Fast and Robust 3D Terrain Surface Reconstruction of Construction Site Using Stereo Camera

Changhun Sung[a], Sang Hoon Lee[a], Young Min Kwon[a] and Pan Young Kim[a]

[a] Hyundai Heavy Industries, Republic of Korea
E-mail: sungch@hhi.co.kr, lshleee@hhi.co.kr, ymkwon@hhi.co.kr, projjang@hhi.co.kr

Abstract –

This paper presents a fast and robust three-dimensional (3D) terrain surface reconstruction system using a stereo camera. The local feature-based 3D terrain surface reconstruction algorithm consists of two major steps: matching and terrain surface reconstruction. In this paper, extracted corners are described by multi-scale descriptors (MSDs), and the matching precision is increased by a quadratic interpolation method. In the terrain surface reconstruction step, the construction terrain surface is modeled using a 3D regular grid plane for its computational efficiency and robustness. The precision of the 3D regular grid plane is improved by inferring the 3D grid vertices from a robustly matched nearest point cloud. The computational complexity of the proposed terrain surface is simplified using 2D and 3D triangle mesh structures.

In order to evaluate the proposed method, different types of 3D reconstruction methods were compared under the conditions of a practical construction site. The results of the experiment show that the MSD-based 3D terrain surface algorithm performed well in terms of robustness and computational efficiency.

Keywords –

3D terrain surface, Stereo Camera, Interpolation, Triangle mesh, Regular grid plane

1 Introduction

Recently, the development of automated construction systems on construction sites for safety and convenience has attracted interest [1,2,3]. In order to operate an automated construction system, it is essential to understand the environment in which construction equipment operates, and the first step of environment recognition is terrain reconstruction. To reconstruct construction site terrain, there are two general approaches: vision-based and Light Detection and Ranging (LiDAR)-based [5].

Figure 1. Example of a construction environment

LiDAR can provide highly precise distance information about the observed terrain. Therefore, when surveying construction sites, LiDAR based terrain reconstruction is normally used [4,5]. However, the use of LiDAR is limited in real-time applications. This is because the post-processing of the three-dimensional (3D) point cloud acquired from LiDAR is a time-consuming task. In addition, LiDAR consumes a large amount of power and it is expensive to mount these devices on real construction equipment.

A vision sensor can provide construction site environment images quickly, at over 30 Hz. Furthermore, it is possible to quickly calculate terrain surface information using current computer vision techniques [6,7,8]. However, there remain some problems when using vision to reconstruct construction site conditions because of its challenging environment. An image of a construction site generally comprises an irregular-appearing surface and textureless ground (see Figure 1). This makes it difficult to use vision-based methods to reconstruct 3D terrain surfaces in practical applications.

In order to deal with this problem, in this paper, we propose a fast and robust 3D terrain surface reconstruction algorithm suitable for construction sites. In the matching step, the quadratic interpolation method is applied to increase the precision of sparse 3D point location. In the 3D terrain surface reconstruction process, regular a grid vertex-based 3D terrain surface reconstruction algorithm is presented. This method efficiently estimated terrain surfaces from 2D and 3D triangle meshes while improving accuracy.

2 Related Work and Proposed Method

2.1 Related Work

In order to infer 3D terrain surface information using a stereo vision sensor, the first step is to find correspondence points between stereo images. There are many approaches to searching for correspondence points. One popular method is to find correspondence points by minimizing global constraints. Graph-cuts [9] and the belief propagation algorithm [10] are well-known global methods for stereo matching. These algorithms have shown good performance [10,11]. However, the selection of parameters such as disparity range and kernel size critically affects the result [9,15].

A local feature-based reconstruction algorithm is another general method. This method finds correspondence points based on interest points represented by a descriptor. In local feature-based reconstruction, how to represent extracted interest points is important, because descriptor distinctiveness significantly affects matching precision. The most widely used local features are SIFT [12] and SURF [14], which are based on gradient information. In various applications, these features have shown good performance. However, their high computational complexity restricts their use in real time applications [15,21].

Recently, a robust and fast feature descriptor, the multi-scale descriptor (MSD), was proposed by Sung et al. [13] for outdoor practical applications. This descriptor already has demonstrated its performance in outdoor applications such as motion estimation and dense 3D reconstruction [3,13]. Therefore, we also used the MSD to describe extracted corner points because of its distinctiveness and computational efficiency.

