High Resolution Colored Surface Reconstruction from Oriented Points

Fatih Calakli
Department of Computer Science
Brown University
Providence, RI 02912
fcalakli@cs.brown.edu

Abstract

In this report we address the problem of reconstructing the surface geometry, topology, and color map of a 3D scene from a finite set of colored oriented points. These data sets are nowadays obtained using a variety of techniques, including multi-view stereo reconstruction methods from multiple 2D images. We describe a novel variational method which reduces the problem to the solutions of sparse systems of linear equations. We first use the point positions and orientation vectors to reconstruct the geometry and topology of a watertight surface represented as an adaptively tesselated polygon mesh. The method then smoothly extrapolates the color information from the points to the surface. Experimental evidence is presented to show that the resulting method produces high quality polygon meshes with smooth color maps which accurately approximate the source colored oriented points.

1 Introduction

This report addresses the problem of reconstructing the geometry and topology of a watertight surface, and a smooth surface color map, from a finite set of colored oriented points. This problem has a wide range of applications in industry, entertainment, virtual reality, archeology, forensics, art, medical imaging, and many other areas. Colored oriented point clouds are nowadays obtained using various optical measuring devices, such as laser scanners and inexpensive structured lighting systems, by other computational means such as multi-view stereo reconstruction algorithms, and also result from large scale simulations. We are particularly interested here in the reconstruction of colored surfaces from multi-view image and video data captured using consumer level digital and video cameras. The problem is particularly challenging due to the non-uniform sampling nature of the data collection processes. Common problems to cope with include: non-uniform sampled data due to concave surfaces, missing data due to inaccessible regions, noisy data due to sensor incapabilities, and distorted data due to misalignment of partial scans. As well as other researchers who have contributed to the field, we are particularly concerned with developing algorithms robust with respect to these issues, and to develop efficient algorithms which can scale up to very large data sets.

The prior art in surface reconstruction methods is extensive. We only discuss here some of the existing methods which reconstruct the geometry and topology of the surface from the point locations and orientation vectors. For a more extensive description of the prior art we refer the reader to a recent survey [20]. Despite the long history, the area is still active. One family of algorithms produces interpolating polygon meshes where some of the input points become vertices of the polygons [3]. Most of these algorithms are based on constructing Voronoi diagrams and Delaunay triangulations of the data points, and many come with guaranteed reconstruction quality [10]. Another family of algorithms are based on implicit function fitting, and produce approximating surfaces which are more desirable in the presence of noise. These methods include [2, 14–16] which reconstruct a binary in-

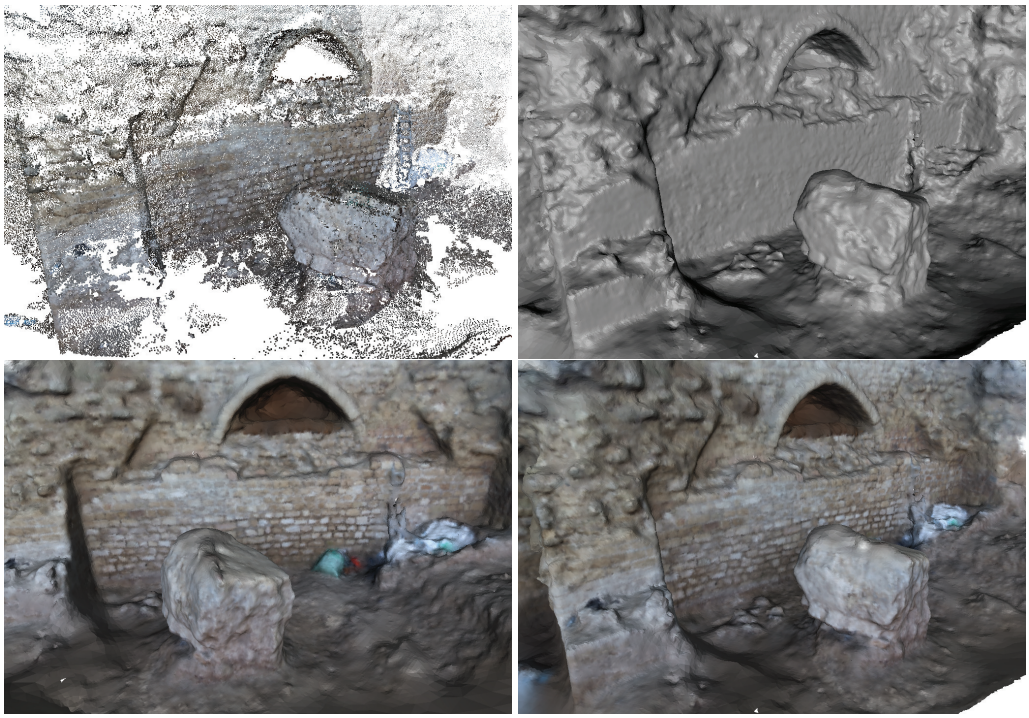

Figure 1: Reconstruction of the side of a castle model: The input point cloud (top-left), Surface reconstructed by the proposed algorithm(top-right), Two views from the surface and color map reconstructed by the proposed algorithm (bottom).

dicator function; [4,5,9,13] which estimate a signed distance function; and [1,6,11,17,18,22] based on local function fitting and blending. Most of these methods reduce the surface reconstruction problem to the solution of a large scale numerical optimization problem, and use schemes based on space partition data structures such as octrees to reduce the computational and storage complexity.

