Vision-based mobile robot localisation and mapping system with Kinect

László Somlyai, Zoltán Vámossy

***Abstract*—**RGBD cameras are increasingly used for mobile robot developments recently. Data of the sensor can be used for creating maps during the navigation and localization made on this. The robot system described in the article is able to create a map in an unknown environment during its navigation and localize its state on it parallel. The system was placed on a modified RC robot car. The process is able to evaluate movement of the vehicle in real time using data from the Kinect sensor placed on the robot and it continuously updates this map using new information. Pictures made by the camera during movement of the system are overlapped with each other. Evaluation of movement is based on join cohesive feature point pairs in each three dimensional point clouds. Definition of transformation between cohesive three dimensional feature points is based on an SVD algorithm. During a multistage iteration, the system minimizes the join fault and deletes point pairs detected wrong.

***Keywords*—**mobile robot navigation, mapping, three dimensional reconstruction, SLAM, Kinect, RGB-D camera, SVD

I. Introduction

THE autonomous mobile robots are performing a specified task during their mission. It is necessary to know its position, obstacles in its environment and the goal position. During the navigation, it senses the environment and the obstacles with its sensors and should be built a map from the travelled area. Such navigation systems are called simultaneous localization and mapping by literature. Shortly: SLAM

Indoor one of the positioning chance is recognizing the specific objects. With the help of camera feature objects, points can be recognised which positions are known. Then with the help of this information position of the robot can be defined. A possible solution is that feature objects, QR codes [1], [2] are placed, which spatial position is accurate. A camera watches these feature objects and defines the position of the vehicle. Know the map of the environment is not necessary in every case. For example, in the case of warehousing robots, a line is painted between the starting and target point [8] which gives the route of the vehicle.

This work was supported by John von Neumann Faculty of Informatics at Obuda University.

L. Somlyai is with the Software Engineering Department of Obuda University, Bécsi út 96/b, H-1034 Budapest (phone: +36-1-666-5549; e-mail: somlyai.laszlo@nik.uni-obuda.hu).

Z. Vámossy is with the Software Engineering Department of Obuda University, Bécsi út 96/b, H-1034 Budapest (phone: +36-1-666-5573; e-mail: vamossy.zoltan@nik.uni-obuda.hu).

In the case of indoor navigation global positioning systems are not reliable, like GPS system [3], [4], [5]. For indoor navigation inertial sensors or other displacement meters can be used because the pre-installed infrastructure is not necessary. A system built up this way is able to evaluate its own displacement with the help of accelerometers.

Earlier a robot vehicle [6] was presented in an article, where localization and map construction of the robot happened only by data of the displacement sensors fixed to the wheel. The main problem was the accumulation of faults on the path of the robot.

Nowadays relatively cheap and small sized sensors [7] (Kinect, Asus Xtion, PrimeSense, etc.) can be found in more and more robot development. Despite their simpler make, they have enough accuracy for sensing the environment of a smaller robot. With the help of sensor, a detailed three dimensional map can be constructed from the environment of the vehicle. The RGB-D cameras are used for SLAM algorithms and three dimensional map construction as a primary sensor [9].

The same element of the works is that displacements between pictures taken in different moments are defined by searching feature points on pictures of the sensors colorful camera. One of the research deals with the three dimensional mapping of the indoor environment [10]. It is built on the sensor called PrimeSense which is very similar to the Kinect sensor. Feature searching happens with SIFT detector. Its disadvantage is that the complex feature searcher has big running time. Spatial locations of features are known because of data of the depth camera. In this way, transformation among conglomerations taken at different times can be defined. An algorithm named RGBD-ICP was developed for joining. Finally, the system clarifies the position with close-loop algorithms on the global map.

The project called "Visual Odometry and Mapping for Autonomous Flight Using an RGB-D Camera" [11] presents a method where controlling of the plane happens by data of an RGB-D camera. Searching of feature points happens by FAST feature point detector. The algorithm is less complex, it has small running time, but it is less invariant for transformations. Inertial system (IMU) is used for evaluation of early displacement. The method mentioned in the "Combining photometric and depth data for lightweight and robust visual odometry" [12] also uses FAST feature detector. The accuracy of the algorithm was corrected by following feature points. Another paper [13] compares three robust feature searching methods in one SLAM system.