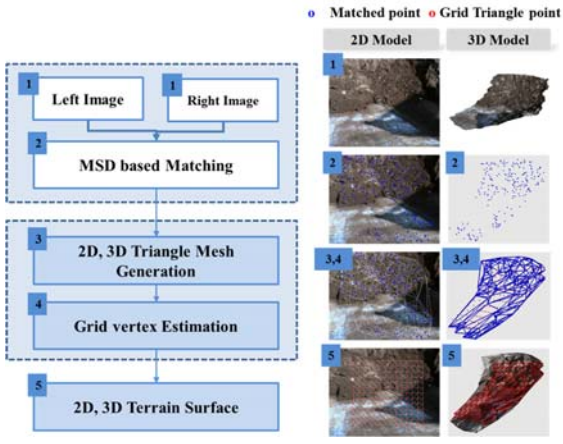

Figure 2. Overall procedure for the proposed 3D terrain surface reconstruction algorithm

2.2 Proposed work

The proposed terrain surface reconstruction method comprises three major steps: MSD-based matching, matched point-based 2D and 3D triangle mesh generation, and a regular grid-based terrain surface creation (see Figure 2). In the matching step, the distribution of the corners and the precision of the 3D point cloud are important, as they significantly affect the accuracy of reconstructed 3D terrain surface. In order to achieve these requirements, an equal number of corners are selected from each sub-region to ensure that the distribution of the interest points is uniform. The precision of the 3D point cloud is improved by quadratic interpolation techniques.

In this paper, the construction site's terrain surface is represented by a regular grid plane. Regular grid planes can dramatically reduce computational burden by inferring regular grid vertices from a robustly estimated triangle mesh. The accuracy of the terrain surface plane is improved by precisely calculated closest 3D points. Furthermore, when reconstructing a 3D terrain surface, we also generate a 2D terrain surface to increase the accessibility to the target surface distance and as information for the operator of the construction equipment.

3 Matching

Various methods have been proposed to extract interest points. One popular detector is the Harris corner detector [17]. This corner detector extracts corner points by eigenvalues analysis. For computational efficiency, the Features from Accelerated Segment Test (FAST) detector has been proposed [18]. This detector finds corner points by comparing the pixel intensity around a candidate point with machine learning techniques. The FAST detector has performed well in various practical tasks [18,21]. In this proposed method, we also find corner points using the FAST detector.

The proposed terrain surface reconstruction algorithm is based on a triangle mesh formed from the nearest extracted corner points. Therefore, in order to reconstruct the terrain surface more efficiency, it is important that the distribution of the corner points is uniform. In order to achieve this goal, in the proposed algorithm, corner points are independently extracted from regularly divided sub-regions. This enables a triangle mesh to be generated evenly over the input image.

After detecting the corner points, each extracted point must be described distinctively for finding correspondence points. Several ways to describe corner points have been proposed [15]. The well-known local feature descriptors are SIFT and SURF [12,14]. These

descriptors have shown good performance in many tasks [15,21]. However, there is a limit to their practical applications because of their computational complexity. In order to overcome this problem, MSD has been proposed for real practical vision application in challenging outdoor conditions [3,13]. Therefore, in order to find correspondence points, each extracted corner point is described by MSD in the proposed method.

In order to keep this paper self-contained, we briefly discuss the concept of MSD. As shown in Figure 3, MSD describes each corner point with three pre-defined scale descriptors $(s1, s2, s3)$. These multiple scale descriptors improve the distinctiveness of the descriptor, as combining the different scales represents the corner point characteristics more efficiently. Furthermore, the computational complexity of MSD is significantly reduced using an integral image.

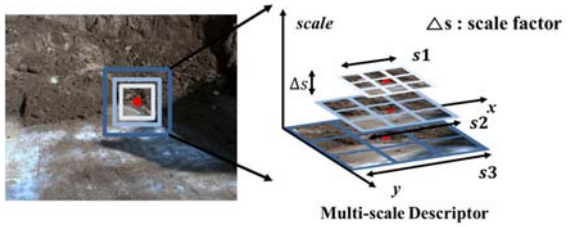

Figure 3. Concept of MSD

Once a pair of correspondence points $x = \{u, v\}$ is found using the MSD descriptor and matching process [3,13,16], the 3D point $X = (u, v, d)^T$ of x is estimated from the difference d, between the matched points, which is called the disparity value. In general, the location of a pixel is presented by an integer value. Therefore, the data type of d is also an integer because it is computed by subtracting corresponding integer-type pixel point locations. This leads to inaccuracy in the 3D point position.

