, t)$ and $I(x + \delta x, y + \delta y, t + \delta t)$ are intensity values of the same pixel, we have:

$$I(x, y, t) = I(x + \delta x, y + \delta y, t + \delta t). \tag{2.1}$$

Suppose that within the displacements $\delta x, \delta y, \delta t$ of the pixel from frame I_t to frame I_{t+1}, the tangential component shown in figure 3 is small enough; then we can write the Taylor expansion of (2.1) as follows:

$$I(x + \delta x, y + \delta y, t + \delta t) = I(x, y, t) + \frac{\partial I}{\partial x}\delta x + \frac{\partial I}{\partial y}\delta y + \frac{\partial I}{\partial t}\delta t + h.o.t.. \tag{2.2}$$

Replacing (2.1) in (2.2), we obtain

$$\frac{\partial I}{\partial x}\delta x + \frac{\partial I}{\partial y}\delta y + \frac{\partial I}{\partial t}\delta t = 0.$$

Dividing by the time displacement factor, it results the commonly used optical flow constraint:

$$\frac{\partial I}{\partial x}\upsilon_x + \frac{\partial I}{\partial y}\upsilon_y + \frac{\partial I}{\partial t} = 0. \tag{2.3}$$

where $\upsilon_x = \dfrac{\delta x}{\delta t}$ and $\upsilon_y = \dfrac{\delta y}{\delta t}$ are the optical flow components on the x and y directions, respectively.

Denoting $I_x = \dfrac{\partial I}{\partial x}$, $I_y = \dfrac{\partial I}{\partial y}$ and $I_t = \dfrac{\partial I}{\partial t}$, (2.3) may be briefly rewritten as

$$(I_x, I_y) \cdot (\upsilon_x, \upsilon_y) = -I_t.$$

Considering $(\dot{u}, \dot{v})^T$ as the optical flow and $I(u, v, t)$ the values for the levels of gray in an image that varies in time. Determining the optical flow implies solving the equation

$$I_x u + I_y v + I_t = 0 \tag{2.4}$$

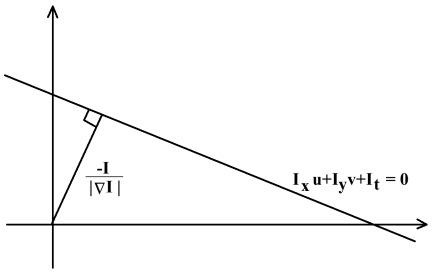

Figure 4: The normal flow.

by using the Lucas Kanade algorithm. We note that (2.4) can be rewritten as:

$$\nabla I^T \mathbf{f} = -I_t, \quad \text{where} \quad \mathbf{f} = \begin{bmatrix} u \\ v \end{bmatrix}, \quad \nabla I = \begin{bmatrix} I_x \\ I_y \end{bmatrix}.$$

When there is not enough information to determine the velocity in the optical flow, the normal velocity is computed; the tangential component v_t cannot be rebuilt (fig. 4). The Lucas Kanade algorithm involves solving the following equation system written in matrix form

$$G\mathbf{f} = b, \tag{2.5}$$

for

$$G = \sum_{\text{neighborhood}} \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}, \quad \text{and} \quad b = \sum_{\text{neighborhood}} \begin{bmatrix} I_x I_t \\ I_y I_t \end{bmatrix},$$

where

$$I_x(x,y) = \frac{I(x+1,y) - I(x-1,y)}{2}, \quad I_y(x,y) = \frac{I(x,y+1) - I(x,y-1)}{2}. \tag{2.6}$$

The equation system (2.5) allows solutions only in the following situations:

- G is invertible;
- G does not contain small entries (which are, generally, provided by the noise), i.e., the eigenvalues $\lambda_1,\ \lambda_2$ of G must not be too small;
- G must be conditioned, i.e., λ_1/λ_2 must not be too large (where λ_1 is the larger eigenvalue).