II. BUILDING UP THE IMPLEMENTED SYSTEM

The developed system can be placed on a smaller mobile robot as well (Fig. 1). A two-wheel drive mobile robot platform has been prepared for testing the system and collecting measurement data. The Kinect sensor and the computer were placed on the platform.

For motor controlling an own motor control electronics was made, which was connected to the computer on a serial port. For final the Kinect sensor was posted to the robot.

Fig. 1. Redone robot car to create own data sets.

For movement following the system uses only the colorful and pixel level depth pictures of the camera. Definition of movement is based on point clouds joining were made during each measurement.

During system operation (Fig. 2) measurements are made continuously with Kinect sensor, while the vehicle is moving. Each measurement made from the actual environment of the sensor (colorful camera picture and pixel level depth information) gives a three dimensional point cloud what is included by the X_n set. The point cloud made at the new place can be joined to the former measurement, if the sensor displaced just a little bit during. During joining displacement is defined between former point clouds and newer measurement. Always the last measurement is given to the global map this way. Measurement faults are largely accumulated to each other.

The algorithm uses three dimensional feature points for joining point clouds made in different times. The point of the method is that searching of feature points happens on the colorful camera picture, which situation is known with the help of the RGB-D cameras data. Using these feature points, a spatial displacement can be defined among measurements made at different places. Estimate of displacement is based on SVD resolution among the set of points.

The three dimensional point cloud is reduced to a simpler two dimensional map during navigation. Only barriers at the level of robots height are showed on the map. A standard route searching algorithms can be used on these simpler maps (wave propagation [6], A*, fuzzy-based obstacle avoidance [22], etc.) for navigation.

Another application of the system is the displacement data are used for joints only. During the measurements position of the sensor is given by accumulated value of specified elemental shifts, compared to the power point. Beside this kind of utilization, we get an IMU unit which relies on visual elements (RGBD-IMU).

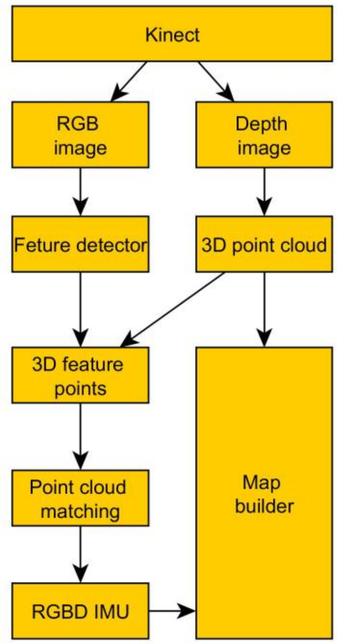

Fig. 2. Graph of the map builder system

III. FEATURE SEARCH

More researches deal with the topic of feature based three dimensional join. The main difference is that feature detector is used because they react for different picture transformations in different ways. From the comparison of feature detectors, the SIFT detector gives the best result for scaling, rotating transformations and for blurriness [17].

One of the developments chose the SIFT detector [10], its disadvantage is that this robust feature searcher has huge running time. The FAST algorithm is a less robust detector, it has small running time. Its disadvantage is that it is less invariant for transformations [11]. Some known feature point detector what have been tried by their efficiency and speed: SIFT [14], SURF [15], ORB [16], FAST. Because of its robust operation and fastness we chose the SURF detector. One paper shows a method for the thermal image matching [18].

On each new measurement data, a preprocessing runs for the fast definition of the spatial position of feature points. During the transformation of depth cameras pixels, every depth points are indexed by their pixel pair on colorful camera picture. In this way, if we are searching for depth information to feature point on a colorful camera picture, it can be found in a short time owing to indexing. Feature points found on the colorful camera picture are in the feature set. In this set their position on the colorful camera picture and spatial position in the depth cameras coordination system are stored. During preprocessing every spatial point is calculated, but it is not a lot of time because the algorithm builds map and three dimensional model beside localization.