In this paper, to enhance the precision of the 3D point position, a quadratic interpolation method is applied as follows:

$$S(x_{r,t}) = ax_{r,t}^2 + bx_{r,t} + c \qquad (1)$$

where $S(x_{r,t})$ indicates the similarity between the target corner location in left image x_l and the tth candidate corner location in right image $x_{r,t}$. The interpolation $S(x_{r,t})$ is computed by comparing the MSD descriptor similarity between x_l and $x_{r,t}$ (see Figure 4).

The quadratic interpolation equation parameters $\{a, b, c\}$ are computed by solving the following linear equation.

$$\begin{bmatrix} x_{r,t-1}^2 + x_{r,t-1} + 1 \\ x_{r,t}^2 + x_{r,t} + 1 \\ x_{r,t+1}^2 + x_{r,t+1} + 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} s(x_{r,t-1}) \\ s(x_{r,t}) \\ s(x_{r,t+1}) \end{bmatrix} \qquad (2)$$

Once the quadratic interpolation parameters have been estimated, we can find the optimal disparity value d_{op} easily by solving the quadratic equation.

Note that interpolation is a highly important step in the proposed terrain surface reconstruction algorithm. Regular grid plane vertices are inferred based on the triangle mesh estimated from the sparse 3D point cloud. Therefore, once the precise sparse 3D point cloud is calculated, accurate grid plane vertices can be obtained. This also positively influences the precision of the regular grid plane.

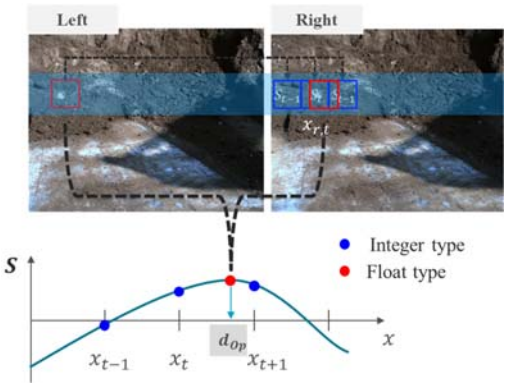

Figure 4. Quadratic interpolation for sparse 3D point cloud improvement

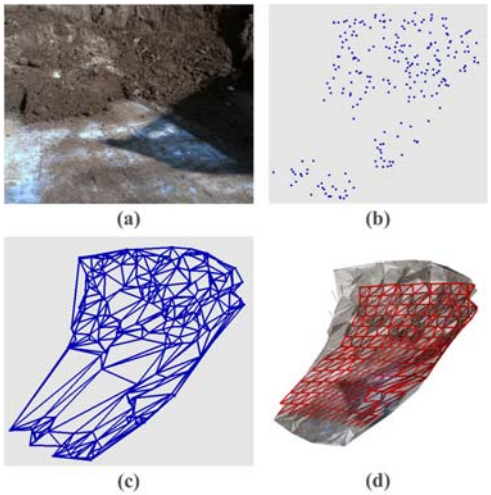

Figure 5. Summary of the proposed 3D terrain surface reconstruction: (a) input image, (b) sparse 3D point cloud, (c) 3D triangle mesh, and (d) reconstructed 3D terrain mesh

4 3D Terrain Surface Reconstruction

The proposed 3D terrain surface reconstruction step comprises three major steps:

(1) sparse 3D point cloud estimation
(2) 2D and 3D triangle mesh generation
(3) 2D and 3D terrain surface creation

Figure 5 shows each step in detail.

Once the disparity value has been estimated by subtracting the location of the corresponding point pair, the 3D point position can be computed based on the stereo camera geometry. The calibration and rectification of the stereo camera is an essential step for obtaining the camera's geometric information. Camera calibration is related to knowing the intrinsic and extrinsic parameters of camera, and the stereo camera rectification step involves horizontally aligning the epipolar line of the stereo images for matching efficiency. In this paper, the well-known open-source library OpenCV [21], was used to calibrate and rectify the stereo images.