Then, the obtained optical flow is given by $\dot{f} = -G^{-1}b$.

2.3 Gateaux derivatives

We may notice from the relations (2.6) which describe the spatial derivatives, that the privileged directions are those given by the $\bar{i}$ and $\bar{j}$ unit vectors, and hence in a 3×3 neighborhood, the pixels in the corners do not contribute to the flow. To compensate

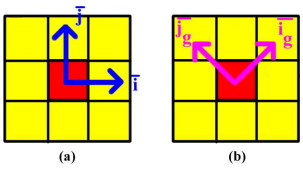

Figure 5: a) Standard directional derivatives; b) Slant directional derivatives.

this drawback, we shall make use of the slant Gateaux (directional) derivatives (see Fig. 5).

The basic discrete spatial derivatives in the rotated with 45 degrees coordinate system are:

$$I_{g_x}(x,y) = \frac{I(x+1,y-1)-I(x-1,y+1)}{2}, \qquad I_{g_y}(x,y) = \frac{I(x+1,y+1)-I(x-1,y-1)}{2}$$

In this case, we add to the spatial derivatives the projection of the slant derivatives to the Cartesian coordinate system:

$$I_x = I_{g_x}\sqrt{2} + \frac{I(x+1,y)-I(x-1,y)}{2}, \quad I_y = I_{g_y}\sqrt{2} + \frac{I(x,y+1)-I(x,y-1)}{2}.$$

By introducing the slant derivatives in the gradient computation significantly reduces the number of pixels that do not satisfy the conditions of existence for the solution.

2.4 Affine transformations

Affine transformations are used to estimate real movements of the object within the image, and also as a tool for indirectly reduce errors that may occur during optical flow computation. Combining the result obtained through this transformation with the optical flow we can emphasize the objects which are located in the robot's environment when it moves on a planar surface.

Let M be the transfer matrix from image I_t to image I_{t+1}, defined as

$$I_{t+1}(i,j) = MI_t(i,j).$$

We further assume that the movement in the image is small enough, which allows to estimate the transfer matrix M with an affine transformation. For each pixel (x,y) from the image at time t, we have the same pixel in the image $t+1$ located at the coordinates (x',y'). The relation between these two positions is given by

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix}.$$

In order to determine the affine coefficients we choose three pairs of pixels and we apply the RANSAC algorithm [4]. The optical flow associated to the affine transformation, named planar optical flow, is estimated to be:

$$\hat{f} = \begin{pmatrix} \hat{u} \\ \hat{v} \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix} - \begin{pmatrix} x \\ y \end{pmatrix},$$

where $\hat{f}$ is the planar optical flow at time t.

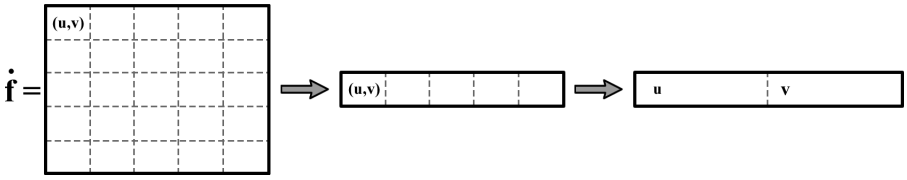

Figure 6: Transformation of optical flow to an input vector for ICA.

