

Application of Surface Reconstruction for Car Undercarriage Inspection

Martin Dörfler, Tomáš Pivoňka, Karel Košnar, and Libor Přeučil
Czech Technical University in Prague, CIIRC, Czech Republic
Email: {Martin.Dorfler, Tomas.Pivonka, Karel.Kosnar}@cvut.cz

Abstract—The method for camera-based 3D reconstruction of car undercarriages is proposed in this paper. It is designed for use in a special security scanner, which is placed under a road level and scans undercarriages of passing cars. The scanner uses mirrors to increase a distance from a camera and to simulate a stereo pair, thus capturing a stereo image by a single camera. The partial 3D models created from individual images are reconstructed by a correlation-based block-matching algorithm. Afterward, these models are transformed to common coordinate base according to visual odometry readings, and the individual pieces are clipped and stitched together to form a seamless model. The reconstruction works at near real-time speed and the complete process is fast enough to enable the inspection without substantial delay. The method was implemented on a prototype and successfully tested on real car undercarriages. During the test, the prototype was able to facilitate number of inspections and successfully detect foreign objects placed on the undercarriage.

Index Terms—stereo reconstruction, computer vision, visual odometry

I. INTRODUCTION

Vehicle inspection is a common part of the security procedures at key locations, such as military bases or entrances to strategic compounds. As part of the inspection routine, the undercarriage must be checked for anomalous devices placed by a malicious actor. In the *Kassandra* project and its previous iterations, a prototype of a camera-based scanner was developed to facilitate the inspection. The device is built into the checkpoint road, and it is able to scan the undercarriage and display it for the operator. A schematic of the system is depicted in Fig. 1.

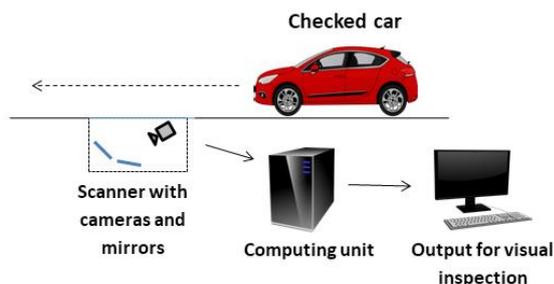


Figure 1. Scheme of the camera-based 3D scanner.

In typical inspections, several pairs of images are recorded as the car passes over the scanner. Dense stereo reconstruction is performed on each image pair, producing several partial 3D models of the undercarriage. These are composed in one unified model and presented to an operator for inspection.

Many complicating factors differentiate this task from the well-investigated problem of a general stereo reconstruction. Due to the proximity of the cameras to the target surface, the stereo pair is very wide, and the viewpoint difference complicates finding correspondences between images. The scanner has to be equipped with artificial lighting, as there is not sufficient natural lighting on the undercarriage. Furthermore, such lightning on the target surface is necessarily uneven and impacts both cameras differently due to the proximity. It is difficult to correctly compute a disparity map on the undercarriage because the surface naturally contains large homogeneous parts. The lack of both texture and detectable image features causes problems for most computer vision methods. The target application requires almost real-time performance. An additional complication is present in the physical realization of the system. The total height of the capturing device is limited because the installation costs would get prohibitively high otherwise.

This paper details the software part of the project, consisting of the camera controls and the algorithms for a reconstruction of the complete undercarriage picture. The history of the project and previous results are listed in Section II. The reconstruction algorithm is described in Section III. A brief overview of the system is given in subsection I.A to frame the context for the software problems, and subsection I.B shows the experimental results.

II. STATE OF THE ART

This work follows on from previous project *Kerberos* [1] focused on the development of a laser-based undercarriage scanner. The scanner consists of three laser rangefinders combined with cameras. The main advantage of this approach is the ability to reconstruct 3D shape regardless of an undercarriage texture. Two of the rangefinders with cameras are placed under the road level and scans an undercarriage of a car going across the scanner. The third rangefinder is placed horizontally, and it measures the car position. Further, the system contains

a camera for a reading of a registration number and display showing a car speed. The maximal car speed for detection is 15 km/h, and display helps a driver not to exceed maximal speed. Apart from creating a model, the system automatically detects changes on the undercarriage based on comparison with previous scans of the same car. This scanner is offered as a commercial product, and it is successfully installed in several strategical compounds in the Czech Republic.



