Depth Map and Normal Direction Fusion

Supervised by Fabio Maninchedda

Fredrik Bjerkås
ETH Zürich
bjerkasf@student.ethz.ch

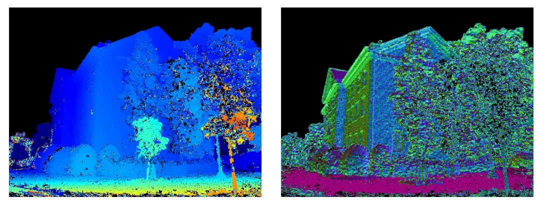
Kunal Shrivastava
ETH Zürich
kunals@ethz.ch

David Trüssel
ETH Zürich
trdavid@ethz.ch

June 23, 2017

Abstract

In this paper, we consider the problem of iso-surface extraction from a signed distance field while preserving sharp features. We have explored two alternatives to the widely used Marching Cubes algorithm, namely Extended Marching Cubes and Dual Contouring. Our results show that both methods provide better feature preservation performance compared to Marching Cubes.



(a) Depth map (b) Normal map

Figure 2: Examples of input to the pipeline

1. Introduction

1.1. Pipeline

The reconstruction pipeline, illustrated in Figure 1, starts with a set of depth and normal maps (seen in Figure 2) which are used as input to a modified version of the TV-Hist algorithm [1]. The algorithm outputs a signed distance field from which an iso-surface can be extracted, and up until now this has been done using the Marching Cubes (MC) [2] algorithm. This method does not preserve sharp features as well as other methods available, so corners and edges will appear rounded in the extracted surface. We have experimented with other iso-surface extraction algorithms like Dual Contouring (DC) [3] and Extended Marching Cubes (EMC) [4] to improve sharp feature preservation, and thus increase the quality of the resulting mesh.

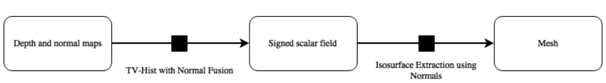


Figure 1: Flowchart of the pipeline

1.2. The TV-Hist Algorithm

The TV-Hist algorithm takes in a set of depth maps, $\{r_i : D_i \to \mathbb{R}\}$ where $D_i \subseteq \mathbb{R}^2$ is the image domain, and computes truncated distance fields $f_i : \Omega \mapsto [-1, 1]$ over a voxel space $\Omega \subseteq \mathbb{R}^3$, see Figure 3. The convention is that f_i has positive sign for carved voxels (points lying in front of the hypothetical surface), and negative for filled voxels. The interval $[-1, 1]$ is sampled by evenly spaced bin centers in each voxel for computational efficiency. This results in the following energy functional which is minimized to find an optimal signed distance function, u:

$$E^{\text{TV-Hist}}(u) = \int_\Omega \left\{ |\nabla u| + \lambda \sum_j n_j |u - c_j| \right\} d\vec{x} \qquad (1)$$

The first term is called the Total Variation (TV) term [5] and its main property is that it penalizes the level sets in u, which in this case is exactly the surface area. The second term is a data fidelity term that measures the distance from u to the bin center c_j with respective frequency n_j. Note that the data term is integrated over the voxel space, hence a histogram is maintained for each voxel.

The baseline for this project is a modified version of the algorithm, developed by last years team in the 3D Vision course working on the same topic, that incorporates normal information in the depth map fusion to improve sharp feature preservation in the distance field generation step. This is done by first obtaining a per-voxel normal estimate by projecting the normal maps into the voxel grid and performing a clustering technique to obtain up to three normals

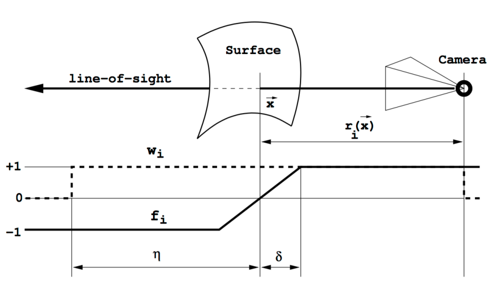


Figure 3: Generation of 3D distance fields from range images. Figure taken from [6].

