3D Reconstruction of Rough Terrain for USARSim Using a Height-map Method

G. Roberts
Intelligent Systems Division, National Institute of Standards & Technology, 100 Bureau Drive, MS 8230, Gaithersburg, MD 20899 USA.
1-301-975-3434,
gael.roberts@nist.gov

S. Balakirsky
Intelligent Systems Division, National Institute of Standards & Technology, 100 Bureau Drive, MS 8230, Gaithersburg, MD 20899 USA.
1-301-975-4791,
stephen.balakirsky@nist.gov

S. Foufou
Laboratoire Electronique, Informatique et Image (LE2i Lab). UMR CNRS 5158, University of Burgundy, 21000, Dijon, FRANCE.
00-333-8039-3805,
sfoufou@u-bourgogne.fr

ABSTRACT

In this paper, a process for a simplified reconstruction of rough terrains from point clouds acquired using laser scanners is presented. The main idea of this work is to build height-maps which are level gray-scale images representing the ground elevation. These height-maps are generated from step-fields which can be represented by a set of side-by-side pillars. Although height-maps are a practical means for rough terrain reconstruction, it is not possible to represent two different elevations for a given location with one height-map. This is an important drawback as terrain point clouds can show different zones representing surfaces above other surfaces.

In this paper, a methodology to create several height-maps for the same terrain is described. Experimental results are shown using the high-fidelity physics-based framework for the Unified System for Automation and Robot Simulation (USARSim).

Keywords

3D Reconstruction, height-map, step-field, point cloud, USARSim.

1. INTRODUCTION

The usefulness of simulation systems for developing agent control systems is well established. The role of the simulation is to provide convincing sensor measurements as input to the control system and to accurately model the system's response to actuator outputs (physics, dynamics, and statics). An important enabler that allows the simulation system to meet these requirements is the existence of an accurate model of the real world.

PerMIS'08, August 19-21, 2008, Gaithersburg, MD, USA
ACM ISBN 978-1-60558-293-1/08/08.

One aspect of this world model is its representation of the ground surface, or a height-map. The required complexity of the ground surface representation is directly related to the agent's application and target environment. For example, an agent that is required to operate in a flat-floored environment with no overhanging obstacles may only require occupancy information in its model. More complex environments that include uneven terrain may require a 2.5D model (a single elevation value per unit area). Even more complex environments that include overhanging obstacles or multi-level terrain may require a full 3D model (multiple elevation values per unit area).

This paper addresses the automatic creation of height-maps from data collected with laser scanners. The overall objective is to be able to recreate complex 3D terrains such as those that may be found at disaster sites (partially destroyed buildings, large amounts of debris, etc.) for use in simulation and in reconstructed physical representations. While the general techniques developed in this paper should be applicable to the general problem, the specific output formats have been tailored for the Unified System for Automation and Robotics Simulation (USARSim) framework [1].

USARSim was initially developed in 2002 as a low cost (under $40) robotics simulator at Carnegie Mellon University (CMU) and the University of Pittsburgh [2]. In 2005, USARSim management was handed off to the National Institute of Standards & Technology (NIST) and it became an open source project hosted on the Sourceforge network (www.sourceforge.net/projects/usarsim). USARSim is based on the UnReal Game Engine from Epic Games[1] and provides a high-fidelity, physics based simulation environment for robotic development. One of the unique features of USARSim lies in its validated sensor and robot models [3]. In addition to its use as a research tool, USARSim is also used as the basis for the RoboCup Rescue Virtual Competition as well as the

[1] Certain commercial software and tools are identified in this paper in order to explain our research. Such identification does not imply recommendation or endorsement by the authors, nor does it imply that the software tools identified are necessarily the best available for the purpose.

IEEE Virtual Automation and Manufacturing Competition. More information on USARSim may be found in [4].
The remainder of this paper is laid out as follows: Section 2 recalls some preliminary definitions regarding the dataset, step-fields, and height-maps. The relation between our work and other works is explained in Section 3. Section 4 gives the general approach adapted in this work. Section 5 details the proposed algorithm for terrain reconstruction using height-maps. Some experimental results are shown and discussed in Section 6. Section 7 concludes and discusses about future extensions of this work.

