A MEDIAN-BASED DEPTHMAP FUSION STRATEGY FOR THE GENERATION OF ORIENTED POINTS

M. Rothermel[a]*, N. Haala[b], D. Fritsch [b]

[a] nFrames GmbH, Stuttgart, Germany - mathias.rothermel@nframes.com
[b] Institute of Photogrammetry, Stuttgart University, Germany -
forename.lastname@ifp.uni-stuttgart.de

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

ABSTRACT:

Due to good scalability, systems for image-based dense surface reconstruction often employ stereo or multi-baseline stereo methods. These types of algorithms represent the scene by a set of depth or disparity maps which eventually have to be fused to extract a consistent, non-redundant surface representation. Generally the single depth observations across the maps possess variances in quality. Within the fusion process not only preservation of precision and detail but also density and robustness with respect to outliers are desirable. Being prune to outliers, in this article we propose a local median-based algorithm for the fusion of depth maps eventually representing the scene as a set of oriented points. Paying respect to scalability, points induced by each of the available depth maps are streamed to cubic tiles which then can be filtered in parallel. Arguing that the triangulation uncertainty is larger in the direction of image rays we define these rays as the main filter direction. Within an additional strategy we define the surface normals as the principle direction for median filtering/integration. The presented approach is straight-forward to implement since employing standard oc- and kd-tree structures enhanced by nearest neighbor queries optimized for cylindrical neighborhoods. We show that the presented method in combination with the MVS (Rothermel et al., 2012) produces surfaces comparable to the results of the Middlebury MVS benchmark and favorably compares to an state-of-the-art algorithm employing the Fountain dataset (Strecha et al., 2008). Moreover, we demonstrate its capability of depth map fusion for city scale reconstructions derived from large frame airborne imagery.

1. INTRODUCTION

3D surface reconstruction from large sets of overlapping imagery has been, and still is, a vivid research topic in photogrammetry and computer vision especially for complex 3D scenes. Driven by advances in digital camera technology and algorithms, limits of automatic image-based 3D data capture were pushed regarding precision, robustness, processing speed and scale. In this work we focus on the problem of depth map fusion for a wide range of applications, including datasets in the domain of large scale airborne mapping. Traditionally aerial imagery is captured in nadir viewing directions enabling reconstruction of 2.5-dimensional (2.5D) Digital Surface Models (DSM). Such DSMs provide detailed roof structures, however reconstructed geometry at facades is limited. While this is sufficient for applications aiming at LOD1 and LOD2 city representations, explicit geometric information on facade elements like doors and windows as well as other vertical objects is frequently required. Since such features are difficult to extract from nadir flights, oblique camera systems operated by unmanned aerial vehicles (UAV) or aircraft are becoming more and more important. Algorithms for depth map fusion derived from such imagery require to properly handle 3D geometry, scale well to the amount of collected data, offer precise reconstructions at high density and guarantee adequate run times.

A tremendous amount of research was conducted in the area of image-based surface reconstruction in the last three decades. For an excellent overview and probably the most popular benchmark of multi view stereo (MVS) systems see (Seitz et al., 2006). Several approaches represent the scene to be reconstructed in object space from an early stage. Typical representatives include level set methods (Pons et al., 2007) and mesh evolution algorithms (Hiep et al., 2009). Patch-based algorithms like (Furukawa and Ponce, 2010) start with high confident surface points and grow the surface utilizing geometric information of the points already reconstructed. Space carving (Kutulakos and Seitz, 1998) starts with a solid and iteratively carves the volumetric entities not being photo-consistent across the views. Scene representation based on 2.5D height fields deliver good results for the reconstruction of elevation data or DSMs, for example (Vogiatzis et al., 2008), (Bethmann and Luhmann, 2015). For these methods, shape priors in the course of matching can be conveniently formulated due to the regular structure of raster representations. In contrast to object space based methods, depth map based matching algorithms represent the surface information by a set of 2D distance or disparity maps. This type of matching methods is quite popular for large scale reconstructions since naturally dividing the problem of reconstruction into multiple subproblems. More precisely, geometry is reconstructed by matching one or a limited set of neighboring views to a reference view using stereo or multi-baseline stereo (e.g. (Okutomi and Kanade, 1993) (Merrell et al., 2007), (Goesele et al., 2007), (Pollefeys et al., 1998). These methods in general produce depth maps which hold redundant surface observations. In order to extract a consistent representation of the scene, depth maps have to be fused eventually.

The problem of depth map fusion is aggravated by the large number and complex nature of effects influencing the variances of observations across depth maps. These are e.g. geometric properties of camera network configuration like the number of images a point is seen in, the intersection angles of images rays, as well as errors from stereo matching like sub-pixel locking, fronto parallel effects or image blur as well as errors introduced by inaccuracies in bundle block adjustment. An early work and purely geometric algorithm for depth map fusion was proposed by Polygon Zippering (Turk and Levoy, 1994). The method generates depthmap-wise triangle meshes by constructing two faces from four adjacent depth estimates. After the alignment of meshes,