In order to infer a regular grid vertex for the 3D terrain surface, we first establish a 2D triangle mesh with the matched point cloud using Delaunay triangulation. Once the 2D triangle mesh set is constructed with the three nearest matched 2D points in the left image, $x_{l,t} = \{u_{l,t}, v_{l,t}\}$, $t = 1, 2, 3$, we then construct the 3D triangle mesh $Ms(\mathrm{x}_l^n)$ using the estimated 3D point $X_{l,t} = \{u_{l,t}, v_{l,t}, d_{l,t}\}$ corresponding to the 2D point $x_{l,t}$, where $d_{l,t}$ is the disparity value of 3D point, $X_{l,k}$ calculated from the camera parameters and quadratic interpolation. The mesh is calculated as follows:

$$Ms_i(X_{l,k}) = a_i u_{l,k} + b_i v_{l,k} + c_i \qquad (3)$$

where Ms_i indicates the i*th* 3D triangle mesh and (a_i, b_i, c_i) are the mesh parameters. The mesh parameters are easily acquired by solving a linear equation with the given matched 3D point set.

Note that construction site can be assumed to be a collection of continuous small surfaces. Construction terrain can be approximately described as a combination of small planes. Therefore, a small-sized triangle mesh reflects the real terrain surface well. Furthermore, we efficiently construct a 3D triangle mesh from the 2D triangle mesh. It is possible to further reduce the computational complexity of the 3D triangle mesh.

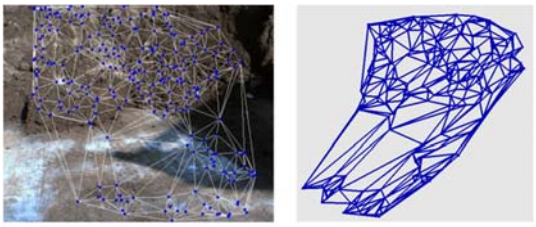

Figure 6. (a) 2D triangle mesh and (b) 3D triangle mesh structure

After building the triangle mesh, we calculate a regular grid of vertices from the 3D triangle mesh. If the k*th* regular grid vertex $x_k = \{u_k, v_k\}$ in the image plane is given, we search for the corresponding 3D triangle mesh Ms_i that contains the target grid vertex using the relationship between the 2D and 3D triangle meshes. We then compute the disparity value d_k of the target grid vertex using the 3D triangle mesh model.

$$d_k = a_i u_k + b_i v_k + c_i \qquad (4)$$

Once the disparity value d_k of x_k is calculated, we can define the 3D position of x_k to be $X_k = \{u_k, v_k, d_k\}$, as previously explained, it is easy to convert the disparity-based 3D position into to Cartesian coordinates using the camera parameters.

Note that proposed regular grid-based 3D terrain surface reconstruction has two major advantages. The first one is that regular grid vertices are inferred from robustly estimated nearest 3D points using the triangle mesh model. This improves the result of the proposed 3D terrain surface. The other one is that the grid vertices are computed by simply solving a linear equation from the triangle mesh model. It is hence possible for the proposed algorithm to be applied in practical applications.

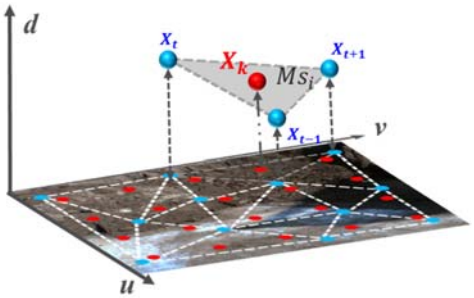

Figure 7. Regular grid vertex estimation

The 3D terrain surface is represented by regular grid plane Gr using the three adjacent grid vertices X_k, k=1, 2, 3 as follows:

$$Gr_i(X_k) = p_i u_k + q_i v_k + r_i \quad (5)$$

where Gr_i indicates the ith regular grid plane containing the three adjacent 3D points, X_k, k=1,2,3. The grid plane parameters (p_i, q_i, r_i) are easily obtained by solving a linear equation with three given grid vertices. Once the grid plane parameters are estimated, we can easily calculate the target area information such as surface slope, area, and distance from the equipment.

Note that the proposed 3D terrain surface is also connected to the corresponding 2D image, similarly to the 2D and 3D triangle meshes. It is a very convenient way to describe the 3D terrain surface. This is because the operator of a piece of construction equipment such as an excavator or wheel loader can intuitively access the 3D information of a certain task area simply by touching the 2D input image using the instrument displays in the cabin.

5 Experiments and Analysis

To evaluate the proposed method, we captured test images from real construction sites to take into account practical conditions. The test images were acquired from a stereo camera mounted on an excavator that consists of two Flea3 USB 3.0 cameras (see Figure 8) [3]. We evaluated each algorithm with four test images Image I (gravel slope), Image II (soil slope), Image III (mud hole), and Image IV (mud bumps). Each test image reflects typical construction environments such as complex surface appearances and various solid materials.