2. PREREQUISITE

This section indicates the type of data used in this project and gives some fundamental definitions.
The purpose of this work is to reconstruct models of rough terrains for use in a simulator. Laser based scanners are used to acquire 3D numerical data (point clouds) representing the ground. Point clouds may be complex, containing ground, surfaces, and heaps of objects. Point clouds are muddled, contain no texture, and their points are not uniformly distributed in space.
In this work, the complexity of point clouds representing real-world scenes is intentionally avoided by only considering point clouds generated artificially. These point clouds are simpler but keep approximately the same characteristics (muddled, not textured, not uniform) as those acquired from real-world scenes. These artificial point clouds are generated as follows: a scene is created with different simple objects (planes, spheres, cylinders …) and points are randomly put onto the surface of each object using their equation. At the end, noise is added to make the clouds more realistic.
Two important models are calculated for each point cloud: a step-field model and a height-map model. A step-field can be represented by a set of side-by-side pillars (see Figure 1) and is computed as follows: starting from a horizontal plane, side by side pillars are placed with predefined dimension (height * width * depth). These pillars will all have identical width and depth, but their height will vary according to the configuration and properties of the point cloud. More details regarding the construction of these pillars are given later.

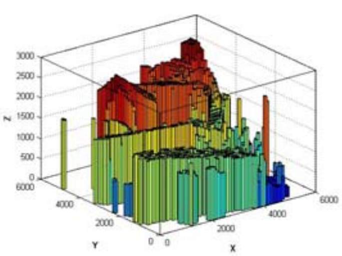

Fig. 1: Example of step-field

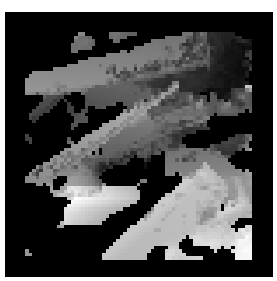

Fig. 2: Example of height-map

The height-map model, also called CEM (Cartesian Elevation Map) or elevation map, is a gray scale picture (based on 256 levels of gray) which represents the height information of the studied terrain (see Figure 2). The lighter the color of a pixel is, the higher the elevation of the terrain is for the location represented by this pixel.

3. RELATED WORK

Since the beginning of mobile robot development, researchers have been striving to find ways for the platforms to navigate through various terrains. Work has even been performed on classifying the difficulty of terrain for robotic traversal. The general idea of our work is to find an adequate representation of rough terrains that will allow simulations of robots to be used to create and evaluate algorithms that will allow mobile robots to be more or less autonomous on rough terrains.
Because robots come in many different shapes, sizes and abilities, simulation is a useful tool that allows one to predict which types of robots will be best-suited for particular types of terrain under consideration. Simulation is also an efficient way to test the impact of algorithm changes on the effectiveness of a given platform. There are two main types of ground mobility: walking robots [5, 6] and rolling robots (with tracks or wheels). Despite a large effort in the area of walking robots, they are not completely adequate for the ground studied for this paper. The reason for this is the rocky and uneven nature of a disaster site. Consequently, rolling robots which are also the subject of studies as regards their efficiency for rough terrains are used [7, 8, 9].
The terrains are created from a cloud of points. These reconstructions are an important area of computer graphics and there are many methods of 3D reconstruction from point clouds. Among these methods, which have the objective of creating a mesh, there are the Delaunay triangulation (which is joined to the Voronoi diagram) [10, 11, 12], related methods such as Ball-Pivoting (pivot of a sphere) [13] or other methods such as the one from Chang et al. [14]. All these methods give very good constructions, but the results are often very costly in memory.
To resolve this problem, other research aims at simplifying the obtained meshing. These simplifications consist in reducing the number of polygons in a meshing by grouping polygons according to certain criteria such as the orientation of polygons or their configurations [15, 16, 17]. Although these mesh simplifications give reasonable results, the processes to do this are very costly in memory compared to the creation of height-maps as discussed in the next paragraph.
Other methods for creating ground representations use height-maps. The concept of height-map is used for several studies including the crossing possibilities of grounds or the construction of obstacles on a ground [18, 19]. Height-maps are useful for path planning and collision avoidance [20], they are also used to represent surfaces having a low-level of detail such as landscapes (e.g., video games).

