Folding meshes: Hierarchical mesh segmentation based on planar symmetry

Technical report CSRG-536, Dynamic Graphics Project, Department of Computer Science, University of Toronto,

Patricio Simari* Evangelos Kalogerakis† Karan Singh‡

Dynamic Graphics Project
University of Toronto
Toronto, Canada

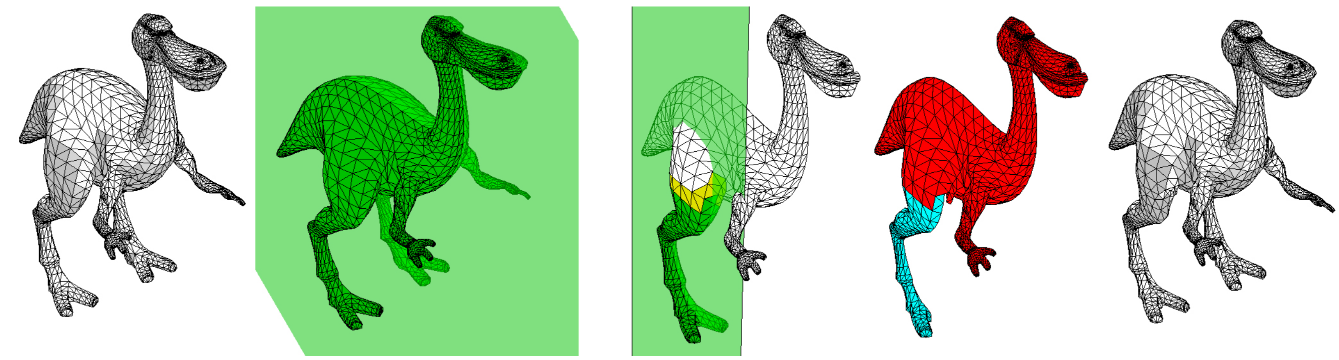

Figure 1: Left to right: **a)** Original dino mesh. **b)** Detected global symmetry plane. **c)** Remaining half dino mesh with a detected local symmetry. The darker shade (green in the electronic version) indicates the region of symmetry and the lighter shade (yellow in the electronic version) indicates faces still in the support region but not included in the symmetric region. **d)** Remaining leaf geometry (the two leaves shaded a different color). **e)** Dino reconstructed from the leaf geometry and symmetry planes.

Abstract

Meshes representing real world objects, both artist-created and scanned, contain a high level of redundancy due to planar reflection symmetries, either global or localized to different subregions. An algorithm is presented for detecting such symmetries and segmenting the mesh into the symmetric and remaining regions. The method, inspired by techniques in Computer Vision, has foundations in robust statistics and is resilient to structured outliers which are present in the form of the non symmetric regions of the data. Also introduced is an application of the method: the *folding tree* data structure. The structure encodes the non redundant regions of the original mesh as well as the reflection planes and is created by the recursive application of the detection method. This structure can then be *unfolded* to recover the original shape with bounded error, which is user specified in the folding process. Applications include mesh compression, repair, skeletal extraction of objects of known symmetry as well as mesh processing acceleration by limiting computation to non redundant regions and propagation of results.

CR Categories: I.3.5 [Computational Geometry and Object Modeling]: Geometric algorithms, languages, and systems—Curve, surface, solid, and object representations

Keywords: mesh segmentation, symmetry, compression, folding meshes

*e-mail: psimari@dgp.toronto.edu
†e-mail: kalo@dgp.toronto.edu
‡e-mail: karan@dgp.toronto.edu

1 Introduction

Symmetry plays a fundamental role in nature, manifested both in the form and function of living organisms. Visually, symmetry is important to humans, as it influences our perceptual understanding of objects in the world. Symmetric patterns, not surprisingly, are an important design principle in guiding the aesthetic and construction of synthetic objects [Arnheim 1954; Gombrich 1969]. Neuroscience research goes so far as to indicate that aspects of symmetry in humans may be hard-wired into our visual processing system [Norcia et al. 2002]. Symmetries are ubiquitous in humans, our environment and our creations of art and architecture.

