A Integrated Depth Fusion Algorithm for Multi-View Stereo

Yongjian Xi, Ye Duan

University of Missouri at Columbia

Abstract. In this paper, we propose a new integrated depth fusion algorithm for multi-view stereo. Starting from an embedding space such as the visual hull, we will first conduct robust 3D depth estimation (represented as 3D points) based on image correlation. Next a volumetric saliency weighted normal vector field is constructed from which a watertight 3D surface can be extracted using the graph cut algorithm. Finally an explicit surface evolution will be conducted to recover the finer geometry details of the recovered shape. The experiments on the benchmark datasets show that our algorithm can obtain high quality reconstruction results that are comparable with the state-of-art methods, with considerable less computational time and complexity.

1 Introduction

Despite significant advancement in interactive shape modeling, creating complex high quality realistic looking 3D models from scratch is still a very challenging task. Recent advancement in 3D shape acquisition systems such as laser range scanners and encoded light projecting system have made directly 3D data acquisition feasible [1]. These active 3D acquisition systems however remain expensive. Meanwhile, the price of digital cameras and digital video cameras keeps decreasing while the quality is improving every day, partially due to the intense competition in the huge consumer market. Furthermore, huge amounts of images and videos are added in internet sites such as Google, etc every day, a lot of which could be used for multiview image-based 3D shape reconstruction [2].

To date, there have been a lot of researches conducted in the area of multiview image-based modeling. The recent survey by Seitz et al. [3] gives an excellent review of the state-of-arts in this area. As summarized by [4], most of the existing algorithms follow a two-stage approach: 1) conduct depth estimation based on local groups of input images; 2) fuse the estimated depth values into a global watertight 3D surface estimation. The depth estimation step is often based on image correlation [5]. The main differences between existing algorithms are in the second stage, the data fusion step, which can be divided into two categories. The first type of data fusion reconstructs the 3D surface by conducting volumetric data segmentation using global energy minimization approaches such as graph cut [6, 7, 8, 10, 17, 28], level-set [11, 12, 13, 14, 23], or deformable models [5, 15, 16, 18]. Recently, people have proposed another types of

data fusion algorithms that are based on local surface growing and filtering [2, 9, 20]. Without global optimization, these types of data fusion algorithms can be computationally more efficient [21, 22].

In this paper, we proposed an integrated depth fusion algorithm that integrates the graph-cut based global optimization with a mean-shift based explicit surface evolution. The energy functional to be minimized is derived from a novel volumetric saliency weighted normal vector field based on an anisotropic kernel density estimation. The experiments on the benchmark datasets show that our algorithm can obtain high quality reconstruction results that are comparable with the state-of-art methods, with considerable less computational time and complexity.

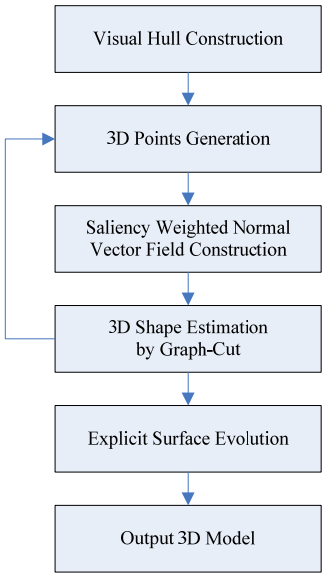


Fig. 1. Flow chart of the algorithm.

1.1 Comparison With Related Works

Our work draws inspirations from a lot of recent advancements in this field. For example, our depth estimation follows the similar pipeline of Hernandez et al. [5] with several modifications to further improve its efficiency (Section 2.2). Unlike [5] which store the depth values into an octree-based volumetric grid, we represent the depth estimations as 3D points thus the accuracy is not restricted by the resolution of the volumetric grid. The graph cut algorithm as well as the divergence based energy functional are originally proposed by Lempitsky et al. [34]. However [34] is focused on the general methodology of surface fitting, where we are developing a new framework for 3D reconstruction from multi-view stereo.

