

# Obstacle Avoidance Procedure and Lee Algorithm Based Path Replanner for Autonomous Mobile Platforms

Maciej Polańczyk, Michał Strzelecki, and Krzysztof Ślot

**Abstract**—The paper proposes solution for two important issues connected to navigation of independent mobile platforms in an unknown environment. First issue relates to obstacle map, estimated based on stereovision images. It provides a basis for further platform path-planning. The main problem that has to be solved in obstacle map derivation is elimination of artifacts resulting from depth estimation. Thus a two-step artifact filtering procedure is proposed, which exploits both within-frame spatial correlations as well as temporal, between-frame correlations to do this task. Second procedure, based on well-known Lees algorithm is designed for obtaining vehicle collisionless path. Such routes need to be updated on-the-fly to take into account moving obstacles or newly detected objects. The main idea of the proposed approach is to identify regions where environment has changed and to execute a procedure of selective path updates. As a result, an optimal path can be derived at a computational expense comparable to the heuristic Lifelong A\* search. Experiment results demonstrate efficiency of the two discussed approaches for platform operation control in real environments, where both static and moving obstacles are present.

**Keywords**—mobile platform motion system, camera motion estimation, map of obstacles, path planning.

## I. INTRODUCTION

THE CAPABILITY of collision-free navigation in an unknown environment is a basic requirement that majority of the autonomous robotic mobile platforms have to satisfy. In many systems, e.g. robots aimed for environment exploration, there is no way to provide up-to-date detailed obstacle map beforehand. In such systems the missing knowledge has to be discovered by the system in real time. To cope with problems of mobile platform operation in dynamically changing and partly-known environments, the concept of the Simultaneous Localization And Mapping (SLAM) was introduced.

SLAM methods build an up-to-date map while robotic devices estimate their own position. While moving, the map is updated every time the new obstacles are encountered. Not only new obstacles are marked on the map, but also the old ones position is updated according to new platform localization. This has to be done in order to preserve spatial relationships between objects. To ensure the localization reliability, complex computations must be performed in real time.

Another problem connected to platform motion control is appropriate path planning to avoid collisions in dynamically

changing environment. Most of path-planning algorithms assume that obstacle distribution is known before the unmanned vehicle sets off [1]. One of the most commonly-used search methods the heuristic A\* algorithm [2], outperforms other ones in terms of the computational burden, however, it can be further improved if accumulated history from previous steps is taken into account. This idea is implemented e.g. in Lifelong Planning A\* [3], which is a tuned version of A\* or the D\* and the D\* lite algorithms [4], [5]. Lees algorithm [6], originally developed for printed circuit board design, can also be harnessed to plan routes for autonomous vehicles. The algorithm is optimal yet simple and its incurred computational cost is low. However, as it does not use any heuristics, it is usually slower than A\*.

The objective of this paper is twofold. First, a procedure for collision avoidance, aimed for autonomous mobile platforms equipped with stereovision camera, is presented. In such systems obstacle localization can be determined from depth image. Unfortunately, errors introduced during disparity calculation can significantly impair correct identification and localization of obstacles. Therefore there is a need to improve process of building map of the obstacles by eliminating artifacts introduced during stereovision depth image formation. To solve this problem a novel approach, based on spatio-temporal filtering, is proposed. In motion control module a fuzzy logic algorithm was used, resulting in an overall hybrid, fuzzy-crisp computational architecture of the proposed system. The experimental results show that the resulting method enables real time navigation in unknown indoor environments such as corridors or office rooms.

Secondly, the extension of original Lee approach was proposed to enable path-replanning mode yielding significant reduction in its execution time and becoming competitive to heuristic search methods.

This paper is organized as follows. Section 2 presents the literature review on obstacle map estimation, Section 3 describes proposed method for such a map building. Section 4 discusses the new path replanner algorithm, Section 5 shows sample experimental results obtained for moving platform implementing these two novel approaches, finally Section 6 concludes the paper.