We compared the MSD-based 3D terrain surface reconstruction (MSD-TS) with three popular 3D terrain reconstruction methods: semi-global block matching (SGBM-TS), block matching (BM-TS), and SURF descriptor-based matching (SURF-TS) methods.

Figure 8. Stereo camera system mounted on an excavator

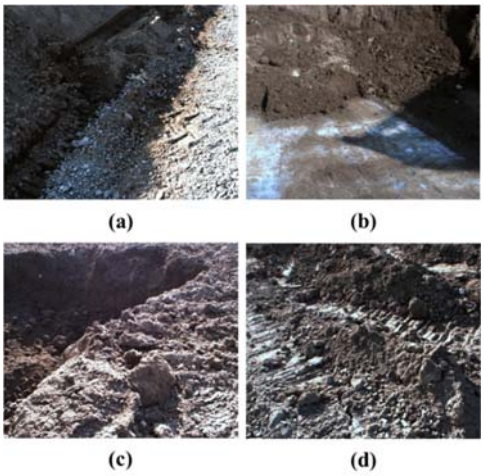

Figure 9. Evaluation image, (a) Image I (gravel slope), (b) Image II (soil slope), (c) Image III (mud hole), and (d) Image IV (mud bumps)

The three comparative algorithms were implemented with the latest OpenCV library [19]. In order to fairly test each algorithm, we used the same 3D terrain surface reconstruction process after finding the correspondence points with each original method. All experiments were performed on a 2.6 GHz single core processor with 8 GB memory.

5.1 Experimental results

The three major steps of the MSD-TS method are shown for each of the four evaluation images in Figure 10. For each test image, the shape of the reconstructed surface is close to the actual slope of the real ground surface.

Note that construction site terrain surfaces consist of textureless and irregular shaped surfaces with different type of solid material such as gravel, mud, and sand. This makes it difficult to reconstruct 3D terrain surfaces. However, given these challenging conditions, the proposed method performed well.

This is explained by the fact that proposed terrain surface model is built from a triangle mesh using a sparse 3D point cloud. Therefore, the accuracy of the sparse 3D points and robustness of triangle mesh plays an important role in the proposed method. The accuracy of the sparse 3D point cloud is improved by quadratic interpolation. The triangle mesh is robustly established with the previously matched nearest three points. This combination of a matched 3D point cloud and robust triangle mesh set has a positive effect on the result of the 3D terrain surface model.

To show the performance of the proposed method clearly, we also compared it the three algorithms, BM-

TR, SGBM-TR, and SURF-TR. As can be seen Figure 11, MSDTR outperformed the BM-TR and SGBM-TR methods for all test images. This is because the matching of patch-based descriptors such as BM and SGBM perform poorly in these kinds of textureless and periodic conditions. This also affected the result of the 3D terrain surface.

SURF-TR achieves a result that was similar to those of MSD-TR because of its good matching performance. However, as mentioned in Section II, SURF-TR has limited suitability for practical applications because of its computational complexity. In contrast, the MSD-TR was fast to calculate in the real 3D terrain surface reconstruction application. We discuss this fact in detail in Section 5.2.

5.2 Computation Time

In Tables I and II, we evaluated the computation time of each method. Although BM-TR and SGBM-TR reconstructed the noisy terrain surface poorly in terms of accuracy, with respect to computation time, BM-TR and SGBM-TR performed well because of the descriptor characteristics of patch-based descriptors.

The average computation time of MSD-TR was about 360 ms. This is far faster than SURF-TR. Furthermore, MSD-TR is 1.2 times faster than SGBM-TR. Note that the operators of construction equipment are generally recommended to operate their machine slowly (under 15 km/h) for safety reasons. In this environment, the computation time of MSD-TR is sufficiently fast to provide ground surface information for automated equipment.

