Vision System for Relative Motion Estimation from Optical Flow

Sergey M. Sokolov, Andrey A. Boguslavsky, Felix A. Kuftin
Keldysh Institute for Applied Mathematics RAS
Moscow, Russia

ABSTRACT

For the recent years there was an increasing interest in different methods of motion analysis based on visual data acquisition. Vision systems, intended to obtain quantitative data regarding motion in real time are especially in demand. This paper talks about the vision systems that allow the receipt of information on relative object motion in real time. It is shown, that the algorithms solving a wide range of practical problems by definition of relative movement can be generated on the basis of the known algorithms of an optical flow calculation. One of the system's goals is the creation of economically efficient intellectual sensor prototype in order to estimate relative objects motion based on optic flow. The results of the experiments with a prototype system model are shown.

This research was supported in part by the grant of RFBR № 08-01-00908.

Keywords: vision system, optical flow, relative motion, open libraries, component approach, contactless odometer.

1. INTRODUCTION

Works in the field of an optical flow calculation have been conducted for more than 30 years. Last decade these methods were used in a wide range of applied problems due to increase of computers computing capacity and the occurrence of specialized graphic processors. There are many articles that have been written on the subject of optical flow methods [1-17]. There are also widely available libraries with an open code, in which the ready applications of the most popular optical flow methods could be found (for example, OpenCV [18], LTI-Lib [19], VXL [20]). Methods of an optical flow appear to be useful for segmentation of images [15], and also for detection of obstacles from moving objects [3].

Despite the growing efficiency of computers, it is prudent to see how the optical flow method can be widely used in different applications but with minimal computing expenses and sustained data accuracy and calculating stability. In order to reduce computing expenses restrictions to the way optical flow calculated and processed may be imposed.
Important questions to be answered are:

- Accuracy of calculations at a low image resolution;
- Selection of an optical flow method for specific targets regarding the optimal parameters values;
- Algorithms' construction where parameters can be adjusted depending on changing conditions.

The work paper set out to address the following tasks:

- Calculation and interpretation of an optical flow using Lucas-Kanade method on pyramids of images given plain - parallel camera movement as well as movement under an angle;
- Determination of how calculation accuracy depended on resulting values of algorithm parameters in various applications;
- Identification of how all algorithms functioned together when a single task was performed in real time.

To tackle the above, available in the KIAM RAS laboratory template of the vision system (VS) program VS software [21] and a custom- made hardware-software complex for studying the parameters of Lucas-Kanade optical flow on pyramids of images and feature points allocation methods were used. VS software is a modular, multi-channel application which allows quickly processing various visual data algorithms and most importantly working with the algorithms of Lucas-Kanade optical flow as well as testing its adjustable parameters.
The offered solutions have been practically tested using (see item 3) a prototype of VS model: the VS for relative movement estimation on the optical flow, created in KIAM VS laboratory.

2. ARCHITECTURE of VISION SYSTEM FOR RELATIVE MOTION ESTIMATION FROM OPTICAL FLOW

During VS creation the following two basic methods were used in order to determine relative movement on an optical flow:

- COTS technology in configuration of hardware and software parts of the system;
- component approach to real time VS design.

The VS system was also formed as a component of information system of mobile devices, inside of which there was programmed an ability to integrate with local and global navigation.

Software
A sparsed optical flow method has been used to achieve maximum efficiency and to decrease computing expenses. The optical flow is not applied throughout the picture, but only at feature points [5, 6, 7, 13].
Any algorithm based on a sparsed optical flow method entails three stages:

- identification of feature points of the image;
- definition of the vectors where feature points are displaced;
- segmentation of the resulting vector field and its interpretation.

These stages would be called processing levels of initial visual data. Besides these traditional operations, the following algorithms were added: algorithms that allow selecting areas of interest in a video camera's view, algorithms that can statistically calculate statistical vectors' displacement and finally, algorithms that can automatically change the parameters of visual data inputs.

Low level algorithms

The functions from OpenCV library [18] were used to completely implement the low level of video data processing. An approach identifying feature points was selected in order to keep the system working in real time and also to be able to address any new issues at a low level from the known optical flow methods [11, 14]. There is a possibility of effective processing of image points by regions of interest. These regions are set by special masks (based on priority of information or as a result of analyzing images from previous video sequence) which are formed by algorithms of top level.

