3-D MAP-BUILDING BASED ON STEREO VISION FOR MOBILE ROBOT NAVIGATION

Adam Babinec

Bachelor Degree Programme (3), FIT BUT

E-mail: xbabin02@stud.fit.vutbr.cz

Supervised by: David Herman

E-mail: iherman@fit.vutbr.cz

Abstract: This paper describes the design and implementation of stereo vision system for robot navigation. The system is based on passive binocular stereo vision. It uses reprojection of a depth map obtained by application of block matching algorithms on image pairs taken by cameras to build up a local map. The local map is represented by voxel grid and supports detection of moving obstacles on ray-casting principle. The visual odometry is calculated by tracking triangulated paired image features detected in the image pairs. We also propose the modification of Iterative Closest Point (ICP) algorithm suitable for the estimation of transformation between two sets of triangulated features using weights derived from descriptor distances of features.

Keywords: stereo vision, robot navigation, Iterative Closest Point, feature matching, transformation estimation

1. INTRODUCTION

Extraction of information from the surrounding environment is one of the most important tasks in robot navigation systems. Mobile robot may not be able to fulfill its tasks without reliable and sufficient knowledge about the environment. For these purposes, passive binocular stereovision is one of the non-invasive methods which allows detection of position and shape of objects in 3-D scene. The aim of this paper is to design and implement a system of passive binocular stereo vision for mobile robot which builds the local 3-D map of the surrounding environment using information obtained from two cameras and estimates the position of robot in this environment.

2. PASSIVE BINOCULAR STEREO VISION

Passive binocular stereo vision uses geometric principle of pinhole cameras to estimate the position and shape of objects seen by both cameras. Every seen point P of the 3-D scene is projected onto 2-D matrices of image sensors in the cameras. Let $p_l(x_l, y_l)$ be projection of point P onto the matrix of the left camera sensor and $p_r(x_r, y_r)$ be projection of point P onto the matrix of the right camera sensor. If the characteristics of both cameras and geometric relations between the left and the right camera are known we can reproduce the position of point P in the 3-D scene using position of points p_l and p_r [1]. The main challenge of this approach is robust search for the point p_r in the image of the right camera which corresponds to the point p_l in the image of the left camera, so that those points are the correct projections of the point P. This search is performed effectively on rectified images which have parallel horizontally-aligned epipolar lines; therefore, the corresponding points p_l and p_r lie on the same horizontal scan-line in both images [3].

The reconstruction of the position of point P is given as [1]:

$$\mathbf{Q}\begin{bmatrix} x_l \\ y_l \\ d \\ 1 \end{bmatrix} = \begin{bmatrix} X \\ Y \\ Z \\ W \end{bmatrix} \tag{1}$$

In (1), $\mathbf{Q}$ is re-projection matrix obtained in stereo camera calibration and d is disparity $d = x_l - x_r$. The reconstructed position of point P is $P = (X/W, Y/W, Z/W)$.

3. VISUAL ODOMETRY

Visual odometry is based on the principle of transformation estimation of feature points in the 3-D space.

3.1. FEATURE POINTS

Feature points are detected in rectified images obtained from both cameras. Descriptor values are extracted for all feature points. The feature points are then paired up using the distances of their descriptors; the pairs whose points do not lie on the same horizontal epipolar line are rejected. We have implemented the following combinations of feature point detection, description and matching:

Detection algorithm	Description algorithm	Matching
SURF	SURF	Squared Euclidean Distance
STAR	SURF	Squared Euclidean Distance
ORB (Oriented BRIEF)	ORB	Hamming Distance
GFTT (Good Features To Track)	-	Lucas Canade with pyramids

Table 1: Implemented combinations of feature point detection, description and matching

3.2. TRANSFORMATION ESTIMATION

The transformation estimation of mobile robot position works on the principle of proposed modified Iterative Closest Point (ICP) algorithm, which takes into account both the position of the points and the descriptor distance of point pairs. Our modification uses the weights of the point pairs in the transformation estimation. Let M and D be two equally big sets of points in 3-D space:

$$d_i = \mathbf{R}\, m_i + T + V_i \mid d_i \in D, m_i \in M, i = 0..N \quad , \tag{2}$$

where $\mathbf{R}$ represents rotation matrix, T represents translation vector, and V_i represents noise vector [2]. Finding optimal transformation $[\hat{\mathbf{R}}, \hat{T}]$ is carried out by the least square method. Optimal translation vector $\hat{T}$ is obtained as the difference of centroids d_C and m_C of sets M and D:

$$d_C = \frac{\sum_{i=0}^{N}(w_i\, d_i)}{\sum_{i=0}^{N}(w_i)};\;\; m_C = \frac{\sum_{i=0}^{N}(w_i\, m_i)}{\sum_{i=0}^{N}(w_i)};\;\; \hat{T} = d_C - m_C \quad , \tag{3}$$