Figure 2. Scanner installed in the road.

The main disadvantage of the Kerberos system is its complicated installation because it has to be placed under the road level (Fig. 2). A more lightweight scanner was developed in associated project Kyklop (in the commercial version, called Kerberos Mobile) [2]. The system works only with one camera, and it returns a 2D model of an undercarriage. It does not contain undercarriages comparison, but it serves well for human visual inspection. It replaces inspection with mirrors or hand cameras, which requires a stop of a vehicle. This system can be a part of the Kerberos system or an independent portable device.

Object detection [3] is a related task that aims to find known object or a class of object in the image. The relevant techniques are of limited applicability in this case, because the detected objects are not known beforehand. Instead, the scan of the undercarriage is compared with a previously stored record of the same vehicle. Image registration techniques (e.g. [4]) could be used at that point.

Visual odometry [5] is a measurement of a moving camera position from its images relative to a captured scene. It is an essential part of many SLAM (Simultaneous Localization and Mapping) systems. The main difference between SLAM and visual odometry is that SLAM contains loop closure detection and global optimization. In most robotic tasks a camera is placed on a vehicle and measures its position in world coordinates [6]. Contrarily, the presented system uses a static camera and measures a movement of a vehicle above it. The older version of the proposed visual odometry method is in detail described in [7], [8]. Even though the method was designed on data from regular cameras with larger images, it can be used in the scanner with mirrors only with slight modifications. The improved version is presented in this article in subsection III.B.

III. THE RECONSTRUCTION PIPELINE

This section describes algorithms for the reconstruction of the 3D model of the car undercarriage. It works with the sequences of stereo-images taken by cameras looking on the system of mirrors while the car is passing over the scanner. The configuration of the mirrors split the image into two virtual cameras forming the stereo pair as well as prolongs the optical distance of the camera from the object. The algorithm consists of several parts, which are described in the following subsections.

A. Pipeline Overview

The image processing pipeline is shown in Fig. 3. During each operation, several pairs of images are captured in sequence. A *visualodometry* (subsection III.B) is used to compute their relative position.

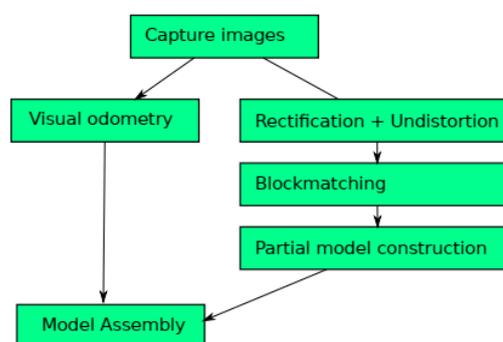


Figure 3. Scheme of the image processing pipeline.

Each pair of images is used to reconstruct a piece of a 3D model. The reconstruction is standard in principle. Most steps (image rectification, triangulation) relies on OpenCV [9] implementation. The *disparity map* is computed by correlation-based block-matching algorithm described in sections III.C and III.D.

The partial model pieces are transformed into the whole model space. By design, there is a significant overlap of the partial models. The final model is composed by trimming the overlapping sections and triangulating the space between resultant edges (Fig. 4).

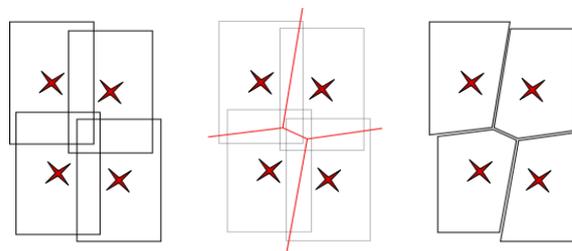


Figure 4. Clipping and assembly of partial models.

