dr inż. Janusz Będkowski
prof. dr hab. inż. Andrzej Masłowski
Instytut Automatyki i Robotyki, Politechnika Warszawska

„SEMANTIC SIMULATION ENGINE" FOR MOBILE ROBOTIC APPLICATIONS

In the paper the „Semantic Simulation Engine" dedicated for mobile robotics applications is shown. Presented software performs mobile robot simulation in virtual environment built from real 3D data that is transformed into semantic map. Data acquisition is done by real mobile robot PIONEER 3AT equipped with 3D laser measurement system. Semantic map building method and its transformation into simulation model (NVIDIA PhysX) is described. The modification of ICP (Iterative Closest Point) algorithm for data registration based on processor GPGPU CUDA (Compute Unified Device Architecture) is shown. The semantic map definition is given including the set of semantic entities and set of relations between them. Methods for localization and identification of semantic entities in 3D cloud of points based on image processing techniques are described. Results and examples of semantic simulation are shown.

SYSTEM SYMULACJI SEMANTYCZNEJ DLA APLIKACJI ROBOTÓW MOBILNYCH

W pracy przedstawiono system symulacji semantycznej „Semantic Simulation Engine" dedykowany aplikacjom robotów mobilnych. Oprogramowanie realizuje symulację robota mobilnego poruszającego się w wirtualnym środowisku powstałym na bazie rzeczywistych pomiarów 3D przekształconych w mapę semantyczną. Pomiary dokonane są z wykorzystaniem rzeczywistego autonomicznego robota mobilnego klasy PIONEER 3AT wyposażonego w laserowy system pomiarowy 3D. Przedstawiono metodę budowy mapy semantycznej oraz metodę transformacji tej mapy do modelu symulacyjnego (NVIDIA PhysX). Przedstawiono autorską modyfikację algorytmu ICP (Iterative Closest Point) zastosowaną do dopasowywania dwóch chmur punktów 3D z wykorzystaniem procesora GPGPU CUDA (Compute Unified Device Architecture). Przedstawiono założenia mapy semantycznej, w tym zbiór podstawowych elementów semantycznych oraz relacji między nimi. Omówiono autorskie metody lokalizowania oraz identyfikacji elementów semantycznych w chmurze punktów 3D z zastosowaniem technik przetwarzania obrazów. Pokazano przykłady działania opracowanego systemu symulacji semantycznej.

1. INTRODUCTION

Semantic information [3] extracted from 3D laser data [4] is recent research topic of modern mobile robotics. In [5] a semantic map for a mobile robot was described as a map that contains, in addition to spatial information about the environment, assignments of mapped features to entities of known classes. In [6] a model of an indoor scene is implemented as a semantic net. This approach is used in [7] where robot extracts semantic information from 3D models built from a laser scanner. In [8] the location of features is extracted by using a probabilistic technique (RANSAC) [9]. Also the region growing approach [10] extended from [11] by efficiently integrating k-nearest neighbor (KNN) search is able to process unorganized point clouds. The improvement of plane extraction from 3D Data by fusing laser data and vision is shown in [12]. The automatic model refinement of 3D scene is introduced in [2], where the idea of feature extraction (planes) is done also with RANSAC. The semantic map building is related to SLAM problem [13]. Most of recent SLAM techniques use camera [14], laser measurement system [15] or even registered 3D laser data [16]. Concerning the registration of 3D scans described in [1] we can find several techniques solving this important issue. The authors of [17] describe ICP algorithm and in[18] the probabilistic matching technique is proposed. In [19] the comparison of ICP and NDT algorithm is shown. In [20] the mapping system that acquires 3D object models of man-made indoor environments such as kitchens is shown. The system segments and geometrically reconstructs cabinets with doors, tables, drawers, and shelves, objects that are important for robots retrieving and manipulating objects in these environments.

A detailed description of computer based simulators for unmanned vehicles is shown in [21]. Also in [22] the comparison of real-time physics simulation systems is given, where a qualitative evaluation of a number of free publicly available physics engines for simulation systems and game development is presented. Several frameworks are mentioned such as USARSim which is very popular in research society [23] [24], Stage, Gazebo [25], Webots [26], Matlab [27] and MRDS (Microsoft Robotics Developer Studio) [28]. Some researchers found that there are many available simulators that offer attractive functionality, therefore they proposed a new simulator classification system specific to mobile robots and autonomous vehicles [29]. A classification system for robot simulators will allow researchers to identify existing simulators which may be useful in conducting a wide variety of robotics research from testing low level or autonomous control to human robot interaction. Another simulation engine – the Search and Rescue Game Environment (SARGE), which is a distributed multiplayer robot operator training game, is described in [30]. On the other hand many simulation environments offer different performance. To ensure the validity of robot models, NIST proposes standardized test methods that can be easily replicated in both computer simulation and physical form [31].

