GENERATING ORIENTED POINTSETS FROM REDUNDANT DEPTH MAPS USING RESTRICTED QUADTREES

M. Rothermel*, N. Haala, D. Fritsch

Institute for Photogrammetry, University Stuttgart, Germany - forename.lastname@ifp.uni-stuttgart.de

KEY WORDS: Matching, Surface, Three-dimensional, Point Cloud, Fusion, Triangulation

ABSTRACT:

In this article we present an algorithm for the fusion of depth images derived by dense image matching (DIM). One key idea of our algorithm is to generate a 2D triangulation for each available depth map in the image sequence using a restricted quadtrees (RQT). On the one hand this guarantees *matching triangulations*, on the other hand this creates the possibility to reduce points in the noise range not contributing to the geometry in a controlled manner. By vertex decimation computational efforts in subsequent processing steps are eased. In order to reduce IO overhead, the algorithm is designed in an iterative way: an initial triangulation is lifted to 3D space and, if pixel footprints are comparable, updated using depths of the subsequent map in the sequence. Previously not observed surface regions or surface patches observed only with adverse precision are removed from the existing model and updated by more appropriate triangulations. Thereby differences in scale across depth maps are handled which is particularly important to preserve details and obtain surfaces with the best reconstruction geometry. To remove outliers visibility constraints are forced. The input is overlapping depth images and their poses in space, the output are point coordinates representing the surface, their respective normals and to some degree spatial neighbourhood information of points represented as a non-watertight mesh. The performance of the algorithm will be evaluated on a close range and a oblique aerial dataset.

1. INTRODUCTION

Image based surface reconstruction has been a vivid research area in the last decades. Advances in both sensors and algorithms enable reconstruction of high quality point clouds providing depth information for almost every pixel at accuracies up to ground sampling distance (GSD) level (Haala and Rothermel, 2012). To guarantee completeness, robustness and precision, image sets for the purpose of 3D reconstruction are typically collected with high redundancy (high image overlap). Particularly the class of image based multi-view reconstruction systems (MVS) (Seitz et al., 2006) which generate one or more depth maps per frame face one problem: spatial information of the same surface point is reconstructed multiple times within the image sequence, which leads to a significant amount of redundant observations varying in precision and reliability. The reason for variances in reconstruction quality of depths across stereo models are manifold and comprise

- Differences in image scale
- Differences in the number of observations available across the views used for depth reconstruction at image matching / forward intersection stage
- Differences in ray intersection angels across views
- Matching errors due to image blur, fronto parallel effects, pixel locking effects, etc.

To reduce data, removing erroneous depths and exploit redundancy an adequate fusion strategy is required. Based on the cleansed data subsequent meshing algorithms can be applied.

Some MVS systems avoid the depth map fusion step by linking surface points directly at the matching stage. Beside the expandability of the integration step these algorithms take advantage of multi-photo consistency measures. Starting with a sparse set of points, PMVS (Furukawa and Ponce, 2010) optimizes position and normals of surface patches based on multi-photo consistency. Iteratively the surface is grown by propagating patches to proximate regions. If no foreground / background information is available, the final oriented point sets are meshed using Poisson Reconstruction (Kazhdan and Hoppe, 2013) followed by a refinement step evaluating photo consistency of each mesh vertex. Starting with an even more coarse mesh produced by Delaunay triangulation of SIFT (Lowe, 1999) feature points the approach proposed by (Hiep et al., 2009) grow / reconstruct detailed surface using variational optimization of a global cost function based on multi-view consistency measures.

The class of MVSs we are focusing on is based on stereo matching producing depth maps for each stereo model which consequently have to be merged. An example of a stereo algorithm which gained popularity in recent years is the Semi-Global Matching method (SGM) proposed by (Hirschmüller, 2008). Due to its high robustness regarding parametrization, limited hardware requirements and moderate computation times it is often the technique of choice for real world applications, such as automotive driver assistance systems (Hermann and Klette, 2012) or the generation of digital elevation or city models (Rothermel et al., 2014). SGM is a stereo method solving the correspondence problem for an image pair, ideally leading to one 3D coordinate or depth per pixel. For high redundant image sets proximate stereo pairs overlap which results in a tremendous amount of (redundant) points. Depth maps used by the method presented in this article were generated using a coarse-to-fine adoption of SGM (Rothermel et al., 2012). Besides reducing memory load and computational costs an approach for the fusion of stereo pairs corresponding to the same reference image is implemented. This fusion is based on geometric consistency of depth estimations and produces high quality depth maps for image sets which provide sufficient accurate orientations. Assuming 2.5D structure of reconstructed surfaces, depth estimations can be fused using orthographic projection onto a plane, rasterization and cell-wise filtering which results in one height value per cell. Thereby the number of points can be reduced significantly whereas redundancy is exploited to

[*] Corresponding author. This is useful to know for communication with the appropriate person in cases with more than one author

This contribution has been peer-reviewed.
doi:10.5194/isprsarchives-XL-3-281-2014

eliminate outliers and increase accuracy. While 2.5D assumption is sufficient for airborne nadir scenarios, applications like oblique aerial images in urban areas or terrestrial data collection require more general approaches. In this article we want to address the problem of depth map fusion capable of preserving real 3D information. Apart from the computation of reliable point coordinates also the computation of robust normals is also of interest, since the latter can serve as input for subsequent meshing techniques as Poisson Reconstruction (Kazhdan and Hoppe, 2013).

