Fusion of Depth Maps with Multiple Scales

Simon Fuhrmann
TU Darmstadt

Michael Goesele
TU Darmstadt

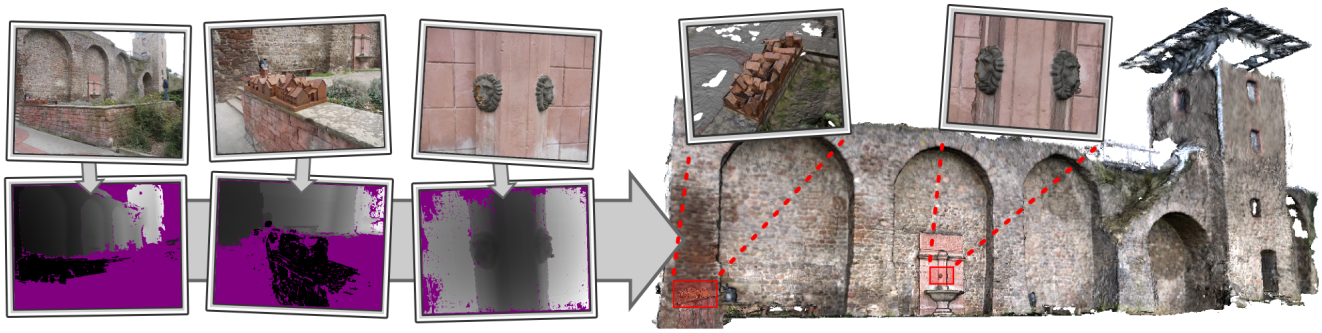

Figure 1: *Reconstruction pipeline. Input photographs (top left) depicting objects at different levels of detail. Multi-view stereo yields depth maps (bottom left), which inherit these multi-scale properties. Our system is able to fuse such depth maps and produce an adaptive mesh (right) with coarse regions as well as fine scale details (insets).*

Abstract

Multi-view stereo systems can produce depth maps with large variations in viewing parameters, yielding vastly different sampling rates of the observed surface. We present a new method for surface reconstruction by integrating a set of registered depth maps with dramatically varying sampling rate. The method is based on the construction of a hierarchical signed distance field represented in an incomplete primal octree by incrementally adding triangulated depth maps. Due to the adaptive data structure, our algorithm is able to handle depth maps with varying scale and to consistently represent coarse, low-resolution regions as well as small details contained in high-resolution depth maps. A final surface mesh is extracted from the distance field by construction of a tetrahedral complex from the scattered signed distance values and applying the Marching Tetrahedra algorithm on the partition. The output is an adaptive triangle mesh that seamlessly connects coarse and highly detailed regions while avoiding filling areas without suitable input data.

CR Categories: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Surface Reconstruction I.3.6 [Computer Graphics]: Methodology and Techniques—Graphics data structures and data types

Keywords: multi-scale depth map fusion, multi-view stereo depth maps, depth map integration, hierarchical signed distance field, surface reconstruction, marching tetrahedra

Links: DL PDF WEB VIDEO

1 Introduction

Surface reconstruction is an important problem with huge practical applications and a long history in computer graphics. The goal is to build high quality 3D surface representations from captured real-word data. Important applications include the preservation of cultural heritage, model reverse engineering, and prototyping in the multi-media industry. Typical inputs to surface reconstruction algorithms are either unorganized points or more structured data such as depth maps. In this work we will focus on the latter kind of data, which is produced by range scanners and some multi-view stereo algorithms. To fully capture an object of interest, multiple overlapping depth maps are necessary, each covering parts of the object surface. In a general acquisition framework, these depth maps need to be aligned into a common coordinate system and fused into a single, non-redundant surface representation. This process is called the *integration* or *fusion* of depth maps.

One source of depth maps are multi-view stereo (MVS) systems, which recently attained renewed interest [Seitz et al. 2006]. These algorithms reconstruct the scene geometry from photographs of the scene by regaining the 3D information lost during capture. Current structure-from-motion systems [Snavely et al. 2008] are able to recover the camera parameters of thousands of photographs under very uncontrolled conditions. This enables modern MVS algorithms to make use of the massive amount of Internet imagery for geometry reconstruction [Goesele et al. 2007; Agarwal et al. 2009; Frahm et al. 2010].

We desire to construct surface representations from the depth maps delivered by these acquisition systems, which is still an unsolved problem and difficult for various reasons. In particular, the photographs may be at different resolutions and show large variations in viewing parameters. The resulting depth maps inherit these properties and imply vastly different sampling rates of the surface. As in almost all acquisition processes, individual depth map samples are not ideal point samples. Instead, they represent the surface at a particular scale depending on viewing distance, focal length and image resolution. The extent of individual pixels when projected into 3D space can therefore dramatically vary in size. We call this the *pixel footprint*. The issue of scale and pixel footprints is crucial and requires particular care when mixing samples at different

scales. To our knowledge, this has not been solved convincingly in the surface reconstruction literature.