To summarize, the main contributions of this paper are: 1) a novel anisotropic kernel density based point saliency estimation algorithm; 2) a novel volumetric saliency weighted normal vector field; 3) a novel data fusion approach that integrates the global optimal graph cut algorithm with an efficient mean-shift based explicit surface evolution algorithm that can reconstruct the 3D surface with high-quality.

2 Algorithm

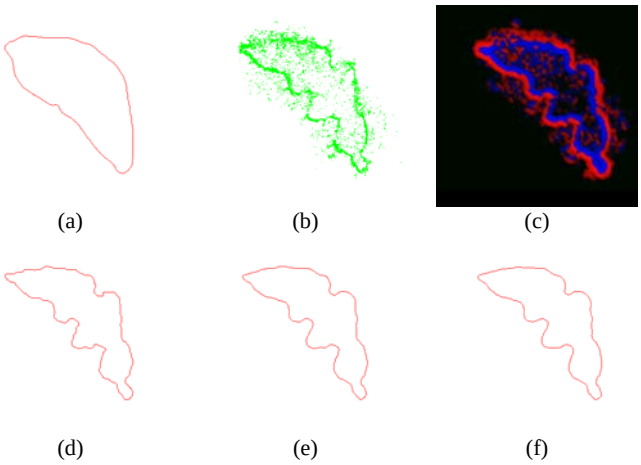


Fig. 2. A 2D slice view of the 3D reconstruction process from the dino ring data [19]. (a) visual hull; (b) points generated by depth estimation; (c) divergence field; (d) shape estimation by graph-cut; (e) shape estimation by graph-cut after another iteration; (f) refined shape estimation by explicit surface evolution.

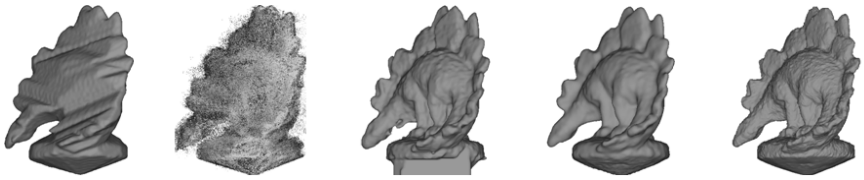


Fig. 3. 3D view of the reconstruction process of the dino ring data [19]. From left to right: visual hull; 3D points generated by depth estimation; shape estimation by graph-cut; shape estimation by graph-cut after another iteration; refined shape estimation by explicit surface evolution.

The entire algorithm consists of five main steps (Fig. 1). Starting from an initial shape estimation such as the visual hull (Step 1), we will conduct depth estimation which will generate a set of 3D points representing the estimated depth (Step 2). A volumetric saliency weighted normal vector field is then constructed based on the 3D points (Step 3), which is then be used to extract a watertight 3D surface by graph cut algorithm (Step

4). Step 2 to Step 4 can be repeated several times. In practice, two to three iterations between Step 2 and Step 4 will be sufficient to obtain a good shape estimation, which will be further improved by the explicit surface evolution step (Step 5). We will discuss each step in details in the following sections. Fig. 2 is a 2D slice view of the reconstruction process of the dino ring dataset of [19]. Fig. 3 shows the intermediate results of the 3D reconstruction process of the dino ring data.

2.1 Visual Hull Construction

The first step of our algorithm is to obtain an initial shape estimation. A bounding box or a visual hull would suffice here. Visual hull is an outer approximation of the observed solid constructed as the intersection of the visual cones associated with all the input cameras [26]. A discrete volumetric representation of the visual hull can be obtained by intersecting the cones generated by back projecting the object silhouettes from different camera views. An explicit shape representation can be obtained by iso-surface extraction algorithms such as Marching Cubes [25].

2.2 3D Points Generation

Once we had an initial explicit shape estimation, we will proceed to 3D depth estimation. First, we need to estimate the visibility of the initial shape with respect to all the cameras. We use OpenGL to render the explicit surface into the image planes of each individual cameras and extract the depth values from the Z-buffer. Given a point on the surface, its visibility with respect to a given camera can then be decided by comparing its projected depth value into the image plane of the given camera with the corresponding depth value stored in the Z-buffer.