4. THE APPROACH TO THE PROBLEM

The purpose of this work is to create a 3D terrain model that will be used in a simulator, consequently the simplest possible construction is necessary in order to reduce the computation costs during simulation. Therefore, in order to have a good simulation, it is very important to have a very simple scene because the Frame Per Second (FPS) is taken into consideration. For a complex scene with big meshed objects, the FPS will be too low for effective rendering and the simulation will certainly be too slow.
Our simulator utilizes the Unreal 3D engine, thus the editor of this engine is used to construct the 3D models either by importing them as meshed scenes or by generating a height-map from point clouds. However, it is only considering the use of height-maps to avoid having bad FPS during the simulation.

5. ALGORITHM DESCRIPTION

The algorithm operates in two steps: (i) compute the step-field from the point cloud, (ii) and then compute the height-map. Several height-maps may be created for the same scene in order to have various layers representing surfaces (or objects) because these surfaces can be positioned one on the top of the other.

5.1. Step-fields elaboration

Having acquired the point cloud, a virtual horizontal plane P(x,y) is created:

$$P(x,y) = Z_{min} - \alpha. \tag{1}$$

Where Z_{min} represents the coordinates of the lowest point of the cloud on the Z axis, and α represents a small distance defined by the user. Therefore, the plane will be "α units" below the point cloud. This plane is bounded by the cloud on the X and Y axes.
Then a grid on this plane is created. Each compartment of this grid represents the possible base of a future pillar. The side sizes of each compartment will be equal to the dimension of the pillar sides. The size of a compartment is given by the user. Thus, the dimensions of the grid (length * width compartments) are a function of the plan size and the compartments size.
A pillar is created in each compartment situated under a non-empty subset of points. The height of this pillar is the average of the heights (Z-components) of the points of the subset. The obtained set of pillars defines the step-field of the scene (terrain), and will be used to construct the height-map.

5.2. Height-map construction

The height-map is built from the step-field. To this end, each pillar created is correlated with a pixel of a gray-scale image. If a pillar has a width of 10 cm, a pixel of the image will also represent a zone of 10 cm in reality. One should notice that if there was beforehand a grid of 64*64 compartments on the initial plane, there will be a height-map of 64*64 pixels.
Height-map being an image of 256 levels of gray, the maximal height (H_{max}) of the pillars is put in correspondence to the highest level of the scales of gray (that is 256). Other heights (h) of pillars will be represented by a proportional gray level g, such that:

$$g = h * 256 / H_{max}. \tag{2}$$

We note that the simulator constrains the height-map to have a number of pixels equal to a power of 2 for each of its sides. For that purpose, padding is added around the height-map obtained previously. This padding will have a value equal to 0 on the scale of the gray-levels. (see Figure 2).

5.3. Construction of different layers

A layer can be assimilated to one height-map or to one surface in the 3D reconstruction.

5.3.1. Why different layers?

The purpose of creating different layers is to rebuild surfaces which are above other surfaces. Effectively, in a point cloud, several objects heaped upon each other can be represented. A perspective scan of a table is a perfect simple example of this situation, because in the associated point cloud, there is the surface of the ground and the top surface of the table.

A height-map is a gray scale picture and with only one gray color. It is not possible to represent different elevations. Thus, different surfaces (or elevations) cannot be represented for a given location with only one height-map.

