

Single Viewpoint Model Completion of Symmetric Objects for Digital Inspection

Alvin J. Law and Daniel G. Aliaga

Department of Computer Science, Purdue University

Abstract

The ability to create complete 3D models of real world objects is an important task for various applications. In digital inspection, complete models allow users to analyze the entirety of an object. However, various difficulties arise for image-based acquisition techniques. First, the viewpoint planning problem must be solved. Second, each of the resulting viewpoint captures must be combined with either zippering or 3D triangulation, both difficult problems. We observe that if an object is symmetric, then the object's symmetry can be exploited so that a single viewpoint capture is sufficient to generate a complete, 3D triangulated model. In our work, three problems of previous approaches to generating complete models are avoided or minimized: 1) we avoid 3D triangulation, 2) we avoid searches for geometry to extend our models, and 3) we minimize viewpoint planning to the selection of a single viewpoint. Our approach also includes algorithms to mitigate global deformations due to capture error. We demonstrate our approach by capturing, reconstructing, and completing several scenes of one or more objects and illustrating several digital inspection methods with these scenes.

Keywords: single viewpoint acquisition; model completion; symmetry detection; digital inspection.

1. Introduction

We present a method to obtain complete 3D models of real-world symmetric objects for use in digital inspection. Digital inspection provides tools for non-experts to closely examine the details of a captured object. For example, an artist, archaeologist, or historian might wish to digitally magnify surface details, to re-light the objects, or to create synthetic illustrations. In these situations, complete 3D models enable the user to digitally inspect more than that which is initially captured – this can be particularly useful when dealing with an artifact fragment where the remainder of the artifact is lost. Being able to quickly and virtually slice a virtually completed object provides a powerful visualization ability. Moreover, a low-cost and easy-to-use approach permits its easy and widespread dissemination.

Thus, a major challenge of digital inspection is to obtain complete, high-resolution models for a comprehensive inspection of the object. While approaches to acquiring complete models have been proposed, each approach has its own challenges. Manual modeling using computer software is a time consuming option. 3D acquisition methods (e.g., structured-light [22], laser scanning [10], or passive stereo [17]) can capture models, but obtaining complete models requires special equipment or multiple viewpoints to observe the entire object. The multiple viewpoint requirement introduces viewpoint planning [23]. Additionally, the acquisitions from multiple viewpoints must be aligned (e.g., with Iterative-Closest-Points (ICP) [3, 21]) and zippered [29]. In both steps, the negative effect of deformations in the acquired fragments must also be addressed. The acquired fragments' points can alternatively be merged via 3D triangulation, a complex problem in itself (e.g., handling noise and avoiding holes or other artifacts [5]).

For scenes with multiple objects, acquiring complete models is even more difficult due to potential visibility constraints being introduced. In fact, if the objects are of high importance and manipulating or re-arranging them is not an option, there may not be a practical configuration that enables capturing all of the objects' surfaces. Altogether, the task of a digital inspection system is confronted with several model acquisition issues and limitations in viewpoint planning, 3D triangulation, and the registration of multiple scans to form a single model.

Our key inspiration is to exploit the symmetry present in objects to extrapolate a plausible complete model of an object from a single viewpoint. In doing so, we minimize or avoid three problems of previous works: 1) the viewpoint planning problem is reduced to choosing a single viewpoint where the object's symmetry is identifiable; 2) 3D triangulation is avoided, and the initial single viewpoint capture is triangulated in 2D image space; and 3) since symmetry can be used to identify geometry for filling in holes and for extending the model beyond its captured borders, searches to find suitable geometry (as in [14] and [25]) to complete the model are unnecessary. The same principles apply with no additional difficulties for scenes with multiple objects, thus enabling the acquisition and inspection of multi-object scenes with minimal additional effort. Nevertheless, there is no explicit guarantee that the extrapolated models are exact matches of the original objects. For example, if a unique feature of the object was not in the initial capture (e.g., a vase's handle), then the requisite geometry to replicate the feature would be missing in the final model. For an approximately symmetric object (e.g., a manmade object intended to be symmetric but is not due to human error), the slight asymmetry would not be captured. Despite these limitations, our method is still suitable for a wide range of objects.

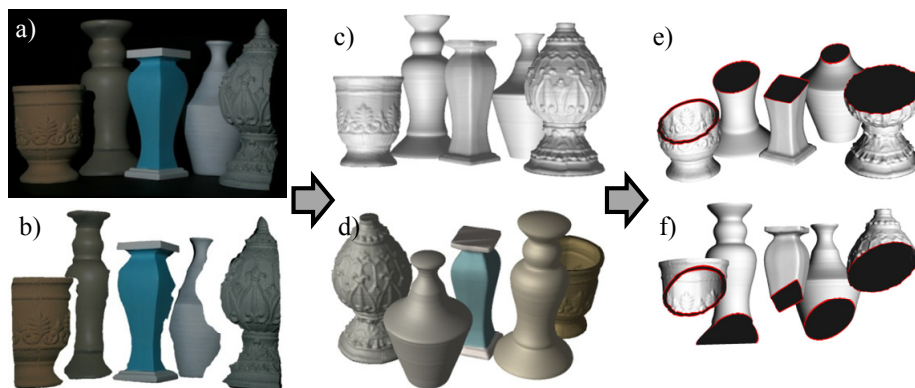


Figure 1: Single viewpoint model completion for digital inspection. *a) Picture of a multi-object scene. Left to right, these objects are ‘Small Vase’, ‘Tall Pedestal’, ‘Pedestal’, ‘Smooth Vase’, and ‘Saint’. b) Initial single viewpoint reconstruction. c) Completed models of the observed objects rendered in grayscale from a vantage point different from the capture viewpoint. d) The completed models texture-mapped. e-f) Interactive digital inspection examples using the objects produced by our method.*

Our approach to 3D model completion for digital inspection consists of four steps (Figure 1). First, from a single viewpoint, we capture a detailed fragment of an object. Second, we manually select the object’s symmetry class – we support three common symmetry classes: bilateral symmetry, rotational symmetry, and surface-of-revolution symmetry – and automatically discover the relevant symmetric features. Third, symmetric geometry fragments are replicated and merged with our custom zipping process to construct a complete model. Then, under user control, we assume the object to be either solid or hollow and either close the mesh or add an inset to create a plausible interior surface for the model. Finally, we interactively generate digital inspection illustrations of the completed models using techniques similar to [1, 19]. Our approach is fully automated aside from choosing an object’s symmetry class and the closing style of the resulting model. We have applied our method to capture individual models ranging up to

900k triangles and multi-object scenes ranging up to 4.1 million triangles, and the resulting models have an average sampling resolution of 0.35mm.