*Corresponding author

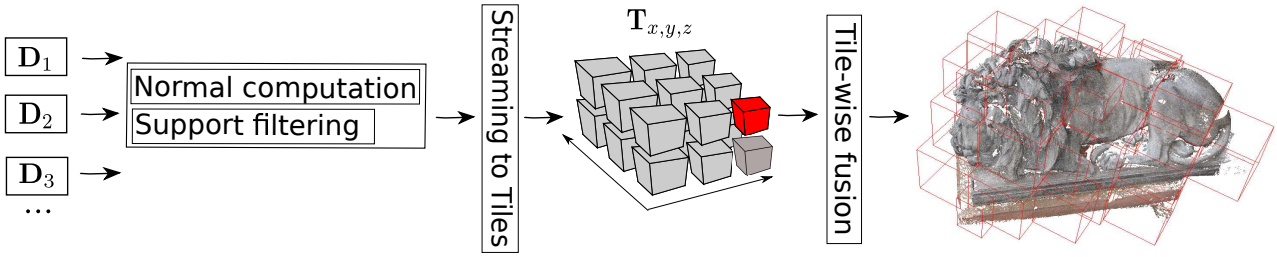

Figure 1: Flow chart of the proposed fusion algorithm. Depth images $\mathbf{D}_i$ are sequentially filtered and point-wise normals are computed. On completion of each image, points and corresponding normals and image ids are streamed to 3D cubes regularly partitioning object space. The single cubes are then subject to the fusion process.

redundant triangles are removed from the boundaries of single patches and remainders are connected. In (Merrell et al., 2007) an approach for real-time fusion of noisy depth maps was presented. Thereby, proximate depth maps are rendered into one reference view, redundant depths are pixel-wise checked for geometric consistency and are eventually filtered using occlusion and confidence checks. After averaging consistent depths a mesh is constructed in image space using quadtrees and lifted to 3D space. In contrast to the latter two local methods, global methods extract surfaces by minimizing a global energy functional typically forcing visibility constraints or smoothness whilst possibly representing input data to a high degree. Global methods tend to deliver more robust results, however, this comes at the cost of less scalability and higher computational costs. An example producing watertight surface meshes was proposed by (Kazhdan et al., 2006), (Kazhdan and Hoppe, 2013). The algorithm operates on oriented point sets and models the surface as an indicator function evaluating to 1 behind the surface and 0 in front of the surface. At the in-front / behind transition the gradient of the vector field is maximal. Oriented points, i.e. 3D coordinates along corresponding normals, are considered as samples of the indicator functions gradient and are used to construct a gradient vector field. The indicator function is given by the function minimizing the absolute difference between vector field and the indicator functions gradient. This then can be casted as a Poisson problem, which can be efficiently solved.

A large portion of depth map fusion methods build up on volumetric range integration (VRIP) (Curless and Levoy, 1996). Typically a signed distance field is computed on a (multi-level) octree structure by the projection of depth estimations from which then a triangulation can be derived, for example using the Marching Cube algorithm (Lorensen and Cline, 1987). Using the same base concept, (Zach et al., 2007) reconstruct a signed distance field in voxel space. Then a surface is extracted by minimizing a global energy functional based on TV-L1 regularization, claiming smoothness and small differences to the zero level set. Employing the L1 norm yields favorably results in the presence of outliers. However, 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. Vertices are sorted to the octree structure 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. Similarly, (Kuhn et al., 2014) proposed a method employing variable voxel sizes defined by observation-wise precision estimates. These precision measures are computed for each disparity based on TV in the disparity maps. The local TV is associated with an error class defining quality which is previously learned using ground truth.

In this work we present a local method for depth map fusion. Similar to (Fuhrmann and Goesele, 2011) we first assign points to an multi-level octree structure according to the pixel footprint. Subsequent filter operations are then carried out on points of the locally lowest tree-levels, whereas high level points are discarded. This way not only local surface sampling is adapted, but also most precise samples can be identified. Moreover, for each point the id of the reference image (Rothermel et al., 2012) and its normal is stored. The latter is derived in image space employing an adaptive, discontinuity preserving triangulation on each depth map based on a RQT (Pajarola, 1998). Due to the triangulating character of MVS, precisions along image rays are expected to be smaller than precisions perpendicular to the rays. When filtering single points, we acknowledge this fact by only incorporating points located in a cylinder oriented in direction of the largest uncertainty, e.g the line of sight. Within a second strategy we filter along the point-wise normals. Approaches employing the L1 norm yield excellent results since outliers possess small influence on integration process. Our method is based on the idea to further restrict the impact of outliers by employing median filtering. The methodology is discussed in detail in section 2. While we mainly use close range data sets to evaluate our concept, additional results in section 3 demonstrate the feasibility of our approach for 3D reconstructions in complex urban areas from oblique aerial images as the aspired main area of application.

2. METHODOLOGY

2.1 Pipeline Overview

In this section we discuss the single modules of the proposed fusion algorithm. The flow chart of the complete workflow is depicted in figure 1. The input is a set of oriented depth images $\mathbf{D}_i$, for which the respective camera parameters, i.e. exterior and interior orientation are known. Therefore, the point coordinates corresponding to a depth $d(x, y)$ can be derived using perspective geometry. Within the first step each depth map is filtered to remove spurious depth estimates. This filtering is based on the number of successfully reconstructed depths in a local neighborhood of each depth observation. The underling assumption is that areas which are successfully reconstructed for large patches are more reliable than patches possessing dimensions of a few pixels only. Then, for each depth in each of the depth maps a normal is computed based on an RQT triangulation adapting to local geometry and noise levels. A more detailed explanation of filtering and normal computation can be found in (Rothermel et al., 2014b). To guarantee scalability, we divide the fusion problem into several sub-problems. More precisely, we spatially partition object space by regular cubes which are then processed independently. Depth map filtering and normal computation are sequentially performed on each depth image. On completion of processing an image, object coordinates for each depth $d(x, y)$ are computed

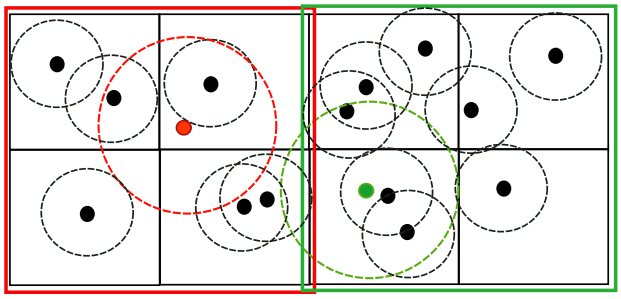

Figure 2: Visualization for the criterion assigning points to the octree. Circles mark the point-wise GSD, points represent the coordinates. Points are assigned to the box they are located in and the side lengths of the box are comparable to the GSD.

and streamed, along its normals and image ID, to its corresponding cube. The points contained by each cube are then subject to a fusion/filter algorithm. Thereby, within a first step point-wise pixel footprints and the vector defined by the ray connecting point and camera center are computed. Then the actual filtering is performed. This involves identification of the points representing the surface with the best precision and subsequent median filtering along the point-wise lines of sight or normals (section 2.2). Beside scalability, cube-wise processing offers the possibility of convenient high-level parallelization.