The classification, understanding and intelligent representation of shape is an active area of research in geometry processing. Recognizing the common presence of symmetries in many real world objects can greatly assist in solutions to various shape representation problems such as simplification, repair, noise removal and skeletal extraction. Of the various types of symmetries found, planar symmetry is perhaps the most commonplace and is thus the focus of this paper. While planar symmetries have been recognized to be an important feature in shape understanding, there has been little work in shape representations that are defined as a structured assembly of symmetric parts. In this paper we present the concept of a folding tree (see figure 3), where an object is defined as a hierarchical union of planar symmetric and asymmetric parts. Each (possibly nested) detection of symmetric part reduces the complexity of representation of said part in half, greatly reducing the representation complexity of many objects.

For folding trees to be useful beyond an academic concept, we must be able to automatically construct them from geometric data such as meshes as well as regenerate the object from its folding tree. Central to folding tree construction is the problem of automatically finding a maximally symmetric part of the object. We observe that most organic objects with planar symmetry are articulated figures with coherent symmetric parts that are connected at the joints of an underlying skeletal structure. Most synthetic objects show a construction history involving symmetric primitives, symmetry creating operations such as reflections, planar symmetry preserving operations like revolves and coherent combinations of various symmetric parts often with some asymmetric refinement. Motivated by these observations we additionally constrain our problem to finding a maximal symmetric part that is a single connected surface component of the object. The constraint has several advantages, including simplifying reconstruction. Multiple surface components with the same planar symmetry are just as easily represented as multiple symmetry nodes that have the same symmetry plane.

The approach to finding a maximally symmetric part is based on an iteratively reweighted least squares algorithm where we simultaneously optimize candidate reflection planes and the region of surface that is symmetric with respect to it. Upon convergence, the overall set of faces is partitioned into the symmetric subset and the remaining faces. Symmetric regions can be represented by the plane of reflection, half the geometry and a segmentation boundary. The algorithm is recursively applied to half the symmetric subset as well as the remaining regions to construct a folding tree decomposition of the object. Object geometry is thus only stored in tree leaves. We then show how the mesh is reconstructed from its folding tree from the bottom up by reflecting symmetric geometry and stitching along segmentation boundaries.

This paper presents two principal contributions. Firstly, we introduce a method capable of detecting global as well as local symmetries in subparts of 3D meshes. Our algorithm has solid foundations on statistical methods for robust iteratively reweighted least squares estimation [Hampel et al. 1986] which has also been used in many recent computer vision applications [Sawhney et al. 1995; Black and Anandan 1996; Stewart 1999]. Secondly, we exploit our symmetry detection approach for *mesh folding*: the elimination of planar symmetry redundancy from the mesh data. Our algorithm is orthogonal to other existing methods for mesh compression [Hoppe 1996; Gueziec et al. 1999; Karni and Gotsman 2000; Isenburg and Alliez 2002] which do not explicitly take advantage of repeating symmetric areas in 3D meshes.

The rest of the paper is organized as follows: section 2 describes the related work on symmetry detection and applications; section 3 describes our method for detecting global and local symmetries on mesh data; section 4 introduces folding trees and describes their construction; section 5 shows our results; and finally, section 6 presents our conclusions and future work.

2 Related Work

Although the computation of symmetries in shapes has been an intriguing area of research in computer vision and computational geometry literature for the last 30 years at least, to our knowledge, there has been little research on symmetry detection in 3D meshes mainly due to the increased complexity of the existing algorithms when extended from the 2D to the 3D case. Our work differs from the few existing approaches for 3D symmetry detection in meshes, in that, we aim at the robust detection of global or local reflection symmetries in parts of a 3D mesh and we exploit these symmetries in order to achieve mesh compression by eliminating faces implied by the discovered symmetry.

The detection of symmetries in 2D and 3D models has mainly been applied in object classification, recognition and reconstruction. Early approaches dealt with symmetries of planar point sets by applying pattern recognition algorithms that search matches in circular strings representing the graphs of polyhedral objects [Atallah 1985; Wolter et al. 1985; Highnam 1986; Jiang et al. 1996].

Despite the optimality of these algorithms that could also detect all the possible symmetries in a shape, they were only able to recover perfect symmetries in 2D and 3D shapes making them useless in the presence of small perturbations, imprecision and noise which is very common in meshes. The problem of approximate symmetries was addressed in [Alt et al. 1988] that considered the problem of computing generic geometric transformations between two point sets. The paper gives a detailed theoretical analysis of the developed algorithm for symmetry, however, it deals with global symmetry and it is unclear if the given algorithm could be extended to three dimensions. Such is also the case of other methods for finding symmetries of symmetric or almost symmetric 2D planar images [Marola 1989; Gofman and Kiryati 1996; Shen et al. 1999].

