to appear in Proc. ICCV'07

Multi-View Stereo via Graph Cuts on the Dual of an Adaptive Tetrahedral Mesh

Sudipta N. Sinha Philippos Mordohai Marc Pollefeys
Department of Computer Science, UNC Chapel Hill, USA

Abstract

We formulate multi-view 3D shape reconstruction as the computation of a minimum cut on the dual graph of a semi-regular, multi-resolution, tetrahedral mesh. Our method does not assume that the surface lies within a finite band around the visual hull or any other base surface. Instead, it uses photo-consistency to guide the adaptive subdivision of a coarse mesh of the bounding volume. This generates a multi-resolution volumetric mesh that is densely tesselated in the parts likely to contain the unknown surface. The graph-cut on the dual graph of this tetrahedral mesh produces a minimum cut corresponding to a triangulated surface that minimizes a global surface cost functional. Our method makes no assumptions about topology and can recover deep concavities when enough cameras observe them. Our formulation also allows silhouette constraints to be enforced during the graph-cut step to counter its inherent bias for producing minimal surfaces. Local shape refinement via surface deformation is used to recover details in the reconstructed surface. Reconstructions of the Multi-View Stereo Evaluation benchmark datasets and other real datasets show the effectiveness of our method.

1. Introduction

We address multi-view reconstruction from a set of calibrated images utilizing both photometric and silhouette information. Several high-quality reconstruction approaches have been recently proposed [14] and have participated in the Multi-View Stereo Evaluation (http://vision.middlebury.edu/mview). Multi-view reconstruction has been formulated as a variational problem and techniques such as level sets and graph cuts have been used to recover a surface that minimizes a surface cost functional regularized using smoothness priors. The main issues that are investigated by methods such as [3, 4, 15] are (1) how to enforce photo-consistency and silhouette constraints, which are complementary and (2) how to estimate visibility. We propose a novel way to address the first question and employ a robust technique for computing photo-consistency that does not require accurate visibility estimation.

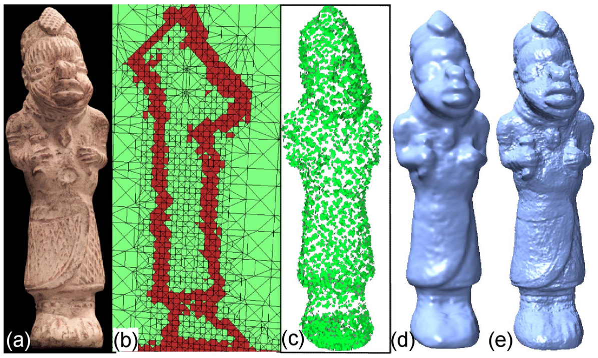

Figure 1. Overview (*statue1* dataset)(a) One of 36 input images. (b) A slice through the adaptive tetrahedral mesh showing the photo-consistent region in red (dark). (c) Quasi-dense patches produced during mesh refinement. (d) The 3D model obtained from graph-cut optimization. (e) The final refined 3D model.

Problems whose solution is a manifold of co-dimension one, such as surface extraction in a volume, can be solved using graph cuts. A formulation often used for 3D reconstruction is to embed a graph in a volume containing the surface and estimate the surface as a cut separating free-space (exterior) from the interior of the object or objects. Vogiatzis et al. [17] were the first to present a graph construction technique for this problem. Their method and those of [3, 6, 16, 19] rely on the silhouettes for determining an exterior and an interior bound for the surface. Photo-consistency is then estimated for the nodes of the geometric graph embedded between the two base surfaces.

While these methods have shown impressive results, the base surfaces impose hard constraints on the topology of the reconstructed objects. Deep concavities, holes and separations not present in the silhouettes cannot be recovered. To circumvent these limitations, we propose a novel method for constructing the graph that is guided by photo-consistency. We adaptively subdivide a coarse tetrahedral mesh to densely sample the photo-consistent parts of the volume. The adaptive subdivision is crucial for achieving high-quality reconstructions since the memory and computation requirements for operating on a uniform 3D grid become prohibitive as the resolution increases. Hornung and Kobbelt's multi-resolution method [6] achieves high-resolution in the region of interest, but it relies heavily on the visual hull being a good approximation of the surface. Since our approach does not require a base surface, it can reconstruct deep concavities and shapes whose visual hulls have a different topology.

An important aspect of multi-view reconstruction is visibility estimation. Lempitsky et al. [9] argue that a local estimation based on surface orientation may suffice instead of attempting to estimate long-range occlusions as the shape estimate used for this can contain errors. Conceding that neither local or global methods are able to estimate visibility reliably, we use a robust matching cost in multiple views that ignores least photo-consistent pairs of image patches, treating occlusion as the source of outliers. Regions that are not photo-consistent are labeled as either interior, exterior or undecided using a voting-based scheme.

After an initial surface is estimated as the minimum cut of the dual graph embedded in this semi-regular tetrahedral mesh, we impose silhouette constraints and obtain the final surface by a second minimum cut on the same graph. These silhouette constraints help to overcome the bias for minimal surfaces which are preferred by graph-cuts. Local shape refinement is then performed along the lines of [3, 4] to capture fine details on the surface.