A purely geometric algorithm for depth map merging was proposed in Polygon Zippering (Turk and Levoy, 1994). The method generates triangle meshes by simply constructing two faces from four adjacent depth estimations. Suspicious triangles are removed by evaluation of the triangle side lengths. After alignment of meshes, redundant triangles are removed from the boundaries of single patches and remainders are connected. However, redundancy is not exploited and visibility constraints are not implemented. Perhaps most similar to our approach is a method presented for the fusion of noisy depth maps (Merrell et al., 2007) in real-time applications. Proximate depths maps are rendered in one reference view. Redundant depths per pixel are checked for geometric consistency and are filtered using occlusion and confidence checks. After consistent depth estimations are averaged a mesh is constructed on the depth maps using quadtrees and lifted to 3D space. In contrast, we emphasize image scale within our fusion method and prefer a restricted quadtree for meshing.

Region quadtrees are data structures spatially separating 2D space. Starting from an initial quad (node), each quad is recursively split into four sub-quads (sub-nodes) until a certain level is reached or each node satisfies a specific criteria. Quadtree triangulations of 2.5D elevation data was covered in multiple works, mainly in the field of computer graphics, for regular and irregular data points, see (Sivan and Samet, 1992), (Pajarola et al., 2002) and (Pajarola, 2002) for an excellent overview. In (Pajarola, 1998) a special type of this structure, a restricted quadtree (RQT), was used for the triangulation of digital elevation data for the purpose of terrain visualization. The idea is to build a RQT on regular 2.5D height data raster from which then a simplified triangulation can be derived. One of the key properties of RQTs is the option to extract matching triangulations from regular grids, meaning that resultant surfaces represented as triangle meshes are crack-free. Moreover, vertices of such triangulations are guaranteed to fulfill specific error criteria, or in other words, allow to discard vertices (linked to depth estimations) satisfying this criteria. We exploit this property to discard depth estimations not contributing to the geometry at an early stage of the fusion process.

A large portion of depth map fusion builds up on volumetric range integration of depth maps (VRIP) (Curless and Levoy, 1996). Typically a signed distance field is computed on a (multi-level) octree structure by projection of depth estimations from which then a triangulation can be derived for example using the Marching Cube algorithm (Lorensen and Cline, 1987). In a similar way (Zach et al., 2007) and (Zach, 2008) reconstruct a zero level set in voxel space. Then a surface is extracted by minimizing a global energy functional based on the TV-L1 norm, claiming smoothness and small differences to the zero level set. Although giving impressive results, the computational effort and memory demands are significant. Moreover, depth samples across views possessing different scales is challenging for VRIP approaches. One example addressing this issue is the scale space representation presented in (Fuhrmann and Goesele, 2011). They build a multi-level octree holding vertices at different scales. Triangles from the depth maps are inserted according to their pixel footprint. This way a hierarchical signed distance field is generated. For iso-surface extraction the most detailed surface representation is preferred.

Our approach can be divided into two stages: Depth map filtering and the actual fusion step. First, in order to remove outliers, depth maps are filtered based on the local density of depth estimations. In the fusion step a RQT is used to extract a matching triangulation of an initial depth map. The error criteria is formulated using local pixel footprints as a measure of precision. If a specific depth value is assumed to be located in the noise band it is not considered for further processing. The compressed sub mesh is then lifted to 3D space and builds the initial part of the model state. Iteratively the model is projected to the next depth map using depth buffering as well as frustum- and backface culling. If the pixel footprints (the image scale) of existing vertices and map values are comparable the existing surface points are refined. If the scale differs, the most accurate surface representation is chosen. Surface patches not contained by the model are triangulated and added. Note that this is a purely geometric approach although regularization was applied during DIM process.

2. METHODOLOGY

In this chapter we describe the single steps performed to extract oriented point sets from a set of oriented depth maps. Each available map is initially filtered to remove obvious errors introduced by the DIM process. This is discussed in section 2.1. In section 2.2 properties of RQT and extraction of matching triangulation from depth maps using an error criteria based on the pixel footprint is explained. In section 2.3 we describe the iterative fusion of depth maps and triangulations to get the final set of oriented points.

2.1 Filtering of Depth Maps

Prior to the actual meshing process the single depth images are filtered in order to remove spurious depth values. Although multi-view approaches may result in rather clean point clouds not all erroneous elements can be detected based on the evaluation of geometric and/or radiometric consistency across multiple views. One major problem, for example, are regions of object borders in combination with low-textured background (see the red circle in 1). In these parts pixel consistency measures are not distinctive and depth estimation is mainly powered by smoothness constraints. As a result edges are over-matched what cannot be reliably detected by consistency constraints across the images. These erroneous depths are typically represented by small-sized pixel patches. In order to filter these elements we compute the connectivity of each pixel within its local neighbourhood. This measure will be referred to as support in the following text. Inspired by the path-wise accumulation process of SGM, for each pixel $\mathbf{x}(x, y)$ the support is computed by the evaluation of consistency with its proximate pixels along 16 image paths.

