Monocular Visual Odometry and 3D Reconstruction

Alexandr Prozorov, Vladimir Volokhov, Andrew Priorov
P.G. Demidov Yaroslavl State University
Yaroslavl, Russia
alexprozoroff@gmail.com, volokhov@piclab.ru, andcat@yandex.ru

***Abstract*–This article concerns the algorithms of stereovision and visual odometry, which can be applied to solve a three-dimensional reconstruction problem through monocular vision and describes the basic stages of the system, advantages and disadvantages of the above-mentioned methods.**

I. Introduction

Due to the wide distribution of digital mobile devices and a significant rate of their productivity growth, there are new ways of using their multimedia capabilities. At the same time, in various fields of digital technology 3D features are becoming increasingly popular; therefore the problem of the visual reproduction of surrounding reality in three dimensions is of great relevance. One of the most common methods to scan objects non-contact for three-dimensional models is stereo vision, i.e. the process to obtain a three-dimensional picture of the world through video data or a set of static images built by a stereo camera. However, 3D reconstruction is possible by using monocular vision. Camera movements allow receiving information about third dimension positions.

To assess the camera movements through a set of visual data, there can be applied another product of image processing science. Visual odometry is the method to assess the position and orientation of cameras by means of analyzing a sequence obtained from these images.

The synthesis of these algorithmic complexes allows scanning the observed scene in three dimensions.

II. Getting the Depth Map of the Observed Scene

A. Classification of Stereo Matching Algorithms

There are several classifications of stereo matching algorithms. Generally, all of the algorithms can be divided into two separate groups of algorithms. There are local methods, where the disparity for each image is computed based on the analysis of a region around point of interest (windows and blocks) and global algorithms based on minimizing the energy functional, where the difference is calculated for all pixels in an image at once. Global methods can be divided into subgroups of algorithms, combined by the type of energy minimization method. These are dynamic programming and graph cut methods. As a rule, the graph cut algorithm gives quite good results for the case of images without high-frequency regions, but they require a large number of machine computational resources; therefore, they are no longer applicable for computing a depth map in real time. Algorithms that match lines of images independently from each other are called one-dimensional. They are less resources-demanding, but are subject to various negative effects, which in return require additional treatment. Between these methods, there is the case of optimization algorithms for the graph sub tree that is constructed on the image pixels.

In this paper, we propose the algorithm that minimizes a cost function of the first order. In addition, it is enhanced by using the hierarchical structure.

B. The Hierarchical Stereo Matching Algorithm through Dynamic Programming

Typically, digital cameras of mobile devices have a significant amount of spurious artifacts in their image. The lens of any digital camera is not the perfect optical instrument, especially for the problems of computer vision. The first priority to implement complex algorithms for machine vision, such as optical flow tracking or building a depth map of the observed scene, is to eliminate the distortion of the lens through image reprojection.

The small size of camera matrix in most of mobile digital devices causes a considerable distortion of the image through thermal noise. The physical dimensions of a pixel matrix do not allow to accumulate a sufficient number of electrons during the exposure time, so their number is closer to the reading noise of the matrix. To reduce the effect of such noise is supposed to use hierarchical method for calculating the stereo correspondence maps.

The problem of stereo correspondence search is to computing the cross-correlation between the pixels of the left and right pictures. We denote the correspondence between pixels as : $M = (M_{xL}, M_{xR})$, where M_{xL} and M_{xR} - pixel indices on the left and right images , respectively. The discrepancy between the pixels δ is limited by two

predefined constants that define the geometry of the camera system and their optical characteristics:

$$\delta_{min} \leq \delta = xL - xR \leq \delta_{max}.$$

Thus, the magnitude of the maximum difference between pixels of the image (the shift of one image relative to the other in this point) will be determined as:

$$\delta_{\text{p}} = \delta_{max} - \delta_{min} + 1.$$

The complete sequence of stereo correspondences $\{M_i\}_{i=1..N}$ should meet a certain number of limitations:

$$M_{i,xL} < M_{j,xL} \& M_{i,xR} < M_{j,xR},$$

where $0 \leq i < j < N_m - 1$. This relation means that every unique pixel cannot belong to two different stereo correspondence. And the last:

$$M_{i+1,xL} = M_{i,xL} + 1 \vee M_{i+1,xR} = M_{i,xR} + 1$$

$$M_{0,xL} = 0 \vee M_{0,xR} = 0.$$

Here $i = 0..N_m - 2$. That is, a sequence of stereo correspondences cannot contain gaps - uncorrelated pixels. If there is a match between any two M pixels, the next pixel of the left and (or) right image must correlate with another image pixel. The sequence of stereo correspondences will be complete when the last pixel of the scan line on the left (right) image match is found [1]. Fig. 1 provides an example to illustrate these conclusions.