IV. THREE DIMENSIONAL FEATURE POINT MATCHING SYSTEM

During map construction, the task is the definition of transformation between point cloud from the actual measurement and reference points of the system. The algorithm selects feature points from the points of actual and former measurement. Later the former measurement D and the incorporated (actual measurement), the feature set S consists the list of corresponding points.

Calculate of the transformation between feature points (T) is made by a multi-stage joining method. In the algorithm, the task is to define a transformation which gives the minimal joining fault between two sets (1).

$$\arg\min_T \sum_i \| D - T * S \| \tag{1}$$

During steps (Fig. 3) transformation between sets is given by the *ISVD(D,S)* algorithm. The algorithm originates from an initial transformation. During iterations, the algorithm deletes those points which Euclidean distance is bigger than a specified error (e_n) using the actual transformations. The e_n iteration is the biggest allowed difference between joined points.

In the first step the algorithm filters feature points with bigger faults. The e_n gives smaller value in every iteration. The base of the algorithm is the SVD resolution [21], its input is the two point sets and an actual estimated transformation. The algorithm defines the optimal transformation between two point sets by SVD resolution and then determines the translation vector from the center of gravity of sets. It is repeated during more steps. The ISVD algorithm stops when a maximum iteration number is reached, or the required accuracy of join is reached.

During multi-step joining those point pairs which look like belonging together but they are not are deleted by the system. It can be seen in the Fig. 4 that there are a lot of fault false point pairs because of the homogeneous surface on pictures made one after another. During join these false feature point pairs can be deleted using that spatial location is known. The reduced, coherent point can be seen on the right side of the figure.

The Table I illustrates how reduces the number of coherent point pairs in each iteration during joint of two point sets. It gives the running time and quality of joint as well. In this case the maximum allowed squared error was 800 in first iteration and it was 120, 80, 45 in the next iterations. An estimated displacement is defined during joining of the former (*ISVD*($X_n;X_{n-1}$)) and the previous (*ISVD*($X_n;X_{n-2}$)) feature points. The algorithm reduced almost 200 feature points to less than 50 during 4 iterations. In the table joint happened to the former and the previous picture. The number of pictures for joining can be adjustable in the system.

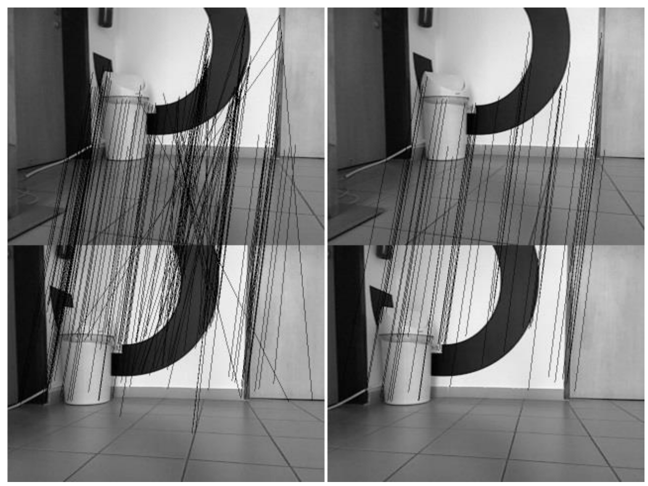

Fig. 4. Results of the standard feature matching. Left image: result of SURF, 52 pieces of feature pair. Right image: result after the matching algorithm, remained 39 pieces of feature pair.

There is enough overlap on some pictures foregoing actual measurement for joining.

For joining two point pairs one metrics is assigned (μ), what gives the quality of a given joint during an estimated T transformation and between D set and S set (2). If the measurement is 0, the joint is unsuccessful. It gives the bigger value, the quality of joint is better.

