CONTINUOUS REALIZATION OF MONOCULAR STRUCTURE FROM MOTION

Jan Klečka

Doctoral Degree Programme (2), FEEC BUT

E-mail: klecka@feec.vutbr.cz

Supervised by: Karel Horák

E-mail: horak@feec.vutbr.cz

Abstract: This paper describes a concept and practical aspects of a realization of an algorithm used for 3D scene reconstruction from multiple views captured by single moving camera. Position and orientation of the camera are a priory unknown and are calculated on-the-fly from tracked point correspondences.

Keywords: Structure from motion, multiple view reconstruction, visual SLAM, monocular SLAM

1. INTRODUCTION

Reconstruction of an environment by processing its pictures taken from different views has become a popular topic in recent years. Two major reasons for this are: their use in mobile robotic, where a single camera can be used e.g. for improving a navigation of an unmanned vehicle, and augmented and virtual reality applications.

Algorithm described in this paper can be referred as SLAM (Simultaneous Localization and Mapping) algorithm, because camera position and orientation are calculated simultaneously with 3D reconstruction, although in this particular realization trajectory is not being considered as part of the result. For single realizations of SLAM algorithms a varied group of sensors is being used e.g. matrix TOF, RGB-D cameras, camera stereo pair. The main difference between usage a single camera and the other sensors is an ambiguity of the final reconstruction. While data from mentioned sensors can be usually processed into reconstruction ambiguous only in a link to the global coordinates system (i.e. zero point and direction has to be chosen) in data captured by single camera can be find only monocular decencies and due to this the reconstruction is, in addition, ambiguous in overall scale (i.e. the distance unit has to be chosen). This property of monocular reconstruction can be useful – same device and algorithm can be used in dimensionally diverse environments.

2. ALGORITHM OUTLINE

Scene reconstruction from camera picture can be done in several ways. In this realization, I use a technique, based on reobservation a set of projections of 3D points, called passive triangulation. This technique can compute points 3D coordinates from at least two observation of its projection (correspondences). Such a reconstruction have a singular state when reconstructed point and camera centers lie all on the same line. This state occurs for example if observations are made from the same center of projection or reconstructed point is in camera movement direction.

So, the reconstruction problem leads to the correspondence search problem which is in dense variant computationally demanding and its computational complexity rapidly increase with the image count; that's why I decided to use only two pictures for reconstruction. However, two consecutive images captured by moving camera will be, most probably, close to the singular state of reconstruction, because their centers of projection will be close together. To avoid this, I develop a fast meth-

od, based on sparse correspondence points tracking, which decide whether the images are plausible for reconstruction.

Several point correspondences (specifically 8 or more) are also necessarily for another triangulation assumption - known camera projection matrix for every used image. I utilize the epipolar geometry constraints to get this information.

Epipolar geometry can be, also, used for facilitation of dense correspondence search by the process called rectification, therefore, deform processed images in such way that all correspondence will be in the same row (or column) of both deformed pictures i.e. correspondence can be searched only in one dimension instead of two.

Algorithm can be sum up into four points:

1. Point tracking, until camera movement, is significant to scene dimensions
2. Computation of camera position and orientation
3. Images rectification
4. Disparity map computation and scene reconstruction

2.1. POINT TRACKING

Realization of point tracking has been done by pyramidal modification Lucas-Kanade algorithm described in [4]. It's a standard feature tracking algorithm, reliable enough for this application. As a feature detector, I use the Harris detector.

Detection whether camera movement is significant to scene dimensions I treated by checking how tracked correspondences fit the 2D homography model. Let's have an image I_1 with feature vector $\mathbf{x}$ and image I_2 with feature vector $\mathbf{x}'$ which corresponds to $\mathbf{x}$ both feature vectors are in homogenous coordinates. Decision whether I_1 and I_2 should be processed by reconstruction algorithm is based these two equations:

$$err = \min_{\mathbf{H}} \left(\sum \left\| \mathbf{x}' \times \mathbf{H}\mathbf{x} \right\| \right) \tag{1}$$

$$J = \mathrm{Med}\left(\mathrm{d}\left(\mathbf{x}', \mathbf{H}\mathbf{x} \right) \right) \tag{2}$$

The equation (1) represents homography estimation by minimization of algebraic error it can be computed fast because it's in homogenous coordinates linear. The equation (2) express the decision criterion so the median of geometric error. The value of this criterion is checked for every picture and when it exceeds predefined threshold pictures are assumed as appropriate for scene reconstruction. Plausible threshold values can differ for different cameras but in my experiments optimal value lie between 20 and 80 most often, I use 50.

2.2. POSITION AND ORIENTATION COMPUTATION

Computation of camera position and orientation thru constraints of epipolar geometry can be done in several ways. I decided to utilize essential matrix decomposition as described in [1]. This approach works with so-called normalized image coordinates $\mathbf{x}_n$ which can be computed from image coordinates using the internal camera parameters.

$$\mathbf{x}_n = D^{-1}(r)\mathbf{K}^{-1}\mathbf{x} \tag{3}$$

Where: $\mathbf{K}$ is the matrix of linear internal camera parameters and $D(r)$ is non-linear distortion function dependent on the distance from camera principal point.