$$S(\mathbf{x}(x,y)) = \sum_{i=1}^{16} s_i(\mathbf{x}(x,y)) \tag{1}$$

The pixel-wise support values are computed evaluating subsequent pixels $\mathbf{x}_n$,$\mathbf{x}_{n+1}$ along a path i as

$$s_i(\mathbf{x}) = \sum_{n=0}^{\infty} T(\mathbf{x}_n, \mathbf{x}_{n+1}) \tag{2}$$

This contribution has been peer-reviewed.
doi:10.5194/isprsarchives-XL-3-281-2014

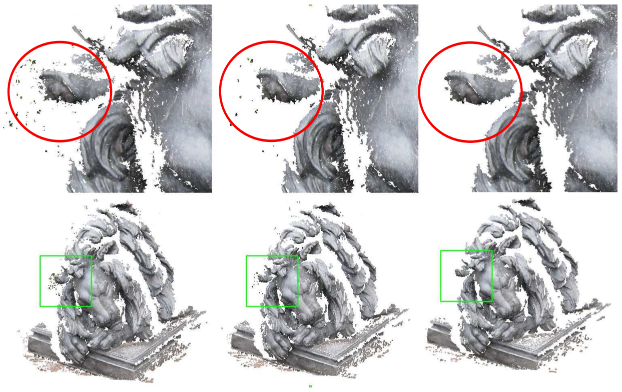


Figure 1: Point clouds corresponding to the filtered depth maps for different thresholds t. Figures in the upper row are detailed views of the area depicted by the green rectangles. Left: $t_s = 0$; Middle: $t_s = 21$; Right: $t_s = 41$

thereby n denotes the offset with respect to $\mathbf{x}$ along the path direction. $T()$ is a operator evaluating to 0 or 1 according to

$$T(\mathbf{x}_n, \mathbf{x}_{n+1}) = \begin{cases} 1 & \textit{for} \quad d(\mathbf{x}_n), d(\mathbf{x}_{n+1}) \quad \textit{valid} \\ 0 & \textit{else} \end{cases} \tag{3}$$

If the final support $S(\mathbf{x}(x,y))$ is below a certain value t_s its depth will not be considered for further processing. Figure 1 displays results for different thresholds t_s. The red circles mark areas of erroneous depths due to overmatching. For a threshold of $t_s = 41$ we got overall satisfying point filtering within our tests. We used this threshold for all of our experiments.

2.2 2.5D Mesh Extraction from Depth Maps using Restricted Quadtrees

Generally there are two different approaches for the RQT construction. The bottom-up strategy initially starts by defining nodes on the lowest (full resolution) level. Then, each node is evaluated based on quadtree and error criteria. The quadtree criterion assures that side lengths of proximate grid cells are the same, the half or the double. This property allows to obtain matching triangulations. The error condition, typically the approximation error describes to what extend geometric errors are introduced if the particular node is removed. If these two conditions are met, nodes are fused and higher level nodes are examined. The major draw back of this approach is that error conditions can not be guaranteed. Although conditions are fulfilled for local evaluations the errors accumulate over multiple levels.

This drawback can be overcome by a top-down construction strategy. Thereby initially the raster data is represented by a single node. Nodes are added recursively if error criteria are not fulfilled. However, compared to the bottom-up methods the implementation is more sophisticated because when adding a node on level l also updates on higher levels $l + n$ might be necessary to maintain the restricted quadtree structure. An efficient implementation is given in (Pajarola, 1998) which is based on dependency graphs. Each node depends on two other nodes from the same or the next upper level. Recursively, in a top-down manner, all nodes are checked for error criteria based on their two dependent nodes and activated if these are fulfilled. If a node is evaluated to be a part of the triangulation then its dependent nodes and their dependencies have to be activated too. From the set of finally activated nodes a matching triangulation can be extracted. For more detailed information of implementation the reader is referred to the original publication.

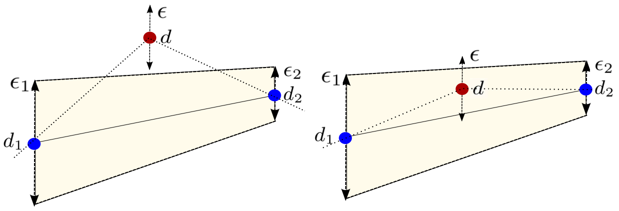


Figure 2: Visualization of the error criterion, point candidate red, dependent node points blue. Left: error criteria is not fulfilled, point candidate is inserted to triangulation. Right: error criteria is fulfilled, point candidate is not inserted to triangulation.