## II. APPROACHES TO ESTIMATION OF THE OBSTACLE MAP

Several approaches that deal with the problem of building up-to-date obstacle maps can be found in literature. Most

of the existing solutions use a laser rangefinder, augmented with monocular/stereo vision [7], [8], [9] or sonar sensors [10], as a source of information on the environment. Such complex systems are characterized by high precision and reliability but are expensive. Therefore, many researches focus on stereovision-only based systems that are more error-prone but much cheaper [11].

In majority of stereovision-based systems imprecision of disparity calculation is compensated by fusion of utilizing a single camera image and depth image. For example in [12] the depth discontinuities in U-V-disparity domain is used to confirm the existence of obstacles. Other researchers have developed sub-pixel displacement methods [13] to enhance the accuracy of disparity.

The proposed approach for constructing reliable obstacle maps exploits only depth information. Disparity derivation errors are corrected by both spatial and temporal filtering. We use original depth information, with its insufficient precision to get a general environment perception, whereas appropriately filtered information, without artifacts, is used for path-planning. The proposed platform-guidance system provides reliable mapping of an environment, correct short-term estimation of the platforms current position and real-time operation.

### III. PROPOSED SOLUTION

The proposed algorithm consists of 3 stages (I) camera motion estimation, (II) building map of obstacles and (III) path planning - as presented on the flowchart in Fig. 1. Detailed description of camera motion estimation algorithm, which is used in the proposed procedure, can be found in previous authors work [14], [15]. Path planning is based on the modified version of Lees algorithm [16]. See Section 4 for details.

#### A. Preliminary Obstacles Detection

Preliminary obstacle detection is performed on the basis of stereovision depth image (Fig. 1d). In parallel with camera motion estimation (Fig. 1e), ambient obstacles are searched for. First, the points that are located at distances larger than  $Z_{max}$  (10m) from the camera are discarded. This way certainty of point localization correctness in subsequent steps increases (disparity for closer points can be estimated more accurately).

Points which Z coordinate is in the range  $0H_{max}$  ( $H_{max}$  platforms height, 0.8m) from the camera are considered as belonging to obstacles and are taken into account for further map construction (Fig.1b). Such assumption enables eliminating points lying on the surface on which mobile platform is moving and located on hanging obstacles that have no impact on safety platform motion (note that to adopt this assumption the camera needs to be mounted in parallel with the ground). Next, points that belong to the considered volume are transformed from camera coordinates to 2D map of obstacles (Fig. 1c). The transformation is done according to the formula (1).

$$\begin{bmatrix} X_m \\ Z_m \end{bmatrix} = \begin{bmatrix} \cos(\beta) & \sin(\beta) \\ -\sin(\beta) & \cos(\beta) \end{bmatrix} \cdot \begin{bmatrix} X \\ Z \end{bmatrix} + \begin{bmatrix} X_c \\ Z_c \end{bmatrix} \quad (1)$$