Our contributions are twofold. First, we present robust symmetry algorithms for our three types of supported symmetry capable of identifying the global symmetry of a model from a fragment of it. Second, we describe a method for producing complete models of the objects from a single viewpoint, including a zippering procedure capable of adapting to irregularities between fragments. We also describe techniques to better improve the quality of symmetry detection as well as mesh zippering where appropriate.

2. Related Work

Our work spans research in acquisition and modeling, symmetry detection, and visualization and rendering. We summarize the relevant research below.

2.1. Acquisition and Modeling

Various techniques exist to acquire high-resolution 3D models using image sequences, lasers, or structured light (e.g., [2, 10, 17, 22]). The motion of an object can also be tracked (e.g., across video frames [11] or by hand [20]) to generate a model, where tracked captures represent viewpoints observing the object [11]. These approaches obtain complete models by increasing the number of capture viewpoints. However, the viewpoint planning problem to decide where to acquire additional data remains a difficult and open problem.

Several techniques have been proposed to complete or expand 3D models from fragments. A priori databases of shapes can provide geometric priors for filling in the holes of a model or for synthesizing new models (e.g., [7, 14, 24]). Searching for a suitable patch in the model itself has also been proposed (e.g., [25]). Symmetry removes the need to search for suitable geometry.

Combining separately acquired mesh fragments traditionally requires an alignment process (e.g., ICP [4, 8, 21] or 4PCS [6]) and zippering [29]. Warping functions (e.g., [14]) and Poisson-based gradient field manipulation (e.g., [30]) have also been proposed in order to alter mesh fragments to fit seamlessly. Symmetrization [13] enforces symmetry in meshes, but the result may incorrectly warp the geometry. Enforcing symmetry during acquisition has also been explored (e.g., for bilateral symmetry [31]). While this would help alleviate capture errors, we opt to alter the mesh’s geometry after acquisition. Combining symmetric fragments is unique in that the fragments are already reasonably aligned and shaped. Thus, our zippering process avoids an alignment process, and only a small amount of mesh warping and interpolation is needed.

2.2. *Symmetry Detection*

Our method requires detecting the global symmetry of an object from a fragment of it. Methods identifying symmetry only in the capture geometry (e.g., [12, 16, 26]) are inadequate since they do not consider the symmetry beyond the capture borders. Voting based techniques (e.g., [12, 16]) may detect an object’s global symmetry, but depending on the amount of global symmetry evident in the capture, local symmetries may be more strongly identified. With our capture fragments, we avoid incorrect classifications from ambiguous configurations (e.g., two captured faces of a rotationally symmetric object may exhibit bilateral symmetry; a surface-of-revolution fragment inevitably exhibits rotational and bilateral symmetry) by letting the user select the symmetry class with a simple interactive selection. Instead, we focus on obtaining plausible, complete models.

A variety of shape-fitting and shape-classification methods have been proposed to detect symmetry from fragments (e.g., fitting point sets to cylindrical,

surface-of-revolution, or helical symmetry [18] or computing curve skeletons [9, 27]). We instead use optimizations to find the necessary symmetric information. Thrun and Wegbreit [28] generate complete point models from a partial capture, but these points would have to be cleaned-up prior to 3D triangulation. Our approach is capable of generating a complete model without 3D triangulation and without requiring a densely sampled object. Pauly et al. [15] introduced a scheme to identify structural regularity in objects. However, their approach relies on a sufficient amount of repetition for robust structure detection. Our method generally requires minimal symmetric repetition.

2.3. *Visualization and Rendering*

Several works have developed visualization and rendering methods that support digital inspection. These methods enhance the visualization of scientific and historically-important objects by exploiting photorealistic and non-photorealistic rendering strategies (e.g., [19]). However, these methods assume a priori captured 3D models and therefore are limited to the quality and completeness of the provided models. Our work provides complete 3D models of symmetric objects for such methods as well as other uses.

3. Method Overview

Our method acquires, completes, and digitally inspects single and multi-object scenes. Figure 2 shows a pipeline for our method. We use a self-calibrating structured-light method (similar to [2]) with one (static) camera for the single viewpoint capture. Then, the captured object points are triangulated using 2D Delaunay triangulation and placed into a kd-tree.

Next, the user chooses the object’s symmetry class: bilateral symmetry, rotational symmetry, or surface-of-revolution symmetry. Each unique, repeating

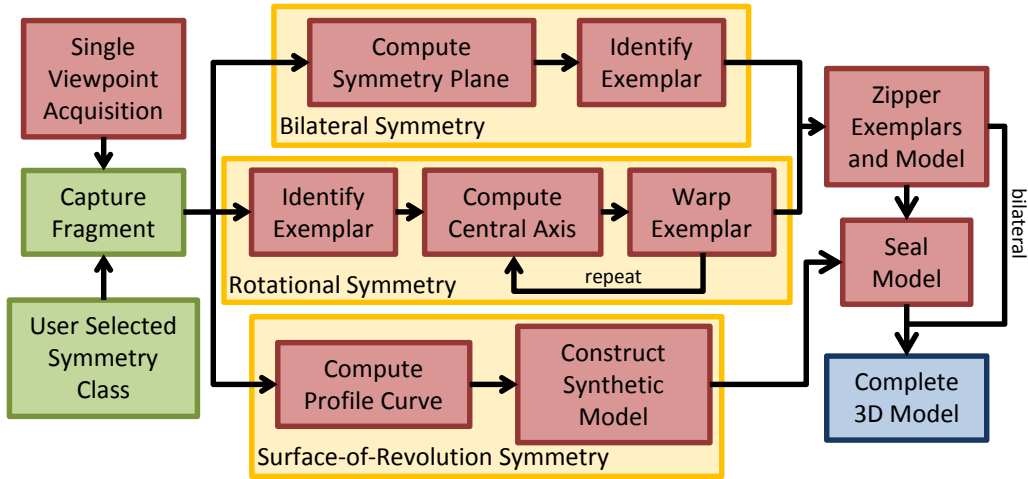


Figure 2. Method pipeline. A pipeline summary of our method.

fragment is called an *exemplar*. Bilateral symmetry exists when an object’s two exemplars are mirrored across a symmetry plane. Rotational symmetry exists when an exemplar is repeated $n_r \geq 3$ times around a central axis (three exemplars is the minimum since a two exemplar object is just a two-sided object). Surface-of-revolution symmetry exists when the exemplar is an infinitely thin slice (profile curve) rotated around a central axis. Symmetry detection is automated after the user specifies the symmetry class.