An interesting extension of that early work which introduced the notion of symmetry distance, meaning how much of a given symmetry an object possesses, was developed in [Zabrodsky et al. 1995]. The approach can evaluate symmetries in the presence of noise and also find locally symmetric regions in 3D objects which were represented as images. The reflection plane of the 3D object is determined by minimizing the symmetry value over all possible reflection planes using a gradient descent algorithm to efficiently locate the plane of maximal symmetry. However, noise and digitization errors create convergence problems that require a coarse-to-fine estimation of symmetry from low to high resolution images. Another recent approach having 3D range images as input was presented in [Thrun and Wegbreit 2005]. A probabilistic measurement model is used to detect global and local symmetries in order to reconstruct partially occluded 3D shape models. Both of the above-mentioned Computer Vision approaches require a 3D image-based representation of the objects which requires uniform sampling of surfaces. This is impractical in computer graphics applications that use surfaces, like 3D meshes, that usually have non-uniform tessellation.

Another original approach that detects the dominant hyperplane of bilateral symmetry in images of 3D objects with a linear time algorithm is presented in [Mara and Owens 1996]. The hyperplane is uniquely defined by the centroid and eigenvectors of the covariance matrix of the object. Although this method is related to ours, it is limited to the detection of the plane of global symmetry and is not robust to outliers or imprecision in the 3D object. Shape descriptors representing a 3D model by a function defined on a canonical domain were used in [Sun and Sherrah 1997] where the correlation of the extended gaussian image of the object is exploited to solve the symmetry detection problem. Although the presented methodology could be used in a wide class of shape descriptors, it suffered from the same limitations. Shape descriptors are also introduced in [Kazhdan et al. 2004a; Kazhdan et al. 2004b] where a collection of spherical functions are used to describe the measure of rotational and reflective symmetry present in a mesh with respect to every axis passing through its center of mass. The descriptors had several desirable properties such as robustness and stability. However, the approach aimed at using symmetry information as a shape descriptor and not at extracting local symmetries.

3 Symmetric region detection

Given a mesh, we wish to find a connected region S of faces that exhibit planar symmetry within tolerance parameter ε. In the case of global symmetry, this region should be the entire mesh. We approach the problem as a model fitting scenario, in which the model consists of the sought plane, and the connected region of symmetry.

Given the presence of structured outliers in the form of the non symmetric regions of the mesh, we interleave solving for the symmetric region S and the plane p based on an iteratively reweighted least squares (IRLS) approach, using the Geman-McClure (GM) robust M-estimator. [Hampel et al. 1986; Forsyth and Ponce 2002]. The GM estimator exhibits excellent behavior in rejected structured outliers with the appropriate choice of the scale factor σ [Sawhney et al. 1995].

Distance metric: given a plane p, we denote the distance from a point r_i, which is the reflection of v_i with respect to p, to a mesh M as $d_i = dist(r_i, M)$.

We compute the distance function *dist* from a reflected vertex to the mesh by taking the minimum point-to-triangle distance from the point to the closest compatible face on mesh M [Rusinkiewicz and Levoy 2001]. We consider a face to be compatible with a given query vertex if the angle between the interpolated normal at the closest point on the face and the vertex's normal is less than 45 degrees.

Solving for the plane: Given the current distances d_i, the GM cost estimator ρ_i and associated weight w_i for each vertex are given by

$$\rho_i = \frac{d_i^2}{\sigma^2 + d_i^2}$$

$$w_i = \frac{1}{d_i}\frac{\partial}{\partial d_i}\rho_i$$

In addition, in order to be robust in the presence of tessellations with varying face sizes, we multiply the obtained weights by their associated vertex areas, i.e. $w_i \leftarrow w_i \frac{1}{3}\sum_{j=1}^{k} area(f_j)$, where $f_1, \ldots, f_k$ are the faces incident on vertex v_i.

For a body which exhibits planar symmetry it is known that its plane of symmetry is perpendicular to a principal axis and contains the object's center of mass [Minovic et al. 1992]. This lets us solve for the current plane of maximum symmetry in a closed form manner by considering the center of mass m and weighted covariance matrix C relative to the weights w_i.

$$m = \frac{1}{s}\sum_{i=1}^{n} w_i v_i$$

$$C = \frac{1}{s}\sum_{i=1}^{n} w_i (v_i - m)(v_i - m)^T,$$

where $s = \sum_i w_i$.