1.1. Related work

Multi-view shape reconstruction has been approached from various angles by the computer vision community. We refer readers to the recent survey by Seitz et al. [14] and only review methods closely related to ours here.

Lhuillier and Quan [10] detect a quasi-dense set of reliable 3D points and reconstruct the surface within a variational framework. The solution is computed by a level set implementation that takes into account 3D points, silhouettes and the images. Pons et al. [12] adopt a level set approach that evolves the shape in order to minimize image prediction error. A multi-resolution scheme is used to escape local minima. Two approaches based on local graph-cut optimization of surface patches combined with space carving and patch growing are presented by Zeng et al. [20].

The first approach based on graph cuts was presented by Kolmogorov and Zabih [8] who use a labeling graph that also encodes visibility constraints. A volumetric graph-cut stereo approach was presented by Vogiatzis et al. [17] who use the visual hull and an inwards offset surface as the source and sink of a graph. The desired surface is the cut that separates the two terminal while maximizing surface photo-consistency. Yu et al. [19] proposed a similar method that operates on the surface distance grid to reduce the minimal surface bias and metrication errors. To counter the bias for minimal surfaces a ballooning term favoring larger volume was proposed by [17]. A better visibility based 'intelligent' ballooning term was recently proposed by [5].

Graph cuts on CW-complexes (duals of meshes) were first used by [7] for optimizing surface functionals. Later globally optimal methods for volumetric stereo were proposed [2, 9]; these did not require initialization via the visual hull. However these methods used a uniform CW-complex which had a prohibitive memory cost limiting them to complexes with coarser resolutions. The CW-complex reduces metrication errors by providing more orientations than those in a voxel grid and a graph-cut on it yields a manifold mesh. Hornung and Kobbelt [6] proposed a multi-resolution approach based on complexes (dual graph embedding of a uniform grid) to address the resolution issue but their method relied on a good visual hull for initialization. In contrast our globally optimal method for volumetric stereo via graph-cuts neither uses the visual hull for initialization nor is limited by the high memory costs because it operates on an adaptive CW-complex.

Many of the above methods use silhouettes only for initialization. They do not enforce silhouette constraints during the main optimization. Hernández and Schmitt [4] combine silhouette and photo-consistency constraints that act as forces on a deformable model representation of the surface. To overcome sensitivity to local minima, gradient vector flow for the texture-driven force is computed in an octree. Tran and Davis [16] initialize their graph using two base surfaces, with the visual hull being the exterior one. They show ways to modify the graph to enforce the inclusion of protrusions and, after a first cut is computed, to pursue concavities. Sinha and Pollefeys [15] introduce a novel geometric graph construction that guarantees exact enforcement of silhouette constraints in a single step but cannot handle complex geometry and topology. Furukawa and Ponce [3] first constrain the rims using dynamic programming followed by iterative graph cut with the rims fixed. Fine geometric details are recovered in a final refinement stage where texture, silhouette-driven and smoothness forces deform the mesh representing the surface as in [4, 5].

Visibility estimation is a critical aspect in multi-view stereo. Several authors [3, 12] use the current estimate of the shape to compute the exact visibility of all points. A simpler approach is to use the initial shape, typically the visual hull, and a restriction on the angle between the optical axes of the cameras and the surface normal for this purpose [17, 6, 19]. Only the latter factor was taken into account by [9]. We opt for a robust approach under which we select the best subset of cameras for each point as in [4].

2. Key Concepts

A major strength of our method is that it does not rely on the visual hull to construct a base surface. As a result, it is able to handle changes in topology, unlike most global methods reviewed above with the exception of [9]. A second advantage is that unlike previous CW-complex based methods, the resolution of our volumetric mesh adapts according to photo-consistency and is finer at places where surfaces are more likely to exist. For textured surfaces, this provides huge memory savings as photo-consistency bands in the volume are fairly thin. Textureless surfaces however tend to create larger zones of photo-consistency; such regions must be finely sampled in our mesh. High resolution is critical for the quality of the reconstruction and our method aims at maximizing sampling density where needed. Along the lines of [6, 9] the presence of 12 different oriented faces in the mesh (as opposed to 6 in a uniform grid) reduces the discretization of the cut surface.

We employ a robust scheme for photo-consistency estimation that selects the most photo-consistent camera pairs to compute the score of a patch. Visibility does not need to be known, since potentially occluded cameras would not be included in the most photo-consistent pairs. We also present a method similar to [5] for determining the likelihood of a cell in the mesh being interior or exterior to the surface utilizing the known visibility of a few surface patches which have been found to be photo-consistent.

2.1. Graph-cut on dual of a volumetric mesh

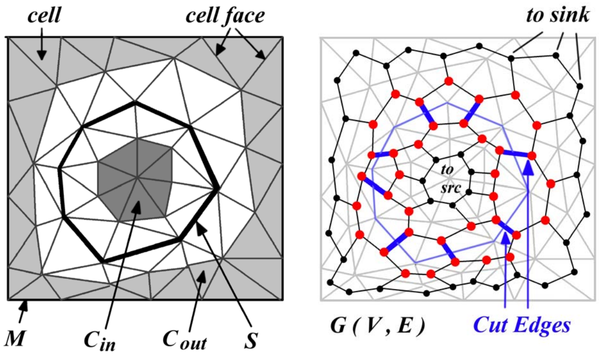

Figure 2. 2D illustration of the graph-cut formulation on the dual graph G of a volumetric mesh M (C_{in} and C_{out} are interior and exterior cells respectively). The value of a cut on G is equal to the cost of a surface S embedded within M.