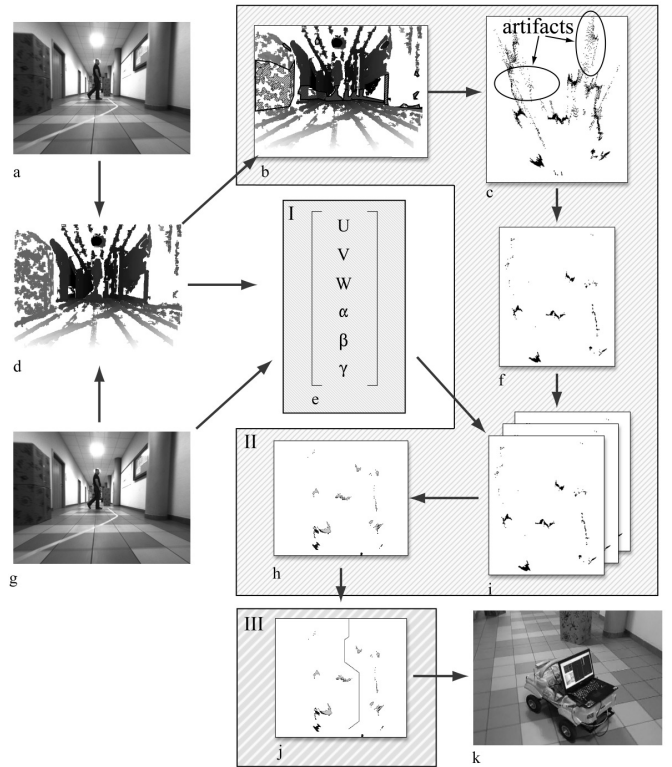


Fig. 1. The flowchart of the proposed algorithm. (a) Right image. (b) Disparity image with marked potential obstacles. (c) Potential obstacles points transformed to 2D map. (d) Depth image. (e) Motion vector. (f) Preliminary map of obstacles (c) after erosion. (g) Left image. (h) Final map of obstacles. (i) Time filtering. (j) Path. (k) Control mobile platform. (I) Camera motion vector estimation. (II) Building map of obstacles. (III) Path planning.

where  $X_m, Z_m$  - point coordinates in map,  $X_c, Z_c$  - point coordinates in real world,  $X, Z$  - camera current position in map,  $\beta$  - direction of current camera movement.

#### B. Building Reliable Up-To-Date Map of Obstacles

Preliminary map of obstacles created in previous step contains artifacts introduced by errors in disparity calculation (Fig. 1c). These artifacts have to be filtered out as they cause misjudgment what is an obstacle. Therefore before points preliminarily classified as belonging to an obstacle are added to the global map of obstacles they undergo spatio-temporal filtration (Figs. 1f, 1c). They are stored in temporary buffer and:

- 1) the  $3 \times 3$  erosion with cross-shaped structuring element is applied to eliminate single points (spatial filtering). The probability that only one point was detected on the obstacle is considered as very low thus such a point can be neglected (Fig. 1f).
- 2) only points that are valid in three subsequent frames (points from each frame are stored in separate temporary buffers (Fig. 1i) are left (temporal filtering) and added to global obstacle map (Fig. 1h).

While the camera is moving, a new region is explored and new obstacles can appear or some can change their position. Thus there is a continuous need to update the global map.

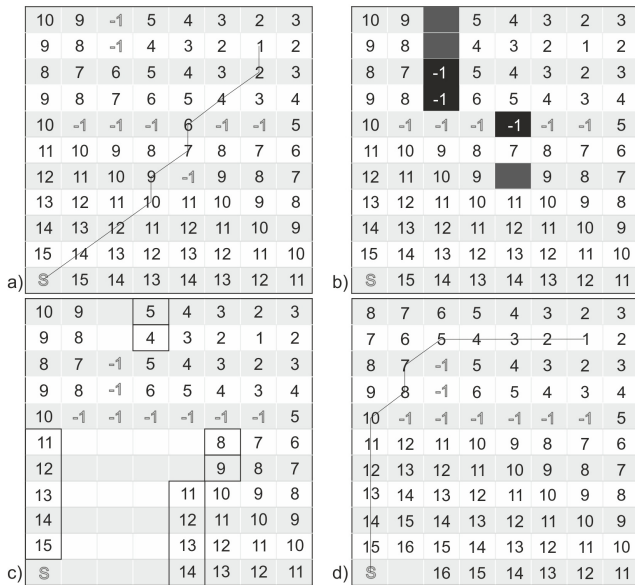


Fig. 2. Key phases of the proposed algorithm. An initial path derived using Lee's algorithm (a), innovation detection (b), determination of expansion register cells (c), new path selection (d).