Our depth estimation is based on the Lambertian assumption, i.e. if a point belongs to the object surface, its corresponding 2D patches in the image planes of its visible cameras should be strongly correlated. Hence starting from a point on the object surface, we can conduct a line search along a defined search direction to locate the best position whose correlation between the corresponding 2D image patches of different visible cameras is the maxima within a certain search range. This idea is first proposed by [5]. Our paper follows the same principle with several modifications. In the following, we will briefly describe our depth estimation method as well as the main differences between our method and the method of [5].

Given a point on the initial surface, we will select a set of (up to) five “best-view” visible cameras based on the point’s estimated surface normal. Each camera in the selected set will serve as the main camera for once. The search direction is defined as the optical ray passing through the optical center of the main camera and the given point. We will uniformly sample the optical ray within a certain range of the given point, and for each sampled position, we will project it into the image planes of the main camera and another camera in the set, respectively. Rectangular image patches centered at the projected locations of the two image planes will be extracted, and the correlation between the two image patches will be computed by similarity measures such as the normalized cross-correlation (NCC) [5].

For a set of five "best-view" cameras, a total of 20 correlation curves will be generated. For each of the correlation curve, the best position (i.e. the point with the highest correlation value) will be selected as the depth estimation. The depth estimations will be represented as 3D points, which will be processed further to construct a new shape estimation of the object.

The main differences between our implementation and the method of [5] are: first, we start the line search from every point on the explicit object surface. The line search in [5] is initiated from every image and the correlation is computed with all the other images, which could be computationally more expensive than ours. Secondly, in [5], for each set of correlation curves computed using the same search direction and the same main camera, only one representative depth estimation is used. While in our method, we avoid this potentially premature averaging by using the depth estimations from all the correlation curves, and postpone the outlier pruning into the subsequent outlier removal step. Thirdly, in [5], the depth estimations are stored in an octree-based volumetric grid, while we store them as discrete points whose accuracy are not restricted by the grid size.

2.3 Saliency Weighted Normal Vector Field Construction

There are two main steps in the data fusion phase. First, a saliency weighted normal vector field is constructed based on the 3D points generated by the above depth estimation step. Next, a watertight 3D surface is extracted from the saliency weighted normal vector field by energy minimization. The saliency weighted normal vector field is constructed by the following three steps: 1) saliency field construction by anisotropic kernel density estimation; 2) normal estimation and consistent normal orientation propagation; 3) volumetric saliency weighted normal vector field construction.

We will detail the three steps of saliency weighted vector field construction in this section. The 3D surface extraction will be discussed in the later sections.

2.3.1 Saliency Field Construction By Anisotropic Kernel Density Estimation

We use the term saliency to represent the likelihood the unknown surface passes through a certain part of 3D space. In this paper, we propose to employ parzen-window based nonparametric density estimation method to compute the saliency of each point.

Given n data points x_i, $i = 1, \ldots, n$ in the d-dimensional Euclidean space R^d, the multivariate kernel density estimate obtained with kernel $k(x)$ and window radius h (without loss of generality, let's assume $h = 1$ from now on), computed in the point x is defined as:

$$f(x) = \frac{C_{k,d}}{n} \sum_{i=1}^{n} k(\| x - x_i \|^2) \quad \textbf{(1)}$$

$C_{k,d}$ is the normalizing constant, $||x||$ is the L_2 norm (i.e. Euclidean distance metric) of the d-dimensional vector x. There are three types of commonly used spherical kernel functions $k(x)$: the Epanechnikov kernel, the uniform kernel, and the Gaussian kernel [27].

For 3D point cloud obtained by depth estimation, the outliers tend to spread in the space randomly, while "real" (we use a quotation here to emphasize the fact that the real surface is unknown) surface points will spread along a thin shell which encloses the real surface object. In other words, the distribution of the outliers is relatively isotropic, while the distribution of the real surface points is rather anisotropic. Hence in this paper, we propose to employ an anisotropic ellipsoidal kernel based density estimation method. More specifically, for anisotropic kernel, the L_2 norm $||x-x_i||$ in Equation (1), which measures the Euclidean distance metric between two points x and x_i will be replaced by the Mahalanobis distance metric $||x-x_i||_M$:

$$|| x - x_i ||_M = ((x - x_i)^t H^{-1} (x - x_i))^{1/2} \quad (2)$$

here H is the covariance matrix defined as:

$$H = DD^T \quad (3)$$

and

$$D = (x_1 - x, x_2 - x, \ldots, x_n - x) \quad (4)$$

Geometrically, $(x - x_i)^t H^{-1} (x - x_i) = 1$ is a three-dimensional ellipsoid centered at x, with its shape and orientation defined by H. Using Single Value Decomposition (SVD), the covariance matrix H can be further decomposed as:

$$H = UAU^T \quad (5)$$

with

$$A = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix} \quad (6)$$

$\lambda_1 \geq \lambda_2 \geq \lambda_3$ are the three eigenvalues of the matrix H, and U is an orthonormal matrix whose columns are the eigenvectors of matrix H.

To compute the anisotropic kernel based density, we will apply an ellipsoidal kernel E of equal size and shape on all the data points. The orientation of the ellipsoidal kernel E will be determined locally. More specifically, given a point x, we will calculate its covariance matrix H by points located in its local spherical neighborhood of a fixed radius. (Without loss of generality, we will assume the radius is 1, which can be done by normalizing the data by the radius). The U matrix of Equation (5) calculated by the covariance analysis is kept unchanged to maintain the orientation of the ellipsoid. The

size and shape of the ellipsoid will be modified to be the same as the ellipsoidal kernel E by modifying the diagonal matrix A as:

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & r \end{bmatrix} \tag{7}$$

r is half of the length of the minimum axis of the ellipsoidal kernel E.

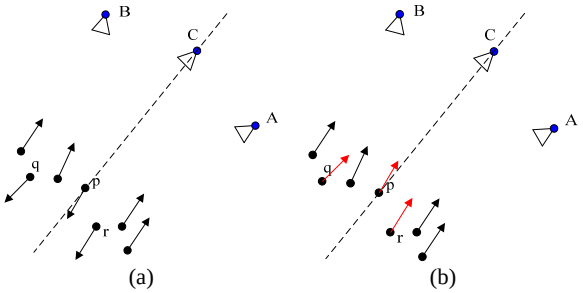


Fig. 4. Camera view direction guided normal orientation propagation. (a) 3D points generated by the depth estimation based on the image correlations between the main camera C and the other adjacent cameras. Each point is shown with its normal vector estimated by the PCA algorithm. The orientation of the normal vectors at point p, q and r are not correct. (b) Based on the view direction of the main camera C, the orientation of the normal vectors at point p, q and r are reversed so that they are now opposite to the view direction of camera C.

2.3.2 Normal Estimation and Consistent Normal Orientation Propagation

Given the 3D point clouds, we can estimate the normal vector at each point based on the Principle Component Analysis (PCA) algorithm [32]. Normal vectors estimated by the PCA algorithm however has an ambiguity of 180 degree so might not be consistently oriented. An orientation propagation is often needed to ensure the consistent orientation of the normal vectors. One way to do this is to first build a graph with each point as a node and the weights of edges between the adjacent points are defined as $1 - \|n_1 \cdot n_2\|$, where n_1 and n_2 are the normal vectors of the two adjacent points, and then compute the minimum spanning tree (MST) from the graph using algorithms such as the Kruskal's algorithm [33] which finds a subset of the edges that forms a tree that includes every vertex in the graph, where the total weight of all the edges in the tree is minimized. At the termination of the algorithm, the normals are adjusted so the two neighbors in the tree have consistent normal orientation.

The above MST based normal orientation propagation approach however is not robust against noises and outliers. In this paper, we propose to utilize external knowledge to guide the normal orientation propagation (Fig. 4). Particularly, since the point clouds are generated by depth estimation based on the image correlations between

different cameras, for a given point *p*, it should be visible to its main camera in the depth estimation step (Section 2.2), i.e. the dot product between the normal vector of the point *p* and the view direction of its main camera should be negative. If not we will reverse the normal orientation at this point.