Let us assume that we are given a volumetric mesh M of the bounding volume with its set of cells and faces denoted by C and F respectively and that some of its cells have been labeled as interior and exterior to the unknown surface. The surface reconstruction problem can then be formulated as finding the most photo-consistent surface embedded within M, which separates the set of interior cells C_{in} from the exterior ones denoted by C_{out}. This can be achieved by minimizing a surface cost functional $\int_S \phi(s)\, ds$, where $\phi(s)$ represents the image discrepancy of an infinitesimal area ds of the unknown surface. In the discrete case, the energy functional becomes $\sum_S \phi(s)$ where S is a set of polygonal faces constituting a candidate surface. The discrete optimization can be formulated as a binary graph-cut problem [1, 9] on the dual graph of M denoted by $G(V, E)$. See Fig. 2 for a 2D illustration. The vertices in V are dual to the cells of M while the directed edges in E are dual to the oriented cell boundaries (faces) of M. The capacity of an edge in E can be derived from the photo-consistency cost of its dual polygonal face. The vertices in V representing cells in C_{in} and C_{out} are connected to the source and sink vertices in the flow graph using edges with infinite capacities. The minimum cut on G can be computed in low-order polynomial time and corresponds to a surface which gives a global minimum of the surface cost functional.

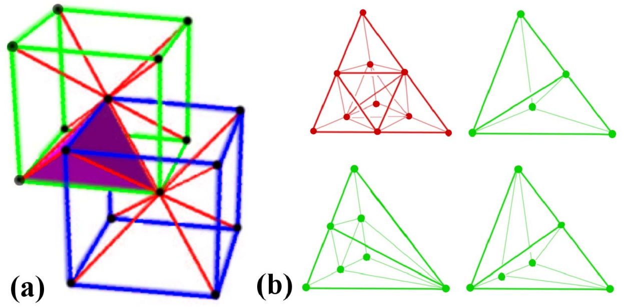

Figure 3. (a) Tetrahedral cell in a BCC lattice (two interlaced grids with diagonal edges added). (b) (Top-left) Red-Refinement (1:8) subdivision. (Rest) Green Refinement (1:2, 1:4) subdivision).

We choose M to be a tetrahedral mesh, motivated by their popularity in mesh-generation [11] and the fact that a minimum cut on its dual graph produces a triangulated surface. The rest of the paper describes (a) how to build a suitable tetrahedral mesh M and (b) how to use this new graph-cut formulation for inferring visibility and enforcing silhouette constraints in the 3D reconstruction problem.

2.2. Photo-consistency driven mesh refinement

Previous graph-cut based reconstruction methods [9, 17] first densely sample voxels on uniform grids to build a graph embedding and then evaluate photo-consistency at all these voxels. This step is more expensive than solving the graph-cut. Here we show how to adaptively sample the volume and avoid evaluating the cost functional in regions which are unlikely to contain the unknown surface. This is achieved by applying a recursive subdivision scheme on a coarse, regular, tetrahedral mesh representing the bounding volume, and adaptively refining the most photo-consistent regions until the desired level of tessellation is reached.

Our base mesh denoted by M_0 is a body-centered cubic (BCC) lattice which comprises the nodes of a 3D grid along with the centers of all the cubic cells (see Fig. 3(a)). It can be thought of as two interlaced cubic lattices. Edges are added between every node in the first grid and its eight diagonal neighbors in the second grid. We choose a simple red-green mesh refinement strategy [11] to obtain a semi-regular mesh from M_0. The mesh obtained after i subdivision steps will be denoted by M_i and its tetrahedral cells and triangular faces by C_i and F_i respectively. A subset of cells in C_i which lie in the photo-consistent region, referred to as the *active* region will be denoted by A_i. The refined mesh M_{i+1} is obtained by applying red-refinement to the cells in A_i and green-refinement to the cells in $C_i - A_i$. A tetrahedron is red-refined into eight tetrahedra as shown in Fig. 3(b) by bisecting its six edges. The shortest of the three possible diagonal edges internal to the tetrahedron must be chosen to make the eight new tetrahedra geometrically similar to the original cell. Green tetrahedra which share faces with red tetrahedra require between one to five edge-splits. Similar to [11], we reduce the various cases of green refinement to the three shown in Fig. 3(b). Green tetrahedra are not geometrically similar to the original BCC tetrahedra and are never subdivided any further.

A photo-consistency measure $g(X) : R^3 \to R$ which computes the likelihood of the 3D point X of being a true surface point is used to find the *active* set $A_{i+1} \subset C_{i+1}$, excluding cells created by green refinement. When the unknown surface passes through a tetrahedral cell, some of its four faces must contain points with a high measure of photo-consistency. We refer to these as *crossing* faces. If none of the faces of a cell contain any photo-consistent points, that cell cannot contain a piece of the surface. We do not refine such cells any further and avoid sampling in their interior. Assuming that the unknown object is large enough not to be completely contained inside a single tetrahedron, a cell must have at least one *crossing* face in order to be labeled *active*. To determine A_{i+1}, we evaluate $g(X)$ on the faces of cells created by red-refinement of A_i and determine the subset of *crossing* faces. Then each *crossing* face labels its two neighboring tetrahedral cells as *active*.