per voxel. The dominant normal direction can then be used to modify the previous energy functional (1) where the TV term penalizes change in all directions equally. In the modified formulation, changes in the direction of the dominant normal are penalized less. The TV term is replaced by

$$\phi(\nabla u) = \max_{p \in \mathcal{W}} \langle \nabla u, p \rangle \quad (2)$$

where $\mathcal{W}$ is a half space sphere aligned to the dominant normal in voxel. A 2-dimensional illustration can be seen in figure 4. By lowering the height h, changes in the direction of the dominant normal n are penalized less, and by increasing the radius R changes in other directions are penalized more.

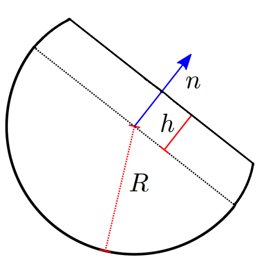


Figure 4: 2-dimensional half sphere aligned to the dominant normal in voxel

2. Iso-Surface Extraction

After obtaining the signed distance field, with negative truncated distance values inside the surface and positive values outside, the next step is to extract a surface. Ideally a surface is extracted as a mesh which is a collection of faces and vertices. The vertices of this mesh are points in 3D space close to the zero level set in the signed distance field. One class of methods used to find these vertices are the cube-based methods which generates polygons for each cube that intersects the contour. The signed distance field generated from TV-Hist is discretized as a uniform grid with scalar values assigned to each grid point. We can iterate over the grid points and consider each one of them as corners of a cube whose dimensions are determined by the resolution of the grid. If any of the grid points (corners) of this cube has a different sign than the others, it means that the contour is passing through it. We can then find the intersecting edges and obtain a vertex for each of them. To do this, we have considered the following (cube based) methods: *Marching Cubes* obtains the zero level point on each edge that intersects the cube. It does not allow us to use normal information to improve the placement of the vertices, and as a result sharp features will appear rounded in the reconstructed mesh. Methods like *Extended Marching Cubes* and *Dual Contouring* does allow us to use normal information to improve the positioning of these vertices, and thus improve sharp feature preservation. Figure 5 illustrates the three methods on a 2-dimensional grid.

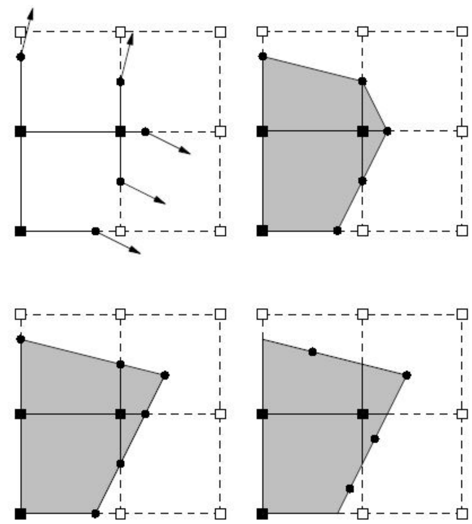


Figure 5: A signed grid with edges tagged by Hermite data (upper left), its Marching Cubes contour (upper right), its Extended Marching Cubes contour (lower left), and its Dual Contour (lower right) [3]

2.1. Extended Marching Cubes

As implied by its name Extended Marching Cubes is an extension of the Marching Cubes algorithm as described by Kobbelt et al. [4]. This surface extraction method adaptively refines those grid cells which contain a piece of the surface. The main goal is to reduce the approximation error from the extracted surface to the real surface which stems from the discretization through the grid representation. Local tangent information of the distance field allows for a better approximation of the surface near a sharp feature. Instead of directly connecting the intersection points of the contour, the contour normal is used to compute a tangent element for each intersection point. Intersecting the tangent elements yields an additional sampling point close to the sharp feature. Fig. 6 illustrates this by a 2D example. The algorithm has the following major steps:

- Check for each voxel if a feature is present (assuming maximum one feature per voxel).
- If a feature is present the gradient information at each edge intersection point is used to define local tangent elements and calculate one new sample point. This new vertex is used as the center of a triangle fan. Otherwise the original MC algorithm is used for a cell without any features.
- Since the additional sample point was added as the center of a triangle fan without considering neighboring voxels, the mesh has to be adjusted by a post-processing step. Edges are flipped if they then connect two new feature points.