Apart from the scale issue, each depth map may be *locally incomplete*, i.e., contain regions without reconstructed depth values, a common artifact of depth maps produced by multi-view stereo (see Figure 1). Additionally, the individual depth values possess errors and deviate from the ground truth surface. These errors depend on the technology used to create the depth maps. Naturally, MVS approaches generate a different kind of (and much larger) error than most active range scanning systems. Given a set of depth maps of an object, some regions of the surface are typically seen by more than one depth map. We want to make use of the redundancy to suppress, or average out noise in the individual depth samples. In contrast to techniques that produce water-tight surfaces, we want to support incomplete representations and leave unobserved regions empty while closing small holes. One well known approach that handles these issues but does not take scale information into account, is volumetric range image integration (VRIP) [Curless and Levoy 1996]. Our proposed approach is volumetric, builds on some ideas developed in VRIP but solves the scale issue while sharing advantageous properties. In particular, our contributions are

- a method to construct a discrete, multi-scale signed distance field capable of representing surfaces at multiple levels of detail, yielding a *hierarchical signed distance field*,
- a processing approach that supplements uncertain signed distance values at high resolution with data from a coarser scale, *regularizing* the distance field,
- defining a *continuous signed distance field* from the hierarchical, incomplete and scattered signed distance values by building a bounded tetrahedral complex, and
- a surface extraction approach based on Marching Tetrahedra [Doi and Koide 1991] to produce output surfaces that are adaptive to the scale of the input data.

The remainder of this paper is organized as follows: We first give an overview over previous work (Section 2) before we describe our main concept in Section 3. A high level discussion of the algorithm is given in Section 4. We show how to construct the hierarchical signed distance field (Section 5) and present a regularization technique by combining data at different resolutions (Section 6). Our approach for surface extraction is described in Section 7. We evaluate the proposed algorithm and present results in Section 8. Finally we conclude our work in Section 9.

2 Related Work

Surface reconstruction is an important topic for which a large variety of techniques have been proposed over the last decades. Most reconstruction techniques aim at generating a piecewise linear surface representation (such as a triangle mesh) from the input data. Methods can be classified into reconstruction from unorganized points and techniques that use the underlying structure of the data. Examples of the former class include the classical work by Hoppe et al. [1992], Moving Least Squares surfaces [Levin 1998], RBF-based techniques [Ohtake et al. 2006], Poisson surface reconstruction (PSR) [Kazhdan et al. 2006], and Voronoi-based reconstruction [Alliez et al. 2007]. These methods operate in the most general setting and do not make any assumptions about the spatial structure of the data. The motivation to deal with a more specific type of input, namely depth maps, is that the acquisition process often provides us with additional information such as connectivity. Although we can always fall back to unorganized point-based reconstruction techniques by projecting all pixels of the depth maps in 3D space to produce unorganized point samples, intuition suggests that we should make use of the additional information to improve upon the results [Levoy 2011].

In fact, most methods that deal with *depth map integration* apply a depth map triangulation step first, where depth samples are connected in image space to form a triangulated surface, which is then lifted to 3D space. We can classify these methods into *parametric surface representations*, *surface-based methods*, and *volumetric methods*.

Early works in the field of depth map fusion impose an object-centered coordinate system for surface integration. Chen and Medioni [1991] apply a global re-parametrization of the depth maps into a unified parameter space. Integration is then simply a matter of averaging in the overlapping areas. Similarly, Higuchi et al. [1994] integrate all data points into an object-centered parametric representation and fit a deformable mesh in order to obtain a smooth model. These techniques assume a simple topology such as a cylinder or a sphere, are therefore *parametric*, and restrict the input to very simple and compact models.

Surface-based methods such as Mesh Zippering [Turk and Levoy 1994] or the co-measurements approach by Pito [1996] select one depth map for each surface region, remove redundant triangles in overlapping regions, and glue the remaining meshes together by connecting the boundaries. These methods can handle noise by local surface averaging of positions, but are very fragile in the presence of outliers and typically fail in regions of high curvature. Interestingly, these methods can, at least in theory, handle arbitrary scales since they attempt to fuse triangulated depth maps directly, and do not re-parameterize the data. Thus these methods work with natural pixel resolution.

Hilton et al. [1996] introduced the idea of representing the surface implicitly using a signed distance field computed from the individual depth maps. Curless and Levoy [1996] took this idea further by taking into account the direction of the sensor uncertainty to model the anisotropic behavior of sensor noise of the acquisition device. As in most volumetric methods, the final surface can then be extracted as zero-level set of the implicit function using standard techniques such as Marching Cubes [Lorensen and Cline 1987]. Hilton and Illingworth [1997] propose a method to reduce memory consumption of implicit functions by constructing an adaptive signed distance field stored in an octree. The octree level is adapted to surface curvature bounding the approximation error. This approach still expects depth maps with similar scale and adapts the octree with respect to geometric properties only.

Zach et al. [2007] cast the problem of depth map integration as global optimization problem, minimizing an energy functional consisting of a total variation regularization with an L_1 data fidelity term. L_1 is more robust than L_2 data fidelity in the presence of noise and outliers, but is also very expensive: Although their method produces impressive results, it is restricted to small and compact objects sampled over regular volumes because computation time and memory consumption quickly become prohibitive.

It is worth noting that Point Set Surfaces [Alexa et al. 2001] based on Moving Least Squares [Levin 1998] can produce implicit functions, which can be evaluated over a hierarchy with respect to approximation error, the local feature size, or even local scale information. Such a technique, however, requires that all sample points (from all depth maps) are located in memory. Another disadvantage is that a Point Set Surface can only define a smooth closed surface. Guennebaud and Gross [2007] describe a technique to define object boundaries but this requires an additional clipping normal for each boundary input point. Similar drawbacks also apply to most other point-based reconstruction techniques.