In this paper a new idea of semantic map that can be transformed to rigid body simulation [35] engine is proposed. It can be used for several applications such as mobile robot simulation, mobile robot operator training. The paper is organized as follows: robot hardware short description is shown in 1.1, chapter 1.2 deals with the approach of 3D data registration, Semantic Simulation Engine is introduced in 1.3, chapters 1.4 and 1.5 are related with experiments and conclusion.

1.1. ROBOT

The robot used is an ActiveMedia PIONEER 3AT, equipped with SARA (Sensor Data Acquisition System for Mobile Robotic Applications). SARA is composed by 2 lasers LMS SICK 100 orthogonally mounted. Bottom laser can rotate, therefore it delivers 3D cloud of points in stop-scan fashion. Fig. 1 shows the hardware and data visualization.

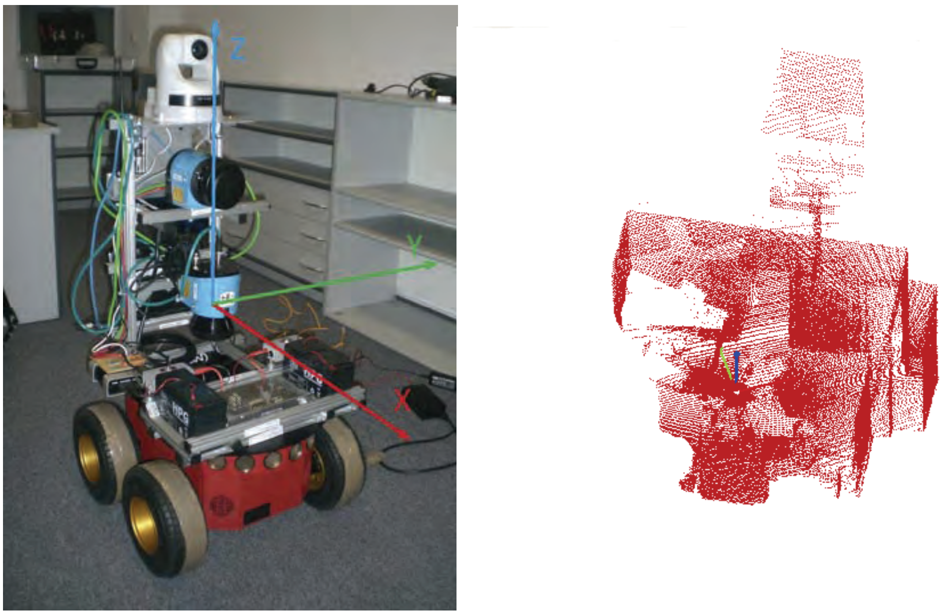

Fig. 1. Robot PIONEER 3AT equipped with SARA (Sensor Data Acquisition System for Mobile Robotic Applications)

1.2. 3D DATA REGISTRATION

Iterative Closest Point algorithm computes rotation and translation between two sets of 3D points [2, 4, 5, 7] – model set M ($|M| = N_m$) and data set D ($|D| = N_d$) by minimization following cost function:

$$E(\mathbf{R},\mathbf{t}) = \sum_{i=1}^{N_m} \sum_{j=1}^{N_d} w_{i,j} \left\| \mathbf{m}_i - \left(\mathbf{R}\mathbf{d}_j + \mathbf{t}\right) \right\|^2 \qquad (1)$$

$w_{i,j}$ is assigned 1 if the i-th point of M correspond to the j-th point in D as in the same space. Otherwise $w_{i,j} = 0$. The iterative ICP algorithm is given:

a) selection of closest points (1NN – 1 nearest neighbor)

b) calculation of transformation (**R,t**) for minimizing equation (1)

c) if criterion of stop is not satisfied back to a)

Calculation of transformation (**R,t**) is performed using reduced equation (1) to the following form:

$$E(\mathbf{R},\mathbf{t}) \propto \frac{1}{N} \sum_{i=1}^{N} \left\| \mathbf{m}_i - \left(\mathbf{R}\mathbf{d}_i + \mathbf{t}\right) \right\|^2 \qquad (2)$$

where $N = \sum_{i=1}^{N_m} \sum_{j=1}^{N_d} w_{i,j}$. The computation of rotation **R** is decoupled from computation of translation **t** using the centroids of points:

$$\mathbf{c}_m = \frac{1}{N}\sum_{i=1}^{N}\mathbf{m}_i, \qquad \mathbf{c}_d = \frac{1}{N}\sum_{i=1}^{N}\mathbf{d}_i, \quad (3)$$