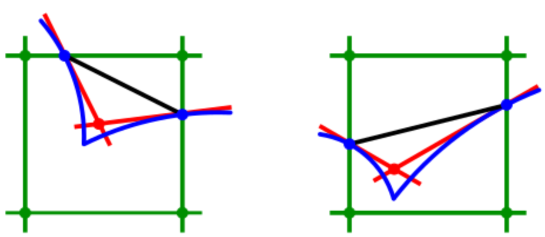


Figure 6: 2D example of using local tangent information in EMC: Grid (green), Real surface (blue), MC (black), EMC (red) [4]

The feature detection mentioned above is used to classify if a cell contains a sharp feature. The heuristic used here is the opening angle of the normal cone spanned by the normals of the corresponding grid points. If the opening angle is smaller than the specified limit, the cell is classified as containing a sharp feature. If a feature is detected it has to be further classified as an edge or a corner using the maximum deviation of the normals. If the maximum deviation is bigger than a specified threshold then the feature is assumed to be a corner. After this classification into edges and corners we then try to add a new sample point close to the feature. The new sample point is the solution of a linear system defined by the tangent elements of the sampling points. This is done by using a Singular Value Decomposition based technique which minimizes the average squared deviation from all tangent elements (see the next section for more details).

The performance gain of EMC compared to MC can be seen in Figure 7 where two functionally represented cubes are misaligned with the sampled grid. This causes the edges of the surface reconstructed with MC to be rounded, while the surface reconstructed with EMC is difficult to distinguish from the ground truth.

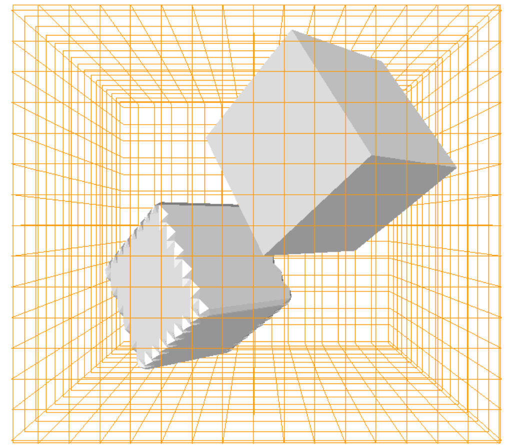


Figure 7: Comparison between MC and EMC. Implicit cubes that are misaligned with the grid. MC (lower left), EMC (upper right)

2.2. Dual Contouring

The main advantage of the EMC method is that it allows us to use normal information to place the vertices associated with cubes that contain features. However one drawback is that we have to explicitly look for features. As an alternative to EMC, the Dual Contouring method does the following:

- For each cube that intersects the contour, generate a vertex x that minimizes the the quadratic error function (QEF), given by

$$E[x] = \sum_i (n_i(x - p_i))^2 \tag{3}$$

  where the point p_i is the i-th intersection between the contour and the cube, and n_i is the normal in that point.

- For each edge that intersects the contour, generate a quadrilateral face connecting the minimizing vertices of the four cubes containing the edge.

The difference to EMC is that we now minimize the QEF for positioning every vertex of the contour, hence there is no need to explicitly look for features. We have implemented this algorithm in Python using numerically stable and fast solvers. Solving the QEF comes down to solving a least squares problem of the form

$$Ax = b \tag{4}$$

where A is a matrix whose i-th column is given by the normal n_i in intersection point i, and b is a vector whose entries are given by the inner products $\langle n_i, p_i \rangle$. Since A is not necessarily a square matrix we compute the Moore-Penrose pseudo-inverse using the Singular Value Decomposition. Due to noisy normals, the singular values of A sometimes become too small, causing numerical instability. As a result the computed vertex x will be placed too far away from the cube and appear as an irregularity in the mesh. To resolve this issue [3] suggests discarding small singular values, and the corresponding left and right singular vectors. The resulting vertex will then not be completely accurate but will still be a good approximation.