In [5] we introduced the Smooth Signed Distance Surface Reconstruction (SSD) method for reconstructing the geometry and topology of a watertight surface. In this report we show how to extend this method to the reconstruction of colored surfaces by estimating an additional continuous color map defined on the reconstructed surface geometry. This color map approximates the colors associated with the data points at the point locations, and extrapolate with continuity those colors to areas where the sampling rate is low. The color map estimation step extends the variational formulation in [5] to accommodate appearance estimation over surfaces defined by implicit functions. As a result, each problem reduces to the solution of a sparse linear system which can be solved independently of the geometry and topology estimation by using an appropriate solver.

We devote Section 2, Section 4, and Section 5 to describe the variational formulation, the discretization, and octree-based implementation, respectively. We present our results in Section 7. Figure 1 shows one example of surface and color map reconstructed with the proposed method. Note the good behavior near boundaries, holes, and regions with uneven sampling.

2 Continuous Formulation

Given a colored oriented point cloud $\mathcal{D} = \{(p_1, n_1, c_1), \ldots, (p_N, n_N, c_N)\}$, where p_i is a surface location sample, n_i is the corresponding surface normal sample oriented towards the outside of the object, and c_i is the corresponding color sample, we are concerned with the problem of reconstructing a surface S defined by an implicit equation $S = \{x : f(x) = 0\}$, where $f : V \to \mathbb{R}$ is the implicit function defined on a bounded volumetric domain V contained in Euclidean three dimensional space, and a smooth color map $g : V \to \mathbb{R}$ defined on the same domain with intensity values, so that $f(p_i) \approx 0$, $\nabla f(p_i) \approx n_i$, and $g(p_i) \approx c_i$, for $i = 1, \ldots, N$.

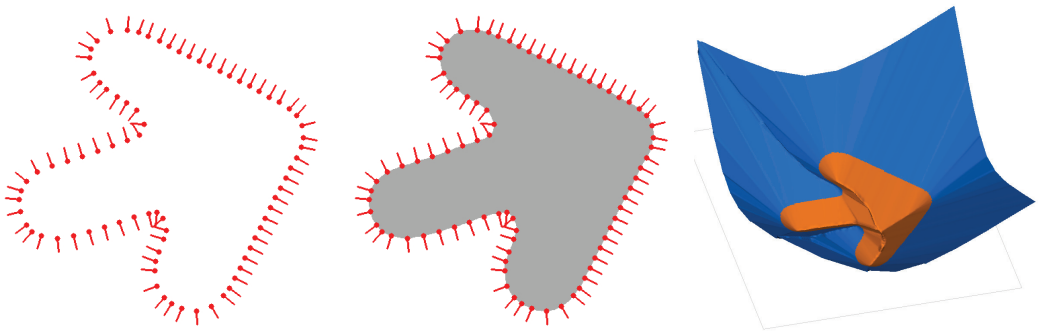

Figure 2: A 2D oriented point cloud (left) as samples of a 2D curve which is represented as a level set of an indicator function (middle) (gray=1,white=0). The graph of the smooth signed distance function (right) estimated by our algorithm from the same point cloud.

2.1 Surface Reconstruction

If the implicit function $f(x)$ is continuous, then the surface S is watertight. Without loss of generality we will assume that $f(x) < 0$ inside and $f(x) > 0$ outside of the object. As a reference, in both FFT Surface Reconstruction [14] and Poisson Reconstruction [15] the implicit function is forced to be the indicator function of the volume bounded by the surface S. This function is identically equal to zero outside, to one inside, and discontinuous exactly on S, as shown in a 2D example in Figure 2.

Usually we are only interested in reconstructing the surface S within a bounded volume V, such as a cube containing all the data points, which may result in an open surface at the boundaries of the volume V, particularly when the points only sample one side of the object, there are holes in the data, or more generally if the points are not uniformly distributed along the surface of the solid object.

The gradient $\nabla f(x)$ of a function $f(x)$ with first order continuous derivatives is a vector field normal to the level sets of the function, and in particular to the surface S. We will further assume that the gradient field on the surface S is unit length, which allows us to directly compare the gradient of the function with the point cloud normal vectors. Since the points p_i are regarded as samples of the surface S, and the normal vectors as samples of the surface normal at the corresponding points, the implicit function should ideally satisfy $f(p_i) = 0$ and $\nabla f(p_i) = n_i$ for all the points $i = 1, \ldots, N$ in the data set. Since these two conditions are hard to satisfy in the presence of measurement noise, we require that these conditions be satisfied in the least-squares sense. As a result, we consider the problem of minimizing the following data energy

$$\mathcal{E}_{\mathcal{D}}(f) = \mathcal{E}_{\mathcal{D}_0}(f) + \lambda_1 \mathcal{E}_{\mathcal{D}_1}(f) \tag{1}$$

where

$$\mathcal{E}_{\mathcal{D}_0}(f) = \frac{1}{N}\sum_{i=1}^{N} f(p_i)^2 \quad \mathcal{E}_{\mathcal{D}_1}(f) = \frac{1}{N}\sum_{i=1}^{N} \|\nabla f(p_i) - n_i\|^2$$

and λ_1 is a positive constant used to give more or less weight to the second energy term with respect to the first term. The normalization by the number of points is introduced to make these two parameters independent of the total number of points. At this point this normalization has no effect, but plays a role after we add a third regularization term.