In this work, we address both multi-resolution input depth maps as well as creating a final surface that is adaptive with respect to the scale of the input data. While most volumetric methods operate on regular grids, some techniques use hierarchical data structures, including [Kazhdan et al. 2006; Hilton and Illingworth 1997; Soucy and Laurendeau 1992]. Sometimes these (and similar) methods are said to be "multi-resolution approaches", which typically means that the resulting mesh is adaptive. However, input data at various scales is not addressed. Further, we show why methods such as VRIP [Curless and Levoy 1996], Poisson Surface Reconstruction [Kazhdan et al. 2006] and Point Set Surfaces [Guennebaud and Gross 2007] cannot be modified in an obvious way to handle multi-scale input.

3 Concepts

In the noise-free case, i.e., if all samples are perfect, we would like to reconstruct a surface from the samples corresponding to the highest resolution information available at that location. Thus, adding infinitely many depth maps with lower resolution should not change the reconstruction. In contrast, many existing techniques converge towards the lower resolution surface if more and more low resolution depth maps are added. This issue is illustrated in Figure 2.

In VRIP, this behavior is very pronounced because the surfaces (obtained by triangulating the individual depth maps) are resampled into the volume without taking the scale into consideration. Combining a single low resolution depth map with a high resolution depth map considerably influences the high-resolution geometry. In PSR, this issue is much less apparent because PSR considers the *density* of the samples, and the *amount* of samples contained in a depth map varies with the scale of the depth map. However, the same behavior can be observed when adding the same *density* of low resolution samples and high resolution samples. Thus, in general, PSR has the same convergence behavior for n low resolution depth maps with $n \to \infty$.

A weighting scheme that leaves out the contribution of coarse information is hard to realize: In a regular volume like in VRIP, the required information is simply not present because the implicit surface is not represented at different scales. In point-based techniques, a single unreliable high-resolution sample potentially prohibits the use of coarser information essential for reliable reconstruction.

A more formal view on the issue of scale is given by scale space theory [Lindeberg 1998]. Given an image, the scale space of the image is constructed by introducing a parameter t of scale and convolving the image signal with a Gaussian filter with variance $t = \sigma^2$, thus representing the image as one-parameter family of smoothed images, which is called the *scale-space representation* of the image. This theory also applies to 3D images such as signed distance fields (SDF). We assume that the SDF of a surface at a given resolution can be approximated by a low-pass filtered SDF of the same surface at higher resolution. The scale space parameter t has the interpretation that image structures of size $t \geq 1$ (in pixel) have largely been eliminated at scale t^2. This interpretation suggests that we should be careful when combining depth maps at different scales. In particular, if we want to keep structures of a specific size in a depth map, say size t, we should avoid naïvly combining it with another depth map at scale t^2.

Another important aspect of our work is that noise in the depth maps is typically coherent between samples from a single depth map, but differs between samples from different depth maps at a different scale. In particular, this is difficult for point-based reconstruction techniques, since they can no longer exploit the fact that the consistency in noise is tied to proximity. In fact, points that are spatially close together may have very different amount of noise and should therefore be processed independently. In our system, we assume a linear correlation between the scale and the expected noise of a sample: For example, if the scale of two samples differs by a factor of two, the depth uncertainty also doubles.

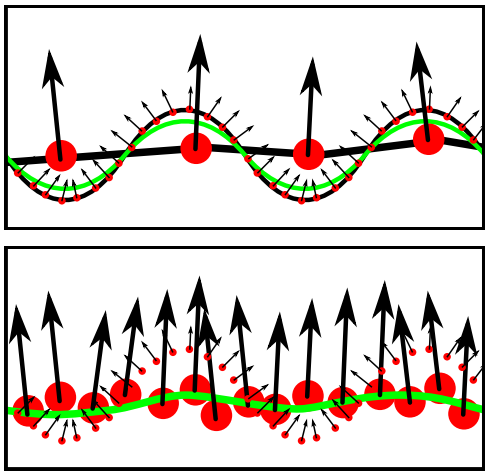

Figure 2: Top: *Point samples (red) with normals for a surface sampled at low and high resolution (black curves). The reconstructed surface (green) degenerates only slightly because the density of high resolution samples dominates.* Bottom: *Adding more and more low resolution samples causes the surface to converge towards the coarse geometry.*

We assume that the input depth maps to our system are band-limited such that they can be triangulated without significant aliasing artifacts. While this is the case for depth maps from most acquisition systems (such as multi-view stereo and structured light scanners), some technologies produce depth maps with different characteristics. For example, LIDAR scanners typically produce samples with a sample spacing that is substantially larger than their footprint. This is due to the very small point spread function of the LIDAR beam and these samples need additional filtering to suppress aliasing artifacts in the triangulated surface.

In the next section we describe the underlying ideas and properties of our technique before presenting the implementation details in Section 5. We decided to approach the reconstruction problem using a volumetric representation of the input data. Note, however, that the principle behind our solution applies to other representations as well.

4 Algorithm

One of our key ideas is to separately aggregate the contributions of the individual depth samples at their corresponding scale. We are therefore able to select a suitable scale for final surface extraction and avoid mixing up different scales. In order to do this, we aggregate geometric information (in form of the SDF) in scale space, i.e., the 3D Euclidean space plus one dimension of scale. We associate a scale with each depth sample, which then only contributes at that specific scale parameter in scale space at its 3D position. We explain how we define the scale of a sample in the next section.

So far averaging of information would not be possible because overlapping regions in the depth maps rarely have exactly the same scale. A common solution is to discretize the scale space into octaves, which yields a hierarchical representation. The levels of the octaves correspond to a doubling of scale, and all samples within

a single octave are combined to produce average surfaces. In the (unlikely) case that all depth maps contribute to a single octave only, the dataset has uniform scale, and our technique gracefully degrades to the VRIP algorithm.