2.2 Median-based Fusion

Motivated by the results obtained by the median-based fusion algorithms for 2.5 surface representations, as for example implemented in (Hirschmüller, 2008), (Rothermel et al., 2014a), we propose a median-based fusion strategy for 3D scenes. The general idea is to first extract the set of surface points $\mathcal{P}$ mapping the object with the highest resolution. This not only recovers the smallest sampling rates of images observing the surface, but also, if image scale is assumed to define precision, recovers the most accurate points. Then, each point $\mathbf{p} \in \mathcal{P}$ is incorporated into the process of median filtering along the vector defined by its translation to the camera center. The motivation for filtering along these rays is based on the fact that precision along the line of sight and perpendicular to these lines are not identical due to properties of forward intersection which in general delivers lower accuracies in viewing directions. The aforementioned algorithms for 2.5D DSM integration define the main filter direction by the normal of the predominant plane fitting the observed surface. Obviously such a plane does not exist for 3D scenes. Therefore, within a second approach we define the main filter direction by the point-wise normals. Since its minor importance to the algorithmic discussion, in the following section we denote both, lines of sight as well as normals, by $\mathbf{n}$ and refer to them as *point-wise filter direction* (PFD).

Derivation of the Most Precise Point Set Representing the Surface In this section we explain how we extract the point set $\mathcal{P}$ representing the most precise surface samples assuming the pixel footprint represents reconstruction quality. Therefore we use a multi-scale octree structure into which all points of a single tile are inserted. Point coordinates and the point-wise pixel footprint define the octree cell an observation is located in. Points located on the lowest level (leaf nodes) of the octree then indicate the most precise surface sample.

Octrees are data structures separating space by regular boxes. Each of the cubes contains 8 daughter cubes regularly subdividing its mother cube. Points to be inserted are assigned to cubes (nodes) if certain criteria are fulfilled. For example, a

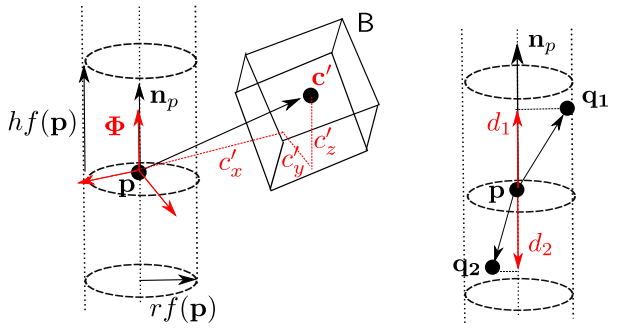

Figure 3: Left: Evaluation of the criterion describing if an octree box contains any points inside a specified cylinder $\mathbf{p}$, $\mathbf{n}_p$, $hf(\mathbf{p})$, $rf(\mathbf{p})$. Conditions are based on the coordinates c'_x, c'_y, c'_z of box center $\mathbf{c}'$ defined w.r.t. the coordinate system Φ. Right: Median filtering along the main filter direction $\mathbf{n}_p$. Translations induced by candidates $\mathbf{q} \in \mathcal{Q}$ are given by their projection on the cylinder axis $\mathbf{n}_p$.

point is assigned to the smallest box fully containing a sphere around the respective point coordinates. Hence, the octree structure implements an ordering scheme of points providing queries for point location as well as neighborhood queries. Our implementation follows the algorithms proposed by (Gargantini, 1982), (Press et al., 2007). For tree traversal, the link between mother nodes and daughter nodes have to be provided. Instead of a more conventional implementation where links between mother nodes and daughter nodes are realized by doubly linked pointers we prefer a implementation based on hash maps, also called Linear Octrees. The main advantage is that the usage of pointers (each 64bit on common hardware) storing mother-daughter and daughter-mother relationships is avoided. This reduces memory requirements, which supports the processing of single tiles in parallel and therefore speed up the integration process.

To identify the point set $\mathcal{P}$ we sort all points $\mathbf{t} \in \mathcal{T}$ contained in a single tile into the octree. Let B be an octree box with the side length s. As visualized in figure 2, a point $\mathbf{t} = [t_x, t_y, t_z]$ with the footprint $f(\mathbf{t})$ is located in B if the point is inside the cell and B is the smallest box satisfying

$$s > t_o(\mathbf{t}) \tag{1}$$

with

$$t_o(\mathbf{t}) = \alpha \frac{f(\mathbf{t}) + \beta \sum_{\mathbf{t} \in \mathcal{T}} f(\mathbf{t})}{1 + \beta}. \tag{2}$$

To be able to extenuate the influence of single footprints $\mathbf{t}_o(\mathbf{t})$ is composed of the local footprint and the average pixel footprints in the tile or the whole data set. For large β a uniform sampling can be derived neglecting scale variances across single observations completely. The parameter α controls the sampling density of the surface, therefore using a large valued α the surface is undersampled and for a small valued α oversampling is enforced which might be desirable for high redundant datasets.