Table 1. Computation Time (ms)

Img		SGBM-TR	SURF-TR	MSD-TR
I	**Matching**	429.9	17,199	358.8
	Terrain Surface	10.7	10.9	10.6
	Total	440.6	17,209.9	369.4
II	**Matching**	415.3	17,131	348.2
	Terrain Surface	9.0	9.4	9.2
	Total	424.3	17,140.4	357.4
III	**Matching**	423.3	17,080	327.2
	Terrain Surface	11.3	11	11.2
	Total	434.6	17,091	338.4
IV	**Matching**	426	17,371	356.8
	Terrain Surface	10.5	10.3	10.2
	Total	436.5	17,381.3	367

Table 2. Total Computation Times (ms)

Img	BM-TR	SGBM-TR	SURF-TR	MSD-TR
I	209.0	440.6	17,209.9	369.4
II	211.1	424.3	17,140.4	357.4
III	212.5	434.6	17,091.0	338.4
IV	209.6	436.5	17,381.3	367.0

6 Concluding Remarks

This paper proposed a robust and fast 3D terrain surface reconstruction algorithm combined with a regular a grid plane for computational efficiency and robustness. In order to enhance the precision of the 3D terrain surface, the vertices of a 3D grid are calculated from a robustly interpolated nearest 3D point set. Computational efficiency is achieved by constructing 2D and 3D triangle mesh structures. The 3D grid vertex is calculated by simply solving a linear equation using the triangle mesh structure.

The experimental results show that MSD-based 3D terrain surface reconstruction performed well in terms of robustness and computation time. The experimental result demonstrate that MSD-TR is suitable for a terrain surface reconstruction system where accuracy and operating time are vital.

References

[1] Yang, M.-D., Chao, C.-F., Huang, K.-S., Lu, L.-Y., & Chen, Y.-P. (2013). Image-based 3D scene reconstruction and exploration in augmented reality. Automation in Construction, 33, 48–60. doi:10.1016/j.autcon.2012.09.017

[2] Brilakis, I., Fathi, H., & Rashidi, A. (2011). Progressive 3D reconstruction of infrastructure with videogrammetry. Automation in Construction, 20(7), 884–895. doi:10.1016/j.autcon.2011.03.005

[3] Sung, C., & Kim, P. Y. (2016). 3D terrain reconstruction of construction sites using a stereo camera. Automation in Construction, 64, 65–77. doi:10.1016/j.autcon.2015.12.022

[4] Arayici, Y. (2007). An approach for real world data modelling with the 3D terrestrial laser scanner for built environment. Automation in Construction, 16(6), 816–829. doi:10.1016/j.autcon.2007.02.008

[5] Du, J.-C., & Teng, H.-C. (2007). 3D laser scanning and GPS technology for landslide earthwork volume estimation. Automation in Construction, 16(5), 657–663. doi:10.1016/j.autcon.2006.11.002

[6] Davison, A. J., Reid, I. D., Molton, N. D., & Stasse, O. (2007). MonoSLAM: Real-Time Single Camera SLAM. IEEE Trans. Pattern Anal. Machine Intell., 29(6), 1052–1067. doi:10.1109/tpami.2007.1049

[7] Brostow, G. J., Shotton, J., Fauqueur, J., & Cipolla, R. (2008). Segmentation and Recognition Using Structure

from Motion Point Clouds. Computer Vision – ECCV 2008, 44–57. doi:10.1007/978-3-540-88682-2_5
[8] Golparvar-Fard, M., Bohn, J., Teizer, J., Savarese, S., & Peña-Mora, F. (2011). Evaluation of image-based modeling and laser scanning accuracy for emerging automated performance monitoring techniques. Automation in Construction, 20(8), 1143–1155. doi:10.1016/j.autcon.2011.04.016
[9] Kolmogorov, V., & Zabih, R. (2001). Computing visual correspondence with occlusions using graph cuts. Proceedings Eighth IEEE International Conference on Computer Vision. doi:10.1109/iccv.2001.937668
[10] Felzenszwalb, P. F., & Huttenlocher, D. P. (2006). Efficient Belief Propagation for Early Vision. International Journal of Computer Vision, 70(1), 41–54. doi:10.1007/s11263-006-7899-4
[11] Hirschmuller, H. (2008). Stereo Processing by Semiglobal Matching and Mutual Information. IEEE Trans. Pattern Anal. Mach. Intell., 30(2), 328–341. doi:10.1109/tpami.2007.1166
[12] Lowe, D. G. (2004). Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision, 60(2), 91–110. doi:10.1023/b:visi.0000029664.99615.94
[13] Changhun Sung, & Myung Jin Chung. (2013). Multi-Scale Descriptor for Robust and Fast Camera Motion Estimation. IEEE Signal Process. Lett., 20(7), 725–728. doi:10.1109/lsp.2013.2264672
[14] Bay, H., Tuytelaars, T., & Van Gool, L. (2006). SURF: Speeded Up Robust Features. Lecture Notes in Computer Science, 404–417. doi:10.1007/11744023_32
[15] Mikolajczyk, K., & Schmid, C. (2003). A performance evaluation of local descriptors. 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings. doi:10.1109/cvpr.2003.1211478
[16] Veksler, O. (2005). Stereo Correspondence by Dynamic Programming on a Tree. 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. doi:10.1109/cvpr.2005.334
[17] Harris, C., & Stephens, M. (1988). A Combined Corner and Edge Detector. Procedings of the Alvey Vision Conference 1988. doi:10.5244/c.2.23
[18] Rosten, E., & Drummond, T. (2006). Machine Learning for High-Speed Corner Detection. Lecture Notes in Computer Science, 430–443. doi:10.1007/11744023_34
[19] G. Bradski, A. Kaehler. Learning OpenCV. O'Reilly Media Inc., 2008.
[20] Beis, J. S., & Lowe, D. G. (1997). Shape indexing using approximate nearest-neighbour search in high-dimensional spaces. Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition. doi:10.1109/cvpr.1997.609451
[21] Gauglitz, S., Höllerer, T., & Turk, M. (2011). Evaluation of Interest Point Detectors and Feature Descriptors for Visual Tracking. Int J Comput Vis, 94(3), 335–360. doi:10.1007/s11263-011-0431-5