We have followed a slightly different approach in our implementation. Instead of discarding the small singular values we replace the corresponding vertex with the Marching Cubes solution. This does not affect the accuracy, and helps make the code quicker as zero level vertices used for Marching Cubes are already computed before minimizing the QEF.

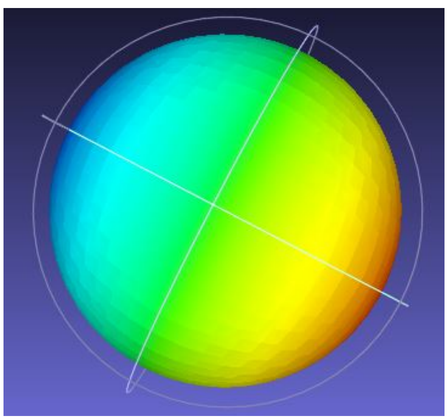


Figure 8: Resulting mesh. Dual contouring was applied to an implicit sphere for testing

The Dual Contouring implementation was applied on a simple implicit sphere defined by a continuous distance function. The function takes in a 3D coordinate and outputs the distance to the surface of the sphere in the direction pointing to the centre. Similar to the signed distance field discussed earlier, the space outside has positive values and the space inside has negative values. The resulting normal function is just the gradient of this function. The result is a smooth sphere with no artifacts, seen in Figure 8, as there is no noise applied in this case. We will see later that when we apply the same algorithm on a noisy cube represented by a discretized signed distance function reconstructed from depth maps, we do not obtain a perfect result.

3. Implementation

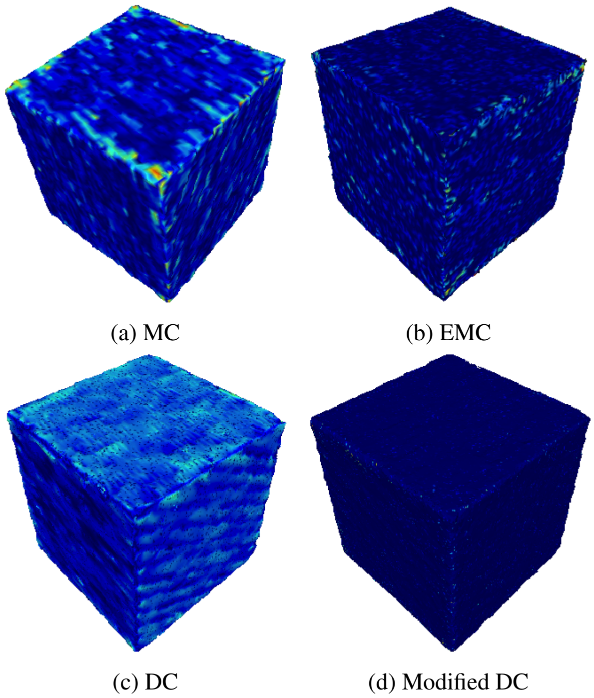
- Extended Marching Cubes was implemented in C++ using a library called IsoEx [7]. The algorithm takes in the scalar distance field (from TV-Hist) stored in a text file. This is a slight modification to the library implementation as described in section 6.
- Dual Contouring was implemented in Python. This code also takes the same text file for the scalar distance field as an input. We used inbuilt Python packages for the mathematical calculations in the algorithm (for example least squares minimization of the QEF). We used the Mayavi toolbox to generate the mesh from the obtained vertices.

The code is submitted along with the report. It takes the scalar distance field corresponding to a $100 \times 100 \times 100$ cube as an input. This distance field was generated by the TV-Hist implementation provided to us. EMC and DC can be run separately by following the instructions in the accompanying read-me files.