After sorting all points of a tile to the octree the initial point set $\mathcal{P}$ is derived by identifying all leaf nodes. Per leaf node one point is generated by averaging coordinates and main filter directions of all contained points. Points from higher octree levels are discarded. Note that within these two steps the number of points is significantly reduced, typical by a factor of 5 to 10.

Median filtering - First Iteration The point set $\mathcal{P}$ comprises points which are reconstructed possessing the smallest pixel footprint within a local neighborhood. However, errors resultant from

registration and propagated from dense stereo, as well as properties of ray intersection angles are not modeled by the pixel-wise footprints. These errors cause the extracted points $\mathcal{P}$ being noisy, hence we median filter the point set $\mathcal{P}$ along the PFD. Thereby, all (in most cases more than one) samples stored in the all leaf nodes are incorporated by the filtering process. Note that this set of leaf node points in general is much larger than $\mathcal{P}$. Within a first step for each $\mathbf{p} \in \mathcal{P}$ a set of neighboring points is derived from the octree. Paying respect to the lager uncertainty in main filter direction as well as to outliers, the neighbors are defined by the set of points $\mathcal{Q}$ located in a cylinder with its central axis given by $\mathbf{p}$ and its PFD ($\mathbf{n}_p$). The cylinder radius and height are dependent of the footprint and specified by $rf(\mathbf{p})$ and $hf(\mathbf{p})$ receptively (see figure 3 (left)). To limit smoothing and artifacts at tile borders we typically choose a rather small radius $r = 1.4$. The tube height in our experiments is set to $h = 15$.

Identification the point set $\mathcal{Q}$ involves nearest neighbor queries on the octree structure. Starting at the mother node, each octree cube is checked if itself or any daughters might contain leaf node points located in the cylinder of question. This query is checked frequently and thus has to be designed carefully. Let B be a candidate octree box with the center $\mathbf{c}$ and the side length s. Moreover let $\mathbf{p}$, $\mathbf{n}_p$, r, h define the cylinder (see figure3 (left)). We construct a Cartesian coordinate system Φ with the origin in $\mathbf{p}$ and the z-axis pointing in direction of $\mathbf{n}_p = [n_x, n_y, n_z]$. The axes of the coordinate system are defined by the columns of the rotation matrix

$$\mathbf{R} = \begin{bmatrix} \mathbf{r}_1^\top \\ \mathbf{r}_2^\top \\ \mathbf{r}_3^\top \end{bmatrix} = \begin{bmatrix} 1 & 1 & -\frac{n_x+n_y}{nz} \\ & (\mathbf{r}_1 \times \mathbf{r}_3)^\top & \\ n_x & n_y & n_z \end{bmatrix} \tag{3}$$

The box center can then be transferred into the coordinates system Φ by

$$\mathbf{c}' = \mathbf{R}(\mathbf{c} - \mathbf{p}). \tag{4}$$

The octree box B may contain points located in the cylinder if

$$\sqrt{(c'_x)^2 + (c'_y)^2} < r + \sqrt{3}s \tag{5}$$

and

$$c'_z < h. \tag{6}$$

The term $\sqrt{3}s$ in equation (5) represents the radius of a sphere enclosing B. If conditions (5) and (6) are not fulfilled traversal of daughter nodes is terminated. If the conditions are fulfilled and additionally the examined box B is a leaf node all points contained by B are a subset of $\mathcal{Q}$.

Once the set of neighbors $\mathcal{Q}$ is identified, all $\mathbf{q} \in \mathcal{Q}$ are checked to be located in the specified cylinder. This is done following equations (4)-(6) by exchanging roles of box centers and points. If not located in the tube, the sample is removed from $\mathcal{Q}$. Additionally for each $\mathbf{q} \in \mathcal{Q}$ the angle between its normal $\mathbf{n}_q$ and $\mathbf{n}_p$ is computed. If this angle is larger than $60°$ the sample is removed from $\mathcal{Q}$ and discarded for further filtering. This way the incorporation of points not representing the same surface is avoided. After removal of suspicious samples the actual filtering is performed. The basic idea of the implemented median filtering is to translate the coordinates of $\mathbf{p}$ along its PFD $\mathbf{n}_p$. The translation is given by the median of translations induced by $\mathbf{q}_i$. More precisely, a translation of a sample $\mathbf{q}_i$ with respect to Φ is given by its projection onto $\mathbf{n}_p$

$$d_i(\mathbf{q}_i) = [\mathbf{q}_i - \mathbf{p}]^\top \mathbf{n}_p, \tag{7}$$

see figure 3 (right). Then the updated coordinates $\mathbf{p}'$ are computed as

$$\mathbf{p}' = \mathbf{p} + \mathbf{n}_p \, median[d_i(\mathbf{q}_i)]. \tag{8}$$

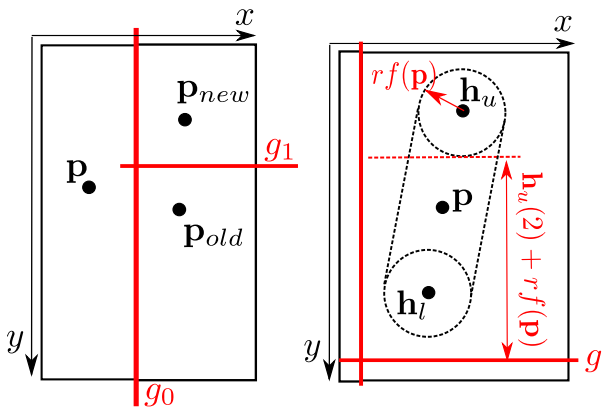

Figure 4: Left: Partitioning scheme of a KD-tree. Each node contains one point, nodes are divided by hyperplanes g_i. Direction of planes are incremented over the dimension D of space $\mathbb{R}^n$ (in this example $\mathbb{R}^2$). Right: Evaluation of the criterion describing if a KD-tree node fully contains a given cylinder. Therefore the D-th component of the coordinates $\mathbf{h}_u(D)+rf(\mathbf{p})$, $\mathbf{h}_l(D)+rf(\mathbf{p})$ have to be completely located on one side of the plane g.