and

$$M' = \{\mathbf{m}_i' = \mathbf{m}_i - \mathbf{c}_m\}_{1,\dots,N} \quad (4)$$

$$D' = \{\mathbf{d}_i' = \mathbf{d}_i - \mathbf{c}_d\}_{1,\dots,N} \quad (5)$$

After replacing (3), (4), (5) in the mean square error function $E(\mathbf{R},\mathbf{t})$ equation (2) takes following form:

$$E(\mathbf{R},\mathbf{t}) \propto \frac{1}{N}\sum_{i=1}^{N}\left\| \mathbf{m}_i' - \mathbf{R}\mathbf{d}_i' - \underbrace{(\mathbf{t} - \mathbf{c}_m + \mathbf{R}\mathbf{c}_d)}_{=\tilde{\mathbf{t}}} \right\|^2 = \frac{1}{N}\sum_{i=1}^{N}\left\|\mathbf{m}_i' - \mathbf{R}\mathbf{d}_i'\right\|^2 - \frac{2}{N}\tilde{\mathbf{t}}\cdot\sum_{i=1}^{N}(\mathbf{m}_i' - \mathbf{R}\mathbf{d}_i') + \frac{1}{N}\sum_{i=1}^{N}\left\|\tilde{\mathbf{t}}\right\|^2 \quad (6)$$

To minimize (6) the algorithm has to minimize following term:

$$E(\mathbf{R},\mathbf{t}) \propto \sum_{i=1}^{N}\left\|\mathbf{m}_i' - \mathbf{R}\mathbf{d}_i'\right\|^2 \quad (7)$$

The optimal rotation is calculated by $\mathbf{R} = \mathbf{V}\mathbf{U}^T$, where matrixes $\mathbf{V}$ and $\mathbf{U}$ are derived by the singular value decomposition of a correlation matrix $\mathbf{C} = \mathbf{U}\mathbf{S}\mathbf{V}^T$ given by:

$$\mathbf{C} = \sum_{i=1}^{N}\mathbf{m}_i'^T\,\mathbf{d}_i' = \begin{bmatrix} c_{xx} & c_{xy} & c_{xz} \\ c_{yx} & c_{yy} & c_{yz} \\ c_{zx} & c_{zy} & c_{zz} \end{bmatrix}, \quad where \quad c_{xx} = \sum_{i=1}^{N} m_{ix}' d_{ix}', c_{xy} = \sum_{i=1}^{N} m_{ix}' d_{iy}', \dots \quad (8)$$

The optimal translation $\mathbf{t}$ is calculated as $\mathbf{t} = \mathbf{c}_m - \mathbf{R}\mathbf{c}_d$ (minimization (6) for $\tilde{\mathbf{t}} = 0$)

1.2.1. SELECTION OF DATA POINTS

Nearest Neighbor Search algorithm aim to optimize the process of finding closest points in two datasets with respect to a distance measure. The NNS problem is stated as follows: given a point set S and a query point q, the goal is to optimize the process of finding the point $p \in S$, which has the smallest distance to q. The space partitioning using Octree improves the search process. Massively parallel computation in CUDA is performed for nearest neighbor search for all points from data set D in parallel. The Octree has 24 levels, therefore 3D xyz space is partitioned into 256 x 256 x 256 buckets. For cubic space 40 m x 40 m x 40 m the bucket dimensions $\approx$0.156 m x 0.156 m x 0.156 m. For each query point separate thread in CUDA architecture performs following algorithm:

a) assign the query point to a thread
b) find bucket for query point using Octree
c) find all neighboring buckets to bucket from b)
d) find closest point in buckets b) + c)

The amount of query point is determined by hardware, in this case 541 x 301 = 162841 3D points.

1.3. SEMANTIC SIMULATION ENGINE

The concept of semantic simulation engine is a new idea, and its strength lies on the semantic map integration with mobile robot simulation. The semantic net is shown in figure 2. The engine basic elements are:

semantic map nodes(entities) L_{sm} = *{Wall, Wall above door, Floor, Ceiling, Door, Free space for door, Stairs…},*
robot simulator nodes(entities) L_{rs} = *{robot, rigid body object, soft body object…},*
semantic map relationships between the entities R_{sm} = *{parallel, orthogonal, above, under, equal height, available inside, connected via joint…},*
robot simulation relationships between the entities R_{rs} = *{connected via joint, position…},*
semantic map events E_{sm} = **robot simulation events** E_{rs} = *{movement, collision between two entities started, collision between two entities stopped, collision between two entities continued, broken joint…}.*