We compute the eigenvectors of C and consider the three planes determined by these vectors and m. For each of these planes we compute the distances d_i and associated costs ρ_i retaining the one of minimum sum cost. This now lets us solve for the support region.

Support region: Given the current ρ values and a candidate face $f = (v_1, v_2, v_3)$ we consider it to be a *support* face if it holds that $\forall_{i\in\{1,2,3\}} d_i \leq 2\sigma$ [Hampel et al. 1986]. We then find the largest connected region of support faces, taking this as our new estimate of S and set the weights for all vertices outside this region to be 0.

The estimation and region finding steps are iterated until convergence.

Initialization and details: Initially we simply define w_i to be the mesh area associated with vertex v_i as defined above, and the initial support region contains all faces.

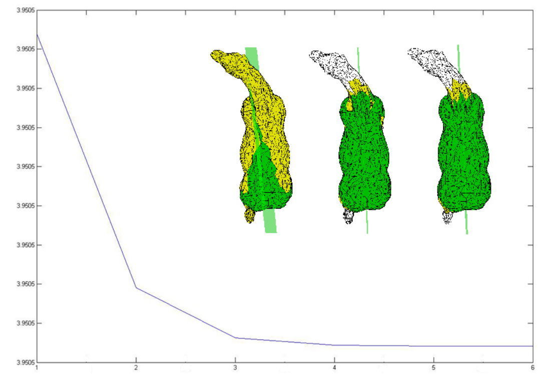

Figure 2: Illustration of algorithm convergence. The plot shows the $\sum_i \rho_i$ for vertices $v_i \in S$ during the fine iterations. The placement of the estimate of the symmetry plane along with support region shaded lightly (yellow in electronic version) and symmetric region shaded darkly (green in the electronic version) for the base segmentation of the horse model. Left to right, iterations 1, 5 and 10 respectively.

Coarse-to-fine symmetry search In order to avoid local minima and accelerate convergence we also use a discrete approximation of the above distance function for a coarse-to-fine approach. For a given vertex v_i and face f_j this distance function $dist_{coarse}(v_i, f_j)$ is defined as the Euclidean distance from v_i to the face plane of f_j if the angle between the normals of v_i and f_j is less than 90 degrees, and infinite otherwise. The distance $dist_{coarse}(v_i, M)$ is defined as the distance to the f_j whose centroid is closest to v_i. Also, during these coarse iterations we set $\sigma = 1.4826 \cdot median(d_i)$, which is a popular estimate of scale [Forsyth and Ponce 2002], not letting it fall below 3ε to avoid instability.

During the fine iterations we set $\sigma = 3\varepsilon$. This setting allows for near-symmetric vertices to be included in the support region albeit with lower weight, and helps convergence.

The coarse distance function is first used until convergence to a local minima, after which the more precise distance function is used. Upon convergence, the symmetric region S is extracted as the largest connected region of faces whose vertex distance values are all below ε. We detect convergence by comparing the current plane estimate with that of the previous iteration checking for a sufficiently small difference.

Convergence: In our experiments both distance functions exhibit very good convergence behavior (to their respective minima). Figure 2 illustrates an example of the convergence properties of our approach.

4 Folding trees

4.1 Definitions

We consider a region R of a mesh M to be a connected subset of the faces of M.

A segmentation $\{R_1, R_2, \ldots, R_n\}$ of a mesh M is a set of mutually exclusive regions whose union results in M.

A folding tree T representing a mesh M is inductively defined as one of the following:

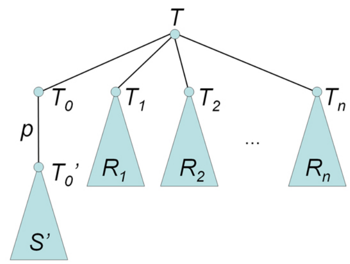

Figure 3: Folding tree construction structure.

- a *leaf* node, which contains mesh data for M.
- a *folding* node, with one subtree S, and a plane of symmetry p.
- a *segmentation* node, with n subtrees $T_1, T_2, ..., T_n$, where T_i is a folding tree for region R_i such that regions $R_1, R_2, ..., R_n$ are a segmentation of the mesh represented by T.

The *unfold* operation can now be defined on a folding tree T as follows.