Median Filtering - Additional Iterations The median-based integration described before enforces the set $\mathcal{P}$ to converge to the median surface. However, for noisy data sets multiple iterations might further improve the final surface. We found two additional iterations are a good trade-off between processing time and improved surface quality. Recall that within the first iteration all points contained in all leaf nodes of the octree were considered for integration. To speed up further iterations we restrict filtering on points $\mathbf{p} \in \mathcal{P}$ solely. As before, $\mathcal{P}$ has to be sorted to a 3D data structure enabling cylinder-based neighborhood queries. An octree as presented in the last section would be suitable for this task. However, we found that for our data sets these queries can be processed faster using KD-trees.

KD-trees are structures partitioning k-dimensional points based on half spaces. Since in this work we are interested in 3D entities, we restrict the following discussion to 3D space. Initially the first two points to be inserted are divided by a plane perpendicular to the axis of the first dimension (x-axis). These spaces define the initial nodes of the tree. For a new point $\mathbf{p}_{new}$ to be inserted the leaf node in which $\mathbf{p}_{new}$ is located in along with the point $\mathbf{p}_{old}$ already contained by the node are identified. The node then is further split by a plane separating $\mathbf{p}_{new}$ and $\mathbf{p}_{old}$ leading to two new leaf nodes, see figure 4 (left). The orientation of the plane is defined orthogonally to the second dimension. Subsequent insertions are performed in a similar way: first the leaf node, the contained point and the dimension d defining the direction of its last separation are identified. Then, the node is divided by a plane possessing a normal in direction of the incremented dimension ($(d+1) \mod 3$), resulting in two new leaf nodes. For implementation details the reader is referred to the algorithm proposed in (Press et al., 2007).

In order to extract the point set $\mathcal{Q}$ containing the points located in a cylinder defined by the point $\mathbf{p}$, its axis $\mathbf{n}_p$, its radius $rf(\mathbf{p})$ and height $hf(\mathbf{p})$ efficient neighborhood queries have to be provided. To identify the node whose daughters hold all points in the tube we identify the node fully containing the tube. Therefore we construct two points $\mathbf{h}_l = \mathbf{p} - \frac{h}{2}\mathbf{n}_p$ and $\mathbf{h}_u = \mathbf{p}_l + \frac{h}{2}\mathbf{n}_p$, see figure 4 (right). Starting at the root of the tree for each node

it is evaluated if the two spheres defined by $\mathbf{h}_l$ and $\mathbf{h}_l$ and the radius $rf(\mathbf{p})$ are fully contained by one of the daughter nodes. Let g be the position of the plane separating the daughter nodes in dimension D. Then the conditions are given by

$$\mathbf{h}_{l,u}(D) + rf(\mathbf{p}) < g \tag{9}$$

and

$$\mathbf{h}_{l,u}(D) - rf(\mathbf{p}) > g. \tag{10}$$

If a node fulfills both conditions (9),(10) and its daughter nodes do not, the node is guaranteed to be the smallest node holding all points located in the cylinder. Once this node is identified, all points contained by the daughters are skimmed and those located in the cylinder define the set $\mathcal{Q}$. Analogously to filtering in the first iteration, all inconsistent samples $\mathbf{q}_i$ possessing PFDs largely differing from $\mathbf{n}_p$ are identified and removed from $\mathcal{Q}$. The remainder is median filtered along $\mathbf{n}_p$.

3. RESULTS

3.1 Fountain Dataset

The first evaluation is carried out on the Fountain dataset (Strecha et al., 2008) for which a ground truth mesh based on LiDAR data is available. Triangles of this mesh were reprojected to the center view to generate a ground truth depth map. Using (Rothermel et al., 2012), an initial point set was obtained by matching each view to its four closest neighbors. These points then were subject to the integration algorithm employing the two filter strategies (normal and line-of-sight filtering). Thereby, the parameters of cylinder radii for oc- and kd-trees were set to our standard parameters $r = 1.4$ and $h = 15$. For comparison, points were also integrated by an state-of-the art algorithm proposed in (Fuhrmann and Goesele, 2011). For evaluation purposes a subpart of the generated points was selected and the depths were compared to the ground truth depth map. Whereas figure 5 displays statistics of residuals, figure 6 displays the subpart of generated points which were meshed using Delaunay triangulation and the color-coded residuals. From the meshes it can be observed that line of sight filtering delivers surfaces possessing slightly less outliers. This is due to the fact that normals, and therefore the direction of cylinders, in areas of limited density might be erroneous. An improvement of sub-pixel accuracy can not be observed. Comparison of the standard deviations of residuals and σ_3 filtered residuals of our method and (Fuhrmann and Goesele, 2011) clarifies that the median-based approach produces clearly less outliers. Again, results of σ_3 filtered residuals are rather similar. However, specifying the multiplier determining the footprint-dependent sampling rate to four (FG-s4, figure6c), the number of outliers is decreased, but so is the point density and therefore the preservation of details. The density in figure 5 denotes the size of the area induced by the filtered, projected points. Our approach delivers lower densities, however a loss of detail in the meshes can not be observed. Table 1 lists the processing times of the single algorithms and shows that the presented approach clearly out-performs the comparison method.