In this work the concept of RQT is applied for constructing meshes from depth maps generated by multi-view matching systems. The goal is to extract robust normals from these meshes on noisy depth estimation whilst preserving as much as possible geometric information. The RQT approach holds the error criteria for each node and gives us the option to extract normals in a controlled manner. For the meshing of elevation data presented in the original work approximation errors are computed in object space and added to the triangulation if larger than a constant threshold ϵ. We base our dynamic error criterion on the pixel footprint assuming that precision of depth maps linearly decreases with the depth (and therefore the GSD). Let n be a node with the depth $d(x,y)$ and its dependent nodes n_1, n_2 with $d_1(x,y)$ and $d_2(x,y)$ respectively. Each depth point is assumed to be measured with an uncertainty of $\epsilon = gsd * t_q$. The node n is inserted to the triangulation if it is not contained by the local noise band which is defined by $d_1 \pm \epsilon_1$ and $d_2 \pm \epsilon_2$ (figure 2). Hence, a depth value is considered to contribute to the geometry if

$$\begin{aligned} d - \epsilon &> \frac{(d_1 + d_2 + \epsilon_1 + \epsilon_2)}{2} \quad \textit{or} \\ d + \epsilon &< \frac{(d_1 + d_2 - \epsilon_1 - \epsilon_2)}{2} \end{aligned} \tag{4}$$

The pixel footprints are approximated by projecting the pixel pitch onto the fronto-parallel plane at depth d, d_1, d_2. Within the mesh generation step we discard triangles made up by vertices varying by more than six times the local pixel footprint in depth. Dependent of ϵ the number depth measurements can be reduced in a controlled way. For an example of the quadtree structure see figure 3. Note that vertices not representing actual structure but local noise can be removed.

2.3 Model Fusion

In this section we discuss how we fuse estimations of all available filtered maps to generate a non-redundant point set and the respective normals indicating the surface orientation. Our overall model $M(i)$ at the iteration step i is represented as a set of i sub-models S_i. Each sub model contains vertices and respective faces of one depth map D_i. To update the model by a depth map D_i we iteratively perform the following four steps:

1. Generate virtual vertex image V_i and virtual face image F_i by projecting vertices and faces of all sub models S_i for $i \in [0, i-1]$

2. Fusion of vertices contained in the depth map D_i and the model vertices stored in image V_i.

3. Invalidation of obsolete vertices and faces in the models S_i. Invalidation of obsolete depth estimates in D_i using the faces stored in F_i.

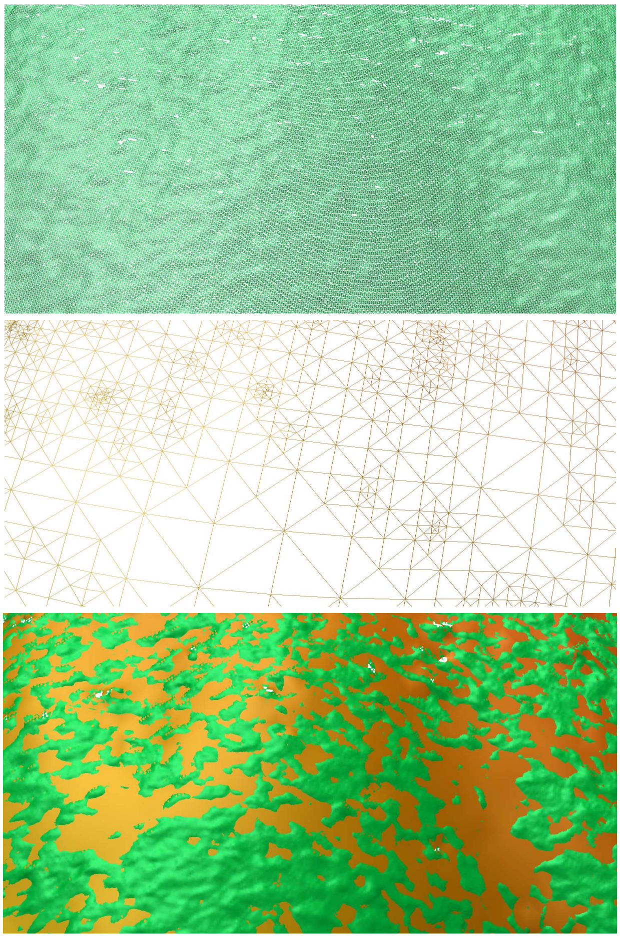


Figure 3: Meshes generated by quadtree approach in dependence of ϵ. Upper: Uncompressed triangulation for $\epsilon = 0$. Middle: RQT triangulation with $\epsilon = 1.2$. Lower: Overlay of shaded surfaces $\epsilon = 0$ (green) and $\epsilon = 1.2$ (orange)

4. RQT meshing of D_i and insertion to M

For the generation of V_i we skim all the models contained by M and project them to a virtual view corresponding to the depth image D_i. Thereby we use backface and frustum culling as well as z-buffering. If a vertex is successfully projected, its pointer is stored to V_i. This way pointers to all vertices observable from the ith frame are stored. With the same intention, all faces of M are projected to the virtual image F_i. This way a set of adjoining faces for each vertex of V_i is derived. Since each vertex can touch a maximum of nine faces F_i has to be designed as a three dimensional data structure. The virtual face image is later used to invalidate redundant or obsolete depth estimations in D_i.

Once the virtual views are generated, the depth estimations $d_i(x, y)$ in the current depth image D_i are fused with depths $d_m(x, y)$ implied by the model vertices $v_i(x, y)$ stored in V_i. If for the same coordinates (x, y) only one of the depths d_i or d_m is available no fusion is carried out. Otherwise the depths are checked for geometric consistency. We evaluate consistency by means of the local pixel footprints p_i of depth image pixel and p_m the footprint of the model depth. Depending on the factor t_c we consider the two candidates to be consistent if

$$|(d_i - d_m)| < (p_i + p_m)t_c. \tag{5}$$

In this case a counter expressing the reliability of an vertex is increased. If equation 5 is not fulfilled the visibility is verified. In the physically impossible case that the depth candidate d_i is located behind the model vertex, d_i is invalidated and the reliability of the vertex v_i is decremented. If the reliability is equal to zero the vertex is marked to be removed from the model. On the other hand, if d_i is in front of the model plane we keep the old model vertex and mark the new depth to be part of the new triangulation.