2.5 ICA

ICA is a statistical technique that separates the independent components of a given signal. In our case, the algorithm separates the optical flow in a linear combination of optical flow on the dominant area and the area with obstacles:

$$\ddot{f} = a_1 \dot{f}_{\text{background}} + a_2 \dot{f}_{\text{obstacle}}$$

To be able to apply the ICA algorithm to the optical flow, we have to rewrite the vector matrix as line vector. The vector field

$$\dot{f} = \begin{pmatrix} (\dot{u},\dot{v})_{11} & (\dot{u},\dot{v})_{12} & \dots & (\dot{u},\dot{v})_{1n} \\ (\dot{u},\dot{v})_{21} & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ (\dot{u},\dot{v})_{m1} & \dots & \dots & (\dot{u},\dot{v})_{mn} \end{pmatrix}$$

is rewritten as line vector

$$\begin{aligned} \dot{f} &\mapsto ((\dot{u},\dot{v})_{11}, (\dot{u},\dot{v})_{12}, \dots, (\dot{u},\dot{v})_{mn}) \\ &\mapsto (\dot{u}_{11}, \dot{u}_{12}, \dots, \dot{u}_{mn} \; ; \; \dot{v}_{11}, \dot{v}_{12}, \dots, \dot{v}_{mn}) \,. \end{aligned}$$

This vector represents the input for the ICA algorithm, which eventually allows to achieve the optical flow reconstruction. The transformations made on the optical flow matrix are shown in figure 6. Given the input flows $\dot{f}$ and $\hat{f}$, after applying the ICA algorithm, we obtain two optical flow fields

$$\begin{cases} \ddot{f} = a_1 \ddot{u}_\alpha + a_2 \ddot{v}_\beta \\ \tilde{f} = a_3 \tilde{u}_\alpha + a_4 \tilde{v}_\beta, \end{cases}$$

where a_1, a_2, a_3, a_4 are constant, and $\ddot{u}_\alpha$, $\ddot{v}_\beta$, $\tilde{u}_\alpha$, $\tilde{v}_\beta$ are the independent components of the flows at time t.

2.6 The decision criterion

Deciding that a pixel belongs to an obstacle or not will further rely on the dominant plane detection algorithm proposed by Naoya Ohnishi and Atsushi Imiya [14]. Due to the fact that the result from ICA does not emphasize the encountered obstacle, we focus on a decision to highlight the obstacles from the image, considering that the dominant plane and obstacle velocities are different. This criterion consists in comparing the variance values of norms from the components u_α and v_β.

For each pixel from the image, the following norms on an $h \times w$ neighborhood are defined:

$$l_{\alpha,\beta} = \{l_{i,j}\}_{i=1,j=1}^{h,w}, \quad \text{where} \quad l_{i,j} = |(u,v)_{i,j}|.$$

The variance is computed using

$$\sigma^2 = \frac{1}{hw} \sum_{i=1,j=1}^{h,w} (l_{ij} - \bar{l})^2, \quad \text{where} \quad \bar{l} = \frac{1}{hw} \sum_{i=1,j=1}^{h,w} l_{ij}.$$

We use the fact that the variance of the output signal corresponding to the obstacle is greater than the variance corresponding to dominant plane.

We build a matrix of norms L, $dim(L_{ij}) = hw$, whose components are defined as:

$$L_{ij} = \begin{cases} l_\alpha & \text{if } \sigma_\alpha^2 > \sigma_\beta^2 \\ l_\beta & \text{otherwise.} \end{cases}$$

In order to use that the dominant plane covers most of the surface of the image, two important aids are the matrix of distances between L_{ij} and the shift towards the median value of the background,

$$\{d_{ij}\}_{i=1,j=1}^{h,w}, \quad d_{ij} = |L_{ij} - median(L_{ij})|.$$

The coordinates where the values d_{ij} are close to zero correspond to the pixels of the dominant plane. The other areas are obstacles lying on the path of the robot. To visualize these objects, the values d_{ij} are normalized in the interval 0-255, and further represented as a monochrome image containing the dominant plane in contrast with the encountered obstacles. This image is explicitly defined as:

$$D(i,j) = \frac{d_{ij} \cdot 255}{\max d_{ij}}.$$

3 Results and conclusions

The image sequence was captured using a high speed camera, at a resolution of 640×480 pixels and capturing speed of 1000 fps. To analyze the captured images, the Lucas Kanade algorithm was used [10], enhanced by considering slant Gateaux derivatives in order to process much more information. The procedure is widely presented in section 2.2. In figure 7 it is shown an example of optical flow object detection which results from the analyzed image sequence. Using the affine transformation we estimated the background shift (fig. 8). The output data of the two algorithms mentioned before, become input signals for ICA. By means of the FastICA version of the ICA algorithm, we decompose the input signals into independent components (background and moving object). Further, using the decision criterion presented in section 2.5 we highlight the objects met by the robot.