Depending on which family $\mathcal{F}$ of functions $f(x)$ is considered as the space of admissible solutions, the problem of minimizing the energy function $\mathcal{E}_{\mathcal{D}}(f)$ of equation (1) may or may not be well conditioned. In particular, this energy function does not specify how the function should behave away from the data points. Many parameterized families of functions have parameters highly localized in space, and the energy function of equation (1) may not constrain all the parameters. If the unconstrained parameters are arbitrarily set to zero, the estimated surface S may end up containing spurious components located far away from the data points. To address this problem we add the following regularization term to the energy function

$$\mathcal{E}_{\mathcal{R}}(f) = \frac{1}{|V|}\int_V \|Hf(x)\|^2 \, dx \tag{2}$$

where $Hf(x)$ is the Hessian matrix of f, the 3×3 matrix of second order partial derivatives of $f(x)$, and the norm of the matrix is the Frobenius matrix norm, i.e., the sum of the squares of the nine elements of the matrix. The integral is over the volume V where we are interested in reconstructing the surface, $|V| = \int_V dx$ is the measure of this volume.

The total energy function in the proposed formulation is a weighted average of the data and regularization terms

$$\mathcal{E}(f) = \mathcal{E}_{\mathcal{D}_0}(f) + \lambda_1 \mathcal{E}_{\mathcal{D}_1}(f) + \lambda_2 \mathcal{E}_{\mathcal{R}}(f) \,, \tag{3}$$

and λ_2 is another positive parameter. Increasing λ_2 with respect to λ_1 produces a smoother result.

Note that the three columns of the Hessian matrix $Hf(x)$ are the partial derivatives of the gradient $\nabla f(x)$ with respect to the three cartesian variables:

$$Hf(x) = \left[\frac{\partial \nabla f(x)}{\partial x_1} \; \frac{\partial \nabla f(x)}{\partial x_2} \; \frac{\partial \nabla f(x)}{\partial x_3} \right]$$

As a result, by forcing the square norm of the Hessian matrix to be close to zero, the regularization term makes the gradient of the function almost constant away from the data points. In the vicinity of the data points the data energy terms dominate the total energy, and make the function to approximate the signed distance function. Away from the data points the regularization energy term dominates the total energy and tends to make the gradient vector field $\nabla f(x)$ constant.

2.2 Color Map Reconstruction

We are also interested in estimating the color map g defined on the same bounded volume V. Since the points p_i are regarded as samples of the surface S, and the colors c_i as samples of the surface color at the corresponding points, the color function should ideally satisfy $g(p_i) = c_i$ for all the points $i = 1, \ldots, N$ in the data set. However, we again require that this condition be satisfied in the least-squares sense. As a result, we consider the problem of minimizing the following data energy

$$\mathcal{Q}_{\mathcal{D}}(g) = \frac{1}{N} \sum_{i=1}^{N} (g(p_i) - c_i)^2 \,. \tag{4}$$

Depending on which family $\mathcal{G}$ of functions $g(x)$ is considered as the space of admissible solutions, the problem of minimizing the energy function $\mathcal{Q}_{\mathcal{D}}(g)$ of equation (4) may or may not be well conditioned. In particular, this energy function does not specify how the function should behave away from the data points. To constraint all the parameters we add the following regularization term to the energy function

$$\mathcal{Q}_{\mathcal{R}}(g) = \frac{1}{|V|} \int_V \|\nabla g(x)\|^2 \, dx \tag{5}$$

where $\nabla g(x)$ is the gradient vector of g, the 3×1 vector of first order partial derivatives of $g(x)$, and the norm of the vector is the L^2-norm. The integral is over the same volume V where we reconstruct the surface, and again $|V| = \int_V dx$ is the measure of this volume. The total energy function in the proposed formulation is a weighted average of the data and regularization terms

$$\mathcal{Q}(g) = \mathcal{Q}_{\mathcal{D}}(g) + \mu \, \mathcal{Q}_{\mathcal{R}}(g) \,, \tag{6}$$

and μ is a positive constant. Increasing μ produces a smoother result. In the vicinity of the data points the data energy term dominate the total energy, and makes the function to approximate the true appearance. Away from the data points the regularization energy term dominates the total energy and tends to make the color function $g(x)$ constant.