Points from global obstacle map that are located within the camera visual proximity (defined as the 10m-long rectangular area in front of the camera restricted by a 100-degree camera viewing angle) are taken into further consideration, as described in [17]. Firstly, it is checked if the analyzed point is present in all three temporary buffers. If it is, no update is needed. If not, the label potentially to remove is assigned with such a point and a follow-up procedure is done. In subsequent steps it is verified whether this point is missing from the temporary buffer because the obstacle is not valid anymore and should be removed or if it was occluded by another obstacle. In order to do this the disparities are compared. If the disparity is:

- 1) Larger than the one resultant from the distance it means that the obstacle was occluded by another one and should not be removed from global map (closer object has larger disparity).
- 2) Otherwise the object changed its position and has to be removed from previous location on the map (some farther object is now within the camera viewing range).

Disparity calculation based on the data from global map is not a trivial task as the map is two-dimensional. Only information about X and Z coordinates is stored. To cope with this problem, disparities for all points in the range of Y: 0 Hmax are computed and compared.

Finally, temporary buffers are updated by new data so that data associated with the latest three images were stored and analyzed. Such updated global map of obstacles is an entry point for the path planning algorithm.

#### IV. THE PATH RE-PLANNING ALGORITHM

Before the proposed path-replanning algorithm is launched, an initial environment map is built and the optimal path is

found using the original Lee's method. To ensure computational efficiency of the algorithm a vector of queues, referred to as an expansion-register is proposed to be used as the data structure for handling the node-expansion process. Each entry of this vector groups cells with the same integer label. Expansion of a node produces a set of new cells, which are placed at the end of a queue corresponding to the subsequent registers row.

The proposed algorithm iterates through the following three steps, each time a new, updated environment map is available (see Fig. 2). The first step identifies cells that have got vacated or taken due to obstacle location changes; it is performed by subtracting two consecutive environment maps. The two possible innovations are handled differently in the subsequent step of the algorithm. If a cell gets vacated (it gets an empty label), the algorithm looks for its neighbour with the lowest possible integer label and registers its location. This neighbour is appended to an appropriate row of the expansion register, for further handling in the third phase of the procedure. If a cell, initially labeled with some integer  $k$ , gets taken by an obstacle, the algorithm first determines all descendants of this lost location. Descendants are defined as these cells within its 4-neighborhood, which are assigned with the subsequent integer  $(k+1)$ . Descendant-tracking is a recursive process, which results in selection of a pool (or pools) of cells with labels that are no longer valid. All the cells surrounding the selected pools (i.e. elements of possibly closed contours) are added to appropriate rows (based on their indices) of the expansion register.

The third phase of the algorithm is the expansion of cells that are stored in the expansion register. Expansion is made row-wise and involves checking, whether some currently analyzed registers cell has any unlabeled neighbors or neighbors with labels that are larger by at least a value of two. If this is the case, such cells are placed at the end of a queue of the consecutive row. Expansion proceeds until none of the two above mentioned conditions hold or until the target cell is visited. In the latter case, the backtracking step is necessary to conclude the procedure.

Sample histograms that show expansion register usage (an estimate of computational burden) for path-planning by means of the extensive search, performed in the initial stage of the proposed method (using Lee's approach) and subsequent path-replanning iterations, are shown in Fig. 3. Clearly, path-replanning typically involves examination of only a fraction of cells that need to be considered if data processing history is not exploited. The proposed algorithm and the introduced way of data representation enable substantial reduction in path-planning computational cost, which has been experimentally validated.

#### V. EXPERIMENTAL RESULTS

This Section discusses experimental results obtained based on two proposed solutions applied for platform motion control. To implement and validate the both approaches a test mobile platform, equipped with the Bumblebee2 stereovision camera [18], was used and several sequences of images, for indoor