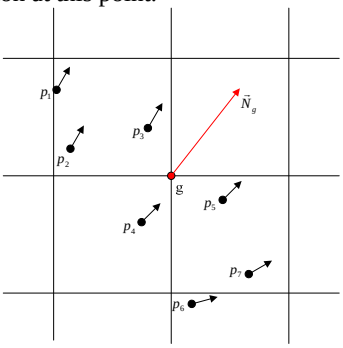


Fig. 5. Construction of the volumetric saliency weighted normal vector field. The normal vector at grid node g is calculated as the weighted sum of the normal vectors of its adjacent points $p_1, p_2, \ldots, p_7$ (Section 2.3.3).

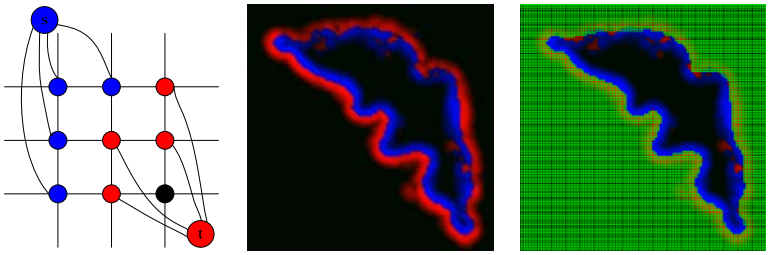
2.3.3 Volumetric Saliency Weighted Normal Vector Field Construction

Once we have estimated the saliency and normal vector at each point, we will proceed to construct a volumetric saliency weighted normal vector field, from which a watertight 3D surface can be extracted by energy minimization. A volumetric grid embedding all the 3D points is first constructed. The saliency and the normal vector of each point are then propagated to its adjacent grid nodes (Fig. 5). Specifically, the saliency S_g of a grid node g is computed as the weighted summation of the saliency S_{p_i} of its adjacent points p_i:

$$S_g = \sum_i C_{p_i} S_{p_i} \tag{8}$$

The weight C_{p_i} is calculated using the aforementioned parzen-window kernel function (Equation (**1**)) based on the anisotropic Mahalanobis distance (Equation (**2**)) between the grid node g and point p_i. Since the kernel we used (e.g. truncated Gaussian kernel) has finite support here, only a finite number of points p_i ($i = 1, \ldots, n$) within the kernel radius has non-zero weights C_{p_i}. The normal vector $\vec{N}_g$ at grid node g is calculated similarly as the weighted summation of the normal vector $\vec{N}_{p_i}$ of its adjacent points p_i:

$$\vec{N}_g = \sum_i C_{p_i} S_{p_i} \vec{N}_{p_i} \tag{9}$$



(a) (b) (c)

Fig. 6. Shape estimation by graph cut. (a) Graph construction for minimization of Equation(**14**): neighboring nodes are connected via *n*-links representing regularization cost. Nodes are also connected to the terminals vial *t*-links based on their divergence value: blue nodes have positive divergence and are connected to the source terminal *s*; red nodes have negative divergence and are connected to the sink terminal *t*; the green node has zero divergence and is not connected to either terminal. (b) A 2D slice of the divergence field for the dino ring dataset of [19]. (c) A 2D slice of the 3D surface (shown as non-green pixels) estimated from (b) by graph cut algorithm.

2.4 3D Shape Estimation by Graph-Cut

Once the volumetric saliency weighted normal vector field is constructed, a water-tight 3D surface *S* can be extracted by energy minimization. We use the following energy functional as suggested by [34]:

$$E = E_{data} + \varepsilon E_{reg} \tag{10}$$

E_{data} is the data alignment term which is the inverse of the flux that enforces the surface alignment with the data orientation:

$$E_{data} = -flux(S) = -\int_S \langle N, v \rangle ds \tag{11}$$

where <,> is (Euclidean) dot product and *N* is unit normal to surface elements *ds* consistent with a given orientation. If vectors *v* is interpreted as a local speed in a stream of water then the absolute value of flux equals the volume of water passing through the hypersurface in a unit of time. The sign of flux will be determined by the orientation of the surface. E_{reg} is the area-based regularization term that maintains the regularity of the extracted surface:

$$E_{reg} = \int_S ds \tag{12}$$

ε is the coefficient of the regularization term E_{reg} that controls the strength of the smoothness in the energy minimization process and is related to the sampling density of the data. In our experiment, we set ε as 0.2.