5.3.2. The process

With one step-field, one height-map is constructed, and with one height-map, one surface is constructed for the simulator. So, in order to have different surfaces in the final 3D model, it is necessary to construct different step-fields for the same scene.
As presented in Section 5.1, a plane is placed below the point cloud and it is subdivided into a grid. Each compartment of the grid is potentially the base of several pillars. Since the height of a pillar is equal to the average of the Z-coordinates of the points above it, to construct different pillars at the same location the points above the compartment are portioned according to their elevations into different subsets. Then, the height of each pillar will be equal to the average of the Z-coordinate of the points of one particular subset. Grouping the obtained pillars into different sets gives different step-fields. The pillars of the same compartment cannot be placed in the same set (every step-field is composed of pillars from different compartments).
Consequently, if there are X sets of points, there will be X pillars for the same location. The number of pillars can be different from one compartment to another. The number of layers equals the maximum number of pillars created on the grid for one compartment.

5.3.3. Classification of pillars

When pillars are created for one compartment of the grid, the first created pillar is placed in the first layer, the second pillar in the second layer and so on. This placement of this pillar leads to layers containing pillars not according to their heights, but according to their creation rank. A pillar which should be in step-field number N can be in step-field number N-1. As a consequence, the built surfaces will not be regular and a pillar replacement process is needed to classify each pillar in the adequate layer.
To classify the pillars, the average elevation of each layer is computed. Then, pillars having heights close to the average of a particular layer are moved to this layer. This process is repeated several times until stability is reached.

6. EXPERIMENTAL RESULTS

Some experimental results are presented and discussed in this section,. Figures 3, 4, 5, and 6 show different reconstruction cases. Figures 3.a, 4.a, 5.a, and 6.a show the point clouds and Figures 3.b, 4.b, 5.b, and 6.b show the associated computed reconstruction. Textures that appear on these reconstructions represent the layers found on the reconstructed surfaces.
In the first example (see Figure 3.a and Figure 3.b), there are 3 horizontal overlaid planes. A point in a plane does not have the same elevation as a point in another plane. By this fact, at the creation of the step-field, it is easy to classify the pillars according to their elevations for each layer, and one may notice remark that the associated height-maps are very homogeneous (there is one height-map for each surface). Of course this is the perfect case where the reconstructed layers (Figure 3.b) correspond to the surfaces of the initial planes.