In figure 9 are shown obstacles detected by the classic Lucas Kanade algorithm. We notice that, at this processing stage, the noise level is high and the obstacle is hard to be identified. In order to reduce the noise and to increase the precision, the slant Gateaux derivatives were used. Besides the classic version, where spatial derivatives

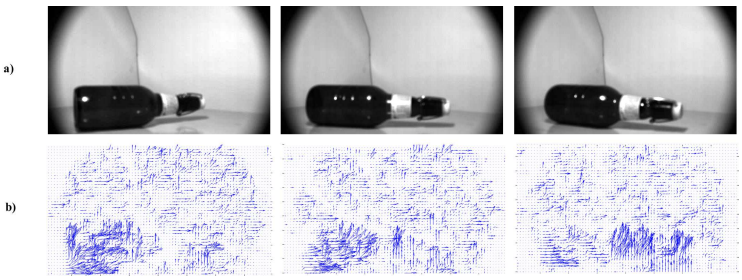

Figure 7: a) Captured images; b) Optical flow through classic Lucas Kanade approach.

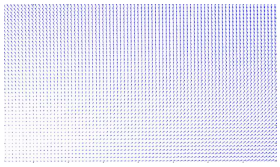

Figure 8: Affine motion.

cover only pixels on vertical and horizontal directions, the slant Gateaux derivatives bring information from the diagonal directions from the neighborhood. The difference between these two approaches can be seen in figure 10, and the improvements consist in reducing the noise and increasing the object detection precision (fig 11). The output of the Lucas Kanade algorithm still exhibits a significant amount of noise. This leads to robot orientation errors because it may interpret the noise as being an obstacle. Thus, the robot cannot generate an optimal trajectory in order to avoid the obstacle. Although the introduction of slant directional derivatives brings improvements, we notice that applying the algorithm on multiple frames decreases the noise. However, in exchange, it induces an error at the objects contour layer. As it is shown in figure 12 the obstacle appears in the image in two overlapped instances with a small offset between them. While using multiple frames, the spatial derivatives remain the same, and the only modification appears at the temporal derivatives which change to

$$I_t = \frac{1}{n}\sum_{k=0}^{n-1} I(x, y, y + k + 1) - I(x, y, t + k).$$

Consequently, by applying the algorithm, we achieved the following (fig. 13a):

Figure 9: Detected object using classic Lucas Kanade optical flow.

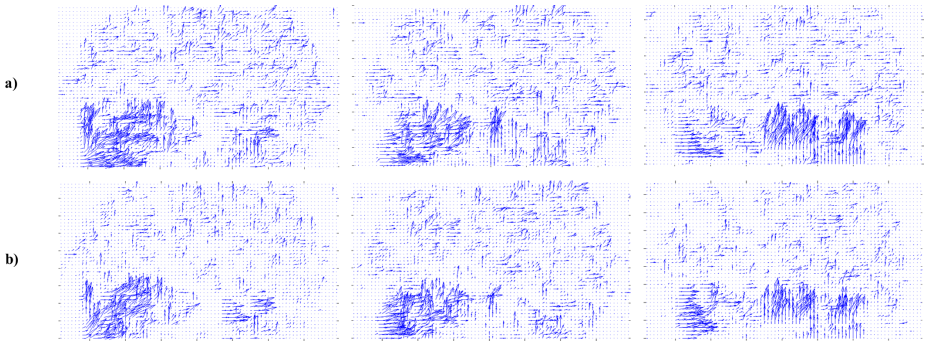

Figure 10: a) Lucas Kanade optical flow; b) Lucas Kanade optical flow with additional slant derivatives.