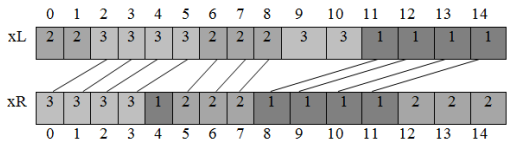


Fig. 1. Graphical representation of a match sequence with a left and a right occlusion. The left scanline is on top. The corresponding disparities are also indicated

In this example the sequence of corresponding pixels pairs found and the sequence of the calculated ones is:

$$D = \{?,?,2,2,2,2,1,1,1,?,?,3,3,3,3\}$$

The resulting sequence includes disparities, which cannot be estimated (occlusion) with simple methods. The obtained sequence of the differences relates to the cost function that consists of three components to be represented as follows:

$$f_c = \sum_{xL=0}^{W-1} \big(D(xL, xR)\big)|_{match} +$$

$$\sum_{xL=0}^{W-1} (\beta)|_{occl} + \sum_{xR=0}^{W-1} (\beta)|_{occl} +$$

$$\sum_{xL=0}^{W-1} \Big(\alpha, \big(D(xL) - D_{-1}(xL)\big)\Big).$$

The first term describes the value of this formula corresponding to the difference between the two pixels, between which it is possible to find the stereo correspondence. Occlusions are characterized by the value of β of the left and right images. The last term takes into account the dimension of the image, considering the values of the differences for the adjacent scan lines [2], [3].

The hierarchical structure implies the creation of a tree image through the decimation of the image [4].

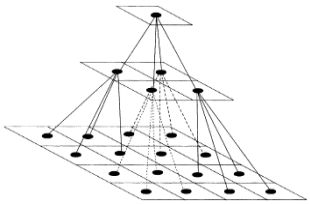


Fig. 2. Picture tree.

The ratio may vary between integer values and multiple lobes. The process of calculating the correct stereo correspondence is shown in Fig. 3:

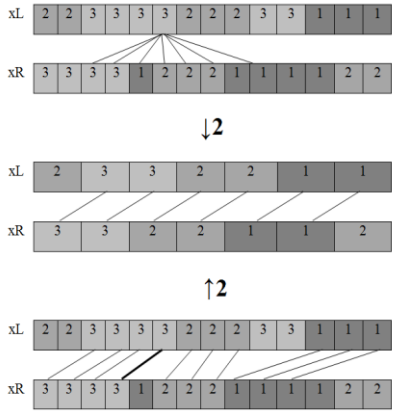


Fig. 3. An overview of the hierarchical stereo calculation

An additional task is to select the type of dissimilarity function. The dissimilarity function shows the degree of correlation between pixels (neighborhood pixels) on the sequences xL and xR. This value is the dominant term in the cost function. There are several ways to specify the dissimilarity function [5]. Let us turn to the most important one.

The simplest case of dissimilarity function is to calculate an absolute difference between the brightness of pixels:

$$D(xL, xR) = |\text{I(xL)} - \text{I(xR)}|.$$

The advantages of this method are its simplicity and calculation speed. However the real conditions severely limit the application of this method due to its weak immunity. Often, the absolute difference method is applied to a two-dimensional image area (Sum of Absolute Differences - SAD). To further reduce the influence of noise images, it is possible to use the sum of squared differences:

$$\mathrm{D_{sqr}}(xL, xR) = |\mathrm{I}^2(\mathrm{xL}) - I^2(\mathrm{xR})|$$

and cross-correlation method:

$$\mathrm{D_{cc}}(xL, xR) = \frac{\sum_{i=-by}^{by}\sum_{j=-bx}^{bx} L_{i,j}R_{i,j}}{\sqrt{\sum_{i=-by}^{by}\sum_{j=-bx}^{bx} {L_{i,j}}^2 \sum_{i=-by}^{by}\sum_{j=-bx}^{bx} {R_{i,j}}^2}}$$

$$L_{i,j} = (I_L(y+i, xL+j) - \overline{I_L})$$

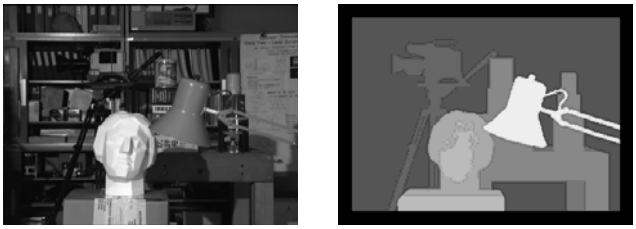
$$R_{i,j} = (I_R(y+i, xR+j) - \overline{I_R})$$

$$\overline{I_L} = \frac{\sum_{i=-by}^{by}\sum_{j=-bx}^{bx} I_L(y+i, xL+j)}{(2bx+1)(2by+1)}$$

$$\overline{I_R} = \frac{\sum_{i=-by}^{by}\sum_{j=-bx}^{bx} I_R(y+i, xR+j)}{(2bx+1)(2by+1)}.$$

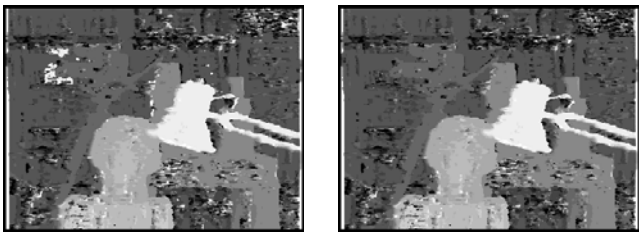
This is the most resistant to image-noise-method for calculate the dissimilarity function.

Comparative results of the hierarchical algorithm are given below.

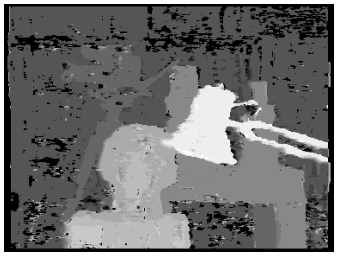


a b

Fig. 4. Tsukuba dataset: a) left image of stereo pair, b) Ground truth.



a b



c

Fig. 5. Resulted SAD disparity maps: a) simple DP method, b) Using hierarchical structure with 3 layers, c) Using hierarchical structure with 6 layers.

III. Visual Odometry through Optical Flows

A. The Problem of Visual Odometry

Visual odometry is the method to assess the position and orientation of the observer, through the analysis of the images sequence from the camera (cameras). The obtained data can be used to determine a travelled trajectory and to map an observed space.

In general, the visual odometry algorithm includes the following steps:

1) to preparation a camera image / stereo camera / omnidirectional cameras
2) to correct an image (eliminate the distortion, etc.)
3) to calculate an optical flow: to detect key features, to compare points between frames, and to construct an optical flow.
4) to check the optical flow vectors for potential errors and dropping emissions
5) to determinate the camera movement from the optical flow
6) to update periodically the set of the key points to be kept track of.

The optical flow is an apparent motion of objects, surfaces or sides of the stage that results from movements of the observer relative to the scene. Many computer vision algorithms are based on analyzing the motions of objects, surfaces or edges - motion detection algorithms, object segmentation, image coding, and stereo matching algorithms, which are mentioned above.

Thus, there are several types of areas in the image possessing of different features. Movements from frame to frame can be traced by the latter. Fig. 6 graphically shows examples of such domains.

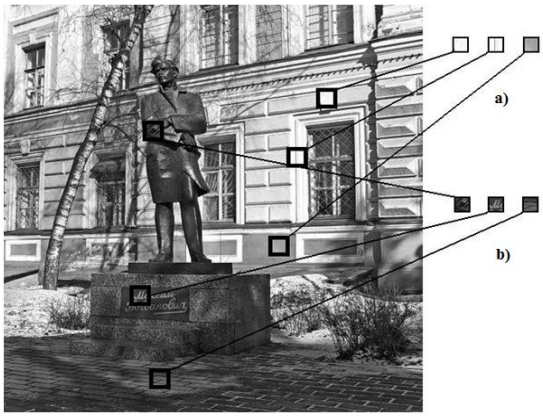


Fig. 6. Domains for feature tracking: a) bad, b) good.