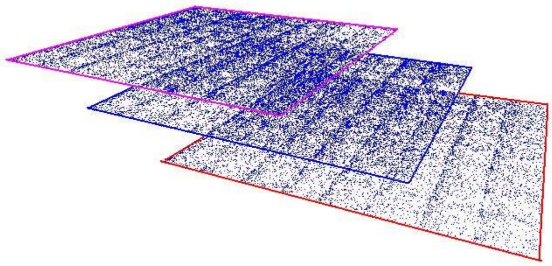

Fig. 3.a: Point cloud representing 3 planes

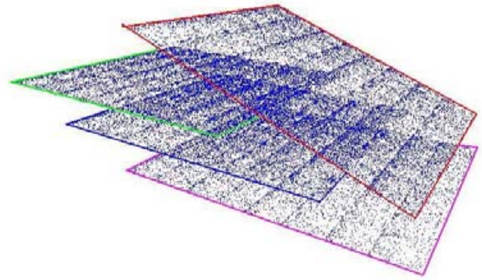

Fig. 4.a: Point cloud representing 4 planes

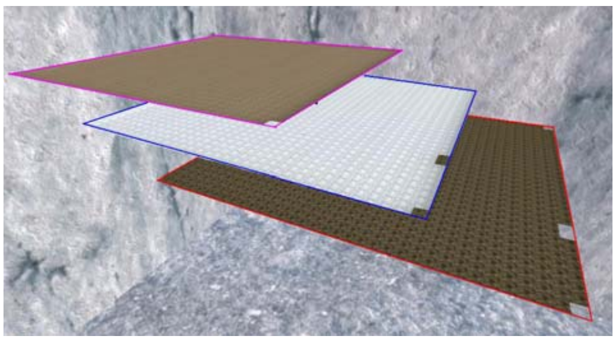

Fig. 3.b: Construction representing 3 planes

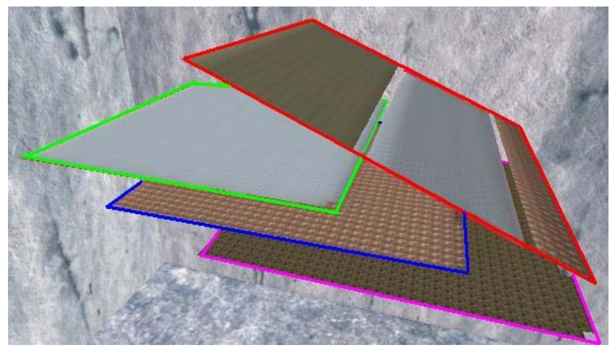

Fig. 4.b: Construction representing 4 planes

The second example (Figure 4.a) represents a scene that is a little more complicated. There are the three planes from Figure 3.a above which a fourth inclined plane is positioned. This fourth plane crosses the elevation of the two highest horizontal planes. Figure 4.b displays the associated construction, where one can see that the three parallel planes are constructed correctly, but the inclined plane presents some discrepancies. This inclined plane is constructed with three different layers because the pillars are classified according to their elevations. Since the inclined plane crosses the elevations of the two other planes, the pillars for the inclined plane are classified in different layers. This classification may lead the pillars of the inclined plane to be positioned in the same layer as the pillars of the other planes with approximately the same elevation.

Notice that the reconstruction of the inclined surface is not continuous. That is explained by the fact that the inclined plane is reconstructed with three layers, and although no data are lost, the continuity of the surface at this location is lost, which creates a discontinuity in the 3D model. The 3D model is used to manipulate a robot in a simulator, and the created holes have the size of the pillars (a small size). Thus the robots can cross over these holes and there is no real impact for the simulation.

The third example (see Figure 5) is more complicated than the second example where the inclined plane does not only cross the elevation of others surfaces, but also crosses one of these planes. The same problem as previously is noticed (the holes because of the discontinuity) and the construction is more complicated too. Actually, even the horizontal planes are built with several layers but the 3D reconstruction is still correct.

The fourth example (see Figure 6) presents a more complex scene where a cylindrical object is positioned on an horizontal plane. The relevance of this scene is to show how one object (the cylinder) which presents a surface above itself is reconstructed. The algorithm cuts the object surface into two parts (see Figure 6.b) in order to create two different layers for the same object.

During the process of creating the ground, there are edge effects. It is during the process of creation of a 3D scene from height-maps that there is a loss of information. This is due to the fact that a pixel of the height-map represents a vertex and not a compartment of the surface in the simulator. This fact has the effect of cutting down the side of the surfaces. Consequently, if there is a ground (or a height-map) which is strewed with holes, these holes will be increased in the final reconstruction, and if the surface is too thin, it will totally disappear.