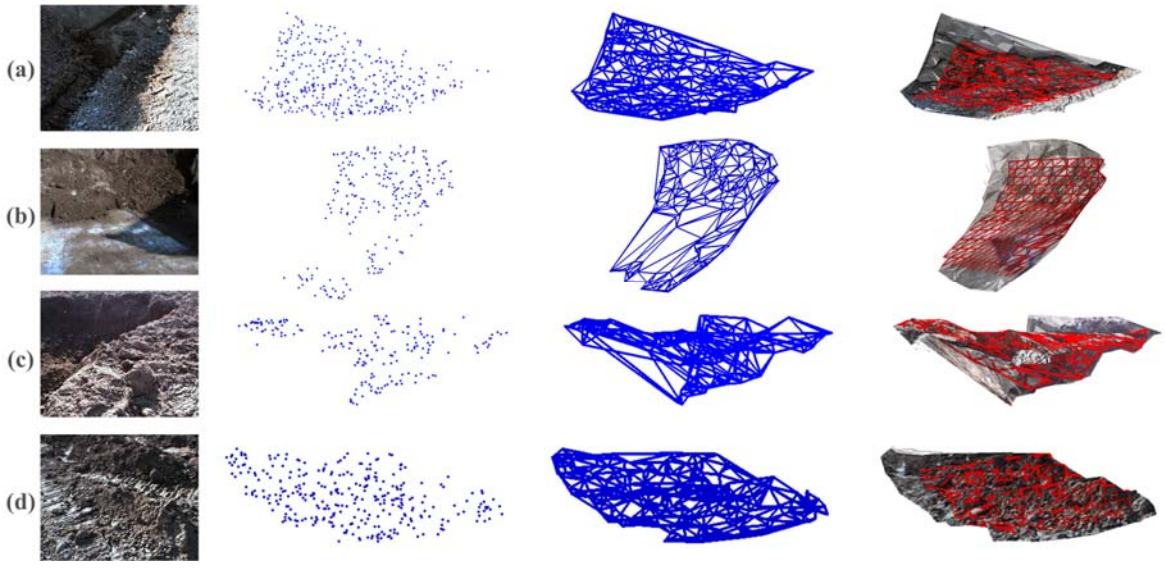

Figure 10. Results of the MSD-TR 3D terrain surface reconstruction method: test images (column 1), sparse 3D reconstruction (column 2), triangle mesh (column 3), and dense reconstruction (column 4)