3 Linearly Parameterized Families

In this section we are concerned with the existence and uniqueness of solution for the variational problems defined by the energy functions $\mathcal{E}(f)$ and $\mathcal{Q}(g)$ of equations 3 and 6 respectively. We restrict the analysis here to families of functions linearly parameterized by a finite number of parameters. Explicitly, a member of this family can be written as a linear combination of certain basis functions

$$f(x) = \sum_{\alpha \in \Lambda} f_\alpha \, \phi_\alpha(x) \tag{7}$$

where α denotes an index which belongs to a finite set Λ, say with K elements, $\phi_\alpha(x)$ is a basis function, and f_α is the corresponding coefficient. The K basis functions are chosen in advance, and are not regarded as variables in the problem. The function can also be written as an inner product

$$f(x) = \Phi(x)^t F \tag{8}$$

where $F = (f_\alpha : \alpha \in \Lambda)$ and $\Phi(x) = (\phi_\alpha(x) : \alpha \in \Lambda)$ are K-dimensional column vectors. Restricting the energy energy $\mathcal{E}(f)$ of equation 3 to this function results in a non-homogeneous and quadratic function $F^t A F - 2b^t F + c$ in the K-dimensional parameter vector F. The matrix A is symmetric and positive definite, and the resulting minimization problem has a unique minimum. The global minimum is determined by solving the system of linear equations $AF = b$.

Similarly, when we restrict the color function $g(x)$ to another linearly parametrized function

$$g(x) = \Psi(x)^t G\,, \tag{9}$$

the energy function $\mathcal{Q}(g)$ of equation 6 results in a non-homogeneous and quadratic energy $G^t B G - 2d^t G + e$ in the K-dimensional parameter vector G. The matrix B is symmetric and positive definite, and the resulting minimization problem has a unique minimum. The global minimum is determined by solving the system of linear equations $BG = d$.

4 Discretization with Discontinuous Function

A hybrid of finite-element and finite differences scheme is presented in [5] to discretize the energy function $\mathcal{E}(f)$ of equation 3: a finite element discretization for the function $f(x)$, and finite differences discretizations for the gradient $\nabla f(x)$ and the Hessian $Hf(x)$. Since all these terms are written as homogeneous linear functions of the same parameter vector F, the problem still reduces to the solution of linear equations $AF = b$. As this particular discretization is applicable to adaptive grids, we adopt a similar approach to discretize the energy function $\mathcal{Q}(g)$ of equation 6. We use a finite element discretization for the color function $g(x)$, and finite differences discretization for the gradient $\nabla g(x)$.

To simplify the notation, we first consider a regular hexahedral grid partition of space. We assume that the volume V is a unit cube $V = [0,1] \times [0,1] \times [0,1]$, and that each of the three axes is split into M equally long segments, resulting in M^3 hexahedral cells and $(M+1)^3$ vertices. In this discretization, the indices $\alpha \in \Lambda$ are the grid cell multi-indices $\alpha = (i, j, k)$, for $0 \le i, j, k < M$. The coefficients g_α are then the values of the color function $g(x)$ at the grid centroids $p_\alpha = ((i+0.5)/M, (j+0.5)/M, (k+0.5)/M)$. Determining which cell C_α each point p_i falls into reduces to quantization. If $p_i \in C_\alpha$ then

$$g(p_i) \approx g_\alpha\,. \tag{10}$$

Since the piecewise constant function leads to zero first derivatives within each cell, the square norm of the gradient $\nabla g(x)$ is a generalized function supported on the square faces shared by neighboring voxels (as Dirac's δ "function"), and the integral over the volume V reduces to a finite sum over the faces. Hence, we only need an approximation to $\|\nabla g(x)\|^2$ at the faces shared by cells. If C_α and C_β are two such cell that share a face, then we use finite differences

$$\nabla_{\alpha\beta} g = \frac{1}{\Delta_{\alpha\beta}} \left(g_\alpha - g_\beta \right) \tag{11}$$

where $\Delta_{\alpha\beta}$ is the Euclidean distance between the centers of the cells. As a result, the integral in the regularization term of equation 5 becomes a sum over all the neighboring cell pairs

$$\mathcal{Q}_{\mathcal{R}}(g) \approx \frac{1}{|V|} \sum_{(\alpha,\beta)} \frac{|V|_{\alpha\beta}}{\Delta_{\alpha\beta}} \left(g_\alpha - g_\beta \right)^2 \tag{12}$$

where $|V|_{\alpha\beta}$ is the area of the face shared by cells C_α and C_β, and $|V| = \sum_{(\alpha,\beta)} |V|_{\alpha\beta}$.

5 Octree-Based Implementation

A regular grid formulation results in high storage requirements and slow algorithms which limit its appeal. Instead, we adaptively construct an octree based on the location of the data points. In

the recursive subdivision algorithm, a cell is split if the number of points contained is larger than a prespecified value, and the cell depth does not exceed another prespecified value. The data set is recursively partitioned into subsets of points $\mathcal{D}_\alpha$ associated with cells C_α. For octrees the multi-indices $\alpha = (L, i, j, k)$ must be augmented with a level (or depth) value L where $0 \leq L \leq L_{\text{MAX}}$. The correspondence between indices α and cells is 1-1.

The finite-differences discretization presented in the previous section for regular hexahedral grids extends in a natural way to octree grids, without any changes. There is one function value g_α associated with each leaf cell of the octree. As a result, within each cell we use g_α to discretize the function $g(x)$, and use the discretization of the gradient $\nabla g(x)$ of equation 11, where the weight $|V|_{\alpha\beta}$ of equation 12 is now the area of the common face.

We first estimate the implicit function $f(x)$ using an iterative cascading multigrid approach described in [5]. We start by solving the problem on a much coarser level than desired using a simple conjugate gradient solver. Then use it to initialize the solution at the next level, which is then refined by the conjugate gradient solver.