Feature points allocation. Let each point of the image be characterized by the function of intensity *I (x, y)*. For the further combination angular points are then selected as feature points. Harris's [6] modified algorithm which reacts to angles is applied to their allocation. Angle in this case is a local distinctive area (location) of the image where the change of intensity function *I* is maximized simultaneously in both directions *x* and *y.* An equation for Harris's detector can be written as [6]:

$$\begin{aligned} R_I(x,y) &= \det(G(x,y)) - k(tr(G(x,y)))^2 \\ &= \lambda_1\lambda_2 - k(\lambda_1+\lambda_2)^2 \end{aligned} \qquad (1)$$

Where *G* – covariance matrix of derivatives function *I(x,y)* (Hessian's functions I of second order):

$$G = \begin{vmatrix} \left(\frac{\partial I}{\partial x}\right)^2 & \frac{\partial I}{\partial x}\frac{\partial I}{\partial y} \\ \frac{\partial I}{\partial x}\frac{\partial I}{\partial y} & \left(\frac{\partial I}{\partial y}\right)^2 \end{vmatrix} \qquad (2)$$

λ_i – Hessian own values, and *k* - empirical value is usually taken out from the interval [0.04, 0.06]. R_I – is called an angle sensitivity function. If its value is negative, then the found location is an edge; if its value is higher than the positively set threshold, then the location is an angle. Locations with positive values R_I lower than the threshold are considered monotonous. Parameter *k* sets operator's sensitivity in that higher the parameter value, the fewer number of angles will be found. Derivatives (according to numerical methods) are in a vicinity of points; therefore high-frequency filtration has already been embedded in the algorithm. Local function maxima are picked out because several neighboring angle points yield maximum values for the angle sensitivity function R_I.

The following algorithm is used to identify feature points:

$$I(x,y) - \begin{cases} \text{feature point}, & \min(\lambda_1,\lambda_2) > \lambda_{\max}\cdot q \\ \text{uniform area}, & \min(\lambda_1,\lambda_2) \le \lambda_{\max}\cdot q \end{cases}$$

Where λ_1, λ_2 - own values of matrix *G* in a considered point, *q* - the parameter setting the quality of feature point (an angle sharpness), $\lambda_{max}=max\ (min\ (\lambda_1^0, \lambda_2^0),\ min\ (\lambda_1^1, \lambda_2^1),\ \ldots,\ min\ (\lambda_1^{sx\cdot sy}, \lambda_2^{sx\cdot sy}))$, *sx, sy* - the size of region of interest (ROI) in which feature points are searched, λ_1^i, λ_2^i - matrix's own values *G* in each point of the image. As in Harris's general detector, several neighboring points of feature location will have $min\ (\lambda_1, \lambda_2) > \lambda_{max}\cdot q$. This is why local maxima of own minimum values as the feature point are chosen. In case it becomes necessary to increase the accuracy of the feature point's location determination, the algorithm's solution is interpolated.

Average level algorithms

Definition of the vectors where feature points are displaced. From now the definition of Gauss pyramid of images will be used. Gauss pyramid of images is a number of images with resolution consistently decreasing by 2. The initial image lays in the pyramid basis. The operation of images' combination calculates vectors fields that translate feature points of the first image into those of the second image. When optical flow is calculated in a traditional way, it would be ideal to compare all points of the image. However, not all points are unique, not all of them are feature points that could be exactly applied to the points from the second image, for example, points found in monotonous areas where brightness of image is the same. This is the reason why optical flow algorithm to such points is not applicable. For image combination, i.e. displacement vector identification, Lucas-Kanade method is used, where minimization condition states that vector displacement:

$$\vec{v}_{opt} \approx G^{-1}\vec{b} \qquad (3)$$

The formula (3), the basic formula of Lucas-Kanade optical flow, states that the vector which fully correlates a point of the first image to that of the second could be found with a margin error. To reduce the error the given method is applied iteratively, i.e. the found vector becomes an input parameter into the algorithm to produce a new more exact vector. The process is repeated until the desired accuracy lever or number of iterations is achieved. Optical flow methods have an essential weakness: they can be applied at small (1-3 pixels) displacement of objects. In order for the algorithm to work with larger displacements, it is applied to the Gauss pyramid of the initial image. First, the vectors at the top level of the pyramid are calculated; the process is repeated until the margin of error sufficient for the application at lower level is reached. These steps are performed for all levels. A vector of an optical flow is resulted at the final stage. This algorithm even given all its advantages has an essential drawback: small errors in calculations at top levels of the pyramid tend to accumulate and increase.