In case of geometric consistency we update the model differentiating two cases. Paying respect to scale differences we again evaluate the local footprints p_i an p_m. If the scale is comparable, the vertex coordinates are projected to the depth map and a depth is bilinearly interpolated. The model vertex coordinates are updated taking into account the number of detections and footprints. Given that scale differences are more significant the surface point providing the smallest resolution is kept and the other is invalidated.

In the course of the fusion process, all model vertices which should be removed as well as the values in the depth maps which should not be considered for triangulation are identified. Subsequently, model vertices are erased by scanning the vertex lists in all sub models S_i. Moreover, faces based on these obsolete vertices are removed. Invalidated depths in the current depth map D_i are not considered for triangulation. However, in order to also remove pixels corresponding to faces between the the invalidated depths, image areas covered by projected faces have to be invalidated. Therefore $F_i(x, y)$ at all invalidated depth $d_i(x)y$ are scanned and pixels covered by the triangles are marked invalid. Note that this is a rather conservative approach of adding new depths since image regions of up to nine adjoining triangles are invalidated per depth.

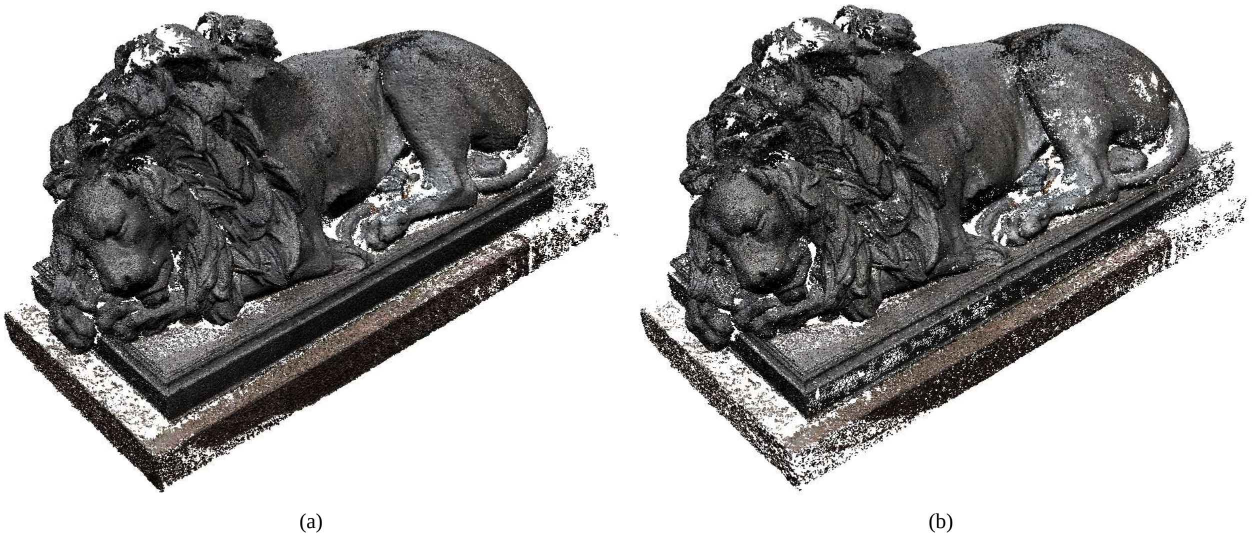
3. RESULTS

3.1 Close Range Dataset

We use the lion data set (publicly available at (Jancosek and Pajdla, 2008)) to evaluate the proposed algorithm regarding performance in presence of scale variances, advanced 3D structure and occlusions. Depth maps for the 56 images were generated using SURE (Rothermel et al., 2012). Then the resulting depth images were fused using two different thresholds $\epsilon = 1.0$ and $\epsilon = 1.2$ for the RQT construction. In our experiments we found these values to be a good trade off between data reduction and preservation of geometry. Moderate changes of ϵ effect point density significantly. The number of original depths estimations was reduced from 62.9 million to 11.7 and 5.7 million points respectively. As expected, the number of points for larger ϵ is smaller. However, as can be seen in figure 4, rather complete surfaces can be obtained. As shown in figure 6 for water tight Poisson reconstructions using an octree level of 11, the visual appearance is rather similar. The surface in large parts possesses the same noise levels and the similar amount of detail. Obtained reconstructions are not consistent in some parts, meaning triangles are overlapping or are located in front of each other. This is mainly due to the fusion of depths representing slanted surfaces. To avoid this, we plan to incorporate the available normal information in future. Figure 5 depicts the color coded normals of the resulting points. Except for few outliers, extracted normals are extracted reliably, which is particularly important for the subsequent extraction of water tight surfaces.