It is most successful to track the flow of points, when the derivative of brightness has a local maximum. When the image is considered as a two-dimensional function, these features are called “corners”. They have enough information to track their movement between adjacent frames of a video stream.

The most wide-spread definition of corners as successful for tracking points, gave Harris [6]. It is based on the matrix of second derivatives of the image intensity, by analogy with the two-dimensional function of the Hessian. Harris corner is the image area, where the autocorrelation matrix of second derivatives has the two largest eigenvalues:

$$M = \begin{bmatrix} \sum\limits_{-k<i,j<k} I_x^2(x+i,y+j) & \sum\limits_{-k<i,j<k} I_{xy}(x+i,y+j) \\ I_{xy}(x+i,y+j) & I_y^2(x+i,y+j) \end{bmatrix}.$$

This means that the region includes edges or texture oriented domains in at least two orthogonal directions. From the last equation, the autocorrelation matrix computation is performed around the point in the limited area window. The window size affects both the result of the algorithm, and the speed.

B. Optical Flow Evaluation

As it is known, the basic equation of the optical flow cannot be solved uniquely since it contains two unknown parameters. Lucas - Kanade algorithm allows bypassing the ambiguity due to the information about the neighboring pixels at each point [7].

The method is based on the assumption that the local neighborhood of each pixel value of the optical flow is constant. It is possible to consider the main optical flow equation for all pixels of the neighborhood and solve the resulting system of equations by the ordinary least squares method.

In general, the assumptions made by using the Lucas – Kanade method, are:

1) Constant brightness;
2) The smallness of the movement;
3) The spatial coherence.

Brightness constancy suggests that when the point moves from frame to frame its brightness remains constant (considering halftones). The smallness of motion implies that the actual increment of time within the motion much more than the scale of the object movement on the image. Finally, neighboring point of the scene forms the surface and moves the same way from frame to frame. And it can be projected to adjacent points in the image.

Assuming that the local window around the pixel of interest p has the same optical flow for all N pixels, the system of equations for the optical flow vector is:

$$\begin{aligned} &I_X(p_q)V_X + I_Y(p_q)V_Y + I_t(p_q) = 0 \\ &I_X(p_2)V_X + I_Y(p_2)V_Y + I_t(p_2) = 0 \\ &\ldots \\ &I_X(p_N)V_X + I_Y(p_N)V_Y + I_t(p_N) = 0, \end{aligned}$$

where $I_X(p_n)$, $I_Y(p_n)$, $I_t(p_n)$ – partial derivatives of the image I to the x and y coordinates and times t.

This system can be represented in matrix form as $A \cdot V = B$, where:

$$A = \begin{bmatrix} I_X(p_q) & I_Y(p_q) \\ \ldots & \\ I_X(p_N) & I_Y(p_N) \end{bmatrix} B = \begin{bmatrix} -I_t(p_q) \\ \ldots \\ -I_t(p_N) \end{bmatrix} V = \begin{bmatrix} V_X \\ V_Y \end{bmatrix}.$$

The resulting system can be solved using the ordinary least squares method. As a result, we obtain the system of equations:

$$A^T AV = A^T B.$$

Then:

$$V = (A^T A)^{-1} A^T B,$$

$$\begin{bmatrix} V_X \\ V_Y \end{bmatrix} = \begin{bmatrix} \sum_i I_X(p_i)^2 & \sum_i I_X(p_i) I_Y(p_i) \\ \sum_i I_X(p_i) I_Y(p_i) & \sum_i I_Y(p_i)^2 \end{bmatrix}^{-1} \begin{bmatrix} -\sum_i I_X(p_i) I_t(p_i) \\ \sum_i I_Y(p_i) I_Y(p_i) \end{bmatrix}.$$