Robot simulator is implemented in NVIDIA PhysX[35]. The entities from semantic map correspond to actors in PhysX. $\mathbf{L_{sm}}$ is transformed into $\mathbf{L_{rs}}$ based on spatial model derived from semantic model i.e. walls, doors and stairs correspond to actors with BOX shapes. $\mathbf{R_{sm}}$ are transformed into $\mathbf{R_{rs}}$ with remark that doors are connected to walls via revolute joint. All entities/relations $\mathbf{R_{sm}}$ has the same initial location in $\mathbf{R_{rs}}$, obviously the location of each actor/entity may change during simulation. The transformation from $\mathbf{E_{sm}}$ to $\mathbf{E_{rs}}$ effects that events related to entities from semantic map correspond to the events related to actors representing proper entities. It is important to emphasize that following events can be noticed during simulation: robot can touch each entity, open/close the door, climb the stairs, enter empty space of the door, damage itself (broken joint between actors in robot arm), brake joint that connects door to the wall etc. It is noteworthy to mention that all robot simulation semantic events are monitored and simulation engine judges them and reports the result to the user.

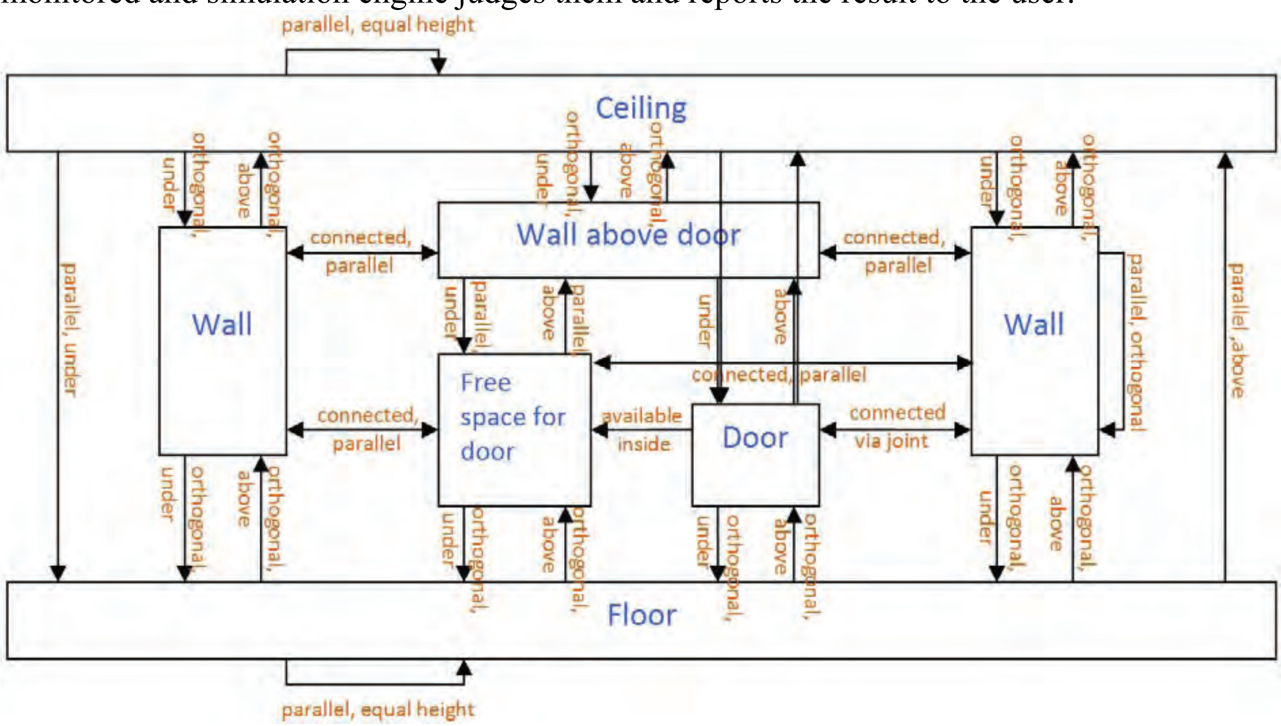

Fig. 2. Semantic net

1.3.1. SEMANTIC ENTITIES LOCALIZATION

The semantic entities localization is implemented using image processing techniques. The idea is that structured entities such as wall, door, stairs correspond to line segments in the image constructed by projection of 3D cloud of points onto XY plane. Fig. 3 shows the procedure.

Input image → Filtering → Dilation → Skeletonization → Hough transform → Output image with lines

Fig. 3. Image processing methods used for prerequisites computation

Input image (where values are real numbers from 0 to 1) is used for prerequisites of semantic entities generation based on image processing methods. The implementation is based on OpenCV image processing library [32].