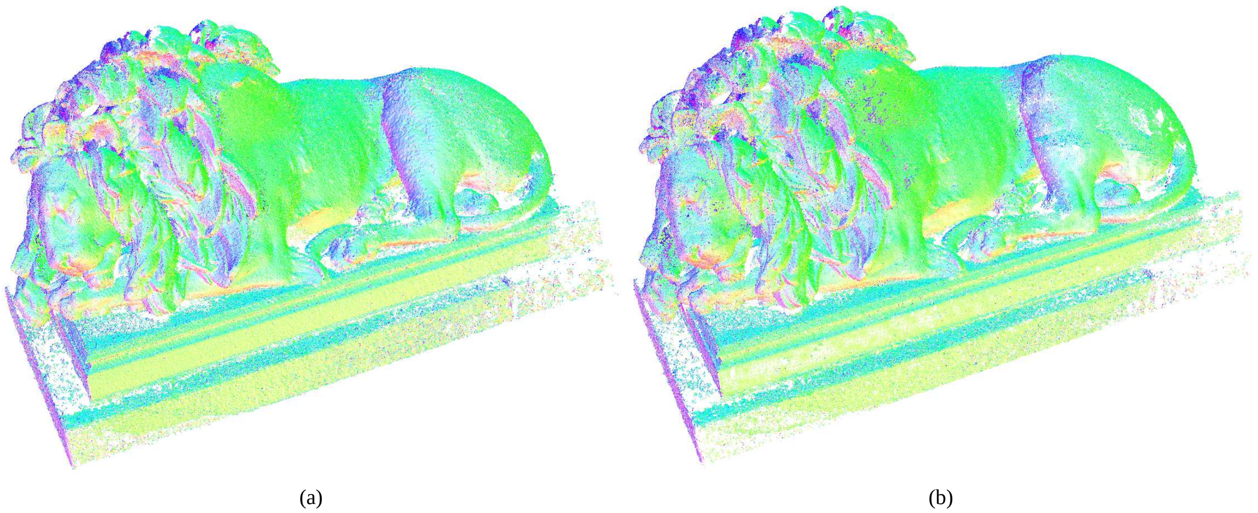
3.2 Oblique Airborne Dataset

Within a second test the algorithm was tested on 24 mid-format images collected using a RCD30 Oblique Penta camera. The average pixel footprint ranges from 6cm to 13cm. The frames



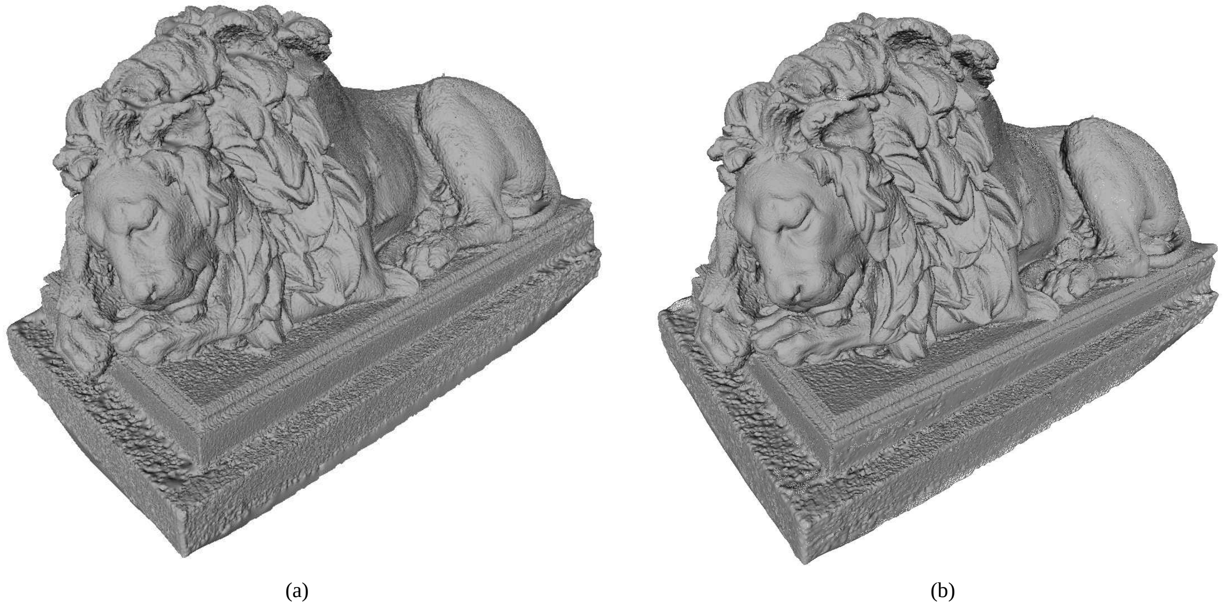
(a) (b)

Figure 4: Resulting oriented points of the lion data set. (a): RQT error threshold $\epsilon = 1.0$. (b): RQT error threshold $\epsilon = 1.2$



(a) (b)

Figure 5: Resulting color coded normals of the lion data set. (a): RQT error threshold $\epsilon = 1.0$. (b): RQT error threshold $\epsilon = 1.2$



(a) (b)

Figure 6: Poison reconstructions based on our oriented point sets. (a): RQT error threshold $\epsilon = 1.0$. (b): RQT error threshold $\epsilon = 1.2$

possess four different viewing directions and form a sub-set of the benchmark data for matching oblique aerial imagery presented in (Cavegn et al., 2014). The 3D structure is not as sophisticated as for the dataset discussed before. Nevertheless, redundancy is limited, particularly for the facades. Furthermore the amount of data is significant. The typical problem of processing oblique data, the variance in image scale and therefore inhomogeneous precisions throughout the point cloud, is well handled by our approach. Figure 7 depicts extracted mesh patches. As before, reconstructions are complete, but some faces are not consistent. Normals are reliably computed and are, along with the point coordinates, sufficient for Poisson reconstruction (figure 8). Thereby in contrast to typical 2.5D datasets 3D structures as balconies and streets below bridges are successfully reconstructed.