As pointed out by [34], combining flux with area based regularization can overcome the shrinking effect of the area-based regularization and improve the reconstruction of elongated structures, narrow protrusions, and other fine details. Based on the divergence theorem for differentiable vector fields, the integral of flux of vector field over surface *S* equals to the integral of vector field's divergence *div(v)* in the interior of *S*:

$$\int_S \langle N, v \rangle ds = \int_V div(v_p) \cdot dp \tag{13}$$

Where *V* is the region enclosed inside *S*. Thus *E* of Equation **(11)** is now:

$$E = \varepsilon \int_S ds - \int_V div(v_p) \cdot dp \tag{14}$$

Equation (**14**) can be solved efficiently using the graph cut algorithm [34]. A typical graph construction is shown in Fig. 6(a): neighboring nodes are connected via *n*-links representing area-based regularization cost. Nodes are also connected to the terminals via one *t*-link based on their divergence value: blue nodes have positive divergence and are connected to the source terminal *s* with weight $div(v_p)$; red nodes have negative divergence and are connected to the sink terminal *t* with weight $-div(v_p)$; the black node has zero divergence and is not connected to either terminal. The weight of the *n*-link is defined as the inverse of the edge length so that the weights of severed *n*-links approximate the surface area [35]. Consequently, a global minimum surface for Equation (**14**) can be found by computing a minimal *s*/*t*-cut in the constructed graph [34].

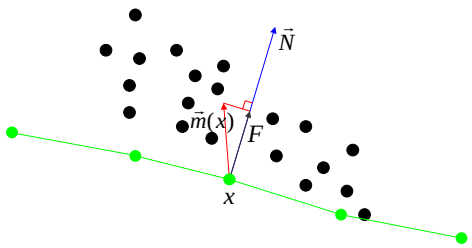


Fig. 7. Explicit surface evolution. The surface (shown in green) will move along its normal vector with speed function *F* which is the inner product between the mean shift vector $\vec{m}(x)$ and the surface normal vector.

2.5 Explicit Surface Evolution

The shape estimated by the graph cut algorithm can be further improved by an explicit surface evolution step guided by the following partial differential equation (PDE):

$$\begin{cases} \dfrac{\partial S}{\partial t} = F\vec{N} \\ F(x) = \vec{m}(x) \cdot \vec{N} \\ \vec{m}(x) = \dfrac{\sum_{i=1}^{n} g(\| x - x_i \|^2) x_i}{\sum_{i=1}^{n} g(\| x - x_i \|^2)} - x \\ g(x) = k'(x) \end{cases} \quad (15)$$

Here S is the 3D surface, t is the time parameter, F is the speed function. $\vec{N}$ is the surface normal vector. $\vec{m}(x)$ is mean shift vector proposed in [27] and is proportional to $\nabla f(x)$, the gradient of the kernel density function $f(x)$ of Equation (1), i.e. the mean shift vector $\vec{m}(x)$ is in the gradient ascent direction and will always point to the direction of the maximal increase in the kernel density $f(x)$. Since the surface evolves along its normal direction (Fig. 7), the speed function F is defined as the inner product between the mean shift vector $\vec{m}(x)$ and the normal vector $\vec{N}$. Note that the speed function F is dynamically calculated at each time step, thus the surface evolution is not be limited by the grid resolution.

Table 1. Running time and reconstruction accuracy.

Dataset	Running time (mins: secs)	# of input images	accuracy	completeness
Temple ring	33:50	47	99.5%	0.53
Temple sparse ring	29:15	16	96.8%	0.72
Dino ring	34:10	48	99.5%	0.46
Dino sparse ring	30:50	16	97.8%	0.42

3 Benchmark Data Evaluation

We had applied our algorithm to the four benchmark datasets: temple ring, temple sparse ring, dino ring, and dino sparse ring from [19]. Table 1 shows the running time and the reconstruction accuracy and completeness obtained from the evaluation site [19]. The running time is based on a Pentium D Desktop PC with CPU 2.66GHz, 2GB

RAM. The graph cut is conducted by using the generic max-flow library implemented by Boykov and Kolmogorov [37].