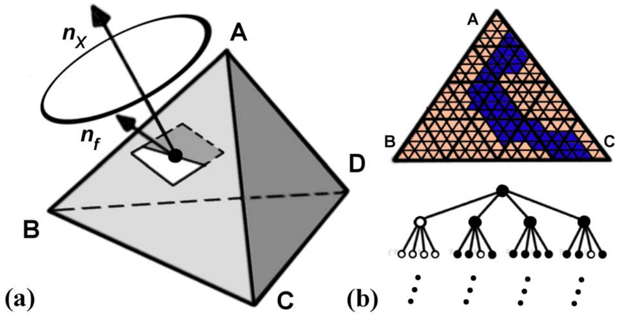

Figure 4. (a) Testing if face ABC with face normal n_f is a *crossing* face; Test patch at P with unit normal n_X for photo-consistency. (b) Computation performed on a triangular lattice (photo-consistent points are blue (dark)) with the results stored in a quad-tree associated with face ABC. Nodes corresponding to *crossing* faces are black in the quad-tree.

2.3. Computing Photo-Consistency

To determine whether a face f is a *crossing* face, we sample a triangular lattice on it as shown in Fig. 4. The spacing between samples in the lattice is selected to prevent aliasing by ensuring that no pixels are skipped when the lattice is projected onto the images. At each lattice position X, we use the normalized cross correlation (NCC) of the image projections of patches placed at X to measure its likelihood of being on the surface. Since the mesh is initially coarse, its faces may not be parallel to true surfaces. In this case, it would be undesirable to compute NCC on the faces themselves. To overcome this, we place multiple patches with different orientations at each point X. The patches and the set of images used for the computation are determined as follows.

At X, we place patches at multiple orientations, each denoted by unit vector n_X. For all points, n_X is chosen from 14 canonical orientations sampled along the corners and faces of a unit cube. For a given orientation n_X, we choose the best k cameras such that the angle between n_X and the direction vector from X towards the camera is at most $60°$. Let us denote this subset of cameras by $P(X)$. If X is a true surface point and n_X is a fair approximation of the surface normal on the unknown surface, then the projection of that patch should have a high matching score for the subset of visible cameras $\subset P(X)$. Since we are only interested in determining whether a point could potentially belong on a surface or not, we use a simple computation for the photo-consistency to reduce computational complexity. We simply place a 1D $1 \times \mu$ grid along the intersection line of the patch and the underlying face f (see Fig. 4). This direction is given by $n_X \times n_f$, where n_f is the normal of the face. This $1D$ grid is now projected into each of the cameras in $P(X)$ and pairwise NCC scores are computed for all such pairs. The photo-consistency score for each camera in $P(X)$ is computed by averaging the best k' NCC scores with the other (k-1) cameras ($k' = \max\{k/2, 3\}$) allowing for matching to succeed despite some occlusion. The score of the best overall camera is retained as the score for the patch at X with orientation n_X. Points with score larger than a global threshold T are labeled photo-consistent. Finally, if a face contains at least 20% photo-consistent points, or at least 20 points if 20% corresponds to a number below 20, we declare it to be a *crossing* face.

This computation could be repeated for every face at every subdivision level during mesh refinement. However this would be highly redundant since during each subdivision level, a large face f splits into four new faces f_1, f_2, f_3 and f_4 whose photoconsistency measures were already computed to decide whether f was a *crossing* face. The solution then is to perform the computation recursively for face f only once and store the results in a quad-tree associated with f (see Fig. 4(b)). Concretely, the root node of the quad-tree corresponds to f while the four children correspond to the faces $\{\ f_i \mid 1 \leq i \leq 4\}$ obtained by bisecting the three edges of f and connecting the mid-points. At each tree node we store: (1) the number of photo-consistent samples (those with matching score $> T$) on the triangle lattice, (2) the total sample-count and (3) the best oriented point for f along-with the set of cameras it correlated on. All such oriented points form a *quasi-dense* reconstruction of the scene (see Fig. 1(c)) and is next used to detect interior and exterior cells. When f is split during subdivision, the four new faces inherit the children of f's quad tree and can be immediately tested for being *crossing* faces.

2.4. Finding the Interior and Exterior

Multiple iterations of mesh refinement produces a set of highly tesselated *active* cells. We will now try to include some of the remaining cells in sets C_{in} and C_{out}. Since the visual hull contains the true shape, any cell which falls outside the visual hull can be labeled as exterior. However, green tetrahedra contained within the visual hull could be either interior or exterior (eg. a deep concavity). The set of quasi-dense oriented surface points recovered during the photo-consistency driven mesh refinement (Section 2.3) allows us to determine which green tetrahedra are part of the true interior. An oriented point p that was photo-consistent in k' views must have been visible from each of those cameras. Hence we path-trace rays from p to all of the camera centers and vote for each cell that the ray intersects along the way. This can be done efficiently by walking along the ray within the tetrahedral mesh and performing ray-triangle intersections. Finally amongst all the green tetrahedra contained within the visual hull, the ones which received votes lower than the 10^{th} percentile are labeled interior, while the ones with votes above the 75^{th} percentile are labeled exterior. Since labeling cells as interior and exterior imposes hard constraints in the reconstruction, we apply the labels conservatively and leave ambiguous cells undecided ie. we re-label them *active*.