We then turn our focus into estimating the color function $g(x)$ on the same level that is used to estimate the implicit function $f(x)$. We need to minimize the total energy of equation 6

$$\mathcal{Q}(g) \approx \frac{1}{N} \sum_{\alpha} \sum_{i:p_i \in \mathcal{D}_\alpha} (g_\alpha - c_i)^2 + \frac{\mu}{|V|} \sum_{(\alpha,\beta)} w_{\alpha\beta} \left(g_\alpha - g_\beta\right)^2 , \tag{13}$$

where $w_{\alpha\beta} = |V|_{\alpha\beta}/\Delta_{\alpha\beta}$. Note that this formulation still reduces to the solution of a linear system system $BG = d$. After accumulating the elements of matrix B and vector d, for reasonably sized problems one can use a direct solver. Instead, we use an iterative linear solver, namely Gauss-Seidel method. We update one g_α value at a time by using the following update equation

$$g_\alpha = \frac{|V| \sum_{i:p_i \in \mathcal{D}_\alpha} c_i + \mu N \sum_{\beta \in \alpha^*} g_\beta w_{\alpha\beta}}{|V||\mathcal{D}_\alpha| + \mu N \sum_{\beta \in \alpha^*} w_{\alpha\beta}} , \tag{14}$$

where α^* are the set of multi-indices of octree leaf cells that share a common face with cell C_α, and $|\mathcal{D}_\alpha|$ is the total number of points in $\mathcal{D}_\alpha$. To compute an RGB color map, we repeat this minimization procedure for each color channel independently.

We use the Dual Marching Cubes algorithm [19] to construct a polygonal approximation of the isolevel zero of the implicit function $f(x)$. Dual Marching Cubes requires values of the implicit function defined at the centroids of the octree leaf cells. The computed implicit function is defined at the vertices of the primal graph. An implicit function value is computed for the centroid of each leaf cell by averaging the values associated with the eight cell corners. On the other hand, the results of the color function are already defined at centroids of the octree leaf cells, so we can directly transfer coefficients g_α to the mesh with no further function evaluation, i.e., color per vertex attribute on the final mesh.

6 Evaluation of Surface Reconstruction Methods

Our algorithm consists of surface reconstruction and color map estimation applied successively. The surface reconstruction step mainly follows the variational formulation presented in Smooth Signed Distance Surface Reconstruction (SSD) [5]. We include this section to motivate the use of SSD against Multi-Level Partition of Unity Implicits (MPU) [17], Poisson Surface Reconstruction (Poisson) [15], and Streaming Surface Reconstruction Using Wavelets (D4 Wavelets) [16]. We

Method	Time (Sec)	Memory (MB)	Polygons
MPU	27	148	378925
Poisson	43	283	319989
D4	17	63	365974
SSD	72	264	346652

Table 1: The running time, the peak memory usage, and the number of triangles in the reconstructed surface of the David's head generated with different methods.

Figure 3: Input point cloud the Chiquita model (top-left), and reconstructions using SSD (top-middle and top-right), MPU (bottom-left), Poisson (bottom-middle), D4 Wavelets (bottom-right) surface reconstruction methods.

evaluate these methods in terms of speed, memory, efficiency, and accuracy. We show that, in terms of the quality of reconstruction, SSD outperforms these other methods even on data sets associated with extreme complications: non-uniform sampling, sensor noise, and missing data. In addition, its performance is still comparable to these methods.

Our initial test case is the head of Michalengelo's David raw dataset of 1 million samples. This dataset is assembled from many range images acquired by a laser scanner. All methods operate on the same data set at a maximal octree depth of 8, and produce surfaces of comparable quality. Table 1 summarizes the performance characteristics of each algorithm. Our second test case is the Chiquita model raw dataset comprising $705,375$ samples. This dataset was acquired by scanning the real world object with an inexpensive 3D structured lighting system [8]. It is an example of *non-uniformly* distributed points with many *inaccurate* normals. Figure 3 compares different reconstructions at a maximal octree depth of 9. MPU produces spurious surface sheets. Although D4

Model	MPU	Poisson	D4	SSD
Armadillo	1.0000	0.4617	0.2383	0.3514
Dragon	0.8779	1.0000	0.5301	0.6810
Horse	0.0752	0.0827	1.0000	0.0551
Igea	1.0000	0.7761	0.5701	0.4018

Table 2: Hausdorff distance between real surfaces and reconstructed surfaces. Each row is normalized by the maximum error to provide relative comparison (lower is better).

Wavelets performs much better, it can not fill in gaps reasonably. Poisson, and SSD Reconstruction both result in pleasing surfaces with subtle differences. To evaluate the numerical accuracy of the reconstruction results we follow the same strategy as in [16]. We first sample points from a known model, then reconstruct surfaces using each method with this point set. We then compute the Hausdorff distance between each reconstructed model and the known model using Metro tool [7]. Table 2 summarizes the result. Note that, in all cases, SSD reconstruction recovers the surfaces with higher degree of accuracy than the Poisson reconstruction. And in two cases (Horse, and Igea) it also outperforms D4 Wavelets.