Fig. 8 are the 3D rendering of our final reconstruction results copied from the evaluation website. Our result is listed under the name “ECCV_216”.

Fig. 8. Reconstruction results of the four Benchmark datasets of [19]. From top to bottom: dino sparse ring, dino ring, temple sparse ring, and temple ring.

4 Conclusion and Future Work

In this paper, we proposed: 1) a novel anisotropic kernel density based point saliency estimation algorithm; 2) a novel volumetric saliency weighted normal vector field; 3) a novel data fusion approach that integrates the global optimal graph cut algorithm with an efficient mean-shift based explicit surface evolution algorithm that can reconstruct the 3D surface with high-quality. The benchmark evaluation of our algorithm is among the top in the evaluation site, with considerable less computational time and complexity.

Currently, our method utilizes the visual hull for initial estimation. In the future we would like to relax this requirement. We will explore the use of bounding boxes and/or feature points as an initialization as were suggested by Furukawa et al. [36] and Quan et al. [24].

5 Acknowledgement

We are very grateful for Steve Seitz et al [3] for providing us the datasets used in the paper and Daniel Scharstein for helping us evaluating the result on the benchmark datasets. We are very thankful for Boykov and Kolmogorov for making their implementation of the max-flow publicly availble [37]. We are very grateful to the support of the Mathematical Sciences Center of Tsinghua University during YD's visit summer 2010.

References

[1] Yang Wang, Xiaolei Huang, Chan-Su Lee, Song Zhangand Zhiguo Li, Dimitris Samaras, Dimitris Metaxas, Ahmed Elgammal, and Peisen Huang. High resolution acquisition, learning and transfer of dynamic 3D facial expressions. Computer Graphics Forum, 23(3):677–686, 2004.

[2] Michael Goesele, Noah Snavely, Brian Curless, Hugues Hoppe, Steven M. Seitz. Multi-View Stereo for Community Photo Collections, Proceedings of ICCV 2007, Rio de Janeiro, Brasil, October 14-20, 2007.

[3] Steve Seitz, Brian Curless, James Diebel, Daniel Scharstein, Richard Szeliski. A Comparison and Evaluation of Multi-View Stereo Reconstruction Algorithms, CVPR 2006, vol. 1, pages 519-526.

[4] Neill Campbell, George Vogiatzis, Carlos Hernández and Roberto Cipolla. Using multiple hypotheses to improve depth-maps for multi-view stereo. ECCV 2008, pages 766-779.

[5] C. Hernandez and F. Schmitt. Silhouette and stereo fusion for 3D object modeling. CVIU, 96(3):367-392, 2004.

[6] G. Vogiatzis, C. Hernandez, P.H.S. Torr, R. Cipolla. Multi-view stereo via volumetric graph-cuts and occlusion robust photo-consistency. IEEE Trans. Pattern Anal. Mach. Intell. 29(12) (2007).

[7] M. Goesele, B. Curless, and S. Seitz. Multi-view stereo revisited. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition. (2006).

[8] A.Hornung, L. Kobbelt. Hierarchical volumetric multi-view stereo reconstruction of manifold surfaces based on dual graph embedding. In: Proc. IEEE Conf. On Computer Vision and Pattern Recognition. (2006).

[9] Y. Furukawa, J. Pons. Accurate, dense, and robust multi-view stereopsis. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition. (2007).

[10] G. Vogiatzis, P. Torr, R. Cipolla. Multi-view stereo via volumetric graph-cuts. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition. (2005).

[11] H. Jin, S. Soatto, and A. Yezzi. Multi-view stereo reconstruction of dense shape and complex appearance. IJCV, 63(3):175-189, 2005.

[12] O. Faugeras and R. Keriven. Variational principles, surface evolution, PDE's, level set methods and the stereo problem. IEEE Trans. on Image Processing, 7(3):336-344, 1998.

[13] S. Soatto, A. Yezzi, and H. Jin. Tales of shape and radiance in multiview stereo. In ICCV, pp. 974-981, 2003.

[14] H. Jin, S. Soatto, and A. Yezzi. Multi-view stereo beyond lambert. In CVPR, vol. 1, pp. 171-178, 2003.