3. Proposed Approach

Our complete approach summarized later in Algorithm 1 begins with tetrahedral mesh generation described in Section 2, followed by the first graph-cut on its dual. This is followed by a second graph cut after interior and exterior sets are augmented by enforcing silhouette constraints and a final local refinement. These are described below.

3.1. Graph Construction

Having generated a tetrahedral mesh M and sets C_{in} and C_{out} we then construct G, the dual graph of M. Vertices in G dual to cells in C_{in} and C_{out} are connected to the *source* and *sink* respectively for the graph-cut. Edge capacities in G are derived from the dual oriented faces in M. Unlike in Section 2.3 where 1D patches were used for speed, the goal here is to minimize a true surface cost functional. To this end, a 2D $\mu \times \mu$ grid, placed on each face f, is projected into the images and their pair-wise NCC scores are combined. We pick the best k cameras at an angle of at most 60° from the surface normal of f. Each of these is chosen as a reference view (as in Sec. 2.3) and correlated with the other k-1 views; the best k' ($k' = \max\{\frac{k}{2}, 3\}$) scores out of these truncated to [0,1] are averaged. The best average score is assigned as the final score ω_f of f. Eq 1 shows how ω_f maps to the edge weight $\phi(f)$ where a_f is the area of face f.

$$\phi(f) = \left(1 - \exp\left(-\tan\left(\frac{\pi}{2}(\omega_f - 1)\right)^2 / \sigma^2\right)\right).|a_f| + \lambda.|a_f| \tag{1}$$

As explained in [17], minimizing the surface functional $\sum_S \phi(s)$ over surfaces S embedded in the volume is equivalent to finding the minimal surface with respect to a Riemannian metric [1] where higher values of σ and lower values of λ produce a more photo-consistent but less smooth surface and vice-versa.

3.2. Enforcing Silhouette Constraints

Variational surface reconstruction approaches have a bias for smaller shapes, as surfaces with a lower total cost are preferred over a more accurate surface which has lower cost per unit area but higher total cost. The energy can be regularized by including a *ballooning* term [9, 17] which acts as a prior for larger shapes. While this can recover protrusions, it also pushes the concave parts outwards thereby significantly reducing the accuracy of the final result. While [5] proposes visibility-based *intelligent* ballooning to address this issue, it only reduces the graph cut bias and preserves concavities better but does not guarantee consistency with the silhouettes. We address this in a different way by enforcing hard constraints in the graph-cut derived from both visibility as well as silhouettes constraints.

Figure. 5 shows $\mathcal{S}_r$ the re-projected silhouette overlaid on the original silhouette $\mathcal{S}$. The re-projection errors are in pixels such as x_1 which fall inside S but outside $\mathcal{S}_r$ and x_2 which fall inside $\mathcal{S}_r$ but outside $\mathcal{S}$. Consider the rays r_1 and r_2 backprojected from x_1 and x_2 and the cells they intersect. The ray r_2 should not meet surface because x_2 is outside the silhouette $\mathcal{S}$, therefore all cells intersected by r_2 can be safely labeled added to C_{out}. On the other hand, r_1 must intersect the surface at least twice. Thus at least one of the cells that r_1 passes through must be an interior cell. For such rays in every view, we intend to mark at least one such cell as interior and add it to our interior set.

We adopt a two-step approach. First, by computing the minimum cut on G as described above, we obtain (a) a triangulated surface (b) a partition of all tetrahedral cells into

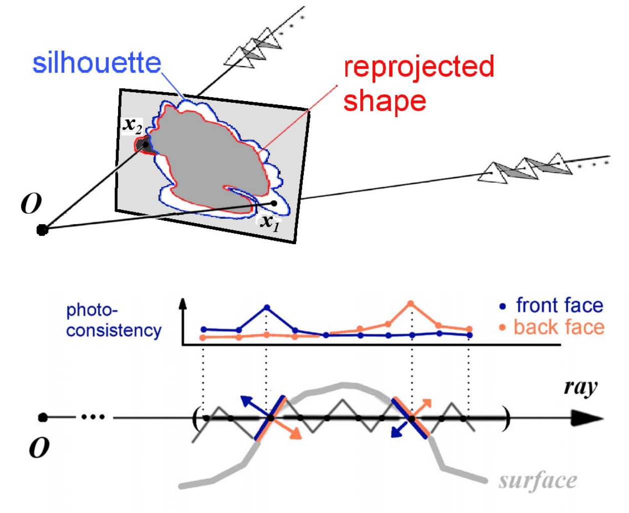

Figure 5. Top: the original silhouette $\mathcal{S}$ and re-projected silhouette $\mathcal{S}_r$ (O is the camera center). x_1 and x_2 indicate re-projection errors (see text for details). Bottom: for x_1, we inspect photo-consistency costs on front and back-faces for all triangles in M which are intersected by the ray back-projected from x_1.