Assigning each sample to exactly one scale can lead to artifacts near the boundaries of the octaves, because contributions are distributed between two neighboring octaves. We therefore transfer geometric information from the coarser octaves to the finer octaves, thus *regularizing* the fine geometry using the coarser one. Unreliable measurements, such as a surface seen at a grazing angle, are pruned from the hierarchical SDF (hSDF). Finally, we extract the isosurface from the hSDF by triangulating the zero-crossing corresponding to the finest geometric information available.

5 Constructing the hSDF

We take as input a set of registered depth maps, generated, e.g., by a range scanner or a multi-view stereo approach, optionally with confidence values and colors. The depth maps are triangulated in image space and the triangulation is lifted to 3D. If the depth disparity between two vertices of a generated triangle is above a threshold, we assume a depth discontinuity and discard the triangle. To dynamically choose the disparity threshold, we use our notion of the pixel footprint. The *pixel footprint* $F_\circ$ is the width (or height) of a fronto-parallel square corresponding to the pixel (u, v) in the image, projected to its 3D location $\vec{x}(u, v)$ on the object. We detect a depth discontinuity between two neighboring pixels if the depth disparity is above a threshold $\rho \cdot F_\circ$, where $F_\circ$ is the footprint of the pixel closer to the camera, and ρ is a user-defined constant. Triangles where at least one edge contains a detected depth discontinuity are discarded. We achieve overall good results with $\rho = 5$.

The next step of our algorithm inserts the triangulated depth maps into the hierarchical signed distance field. Our hierarchy corresponds to a primal octree, where each cube has eight voxels in the corners and is subdivided into eight sub-cubes. These sub-cubes create 27 new voxels, eight of these voxels coincide with voxels of the parent node. We explicitly keep these duplicated voxels to represent information at different levels of the hierarchy. Technically, we do not explicitly store the octree hierarchy, but insert all voxels into a map data structure, which maps the voxel index (l, I_l) to the voxel data. The index is composed of the level l and the index $I_l \in \{0, \ldots, 2^{3l} - 1\}$ within that level and uniquely determines the position of a voxel with respect to the root node's axis aligned bounding box. Initially, the data structure is empty and voxels are created as they are requested for the first time.

The triangles of each depth map are inserted one after another. For each triangle a decision is made which octree level it affects. Again, we use our notion of the pixel footprint to make that decision. Each vertex of the triangle carries an associated footprint size from the depth map pixel that generated the vertex, and we declare the smallest footprint $F_\circ$ of the three triangle vertices as the representative footprint of the triangle $F_\triangle$. To sample the triangle, we enforce that the footprint $F_\square$ of octree cells is smaller or equal to $F_\triangle \cdot \lambda^{-1}$, where λ is the sampling rate. We typically set $\lambda = 1$ and define $F_\square$ as the edge length of the octree cell (i.e., the spacing between voxels). The appropriate octree level l_T for triangle T is efficiently found by taking the binary logarithm of the root node's footprint $F_\square^R$ divided by the maximum sample spacing $F_\triangle \cdot \lambda^{-1}$:

$$l_T = \lceil \log_2 \left(\frac{F_\square^R \cdot \lambda}{F_\triangle} \right) \rceil \tag{1}$$

Once we computed the level l_T of triangle T, we need to identify all affected voxels, i.e., those voxels that fall in a band around the triangle. This is controlled by the *ramp length*, see Curless and Levoy [1996] for details. The ramp length $\gamma \cdot F_\square$ is calculated by multiplying the footprint $F_\square$ of the octree cell with the ramp size factor γ; thus the ramp length is constant for each octree level. The ramp length factor should be chosen according to the expected maximum noise of the data set and the scanning technology; reasonable parameter values are between $\gamma = 2$ for clean, range scanned data and $\gamma = 8$ for MVS datasets with heavy noise.

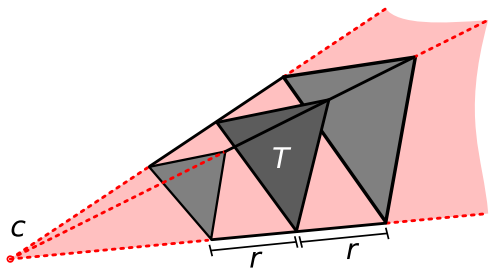

Figure 3: *Truncated tetrahedron created by shooting rays from the sensor center c through the vertices of triangle T. The ramp length is denoted by r.*

To identify affected voxels, we extrude the triangle T by following the rays from the sensor center through the triangle vertices. We bound the resulting cone and limit the volume to the ramp length around the triangle; this yields a tetrahedron with one corner truncated, see Figure 3. To simplify things, we create the bounding box of the truncated tetrahedron and analytically identify the indices of all voxels, yet created or not, inside the bounding box. We calculate the signed distance from each voxel to the triangle by shooting a ray from the camera center through the voxel and either create or update the voxel on hitting the triangle.

When creating a new voxel, we assign a weight value in addition to the distance to the voxel. Our weight value is calculated similar to VRIP [Curless and Levoy 1996]: We multiply individual weights for angle deviation (the dot product between the ray and the hit point normal), a truncated tent weight function of the absolute distance, and the confidence value at the hit point, linearly interpolated from the mesh vertices. When updating a voxel x, we use the following cumulation rules [Curless and Levoy 1996]:

$$W_{i+1}(x) = W_i(x) + w_{i+1}(x) \tag{2}$$

$$D_{i+1}(x) = \frac{W_i(x)D_i(x) + w_{i+1}(x)d_{i+1}(x)}{W_{i+1}(x)} \tag{3}$$

where $d_i(x)$ and $w_i(x)$ are the signed distance and weight values from the ith range image, and $D_i(x)$ and $W_i(x)$ are the cumulative signed distance and weight values after inserting the ith range image. For the rest of this paper, we interchangeably use the terms *weight* of a voxel and *confidence* of a voxel.