In general it is expedient to use the weighted ordinary least squares method. You can give more weight to pixels that are closer to the central pixel and less weight on the pixels that are closer to the edge of the window. In this case, the optical flow is:

$$\begin{bmatrix} V_X \\ V_Y \end{bmatrix} = \begin{bmatrix} \sum_i w_i I_X(p_i)^2 & \sum_i w_i I_X(p_i) I_Y(p_i) \\ \sum_i w_i I_X(p_i) I_Y(p_i) & \sum_i w_i I_Y(p_i)^2 \end{bmatrix}^{-1} \begin{bmatrix} -\sum_i w_i I_X(p_i) I_t(p_i) \\ \sum_i w_i I_Y(p_i) I_Y(p_i) \end{bmatrix},$$

where w_i – diagonal matrix containing the weights that are assigned to the pixels p_i.

Many algorithms for tracking key points based on the theory of this method. In general, they follow the idea of the Lucas – Kanade algorithm, but additional tools are used to improve the calculation accuracy ([8]).

C. Errors Elimination

The process of checking the vectors to the potential error is a necessary step in calculating the camera movement. The process of averaging the found motion vectors is as follows. The large number of found vectors means that any single emission weakly affect the final calculated camera movement. However, if we have the small number of vectors between frames, the elimination of erroneous values is a very important task.

There are different evaluation criteria for invalid vectors. The length of the vectors can be limited by a certain threshold. The threshold can be defined strictly, or adaptively. It based on the mean motion in the frame. The most versatile method is adaptive threshold computation through the average value in a certain area around the current point. Basing on this method, we can derive a condition for the length of the *i*-th motion vectors l_i:

$$\left| l_i - \frac{1}{N} \sum_{-K<p,q<K} (l(x+p, y+q) \right| < m,$$

where N - the number of vectors in the given area, and the size of the K value and the maximum difference m are chosen according to the requirements for accuracy. Also, the number of found motion vectors is the factor to choose the size of the area to average vectors.

a)

b)

c)

Fig. 7. Elimination of wrong value: a) found vectors, b) vectors after elimination process, c) eliminated vectors

D. Movement Calculation

Instant motion vector obtained by computing the optical flow between two consecutive frames has only an estimation of the real motion. The total error is shaping up from errors of the optical flow search, and the weakness of found key domains. To estimate a motion vector, it is convenient to use Kalman filter.

The real value of the motion vector is described by a discrete equation:

$$x(j+1) = A(j)x(j) + q(j),$$

where $q(j)$ – "white noise" with the inconsistency of the two neighboring values.

$$\left\langle q(j)q^T(l) \right\rangle = Q(j)\delta_{jl}.$$

The result of the motion vector observation can be written as:

$$z(j) = C(j)x(j) + r(j).$$

It is supposed that the priors estimation of the motion vector and the error covariance matrix x^- and P_x^- are known. Let's construct a regression equation between the random vectors: the error of estimation $x^- - x$ and the residual difference $z^- - z = Cx^- - z$:

$$x^- - x = K\left(Cx^- - z\right).$$
$$\left\langle \left(x^- - x\right)\left(Cx^- - z\right)^T \right\rangle = K\left\langle \left(Cx^- - z\right)\left(Cx^- - z\right)^T \right\rangle.$$

Since $z = Cx + r,$ we get:

$$\left\langle \left(x^- - x\right)\left(x^- - x\right)^T C^T \right\rangle - \left\langle \left(x^- - x\right)r^T \right\rangle =$$
$$K\left\langle \left[C\left(x^- - x\right) - r\right]\left(x^- - x\right)C^T - r^T \right]\right\rangle.$$

Given the uncorrelated r and error $x^- - x$, we have:

$$\mathrm{P}_x^- C^T = K\left[C\,\mathrm{P}_x^- C^T + R\right],$$

where R – error covariance matrix. Then:

$$K = \mathrm{P}_x^- C^T\left[C\,\mathrm{P}_x^- C^T + R\right]^{-1}.$$

After, we need to estimate a posteriori matrix P_x^+

$$\mathrm{P}_x^+ = \left\langle \left(x^+ - x\right)\left(x^+ - x\right)^T \right\rangle =$$
$$\left\langle \left[\left(x^- - x\right) + K\left(z - Cx^-\right)\right]\left[\left(x^- - x\right) + K\left(z - Cx^-\right)\right]^T \right\rangle =$$
$$= \left\langle \left(x^- - x\right)\left(x^- - x\right)^T \right\rangle + \left\langle \left(x^- - x\right)\left(z - Cx^-\right)^T \right\rangle K^T +$$
$$K\left\langle \left(z - Cx^-\right)\left(x^- - x\right)^T \right\rangle + + K\left\langle \left(z - Cx^-\right)\left(z - Cx^-\right)^T \right\rangle K^T,$$