C^1_{in} (interior cells) and C^1_{out} (exterior cells). The triangulated surface is then re-projected into all the images and the sets of erroneous pixels (such as x_1 and x_2) are determined. Pixels such as x_2 add cells to the set C_{out}. Pixels such as x_1 are processed to mark some additional cells in M as interior; these are added to C^a_{in}, the augmented set of interior cells. The candidate cell is chosen as follows. We first find the sequence of tetrahedral cells in M that ray r_1 cuts through and sort them by distance from the reference camera. Cells in this sequence that fall outside the visual hull are excluded, leaving groups of contiguous tetrahedral cells each of which we will refer to as a *segment*. For each *segment*, we orient the triangles (faces of the cells) consistently with respect to the ray. Let us first consider the simpler case when r_1 intersects the surface twice (see Fig. 5). This ray must meet a triangle f_f whose *front-face* is photo-consistent before it meets a triangle f_b whose *back-face* is photo-consistent. A few tetrahedral cells within such a depth-interval can be chosen as interior. More specifically, we look for the maxima of front-face photo-consistency and find the next significant peak in back-face photo-consistency (within 0.8 of the global maximum) for faces along r_1 in the same *segment* to determine a conservative depth interval for the interior. We then pick the center-most cell in this depth interval and add it to C^a_{in}. This step is highly redundant and we pick candidates (a hard constraint) only when we are sure about a cell being interior. We skip pixels with multiple *segment*s per ray and let a more favorable view enforce silhouette constraints there. In our experiments processing only a few pixels was sufficient to recover all the protrusions. It is better to enforce a minimal number of additional hard constraints for silhouette consistency since performing this step exhaustively increases the likelihood of including incorrect constraints.

A second minimum-cut is now computed on the same graph G but with the augmented interior set C^a_{in} as *source* and augmented exterior set C^a_{out} as *sink*. This new triangulated surface satisfies silhouette constraints upto a few pixels (the actual value depends on the cell resolution of M and is typically in the range of 1-5 pixels in the images. An analogy can be drawn between our approach and the graph-cut based Grab-cut [13] segmentation method, where iterative graph-cut optimization is performed while the user interactively provides additional hard constraints. In a similar fashion, we use silhouettes for generating reliable hard constraints (automatically in our case) as described above and perform a second graph-cut iteration to correct the shortcomings of the first one.

```
**Input**: images $\{I\}$, cameras $\{P\}$, bounding-box $B$
**Output**: polygonal mesh model $H$

$M_0 \leftarrow$ BuildBCCMesh($B$);
$Q = \{\ \}$;
for $i \leftarrow 0$ to $m$-1 do
    patches $\leftarrow$ ComputeMatchingCost($F_i$);
    $A_i \leftarrow$ FindActiveCells($M_i, F_i$);
    $M_{i+1} \leftarrow$ MeshRefine($M_i$);
    $Q \leftarrow Q \cup$ patches ;
end
$C_{in}, C_{out} \leftarrow$ MarkInteriorExterior($M_m, Q$) ;
$G \leftarrow$ SetupGraphCut($M_m, C_{in}, C_{out}$) ;
$[S_1, C^1_{in}, C^1_{out}] \leftarrow$ FindMinCut($G$) ;
foreach camera j in $\{P\}$ do
    $K_j \leftarrow$ RenderSilhouettes($S_1, P_j$) ;
end
$C^a_{in} = C^1_{in}$;   $C^a_{out} = C_{out}$;
foreach camera j in $\{P\}$ do
    $C^a_{in}, C^a_{out} \leftarrow$ EnforceSilhouettes($I_j, K_j$) ;
end
$G^{'} \leftarrow$ SetupGraphCut($M_m, C^a_{in}, C_{out}$) ;
$S_2 \leftarrow$ FindMinCut ($G^{'}$) ;
$H \leftarrow$ RefineShape ($S_2$);
```

Algorithm 1: The Complete Algorithm.

3.3. Surface Refinement

Finally, local optimization is used to refine the shape locally to remove discretization errors introduced by the graph cut reconstruction. The triangulated minimum-cut surface mesh is iteratively deformed and remeshed during local refinement similar to [3, 4]. Vertices of the mesh are displaced by a combination of smoothness, silhouette and texture forces. The smoothness force is computed by the

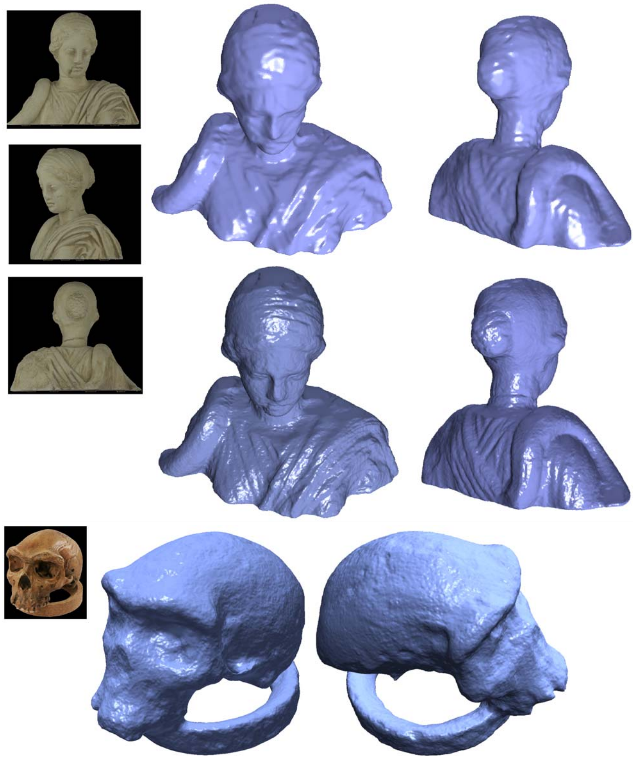

Figure 6. [Top] *statue3* dataset: Three of the input images. The reconstructed surface from the graph-cut step is shown on the top row while the final 3D model after refinement is displayed in the middle row. [Below] *skull* dataset: Two views of the final model.