B. Visual Odometry

The proposed visual odometry method is a feature-based 3D-to-2D method according to the classification from [5]. In feature-based systems, a camera movement between two consecutive stereo-images is computed based-on visual feature matching. 3D-to-2D methods triangulate points only from one stereo image and searches for the best transformation, which reprojects

them to the second image with a minimal reprojection error. A pipeline of the visual odometry is depicted in Fig. 5.

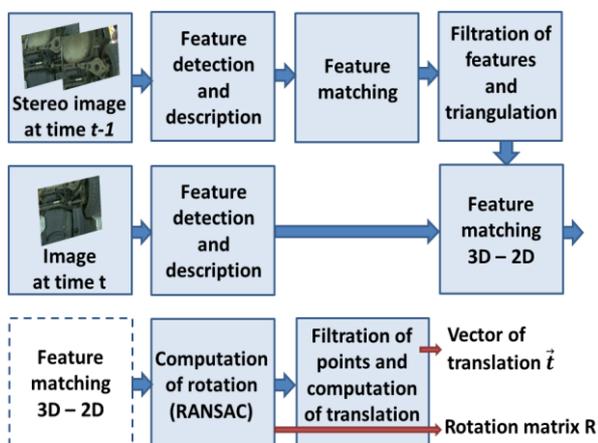


Figure 5. Scheme of visual odometry pipeline.

Inputs to the presented visual odometry method are stereo image in time $t-1$ and one image in time t (i.e. one from subsequent stereo image). Outputs are a rotation matrix and a vector of translation representing motion between stereo-images. All camera matrices and an essential matrix of stereo-camera are known from a camera calibration. The method is composed of following subsequent steps: detection and description of features, matching features in a stereo image and triangulation of points, matching 3D points and features in a third image, robust computation of rotation and computation of translation.

The most significant change from [8] is a replacement of the ORB feature detector and descriptor by AKAZE ones. Due to using of mirrors, the number of cameras was decreased to two and a whole stereo image is captured only by one camera. It caused a substantial decrease in image information and ORB was not able to match features between images successfully. Based on experimental testing on new experimental data, the AKAZE detector was selected.

In the first step, features from a stereo image in time $t-1$ are detected and matched. To ensure a uniform distribution of features, an image is divided into several rectangle areas for feature detection. Corresponding features are selected as features with the mutually closest descriptors. The correspondences are filtered out if the distance of one point from its epipolar line is larger than an assigned threshold. Afterward, 3D positions of points are computed. Points with larger or closer distance from the camera than assigned thresholds are removed. It is assumed that undercarriage distances are varying only in a fixed interval.

Filtered 3D points are matched to features in the third image to gain 3D to 2D correspondences. Both features of a 3D point are matched separately, and only features with the same closest feature in the third image are retained. If some features are too close in the image, only the feature with the strongest descriptor is kept.

Because some of the correspondences are still incorrect, the RANSAC algorithm is used. Rotation and translation are computed from 3 correspondences by P3P problem solver [10], which returns up to four possible solutions. The correct solution is selected by its testing for a 4th point. The output of RANSAC is a transformation with most of the points with reprojection error under an assigned level.

The translation is computed only for points satisfying RANSAC condition. Only half of them with reprojection error closest to a median is selected to create an overdetermined system of linear equations. The only unknown is particular elements of the translation vector because rotation is taken from the previous step. The least-square solution of the equations is the final vector of translation.

A transformation between not adjacent images is a concatenation of particular transformations. There is not used any bundle adjustment, loop closing or global optimization, because there is assumed that a movement of a car is straightforward and image overlapping between not adjacent images is minimal.

C. Surface Reconstruction

The disparity map is computed by a variant of the low texture stereo algorithm published in [11]. The algorithm as published was used for scene reconstruction in autonomous navigation applications, robot vision, and similar situations. It is designed to deal with difficulties in constructing models of walls, road segments and similar scenes where large areas with homogeneous texture are common, but also critical to reconstruct correctly. While specifics are different, the primary problem of large untextured areas in the input is present in our application as well. Although more sophisticated algorithms exist (e.g. [12]), strict limits on the running time of this application limit their use.