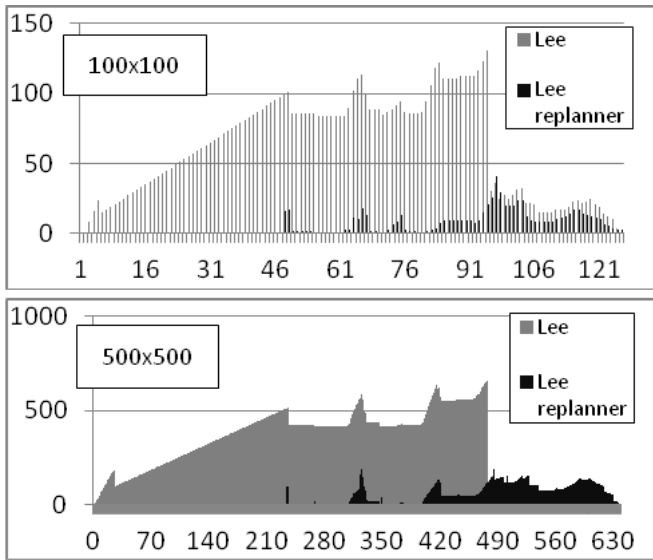


Fig. 3. Typical expansion-register usage (number of cells to be expanded at subsequent steps) for Lee algorithm and the proposed method (for 100x100 grid exploration and for 500x500 grid exploration).



Fig. 4. The test mobile platform.

environment, were captured. This platform is presented in Fig. 4.

#### A. Obstacle Map Estimation

In order to increase computational efficiency, the proposed algorithm is implemented in four independent parallel threads. The first thread is responsible for capturing stereovision images and calculating disparity. In the second one the camera motion vector is estimated. Obstacles map derivation and path planning is performed by third thread. The fourth thread controls mobile platform motion.

Figure 5 presents the experimental results. In Fig. 5a the right image captured by stereovision camera is shown. Fig. 4b

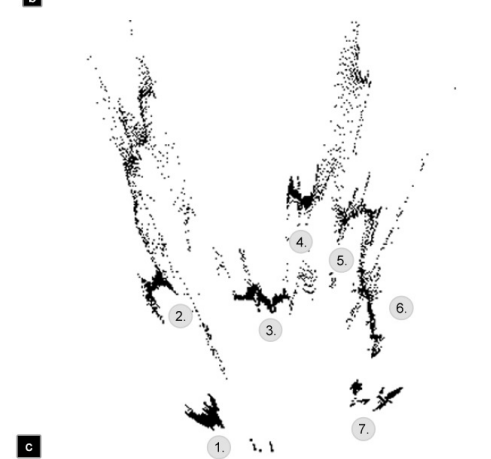
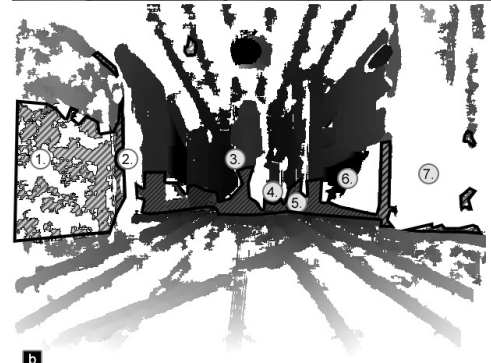


Fig. 5. The experimental results of proposed solution. (a) The frame captured from stereovision camera. (b) Disparity image; points being analyzed are marked with slanting lines; the darker color the further localized point; point for which disparity could not be calculated are marked white. (c) Potential obstacles points transformed to 2D map. Artifacts are visible. (d) Map of obstacles after spatial-time filtering.

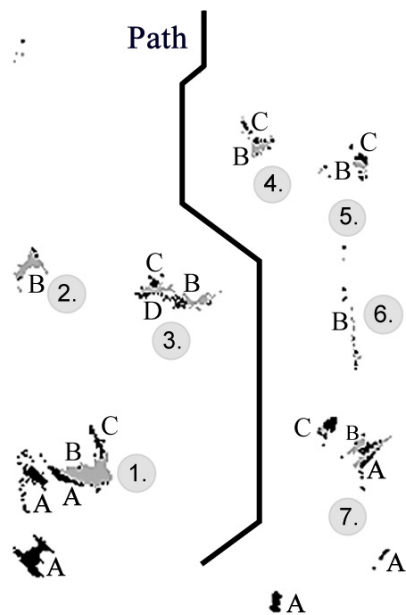


Fig. 6. Global map of obstacles with found path, A obstacles outside the camera vision range, B obstacles within camera vision range, C obstacles hidden by other objects, D obstacles to remove (not valid any more).