- The mesh data M if T is a leaf.
- $unfold(S) \cup reflect(unfold(S), p)$, where S is the unique subtree of T, p is the reflection plane,
- $\cup_{i=1}^{n} unfold(T_i)$ where $T_1, T_2, ..., T_n$ are the subtrees of T.

Here, *reflect* indicates the mesh resulting from planar reflection of the argument mesh's vertices with respect to the argument plane.

4.2 Folding tree construction

Given a mesh M, a folding tree T that represents it can be constructed in preorder through repeated application of the segmentation method of section 3. First, we apply the segmentation algorithm to M to find a subregion of planar symmetry, S, also obtaining the plane of symmetry p. We remove S from M and consider the set of remaining connected components $\{R_1, R_2, ..., R_n\}$. Note that in the case of a global symmetry, this set will be empty. We now construct a folding tree T with $n+1$ children $T_0, T_1, ..., T_n$, each representing $S, R_1, R_2, ..., R_n$ respectively. We know S to be symmetric with respect to plane p, so we can now *fold* S, retaining half of its surface S'. In particular, T_0 will be a folding node, labeled with p, and its child T_0' will represent S'. The resulting structure is illustrated in figure 3. The subtrees $T_0', T_1, ..., T_n$ can now be created recursively with regions $S', R_1, ..., R_n$ respectively as inputs.

When discarding half of the faces of a particular region, it must be decided which half to keep, which to discard, and which to modify, if any. Because of varying tessellation and the provided tolerance, both sides need not be identical. In our implementation we count the number of faces on each side of the plane and keep the side with the most faces in order to preserve detail. Alternatively, we could keep the half with less faces in order to minimize storage.

Faces with all vertices on the discarding side of the plane are removed. In addition, the vertices that remain on the discarded side due to straddling faces are projected onto the plane.

It should also be noted that the tolerance ε parameter for the subtree associated with a folding node should be halved. This guarantees that carryover errors during the unfolding process will not cause a violation of the root tolerance.

Mesh	# Orig. f's	#Nonred. f's	Comp.	tree height	Time
Dino	6638	3142	52.7%	3	152 sec
Horse	3306	2672	19.2%	3	24 sec
Chair	5736	2460	57.1%	4	58 sec
Table	5056	1332	73.7%	2	160 sec
Hammer	4360	677	84.5%	7	174 sec
Triceratops	5660	2447	56.8%	7	202 sec
Eagle	33072	15808	52.2%	6	936 sec
Queen	3360	600	82.1%	4	120 sec

Table 1: Results for seven characteristic meshes. Columns from left to right: mesh name, number of faces in the original mesh, number of non redundant faces stored in folding tree leaves, compression achieved according to 1 - ratio of the previous two columns, height of the folding tree, and running time.

The recursive construction of the tree may be stopped, triggering the creation of a leaf node, by using one or more criteria: for example, when the number of faces in the mesh is below a given threshold, when the area of the mesh is below a certain percentage of the original mesh, or when the number of recursive folds exceeds a given maximum. Our implementation allows for any or all three.

4.3 Unfolding and mesh reconstruction

The procedure for unfolding a tree consists of a postorder traversal according to the definition of subsection 4.1. Upon reconstruction, because of the tolerance parameter of the region finding algorithm, as well as differences in tessellation, the resulting mesh may have gaps. In order to avoid a general mesh repairing problem, we implement an approach which consists of labeling vertices as *tear vertices* during the tree construction process at each level of recursion. At a segmentation node a vertex is labeled as a tear vertex if it is on a segmentation boundary. At a folding node, the vertex simply inherits its tear attribute from the previous level. Thus, on the leaf mesh data, a vertex has a stack, the size of the leaf's depth in the tree, indicating if is a tear vertex at each level. After processing a segmentation node, tear paths are found along the mesh, and these are stitched and smoothed using constrained Laplacian smoothing [Funkhouser et al. 2004]. Alternatively, the mesh can be repaired using standard techniques and software [Gueziec et al. 1998; Turk. and Levoy 1994; Ju 2004].

5 Results

The implementation of the symmetry detection algorithm and the folding tree representation of meshes, as described in sections 3 and 4, has been developed in Matlab 7. The user defines the tolerance ε of the algorithm and the criteria for stopping the hierarchical segmentation of the mesh. The default value of ε is 2% of the bounding box diagonal of the mesh. The default criterium for terminating the recursion is that the total area of the current region is less than 5% of the total mesh area. We present characteristic results, concerning mesh compression, the depth of the hierarchical segmentation, initial and reconstructed meshes, as well as running times in table 1 and figures , 4 and 5. Our tests were run on an Intel Pentium M 2.13GHz processor with 1 GB of RAM.