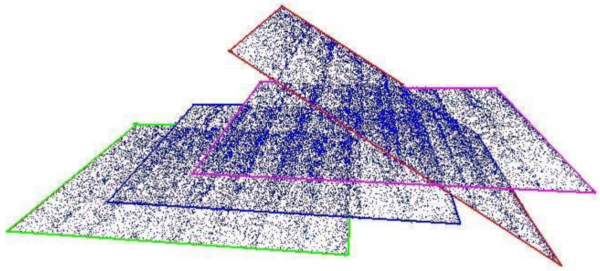

Fig. 5.a: Point cloud representing 4 planes with intersection

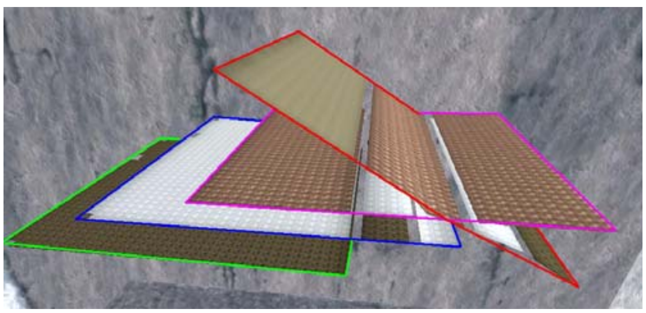

Fig. 5.b: Construction representing 4 planes with intersection

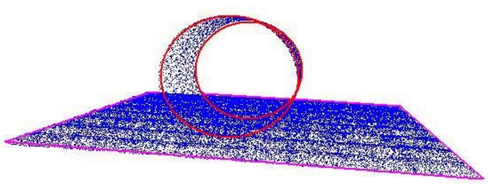

Fig. 6.a: Point cloud representing a cylinder onto a plane

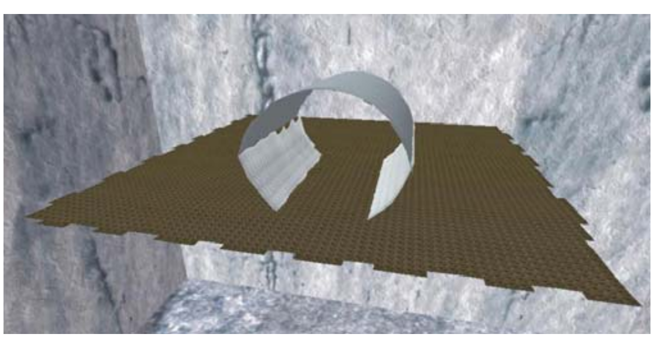

Fig. 6.b: Construction representing a cylinder onto a plane

7. CONCLUSION

We have presented a method to build a simplified 3D scene from point clouds using height-maps. The use of several height-maps allows us to reconstruct in a very simplistic way different scenes, which could have had a heavy and high cost mesh based reconstruction. Furthermore, the elaboration of overlapped height-maps representing grounds to be analyzed permits one to simplify the reconstruction of overlapped surfaces. This new reconstruction method gives satisfactory results for relatively simple scenes. However, height-maps based reconstruction methods are still limited for ground representations, which become complex by the presence of several kinds of objects. The height-maps method works very well for the generation of grounds having few discontinuities, but fails with complex grounds with many discontinuities or with important variations. A future work is to validate this method in two different ways. The first is to create a terrain as similar as possible with the reality. The second way is to compare the similarity between the behaviors of a robot on the terrain in the simulation and in the real world. Namely, the second way is more important than the first one.

The objective is to reconstruct real rough terrains for a simulation, thus we can have very complex scenes. In these scenes, we can differentiate what we call the simple parts of the scene and the complex parts of the scene. To differentiate these two sorts of parts, the local density of points and the shape of the representation of the points are taken in consideration. In this way, a smooth shape will be assimilated to a simple part, and a small area with much variation will be assimilated to a complex part. In order to improve our algorithm, we envisage combining the overlapped height-maps method (for the simple parts of the ground) with a simple mesh-based method (for the complex parts of the ground). Actually, we can also define the boundary between a simple and a complex part with the effectiveness of local reconstructions using the overlapping height-maps method.

8. REFERENCES

[1] Carpin, S.; Lewis, M.; Wang, J.; Balarkirsky, S.; Scrapper, C.; USARSim: a robot simulator for research and education. Proceedings of the IEEE 2007 International Conference on Robotics and Automation, Page(s):1400-1405

[2] Balaguer, B.; Balakirsky, S.; Carpin, S.; Lewis, M.; Scrapper, C.; "USARSim: A Validated Simulator for Research in Robotics and Automation", Workshop on "Robot Simulators: Available Software, Scientific Applications, and Future Trends" at IEEE/RSJ 2008.

[3] Carpin, S.; Stoyanov, T.; Nevatia Y.; Lewis, M.; Wang, J.; Quantitative Assessments of USARSim Accuracy. Proceedings of PerMIS 2006, NIST Special Publication 1062, August 2006.

[4] Usarsim homepage: http://sourceforge.net/projects/usarsim (Accessed: 2008)

[5] Pongas, D.; Mistry, M.; Schaal, S.; "A Robust Quadruped Walking Gait for Traversing Rough Terrain", Robotics and Automation 10-14 April 2007 Page(s):1474 – 1479.

[6] Rebula, J.R.; Neuhaus, P.D.; Bonnlander, B.V.; Johnson, M.J.; Pratt, J.E.; "A Controller for the LittleDog Quadruped Walking on Rough Terrain", Robotics and Automation, 2007 IEEE International Conference on 10-14 April 2007 Page(s):1467 – 1473.