approach of [18] which prevents the surface from shrinking while silhouette forces are computed as described in [3]. To compute the texture force, we use the normal vector at a vertex to determine a set of k cameras and compute a photo-consistency score (see Section 3.1) at multiple offsets from the current vertex location along its surface normal. A red-green 2D mesh subdivision scheme [11] is used to remesh the model after bisecting edges which project to more than 2 pixels in the best reference view.

4. Results

We have reconstructed several real multi-view datasets using our approach as shown in Figs. 1,6,7 and 8. Datasets *statue1, statue2* and *statue3* contain 36 images (6 Mpixels) each, captured using a turntable. The *head* dataset contains 21 640 × 480 images without good color calibration while the *skull* dataset contains 24 2000 × 2000 images.

We have participated in the Multi-View Stereo Evaluation (the reconstructions are shown in 7(a)). This evaluation provides metrics on the accuracy and completeness of the reconstruction. The accuracy metric is the distance d such that 90% of the reconstructed is within d from the ground truth surface. Completeness is defined as the percentage of the ground truth surface within $1.25mm$ of the model. The accuracy and completeness of our reconstruction for the 47-view *templeRing* dataset were $0.79mm$ and 94.9% respectively. The same metrics for the 48-view *dinoRing* dataset were $0.69mm$ and 97.2%.

Fig. 8 illustrates the results of an experiment performed to demonstrate that our method is not limited by the topology of the base surface. While the visual hull built from all 36 images has the correct topology, the visual hull built after omitting 10 images (the separation between the arm and body is observed in these) has genus three. Our method still recovers a model with the correct topology (see Fig. 8(e,f)).

The critical parameters of our algorithm are chosen as follows. The patch size μ is typically 11 pixels while the photo-consistency threshold T is chosen in the range of 0.4-0.7 (a fraction between 0 and 1). A lower T is more conservative and retains more cells as *active*. The surface functional parameters of σ is set to 0.1 in all our experiments and λ is varied between 1 to 10. The stopping criterion for recursive mesh refinement is based on the size of the finest cells in the images; we typically stop when this is in the range of 1 to 5 pixels.

Our method requires a smaller fraction of graph vertices compared to approaches which construct uniform grid graphs in the interior of the visual hull. Our mesh typically has between 2-10 million cells and total running time are typically 1 to 2 hours for each reconstruction.

5. Conclusions

We have presented a novel multi-view reconstruction method that recovers surfaces at high resolution by performing a graph-cut on the dual of an adaptive volumetric mesh created by photo-consistency driven subdivision. We do not need good initializations and are not restricted to a specific surface topology (a limitation with base surfaces). Our graph-cut formulation enforces silhouette constraints to counter the bias for minimal surfaces. In future we will investigate whether substituting the hard constraints we enforce (interior or exterior labels) with per-cell data penalties in the energy functional (similar to graph-cut based Markov Random Field optimization) produces better results. Our current implementation of local refinement needs improvement. This will be addressed in future work.

Acknowledgements We thank the authors of [3, 20, 5] for the *skull*, *head* and the three *statue* datasets respectively. The support of the Packard Foundation and the NSF Career award IIS 0237533 is gratefully acknowledged.

References

[1] Y. Boykov and V. Kolmogorov. Computing geodesics and minimal surfaces via graph cuts. In *ICCV*, pp. 26–33, 2003.

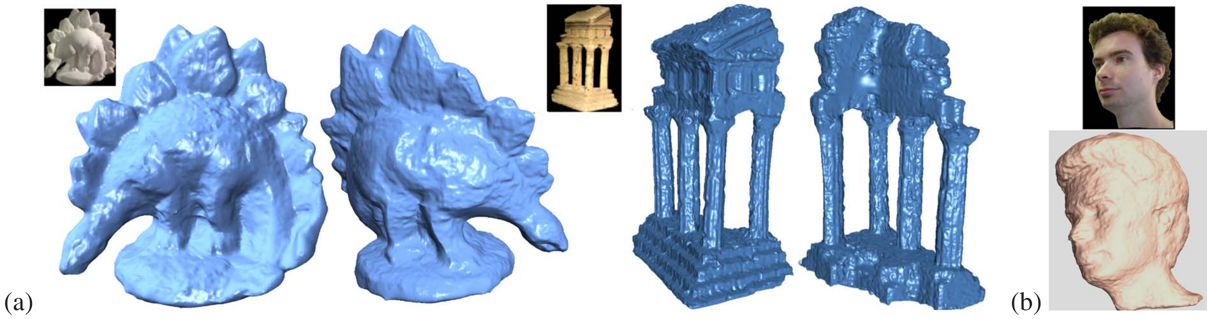

Figure 7. (a) Middlebury Multi-view Stereo Evaluation benchmarks: (left) *dinoRing*, (right) *templeRing*. (b) *head* dataset reconstructed from a set of 21 images lacking perfect color calibration.