The principle of the algorithm is to compute the disparity map by block-matching, and subsequent filtering of the map by an adaptive window filter. Details of the steps are described below.

As a preprocessing, a Sobel filter is applied on the input to emphasize the texture and reduce noise in the images. The block-matching is performed to match pixels in both images. The original algorithm uses cosine distance as a cost function for block-matching. In order to deal with the illumination of reflective surfaces that compose a notable part of the undercarriage, *block correlation* (subsection 3.4) was employed instead.

The filtering is guided by edges detected in the original input data. Moving weighted average window filtering is applied on a horizontal and vertical line in the cost function, with reduced weight for pixels that are separated by a line from the window center. For more details, refer to the original paper [11].

In the first pass, a corresponding matching block is selected by a simple optimum of the filtered cost function. If necessary, the precision and details of the model may be increased in several successive steps, where the model is refined by recomputing the cost function with

increasingly smaller filter windows to capture fine details. In these refinement steps, a combination of non-maxima suppression and quadratic distance penalty is used to preserve high-scale shape and to limit updates to small or strongly salient ones.

$$Q_{dx} = \rho_{dx} - p_Q(dx - h)^2 - p_L|dx - h| \quad (1)$$

The matching block is selected by the following criteria. Only peak values in correlation for the current level are taken into account. Each peak is evaluated according to (1). The disparity for this pixel calculated in the previous pass is introduced as a hint h . The peak at the location of the hint has a full value of the correlation, all the others are penalized according to the distance from the hinted location. The intention here is to use the peak closest to h unless a significantly better candidate appears elsewhere.

D. Fast Block Correlation

The previous algorithm relies on a similarity measure for areas in the image called here *block correlation*, named in reference to block similarity measure used in other block-matching algorithms. The measure is defined in (2).

$$r_{xy} = \frac{\sum x_i y_i - n\bar{x}\bar{y}}{\sqrt{\sum x_i^2 - n\bar{x}^2} \sqrt{\sum y_i^2 - n\bar{y}^2}} \quad (2)$$

Efficient computation of this measure is a key to the efficacy of the reconstruction algorithm, as each block of the left image needs to be compared to a range of blocks in the right image, for a total of $x * y * dx$ computations. The block size also tends to be large. A dynamic programming formulation of the problem is used to keep the complexity manageable.

The block computation involves wide use of a sliding window algorithm. The images are parsed sequentially, and the sliding window is used for simultaneous computation of a running sum of several succeeding values, squares of these values, and multiplication of corresponding values in L and R (corresponding according to the range of scanned disparity). The sliding window is used again for each column to obtain the sums for each respective block. Thus we obtain $\sum_W L_{x,y}, \sum_W L_{x,y}^2, \sum_W R_{x,y}, \sum_W R_{x,y}^2 \wedge \sum_W L_{x,y} R_{x,y}$.

From these, we can obtain correlation matrix by (3). The computation can be performed in a single pass. The resulting algorithm has a complexity of $O(n * dx)$. Specifically, the computational complexity is not dependent on the window size.

$$C(x, y, d) = \frac{\sum L R - \frac{\sum L \sum R}{w^2 w^2}}{\sqrt{\sum L^2 - \frac{\sum L^2}{w^2}} \sqrt{\sum R^2 - \frac{\sum R^2}{w^2}}} \quad (3)$$

E. Model Assembly

The reconstruction algorithm detailed above computes a disparity map for each stereo pair. That can be converted by triangulation into a mesh of points in the coordinate system of the camera. In the final step, these meshes need to be joined into a complete model (Fig. 6).

Considering the model in the coordinate space of the target undercarriage, the visual odometry results give us the relative positions of the camera when taking each image. The partial meshes are shifted to their relevant position. By design, there is a significant overlap at the edges. The center of each model patch is computed, and the whole model is split into Voronoi cells according to the distance to the patch centers. Of each patch, only the points that are in its cell are kept (Fig. 4). The points on the cell edges are connected to fill in the gaps in the model.