In the OpenCV implementation depending on flags, the pyramid of images can be constructed in advance, or the function of finding an optical flow would call it before the calculation begins. The size of a pyramid is chosen based on a rough estimate of visible plain-parallel displacement of an image (or from the prior information, or from a previously calculated vector). Two times as many displacements are found with every new level of a pyramid.

If the feature point of the first image appears closed on the second or falls out of the image area, there are two approaches to address this situation: either such point is marked as the one for which a conformity is not found or the algorithm would substitute the point with another with similar features to yield a false vector i.e. an optic flow vector that does not correspond to the true objects' movement on the scene. False vectors would be filtered at the next stage of the algorithm.

False vectors filtration. Depending on the application, vectors' filtration may be more challenging than the actual determination of optical flow. Undoubtedly, accuracy and stability of the solution for a specific task is depended on optical flow vectors' segmentation. Also at that stage an image comparisons in a camera objective (the so-called visible image) to the real movement of the camera (objects relevant to the camera) is made; and the resulting data would serve for subsequent calibration of algorithms.

At a described stage of research, the relative movements, in which the visible movement was either plain-parallel or the movement of camera's sensor plain under a constant angle to the surface's plain, were considered.

Finding an average vector for an optical flow of such camera movement is not that difficult. It is the vector whose coordinates equal to the sum of corresponding coordinates of all vectors divided by the number of vectors (given corrections of camera movements under an angle).

The filtration of false vectors of optical flow is better conducted based on direction or length, keeping those, whose directions or lengths lay within the acceptable for the average vector's margin of error. The resulting filtration data could be used iteratively: first, find an average vector, then, reject vectors considerably deviating from the average vector's directions or lengths, finally, find a new average vector, reject, and reduce the error. Thus, the accuracy of average vector's calculation as it relates to the plain-parallel movement rises. Since restrictions to the camera movement are imposed a priory, comparing an average vector to the real world metric, movement parameters could be obtained.

Top-level algorithms

Top-level algorithms set out the functionality for the system itself beginning from the visual data collection and ending with the results about a certain relative movement.

The initial a priori data for the algorithms are

- location and characteristics of attainable visual fields (resolution, zoom, responsiveness of the visual data channels);
- location specifics and images' contours of the observed objects (in order to form masks and necessary resolution for the level of Gauss pyramid;
- possibility to obtain additional data about observed objects.

The target task of top-level algorithms, the relative motion estimation, is divided into several subtasks::

- selection of the relative motion model;
- the model verification/specification;
- calculation of relative motion quantitative characteristics.

Under the fixed functioning conditions the part of these subtasks can be passed.

For example, in a mode of functioning as contactless odometer, it is possible to consider, that the scene scale is fixed, and the motion model is limited by the kinematic scheme of a vehicle or the robot. Then the decision of the first subtask is replaced with use of an aprioristic set of motion models. Similar simplification is applicable and in case of estimation of the inspection tool movement under a processed surface by mechanical machining.

Mathematical models describe a priori certain type of relative motion and can change operatively on signals from the outside or by results of the current visual scene analysis. Besides, these models can consider the model of the vehicle on which the acquisition device is mounted. For example, it can be a model of a car suspension, allowing to correct the image scale on the next analyzed video sequence image.

Top-level algorithms are developed with the original approach to the images analysis on the basis of a combination of "top-bottom" and "bottom-top" methods [21, 22].

The hardware

The structure of the system is modular and open:

- One or several acquisition devices (a video camera with optical system) + the input channel for the image transfer into the computing - control device.
- The computing - control device (universal computer or specialized computing device).
- Software (mathematical models; top-, middle- and low-level algorithms).

The VS is composed of the following items. Different types of videocameras with various lenses were used as acquisition devices, as well as inexpensive analogue videocameras with framegrabber or TV-tuner, digital Web-cameras and more expensive Ethernet cameras such as those with progressive scan. Notebook and barebones systems were used for computing and control.

The VS architecture has been tested on the contactless odometer prototype model (fig. 1).