presents the stereovision disparity image being the base of all calculations - the darker color the farther localized point. Points for which disparity could not be calculated are marked in white. Regions marked with slanted lines correspond to potential obstacles. In Fig. 5c the preliminary map of obstacles is shown. Not only obstacle points (labeled 1-7) but also artifacts are visible. Most of the artifacts are removed after spatio-temporal filtering Fig. 5d. In Fig. 6 the global map of obstacles, built during the process of exploration, is presented. The thick dark line shows the optimal path. The A-D labels mean respectively: A obstacles outside the camera vision range; B obstacles within camera vision range; C obstacles hidden by other objects; D obstacles to remove (not valid any more).

Camera motion estimation error is about 5%. Such errors propagate and after long-term exploration a deviation between real camera position and estimated one can reach unacceptable level and result in wrong obstacles mapping. This is a common problem of majority of algorithms based on visual odometry, so there is a need of periodical corrections of localization process, using some other input modality, such as e.g. GPS.

Also, as the process of detecting objects and building a map is based only on disparity image, homogenous obstacles can be missed. The possible solution to this problem is to consider additional information from some other type of sensors or introduce to the system some additional module of object detection based on raw image (not a depth-map) analysis.

In the Table 1 the average times of computation for every stage of the proposed algorithm are presented. All calculations were performed using 2.5 GHz quad core computer. Resolution of images was 640x480. It shows that the path planning module is a weak point of the proposed solution. In order to cope with this problem some improvements were introduced,

TABLE I  
AVERAGE TIMES OF COMPUTATION

Stage	Time [ms]	Thread
Image acquisition and disparity calculation	60	1
Camera motion estimation	40	2
Obstacles detection	15	3
Map updating	20	3
Path planning	5	3
Mobile platform control	<1	4

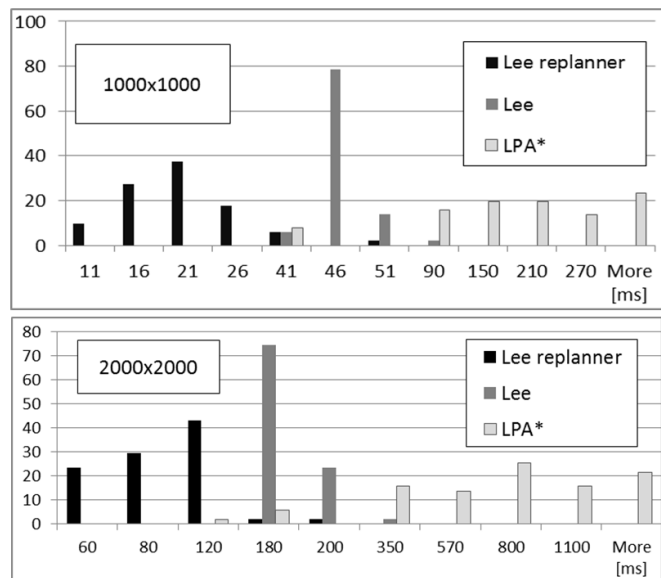


Fig. 7. Path-planning time histograms for the proposed method, standard Lees procedure and LPA\* algorithm for two grids: 1000x1000 cells and 2000x2000 cells. A total of 100 000 maps (derived from 10 different obstacle distributions by making 10000 steps of random obstacle displacements) were analyzed.

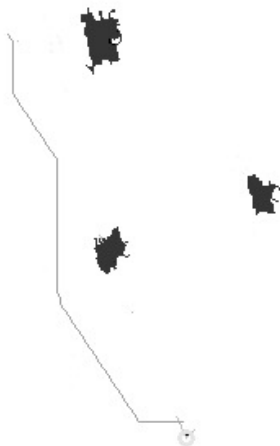
e.g. updating a path in near neighborhood only. Also, one can consider entirely different, more computationally efficient path-planning algorithms, where sub-optimal path selection is sufficient. Comparing the performance with other solutions tested on similar equipment (the same camera, 2.4GHz processor), a significant improvement can be observed (average computational time for images with resolution 640x480 pixels for proposed algorithm does not exceed 133ms whereas in [12] for images with resolution 320x240 it is 230ms). It results in more reliable system, being able to avoid obstacles in dynamically changing environment.