and finally:

$$\mathrm{P}_x^+ = \mathrm{P}_x^- - KC\,\mathrm{P}_x^{-1}.$$

Once the information is received, it is necessary to define and assess the state vector and the covariance matrix of the estimation error. To move to the next point in time, you need to get back priori (predicted) estimates x_{j+1}^- and P_{j+1}^-. For the a priori estimation of the state vector:

$$x_{j+1}^- = A_j x_j^+.$$

The error covariance matrix can be obtained as follows:

$$\mathrm{P}_{j+1}^+ = \left\langle \left(x_{j+1}^- - x_{j+1}\right)\left(x_{j+1}^- - x_{j+1}\right)^T \right\rangle =$$
$$= \left\langle \left[A_j x_j^+ - \left(A_j x_j + q_j\right)\right]\cdot\left[A_j x_j^+ - \left(A_j x_j + q_j\right)\right]^T \right\rangle$$
$$= A_j\,\mathrm{P}_j^+\,A_j^T + Q_j,$$

Q_j – covariance matrix of "excitatory noise" q.

Now, the new values are known. This is the starting position for the next definition posteriori values x_{j+1}^+ and P_{j+1}^+ and etc. The whole algorithm of the discrete Kalman filter is thus reduced to the following actions [9]:

$$x_j^+ = x_j^- + K_j\left(z_j - C_j x_j^-\right),$$
$$K_j = \mathrm{P}_j^-\,C_j^T\left[C_j\,\mathrm{P}_j^-\,C_j^T + R_j\right]^{-1},$$
$$\mathrm{P}_j^+ = \mathrm{P}_j^- - K_j C_j\,\mathrm{P}_j^-,$$
$$x_{j+1}^- = A_j x_j^+,$$
$$\mathrm{P}_{j+1}^- = A_j\,\mathrm{P}_j^+\,A_j^T + Q$$

To make this recursive algorithm work, we must set a priori initial state vector and the covariance matrix. Since they may differ from the true values of these characteristics, these incorrectly given initial conditions will give a distorted assessment of the estimated state vector. However, with the passage of time, these distortions will be negated and this filter will work steadily.

We suppose that observer moves in two dimensions (x, y), which lies in the plane of motion. Each cycle is evaluation of optical flow in each of the dimensions (z_X, z_Y). The motion will be defined as:

$$\begin{pmatrix} x^+ \\ y^+ \end{pmatrix} = \begin{pmatrix} x^- \\ y^- \end{pmatrix} + \begin{pmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{pmatrix}\cdot\left(\begin{pmatrix} z_X \\ z_Y \end{pmatrix} - \begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix}\begin{pmatrix} x^- \\ y^- \end{pmatrix}\right)$$

Thus, this implementation of the Kalman filter takes into account the four dynamic parameters (two-dimensional coordinates and their increments) with two measured values.

The final stage of the development is the association of three-dimensional models based on the frames into a single three-dimensional map of the space [10], [11]. We suppose that the motion is in two dimensions, that is, the position of each model is defined by three parameters: $P_i = (x_i, y_i, \alpha_i)$, where x and y are the coordinates in the plane XY, and α is angle of rotation around the axis z. The total angle of rotation with respect to the initial position of the camera angle is determined by the vector obtained by summing the geometric instantaneous displacement vectors of the camera between shots, or:

$$\alpha_{sum} = \sum_i \alpha_i,$$

where α_i – camera angle between adjacent frames.

The following figures show the results of the computation of depth map and 3d reconstruction.