F. Model Comparison

When the vehicle's model is completely reconstructed, and the system has a previous model of the same vehicle, these two models are compared, and spatial differences are highlighted for the human operator. The system identifies the vehicle by a registration plate. The reading of the registration plate is done by a separate system and is out of this paper's scope.

The resulting model is captured in a relative frame of reference of the first virtual camera. Therefore, the model reconstructed from each passage of the vehicle can be shifted according to the vehicle's exact trajectory over the scanner. The model registration is the first step as we need to align the models' positions to match each other. The most suitable approach is to use the Iterative Closest Point (ICP) alignment method. The model is relatively flat, as the length of the car is around 4 meters, and the depth of the car undercarriage is approximately 20 centimeters. Due to the flatness, it is beneficial to use the color information of the model for the model registration. We use the generalized color-supported iterative closest point [13] implementation from Point Cloud Library [14]. The color of the vertices is represented in $\mathbf{L} * \mathbf{a} * \mathbf{b}$ color space (or CIELAB), a color-opponent space with dimension \mathbf{L} for lightness and \mathbf{a} and \mathbf{b} for the color-opponent dimensions. As there is no relation between Euclidean spatial distance and distance in the color space, the color weight factor α is introduced. The significance of the color is controlled by this factor. Usage of the color increases the precision of alignment in the x-y plane, especially in the presence of reconstruction residual artifacts on edges of the model.

Two models are now aligned, and we want to detect spatial differences between them. As the size of the model is in orders of magnitude of millions of vertices, the comparison is made using the octree spatial structure implemented in Point Cloud Library [14]. All the vertices from the model created from the previous vehicle's passage are inserted into the octree structure. Then, we switch the octree buffer. It resets the octree but keeps the structure of the tree in the memory. Now, we insert the vertices from the currently made model. All the vertices from the octree voxels that are not presented in the first model are obtained. These detected vertices represent spatial differences. The size of the octree voxels significantly influences the computation speed and minimum size of detected change.

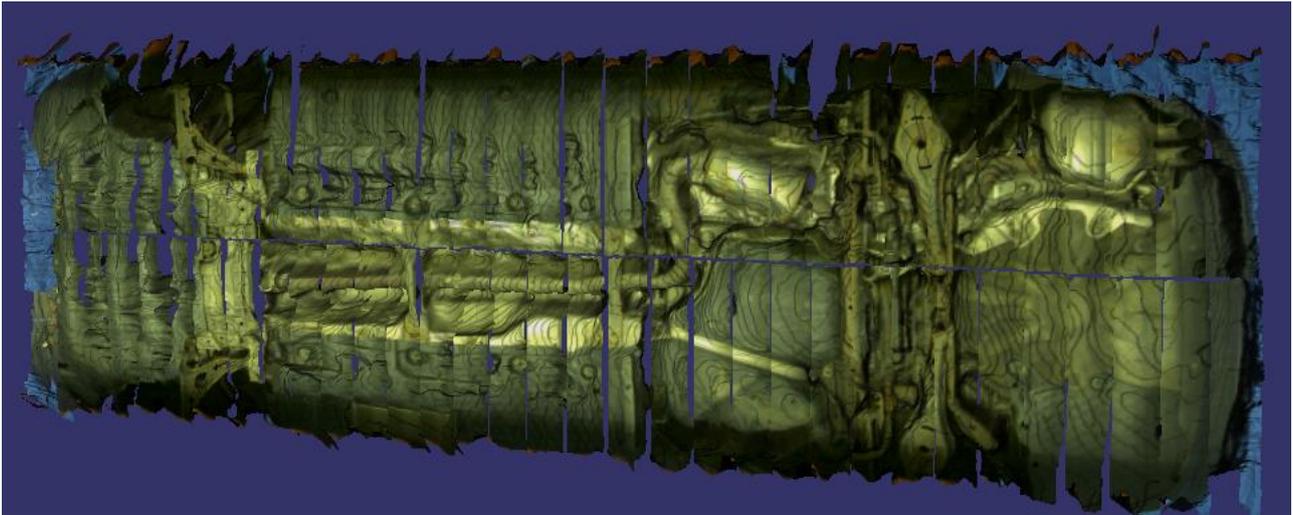


Figure 6. Assembled model of the undercarriage.