Filtering box reduces noise from image. The structuring element used for this operation is

$$strel = \begin{bmatrix} 1_{i=-1,j=1} & 1_{i=0,j=1} & 1_{j=1,j=1} \\ 1_{i=-1,j=0} & 0_{i=0,j=0} & 1_{i=1,j=0} \\ 1_{i=-1,j=-1} & 1_{i=0,j=-1} & 1_{i=1,j=-1} \end{bmatrix} \tag{9}$$

For each pixel $p_{k,l}$ from binary image, where k = 1:510, l = 1:510, following equation is solved.

$$p_{res(k,l)} = \sum_{i=-1}^{1} \sum_{j=-1}^{1} strel_{i,j} \cdot p_{k+i,l+j} \cdot \left(|i| + |j|\right) \tag{10}$$

if $p_{res(k,l)} > 0$ and $p_{k,l} = 0$ then, $p_{out(k,l)} = 1$ else $p_{out(k,l)} = 0$.

Dilation box mathematical morphology operation increase the width of binary objects in the image. The OpenCV function cvDilate [32] dilates the source image using the specified structuring element that determines the shape of a pixel neighborhood over which the maximum is taken: dst = dilate(src,element):

$$dst(k,l) = \max_{(i,j) in element} src(k+i, l+j) \tag{11}$$

Skeletonization box – neighboring objects are going to be connected for better hough transform result. Skeletonization based on classical Pavlidis [33, 34] algorithm gives the output as thin lines.

Hough transform box is used for obtaining line segments. Used Hough transform variant is CV_HOUGH_PROBABILISTIC – probabilistic Hough transform (more efficient in case of picture containing a few long linear segments). It returns line segments rather than the whole lines. Every segment is represented by starting and ending points.

1.3.2. SEMANTIC ENTITIES IDENTIFICATION

Fig. 4 demonstrates the result of semantic entities localization procedure where each line correspond to prerequisite of semantic object.

Fig. 4. Image processing for semantic entities localization. Left – input image, right – line segments that are the prerequisites of semantic objects

Each line corresponds to wall prerequisite. The set of lines is used to obtain segmentation of 3D cloud of points, where different walls will have different label. For each line segment form the orthogonal plane$_{orth}$ to plane$_{OXY}$ is computed. It should be noted that the intersection between this two planes is the same line segment. All 3D points which satisfy the condition of distance to plane$_{orth}$ have the same label. Fig. 5 shows the result of segmentation of 3D cloud of points.

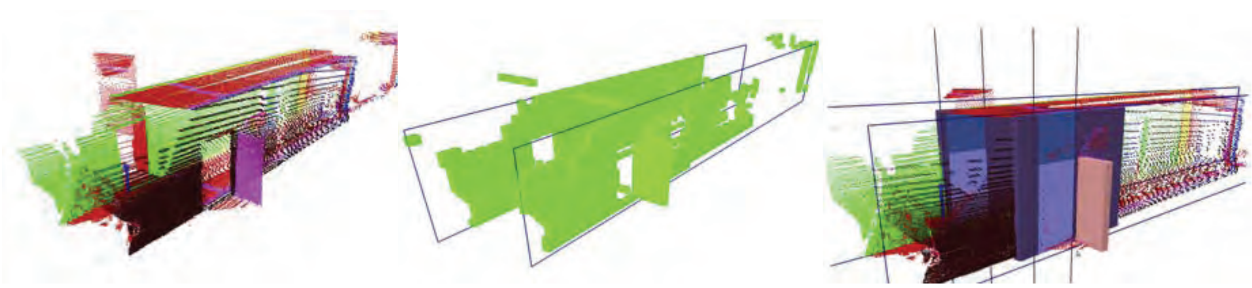

Fig. 5. Left – segmentation of 3D Cloud of points, middle – cubes containing measured points, Right – semantic model

In the first step all prerequisites of walls were checked separately. To perform the scene interpretation semantic net is proposed (Fig. 2). The interpretation of the scene comprises generic architectural knowledge like in [6], [2] and [8]. Nodes of a semantic net represent entities of the world, the relationships between them are defined. Possible labels of the nodes are *L = {Wall, Wall above door, Floor, Ceiling, Door, Free space for door}*. The relationships between the entities are *R = {parallel, orthogonal, above, under, equal height, available inside, connected via joint}*. The semantic net can easily be extended to more entities and relationships which determine a more sophisticated feature detection algorithms. In this case the feature detection algorithm is composed by the method of cubes generation (Fig. 5 middle), where each cube should contain measured 3D point. In the second step of the algorithm wall candidates are chosen. From this set of candidates, based on relationships between them, proper labels are assigned and output model is generated (Fig. 5 right).