### B. Lee-based Path Replanner

Computational complexity of the proposed algorithm is a function of obstacle distribution. Therefore, algorithms performance has been evaluated experimentally by selecting random obstacle configurations and updating obstacle locations by random displacements, thus imitating consecutive snapshots of the monitored environment. Each obstacle configuration was composed of rectangular objects of width/height changing randomly from one pixel to 10% of the map side size. Total area taken by obstacles have been varying from 10% to 40% of the map size. The route calculations were performed on a 2.53GHz processor. The results are shown in Fig. 7

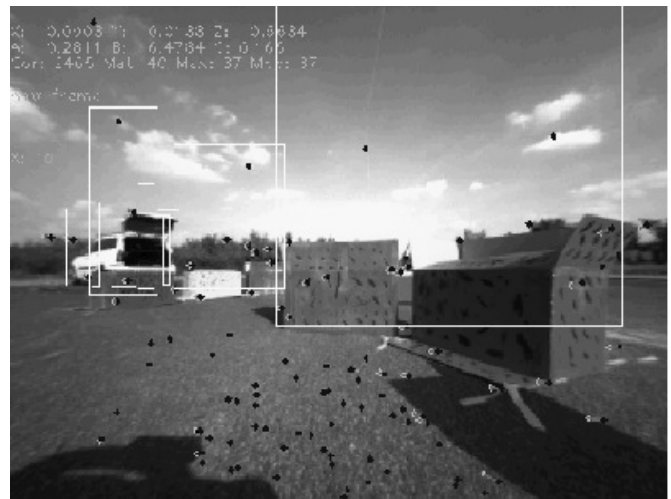


a)



b)

Fig. 8. Results for path planning. The image from stereo camera (a), corresponding obstacle map with marked collisionless path (b).



a)



b)

Fig. 9. Results for path planning. The image from stereo camera (a), corresponding obstacle map with marked collisionless path (b).

TABLE II  
AVERAGE TIMES OF COMPUTATION

Map size [cells]	Lee	LR	LPA*	Path length [cells]
100x100	<1	<1	<1-10	95-150
500x500	8	2-6	1-300	470-750
1000x1000	45	15-30	3-1000	950-1500
2000x2000	180	50-120	10-3000	1900-3000

and Table 2, together with results obtained for the heuristic Lifelong Planning A\* algorithm [3], executed on the same data.

As the search-procedure origin changes in time if assigned to a moving vehicle, we decided to swap the two terminal points of the path and to compute a path that starts from the target and terminates at the origin. Several sequences of images and planned paths, for dynamic outdoor and indoor environment were captured. Sample results are presented in Figs. 8 and 9.

## VI. CONCLUSION

The presented algorithm correctly builds the obstacle map in majority of tested cases. Introduced spatio-temporal filtering of artifacts improved quality of maps significantly. The precision of obstacle localization allows avoiding collisions and reaching the destination. Camera motion estimation is performed with ca. 17 frames/s and path planning with ca. 7 frames/s. The average delay between the path updates is ca. 230ms.

Further research will focus on modifications of the path planning procedure, aimed at reducing its computational burden. Additionally, the way the map of obstacles is updated will be modified to prevent from removing of fast moving objects, which is the case for the current algorithm.

The procedure proposed for planning collisionless routes for unknown environments proved to be several times faster than original Lees algorithm. For the dynamically changing maps considered in this study, also the heuristic LPA\* algorithm did not perform well. This was caused by frequent labyrinth-like obstacle configurations, resulting in instability of heuristic

search performance (i.e. frequent jumps during execution time). As computational complexity of the proposed algorithm is low and the resulting paths are optimal, it becomes an attractive candidate for application in run-time routing, e.g. for unmanned moving vehicles.