4. Experiments and Results

In order to evaluate our implementations of EMC and DC, we first obtained a signed distance field by using depth and normal maps created from functionally generated cubes with some added noise as input to the modified TV-Hist algorithm. This simplified the evaluation of the performance of the implementations, since the ground truth was available. Although it is possible to evaluate the improvements achieved by these methods just by looking at the resulting meshes, it is not a reliable approach. At a low resolution, the improvements might be visible, but at a higher resolution it is difficult to distinguish the MC result from the results of EMC or DC with the naked eye. Therefore, we used a metric called Hausdorff Distance [8][9] which measures the extent to which each point of a "model" set lies near some point of an "image" set and vice versa. Thus this distance can be used to determine the degree of resemblance between two objects that are superimposed on one another. We used this to evaluate the performance of the reconstructions of the cube model by measuring the closeness between the reconstructions and the ground truth. We sampled 10 000 points from each of the reconstructed surfaces and calculated the Hausdorff distances to the ground truth using Meshlab [10]. We then generated error heat maps in order to visualize the results, which can be seen in Figure 9. The red areas are further away from the ground truth, and the blue areas are closer.

Note the high error values in the Marching Cubes result near the edges and corners. DC is better at reconstructing edges and corners than MC, but it actually performs worse on the flat regions. In the modified version of DC where



(a) MC (b) EMC

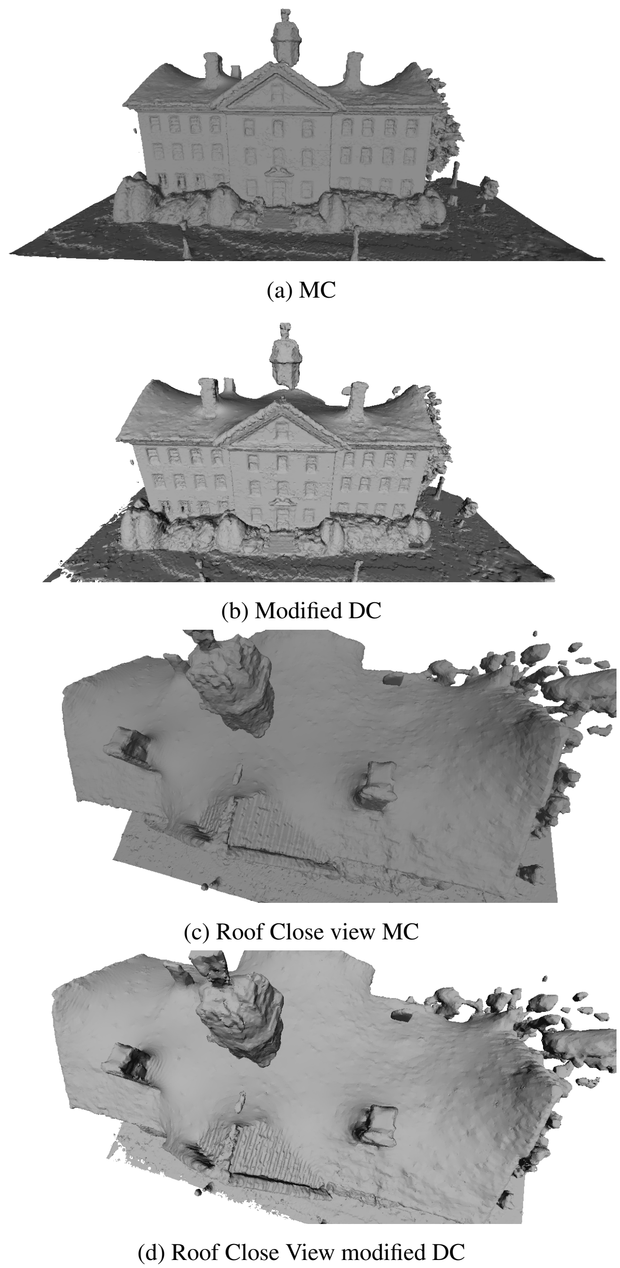
(c) DC (d) Modified DC

Figure 9: Heat maps visualizing the distance error between the reconstructed surface and the ground truth.