where w_i is the weight of point pair $(d_i,\ m_i)$. Optimal rotation matrix can be obtained by singular value decomposition of matrix $\mathbf{H}$ [2]:

$$\mathbf{H} = \sum_{i=0}^{N}\left((m_{Ci} w_i)\,(d_{Ci} w_i)^T\right);\;\; \mathbf{H} = \mathbf{U\Lambda V}^T \quad , \tag{4}$$

where $d_{Ci} = d_i - d_C$ and $m_{Ci} = m_i - m_C$ and $\mathbf{U\Lambda V}^T$ is singular value decomposition of matrix $\mathbf{H}$. Then optimal rotation matrix $\hat{\mathbf{R}}$ is [2]:

$$\hat{\mathbf{R}} = \mathbf{V}\,\mathbf{U}^T \tag{5}$$

The value of weight w_i of point pair $(d_i,\ m_i)$ is derived from the sum of descriptor distances of points d_i and m_i:

$$w_i = \frac{1}{\text{dist}\,(d_{iL}, d_{iR}) + \text{dist}\,(m_{iL}, m_{iR}) + \text{dist}\,(d_{iL}, m_{iL}) + \text{dist}\,(d_{iR}, m_{iR}) + b} \quad , \tag{6}$$

where $\text{dist}(p_x, p_y)$ is distance of descriptors of 2-D points p_x and p_y. Points d_{iL}, d_{iR}, m_{iL}, and m_{iR} are projections of points d_i and m_i on the matrix of left and right camera sensor. Constant b is a positive real number. The bigger the constant b is, the less important are the distances of descriptors.

4. LOCAL 3-D MAP

Disparity map D_m is created by using classic Block Matching algorithm or Semi-Global Block Matching algorithm. These algorithms are used to re-project points of rectified image taken by the left camera into the 3-D space. Disparity d of point $p_l(x, y)$ in the left camera image is [1]:

$$d = D_m(x, y) \tag{6}$$

By using the formula (1) it is possible to reconstruct the 3-D position of all points of the left camera image for which disparity values are defined. This creates a point cloud in the 3-D space which is then transformed according to the robot's position and orientation, and it is subsequently mapped into a local 3-D map. The local map is represented as voxel grid stored in an octal tree. Each voxel of the local map can be marked as free, occupied, or undefined. Mapping of point cloud onto the local map is divided into two steps.

Firstly, the point cloud is filtered and voxelized. Moving obstacles are detected in the local map using ray-casting from camera position into each voxel of the point cloud. If there is an occupied voxel in the local map in the trajectory of casted ray, it is signalized as a moving obstacle and rejected from the local map. Then all voxels in the trajectory of casted ray are set as free voxels.

Subsequently, the voxelized point cloud is inserted into the local map, so that when voxelized point cloud is occupied but the respective voxel in the local map is set as free, the voxel of voxelized point cloud is not inserted into the local map and it is signalized as a moving obstacle. Otherwise, the respective voxel of the local map is set as occupied. In this way, it is possible to detect moving obstacles in the environment. Moving obstacles do not affect the local map, thus the local map holds only data about static objects in the environment.

5. CONCLUSION

The proposed passive binocular stereo vision system was implemented in Qt framework using libraries OpenCV and PCL. Its functionality was confirmed by an application on the mobile robot Quido which is part of the Roboauto project. The system has a graphic user interface and it supports recording and replaying the scenes seen by a stereo camera. A stereo camera calibration tool and an application for parameter tuning of disparity map algorithms were also created. The current implementation shows that the combination of visual odometry based on ORB feature detection, matching and point cloud re-projection based on the disparity map computed by the classic Block Matching algorithm yields the best results in a reasonable computational time.

6. ACKNOWLEDGEMENT

This work has been prepared with the support of Research plan MSM 0021630528 of the Ministry of Education, Youth and Sports of Czech Republic.

7. REFERENCES

[1] Bradski, G., Kaehler, A.: *Learning OpenCV*. First Edition. Sebastopol, CA: O'Reilly Media, 2008. s. 415-453. ISBN: 978-0-596-51613-0.

[2] Eggert, D.: Estimating 3-D rigid body transformation: a comparison of four major algorithms. In: *Machine Vision and Applications.* New York, NJ: Springer-Verlag, 1999. Volume 9. s. 272-290. ISSN: 0932-8092.

[3] Loop, C., Zhang, Z.: Computing Rectifying Homographies for Stereo Vision. In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. Fort Collins, CO, USA, 1999. Volume 1. s. 125-131. Technical Report. ISBN: 0-7695-0149-4.