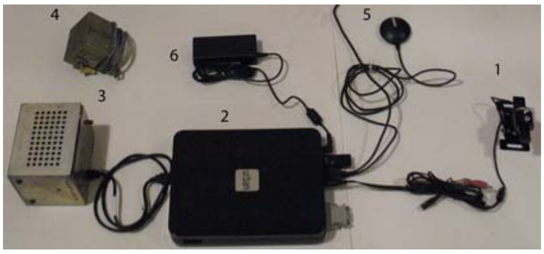

Fig. 1. General view of prototype model components of the navigating system including contactless odometer. 1 – the videocamera on fixing platform; 2 – the computing and control unit; 3 – the three-dimensional accelerometer; 4 – the single-dimensional accelerometer; 5 – the GPS-receiver antenna; 6 – the power converter block.

3. EXPERIMENTS WITH A SYSTEM PROTOTYPE MODEL

Mobile robots

Results of research of VS for contactless estimation of movements are tested for improvement of an information support of two mobile robots: MRK-27 and "Trikol".

Remotely controlled robot MRK-27 (CDTB PR Bauman's Moscow State Technical University) (fig. 2) – the small-serial robot on the track-type chassis. It is actively used in many applied tasks.

In experiments with this robot the field of view «under feet» has been used. It has provided exact and fast estimation of robot movement. The results data can be used in a robot control contour. Robot movements occurred on smooth concrete floor with speed of about 1 m/s. Movement trajectories were rectilinear and curvilinear. Distances of 10-30 m have been fixed by contactless odometer to within 0,01 m (fig. 3).

Fig. 2. The registering block of the contactless odometer is mounted in a MRK-27 grasping unit.

On fig.3 the example of the displacement vectors detection for concrete floor scenes is shown. Frequency of the displacement vectors generation not less than 10 Hz.

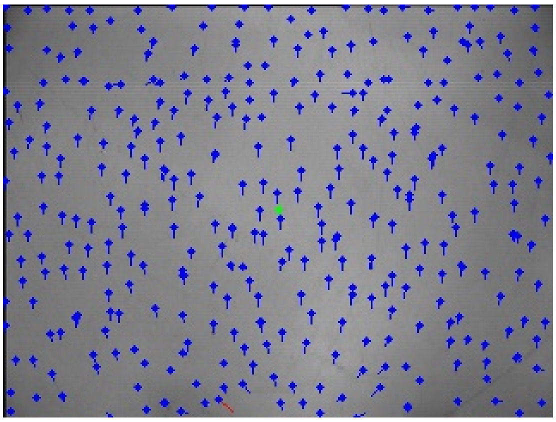

Fig. 3. Displacement vectors estimated by VS prototype model at MRK-27 robot movement on a smooth concrete floor.

“Trikol” is a laboratory maneuverable three-wheeled robot with the high autonomy capability (fig. 4).

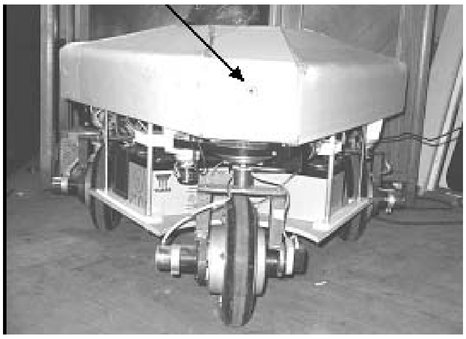

Fig. 4. “Trikol” robot general view. The arrow specifies a placement of the forward looking videocamera. Its field of view is used for contactless movement estimation.

Fig. 5. Movement vectors estimated at the “Trikol” forward looking videocamera field of view.

Rail transport

In experiments, the fields of views of the computer vision systems intended for inspection and measurements of parameters of various railway infrastructure objects have been used (fig.6-9). In these fields of view the regions of interest were automatically allocated. In these regions estimation of movement of a rolling vehicles relating a railway was made. So, to calculate the way based on visual data from a field of view intended for the control of a ballast section (fig.8), rails images were extracted, and relevant regions for an optical flow calculation and for the movement vectors estimation were located.