Method	ours-los	ours-n	FG-s4	FG-s2	FG-s1
time (min)	1.8	1.6	36	92	519

Table 1: Processing times for filtering the complete Fountain dataset of our algorithm and the method proposed in (Fuhrmann and Goesele, 2011).

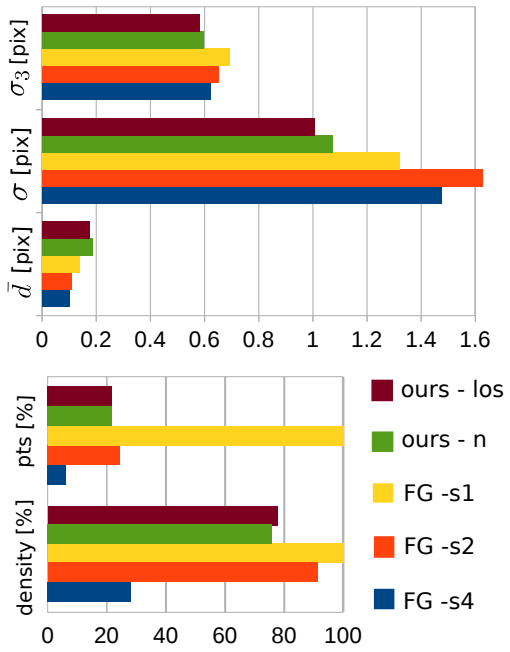

Figure 5: Mean differences, standard deviations and density analysis for the sub-area of the fountain dataset displayed in figure 6. Comparison of our algorithms with main filter directions along lines of sight (los) and normals (n) and (Fuhrmann and Goesele, 2011) using different GSD multipliers (1,2,4).

3.2 Middlebury Dataset

The second evaluation is carried out on the Middlebury MVS benchmark data set (Seitz et al., 2006). Therefore we generated depth maps for the temple and dino datasets in 'full' image configurations. Again, we used the aforementioned SGM-based MVS to reconstruct depth images. Thereby, each image is treated as reference image which is stereo matched against the 8 closest neighboring views. By multi-view forward intersection and constraints on geometric consistency redundant observations across the set of disparity maps are refined resulting in one depth map per view. Since the evaluation is performed on triangle meshes, we apply filtering along normals to extract oriented points and subsequently carry out a Poisson reconstruction (Kazhdan et al., 2006). In order to extract a surface close to the oriented points generated by our approach an octree depth similar to our octree is specified and no constraints regarding the minimal number of points per octree cell is applied. Figure 8 depicts the results for the two data sets. Median filtering was carried out as described in section 2.2 using 2 iterations. Since depth observations of the dino dataset turned out to be more noisy, the parameters for the tube radius was set to $r = 4$ instead of $r = 1.4$ for the temple dataset. The dynamic sampling width for both data sets were set to $\alpha = 2$ and $\beta = 0$ (see equation (2)). The first column in figure 8 shows the points contained by all refined depth maps. The second column depicts the oriented point set derived by the proposed algorithm. As can be seen the gross of outliers are removed and small details are preserved: for the temple data set 99.5% of observations possess an average deviation of 0.55mm to the ground truth and 0.47mm for the dino data set respectively (figure 7). The actual pixel footprints are in the range of 0.1-0.4mm. Dependent on accuracy and completeness levels given on the Middlebury evaluation page, the algorithm ranks in the range of 13 to 1 (with several others).The run times for the fusion process on a single core clocked at 3.3 GHz amounted 171 seconds for the Dino and 78 seconds for the temple data set.