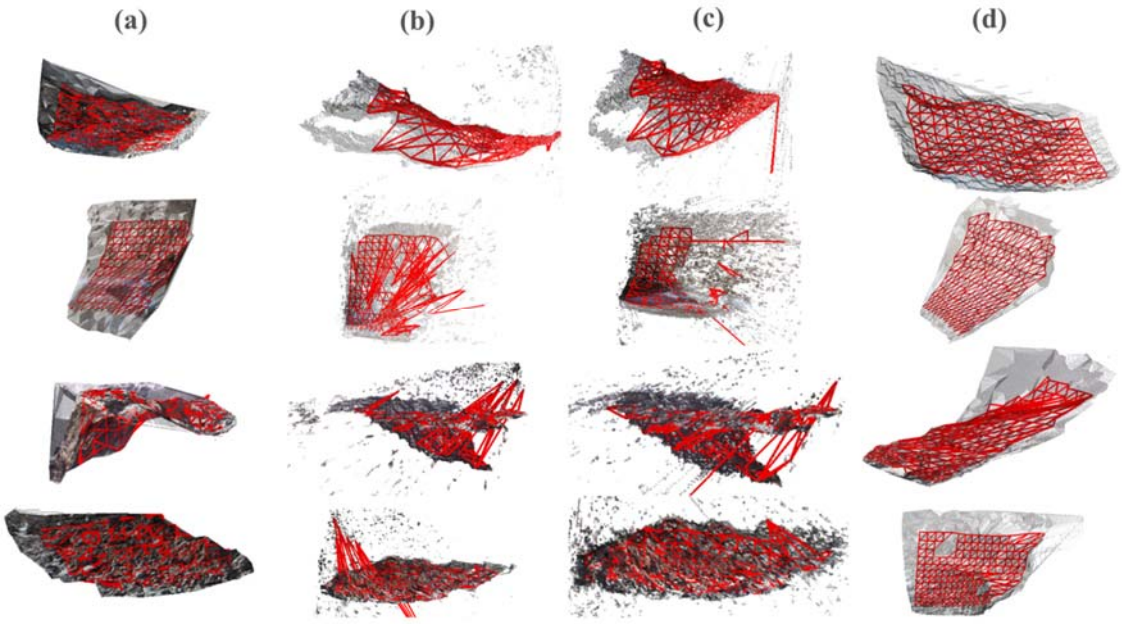

Figure 11. Comparison of the results of different 3D terrain surface reconstruction algorithms: (a) MSD-TR, (b) BM-TR, (c) SGBM-TR, and (d) SURF-TR

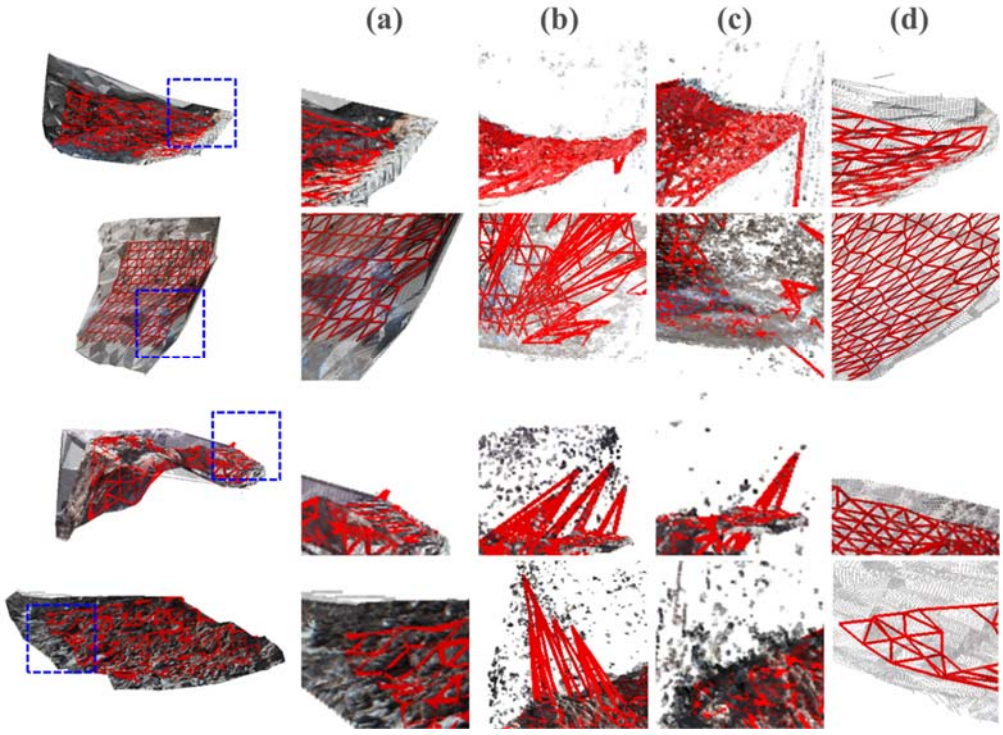

Figure 12. Magnified results of (a) MSD-TR, (b) BM-TR, (c) SGBM-TR, and (d) SURF-TR