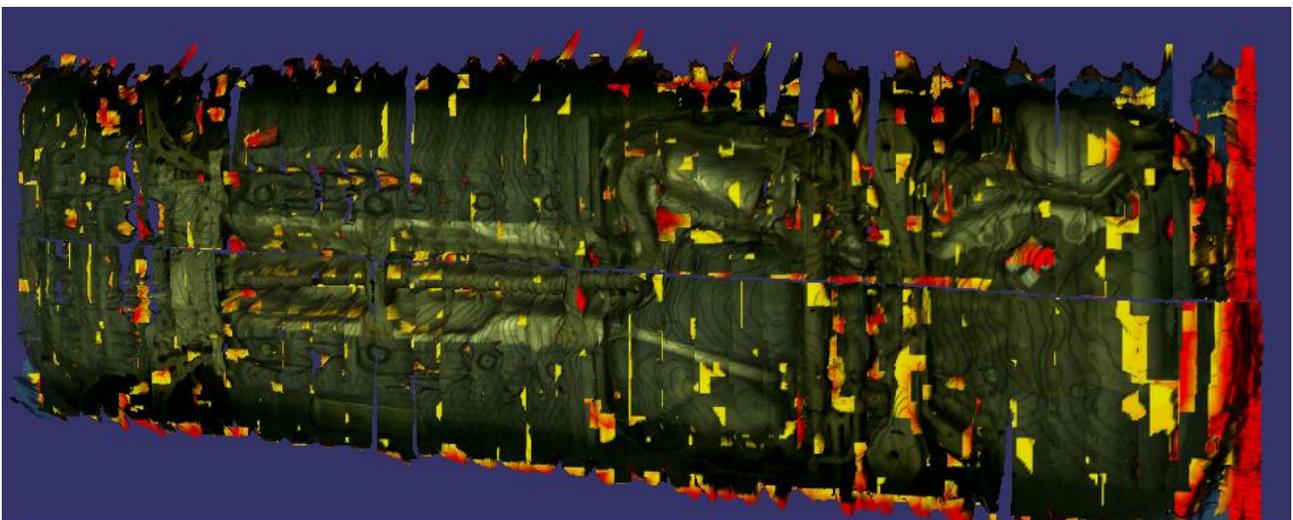


Figure 7. Comparison with the model differences highlighted.

We want to compute the significance of the change as well. For each vertex in the changed part, we calculate the distance to the previous model's surface. As the distance of the prior model is bigger, the change is more significant, and this significance is presented to the user in the form of a color map (Fig. 7).

IV. EXPERIMENTS

A. Hardware Prototype

The prototype of the scanner consists of two RGB cameras, mirrors and artificial lighting (Fig. 8). The camera resolution is 1920×1200 px, but each frame is divided by mirrors into two images 1920×530 px. In this way, the system needs only one physical camera for each stereo pair. The system with mirrors allows increasing an undercarriage distance from a camera with the same sensor depth because most of the necessary distance is situated parallel to the ground. Further, it is possible to protect the cameras from falling debris without intruding in their field of view. On the other hand, the field of view in the longitudinal direction is limited. The cameras were calibrated by [15].

The overlapping field of view of a camera pair covers one side of the undercarriage. Two pairs are needed to reliably cover the entire width. Fig. 9 shows an example of the gathered data. An example of the resultant disparity map is in the Fig. 10.



Figure 8. Device used to capture the images. Mirrors are used to increase effective distance from the target.



Figure 9. Stereo image of undercarriage captured by one camera and two mirrors.



Figure 10. Example of disparity map.

B. Reconstruction Results

The physical experiment was performed at the facility of the industrial partner in the project. The prototype (as described in subsection I.A) was placed in the service pit in the access road and used to gather several sets of data from a passing vehicle. In total, 28 scans were performed on two vehicles under different illumination conditions, both with additional objects placed on the undercarriage and without.