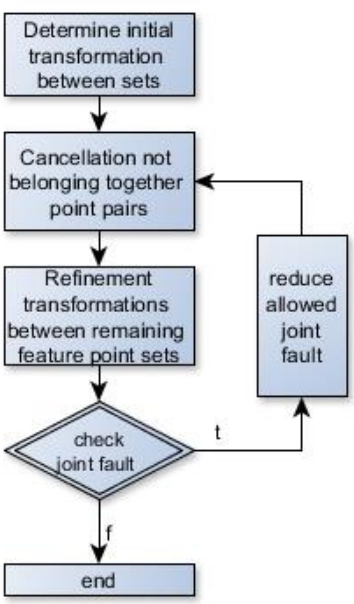

Fig. 3. Steps of the multi-stage joining method

		SURF feature pair	Iterations			
			1.	2.	3.	4.
ISVD(X_n,X_{n-1})	Remaining 3D feature (bit)	217	78	69	63	57
	Runtime (ms)	29	9	10	8	10
	Fitting quality (μ)		1.6	1.4	3.1	4.2
ISVD(X_n,X_{n-2})	Remaining 3D feature (bit)	207	69	65	59	49
	Runtime (ms)	28	8	9	8	7
	Fitting quality (μ)		1.8	1.7	2.8	3.2

Table 1. Runtime of ISVD algorithm

$$\mu = \frac{\sum_{i=0}^{|D|}\left(\sqrt[2]{\left(d_{ix}-s'_{ix}\right)^2+\left(d_{iy}-s'_{iy}\right)^2+\left(d_{iz}-s'_{iz}\right)^2}\right)}{|D|} \qquad (2)$$

$$, where \quad S'=T*S$$

One advantage of the algorithm is that searching of transformation happens among point pairs earlier founded. In this way the number of algorithms iteration are significantly reduced. Table II shows the accumulated fault after 60 pieces three dimensional alignment (Fig. 5). The sensor was turned in 360 degrees during measurement. With different methods in case of same feature detector and parameters it can be seen that the rotational and translation faults are smaller at ISVD algorithm, than using a standard ICP algorithm [10], [19].

V. TESTING OF ALGORITHM

Fig. 5. Offical test environment. The sensor was turned in 360 degrees during measurement.

The earlier presented methods were tested on some own data sets. In case of own measurements the real spatial position of the sensor is not available. The starting position and the target position is the same at every measurement. The robot ran a whole circle in the laboratory during measurements, or the sensor was rotated in 360 degrees on datasets, in this way its first and the last position is the same. At these data piles the examination of the algorithm is concentrated only on faults accumulated on whole area ran by the robot. The sensor goes back to the starting point in every case, its accurate route is unknown. On the next figure the measurement datasets two and three dimensional reconstruction can be seen (Fig. 6, Fig. 7). The reconstruction was made by the own method. On this datasets the robot ran a whole circle in the laboratory. During this time it created 310 pictures. All accumulated translation error was x = 250mm, y =2.5mm, z = 15mm and the rotation error was ϕ = 0.9, Θ = 1, ψ = 9 degrees.

	ICP	ISVD
ϕ [rad]	-0.0365	-0.0336
Θ [rad]	0.0019	0.0058
ψ [rad]	-0.1124	-0.0404
x' [mm]	-183.9	-99.2
y' [mm]	617.4	380.3
z' [mm]	-231.1	-264.7
Rot. error [rad]	0.05	0.026
Tr. error [mm]	343.8	248

Table 2. Comparison of ICP and ISVD algorithm

Testing of algorithms was happened on an Intel Core 2 Quad, 2.66 GHz, 8GB RAM desktop computer. In this computer the average running time of tests was 180ms at joints (Table III). Running time of the algorithm has three main parts: searching feature points in every new picture, the creation of three dimensional point cloud as well as these

	Full processing time (ms)	Feature (ms)	3D point cloud (ms)	Fitting (ms)
orb2	143	19	11	113
surf1	175	121	11	43
surf2	183	118	11	54
surf4	221	119	11	91

Table 3. Runtime of the point cloud matching

points are indexed by their position on the colourful camera and definition of transformation among the set of points. Running time at the definition of transformation among the set of points is influenced by pictures jointed to the number of former measurements. The first step is to find feature pair between the current and former pictures, then the ISVD algorithm defines a transformation based on feature pairs founded.