The essential matrix $\mathbf{E}$ determined by the following equation:

$$\mathbf{x}'_n \mathbf{E} \mathbf{x}_n = 0 \tag{4}$$

This matrix can be scaled and decomposed using singular value decomposition into:

$$\mathbf{U}\,\mathrm{diag}(1,\ 1,\ 0)\mathbf{V}^T = \mathbf{E} \tag{5}$$

Where $\mathbf{U}$ and $\mathbf{V}$ are orthogonal matrixes.

The decomposition then can be used for evaluation the external parameters of suitable projection matrixes $\mathbf{P}_n = [\mathbf{I} \mid \mathbf{0}]$ and $\mathbf{P}'_n = [\mathbf{R}_{rk} \mid \mathbf{t}_{rk}]$ in following way: Let's have constant matrix $\mathbf{W} = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$, then:

$$\mathbf{R}_{rk} = \{\mathbf{U}\mathbf{W}\mathbf{V}^T; \mathbf{U}\mathbf{W}^T\mathbf{V}^T\} \qquad \mathbf{t}_{rk} = \pm s\mathbf{u}_3 \tag{6}$$

Where: $[\mathbf{R}_{rk} \mid \mathbf{t}_{rk}]$ is the relative coordinate transformation between camera coordinates system in step $k-1$ and k, $\mathbf{u}_3$ is the third column of $\mathbf{U}$ and s is the scale factor.

As we can see this procedure leads to four possible solutions (two possible $\mathbf{R}_{rk}$ and two possible $\mathbf{t}_{rk}$ cause four combinations of $\mathbf{P}'_n$) the rights solution is the one which causes 3D reconstruction with positive depth in coordinates systems of both cameras. I check this using randomly chosen point correspondence.

Actual camera position is determined by global coordinates system transformation which is given by combination of all previous relative transformations:

$$[\mathbf{R}_{gk} \mid \mathbf{t}_{gk}] = \prod_{i=1}^{k} [\mathbf{R}_{ri} \mid \mathbf{t}_{ri}] \tag{7}$$

Finally, the scale factor should be determined. Because in my solution the true "metric" scale factor is ambiguous I fix the factor value for the initial image pair to 1. In the other pairs I hold the reconstruction scale by minimizing following criterion:

$$S_{crit} = \sum d\left([\mathbf{R}_{gk} \mid \mathbf{t}_{gk}]\hat{\mathbf{X}}, \mathbf{x}'_n\right)^2 \tag{8}$$

Where $\hat{\mathbf{X}}$ is the 3D reconstruction of $\mathbf{x}'_n$ made from image $k-2$ and $k-1$.

2.3. RECTIFICATION

The goal of rectification is to transform both images geometrically in such way that all epipolar lines will be parallel. Todays are widely used two rectification algorithms. The first is the linear rectification which is based on deforming both pictures by homography transformations which map epipoles into the infinity. But this concept will fail when one of the epipoles lie inside processed image. The second algorithm is the polar rectification [2]. This approach has no limitation and is based on remaping pictures into polar coordinates with the center in epipole; every corresponding epipolar lines are mapped on the same row. I chose to use the polar rectification because it can handle the general movement of the camera.