the erroneous vertices are replaced with MC vertices, we see that the overall error is significantly less on edges and corners, and even the faces are less erroneous (except few isolated points). The reason for this is that the regions where the error was large due to noisy normals we no longer rely on normals, but switch back to the original MC solution. [3] suggests rejecting singular values that are very small in the least squares optimization. The DC result in Figure 9c, was obtained by rejecting singular values that were smaller than 0.1 times the largest singular value. The result was however not better than the modified DC. This is generally a problem with normal DC, where determining which singular values to reject might not be straightforward in every case. Further it can be seen in Figure 9b that the faces of the EMC reconstruction are as accurate as Figure 9a and 9d, this is because the faces in all these 3 solutions are more or less obtained from MC. EMC does not solve the QEF where there are no features and modified DC replaces bad solutions inflicted by noise with the MC solution. The faces in modified DC are indeed a mix of the DC solution where it is acceptable, and the MC solution.

In an attempt to find a single quantitative number to evaluate the reconstructions, we calculated the root mean squared (RMS) Hausdorff distance errors. However, there did not seem to be a clear relation between the RMS values and what we observed in the heat maps.

We took the method that provided the best performance on the small model, namely the modified version of DC, and applied it to a larger and more complex building dataset. Here, the ground truth was not available and therefore the Hausdorff metric could not be applied. Instead, we compared our result to the baseline reconstruction method done with MC. As can be seen in Figure 10, the modified DC reconstruction has slightly sharper features, e.g. the bricks on the roof is more detailed in Figure 10d than in Figure 10c.



(a) MC

(b) Modified DC

(c) Roof Close view MC

(d) Roof Close View modified DC

Figure 10: Comparison on Building Data Set

5. Conclusion and Further Work

Both EMC and DC proved to be better at preserving sharp features than MC. Quality wise modified DC performed slightly better than EMC, because it does not look for features, but obtains a vertex in accordance with the normals at every cube in the grid. The advantage is that when a relatively smooth/flat surface has a small feature, DC will manage to capture it in the reconstruction where EMC will not. However the improvements become less obvious when the resolution of the reconstruction is large. The improved sharp feature preserving performance also comes at a cost of computational efficiency because the DC solution has to solve a QEF for every grid point.

For further work we propose the following points:

- Find a better metric than Hausdorff distance to assess the quality of the reconstruction. A metric that only takes the improvement in the vicinity of sharp features into account would be most desirable.
- The depth map fusion is based on a scalar distance field, however the EMC algorithm might work better using a directed distance field as proposed in [4]. Therefore a directed distance field should be generated earlier in the pipeline to better integrate the TV-Hist algorithm with EMC. A directed distance field represents the distance from each grid point to the closest zero level point in each cardinal direction.
- Improve the run time performance of the Dual Contouring algorithm by using an Octree representation for adaptive resolution. Existing implementations of Dual Contouring are using Octrees, but they are intended for re-meshing.
- Find a better way to handle small singular values in the least squares problem in the DC implementation.

6. Our Contribution

Extended Marching Cubes: Although we found C++ libraries which implemented the EMC algorithm, most of them were designed only to be used with vector distance grids. This kind of grid stores vector distances, that is each grid point stores 3 orthogonal components of distance to the closest zero level point. The output of the TV-Hist implementation was a scalar distance function. This had to be converted to a vector distance field which was done using the normal information. We then used functions from the IsoEx library to run EMC.

Dual Contouring: This was implemented from scratch. The implementation was done in Python and it is tailored for a discretized scalar distance grid. This includes a function that interpolates the signed distance values at any non-integer point in 3D space. We also came up with a modified algorithm which is run-time performance wise somewhere between EMC and DC. The original DC paper already suggests something similar by proposing a numerically stable least squares approach and later rejecting small singular values [3]. We have just taken a less complex approach which works reasonably well. We do not claim that this will reduce time complexity of the algorithm in every case.