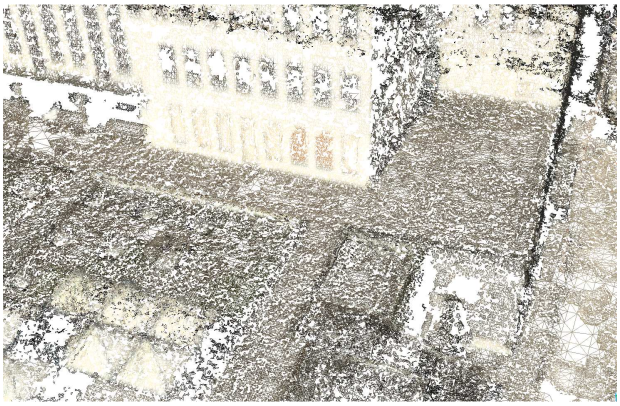


Figure 7: Mesh patches generated from oblique airborne imagery. RQT error threshold $\epsilon = 2.4$

4. CONCLUSION

In this article we presented an algorithm iteratively merging depth maps derived by DIM. Thereby the model state is represented as a set of oriented points and faces. The current model state is updated by either supplementing new surface patches by RQT triangulations of the depth maps or refining existing patches. If the pixel footprints of model entities and candidates from the depth maps largely vary, only the samples corresponding to the higher resolution are considered. Sampling distance of depth maps is dynamically controlled using a RQT. Moreover, visibility constraints are implemented aiming at removing outliers. We tested the approach on an aerial oblique and a close range dataset. For both, the number of original points were reduced by up to 90% percent. Normals were computed reliably and outliers were removed such that subsequent Poisson reconstructions gave satisfying results. For our test dataset increased sampling rates resulted in a vertex reduction of factor 2 while giving comparable results. This is because spatially proximate samples not contributing to the geometry are excluded from further processing by the RQT triangulation. However, the property of matching triangulations is not maintained.

So far normals only serve as input for the extraction of watertight meshes. In future work we plan to incorporate the surface normals into the fusion process to further improve precision, increase consistency of faces and eliminate depth estimations detected on slanted surfaces. Other issues we want to address is parallelization of codes and the capability of processing image tiles. This would allow for processing large frame imagery in a more efficient way enabling airborne applications at even larger scales.

REFERENCES

Cavegn, S., Haala, N., Nebiker, S., Rothermel, M. and Tutzauer, P., 2014. Benchmarking high density image matching for oblique airborne imagery. In: Photogrammetric Computer Vision (to appear).

Curless, B. and Levoy, M., 1996. A volumetric method for building complex models from range images. In: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques, ACM, pp. 303–312.

Fuhrmann, S. and Goesele, M., 2011. Fusion of depth maps with multiple scales. In: Proceedings of the 2011 SIGGRAPH Asia Conference, SA '11, ACM, New York, NY, USA, pp. 148:1–148:8.

Furukawa, Y. and Ponce, J., 2010. Accurate, dense, and robust multiview stereopsis. IEEE Transactions on Pattern Analysis and Machine Intelligence 32(8), pp. 1362–1376.

Haala, N. and Rothermel, M., 2012. Dense multi-stereo matching for high quality digital elevation models. PFG Photogrammetrie, Fernerkundung, Geoinformation 2012(4), pp. 331–343.

Hermann, S. and Klette, R., 2012. Evaluation of a new coarse-to-fine strategy for fast semi-global stereo matching. In: Proceedings of the 5th Pacific Rim conference on Advances in Image and Video Technology - Volume Part I, PSIVT'11, Springer-Verlag, Berlin, Heidelberg, pp. 395–406.

Hiep, V. H., Keriven, R., Labatut, P. and Pons, J.-P., 2009. Towards high-resolution large-scale multi-view stereo. In: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, IEEE, pp. 1430–1437.

Hirschmüller, H., 2008. Stereo processing by semiglobal matching and mutual information. IEEE Transactions on Pattern Analysis and Machine Intelligence 30, pp. 328–341.

Jancosek, M. and Pajdla, T., 2008. Cmpmvs - multi-view reconstruction software. In: Proceedings of the International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT).

Kazhdan, M. and Hoppe, H., 2013. Screened poisson surface reconstruction. ACM Trans. Graph. 32(3), pp. 29:1–29:13.

Lorensen, W. E. and Cline, H. E., 1987. Marching cubes: A high resolution 3d surface construction algorithm. In: Proceedings of the 14th annual conference on Computer graphics and interactive techniques, SIGGRAPH '87, ACM, New York, NY, USA, pp. 163–169.