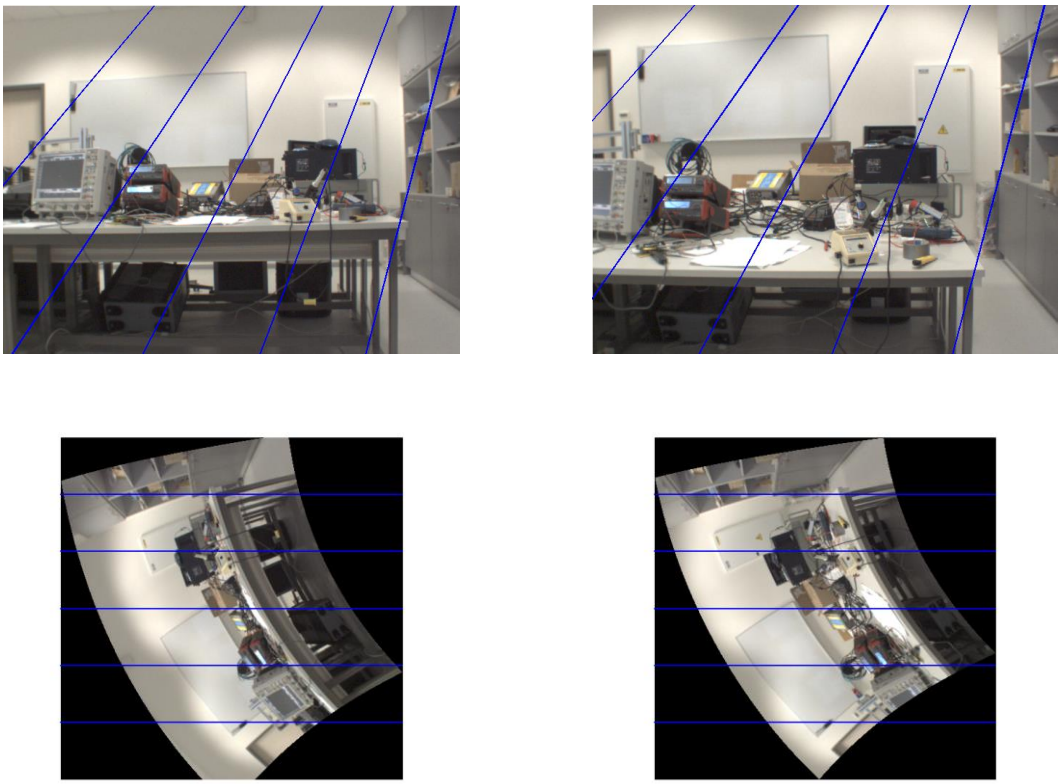

Figure 1: Example of polar rectification with few epipolar lines

2.4. RECONSTRUCTION

After rectification, the dense correspondence search can be performed. The result of such pixel-wise search in rectified coordinates is usually called disparity map. It exists several methods for disparity map generation I decided to use the semi-global method described in [3]. Because in rectified coordinates the correspondences can be found only in the same row (eventually column) disparity map is usually presented as a grayscale image of the differences in the column (eventually row) coordinate.

With known correspondences reconstruction can be done by homogenous triangulation method described in [1]. This requires a recalculation correspondences from polar coordinates and search for null space vector of 4x4 rank three matrix.

$$\begin{bmatrix} x\mathbf{p}^{3\mathrm{T}} - \mathbf{p}^{1\mathrm{T}} \\ y\mathbf{p}^{3\mathrm{T}} - \mathbf{p}^{2\mathrm{T}} \\ x'\mathbf{p}'^{3\mathrm{T}} - \mathbf{p}'^{1\mathrm{T}} \\ y'\mathbf{p}'^{3\mathrm{T}} - \mathbf{p}'^{1\mathrm{T}} \end{bmatrix} \mathbf{X} = \mathbf{0} \tag{9}$$

Where: $\mathbf{p}^{i}$ is the i-th row of the matrix $\left[\mathbf{R}_{g(k-1)} \mid \mathbf{t}_{g(k-1)}\right]$, $\mathbf{p}'^{i}$ is the i-th row of the matrix $\left[\mathbf{R}_{gk} \mid \mathbf{t}_{gk}\right]$ and x, y, x', y' correspondence coordinates transformed from polar rectification.

3. RESULTS

I implemented whole described algorithm in C++ using the OpenCV libraries and perform the real data experiment. As image source, I use a calibrated CCD camera with wide-angle lens, resolution 640x480. I move the camera manually by my hand.

Figure 2: Example of result of the real data experiment

4. CONCLUSION

I developed an algorithm which can conceptually provide multiple view reconstruction from monocularly captured image sequence. The main weakness of this algorithm is a drift of the scale estimation which is caused by poor $\hat{\mathbf{X}}$ estimate in equation (8). Further work will focus mainly on improving the quality of these point reconstructions.

ACKNOWLEDGEMENT

The completion of this paper was made possible by the grant No. FEKT-S-14-2429 - „The research of new control methods, measurement procedures and intelligent instruments in automation" financially supported by the Internal science fund of Brno University of Technology.

REFERENCES

[1] HARTLEY, Richard a Andrew ZISSERMAN. *Multiple view geometry in computer vision*. 2nd ed. Cambridge: Cambridge University Press, 2003. ISBN 05-215-4051-8.

[2] POLLEFEYS, M., R. KOCH a L. VAN GOOL. A simple and efficient rectification method for general motion. In: *Proceedings of the Seventh IEEE International Conference on Computer Vision* [online]. IEEE, 1999, s. 496-501 vol.1 [cit. 2016-03-04]. DOI: 10.1109/ICCV.1999.791262. ISBN 0-7695-0164-8. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=791262

[3] HIRSCHMÜLLER, Heiko. Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on Pattern Analysis and Machine Intelligence* [online]. 2008, **30**(2), 328-341 [cit. 2015-02-26]. DOI: 10.1109/TPAMI.2007.1166. ISSN 01628828.URL: http://ieeexplore.ieee.org.ezproxy.lib.vutbr.cz/xpl/articleDetails.jsp?arnumber=4359315

[4] BOUGUET, Jean-Yves. Pyramidal Implementation of the Lucas Kanade Feature Tracker. *Intel Corporation Microprocessor Research Labs* [online]. 2000 [cit. 2016-03-07]. URL: http://robots.stanford.edu/cs223b04/algo_tracking.pdf