The image processing methods are also used for stairs prerequisites generation. The result of this procedure is shown in Fig. 6, where red rectangle corresponds to stairs prerequisite.

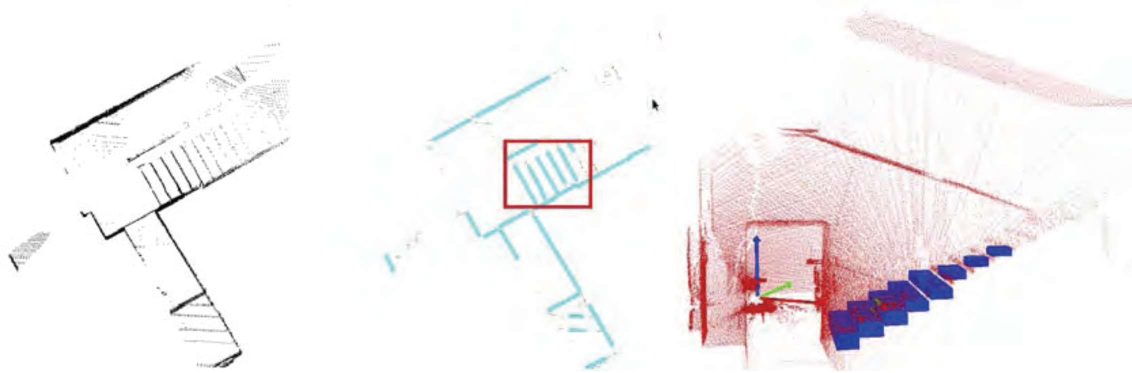

Fig. 6. Left – Input image, middle – stairs prerequisite, right – semantic model

It is important to emphasize that the set of parallel lines in the same short distance between each other can be a projection of stairs. Possible labels of the nodes are *L = {stair}*. The relationships between the entities are *R = {parallel, above, under}*. Fig. 6 – right shows resulting model of stairs generated from 3D cloud of points. In this spatial model each stair (except first and last one obviously) is in relation *r=ablove&parallel* with the previous one and in relation *r=under&parallel* with next one.

1.4. EXPERIMENTS

Fig. 7 show 3D data registration result performed using NVIDIA GF9800 GPU. The processing time for one ICP iteration equals 300ms average. In this particular case ICP need 10 iterations to solve minimization of error function (1), therefore 541 x 301 data points are aligned in 3 s.

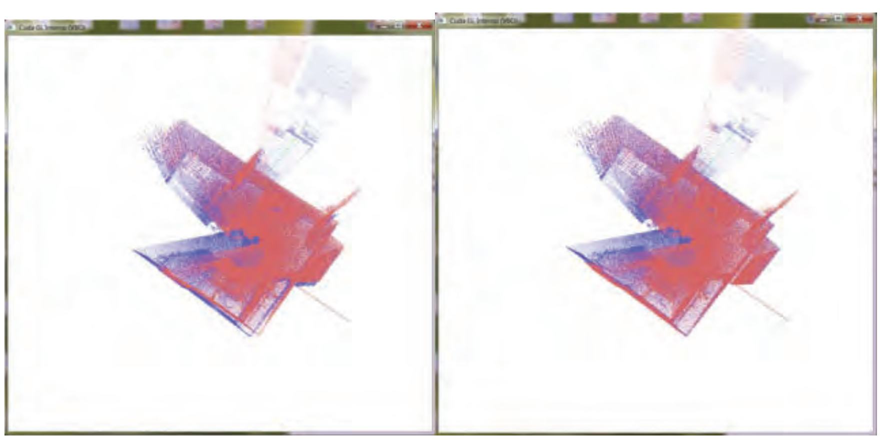

Fig. 7. Data registration

Fig. 8 shows transformed semantic model into NVIDIA PhysX simulation and in the same time integrated with inspection robot model.

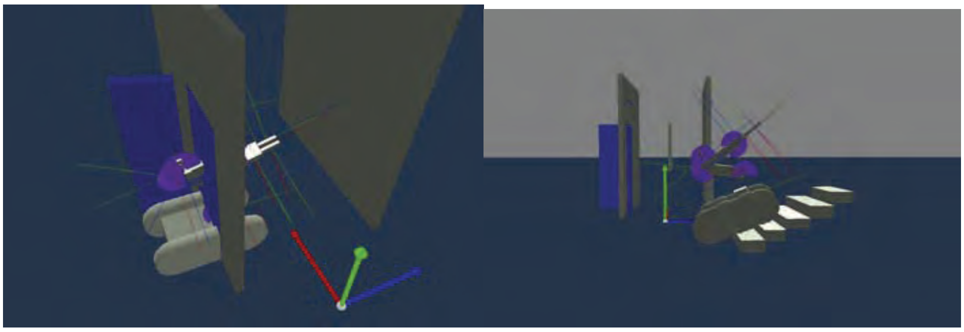

Fig. 8. Transformed semantic model into NVIDIA PhysX simulation

It is important to emphasized that virtual model is located in the scene build from real 3D data. The interaction between all entities in PhysX model are monitored and reported, for example following events: robot entering empty space of door, robot touching wall, broken joint (door, robot arm) etc., robot climbing stairs.