7 Results

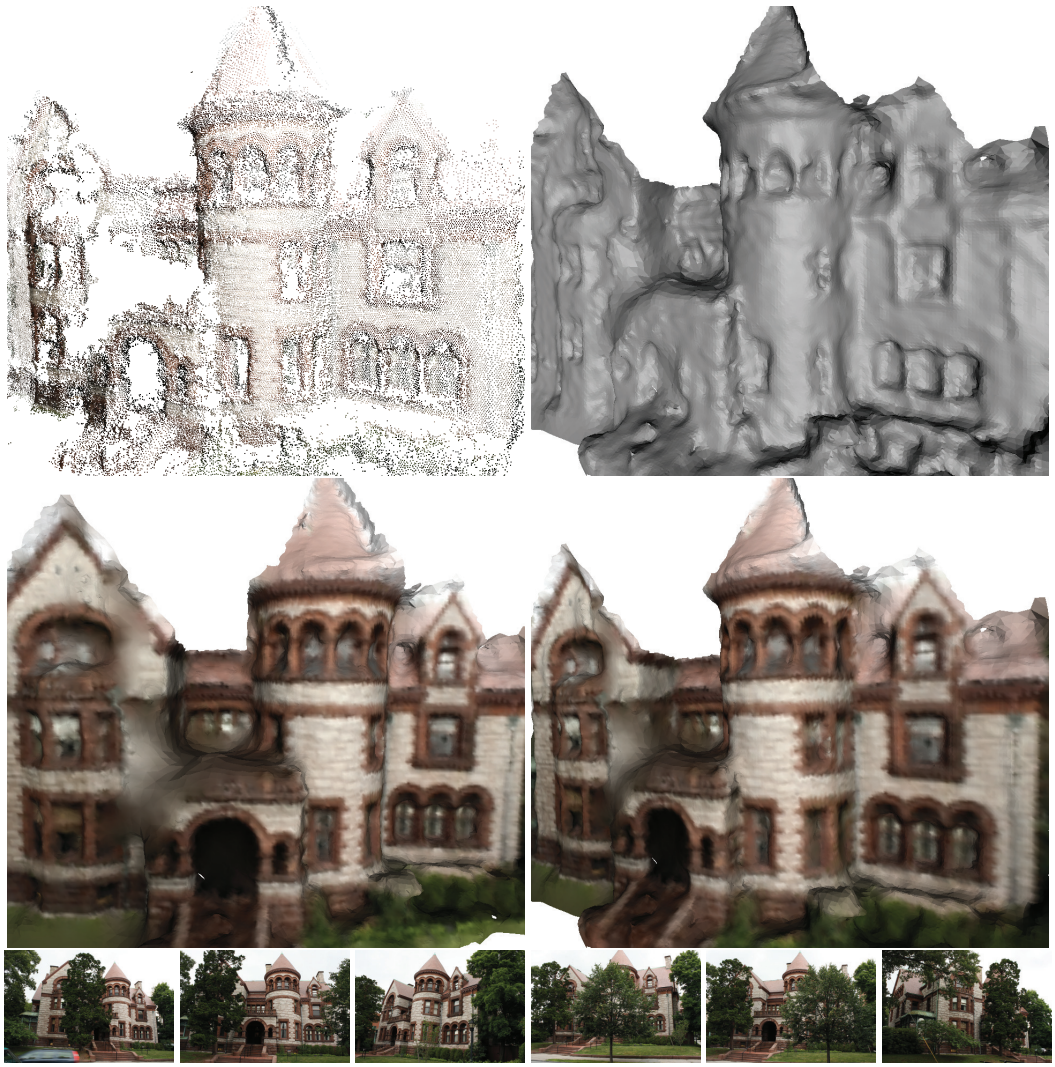

Figure 4: Reconstruction of a building facade. Top row: The input point cloud, and surface geometry reconstructed by the proposed algorithm. Middle row: Two views from the surface and color map reconstructed by the proposed algorithm. Bottom row: 6 examples from the set of 14 images that are used for shape acquisition.

We apply our algorithm to colored oriented point clouds retrieved using the well-known multiview stereo (MVS) pipeline consisting of two publicly available algorithms Bundler [23], and the Patch-based MVS (PMVS) [12]. The goal of multiview stereo is to reconstruct a 3D model from images taken from known camera viewpoints. To learn the viewpoints, we use *Bundler* which takes a set

of images as input and accurately estimates the camera viewpoint per each image. We then use the Patch-based MVS (PMVS) algorithm which takes the same set of images and the estimated camera parameters as input and produces 3D models represented as colored point clouds with accuracy nearly comparable with laser scanners [21]. PMVS is based on the idea of correlating measurements from several images at once to derive globally consistent 3D surface samples. We use a standard digital camera for all the data sets except for the face dataset. In this case, we have to use a set of synchronized cameras as the person could move during data collection procedure. In addition, each camera is calibrated with respect to a common coordinate system.