Then, to complete models with bilateral or rotational symmetry, exemplars are copied to extend or complete the model’s borders. Next, the exemplars are zippered together. Like [14], mesh warping is performed at the overlapping areas and with a locally fitted smooth surface model prior to zippering in order to combat the effects of capture deformations and to smooth out the transition between exemplars. Finally, rotationally symmetric models are closed either by adding an inset to the model to provide a sense of volume or by capping the top and bottom of the model. Due to limitations in the captures of objects with bilateral symmetry (i.e., if the symmetry plane is detectable from a single

viewpoint, a complete exemplar cannot be acquired), we abstain from closing these models. For surfaces-of-revolution, we generate a synthetic model by rotating the profile curve around its central axis. The models are then closed as describe previously. Model completion for all three types of symmetric objects is fully automated.

Finally, interactive digital inspection is enabled for any of our captured objects using several visualization and rendering algorithms.

4. Symmetry Detection

The goal of our symmetry detection algorithms is to identify the overall symmetric patterns of an object using only a fraction of the object as input. For each of our three supported symmetry classes, we derive an error function to measure the difference from the expected point locations to the actual points captured without needing an a priori correspondence of symmetric points. We use an implementation of Levenberg-Marquardt nonlinear least squares to find the optimal symmetry parameters. Our symmetry detection algorithms are designed as nonlinear least squares optimizations robust to outlier points not following the expected symmetry. Without loss of generality, we assume the objects to be roughly vertical; if not, the model can be rotated prior to computation. This assumption simplifies the equations in the following sections but does not prevent the completion of objects without a vertical symmetry axis or plane. For efficiency, all point searching is done in pre-constructed kd-trees.

4.1. Viewpoint Selection

The viewpoint used for the initial capture should generally observe as much of the objects' surfaces as possible. For any scene, selecting a single viewpoint is simpler than viewpoint planning, and we assume an appropriate viewpoint exists.

In scenarios with inadequate observed symmetry, manual intervention is required to select initial guesses for our optimizations.

For objects exhibiting bilateral symmetry, a suitable viewpoint is one in which the expected symmetry plane is neither parallel nor perpendicular to the viewing direction, and geometry on both sides of the symmetry plane is reasonably observable in the initial capture.

For objects with rotational symmetry, we select a viewpoint such that the initial capture contains more than one and ideally two or more exemplar instances. Depending on n_r , a different amount of angular coverage of the object is required (e.g., $n_r = 3$, requires seeing 240° of the object; if $n_r = 6$, only 120° of the object is needed). Thus, as n_r increases, the object's angular span requirement decreases, making models with high repetition simpler to complete.

Symmetry detection for a surface-of-revolution requires only a sliver of the object in the initial capture. Theoretically, three point samples are adequate to define a point on each profile curve. In practice, accuracy improves with considerably more points to minimize the effect of noise and global deformations. We assume the selected viewpoint can observe the surface-of-revolution for any z (or height) value; else, the profile curve will be estimated only at some z values.

4.2. *Bilateral Symmetry*

To detect bilateral symmetry, model points are paired with points at the corresponding reflected location across a symmetry plane (Figure 3a). Calculation of bilateral symmetry parameters is automated and typically takes 5-10 minutes.

Given a collection of object points p_i , we search for a symmetry plane that minimizes the sum of differences between the mirrored location of p_i and the closest found pairing point q_i . The equation set that encapsulates this symmetry is

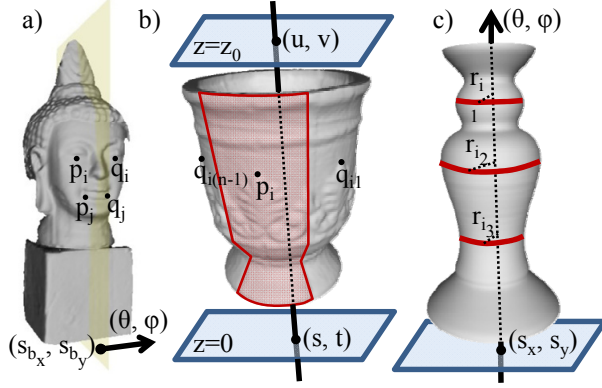


Figure 3: Symmetry Detection. a) Bilateral symmetry is determined by a plane of symmetry. b) Rotational symmetry searches for a central axis about which n_r exemplars rotate. c) Surface-of-revolution symmetry fits circles perpendicular to a central axis.

$$\sum_i \left\| q_i - \left(p_i + 2 \left(\frac{s_n}{\|s_n\|} \cdot (s_b - p_i) \right) \frac{s_n}{\|s_n\|} \right) \right\| \rightarrow 0 \quad (1)$$

where (p_i, q_i) defines an object point and its paired mirrored point, and s_b and s_n are the point and normal defining the symmetry plane. Using trigonometric functions, we remove variables in the denominator and reduce the number of unknowns by observing that (i) twist of the normal about itself is useless, and (ii) s_b intersects the xy -plane because we assume the symmetry plane to be roughly vertical. Thus, s_n is represented in spherical coordinates, and s_b lies on the xy -plane. The resulting equation set is:

$$\sum_i \left\| q_i - \left(p_i + 2 \left(\begin{bmatrix} \cos \theta \sin \phi \\ \sin \theta \sin \phi \\ \cos \phi \end{bmatrix} \cdot \left(\begin{bmatrix} s_{b_x} \\ s_{b_y} \\ 0 \end{bmatrix} - p_i \right) \right) \begin{bmatrix} \cos \theta \sin \phi \\ \sin \theta \sin \phi \\ \cos \phi \end{bmatrix} \right) \right\| \rightarrow 0 \quad (2)$$

where θ , ϕ , s_{b_x} , and s_{b_y} are the unknowns. We pair p_i and q_i if q_i is found within a distance threshold t_d of the mirrored point. If no q_i is found (e.g., a hole in the geometry), then p_i is ignored. Each p_i creates one equation for the optimization. The initial guess is a plane perpendicular to the ground computed by varying θ from 0° to 180° and choosing the θ which yields the smallest error. In our experiments, the error was smoothly varying across θ , making this initial guess

reasonable. Despite the vertical nature of the initial guess plane, the optimization does not limit the final symmetry plane to remain vertical.

The collection of points p_i is defined as a subset of the captured points which adequately spans the object’s captured surface. This set is made by gathering points (typically 100-200) from the top of the kd-tree. To prevent outliers in the optimization and incorrect symmetry detection, each p_i selected should have a point within t_d of its mirrored location of the initial guess plane to prevent computing an incorrect symmetry plane.