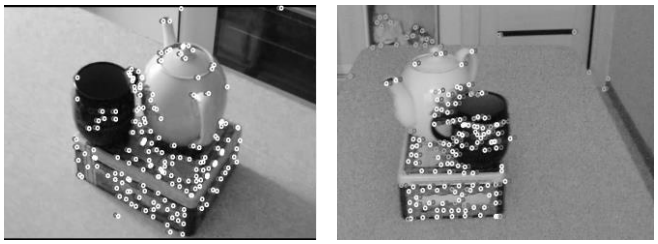


Fig. 8. Detected key features.

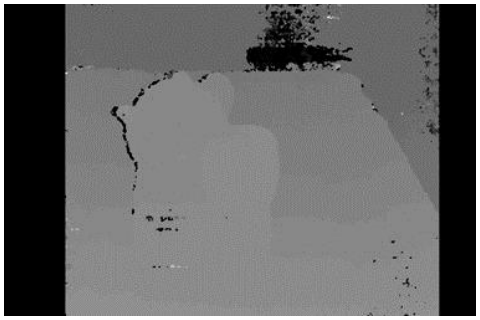


Fig. 9. Depth map computed by Hierarchical DP-based algorithm.

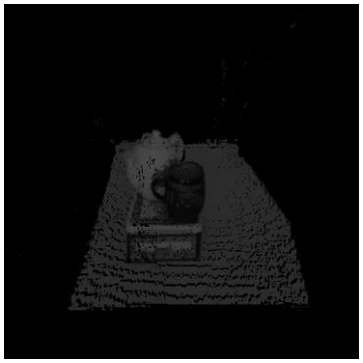


Fig. 10. Resulted 3D model (represented as points cloud) obtained by moving monocular camera.

IV. Conclusion

This article presents a set of algorithms that allows to process three-dimensional reconstruction of the scene through monocular vision. The basic steps to calculate the depth map of the scene, to find the key points on the image, optical flow and movement of the camera are discussed.

The methods to reduce the effects of mobile camera noise and optical flow computation errors are presented. Using the hierarchical structure in the depth map generation algorithm we can greatly improve the quality of obtained depth maps and the overall system performance. Filtering false optical flow vectors can reduce the error in the calculation of the final camera movement.

V. References

[1] D. Scharstein, R. Szeliski, “A taxonomy and evaluation of dense two-frame stereo correspondence algorithms”, *Int. Journal of Computer Vision*. April-June 2002, vol. 47, pp. 7–42.

[2] A. F. Bobic, S.S. Intille, “Large occlusion stereo”, *Int. Journal of Computer Vision.* 1999, vol. 33(3), pp. 181–200.

[3] V. Kolmogorov, R. Zabih, “Computing visual correspondence with occlusions using graph cuts”, *ICCV*, 2001, vol. 2, pp. 508–515.

[4] Kwang Hee Won, Soon Ki Jung, “Hierarchical Pyramid Based Stereo Matching Algorithm”, *Lecture Notes in Computer Science.* Volume 6915, 2011, pp 693-701.

[5] Y. Ohta, T. Kanade, “Stereo by intra- and inter- scanline search using dynamic programming”, *IEEE TPAMI*. 1985, vol. 7(2), pp. 139–154.

[6] C. Harris, M. Stephens, “A combined corner and edge detector”, *Proceedings of the 4th Alvey Vision Conference*. 1988, pp. 147–151.

[7] B. Lucas, T. Kanade, “An iterative image registration technique with an application to stereo vision”, *Proceedings of Imaging Understanding Workshop*. 1981, pp. 121–130.

[8] J. Shi, C. Tomasi, “Good Features to Track”, *IEEE Conference on Computer Vision and Pattern Recognition*. 1994, pp. 593–600.

[9] J. Hamilton, “Time Series Analysis”, *Princeton University Press*, Chapter 13, 1994.

[10] S. Neto, M. Bueno, T. Farias, J. Lima, V. Teichrieb, J. Kelner, I. Santos, “Experiences on the Implementation of a 3D Reconstruction Pipeline”, *International journal of modeling and simulation for the petroleum industry*. February 2008, vol. 2, no. 1.

[11] E.V. Davydenko, A.L. Priorov, “Automatic determination of the position of a video camera in a system of laser optical triangulation”, *Measurement Techniques*. 2008, V. 52, №8 pp. 841-845.

[12] J. Bouguet. Camera Calibration Toolbox for Matlab, Web: http://www.vision.caltech.edu/bouguetj/calib_doc.