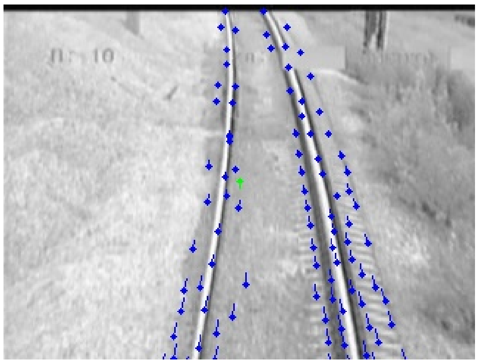

Fig. 6. Movement vectors estimation based on areas of a railroad ties lattice along rail threads in a field of view intended for the control of a ballast section (speed of movement of 15-20 m/s).

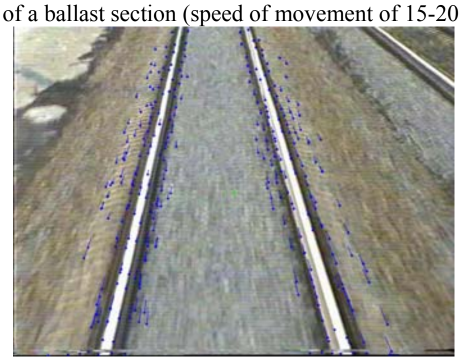

Fig. 7. Movement vectors estimation based on areas of a railroad ties lattice along rail threads in a field of view intended for the control of a ballast section (speed of movement of 20-25 m/s).

In another system intended for the rail ties surface and rail joints control, in the field of view search of railroad ties at first was performed. Then, relative to them regions of interest were positioned. Finally, the calculation of an optical flow and estimation of movement vectors was made.

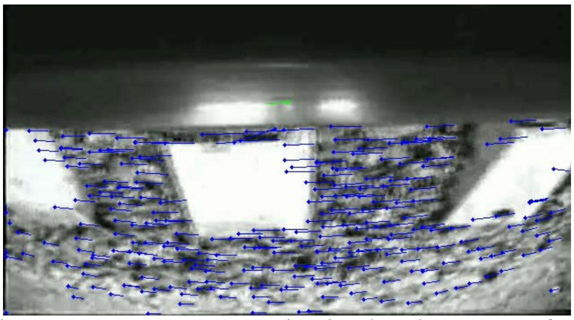

Fig. 8. Movement vectors estimation based on areas of a railroad ties lattice along rail threads in a field of view intended for the rail joints control (speed: 15-20 m/s).

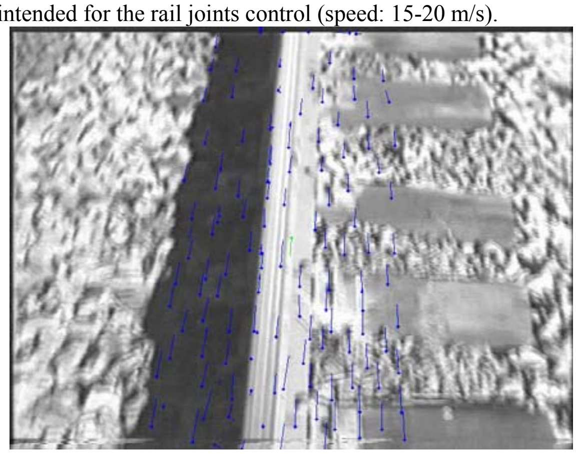

Fig. 9. Movement vectors estimation based on areas of a railroad ties lattice along rail threads in a field of view intended for the railroad ties control (speed: 15-20 m/s).

In all listed systems the specified restrictions on the speed of movement were defined by that fact that cameras intended for inspection of relatively large object rather than special video cameras were used. These cameras correspond to the TV standard. The use in the same configuration of progressive video cameras and the selection of local regions of interest allows to arrive at the solution of the movement estimation problem for a rolling stock with the speed up to 250 km/hour and accuracy of 0,02% from a way (by visual data from a forward looking field of view) and to 120 km/hour and accuracy 0,01 % from a way by visual data from an «under feet looking» field of view.

Motor transport

The contactless odometer problem is present for various mobile laboratories, for which the exact binding of measurements is important. The described VS has been tested as a part of mobile laboratory for operative diagnostics of a road surface. In this laboratory there are several video cameras for the various parameters of a road surface control. For experiments on contactless odometer the field of view of the forward looking camera has been chosen (fig. 10). On fig. 12 the closed trajectory of the mobile laboratories movement is presented. This trajectory restored according to contactless odometer measurements.