4.3. Rotational Symmetry

To detect rotational symmetry, we find an exemplar that repeats n_r times around a central axis. We alternate 2-3 times between two optimizations. First, we solve for the central axis given an exemplar estimate (Figure 3b). Then, we warp the exemplar given a central axis estimate. Calculation of rotational symmetry is automated after the user chooses n_r and typically takes 40-60 minutes.

Solving for the central axis uses the object points p_i to find corresponding points q_{ij} (point i on repeated exemplar $j = 0 \dots n_r - 1$) surrounding the to-be-estimated central axis. Since the central axis is assumed to be roughly vertical, our method uses a two-plane parameterization for it (i.e., the direction of the axis can be expressed as two angles θ and ϕ at the expense of using trigonometric functions). The st -plane is placed at $z = 0$ and the uv -plane at some $z = z_0$. Further, s and t can be assumed zero by subtracting $(s, t, 0)$ from all p_i . Thus, using a nonlinear least squares optimization, we wish to minimize the expression

$$\sum_i \sum_j \left\| q_{ij} - R_{j\theta_{n_r}}(u, v) \left(p_i - \begin{bmatrix} s \\ t \\ 0 \end{bmatrix} \right) \right\| \rightarrow 0 \quad (3)$$

where the unknowns are s , t , u , and v , $\theta_{n_r} = 2\pi/n_r$, and $R_{j\theta_{n_r}}(u, v)$ is a rotation matrix by $j\theta_{n_r}$ about the axis (u, v, z_0) . The axis need not be normalized; instead, the normalization is embedded in $R_{j\theta_{n_r}}(u, v)$. As with bilateral symmetry detection, each point p_i yields an equation for the optimization.

The initial capture points are separated into two groups, the exemplar and the minimization group. The exemplar is formed by taking the angularly central $\theta_{n_r} + \delta$ span of points from the initial capture with δ angular padding to ensure that repeated exemplars overlap. The minimization group consists of all the remaining points from the initial capture and is the set of points p_i used during minimization. While two or more copies of the exemplar in the initial capture are ideal, our algorithm is robust enough to handle a capture with less than two (but greater than one) copies of the exemplar.

Each iteration of this optimization attempts to best fit the sampled model points to the rotationally symmetric model implied by Equation 3. A rotationally symmetric model is created each iteration by rotating the exemplar n_r times, θ_{n_r} each time, about the current estimated central axis and by placing exemplar copies at each rotation. Then, the distances between each minimization point p_i and its closest found exemplar copy point q_{ij} are summed together. For efficiency, all exemplar copy points q_{ij} are placed into a kd-tree for each synthetically-made rotational model. In practice, the tree's size can be reduced by omitting exemplar copies which do not span the expected location of the minimization points. To initialize the optimization, Equation 3 is first solved with u and v set to zero. The effects of outliers are mitigated by the large number of points used.

Our second optimization is used to warp the exemplar. We use an estimate of the central axis to form an orthogonal basis (u, v, w) , setting v to the central

axis. Then, our warping optimization consists of solving for three scales along the basis's axes (s_u, s_v, s_w) and two rotations (r_u, r_w) . Rotation about v is omitted since phase shifts are irrelevant. For each iteration during this optimization, we warp the exemplar using (s_u, s_v, s_w) and (r_u, r_w) and minimize two sets of equations. The first set of equations is described by Equation 3 and ensures that rotational symmetry is maintained. The second set of equations is designed to preserve the overall size of the exemplar (else s_u , s_v , and s_w may shrink the model to a point since all points would then be at zero distance) by measuring the difference between a set of mesh triangle edges' original lengths against their current lengths. The two equation sets are weighted in importance; typically, we use a 3:1 importance ratio between Equations 3 and the edge equations.

4.4. Surface-of-Revolution Symmetry

To identify surface-of-revolution symmetry, our approach partitions the model into a set of approximately horizontal slices such that the points of each slice lie on a circle centered about a central axis (Figure 3c). In this case, the exemplar is an infinitely thin vertical slice known as a profile curve. Calculating surface-of-revolution symmetry is automated and takes 10-15 minutes.

Using a set of model points p_{ij} from a slice (point i on slice j), the equation set to minimize is

$$\sum_i \sum_j \left(\frac{\|(s_2 - s_1) \times (s_1 - p_{ij})\|}{\|s_2 - s_1\|} - r_j \right)^2 \rightarrow 0 \quad (4)$$

where each slice has an unknown radius r_j and s_1 and s_2 are two unknown 3D points defining the central axis. To remove variables in the denominator, we again use a two-plane parameterization and spherical coordinates. By setting s_1 to the origin and using spherical coordinates to represent s_2 , the formulation becomes

$$\sum_i \sum_j \left(\left\| \begin{bmatrix} \cos \theta \sin \phi \\ \sin \theta \sin \phi \\ \cos \phi \end{bmatrix} \times \begin{bmatrix} s_x - p_{i_x} \\ s_y - p_{i_y} \\ 0 - p_{i_z} \end{bmatrix} \right\| - r_j \right)^2 \rightarrow 0 \quad (5)$$

where the unknowns are the r_j 's and θ , ϕ , s_x , and s_y which define a central axis through $(s_x, s_y, 0)$ and $(s_x + \cos \theta \sin \phi, s_y + \sin \theta \sin \phi, \cos \phi)$. The number of slices is selected to obtain more than three points but less than a few hundred points per slice to balance the number and accuracy of the r_j 's. Each slice results in an equation for the optimization. Since we require four more equations (to solve for θ , ϕ , s_x , and s_y otherwise the system is underdetermined), we partition the four largest slices into two groups each to obtain four more equations. A small subset of each slice's points is used to initialize each r_j , and the xy -center of the model is used to initialize s_x and s_y . The impact of outlier points is averaged away since each final r_j value is computed using a slice of many capture points.

5. Model Completion

Model completion uses the initial capture of the object with its identified symmetry as input and produces a new model as output. For objects with bilateral or rotational symmetry, a model is generated by copying the exemplar appropriately, zippering the exemplars, and closing the model (for objects with rotational symmetry). Model completion takes 10-20 minutes depending on the amount of zippering required. For surfaces of revolution, the detected profile curve is sufficient to generate a synthetic model within seconds.