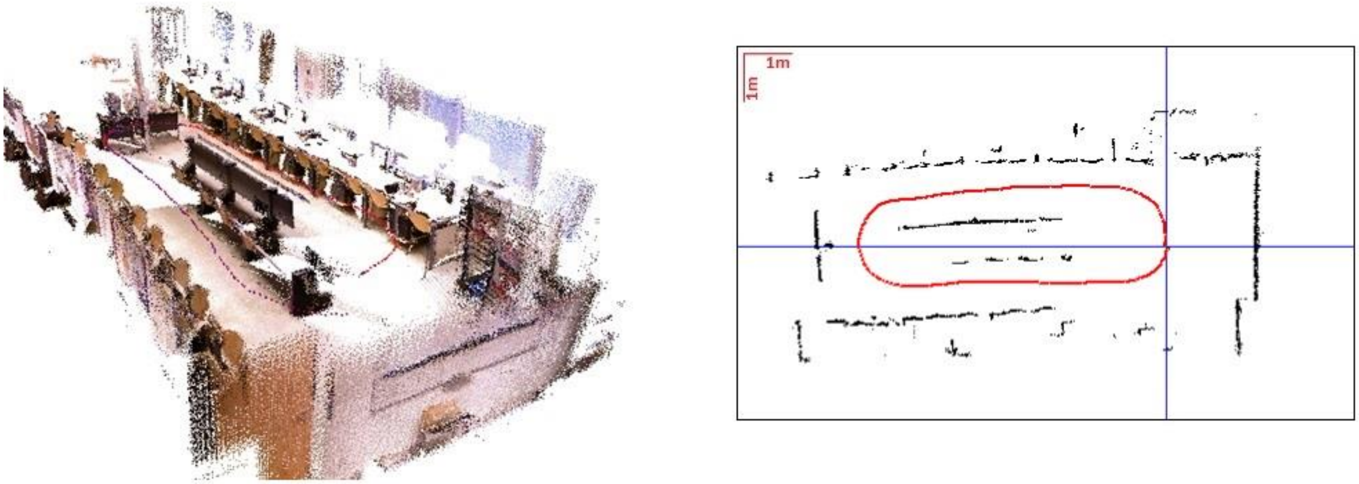

Fig. 6. Result of the algorithm on own datasets. This dataset includes 310 image pair. The image shows three dimensional and two dimensional maps from one laboratory of the university.

VI. CONCLUSIONS

For feature detector the SURF detector was chosen as a result of testing. The system makes the feature searching at every new picture and the conversion of depth camera into three dimensional points. It will be indexed based on the place

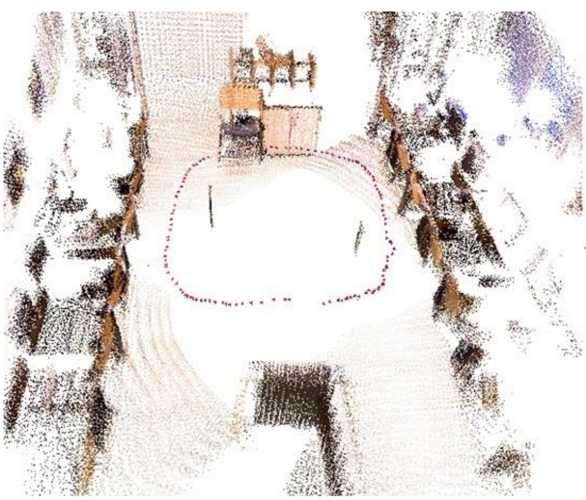

Fig. 7. Laboratory test environment. The robot ran a whole circle on laboratory and created 250 images.

on colorful camera picture, in this way the running time of the later processing is significantly decreased. The system was tested on own datasets. The ISVD algorithm made for joining searches the best transformation between two point clouds made sequentially during more iteration. The algorithm deletes the not cohesive point pairs during its iteration. In this way it gives a more accurate evaluation of transformations between two sets. This method is able to filter fault feature pairs on recurring areas.

During testing good results were reached with the help of SURF feature detector. During experiments the average joining displacement fault was 9.9 mm and rotation error was 0.38 degrees. The best result was when the joining happened to 4 or 8 earlier picture and the final displacement was determined based on weighted average of given results. During 4 earlier measurements average running time of the algorithm was 220 ms, which allows four and half picture joining per second. It is fast processing time in case of similar robust detectors [10], [13]. During system testing further datasets were made beside examinations made in laboratories. One result ran on this dataset can be seen in Fig. 8. The data set was made in a multi-roomed flat. The figure shows the three dimensional reconstruction of the environment with the route of the vehicle.