Fig. 10. The general view of Mobile Laboratory for Operative Diagnostics of a road surface (MLOD).

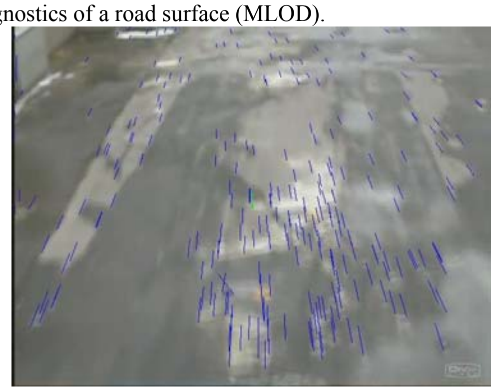

Fig. 11. Movement vectors estimated at the MLOD mobile laboratory movement on wet asphalt road with the 2-9 m/s speed.

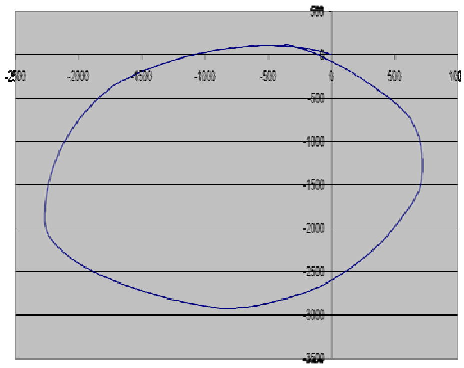

Fig. 12. A trajectory of the MLOD laboratory movement. It is constructed according to contactless odometer measurements based on results of a circular route tour. At a circular route of 195 m length, an error of moving estimation has made 0,5 m.

Machining process

The real example of the demand for the contactless relative movement's estimation of other scale is measuring the movement of the inspection tool at the time of operative quality control of a processed surface during mechanical machining.

In this task the measuring head moves at a 10 mm distance from the processed surface. Moving speed is from 5 to 30 cm/minute. The size of a controllable field of view is 10x10 mm. In the same area, the horizontal rectangular zones along head movement are selected. The received visual data along with the main data check, the movement of the measuring head is determine as well as the binding of measurements to a place on a surface of a controllable detail. The capturing of TV standard frames for the displacement of an observable surface from a frame to a frame makes up from 2 to 15 pixels. The described VS yielded positive results.

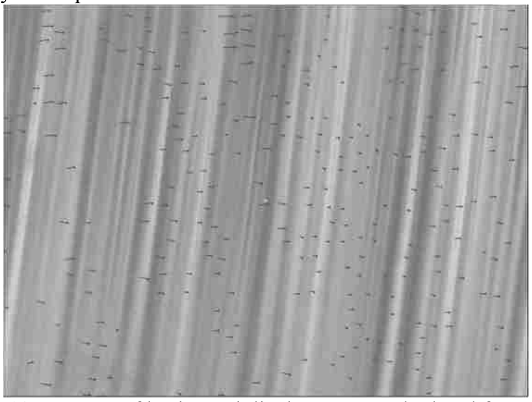

Fig. 13. Vectors of horizontal displacement, calculated from the field of view of a measuring head moving under processed surface.

4. CONCLUSION

In this work paper it has been shown through a number of experiments that the rational use of well-known low-level optical flow algorithms could solve a wide range of tasks where an estimate of parameters of relative objects movement is needed .

Thus, the following issues with the optical flow analysis have been addressed:

- big volume of processed data;
- texture variability (structure of a underlying surface);
- the errors in feature points correspondences.

The usage of previously used hardware and software makes the offered decisions more economically attractive.

The cost of equipment used in the majority of considered configuration is in range 1500-2000$. For variants automobile and railway contactless odometers, capable to work on speeds more than 60 km per hour this cost increases approximately in 2 times (for the account of high-speed videocameras with progressive scan).

Designing a special low-level software package would reduce the optical flow computing costs and provide optimization for the overall performance of the system of relative objects movement estimation based on an optical flow.

The following steps are considered to be appropriate to address the described approach:

- based on optical flow estimation of complex rotary movements of mobile objects (roll, pitch, yaw);
- fusion and sharing of data on relative movement from the optical flow channel with other local navigation systems, such as the vertical sensor and accelerometer-based local navigating system;
- development of methods for adaptive adjustment in uncontrolled light conditions (in considerable limits).