5.1. Exemplar Copying

The exemplar is copied with respect to the object's symmetry to complete the model. Each exemplar is extended to make an overlap region for fitting purposes prior to zippering. The overlap also helps reduce the amount of noise in

the final model. For bilateral symmetry, the exemplar consists of geometry with no corresponding geometry on the opposite side of the symmetry plane, and a mirrored copy of the exemplar is appended to the model. Adjacent geometry to the exemplar is also copied to create the overlap. For objects exhibiting rotational symmetry, a $\theta_{n_r} + \delta$ angular span exemplar is replicated n_r times around the model’s central axis. The overlap between exemplars is generated by the additional δ angular span. In both cases, no alignment process is required since exemplars are symmetric.

While analogous exemplar copying can be used to fill in holes in the geometry (provided the symmetric geometry exists), we focus our efforts on extending the models’ borders and generating complete 3D models.

5.2. Model Zippering

The model and exemplar mesh fragments are unified via our custom zippering procedure. Our zippering method, like [14] and [29], warps the exemplars for a smooth zippered seam. This warping helps alleviate issues with minor asymmetries in the object.

First, model points in and near the overlap are interpolated with their closest exemplar point to provide a smooth transition between model and exemplar geometry (Figure 4a). Interpolation weights are determined with respect to the points’ distances from the edges of the overlap region. For a more gradual transition from exemplar geometry to model geometry, additional model points near the overlap region are interpolated with respect to an estimated local proxy geometry. This local proxy geometry is designed as an extension to the local topology of the exemplar. For example, in a flat overlap region, a plane may be used. For cylindrical objects (e.g., ‘Small Vase’ in Figure 5a), a surface-of-

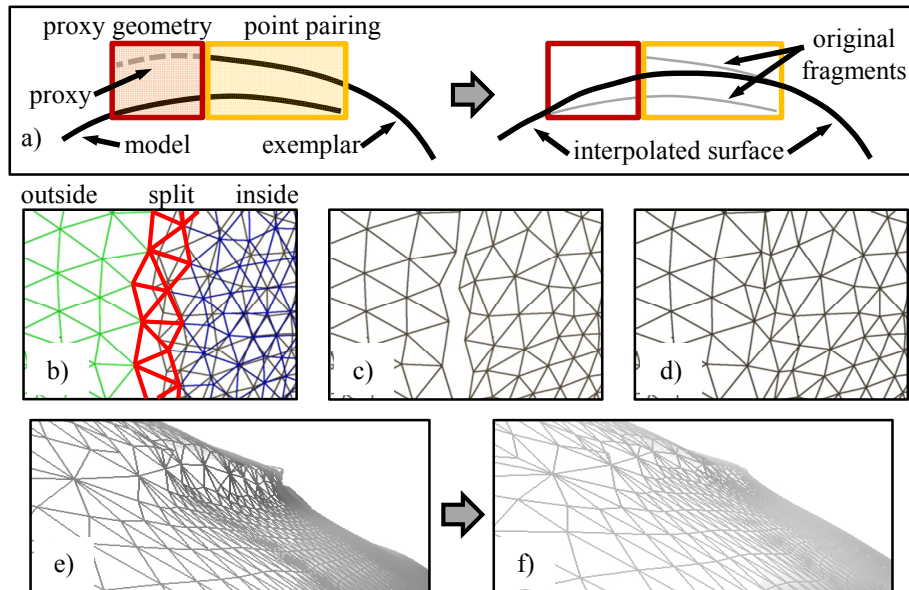


Figure 4: Model Completion. a) Interpolation in the overlap region. Proxy geometry is used in the red box and point pairing is used in the orange box. b) Classification of model (left) and exemplar triangles (right): green = outside exemplar triangles, blue = inside exemplar triangles, red = split exemplar triangles, and black = model triangles. c) The same area after discarding triangles. d) Model and exemplar zippered together. e) Example zippered region without using local proxy geometry to extend the interpolation region. f) Zippered region using local proxy geometry to smooth out a bump.

revolution may be used. Figure 4d-e shows the impact of using proxy geometry to expand a transition region on ‘Small Vase’. Here, the proxy geometry is approximated with a surface-of-revolution. To ensure smoothness, we apply an iteration of Laplacian smoothing after interpolation.

The copied exemplar triangles are then classified as one of three cases (Figure 4b-c): 1) triangles *inside* the model (completely overlapping the model triangles) are discarded since model geometry already exists, 2) triangles *outside* the model (completely non-overlapping the model triangles) introduce new geometry and are retained, and 3) triangles which *split* the model at the exemplar-

model boundary (part of the triangle is outside and part of the triangle is inside) are removed. Each exemplar triangle is classified by computing a local plane approximation of the triangle and projecting the triangle and the model's nearby triangles onto the plane. If the exemplar triangle's three vertices all project into a projected model triangle, the triangle is considered *inside* the model. Triangles with no vertex projecting into a model triangle are considered *outside* the model, and all remaining triangles are considered to be *splitting* the model.

Lastly, the edges of *outside* triangles which were shared with a *split* triangle are connected to the edges of nearby model triangles. First, both model and exemplar edges are organized into groups of ordered, connected edges. Then, for each group of exemplar edges, we walk up the edge, finding the next nearest model edge and triangulating between the two edges (Figure 4d). In each walking step, an exemplar or model edge is advanced depending on which walk minimizes the gap distance between exemplar and model points along the current edges. Thin, elongated triangles are avoided by preventing the gap from getting too large. If an elongated triangle is produced, the exemplar edge group is cut, and the remainder is processed as a new connected edge group. This assures that the triangles added have similar edge lengths to those of neighboring triangles.

5.3. Model Closing

We define two ways to close rotationally symmetric and surface-of-revolution models: 1) we cap the top and bottom of the model, and 2) we provide volume by connecting an inset copy of the model to the original model.