1.5. CONCLUSION

In the paper the semantic simulation engine that is used for merging real semantic data with NVIDIA PhysX mobile robot simulation is shown. The approach can be used for further development of sophisticated training tools i.e. AR (Augmented Reality), where real robots will be used for environment modeling. New approach of image processing techniques in the process of semantic entities localization and identification is shown. It can be developed for another objects recognition in the INDOOR environment. It is shown the new application for semantic mapping – the mobile robot semantic simulation, where identified and modeled semantic objects interact with predefined simulation entities. In our opinion the approach deliver powerful tool for INDOOR environment inspection and intervention in which operator can use semantic information to interact with entities. Future work will be related to the integration of data registration techniques and augmented reality approach.

1.6. REFERENCES

1. M. Magnusson, T. Duckett, A. J. Lilienthal, 3D scan registration for autonomous mining vehicles, Journal of Field Robotics 24 (10) (2007), pp. 803–827.
2. A. Nüchter, H. Surmann, J. Hertzberg, Automatic model refinement for 3D reconstruction with mobile robots, in: Fourth International Conference on 3-D Digital Imaging and Modeling 3DIM 03, 2003, p. 394.
3. M. Asada, Y. Shirai, Building a world model for a mobile robot using dynamic semantic constraints, in: Proc. 11 th International Joint Conference on Artificial Intelligence, 1989, pp. 1629–1634.
4. A. Nüchter, O. Wulf, K. Lingemann, J. Hertzberg, B. Wagner, H. Surmann, 3D mapping with semantic knowledge, in: IN ROBOCUP INTERNATIONAL SYMPOSIUM, 2005, pp. 335–346.
5. A. Nüchter, J. Hertzberg, Towards semantic maps for mobile robots, Robot. Auton. Syst. 56 (11) (2008), pp. 915–926.

6. O. Grau, A scene analysis system for the generation of 3-d models, in: NRC '97: Proceedings of the International Conference on Recent Advances in 3-D Digital Imaging and Modeling, IEEE Computer Society, Washington, DC, USA, 1997, p. 221.
7. A. Nüchter, H. Surmann, K. Lingemann, J. Hertzberg, Semantic scene analysis of scanned 3D indoor environments, in: Proceedings of the Eighth International Fall Workshop on Vision, Modeling, and Visualization (VMV 03), 2003.
8. H. Cantzler, R. B. Fisher, M. Devy, Quality enhancement of reconstructed 3D models using coplanarity and constraints, in: Proceedings of the 24th DAGM Symposium on Pattern Recognition, Springer-Verlag, London, UK, 2002, pp. 34–41.
9. M. A. Fischler, R. Bolles, Random sample consensus. a paradigm for model fitting with apphcahons to image analysm and automated cartography, in: Proc. 1980 Image Understanding Workshop (College Park, Md., Apr i980) L. S. Baurnann, Ed, Scmnce Apphcatlons, McLean, Va., 1980, pp. 71–88.
10. M. Eich, M. Dabrowska, F. Kirchner, Semantic labeling: Classification of 3D entities based on spatial feature descriptors, in: IEEE International Conference on Robotics and Automation (ICRA2010) in Anchorage, Alaska, May 3, 2010.
11. N. Vaskevicius, A. Birk, K. Pathak, J. Poppinga, Fast detection of polygons in 3D point clouds from noise-prone range sensors, in: IEEE International Workshop on Safety, Security and Rescue Robotics, SSRR, IEEE, Rome, 2007, pp. 1–6.
12. H. Andreasson, R. Triebel, W. Burgard, Improving plane extraction from 3D data by fusing laser data and vision, in: Proceedings of the 20 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2005, pp. 2656–2661.
13. J. Oberlander, K. Uhl, J. M. Zollner, R. Dillmann, A region-based slam algorithm capturing metric, topological, and semantic properties, in: ICRA'08, 2008, pp. 1886–1891.
14. B. Williams, I. Reid, On combining visual slam and visual odometry, in: Proc. International Conference on Robotics and Automation, 2010.
15. L. Pedraza, G. Dissanayake, J. V. Miro, D. Rodriguez-Losada, F. Matia, Bs-slam: Shaping the world, in: Proceedings of Robotics: Science and Systems, Atlanta, GA, USA, 2007.
16. M. Magnusson, H. Andreasson, A. Nüchter, A. J. Lilienthal, Automatic appearance-based loop detection from 3D laser data using the normal distributions transform, Journal of Field Robotics 26 (11–12) (2009), p. 892–914.
17. P. J. Besl, H. D. Mckay, A method for registration of 3-d shapes, Pattern Analysis and Machine Intelligence, IEEE Transactions on 14 (2) (1992), 239–256.
18. D. Hahnel, W. Burgard, Probabilistic matching for 3D scan registration, in: Proc. of the VDI - Conference Robotik 2002 (Robotik), 2002.
19. M. Magnusson, A. Nüchter, C. Lörken, A. J. Lilienthal, J. Hertzberg, Evaluation of 3D registration reliability and speed – a comparison of icp and ndt, in: Proc. IEEE Int. Conf. on Robotics and Automation, 2009, pp. 3907–3912.
20. R. B. Rusu, Z. C. Marton, N. Blodow, M. Dolha, M. Beetz, Towards 3D point cloud based object maps for household environments, Robot. Auton. Syst. 56 (11) (2008), pp. 927–941.
21. J. Craighead, R. Murphy, J. Burke, B. Goldiez, A survey of commercial and open source unmanned vehicle simulators, in: Proceedings of ICRA, 2007.
22. A. Boeing, T. Braunl, Evaluation of real-time physics simulation systems, in: GRAPHITE '07: Proceedings of the 5th international conference on Computer graphics and interactive