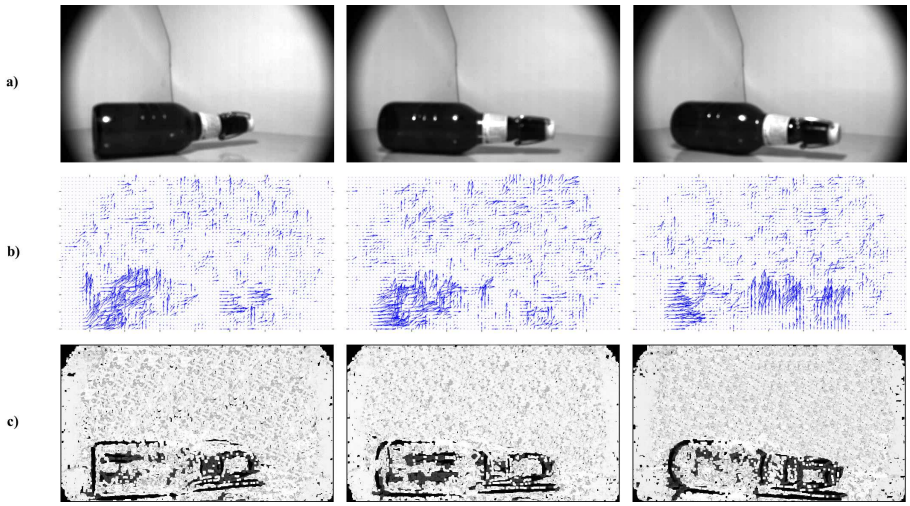

Figure 11: a) Camera-provided sequence of frames; b) Lucas Kanade optical flow with additional slant derivatives; c) Detected object.

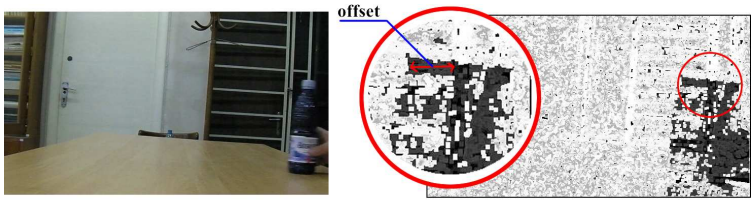

Figure 12: a) Captured images; b) Detected object through Lucas Kanade optical flow using slant Gateaux derivatives and temporal derivative on 4 frames with contour layer offset error.

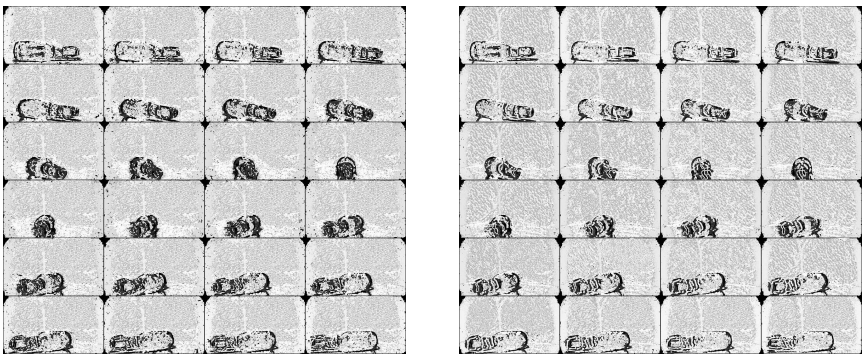

Figure 13: Results obtained using the proposed algorithm: a) applied on two subsequent frames; b) applied on four subsequent frames.