To cap the top or bottom of a mesh, a plane is placed perpendicular to the central axis and automatically moved into place near the top or bottom of the mesh. The intersection of the plane and mesh should define a closed-loop contour

Object	Symmetry Type	Captured Points	Captured Triangles	Final Points	Final Triangles	Ave. Edge Length (mm)
Small Vase	Rotational	109.7k	218.6k	904.1k	1,808.3k	0.2835
Saint	Rotational	159.0k	316.8k	671.4k	1,342.3k	0.3950
Pedestal	Rotational	133.0k	264.7k	336.0k	671.7k	0.4440
Smooth Vase	S. of R.	78.5k	156.5k	72.4k	144.7k	1.0226
Tall Pedestal	S. of R.	106.5k	212.1k	72.0k	144.0k	1.2261
Urn	S. of R.	103.9k	206.7k	56.5k	113.0k	1.5620
Buddha	Bilateral	203.9k	406.4k	272.5k	543.2k	0.2864
Five-Object	Multiple	586.7k	1,168.8k	2,055.9k	4,111.1k	0.4056
Three-Object	S. of R.	658.0k	1,312.9k	169.6k	339.1k	1.5981

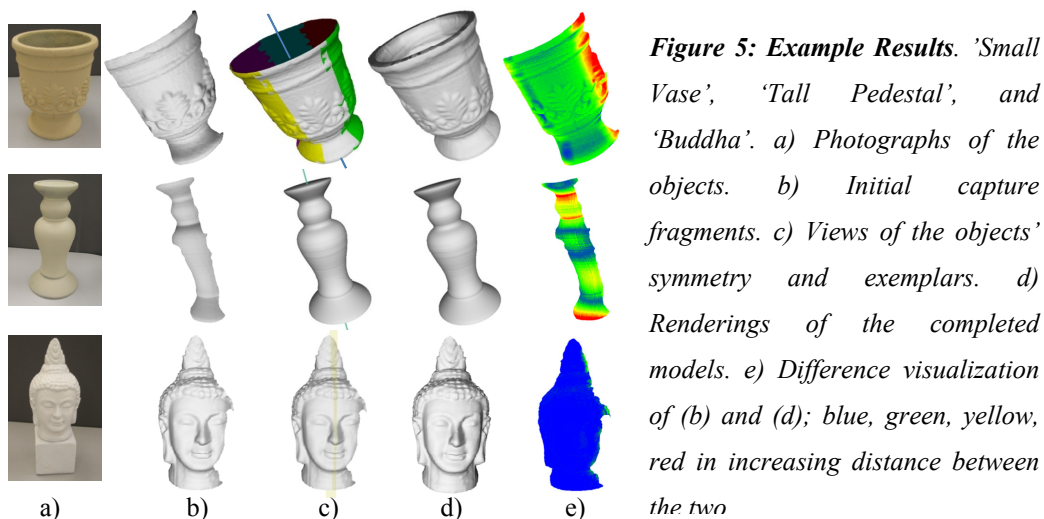
Table 1: A summary of the example datasets we acquired, completed, and inspected.

around the model which culls minimal geometry. Triangles intersecting the plane are clipped, and the resulting planar contour is triangulated in 2D.

An inset of the mesh is created by copying all the triangles of the model and uniformly moved them towards the central axis. Each inset edge is paired with its corresponding original model edge, creating a quadrilateral which is triangulated to connect the model with the inset.

6. Results and Discussion

We have implemented a prototype of our method in C/C++. To project structured-light patterns and capture images, we use 1400x1050 Optoma EP910 projectors and a 10MP Canon Digital Rebel XTi camera. We applied our method to the automatic modeling and completion of several real-world objects possessing symmetry. Details of the acquired, completed, and inspected objects are in Table 1. The average triangle edge length was computed to measure the level of detail of each of the completed models. In general, we achieved triangles with edges lengths of about 0.35mm. The surface-of-revolution models show longer edge lengths since they were synthetically generated. Nonetheless, since



the profile curve accurately represents the surface, these models still exhibit all the original surface details. All initial captures were acquired from a single viewpoint. Symmetry detection and model completion took approximately 20-30 minutes for bilaterally symmetric objects, 60-70 minutes for rotationally symmetric objects, and 10-15 minutes for surfaces-of-revolution.

Our framework provides several digital inspection tools for interactively focusing on surface details as shown in Figures 1 and 8. The inspection tools enable instant relighting, shading exaggeration, depth-based effects, object slicing, and iso-distance curves, similar to those presented in [1, 19].

Figure 5 shows the steps taken to achieve our final models for each type of symmetry. Figure 5a shows photographs of the objects, Figure 5b shows the initial captures of the objects, Figure 5c shows the discovered symmetry characteristics and exemplars, and Figure 5d shows the resulting complete models. For 'Small Vase' and 'Tall Pedestal', we build complete 3D models. For 'Buddha', we complete the left side of its face (particularly the left ear). We omit the Buddha's base during computations since its geometry is trivial. Figure 5e

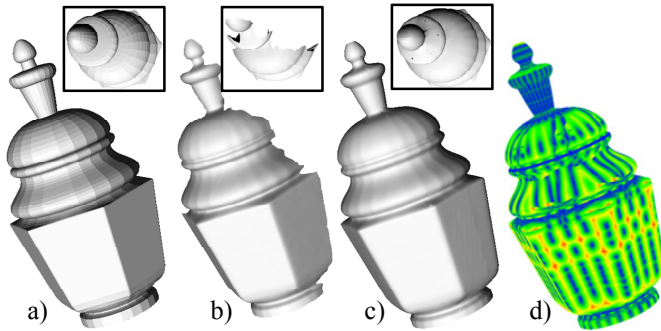


Figure 6: Model Error. Comparison of our model against its ideal target. a) Ideal target model. b) Initial capture fragment. c) Our complete model. d) Visualization of reconstruction error.

shows a visualization of the difference between the initial capture geometry and the completed model geometry. The sequence blue, green, yellow, to red shows an increasing distance between the capture fragment and the final model ranging from 0mm to 4mm (distances above 3mm are clamped to red). The completed ‘Buddha’ (bilateral symmetry) was nearly identical to its original fragment since it only underwent minimal warping during zippering. ‘Small Vase’ (rotational symmetry) underwent mild warping due to the exemplar warping optimization and stronger warping at the overlap regions due to the zippering routine. ‘Tall Pedestal’ (surface-of-revolution symmetry) contained the most differences between its original fragment and its completed model since the initial fragment was susceptible to global deformations. Nonetheless, for all symmetry classes, a reasonable, complete model is constructed.