A particular surface structure can appear quite differently depending on the scale of the image (or depth map) representing the surface. Thus different geometric representations of the same surface may deviate from each other. This deviation can potentially lead to duplicated surfaces in the output mesh, see Figure 4 for an illustration. To avoid duplicated surfaces, one could use infinitely large ramps, such that voxels at coarser levels are overridden, but this is not feasible in practice. If the deviating surface at level l is, however, within the ramp of the surface of level $l + 1$, the duplication is detected and overridden. To make this mechanism work over several scales, we additionally insert the depth maps into a few coarser levels. In our experiments, we always use four coarser levels.

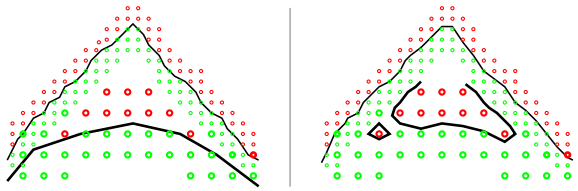

Figure 4: *Left: A high-resolution and a low-resolution depth map of the same surface and the voxels that sample the depth maps at different scales. Green and red voxels are in front of and behind the surface, respectively. Right: The isosurface extracted from the distance field yields artifacts, which we avoid by inserting at coarser scales.*

6 Regularizing the hSDF

The hierarchical signed distance field now contains a sampled representation of all input depth maps at various scales, depending on the footprints of the inserted triangles. The representation is incomplete, i.e., contains holes in unseen or unreconstructed regions. But even in areas where data is available, it might be very unreliable, having a low confidence value caused by, e.g., uncertain reconstruction or surfaces seen at grazing angles. Our goal is to improve these unreliable samples by transferring distance and confidence measures from coarser levels where available.

We make a pass through the hSDF in coarse-to-fine order, searching for occupied voxels with confidence w_l below a threshold τ_0. For each of these sub-confident voxels at level l, we interpolate its distance value from distance values at coarser level $l-1$ if all required voxels at that level are occupied. The number of required voxels varies, depending on the voxel position in the hierarchy: If a voxel at level l coincides with a voxel at level $l-1$, only the coinciding voxel is required for "interpolation". The other possible configurations require two, four, or eight voxels at the coarser level for interpolation.

We perform a weighted blend of the distance and weight values. Since the weight at the coarser level $l-1$ can be arbitrary high, we adapt the reasoning of Mitchell [1987] to this case (similar to Gortler et al. [1996]) and clamp the confidence values to τ_0 to avoid oversmoothing. τ_0 can be seen as saturation threshold. The blended voxel x with distance $\tilde{d}_l$ and weight $\tilde{w}_l$ at level l then becomes:

$$\tilde{d}_l = \frac{d_l \cdot w_l + d_{l-1} \cdot (\tau_0 - w_l) \cdot \min(1, \frac{w_{l-1}}{\tau_0})}{w_l + (\tau_0 - w_l) \cdot \min(1, \frac{w_{l-1}}{\tau_0})} \quad (4)$$

$$\tilde{w}_l = w_l + (\tau_0 - w_l) \cdot \min(1, \frac{w_{l-1}}{\tau_0}) \quad (5)$$

If the blended confidence $\tilde{w}_l$ remains below a second threshold $\tau_1 \leq \tau_0$, we delete the voxel from the octree since it is not reliable enough for reconstruction. Voxels that could not be updated due to missing information at level $l-1$ with confidence $w_l < \tau_1$ are deleted.

The *saturation threshold* τ_0 as well as the *confidence threshold* τ_1 need to be chosen according to the dataset. The confidence threshold is similar to the one in VRIP, that is often used to remove clutter in the reconstruction. If the dataset contains little redundancy, i.e., most regions are observed by one or two depth maps only, reasonable values are $\tau_0 = 0.5$ and $\tau_1 = 0.1$. For scenes with higher redundancy where regions are seen by more depth maps, the thresholds can be increased.

After all voxels have been processed, there are still duplicated voxels at different levels in the hierarchy. Since all unconfident voxels have been deleted, we ultimately trust in the remaining voxels at the highest resolution. Hence we take another pass through the octree in fine-to-coarse order and delete for each occupied voxel at level l all coinciding voxels at coarser levels $\{l-i \mid 1 \leq i \leq l\}$.

7 Extracting the ISO Surface

In the previous step we converted the hierarchical signed distance field to a scattered signed distance field by deleting unconfident and duplicated voxels. Each voxel has an associated distance value as well as optional per-voxel attributes. Although the 3D positions of the voxels are structured as they are derived from a primal octree, isosurfacing turns out to be a difficult problem.

Most prior methods apply the Marching Cubes (MC) algorithm [Lorensen and Cline 1987] to the implicit function, but this only works for regular samplings. Several approaches have been developed to get around this limitation, some of them require knowledge of the original signed distance function, others demand restrictions on the octree topology, i.e., require that the level difference between adjacent leaf nodes must not be greater than one. Dual methods pose less restrictions but require hermite data [Ju et al. 2002] or can introduce topological artifacts [Schaefer and Warren 2005]. A more recent technique [Kazhdan et al. 2007] solves these issues but requires an octree where each non-leaf node has all eight children allocated. Schroeder et al. [2004] use a unique global point numbering (vertex indices) to produce compatible triangulations across cell boundaries. We are, however, not aware of a suitable modification that generalizes the method to incomplete octrees. So none of the direct methods we know of applies to our case.