Lowe, D. G., 1999. Object recognition from local scale-invariant features. In: Computer vision, 1999. The proceedings of the seventh IEEE international conference on, Vol. 2, Ieee, pp. 1150–1157.

Merrell, P., Akbarzadeh, A., Wang, L., Frahm, J.-M. and Nistr, R. Y. D., 2007. Real-time visibility-based fusion of depth maps. In: In Int. Conf. on Computer Vision and Pattern Recognition.

Pajarola, R., 1998. Large scale terrain visualization using the restricted quadtree triangulation. In: Visualization '98. Proceedings, pp. 19–26.

Pajarola, R., 2002. Overview of quadtree-based terrain triangulation and visualization.

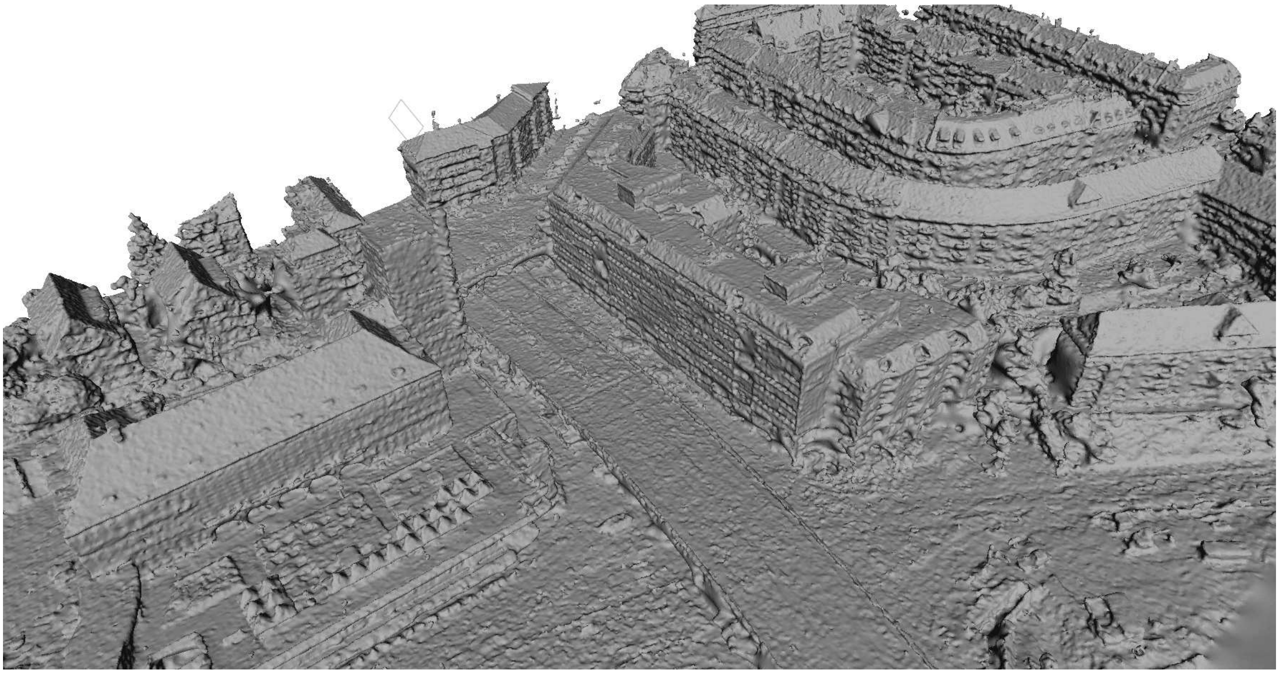


Figure 8: Poisson reconstruction based on our point sets. RQT error threshold $\epsilon = 2.4$

Pajarola, R., Antonijuan, M. and Lario, R., 2002. Quadtin: Quadtree based triangulated irregular networks. In: Proceedings of the conference on Visualization'02, IEEE Computer Society, pp. 395–402.

Rothermel, M., Bulatov, D., Haala, N. and Wenzel, K., 2014. Fast and robust generation of semantic urban terrain models from uav video streams. In: International Conference on Pattern Recognition, Stockholm, Sweden.

Rothermel, M., Wenzel, K., Fritsch, D. and Haala, N., 2012. Sure: Photogrammetric surface reconstruction from imagery. In: Proceedings LC3D Workshop, Berlin.

Seitz, S. M., Curless, B., Diebel, J., Scharstein, D. and Szeliski, R., 2006. A comparison and evaluation of multi-view stereo reconstruction algorithms.

Sivan, R. and Samet, H., 1992. Algorithms for constructing quadtree surface maps. In: Proc. 5th Int. Symposium on Spatial Data Handling, pp. 361–370.

Turk, G. and Levoy, M., 1994. Zippered polygon meshes from range images. In: Proceedings of the 21st annual conference on Computer graphics and interactive techniques, SIGGRAPH '94, ACM, New York, NY, USA, pp. 311–318.

Zach, C., 2008. Fast and high quality fusion of depth maps. In: Proceedings of the International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT), Vol. 1.

Zach, C., Pock, T. and Bischof, H., 2007. A globally optimal algorithm for robust tv-l 1 range image integration. In: Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on, IEEE, pp. 1–8.