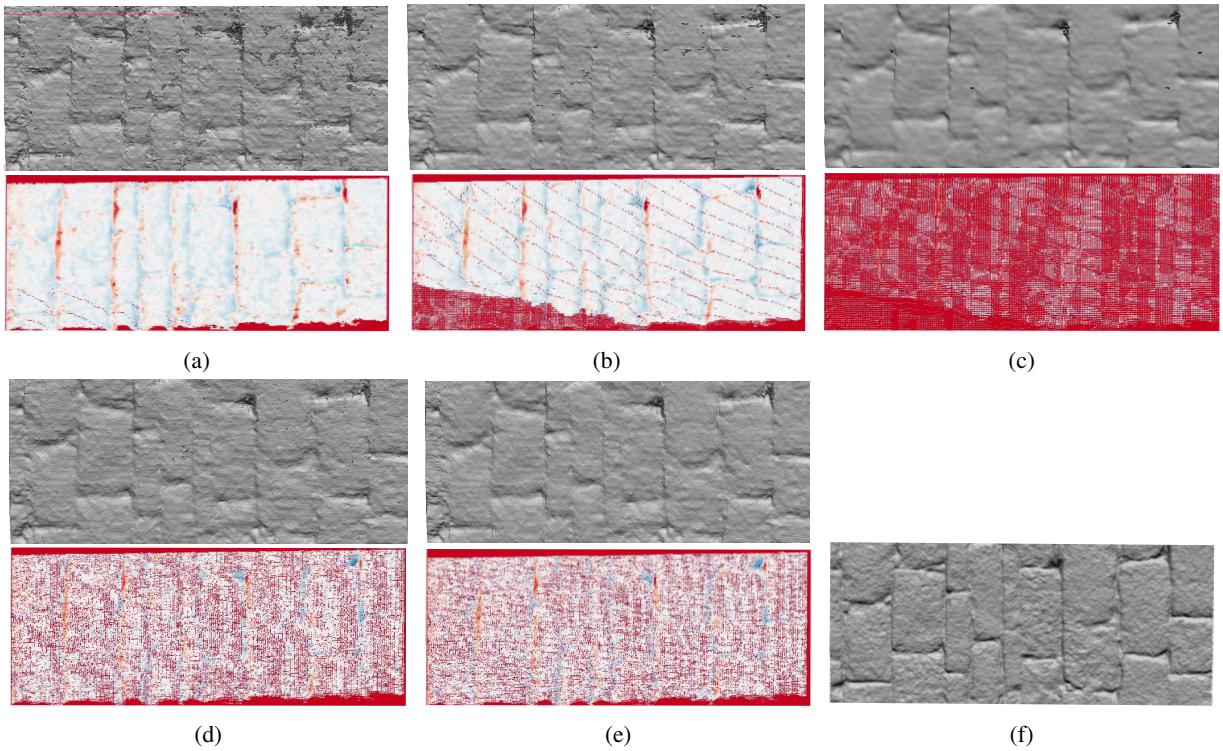

Figure 6: Meshes derived from filtered point clouds and differences of reprojected points to ground truth mesh (dark blue / dark red correspond to +5 / -5 GSD). (a)-(c): Results derived by (Fuhrmann and Goesele, 2011) employing gsd-wise multipliers of 1, 2 and 4. (d) and (e): our approach with filtering along normals and in direction of line of sight. (f): ground truth.

Figure 8: Results for the temple and dino data sets. From Left to right: raw point cloud; oriented points resulting from our method; mesh generated by Poisson reconstruction using our oriented points; ground-truth mesh.

Sort By	Temple Full 312 views Acc [mm]	Comp [%]	Temple Ring 47 views Acc [mm]	Comp [%]	Temple Sparse 16 views Acc [mm]	Comp [%]	Dino Full 363 views Acc [mm]	Comp [%]	Dino Ring 48 views Acc [mm]	Comp [%]	Dino Sparse 16 views Acc [mm]	Comp [%]
Guillemaut	0.54	99.0	1.00	97.6	1.11	96.2	0.46	100	0.77	99.5	0.89	98.0
Furukawa 2	0.68	99.3	0.69	99.1	0.76	99.2	0.41	99.9	0.46	99.6	0.56	99.2
Semerjian	0.75	97.8	4.14	58.7			0.46	99.9	0.53	99.6		
Galliani	0.48	99.2	0.6	99.1	0.71	97.0	0.39	99.9	0.39	99.4	0.55	98.6
Furukawa 3	0.61	99.6	0.6	99.6	0.79	99.3	0.43	99.8	0.38	99.8	0.52	99.2
Savinov	0.53	99.7	0.63	99.5	0.89	97.8	0.34	99.8	0.33	99.9	0.44	99.7
ECCV2016_104	0.53	99.6	0.62	99.5	0.71	98.1	0.55	99.8	0.61	99.7	0.62	98.0
Schroers	0.89	99.1	0.88	96.4	3.3	62.9	0.44	99.7	0.43	99.7	0.71	98.6
Habbecke	0.81	98.0					0.56	99.7				
Rothermel	0.55	99.4					0.47	99.7				

Figure 7: Results for the Middlebury benchmark using the dino dataset.

3.3 Aerial Dataset

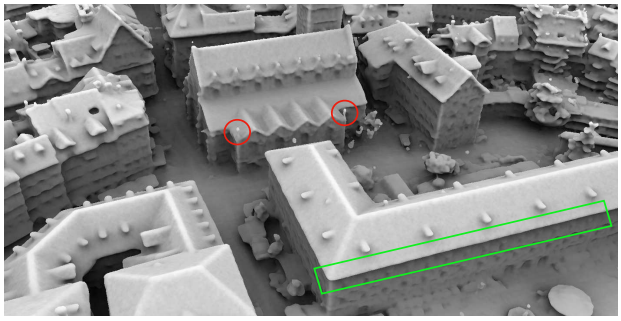

Figure 9: Mesh representation of the urban scenario. Clear edges (green rectangle) and small details as poles on the church roof (red circles) are successfully reconstructed.

Within a second test, the capability of processing large scale scenes is shown. Therefore we generated depth maps of a urban scenario using aforementioned MVS. The imagery was captured with an medium format Leica RCD30 Penta oblique system. Nadir imagery possesses an overlap of 80% in-strip and 75% cross-strip. The oblique cameras were mounted with an angle of $35°$. The size of the average pixel footprint is 6.6 cm across the whole data set. After filtering and normal computation in image space single depths are converted to points and streamed into 40 meters cubes subdividing object space. The points then are tile-wise filtered by sub-sampling with $\alpha = 2$ and $\beta = 0$ using tube radius of two GSD. Since no ground truth is available for the data set this test is of more qualitative nature, more precisely, preservation of details, artifacts at tile borders and correctness of topology are evaluated visually. For visual inspection, a mesh was computed using a Poisson reconstruction based on oriented points generated by our approach (see figure 9). As can be seen edges are extracted clearly (green rectangle) and details like the poles are reconstructed successfully. Virtually no outliers are contained in the reconstructed surface and narrow alleys are topological correct. The depicted area is computed from 25 tiles. Albeit some artifacts are introduced at tile borders of oriented points, these are not visible in the mesh representation due to smoothing in the course of mesh extraction. Figure 10 depicts a reconstruction of a larger part of the urban data set.