[15] Y. Duan, L. Yang, H. Qin, and D. Samaras. Shape reconstruction from 3D and 2D data using PDE-based deformable surfaces. In ECCV, vol. 3, pp. 238-251, 2004.

[16] C. Hernandez, F. Schmitt. Multi-stereo 3D object reconstruction, Proceedings of 3D Data Processing Visualization and Transmission, 2002. 159- 166.
[17] S. Sinha and M. Pollefeys. Multi-view reconstruction using photo-consistency and exact silhouette constraints: A maximum-flow formulation. In ICCV, pp. 349-356, 2005.
[18] Yasutaka Furukawa and Jean Ponce. Carved Visual Hulls for Image-Based Modeling, volume 1, pp. 564-577. ECCV 2006.
[19] The multi-view stereo evaluation web site at http://vision.middlebury.edu/mview/.
[20] M. Habbecke, L. Kobbelt. A surface-growing approach to multi-view stereo reconstruction. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (2007).
[21] P. Merrell, A. Akbarzadeh, L. Wang, P. Mordohai, J-M. Frahm, R. Yang, D. Nister, M. Pollefeys. Real-time visibility-based fusion of depth maps. In: Proc. 11th Intl. Conf. on Computer Vision. (2007).
[22] D. Bradley, T. Boubekeur, W. Heidrich. Accurate multi-view reconstruction using robust binocular stereo and surface meshing. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition. (2008).
[23] Maxime Lhuillier, Long Quan. A Quasi-Dense Approach to Surface Reconstruction from Uncalibrated Images. IEEE Trans. Pattern Anal. Mach. Intell. 27(3): 418-433 (2005).
[24] Long Quan, Jingdong Wang, Ping Tan, Lu Yuan. Image-Based Modeling by Joint Segmentation. International Journal of Computer Vision 75(1): 135-150 (2007)
[25] William E. Lorensen, Harvey E. Cline. Marching Cubes: A high resolution 3D surface construction algorithm. In: Computer Graphics, Vol. 21, Nr. 4, July 1987.
[26] A. Laurentini. The visual hull concept for silhouette-based image understanding. IEEE Trans. Pattern Analysis and Machine Intelligence. Pages: 150-162, 1994.
[27] Comaniciu, Dorin, Peter Meer. Mean Shift: A Robust Approach Toward Feature Space Analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence. 24 (5): 603–619. 2002.
[28] V. Kolmogorov and R. Zabih. Multi-camera scene reconstruction via graph cuts. In ECCV, vol. III, pp. 82-96, 2002.
[29] Hong-Kai Zhao, S. Osher, R. Fedkiw. Fast surface reconstruction using the level set method, Proceedings of IEEE Workshop on Variational and Level Set Methods in Computer Vision, 2001. 194-201.
[30] Hong-Kai Zhao, S. Osher, B. Merriman, and M. Kang. Implicit and non-parametric shape reconstruction from unorganized points using variational level set method. Computer Vision and Image Understanding, 80(3):295–319, 2000.
[31] Vicent Caselles, Ron Kimmel, Guillermo Sapiro, Catalina Sbert. Three Dimensional Object Modeling via Minimal Surfaces. ECCV (1) 1996: 97-106.
[32] Hugues Hoppe, Tony DeRose, Tom Duchamp, John Alan McDonald, Werner Stuetzle: Surface reconstruction from unorganized points. SIGGRAPH 1992: 71-78.
[33] Joseph. B. Kruskal: On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem. In: Proceedings of the American Mathematical Society, Vol 7, No. 1 (Feb, 1956), pp. 48–50.
[34] Victor S. Lempitsky, Yuri Boykov: Global Optimization for Shape Fitting. CVPR 2007.
[35] Yuri Boykov, Vladimir Kolmogorov: Computing Geodesics and Minimal Surfaces via Graph Cuts. ICCV 2003: 26-33.
[36] Yasutaka Furukawa, Jean Ponce: Accurate, Dense, and Robust Multi-View Stereopsis. CVPR 2007.
[37] Yuri Boykov, Vladimir Kolmogorov: An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Vision, In IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 26, no. 9, pp. 1124-1137, Sept. 2004.