Figure 6 illustrates our method’s reconstruction accuracy. Using a synthetic model of a rotationally symmetric object (Figure 6a), we 3D-printed the object and captured it from a single viewpoint (Figure 6b). Then, we used our method to construct a complete model of the object (Figure 6c). The insets to the upper right of Figures 6a-c show a different view of the models. The per-point difference between our completed model and that of its synthetic target is shown in Figure 6d (a color map similar to that in Figure 5e is used but with a range of 0mm to 10mm). We estimate the accuracy of our acquisition system – and thus the

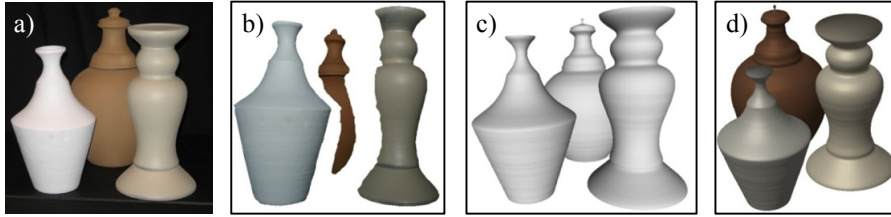


Figure 7: Multi-Object Scene. Three objects with surface-of-revolution symmetry. a) Scene photograph. Left to right, ‘Smooth Vase’, ‘Urn’, and ‘Tall Pedestal’. b) Single viewpoint capture fragments. c) Synthetic rendering of completed models. d) Texture-mapped rendering.

accuracy of acquiring the complete object from multiple viewpoints – at a 2mm error range (using 1024x1024 projector pixels, 15° between object-camera and object-projector directions, and 1m object-camera distance). The wavy error-pattern of the visualization’s colors is due to the acquisition method. Our model completion’s average reconstruction error was 3.6mm. Thus, while our approach does not capture the completed model at the same accuracy as structured-light, its accuracy is similar.

Figure 1 shows an example of our system’s ability to generate complete models of a multiple object scene. We arranged five objects exhibiting rotational and surface-of-revolution symmetry and performed a single viewpoint initial capture (Figures 1a-b). Figure 1c shows a rendering of the completed models, and Figure 1d shows the models texture-mapped (the captured scene image used for texture-mapping is given a matte-tone via image processing). Figure 1e shows our interactive digital inspection tools applied on the scene.

Figure 7, another multi-object capture, demonstrates the robustness of our surface of revolution symmetry detection. Figure 7a shows a photograph of the three objects, left to right ‘Smooth Vase’, ‘Urn’, and ‘Tall Pedestal’. From the capture viewpoint, only a thin slice of the ‘Urn’ object is visible. Nonetheless, we are able to reconstruct a complete model as shown in Figures 7c-d.

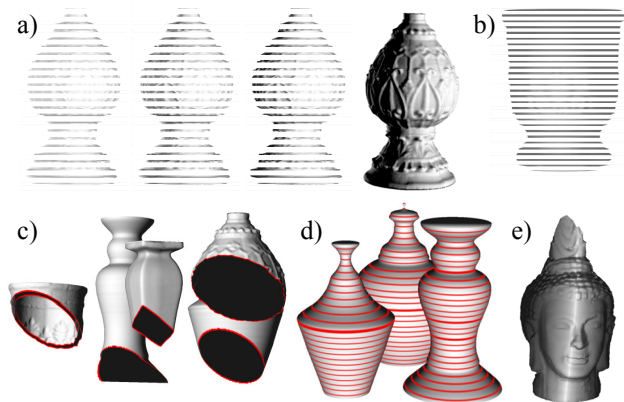


Figure 8. Digital Inspection.
 Visualizations produced by our methods. a) Shading exaggeration from mild to strong. b) Depth-based detail modulation. c) Depth-based object slicing of several objects. d) Iso-distance line rendering of several models. e) Depth-based illumination.

Figure 8 shows several interactive digital inspection techniques from [1] applied to our completed models. Figure 8a shows mild to strong shading exaggeration. Figure 8b is an example of depth-based detail modulation where the details further away are greyed out. Figure 8c demonstrates a depth-based effect by slicing several objects with a plane. Figure 8d shows iso-distance lines on our three surfaces of revolution. Finally, Figure 8e shows ‘Buddha’ with depth-based illumination, where depth is defined as the distance from a user-oriented plane.

Limitations. Our system does have some limitations. The symmetry detection optimizations may fall into a local minima and discover the wrong plane or axis. For our objects, this rarely occurred. Automatic detection for this case is not provided but rather left to interactive feedback for suggesting new initial values. Such manual intervention does not take more than one minute.

The use of interpolation and a small amount of smoothing during zippering may cause some small details to be blurred in exchange for a smoother transition from patch to patch. Also, the effectiveness extending the transition region depends on the accuracy of the proxy geometry used.

Lastly, the single viewpoint capture may contain discontinuities or holes within it. The presence of these artifacts may distort the calculation of symmetry patches. It is worth noting that our approach can also be used to complete such exemplars provided the geometry exists in other captured exemplars.

7. Conclusions and Future Work

We have proposed a technique for capturing and reconstructing symmetric objects from a single viewpoint yielding plausible complete models for use in digital inspection. Using an optimization-based technique, we discover and transfer the symmetries of the geometric structure and zipper together the resulting model. We do not require searches for patch geometry and are able to produce entire models from fragments. While our pipeline is not fully automatic, user intervention is limited to selecting the symmetry case and setting a few parameters. Complete 3D models enrich interactive digital inspection renderings and provide the user with a more compelling experience for analyzing the objects.

Aside from digital inspection, we expect this tool to be useful for a variety of end-users. For future work, we wish to explore the use of this system with archaeologists who often want to quickly capture and complete an object. By adding a fast feedback loop (e.g., low-resolution processing), a user can rapidly predict the reconstruction and change the viewpoint or take additional views as needed until a satisfactory object is obtained. We are also interested in supporting non-Lambertian objects – this requires enhancements to the capture process and to the ability for extracting the true (diffuse) surface albedo from captured images. We would also like to support multiple symmetries on an object and expand our class of supported symmetries.

References

- [1] D. Aliaga, Digital Inspection: An Interactive Stage for Viewing Surface Details, in: Proc. of ACM Sym. on Interactive 3D Graphics, 53-60, 2008.
- [2] D. Aliaga, Y. Xu, A Self-Calibrating Method for Photogeometric Acquisition of 3D Objects, in: IEEE Transactions on Pattern Analysis and Machine Intelligence, to appear, 8 pages, 2009.
- [3] P.J. Besl, N.D. McKay, A Method for Registration of 3-D Shapes, in: IEEE Trans. on Pattern Analysis and Machine Intelligence, 239-256, 1992.
- [4] B.J. Brown, S. Rusinkiewicz, Global Non-Rigid Alignment of 3D Scans, in: Proc. of ACM SIGGRAPH, ACM Trans. on Graphics, 26(3), 2007.
- [5] T.K. Dey, S. Goswami, Tight Cocone: A Water Tight Surface Reconstructor, in: Proc. of ACM Sym. on Solid Modeling and Applications, 127-134, 2003.
- [6] D. Aiger, N.J. Mitra, D. Cohen-Or, 4-Points Congruent sets for Rubust Pairwise Surface Registration, in: Proc. of ACM SIGGRAPH, ACM Trans. on Graphics, 27(3), 2008.
- [7] T. Funkhouser, M. Kahzdan, P. Shilane, P. Min, W. Kiefer, A. Tal, S. Rusinkiewicz, D. Dobkin, Modeling by Example, in: Proc. of ACM SIGGRAPH, ACM Trans. on Graphics, 23(3), 652-663, 2004.
- [8] L. Ikemoto, N. Gelfand, M. Levoy, A Hierarchical Method for Aligning Warped Meshes, in: Proc. of 3D Digital Imaging and Modeling, 434-441, 2003.
- [9] I.K. Lee, Curve Reconstruction From Unorganized Points, in: Computer Aided Geometric Design, (17)2, 161-177, 2000.