Fig. 8. A large 3D reconstruction from a flat.

REFERENCES

[1] Hiroyuki Kobayashi, A new proposal for self-localization of mobile robot by self-contained 2D barcode landmark SICE Annual Conference, pp. 2080-2083, Akita, Japan, 2012

[2] Gijeong Jang, Sungho Kim, Wangheon Lee, Inso Kweon, Color Landmark Based Self-Localization for Indoor Mobile Robots Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference, Vol 1., ISBN: 0-7803-7272-7, pp. 1037-1042, Washington, 2002

[3] Molnar A., Vamossy Z., Navigation of a GPS Based Robot Vehicle In: Proceedings of the 13th International Workshop on Robotics in Alpe-Adria-Danube Region, pp. 408-413, ISBN: 80-7204-341-2, Brno, Csehszlovakia, 2004

[4] Stojcsics D., Molnar A., Fixed-wing small-size UAV navigation methods with HIL simulation for AERObot autopilot In: Proceedings of SISY 2011, 9th IEEE International Symposium on Intelligent Systems and Informatics, pp. 241-245, ISBN: 978-1-4577-1975-2, Budapest, 2011

[5] D. Stojcsics, L. Somlyai, A. Molnár Unmanned aerial vehicle guidance methods for 3D aerial image reconstruction ICCC 2013, Proceedings of IEEE 9th International Conference on Computational Cybernetics, ISBN: 978-1-4799-0060-2, pp. 321-326., Tihany, Hungary, 2013

[6] Csaba Gy., Somlyai L., Vamossy Z., Differences Between Kinect and Structured Lighting Sensor in Robot Navigation In: Proc. SAMI 2012, 10th IEEE Jubilee International Symposium on Applied Machine Intelligence and Informatics, pp. 85–90, ISBN: 978-1-4577-0195-5, IEEE Catalog Number CFP1208E-CDR, Herl'any, Slovakia, January 26-28, 2012

[7] H. Gonzalez-Jorge, B. Riveiro, E. Vazquez-Fernandez, J. Martínez-Sánchez, P. Arias, Metrological evaluation of Microsoft Kinect and Asus Xtion sensors In: Journal of the International Measurement Confederation, pp. 1800-1806, ISSN: 0263-2241, 2013

[8] Jackrit Suthakorn, Andrew B. Cushing, Gregory S. Chirikjian, An Autonomous Self-Replicating Robotic System Advanced Intelligent Mechatronics, Proceedings IEEE/ASME International Conference on, Vol 1., pp. 137-142, DOI:10.1109/AIM.2003.1225085, Japan, 2003

[9] Scaramuzza, D. and Fraundorfer, F., Part I the first 30 years and fundamentals IEEE Robotics & Automation Magazine Vol. 18, No. 4, pp. 80-92, 2011

[10] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox., RGB-D Mapping: Using Depth Cameras for Dense 3D Modeling of Indoor Environments Proc. of the International Symposium on Experimental Robotics (ISER), 2010

[11] A. Huang , A. Bachrach , P. Henry , M. Krainin , D. Maturana , D. Fox , N. Roy, Visual Odometry and Mapping for Autonomous Flight Using an RGB-D Camera Int. Symposium on Robotics Research (ISRR), 2011

[12] M. Nowicki, P. Skrzypezynski, Combining photometric and depth data for lightweight and robust visual odometry Mobile Robots (ECMR), 2013 European Conference on, DOI: 10.1109/ECMR.2013.6698831, pp. 125 - 130., Barcelona, 2013

[13] F. Endres, J. Hess, N. Engelhard, J. Sturm, D. Cremers, W. Burgard An evaluation of the RGB-D SLAM system Proc. IEEE Int. Conf. on Robot. and Automat., pp. 1691–1696., St. Paul, 2012

[14] David G. Lowe, Distinctive image features from scale-invariant keypoints International Journal of Computer Vision, pp. 9-110, 2004