Tests confirmed that despite almost ideal weather conditions, strong artificial illumination is necessary for the scanner to operate correctly. The illumination as implemented in the prototype proved to be sufficient for the task.

The system was able to build a model of the undercarriage (Fig. 6). The level of detail was sufficient to enable the detection of introduced test objects, both visually and by comparing the resultant mesh.

V. CONCLUSION

The primary aim of the Kassandra project is to improve upon previous undercarriage scanner and deliver a device able to compute a 3D model of an undercarriage from camera images. In this paper, we present algorithms to construct a piecewise model of the undercarriage and assemble it into a complete model. The functionality of the prototype was verified on a testing site. The system is able to detect a vehicle passing over the scanner, make a 3D model of the complete undercarriage, and display it for the operator.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

Martin Dörfler was responsible for the reconstruction algorithms. Tomáš Pivoňka implemented the odometry computation. Karel Košnar was responsible for the GUI development, model comparison. Libor Přeučil was in the managerial role and overall leading the project on part of

the CTU. The experimentation was performed by cooperation of all the authors. The text was likewise collaborative effort, with each author covering their part of the solution.

ACKNOWLEDGEMENT

The project is solved in collaboration with VOP CZ, s.p. and it is supported by the Ministry of the Interior of the Czech Republic under the project number VI2VS/461 and the Grant Agency of the CTU (No. SGS18/206/OHK3/3T/37).

REFERENCES

- [1] K. Košnar, T. Krajník, and L. Přeučil. (2013). Kerberos - 3D undercarriage inspection system. [Online]. Available: <http://imr.ciirc.cvut.cz/uploads/Industry/kerberos.pdf>
- [2] Kerberos mobile wireless. [Online]. Available: http://www.vop-security.cz/eng/assets/kerberos_mobile_eng.pdf
- [3] E. Bostanci and B. Bostanci, "Object localization and spatial analysis using computer vision," *International Journal of Machine Learning and Computing*, vol. 1, pp. 120-124, 2011.
- [4] E. H. KalatehJari, M. M. Hosseini, and A. A. Gharahbagh, "Image registration based on a novel enhanced scale invariant geometrical feature," *International Journal of Machine Learning and Computing*, vol. 2, no. 5, p. 667, 2012.
- [5] D. Scaramuzza and F. Fraundorfer, "Visual odometry [tutorial]," *IEEE Robotics Automation Magazine*, vol. 18, no. 4, pp. 80-92, Dec. 2011.
- [6] I. Cvišić, J. Cesić, I. Marković, and I. Petrović, "SOFT-SLAM: Computationally efficient stereo visual simultaneous localization and mapping for autonomous unmanned aerial vehicles," *Journal of Field Robotics*, vol. 35, no. 4, pp. 578-595, 2018.
- [7] T. Pivoňka, "Automatic test of visual feature detectors and descriptors for visual odometry," in *Proce. the International Student Scientific Conference Poster*, 2019, pp. 198-201.
- [8] T. Pivoňka, K. Košnar, M. Dörfler, and L. Přeučil, "Visual odometry for vehicles' undercarriage 3d modelling," in *Modelling and Simulation for Autonomous Systems*. J. Mazal, Ed., Springer International Publishing, 2019, pp. 111-120.
- [9] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.
- [10] L. Kneip, D. Scaramuzza, and R. Siegwart, "A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation," in *Proc. CVPR*, June 2011, pp. 2969-2976.
- [11] L. T. Sach, K. Atsuta, K. Hamamoto, and S. Kondo, "A robust stereo matching method for low texture stereo images," in *Proc. IEEE-RIVF International Conference on Computing and Communication Technologies*, July 2009, pp. 1-8.
- [12] M. Li, B. Guo, and W. Zhang, "An occlusion detection algorithm for 3d texture reconstruction of multi-view images," *International Journal of Machine Learning and Computing*, vol. 7, pp. 152-155, 2017.
- [13] M. Korn, M. Holzkoth and J. Pauli, "Color supported generalized-ICP," in *Proc. International Conference on Computer Vision Theory and Applications*, Lisbon, 2014, pp. 592-599.
- [14] R. B. Rusu and S. Cousins, "3D is here: Point Cloud Library (PCL)," in *Proc. IEEE International Conference on Robotics and Automation*, Shanghai, 2011, pp. 1-4.
- [15] A. Richardson, J. Strom, and E. Olson, "AprilCal: Assisted and repeatable camera calibration," in *Proc. the IEEE/RSJ International Conference on Intelligent Robots and Systems*, November 2013.