In the chair model, we find the vertical plane of global symmetry then each cushion, which was a separate connected component, was

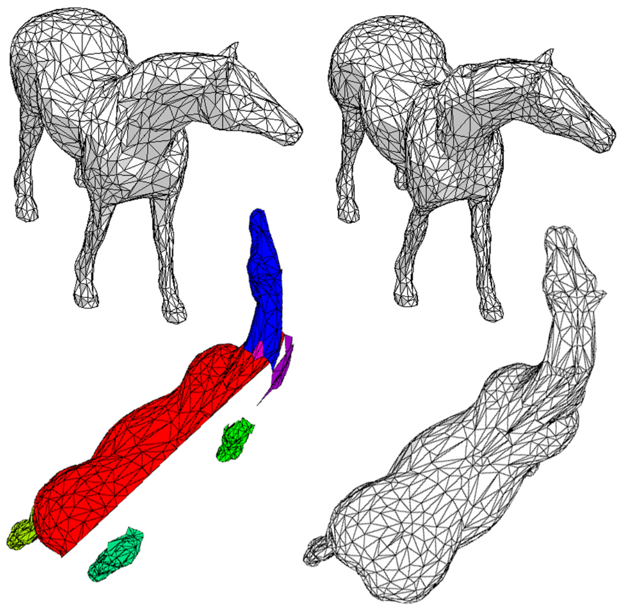

Figure 4: **Top left:** original horse model. **Bottom left:** resulting tree leaves after folding. Note that the local symmetry of the articulated head was detected. **Top right and bottom right:** the reconstructed horse model.

folded through three perpendicular planes. In the case of the table, there are two vertical and orthogonal nested folds. The hammer is firstly folded in half through a vertical global plane of symmetry. Then, the handle ends up being divided twice more. The circular portion of the head is also subdivided three more times recursively. In the case of the triceratops and eagle models simplification is more difficult, but even in these cases we find a global plane of symmetry and then local symmetries in the legs, tail and body for the triceratops model, and in the wings and upper legs of the eagle. Finally in case of the queen chess piece, all planar symmetries are recursively discovered resulting in one eighth of the original surface being stored.

6 Conclusions and future work

In this paper, we have proposed a novel approach for finding and exploiting local and global planar symmetries in 3D meshes. We have presented a new compact representation of meshes, called folding trees, which represent the original mesh by only encoding the non redundant regions as well as the planes of symmetry.

Given the fact that real objects, both organic and synthetic, often exhibit this type of data redundancy and human perception is strongly related to the notion of symmetry, a significant number of applications based on our methodology can further be developed. The elimination of faces which are repeated in redundant areas of global and local symmetries lead to new mesh compression schemes that can be used for mesh storage, processing and transmission. Automatic reconstruction and repairing of the meshes, driven by the extracted symmetries, is also another interesting field of application of our method. The folding tree representation could also be potentially applied to the extraction of skeletons of the represented objects by finding articulations.