[15] Herbert Bay, Andreas Ess, Speeded-UpRobustFeatures (SURF) Computer Vision and Image Understanding, Vol. 110, Issue 3, pp. 346-359, 2008

[16] Rublee, E., Rabaud, V., Konolige, K. and Bradski, G. ORB: an efficient alternative to SIFT or SURF Proc. IEEE Int. Conf. on Computer Vision (ICCV), pp. 2564–2571., 2011

[17] Luo Juan, Oubong Gwun, A Comparison of SIFT, PCA-SIFT and SURF International Journal of Image Processing (IJIP), Vol. 3, Issue 4, pp. 143-152, 2010

[18] Vamossy, Z. Thermal image fusion Intelligent Systems and Informatics (SISY), 2012 IEEE 10th Jubilee International Symposium on, pp. 385-388, E-ISBN: 978-1-4673-4749-5, DOI: 10.1109/SISY.2012.6339549, Subotica, 2012

[19] S. Seeger, X. Laboureux, G. Hgausler, An Accelerated ICP-Algorithm Optical Sensing, Metrology and INspection, Annual Report, pp. 32, 2001

[20] Adam Schmidt, Michał Fularz, Marek Kraft, Andrzej Kasi´nski, Michał Nowicki An Indoor RGB-D Dataset for the Evaluation of Robot Navigation Algorithms Prof. of Int. Conf. on Advances Concepts for Intelligent Vision Systems, Lecture Notes in Computer Science Vol. 8192, pp. 321-329

[21] K. S. Arun, T. S. Huang, S. Blostein Least-squares fitting of two 3-D point sets IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 9, No. 5, pp. 698–700., 1987

[22] Gy. Csaba Improvement of an Adaptive Fuzzy-Based Obstacle Avoidance algorithm using Virtual and Real Kinect Sensors ICCC 2013, Proceedings of IEEE 9th International Conference on Computational Cybernetics, ISBN: 978-1-4799-0060-2, Tihany, Hungary, 2013

László Somlyai was born in Kaposvar, Hungary on 14 April 1987, and received his BSc degree in 2009 on ELECTRICAL ENGINEER and he received his MSc degree in 2011 on INFORMATICS at Obuda University in Budapest, Hungary. He started Ph.D. research in 2011 at the APPLIED COMPUTER SCIENCE Doctoral School on same University. The title of his Ph.D. thesis is "Realtime mapping of the Mobile robots environments". Since 2009 he is an engineer at Software Engineering Department of Obuda University in Budapest, Hungary. He is responsible for the introductory course in electronics and digital technics. His research interests are autonomous mobile robots, computer vision, simultaneous localization and mapping, embedded electronics systems.

Zoltán Vámossy was born in Budapest, Hungary on 12 January 1960, and received M.Sc. degree in 1984 on specialization mathematics in FACULTY OF MECHANICAL ENGINEERING at the Technical University of Budapest, Hungary. He received the Ph.D. degree in 2010 at the KALMAN KANDO DOCTORATE SCHOOL OF MULTIDISCIPLINARY SCIENCES IN ENGINEERING of Budapest University of Technology and Economics, Hungary. The title of his Ph.D. thesis was "Mobil Robot Navigation Using PAL Optics Based Computer Vision".

After receiving the M.Sc. degree, he worked as a software engineer at the VIDEOTON Research Institute. Since 1989 he works at Obuda University in different positions. He taught programming languages, robotics and image processing. Since 2004 he is an associate professor at Software Engineering Department of Obuda University in Budapest, Hungary, and is responsible for a specialization that focuses computer vision and robotics. His supervised students won more than thirty prizes at national students' contest. He was an invited lecturer the Intensive Program on Computer Vision and Summer School of Image Processing summer courses. His research interest focuses on mobile robot navigation using computer vision and he published more than hundred papers.

Dr. Vámossy is the member of IEEE. He is a reviewer of several journals and conferences. In 2004 he was awarded by the Hungarian Chief Information Officers' Association: "IT trainer of the year" and he also got the "Master Teacher Gold Medal award" from the National Council of Student Research Societies.