[7] Fukuda, T.; Zhang, X.; Hasegawa, Y.; Matsuno, T.; Hoshino, H.; "Preview posture control and impact load control of rough terrain vehicle with interconnected suspension", Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004

IEEE/RSJ International Conference on Volume 1, 28 Sept.-2 Oct. 2004 Page(s):761 - 766 vol.1,
[8] Udengaard, M.; Iagnemma, K.; “Kinematic analysis and control of an omnidirectional mobile robot in rough terrain”, Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on Oct. 29 2007-Nov. 2 2007 Page(s):795 – 800.
[9] Nakamura, S.; Faragalli, M.; Mizukami, N.; Nakatani, I.; Kunii, Y.; Kubota, T.; “Wheeled robot with movable center of mass for traversing over rough terrain”, Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on Oct. 29 2007-Nov. 2 2007 Page(s):1228 – 1233.
[10] Ledoux, H.; “Computing the 3D Voronoi Diagram Robustly: An Easy Explanation”, Voronoi Diagrams in Science and Engineering, 2007. ISVD '07. 4th International Symposium on 9-11 July 2007 Page(s):117 - 129.
[11] Tse, R.; Gold, C.; Kidner, D.; “Using the Delaunay Triangulation/ Voronoi Diagram to extract Building Information from Raw LIDAR Data”, Voronoi Diagrams in Science and Engineering, 2007. ISVD '07. 4th International Symposium on 9-11 July 2007 Page(s):222 – 229.
[12] Labatut, P.; Pons, J.-P.; Keriven, R.; “Efficient Multi-View Reconstruction of Large-Scale Scenes using Interest Points, Delaunay Triangulation and Graph Cuts”, Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on 14-21 Oct. 2007 Page(s):1 – 8.
[13] Bernardini, F.; Mittleman, J.; Rushmeier, H.; Silva, C.; Taubin, G.; “The ball-pivoting algorithm for surface reconstruction”, Visualization and Computer Graphics, IEEE Transactions on Volume 5, Issue 4, Oct.-Dec. 1999 Page(s):349 – 359.
[14] Ming-Ching, C.; Leymarie, F.F.; Kimia, B.B.; “Surface Reconstruction from Point Clouds by Transforming the Medial Scaffold”, 3-D Digital Imaging and Modeling, 2007. 3DIM '07. Sixth International Conference on 21-23 Aug. 2007 Page(s):13 – 20.
[15] Hua-Hong, C.; Xiao-nan, L.; Ruo-tian, L.; “Surface Simplification Using multi-edge mesh collapse”, Image and Graphics, 2007. ICIG 2007. Fourth International Conference on 22-24 Aug. 2007 Page(s):954 - 959.
[16] Qu, L.; Meyer, G. W.; “Perceptually Guided Polygon Reduction”, Visualization and Computer Graphics, IEEE Transactions on Volume 14, Issue 5, Sept.-Oct. 2008 Page(s):1015 – 1029.
[17] In Kyu, P.; Sang Wook, L.; Sang Uk, L.; “Shape-adaptive 3D mesh simplification based on local optimality measurement”, Computer Graphics and Applications, 2002. Proceedings. 10th Pacific Conference on 9-11 Oct. 2002 Page(s):462 – 466.
[18] Asada, M.; “Building a 3D world model for mobile robot from sensory data”, Robotics and Automation, 1988. Proceedings., 1988 IEEE International Conference on 24-29 April 1988 Page(s):918 – 923, vol.2.
[19] Asada, M.; “Map building for a mobile robot from sensory data”, Systems, Man and Cybernetics, IEEE Transactions on Volume 20, Issue 6, Nov.-Dec. 1990 Page(s):1326 – 1336.
[20] Gutmann, J.S.; Fukuchi, M.; Fujita, M.; “A Floor and Obstacle Height Map for 3D Navigation of a Humanoid Robot”, Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on 18-22 April 2005 Page(s):1066 – 1071.