4. CONCLUSIONS AND FUTURE WORK

In this article we presented a median-based approach for the fusion of depth maps. Point coordinates are sub-sampled using a multi-level octree. By sorting points to the tree, an initial point set favoring observations possessing the smallest pixel footprint is identified. Iterative median filtering of points in a cylinder along point-wise lines of sight or normals lead to an improved set of oriented points. Within our tests concerning the filter direction, it was shown that filtering along lines of sight yields surfaces with less outliers compared to filtering along point normals. This is due do the fact that normal computation might fail in areas of limited density. However, accuracy of the derived surfaces was comparable. Moreover, compared to results derived by a state-of-the-art algorithm, our method produces less outliers, similar accuracy and superior processing times. Evaluation on the Middlebury MVS benchmark showed that the implemented algorithm in combination with (Rothermel et al., 2012) and (Kazhdan et al., 2006) performs well on completeness and precision: for the temple data set 99.5% of observations possess an average deviation of 0.55mm to the ground truth and an average deviation of 0.47mm for the dino data set, respectively. We showed that by spatially subdividing point clouds using the proposed tiling scheme the method scales well and makes the algorithm suited for reconstruction of complex urban scenes from high resolution airborne imagery. Although specifying the main filter direction along the surface normal and incorporating only limited number of neighboring points for filtering, artifacts are visible at some parts of the tile borders. Albeit being eliminated in the course of integration within the subsequent Poisson reconstruction, this problem can be eliminated by extension of single cubes using a small apron. Furthermore, a loss of detail is observed and bulges at edges are introduced during meshing. Since the oriented points are of good quality, we plan to use an algorithm for mesh extraction not incorporating further optimization causing additional smoothing. Moreover, re-meshing or mesh connection at tile borders has to be investigated to fully enable the generation of consistent large-scale surface meshes.

REFERENCES

Bethmann, F. and Luhmann, T., 2015. Semi-global matching in object space. The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences 40(3), pp. 23.

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, SIGGRAPH '96, ACM, New Orleans, LA, USA, 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.

Gargantini, I., 1982. An effective way to represent quadtrees. Commun. ACM 25(12), pp. 905–910.

Goesele, M., Snavely, N., Curless, B., Hoppe, H. and Seitz, S. M., 2007. Multi-view stereo for community photo collections. In: 11th International Conference on Computer Vision, 2007. ICCV 2007., IEEE, pp. 1–8.

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, pp. 1430–1437.

Hirschmüller, H., 2008. Stereo processing by semiglobal matching and mutual information. Pattern Analysis and Machine Intelligence, IEEE Transactions on 30(2), pp. 328–341.

Figure 10: Larger part of the mesh generated from the airborne oblique data set. Visible tile overlaps are due to shading and not to variances in geometry.

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

Kazhdan, M., Bolitho, M. and Hoppe, H., 2006. Poisson surface reconstruction. In: Proceedings of the Fourth Eurographics Symposium on Geometry Processing, SGP '06, Eurographics Association, Aire-la-Ville, Switzerland, pp. 61–70.

Kuhn, A., Mayer, H., Hirschmüller, H. and Scharstein, D., 2014. A tv prior for high-quality local multi-view stereo reconstruction. In: 3D Vision (3DV), 2014 2nd International Conference on, Vol. 1, IEEE, pp. 65–72.

Kutulakos, K. N. and Seitz, S. M., 1998. What do photographs tell us about 3d shape? Technical report, Computer Science Dept., U. Rochester.

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.

Merrell, P., Akbarzadeh, A., Wang, L., Mordohai, P., Frahm, J.-M., Yang, R., Nistér, D. and Pollefeys, M., 2007. Real-time visibility-based fusion of depth maps. In: 11th International Conference on Computer Vision, 2007. ICCV 2007., IEEE, pp. 1–8.

Okutomi, M. and Kanade, T., 1993. A multiple-baseline stereo. Pattern Analysis and Machine Intelligence, IEEE Transactions on 15(4), pp. 353–363.

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

Pollefeys, M., Koch, R., Vergauwen, M. and Van Gool, L., 1998. Metric 3d surface reconstruction from uncalibrated image sequences. In: 3D Structure from Multiple Images of Large-Scale Environments, Springer, pp. 139–154.

Pons, J.-P., Keriven, R. and Faugeras, O., 2007. Multi-view stereo reconstruction and scene flow estimation with a global image-based matching score. Int. J. Comput. Vision 72(2), pp. 179–193.

Press, W. H., Teukolsky, S. A., Vetterling, W. T. and Flannery, B. P., 2007. Numerical Recipes 3rd Edition: The Art of Scientific Computing. 3 edn, Cambridge University Press, New York, NY, USA.

Rothermel, M., Bulatov, D., Haala, N. and Wenzel, K., 2014a. Fast and robust generation of semantic urban terrain models from uav video streams. In: 2014 22nd International Conference on Pattern Recognition (ICPR), IEEE, pp. 592–597.

Rothermel, M., Haala, N. and Fritsch, D., 2014b. Generating oriented pointsets from redundant depth maps using restricted quadtrees. The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences 40(3), pp. 281.

Rothermel, M., Wenzel, K., Fritsch, D. and Haala, N., 2012. Sure: Photogrammetric surface reconstruction from imagery. In: Proceedings LC3D Workshop, Berlin, Vol. 8, pp. 1–9.

Seitz, S. M., Curless, B., Diebel, J., Scharstein, D. and Szeliski, R., 2006. A comparison and evaluation of multi-view stereo reconstruction algorithms. In: Computer Vision and Pattern Recognition. CVPR 2006, pp. 519–528.

Strecha, C., von Hansen, W., Gool, L. V., Fua, P. and Thoennessen, U., 2008. On benchmarking camera calibration and multi-view stereo for high resolution imagery. In: Computer Vision and Pattern Recognition, 2008. CVPR 2008., IEEE, pp. 1–8.

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.

Vogiatzis, G., Torr, P. H. S., Seitz, S. M. and Cipolla, R., 2008. Reconstructing relief surfaces. Image Vision Comput. 26(3), pp. 397–404.

Zach, C., Pock, T. and Bischof, H., 2007. A globally optimal algorithm for robust tv-l1 range image integration. In: 11th International Conference on Computer Vision, 2007. ICCV 2007., IEEE, pp. 1–8.