Copyright © 2021 by the authors. This is an open access article distributed under the Creative Commons Attribution License (CC BY-NC-ND 4.0), which permits use, distribution and reproduction in any medium, provided that the article is properly cited, the use is non-commercial and no modifications or adaptations are made.



Martin Dörfler has received his master's degree in theoretical computer science at the Faculty of Math & Physics of the Charles University. Currently he is working on his PHD in Intelligent and Mobile Robotics group of Czech Institute of Informatics, Robotics and Cybernetics, CTU. His research interests are mobile robot localization, navigation, and computer vision.



Tomáš Pivoňka has received his master's degree in robotics at Faculty of Electrical Engineering of Czech Technical University in Prague (CTU) in 2018, where he continues in Ph.D. study program Artificial Intelligence and Biocybernetics. He works at the Intelligent and Mobile Robotics Group of Czech Institute of Informatics, Robotics and Cybernetics, CTU. His main research interests are visual localization, navigation, and computer vision.



Dr. Karel Košnar is currently an assistant professor at the Czech Institute of Informatics, Cybernetics, and Robotics, Czech Technical University (CTU) in Prague. He obtained the Ing. (=MSc.) degree in Technical Cybernetics in 2004 and Ph.D. in Artificial Intelligence in 2011 both from Faculty of electrical engineering, Czech Technical University in Prague.

Recently, he spends 10 months at Technische Universität Berlin. He is an author or co-author of more than 20 publications on international conferences and journals. His H-index is 5 according Scopus with 121 citations (70 citations according WOS).

Karel Košnar was a principal investigator of two national projects supported by MEYS and Czech Science Foundation. He participated also on several European and national projects including EU FP7 and H2020 projects. His research interests include scene understanding and spatial cognition for mobile robots.



Dr. Libor Přeučil obtained his MEE (Ing.) in Control Engineering -Robotics in 1985, Ph.D. (CSc.) in Computer Vision and Medical Image Processing in 1993 both in Technical Cybernetics from the Czech Technical University in Prague (CTU).

Currently, he is assistant professor, heading Intelligent and Mobile Robotics (IMR <http://imr.ciirc.cvut.cz>) laboratory and recently 2013 also co-founder of Center for Advanced

Field Robotics bringing together major mobile robotics research labs and industry across the Czech Republic. He coordinated FET IST project on Building Presence thru Localization for Hybrid Telematic Entities (PeLoTe) and acted as the national coordinator of the 6FP European Robotic Network of Excellence (EURON2) and was responsible for 2 Integrated Projects within the EC Framework Programme 7 on evolutionary robotics and bio-inspired robotics entitled SYMBRION and REPLICATOR and a H2020 project in robotics-posed advanced logistics entitled SafeLog (2016-2020). Libor Přeučil's research interests are posed into the area of intelligent and autonomous robotics -- control of self-guided autonomous vehicles, uncertain sensory information processing and sensor fusion technology for UAVs and UGVs -- primarily targeting robot localization, navigation and mapping and map keeping for uncontrolled, human-oriented, natural and urban spaces with no support infrastructures.

Dr. Přeučil's interests also comprise multi-robot and swarm systems, cover hybrid human-robot issues, etc. Besides 250+ papers at conferences, Best Paper Award (co-author in 2012, 2014, 2015), Libor Přeučil has 56 papers in high-profile impacted journals in the respective fields of Robotics, Autonomous Systems, Cyber-Physical Systems and neighboring areas with 166 citations (excluding self-citations and with average citations per item at 2.96) yielding the H-index of 7 (WoS).