Our future research will be focused on both the development of such

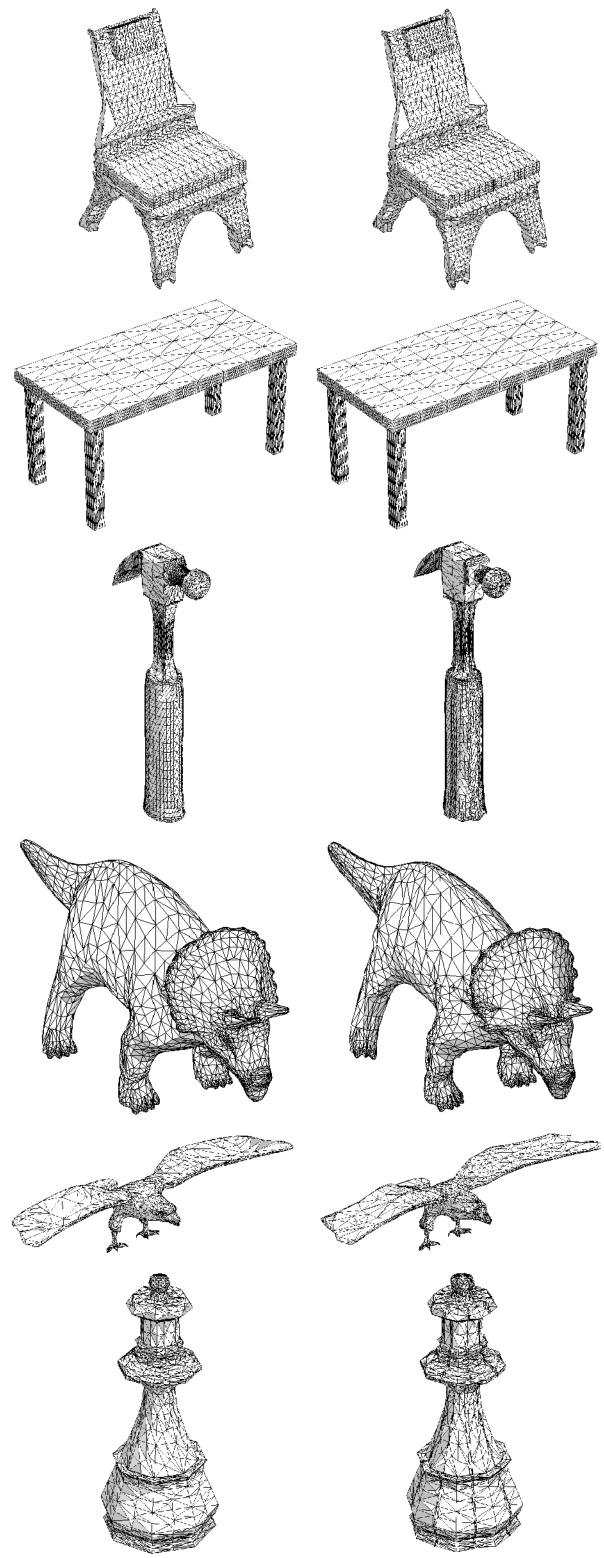

Figure 5: **Left:** original mesh data. **Right**: resulting model from unfolding of folding-tree representation. Please refer to table 1 for details.

applications as well as the exploitation of other types of symmetries (e.g. spherical and axial symmetries) in 3D meshes that can open up new implementations and extensions of our proposed methodology.

References

ALT, H., MEHLHORN, K., WAGENER, H., AND WELZL, E. 1988. Congruence, similarity, and symmetries of geometric objects. *Discrete Compututational Geometry 3*, 3, 237–256.

ARNHEIM, R. 1954. *Art and Visual Perception: A Psychology of the Creative Eye*. University of California Press, Berkeley.

ATALLAH, M. 1985. On symmetry detection. *IEEE Trans. Computers 34*, 7, 663–666.

BLACK, M. J., AND ANANDAN, P. 1996. The robust estimation of multiple motions: parametric and piecewise-smooth flow fields. *Computer Vision and Image Understanding 63*, 1, 75–104.

FORSYTH, D. A., AND PONCE, J. 2002. *Computer Vision: A Modern Approach*, first ed. Prentice Hall.

FUNKHOUSER, T., KAZHDAN, M., SHILANE, P., MIN, P., KIEFER, W., TAL, A., RUSINKIEWICZ, S., AND DOBKIN, D. 2004. Modeling by example. *ACM Trans. Graph. 23*, 3, 652–663.

GOFMAN, Y., AND KIRYATI, N. 1996. Detecting symmetry in grey level images: The global optimization approach. In *Proceedings of the 13th International Conference on Pattern Recognition*, 951–956.

GOMBRICH, E. H. 1969. *Art and Illusion: A Study in the Psychology of Pictorial Representation*. Princeton University Press, Princeton.

GUEZIEC, A., TAUBIN, G., LAZARUS, F., AND HORN, W. 1998. Converting sets of polygons to manifold surfaces by cutting and stitching. In *VIS '98: Proceedings of the conference on Visualization '98*, IEEE Computer Society Press, Los Alamitos, CA, USA, 383–390.

GUEZIEC, A., BOSSEN, F., TAUBIN, G., AND SILVA, C. 1999. Efficient compression of non-manifold polygonal meshes. In *Proceedings of the 10th IEEE Visualization 1999 Conference (VIS '99)*, IEEE Computer Society, Washington, USA.