#### REFERENCES

- [1] J.-C. Latombe, *Robot Motion Planning*. Boston, MA: Kluwer Academic Publishers, 1991.
- [2] P. Wait Hart, N. Nilsson, and B. Rafael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, pp. 100–107, 1968.
- [3] S. Koenig, M. Likhachev, and D. Furcy, "Lifelong Planning A\*," *Artificial Intelligence Journal*, vol. 155, no. 1–2, pp. 93–146, 2004.
- [4] A. Stentz, "Optimal and efficient path planning for partially-known environments," *Proceedings of International Conference on Robotics and Automation, San Diego*, vol. 4, pp. 3310–3317, 1994.
- [5] S. Koenig and M. Likhachev, "D\* Lite," *Proceedings of the AAAI Conference of Artificial Intelligence*, pp. 476–483, 2002.
- [6] C. Y. Lee, "An algorithm for path connection and its applications," *IRE Transactions on Electronic Computers*, vol. EC-10, no. 3, pp. 346–365, 1961.
- [7] J. A. Castellanos, J. Neira, and J. D. Tardos, "Multisensor fusion for simultaneous localization and map building," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 6, pp. 908–914, 2001.
- [8] A. Murarka, J. Modayil, and B. Kuipers, "Building Local Safety Maps for a Wheelchair Robot using Vision and Lasers," in *3rd Canadian Conference on Computer and Robot Vision*, 2006, p. 25.
- [9] R. Labayrade, C. Royere, D. Gruyer, and D. Aubert, "Cooperative Fusion for Multi-Obstacles Detection With Use of Stereovision and Laser Scanner," *Autonomous Robots*, vol. 19, pp. 117–140, 2005.
- [10] L. Xue-Cheng, K. Cheong-Yeen, S. G. Shuzhi, and A. Al Mamun, "Online map building for autonomous mobile robots by fusing laser and sonar data," *Mechatronics and Automation*, vol. 2, pp. 993–998, 2005.
- [11] M. Agrawal, K. Konolige, and R. C. Bolles, "Localization and Mapping for Autonomous Navigation in Outdoor Terrains: A Stereo Vision Approach," *Applications of Computer Vision*, vol. 7, 2007.
- [12] M. Bai, Y. Zhuang, and W. Wang, "Stereovision based obstacle detection approach for mobile robot navigation," in *International Conference on Intelligent Control and Information Processing (ICICIP)*, 2010, pp. 328–333.
- [13] L. Young-Chul, L. Chung-Hee, K. Soon, and J. Woo-Young, "Distance Estimation Algorithm for Both Long and Short Ranges Based on Stereo Vision System," in *Intelligent Vehicles Symposium*, 2008, pp. 841–846.
- [14] M. Polańczyk, P. Barański, and M. Strzelecki, "The application of Kalman filter in visual odometry for eliminating direction drift," in *International Conference on Signals and Electronic Systems*, 2010, pp. 131–134.
- [15] M. Polańczyk, A. Owczarek, M. Strzelecki, and K. Ślot, "Stereovision-based Obstacle Avoidance Procedure for Autonomous Mobile Platforms," *Proceedings of Hybrid Artificial Intelligent Systems*, vol. 1, pp. 206–213, 2011, Springer LNAI.
- [16] M. Polańczyk, M. Strzelecki, and K. Ślot, "Lee-algorithm based path replanner for dynamic environments," in *Proceedings of the International Conference on Signals and Electronic Systems*, Wrocław, Poland, 18–21 September 2012, CD ROM Edition.
- [17] P. Skulimowski and P. Strumiłło, "Surface detection in a stereo image sequence," *Image Process Techniques*, 2006, (in polish).
- [18] BumbleBee2, <http://www.ptgrey.com/products/bumblebee2/index.asp> (accessed November 2012).