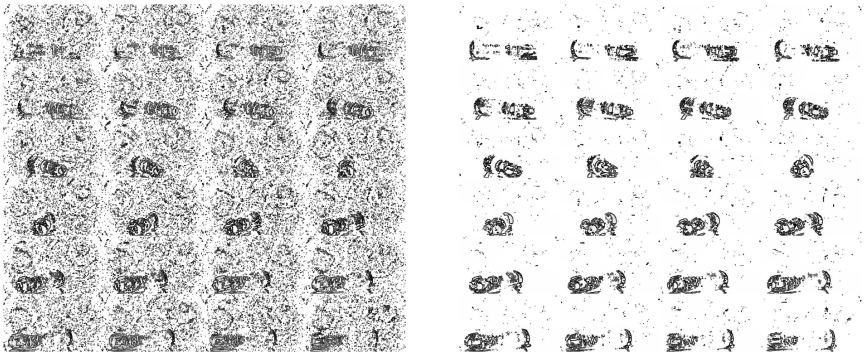

Figure 14: Results obtained using classic Luckas Kanade algorithm: a) applied on two subsequent frames; b) applied on four subsequent frames.

- the algorithm applied on pairs of two frames:
  - detects any movement in the frame;
  - the edges are emphasized;
  - the moving object is pointed out relative to the dominant plane;
  - noise is reduced and precision is increased;
- the algorithm applied on multiple frames (fig. 13b) exhibits, besides the characteristics from above, the feature that the noise is significantly reduced, with the cost of decreasing the precision.

Compared to the classical approach, our method provides better results. To illustrate this, on two consecutive frames (see figs 13a and 14a) the classical approach leads to results which exhibit strong noise, which leads to difficulties in recognizing the object. Under identical conditions, our method significantly reduces the noise, providing a clearer image. On four consecutive frames (see figs 13b and 14b) our proposed method offers a better object contour, while the classical approach reduces noise and leads to a more fragmented object contour. The noise which appears on the background in our method is less intense compared with the contour, leading to a clearly visible object.

	Optical flow with Lucas Kanade classical approach		Optical flow with directional derivatives approach	
No. of frames	2	4	2	4
Affine transformation	0.051526	0.056226	0.052622	0.055933
Lucas Kanade	10.250007	10.277283	11.388679	11.629944
ICA	0.207987	0.200054	0.208476	0.419176
Decision	22.458741	22.874521	22.604373	22.808103
Total	32.968.261	33.368.084	41.254.150	59.283.172

Table 1. Approximative execution time in seconds for one processing cycle.

From table 1 we can notice that the execution time is independent from the number of images in the analyzed sequence. Using the slant Gateaux derivatives increases the execution time but the difference is small compared with the image quality obtained as algorithm output. In terms of execution time, the proposed method is slower, but in terms of image quality, the results are significantly improved. Even if the average execution time is a little bit higher, it is worth to apply the improved algorithm because the detected obstacle is much more visible.

The further concerns of the present research aim to increase the object detection precision, still maintaining the quality of the final image. Also, further research focuses on applying the algorithm on images captured using a stereoscopic algorithm which induces directly the depth information. This allows the differentiation between two or more objects that may appear in the visual field of the robot. Therefore an ordering of the objects in the image can be achieved, from the point of view of the distance to the robot without introducing a distance sensor which may be expensive or difficult to implement. In this respect, a drawback which has to be overcome is the fact that the robot interprets multiple objects as one single object located at a common certain distance from the camera.

Acknowledgements. The authors would like to thank Professor Mona Mihăilescu from the Digital Holography Laboratory of the *Politehnica* University of Bucharest for the provided support and expertise, regarding the handling of hardware for capturing fast-speed image sequences.

References

[1] J.Y. Bouguet, *Pyramidal Implementation of the Lucas Kanade Feature Tracker, Description of the algorithm*, Intel Corporation, Micropocessor Research Labs., Key: citeulike: 1987163.