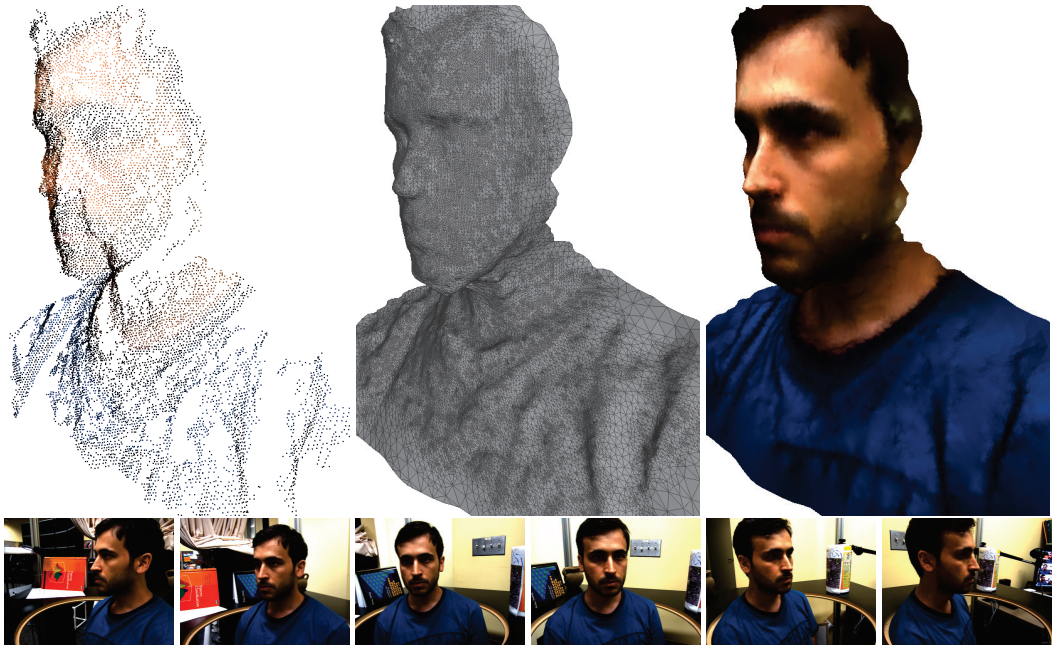

Figure 5: Reconstruction of a human frontal view. Top row: the input point cloud (top-left), surface geometry (top-middle), and color map (top-right) reconstructed by the proposed algorithm. Bottom row: 6 images that are used for shape acquisition.

Our initial test case is the facade of the Applied Mathematics building of Brown University raw dataset of $66,935$ points retrieved from 14 photographs. Each image is captured at resolution 1282×854 with a standard digital camera. The proposed method results in a model consisting of $77,603$ polygons when a maximal octree of depth 10 is used. Fig. 4 shows the reconstruction of the model, and the input point cloud. Our second test case is the frontal view of a human raw dataset of $12,346$ points retrived from 6 photographs.Each image is captured at resolution 1024×768 with a time-synchronized camera system consisting of 6 cameras. The proposed method results in a model consisting of $64,909$ polygons when a maximal octree of depth 9 is used. Fig. 5 shows the reconstruction of the model, and the input point cloud. Our third test case is the brick with Mesopotamian cuneiform inscriptions raw dataset of $333,401$ points retrieved from 21 photographs. Each image is captured at resolution 3072×2304 with a standard digital camera. The proposed method results in a model consisting of $1,034,663$ polygons when a maximal octree of depth 11 is used. Fig. 6 shows the reconstruction of the model, and the input point cloud. Our last test case is the footprint of a human wearing a boot raw dataset of $180,885$ points. This dataset is retrieved from 11 photographs. Each image is captured at resolution 4272×2848. The proposed method results in a model consisting of $897,273$ polygons when a maximal octree of depth 11 is used. Fig. 7 shows the reconstruction of the model, and the input point cloud.

All the experiments are conducted on a PC with a Intel Core 2 Duo 2.26 GHz processor and a 4GB ram. A non-threaded in-core CPU implementation takes 3-15 minutes depending on the maximum depth of the tree that will be used for reconstruction. It consumes %90 of the time for surface reconstruction, and the rest %10 for color map reconstruction. We demonstrate that this method produces watertight surfaces with pleasing smooth color maps even on data sets with various complications:

Figure 6: Reconstruction of a brick with cuneiform, Mesopotamian, 859-840 BCE, clay, Overall 35 x 35 x 11 cm, Williams College Museum of Art, Williamstown, MA, Gift of Professor Edgar J. Banks and Dr. John Henry Haynes, Class of 1876,(20.1.33.A). Top row: the input point cloud (left), and surface geometry (right) reconstructed by the proposed algorithm. Middle row: Two views from surface and color map reconstructed by the proposed algorithm. Bottom row: 6 examples from the set of 21 images that are used for shape acquisition.

non-uniform sampling, sensor noise, and missing regions. In addition, we observe that it is quite in terms of time and memory usage.

8 Conclusion

This report presented a complete solution to the problem of reconstructing a 3D scene's surface and color map from a finite set of colored oriented points. We used an existing variational formulation for the problem of reconstructing a watertight surface defined by an implicit equation. We then extended this by introducing a new variational formulation for the problem of estimating a smooth appearance defined over the surface. We showed that a simple hybrid FE/FD discretization still

Figure 7: Reconstruction of a shoe print. Top row: the input point cloud (left), and surface geometry (right) reconstructed by the proposed algorithm. Middle row: Two views from surface and color map reconstructed by the proposed algorithm. Bottom row: 6 examples from the set of 11 images that are used for shape acquisition.

reduced the problem to the solutions of sparse linear systems. A proof-of-concept implementation has been shown to produce high-quality polygon meshes with smooth color-per-vertex function.