5. REFERENCES

[1] A. Akbarzadch et al., "Towards urban 3D reconstruction from video", **Third International Symposium on 3D Data Processing, Visualization and Transmission**, June 2006.

[2]C. Fermuller, D. Shulman, and Y. Aloimonos, “The Statistics of Optical Flow”, **Computer Vision and Image Understanding** 82, pages 1–32, 2001

[3] C. Braillon, C. Pradalier, J. Crowley, C. Laugier, "Real-time moving obstacle detection using optical flow models", **Proc. of the IEEE Intelligent Vehicle Symp**., 2006, pp. 466-471.

[4] A. Dev, B.J.A Krose, F.C.A. Green, "Heading Direction for a Mobile Robot from Optical Flow", **Proc. of the IEEE International Conference on Robotics & Automation**, Leuven, Belgium, May 1998.

[5] P.H.S. Torr, A Zisserman, "Feature-Based Method for Structure and Motion Estimation", **Vision Algorithms: Theory and Practice**, 2000, pp 278-294.

[6] C. Harris, M. Stephens, "A Combined Corner And Edge Detector", **Proceedings of the 4th Alvey Vision Conference**, 1988, pp 147-151.

[7] J. Weijer, "Robust Optical Flow from Photometric Invariants", **ICIP**, Vol. 3, 2004, pp.1835-1838.

[8 B.K.P. Horn, B.G. Schunk, "Determining Optical Flow", **Artificial Intelligence**, Vol. 2, 1981, pp 185-203.

[9] J.L. Barron, D.J. Fleet, S.S. Beauchemin, T.A. Burkitt, "Performance of Optical Flow Techniques", **Computer Society Conference on Computer Vision and Pattern Recognition**, 1992, pp. 236-242.

[10] T. Brox, A. Bruhn, N. Papenberg, J. Weickert, "High Accuracy Optical Flow Estimation Based on a Theory for Warping", **Proc. 8th European Conference on Computer Vision**, Springer LNCS 3024, vol. 4, 2004, pp. 25-36.

[11] S. Baker, I. Matthews, "Lucas-Kanade 20 Years On: A Unifying Framework", **IJCV**, Vol. 56, No. 3, March 2004, pp. 221-255.

[12] J.L. Barron, N.A. Thacker, "Tutorial: Computing 2D and 3D Optical Flow", **Tina Memo No. 2004-012**, 2005.

[13] J. Shi, C. Tomasi, "Good features to track", **Proc. IEEE Comput. Soc. Conf. Comput. Vision and Pattern Recogn.**, 1994, pp. 593-600.

[14] J.Y. Bouguet, "Pyramidal Implementation of the Lucas Kanade Feature Tracker", **Intel Corporation, Microprocessor Research Labs (2000)**, http://www.intel.com/research/mrl/research/opencv/

[15] M. Zucchelli, J. Santos-Victor, H. Christensen, "Multiple Plane Segmentation Using Optical Flow", **ISR Conference**, 2002.

[16] J. Campbell, R. Sukthankar, I. Nourbakhsh, "Visual Odometry Using Commodity Optical Flow", **AAAI**, 2004.

[17] J. van de Weijer, Th. Gevers "Robust Optical Flow from Photometric Invariants"”, **ICIP**, Vol. 3, 2004, pp. 1835-1838.

[18]http://www.intel.com/technology/computing/opencv/index.htm – OpenCV support site.

[19] http://ltilib.sourceforge.net/ – LTI-Lib support site.

[20] http://vxl.sourceforge.net/ – VXL support site.

[21] A.A. Boguslavsky, S.M. Sokolov, "Component Approach To The Applied Visual System Software Development", **7th World Multiconference on Systemics, Cybernetics and Informatics (SCI 2003)**, July 27-30, Orlando, Florida, USA, 2003.

[22] A.A. Boguslavsky, V.V. Sazonov, S.M. Sokolov, A.I. Smirnov, K.U. Saigiraev, "Automatic Vision-based Monitoring of the Spacecraft Docking Approach with the International Space Station", **Proc of the First International Conference on Informatics in Control, Automation and Robotics**, Setúbal, Portugal , Vol. 2, 2004, pp. 79-86.