7.1 Creating the Complex

We therefore use a more general approach and consider our voxels as scattered samples of the signed distance field. We apply a global Delaunay tetrahedralization [Doi and Koide 1991] to all voxel positions. This yields a tetrahedral complex that covers the convex hull of all voxels. The downside of this approach is that the shape of the data domain (which is typically not convex) is not taken into account, and some tetrahedra connect unrelated parts of the distance field with each other. Thus erroneous interpolation between unrelated regions becomes possible, creating phantom surfaces.

To remove these tetrahedra, we define a neighborhood relation on the voxels. We then delete all tetrahedra that contain at least one edge between non-neighboring voxels. When this relation is carefully designed, we can not only detect those tetrahedra that bridge unrelated parts of the implicit function but also exploit the generic Delaunay tetrahedralization to fill small holes in the surface, by keeping some tetrahedra that would have been removed otherwise.

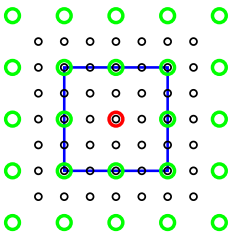

Figure 5: *The voxel neighborhood in 2D. The neighborhood of the center voxel (red) at level l consists of all voxels at levels $\geq l$ within the blue square (the 9-neighborhood at level l). In addition, the n-ring (here: 1-ring) around the blue square at each level $\geq l$ is also part of the neighborhood of the red voxel.*

Figure 6: *Data sets with negligible scale differences.* Left: *The Stanford Bunny reconstructed from the original range images.* Right: *The temple data set from the multi-view stereo evaluation effort from Seitz et al. [2006].*

Consider two voxels A and B at different levels $L(A)$, $L(B)$. Without loss of generality, assume A is the voxel at a coarser level. We define voxels A and B as neighboring if B is contained in the cube that is spanned by the 27-neighborhood of voxel A. Additionally, we enlarge this neighborhood by n voxels in each direction at the finer level of B. Thus, we can express the maximum extent of the neighborhood in each direction at the level of B as $2^{L(B)-L(A)}+n$. The same rule applies for voxels at the same level, which yields a neighborhood distance of $2^0 + n = n + 1$. We typically use $n = 2$ for very conservative hole filling. See Figure 5 for an illustration of the neighborhood relation in 2D.

7.2 Extracting the Surface

We now apply the Marching Tetrahedra algorithm [Doi and Koide 1991] to the resulting tetrahedral mesh. The extracted surface mesh is adaptive, with fewer triangles in regions where only coarse information is present and more triangles in detailed regions modeled by high-resolution depth maps. Due to the nature of the Marching Tetrahedra algorithm, the triangulation contains a lot of poorly shaped triangles that do not contribute much to the accuracy of the surface. To address this, we optimize the tetrahedralization for surface extraction similar to Schaefer and Warren [2005]. For each edge in the tetrahedral mesh with a zero crossing that is located very close to one of the edge vertices, we pull the vertex along the edge onto the crossing and set its distance value to zero. In combination with a simple modification of the Marching Tetrahedra algorithm to prevent zero-area faces, this results in significantly less degenerate triangles, see Figure 7.

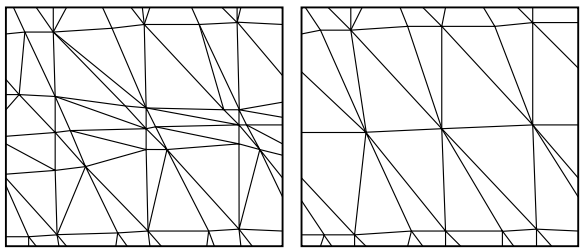

Figure 7: *Optimized isosurface extraction. The original MT triangulation (left) and the optimized triangulation (right) obtained by pulling vertices of the tetrahedral mesh to the zero crossing.*

Applying Marching Tetrahedra is an attractive choice for isosurface extraction because of its stability, simplicity and performance. The downside of this simple approach, however, is the vast amount of tiny triangles that are created. Delaunay-based meshing algorithms [Boissonnat and Oudot 2005] are more involved but produce well sampled, guaranteed-quality triangle meshes.

8 Evaluation and Results

We now present some results on various datasets. Figure 6 (left) shows a reconstruction of the Bunny dataset from laser scanned range images provided by the *Stanford Scanning Repository*. This dataset has negligible scale variations, thus triangles are typically inserted at a constant level. In this case, our algorithm gracefully degrades to the behavior of and produces very similar results to VRIP. One difference, however, is that the parameters of our method are more intuitive. We simply set the sampling rate $\lambda = 1$ and the ramp size factor $\gamma = 4$, and our algorithm determines the appropriate voxel spacing, which needs to be explicitly specified in VRIP. Since all depth values have more or less the same footprint, we can omit the regularization step, setting $\tau_1 = \tau_0 = 0$.

Figure 6 (right) shows a reconstruction of the Temple data set from the Middlebury MVS evaluation effort [Seitz et al. 2006]. We first recovered the depth maps from the 312 input images using the MVS system by Goesele et al. [2007], and fused all depth maps using ramp size factor $\gamma = 8$ and sampling rate $\lambda = 1$. Note that we used the same MVS system for all reconstructions.