techniques in Australia and Southeast Asia, ACM, New York, NY, USA, 2007, pp. 281–288.
23. J. Wang, M. Lewis, J. Gennari, Usar: A game-based simulation for teleoperation, in: Proceedings of the 47th Annual Meeting of the Human Factors and Ergonomics Society, Denver, CO, Oct. 13–17, 2003.
24. J. Wang, M. Lewis, J. Gennari, A game engine based simulation of the nist urban search and rescue arenas, in: Proceedings of the 35th conference on Winter simulation: driving innovation, December 07–10, New Orleans, Louisiana, 2003.
25. R. B. Rusu, A. Maldonado, M. Beetz, I. A. Systems, T. U. Mnchen, Extending player/stage/gazebo towards cognitive robots acting in ubiquitous sensor-equipped environments, in: Accepted for the IEEE International Conference on Robotics and Automation (ICRA) Workshop for Network Robot System, 2007, April 14, 2007.
26. L. Hohl, R. Tellez, O. Michel, A. J. Ijspeert, Aibo and Webots: Simulation, Wireless Remote Control and Controller Transfer, Robotics and Autonomous Systems 54 (6) (2006), pp. 472–485.
27. T. Petrinic, E. Ivanjko, I. Petrovic, Amorsim a mobile robot simulator for Matlab, in: Proceedings of 15th International Workshop on Robotics in Alpe-Adria-Danube Region, June 15–17, Balatonfred, Hungary, 2006.
28. C. Buckhaults, Increasing computer science participation in the first robotics competition with robot simulation, in: ACMSE 47: Proceedings of the 47th Annual Southeast Regional Conference, ACM, New York, NY, USA, 2009, pp. 1–4.
29. J. Craighead, R. Murphy, J. Burke, B. Goldiez, A robot simulator classification system for hri, in: Proceedings of the 2007 International Symposium on Collaborative Technologies and Systems (CTS 2007), 2007, pp. 93–98.
30. J. Craighead, J. Burke, R. Murphy, Using the unity game engine to develop sarge: A case study, in: Proceedings of the 2008 Simulation Workshop at the International Conference on Intelligent Robots and Systems (IROS 2008), 2008.
31. J. Craighead, Distributed, game-based, intelligent tutoring systems the next step in computer based training?, in: Proceedings of the International Symposium on Collaborative Technologies and Systems (CTS 2008), 2008.
32. http://opencv.willowgarage.com/wiki/
33. S.-W. Lee, Y. J. Kim, Direct extraction of topographic features for gray scale character recognition, IEEE Trans. Pattern Anal. Mach. Intell. 17 (7) (1995) 724-729.
34. L. Wang, T. Pavlidis, Direct gray-scale extraction of features for character recognition, IEEE Trans. Pattern Anal. Mach. Intell. 15 (10) (1993), pp. 1053–1067.
35. J. Bedkowski, M. Kacprzak, A. Kaczmarczyk, P. Kowalski, P. Musialik, A. Maslowski, T. Pichlak, Rise mobile robot operator training design, in: 15th International Conference on Methods and Models in Automation and Robotics (CD-ROM), Miedzyzdroje, Poland, 2010.