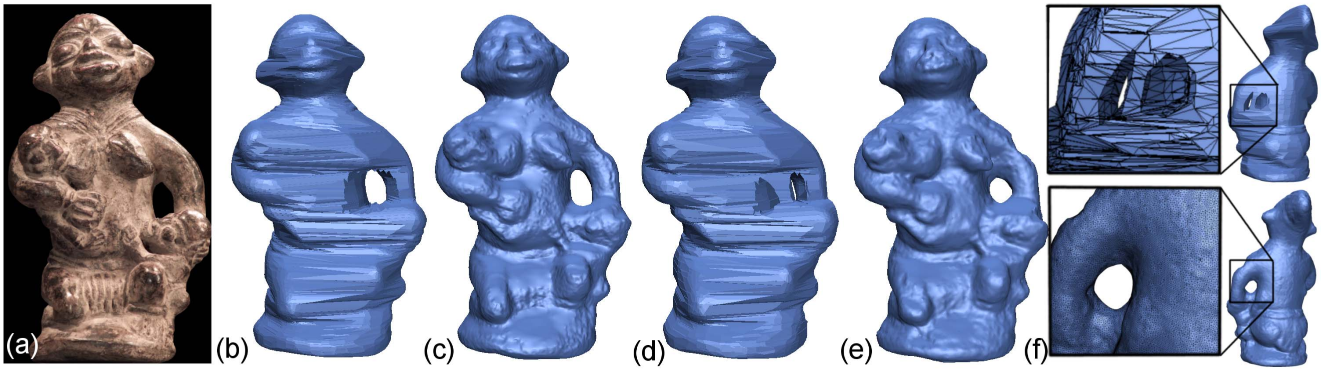

Figure 8. *statue2* dataset: (a) One of the input images (note that in our experiments we leave this image out). (b) Visual Hull from all 36 images. (c) Our result using all 36 images. (d) Visual Hull from 26 images (10 out of 14 views which see the gap between arm and body are left out) has genus 3. (e) Our model using these 26 images has the correct topology (genus 1). (f) Zoomed-in rear view of (top) visual hull (bottom) our result using these 26 views.

[2] Y. Boykov and V. Lempitsky. From photohulls to photoflux optimization. In *BMVC*, pp. 1149–1158, 2006.

[3] Y. Furukawa and J. Ponce. Carved visual hulls for image-based modeling. In *ECCV*, pages I: 564–577, 2006.

[4] C. Hernández-Esteban and F. Schmitt. Silhouette and stereo fusion for 3d object modeling. *CVIU*, 96(3):367–392, 2004.

[5] C. Hernández-Esteban, G. Vogiatzis, and R. Cipolla. Probabilistic visibility for multi-view stereo. In *CVPR*, 2007.

[6] A. Hornung and L. Kobbelt. Hierarchical volumetric multi-view stereo reconstruction of manifold surfaces based on dual graph embedding, In *CVPR*, 2006.

[7] D. Kirsanov and S. Gortler. A discrete global minimization algorithm for continuous variational problems, Harvard CS Technical Report TR-14-04, 2004.

[8] V. Kolmogorov and R. Zabih. Multi-camera scene construction via graph cuts. In *ECCV*, pp. 82–96, 2002.

[9] V. Lempitsky, Y. Boykov, and D. Ivanov. Oriented visibility for multiview reconstruction. In *ECCV*, pp. 226–238, 2006.

[10] M. Lhuillier and L. Quan. A quasi-dense approach to surface reconstruction from uncalibrated images. *PAMI*, 27(3):418–433, 2005.

[11] N. Molino, R. Bridson, J. Teran, and R. Fedkiw. A crystalline, red green strategy for meshing highly deformable objects with tetrahedra. In *12th International Meshing Roundtable*, pp. 103–114, 2003.

[12] J. Pons, R. Keriven, and O. Faugeras. Modelling dynamic scenes by registering multi-view image sequences, In *CVPR*, 2005.

[13] C. Rother, V. Kolmogorov, and A. Blake. "Grabcut": interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph.*, 23(3):309–314, 2004.

[14] S. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *CVPR*, pp. 519–528, 2006.

[15] 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.

[16] S. Tran and L. Davis. 3d surface reconstruction using graph cuts with surface constraints. In *ECCV*, pp. 219–231, 2006.

[17] G. Vogiatzis, P. Torr, and R. Cippola. Multi-view stereo via volumetric graph-cuts, In *CVPR*, 2005.

[18] J. Vollmer, R. Mencl and H. Môller. Improved laplacian smoothing of noisy surface meshes. In *Computer Graphics Forum*, volume 18(3), pp. 131–138, 1999.

[19] T. Yu, N. Ahuja, and W. Chen. SDG cut: 3d reconstruction of non-lambertian objects using graph cuts on surface distance grid. In *CVPR*, pp. 2269–2276, 2006.

[20] G. Zeng, S. Paris, L. Quan, and F. Sillion. Accurate and scalable surface representation and reconstruction from images. *PAMI*, 29(1):141–158, 2007.