- [10] M. Levoy, K. Pulli, B. Curless, S. Rusinkiewicz, D. Koller, L. Pereira, M. Ginzton, S. Anderson, J. Davis, J. Ginsberg, J. Shade, D. Fulk, The Digital Michelangelo Project: 3D Scanning of Large Statues, in: Proc. of ACM SIGGRAPH, ACM Trans. on Graphics, 131-144, 2000.
- [11] N.J. Mitra, S. Flory, M. Ovsjanikov, N. Gelfand, L. Guibas, H. Pottmann, Dynamic Geometry Registration, in: Proc. of Sym. on Geometry Processing, 173-182, 2006.
- [12] N.J. Mitra, L.J. Guibas, M. Pauly, Partial and Approximate Symmetry Detection for 3D Geometry, in: Proc. of ACM SIGGRAPH, ACM Trans. on Graphics, 25(3), 560-658, 2006.
- [13] N.J. Mitra, L.J. Guibas, M. Pauly, Symmetrization, in: Proc. of ACM SIGGRAPH, ACM Trans. on Graphics, 26(3), 1-8, 2007.
- [14] M. Pauly, N.J. Mitra, J. Giesen, M. Gross, L.J. Guibas, Example-based 3D Scan Completion, in: Proc. of ACM Sym. on Geometry Processing, 23-32, 2005.
- [15] M. Pauly, N.J. Mitra, J. Wallner, H. Pottmann, L.J. Guibas, Discovering Structural Regularity in 3D Geometry, in: Proc. of ACM SIGGRAPH, ACM Trans. on Graphics, 27(3), 1-11, 2008.
- [16] J. Podolak, P. Shilane, A. Golovinskiy, S. Rusinkiewicz, T. Funkhouser, A Planar-Reflective Symmetry Transform for 3D Shapes, in: Proc. of ACM SIGGRAPH, ACM Trans. on Graphics, 25(3), 549-559, 2006.
- [17] M. Pollefeys, L. Van Gool, M. Vergauwen, F. Verbiest, K. Cornelis, J. Tops, R. Koch, Visual Modeling with a Hand-held Camera, in: International Journal of Computer Vision, 59(3), 207-232, 2004.

- [18] H. Pottmann, T. Randrup, Rotational and Helical Surface Approximation for Reverse Engineering, in: *Computing (Journal)*, 60(4), 307-322, 1998.
- [19] S. Rusinkiewicz, M. Burns, D. DeCarlo, Exaggerated Shading for Depicting Shape and Detail, in: *Proc. of ACM SIGGRAPH, ACM Trans. on Graphics*, 25(3), 1199-1205, 2006.
- [20] S. Rusinkiewicz, O. Hall-Holt, M. Levoy, Real-Time 3D Model Acquisition, in: *Proc. of ACM SIGGRAPH, ACM Trans. on Graphics*, 21(3), 438-446, 2002.
- [21] S. Rusinkiewicz, M. Levoy, Efficient Variants of the ICP Algorithm, in: *Proc. of 3D Digital Imaging and Modeling*, 145-152, 2001.
- [22] D. Scharstein, R. Szeliski, High-Accuracy Stereo Depth Maps Using Structured Light, in: *Proc. of IEEE Computer Vision and Pattern Recognition*, 1, 195-202, 2003.
- [23] W. Scott, G. Roth, J.F. Rivest, View Planning for Automated Three-Dimensional Object Reconstruction and Inspection, in: *ACM Computing Surveys*, 35(1), 64-96, 2003.
- [24] I. Sebe, Y. Suya, U. Neumann, Rapid Part-based 3D Modeling, in: *Proc. of ACM Sym. on Virtual Reality Software and Technology*, 143-146, 2005.
- [25] A. Sharf, M. Alexa, D. Cohen-Or, Context-based Surface Completion, in: *Proc. of ACM SIGGRAPH, ACM Trans. on Graphics*, 23(3), 878-887, 2004.
- [26] P. Simari, E. Kalogerakis, K. Singh, Folding Meshes: Hierarchical Mesh Segmentation Based on Planar Symmetry, in: *Proc. of Eurographics Sym. on Geometry Processing*, 111-119, 2006.

- [27] A. Tagliasacchi, H. Zhang, D. Cohen-Or, Curve Skeleton Extraction from Incomplete Point Cloud, in: Proc. of ACM SIGGRAPH, ACM Trans. On Graphics, (28)3, 2009.
- [28] S. Thrun, B. Wegbreit, Shape from Symmetry, in: Proc. of IEEE International Conference on Computer Vision, 1824-1831, 2005.
- [29] G. Turk, M. Levoy, Zippered Polygon Meshes from Range Images, in: Proc. of ACM SIGGRAPH, ACM Trans. on Graphics, 311-318, 1994.
- [30] Y. Yu, K. Zhou, D. Xu, X. Shi, H. Bao, B. Guo, H.Y. Shum, Mesh Editing With Poisson-Based Gradient Field Manipulation, in: Proc. of ACM SIGGRAPH, ACM Trans. on Graphics, 23(3), 644-651, 2004.
- [31] H. Zabrodsky, D. Weinshall, Using Bilateral Symmetry to Improve 3D Reconstruction from Image Sequences, in: Computer Vision and Image Understanding, 67(1), 48-57, 1997.



Alvin J. Law is a Ph.D. student in Computer Science at Purdue University. He is a member of the Computer Graphics and Visualization Laboratory, and his research interests are in appearance editing and 3D modeling. He obtained a B.S. from Cornell University and a M.S. from Purdue University.



Daniel G. Aliaga is an Associate Professor of Computer Science at Purdue University. He is a researcher in computer graphics and computer vision, and in particular in acquiring, modeling, and rendering 3D objects and scenes. Dr. Aliaga obtained his Ph.D. degree from the University of North Carolina (UNC). He has served on numerous program committees, on several NSF panels, as journal editor, as conference and paper chair, and authored over 60 papers.