Our next dataset consists of over 700 photographs of the Cathedral of Notre Dame de Paris downloaded from Flickr with vastly different resolutions and viewing parameters. This dataset is challenging, contains very uncontrolled images taken with different cameras, thus the depth maps are tainted with a lot of noise (see Figure 8 for our surface reconstruction result). Since people tend to make most photos of the center portal, we focused our attention to that region for a comparison with VRIP and Poisson, see Figure 9. To make the comparison fair, we specified a bounding box around the center portal for VRIP and Poisson, and reconstructed only within this region. Even though VRIP and Poisson operated on a subset of the data, our reconstruction shows more detail and yields a more crisp result, but also more noise. The influence of coarse depth maps on the VRIP reconstruction quality is clearly visible; for Poisson this effect is less visible but still noticeable.

We captured a dataset called *Stones* (118 photographs) that shows a metal door next to a wall built of stones. Each image represents the geometry at a different scale as we moved closer to the wall while taking the images. The reconstruction consists of various, seamlessly connected scales, from coarse regions to highly accurate geometry on the order of millimeters (see Figure 10).

Finally, we evaluate our reconstruction pipeline with a large MVS dataset called *Citywall* (see Figure 1) consisting of 561 photographs and corresponding depth maps. We reconstructed each depth map with resolution 500×375. In this scene, the footprint of the individual depth samples varies dramatically. A focus in this scene was the detailed reconstruction of the fountain with its two lion heads and the replica of the historic city, see Figures 11 and 12.

Figure 8: *Some input photographs of the facade of Notre Dame de Paris that show the variations in scale (top row), and a surface reconstruction from about 700 depth maps (bottom row).*

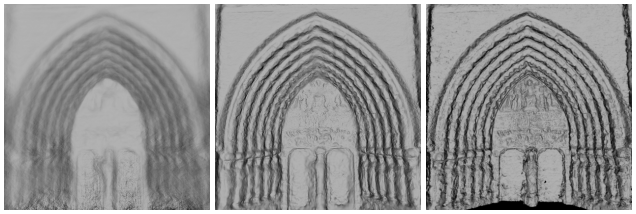

Figure 9: *A comparison of VRIP (left), Poisson surface reconstruction (middle) and our reconstruction (right). Both VRIP and Poisson reconstructions are smoother but also less detailed.*

We compare our full reconstruction with a Poisson surface reconstruction of the fountain. To do this, we clipped all samples with a bounding box around the fountain and provided the clipped point set to PSR. The reconstruction clearly shows an overly smooth result, caused by many low-resolution samples from depth maps taken further away from the surface (see Figure 13).

The running time and memory consumption of our algorithm is dependent on the amount, resolution, and scale of the input depth maps. Reconstruction details are given in the Table 1. The table shows the name of the dataset, the number of fused depth maps, the resolution of the depth maps (if uniform), the time required for reconstruction, and the number of generated voxels.

9 Conclusion

We presented a hierarchical, volumetric approach for depth map fusion that takes into account the scale (or footprint) of the individual depth samples to extract adaptive, high-quality surfaces. Although the basic principle of our algorithm is inspired by VRIP [Curless and Levoy 1996], the new algorithm is, to our knowledge, the first successful attempt to handle multi-resolution data. Our results show a clear improvement over traditional depth map fusion techniques.

Acknowledgements: This work was supported in part by the DFG Emmy Noether fellowship GO 1752/3-1. We want to thank the following Flickr users for permission to use their images in Figure 8, ordered by appearance: *Brian Jeffery Beggerly*, *Eric Wilcox*, *Yvonne Yuen*, *Sara Hopkins*, and *Brian Beaver*.

Figure 10: *The top row shows some input images from the* Stones *dataset, and the middle row shows the reconstructed scene. The bottom row shows a close-up view of our reconstruction with and without texture (left, middle), and the corresponding VRIP reconstruction (right).*

Name	DMs	Resolution (pixel)	Time (hrs, min)	Voxel (10^6)
Notre Dame	715	mixed	7h + 1h + 4m	103
Stones	118	1000×750	2h + 23m + 2m	41
Citywall	564	500×375	6h + 1h + 4m	49

Table 1: *Statistics of the reconstruction results. The individual timings are for constructing the octree, building the tetrahedral mesh and extracting the isosurface, respectively.*

References

AGARWAL, S., SNAVELY, N., SIMON, I., SEITZ, S. M., AND SZELISKI, R. 2009. Building Rome in a day. In *Proc. ICCV*, 72–79.

ALEXA, M., BEHR, J., COHEN-OR, D., FLEISHMAN, S., LEVIN, D., AND SILVA, C. T. 2001. Point set surfaces. In *Proc. VIS*, 21–28.

ALLIEZ, P., COHEN-STEINER, D., TONG, Y., AND DESBRUN, M. 2007. Voronoi-based variational reconstruction of unoriented point sets. In *Proc. SGP*, 39–48.

BOISSONNAT, J.-D., AND OUDOT, S. 2005. Provably good sampling and meshing of surfaces. *Graphical Models 67*, 405–451.

CHEN, Y., AND MEDIONI, G. 1991. Object modeling by registration of multiple range images. In *Proc. ICRA*, 2724–2729.

CURLESS, B., AND LEVOY, M. 1996. A volumetric method for building complex models from range images. In *Proc. SIGGRAPH*, 303–312.