9 Acknowledgements

The authors would like to thank Donald H. Sanders for providing us the images of the cuneiform brick, and the Williams College Museum of Art, Williamstown, MA for granting us permission to use the images. The work described herein was supported by the National Science Foundation under Grant No. IIS-0808718, CCF-0729126, and CCF-0915661.

References

[1] M. Alexa, J. Behr, D. Cohen-Or, S. Fleishman, D. Levin, and C. Silva. Computing and Rendering Point Set Surfaces. *IEEE Transactions on Visualization and Computer Graphics*, pages 3–15, 2003.

[2] P. Alliez, D. Cohen-Steiner, Y. Tong, and M. Desbrun. Voronoi-based variational reconstruction of unoriented point sets. In *Proceedings of the fifth Eurographics symposium on Geometry processing*, pages 39–48. Eurographics Association, 2007.

[3] F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva, and G. Taubin. The Ball-Pivoting Algorithm for Surface Reconstruction. *IEEE Transactions on Visualization and Computer Graphics*, 5(4):349–359, 1999.

[4] J. Boissonnat and F. Cazals. Smooth surface reconstruction via natural neighbour interpolation of distance functions. *Computational Geometry*, 22(1):185–203, 2002.

[5] F. Calakli and G. Taubin. SSD: Smooth Signed Distance Surface Reconstruction. *Computer Graphics Forum*, 30(7), 2011. http://mesh.brown.edu/ssd/.

[6] J. Carr, R. Beatson, J. Cherrie, T. Mitchell, W. Fright, B. McCallum, and T. Evans. Reconstruction and Representation of 3D Objects with Radial Basis Functions. In *Proceedings of ACM Siggraph*, pages 67–76. ACM, 2001.

[7] P. Cignoni, C. Rocchini, and R. Scopigno. Metro: measuring error on simplified surfaces. Technical report, Paris, France, France, 1996.

[8] D. Crispell, D. Lanman, P. Sibley, Y. Zhao, and G. Taubin. Beyond silhouettes: Surface reconstruction using multi-flash photography. In *Third International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT'2006)*, pages 405–412. IEEE, 2006.

[9] B. Curless and M. Levoy. A Volumetric Method for Building Complex Models from Range Images. In *Proceedings of ACM Siggraph*, pages 303–312. ACM, 1996.

[10] T. Dey. *Curve and Surface Reconstruction: Algorithms with Mathematical Analysis*. Cambridge Univ Pr, 2007.

[11] S. Fleishman, D. Cohen-Or, and C. Silva. Robust Moving Least-Squares Fitting With Sharp Features. *ACM Transactions on Graphics (TOG)*, 24(3):544–552, 2005.

[12] Y. Furukawa and J. Ponce. Accurate, dense, and robust multi-view stereopsis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(8), 2009.

[13] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Surface Reconstruction from Unorganized Points. In *Proceedings of ACM Siggraph*, volume 26, pages 71–71, 1992.

[14] M. Kazhdan. Reconstruction of Solid Models from Oriented Point Sets. In *Proceedings of the 3rd. Eurographics symposium on Geometry processing*, page 73. Eurographics Association, 2005.

[15] M. Kazhdan, M. Bolitho, and H. Hoppe. Poisson surface reconstruction. In *Proceedings of the fourth Eurographics Symposium on Geometry Processing*, pages 61–70. Eurographics Association, 2006.

[16] J. Manson, G. Petrova, and S. Schaefer. Streaming Surface Reconstruction Using Wavelets. *Computer Graphics Forum*, 27(5):1411–1420, 2008.

[17] Y. Ohtake, A. Belyaev, M. Alexa, G. Turk, and H. Seidel. Multi-Level Partition of Unity Implicits. *ACM Transactions on Graphics (TOG)*, 22(3):463–470, 2003.

[18] Y. Ohtake, A. Belyaev, and H. Seidel. A Multi-Scale Approach to 3D Scattered Data Interpolation with Compactly Supported Basis Functions. In *International Conference on Shape Modeling and Applications*, pages 153–161, 2003.

[19] S. Schaefer and J. Warren. Dual marching cubes: Primal contouring of dual grids. *Computer Graphics Forum*, 24(2):195–201, 2005.

[20] O. Schall and M. Samozino. Surface from scattered points: A brief survey of recent developments. In B. Falcidieno and N. Magnenat-Thalmann, editors, *1st International Workshop on Semantic Virtual Environments*, pages 138–147, Villars, Switzerland, 2005. MIRALab.

[21] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *Proc. of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 519–528, 2006.

[22] C. Shen, J. O'Brien, and J. Shewchuk. Interpolating and Approximating Implicit Surfaces from Polygon Soup. In *Proceedings of ACM Siggraph*, pages 896–904, 2004.

[23] N. Snavely, S. Seitz, and R. Szeliski. Photo tourism: exploring photo collections in 3d. In *ACM Transactions on Graphics*, volume 25, pages 835–846, 2006.