[2] J. Choi, M. Choi, W.K. Chung, *Topological localization with kidnap recovery using sonar grid map matching in a home environment*, Robotics and Computer-Integrated Manufacturing, 28, 3 (2012), 366-374.

[3] G. Cielniak, T. Duckett, A.J. Lilienthal, *Data association and occlusion handling for vision-based people tracking by mobile robots*, Robotics and Autonomous Systems, 58, 5 (2010), 435-443.

[4] M.A. Fischler, R.C. Bolles, *Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography*, Commun. Assoc. Comp. Mach. 24, 6 (1981), 381-395.

[5] S. Guadarrama, A. Ruiz-Mayor, *Approximate robotic mapping from sonar data by modeling perceptions with antonyms*, Info. Sci. 180, 21 (2010), 4164-4188.

[6] B.K.P. Horn, B.G. Schunk, *Determining optical flow*, Artificial Intelligence, 17 (1981), 185-204.

[7] A. Hyvarinen, E. Oja, *Independent component analysis: algorithms and applications*, Neural Networks 13 (2000), 411-430.

[8] I. Ishii, T. Taniguchi, K. Yamamoto and T. Takaki, *1000 fps Real Time Optical Flow Detection System*, Image processing: Machine Vision Applications III, Proc. SPIE 7538, 75380M (2010), doi: 10.1117/12.838676.

[9] J. Li, S.G. Nikolov, N. E. Scott-Samuel, C. P. Benton, *Reliable real-time optical flow estimation for surveillance applications*, Crime & Security (2006), 402-407.

[10] B.D. Lucas, T. Kanade, *An iterative image registration technique with an application to stereo vision*, Proc. 1981 DARPA Image Understanding Workshop, 1981, 121-130.

[11] Y. Matsushita, J. Miura, *On-line road boundary modeling with multiple sensory features, flexible road model, and particle filter*, Robotics and Autonomous Systems, Special Issue ECMR 2009, 59, 5 (2011), 274-284.

[12] O. Naoya, I. Atsushi, *Dominant plane detection using optical flow and independent component analysis*, MLDM 2005, LNAI 3587, Springer, 2005, 497-506.

[13] N. Ohnishi, A Imiya, *Independent component analysis of optical flow for robot navigation*, Neurocomputing 71 (2008), 2140-2163.

[14] N. Ohnishi, A. Imiya, *Dominant plane detection using optical flow and independent component analysis*, BVAI 2005, LNCS 3704, Springer, 2005, 487-496.

[15] Z. Silar, *Railway crossing monitoring by camera using optical flow estimation*, Electronical Technical Journal of Technology, Engineering and Logistic in Transport, 3, 4 (2010), 272-279.

[16] H. Teimoori, A.V. Savkin, *Equiangular navigation next term and guidance of a wheeled mobile previous termrobotnext term based on previous termrangenext term-only measurements*, Selected papers from the 2007 European Conference on Mobile Robots (ECMR '07), Robotics and Auton. Syst. 58, 2 (2010), 203-215.

[17] W. Yaonana, Y. Yimina, Y. Xiaofanga, Z. Yia, Z. Yuanlia, Y. Fenga, T. Leia, *Autonomous mobile robot navigation system designed in dynamic environment based on transferable belief model*, Measurement 44 (2011), 1389-1405.

Authors' addresses:

Stefan Dumitru and Danut Bucur
Department of Mechatronics, Institute of Solid Mechanics of the Romanian Academy,
15 Str. C-tin Mille, Bucharest, 10141, Romania.
E-mail: dumitru_sa@yahoo.com , xdan.bucur@yahoo.com

Vladimir Balan,
Univ. Politehnica of Bucharest, Fac. of Applied Sciences, Dept. of Math.-Informatics,
Splaiul Independentei 313, RO-060042, Bucharest, Romania.
e-mail: vladimir.balan@upb.ro , vbalan@mathem.pub.ro