7. Work Distribution

- David was responsible for implementing EMC algorithm. He modified the Iso-ex library [7] to be compatible with the existing C++ implementation of TV-Hist. He helped produce the poster for the final presentation, and participated in the scientific paper presentation.
- Kunal was responsible for implementing the DC algorithm, which he did using Python. He also implemented and tested a modified version of the DC algorithm described in this paper under the DC paragraph. Additionally, Kunal produced material for both the scientific paper presentation and the final poster presentation.
- Fredrik did a theoretical review of the TV-Hist algorithm, and a literature review on possible iso-surface extraction methods suited for this project. He worked on a method called Cubical Marching Squares which did not perform well with the signed distance field produced by TV-Hist, and was therefore abandoned. He produced summaries of both the paper presented by the group in class, and of the paper presented when the group acted as a moderator. Helped design the poster for the final presentation.

References

[1] Christopher Zach. "Fast and high quality fusion of depth maps". In: *In 3DPVT*. 2008.

[2] William E. Lorensen and Harvey E. Cline. "Marching Cubes: A High Resolution 3D Surface Construction Algorithm". In: *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH '87. New York, NY, USA: ACM, 1987, pp. 163–169. ISBN: 0-89791-227-6. DOI: 10.1145/37401.37422. URL: http://doi.acm.org/10.1145/37401.37422.

[3] Tao Ju et al. "Dual Contouring of Hermite Data". In: *ACM Trans. Graph.* 21.3 (July 2002), pp. 339–346. ISSN: 0730-0301. DOI: 10.1145/566654.566586. URL: http://doi.acm.org/10.1145/566654.566586.

[4] Leif P. Kobbelt et al. "Feature Sensitive Surface Extraction from Volume Data". In: *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH '01. New York, NY, USA: ACM, 2001, pp. 57–66. ISBN: 1-58113-374-X. DOI: 10.1145/383259.383265. URL: http://doi.acm.org/10.1145/383259.383265.

[5] Leonid I. Rudin, Stanley Osher, and Emad Fatemi. "Nonlinear Total Variation Based Noise Removal Algorithms". In: *Proceedings of the Eleventh Annual International Conference of the Center for Nonlinear Studies on Experimental Mathematics : Computational Issues in Nonlinear Science: Computational Issues in Nonlinear Science*. Los Alamos, New Mexico, USA: Elsevier North-Holland, Inc., 1992, pp. 259–268. URL: http://dl.acm.org/citation.cfm?id=142269.142312.

[6] Christopher Zach, Thomas Pock, and Horst Bischof. "A Globally Optimal Algorithm for Robust TV-L1 Range Image Integration." In: *ICCV*. IEEE Computer Society, 2007, pp. 1–8. ISBN: 978-1-4244-1631-8. URL: http://dblp.uni-trier.de/db/conf/iccv/iccv2007.html#ZachPB07.

[7] Mario Botsch. *IsoEx Library [Extended Marching Cubes Implementation]*. https://www.graphics.rwth-aachen.de/IsoEx/. 2011.

[8] D. P. Huttenlocher, G. A. Klanderman, and W. A. Rucklidge. "Comparing Images Using the Hausdorff Distance". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 15.9 (Sept. 1993), pp. 850–863. ISSN: 0162-8828. DOI: 10.1109/34.232073. URL: http://dx.doi.org/10.1109/34.232073.

[9] Paolo Cignoni, Claudio Rocchini, and Roberto Scopigno. "Metro: measuring error on simplified surfaces". In: *Computer Graphics Forum*. Vol. 17. 2. Blackwell Publishers. 1998, pp. 167–174.

[10] Paolo Cignoni et al. "MeshLab: an Open-Source Mesh Processing Tool". In: *Eurographics Italian Chapter Conference*. Ed. by Vittorio Scarano, Rosario De Chiara, and Ugo Erra. The Eurographics Association, 2008. ISBN: 978-3-905673-68-5. DOI: 10.2312/LocalChapterEvents/ItalChap/ItalianChapConf2008/129-136.