HAMPEL, F. R., RONCHETTI, E. M., ROUSSEEUW, P. J., AND STAHEL, W. A. 1986. *Robust Statistics: The Approach Based on Influence Functions*. Wiley-Interscience, New York, USA.

HIGHNAM, P. T. 1986. Optimal algorithms for finding the symmetries of a planar point set. *Inf. Process. Lett. 22*, 5, 219–222.

HOPPE, H. 1996. Progressive meshes. In *SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, ACM Press, New York, USA, 99–108.

ISENBURG, M., AND ALLIEZ, P. 2002. Compressing polygon mesh geometry with parallelogram prediction. In *Proceedings of the conference on Visualization '02*, IEEE Computer Society, Washington, USA, 141–146.

JIANG, X., YU, K., AND BUNKE, H. 1996. Detection of rotational and involutional symmetries and congruity of polyhedra. *Visual Computing 12*, 4, 193–201.

JU, T. 2004. Robust repair of polygonal models. *ACM Trans. Graph. 23*, 3, 888–895.

KARNI, Z., AND GOTSMAN, C. 2000. Spectral compression of mesh geometry. In *SIGGRAPH '00: Proceedings of the 27th annual conference on computer graphics and interactive techniques*, ACM Press/Addison-Wesley Publishing Co., New York, USA, 279–286.

KAZHDAN, M., CHAZELLE, B., DOBKIN, D., FUNKHOUSER, T., AND RUSINKIEWICZ, S. 2004. A reflective symmetry descriptor for 3d models. *Algorithmica 38*, 1, 201–225.

KAZHDAN, M., FUNKHOUSER, T., AND RUSINKIEWICZ, S. 2004. Symmetry descriptors and 3d shape matching. In *SGP '04: Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, ACM Press, New York, NY, USA, 115–123.

MARA, D. O., AND OWENS, R. 1996. Measuring bilateral symmetry in digital images. In *In Proceedings of IEEE TENCON - Digital Signal Processing Applications*, vol. 1, 151–156.

MAROLA, G. 1989. On the detection of the axes of symmetry of symmetric and almost symmetric planar images. *IEEE Trans. Pattern Anal. Mach. Intell. 11*, 1, 104–108.

MINOVIC, P., ISHIKAWA, S., AND KATO, K. 1992. Three dimensional symmetry identification, part i: Theory. Tech. Rep. 21, Kyushu Institute of Technology, Japan.

NORCIA, A., CANDY, R., PETTET, M., VILDAVSKI, V., AND TYLER, C. 2002. Temporal dynamics of the human response to symmetry. *Journal of Vision 2*, 2 (3), 132–139.

RUSINKIEWICZ, S., AND LEVOY, M. 2001. Efficient variants of the icp algorithm. *In proceedings of the Third International Conference on 3-D Digital Imaging and Modeling*, 145.

SAWHNEY, H. S., AYER, S., AND GORKANI, M. 1995. Model-based 2d&3d dominant motion estimation for mosaicing and video representation. In *Proceedings of the Fifth International Conference on Computer Vision*, IEEE Computer Society, Washington, USA, 583.

SHEN, D., IP, H. S., CHEUNG, K. T., AND TEOH, E. K. 1999. Symmetry detection by generalized complex moments: A close-form solution. *IEEE Trans. Pattern Anal. Mach. Intell. 21*, 5, 466–476.

STEWART, C. V. 1999. Robust parameter estimation in computer vision. *SIAM Rev. 41*, 3, 513–537.

SUN, C., AND SHERRAH, J. 1997. 3d symmetry detection using the extended gaussian image. *IEEE Trans. Pattern Anal. Mach. Intell. 19*, 2, 164–168.

THRUN, S., AND WEGBREIT, B. 2005. Shape from symmetry. In *Tenth International Conference on Computer Vision*, IEEE Computer Society, vol. 2, 1824–1831.

TURK., G., AND LEVOY, M. 1994. Zippered polygon meshes from range images. In *SIGGRAPH '94: Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, ACM Press, New York, NY, USA, 311–318.

WOLTER, J. D., WOO, T. C., AND VOLZ, R. A. 1985. Optimal algorithms for symmetry detection in two and three dimensions. *Visual Compututer 1*, 37–48.

ZABRODSKY, H., PELEG, S., AND AVNIR, D. 1995. Symmetry as a continuous feature. *IEEE Trans. Pattern Anal. Mach. Intell. 17*, 1154–1166.