Figure 11: *A detailed view on the fountain of the Citywall dataset.*

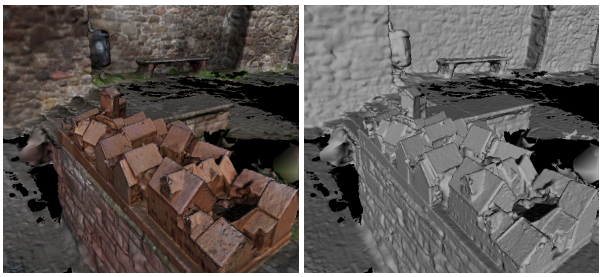

Figure 12: *A detailed view on a miniature replica of a historic city contained in the Citywall dataset.*

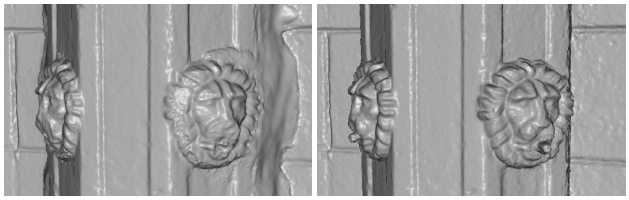

Figure 13: *Poisson surface reconstruction on a small bounding box around the fountain (left). The reconstruction yields a smooth and flat result whereas our result (right) features more detailed geometry (e.g., compare the spout at the mouth).*

DOI, A., AND KOIDE, A. 1991. An efficient method of triangulating equi-valued surfaces by using tetrahedral cells. *IEICE Trans. E74*, 1, 214–224.

FRAHM, J.-M., GEORGEL, P., GALLUP, D., JOHNSON, T., RAGURAM, R., WU, C., JEN, Y.-H., DUNN, E., CLIPP, B., LAZEBNIK, S., AND POLLEFEYS, M. 2010. Building Rome on a cloudless day. In *Proc. ECCV*, 368–381.

GOESELE, M., SNAVELY, N., CURLESS, B., HOPPE, H., AND SEITZ, S. M. 2007. Multi-view stereo for community photo collections. In *Proc. ICCV*.

GORTLER, S. J., GRZESZCZUK, R., SZELISKI, R., AND COHEN, M. F. 1996. The lumigraph. In *Proc. SIGGRAPH*, 43–54.

GUENNEBAUD, G., AND GROSS, M. 2007. Algebraic point set surfaces. In *Proc. SIGGRAPH*.

HIGUCHI, K., HEBERT, M., AND IKEUCHI, K. 1994. Building 3D models from unregistered range images. In *Proc. ICRA*, 2248–2253.

HILTON, A., AND ILLINGWORTH, J. 1997. Multi-resolution geometric fusion. In *Proc. 3DIM*, 181–188.

HILTON, A., STODDART, A., ILLINGWORTH, J., AND WINDEATT, T. 1996. Reliable surface reconstruction from multiple range images. In *Proc. ECCV*. 117–126.

HOPPE, H., DEROSE, T., DUCHAMP, T., MCDONALD, J., AND STUETZLE, W. 1992. Surface reconstruction from unorganized points. In *Proc. SIGGRAPH*, 71–78.

JU, T., LOSASSO, F., SCHAEFER, S., AND WARREN, J. 2002. Dual contouring of hermite data. In *Proc. SIGGRAPH*, 339–346.

KAZHDAN, M., BOLITHO, M., AND HOPPE, H. 2006. Poisson surface reconstruction. In *Proc. SGP*, 61–70.

KAZHDAN, M., KLEIN, A., DALAL, K., AND HOPPE, H. 2007. Unconstrained isosurface extraction on arbitrary octrees. In *Proc. SGP*, 125–133.

LEVIN, D. 1998. The approximation power of moving least-squares. *Mathematics of Computation 67*, 1517–1531.

LEVOY, M., 2011. Range data versus 3D models - a caveat on the use of these models. `http://graphics.stanford.edu/data/3Dscanrep/`.

LINDEBERG, T. 1998. Feature detection with automatic scale selection. *International Journal of Computer Vision 30*, 2, 79–116.

LORENSEN, W., AND CLINE, H. 1987. Marching Cubes: a high resolution 3D surface construction algorithm. *Proc. SIGGRAPH 21*, 5, 79–86.

MITCHELL, D. P. 1987. Generating antialiased images at low sampling densities. In *Proc. SIGGRAPH*, 65–72.

OHTAKE, Y., BELYAEV, A., AND SEIDEL, H.-P. 2006. Sparse surface reconstruction with adaptive partition of unity and radial basis functions. *Graphical Models 68*, 1, 15–24.

PITO, R. 1996. Mesh integration based on co-measurements. In *Proc. ICIP*, 397–400.

SCHAEFER, S., AND WARREN, J. 2005. Dual marching cubes: Primal contouring of dual grids. *Computer Graphics Forum 24*, 195–201.

SCHROEDER, W. J., GEVECI, B., AND MALATERRE, M. 2004. Compatible triangulations of spatial decompositions. In *Proc. VIS*, 211–218.

SEITZ, S. M., CURLESS, B., DIEBEL, J., SCHARSTEIN, D., AND SZELISKI, R. 2006. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *Proc. CVPR*, 519–528.

SNAVELY, N., SEITZ, S. M., AND SZELISKI, R. 2008. Skeletal sets for efficient structure from motion. In *Proc. CVPR*.

SOUCY, M., AND LAURENDEAU, D. 1992. Multi-resolution surface modeling from multiple range views. In *Proc. CVPR*, 348–353.

TURK, G., AND LEVOY, M. 1994. Zippered polygon meshes from range images. In *Proc. SIGGRAPH*, 311–318.

ZACH, C., POCK, T., AND BISCHOF, H. 2007. A globally optimal algorithm for robust TV-L_1 range image integration. In *Proc. ICCV*.