Harmonic Shape Images: A 3D Free-form Surface Representation and Its Applications in Surface Matching

Dongmei Zhang

CMU-RI-TR-99-41

Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

*Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
in the field of Robotics*

Thesis Committee:
Martial Hebert, Chair
Takeo Kanade
Michael Erdmann
Katsushi Ikeuchi

November 16, 1999

Abstract

Surface matching is the process of determining whether two surfaces are equivalent in terms of shape. The usual steps in surface matching include finding the correspondences on the two surfaces being compared and computing the rigid transformation between the two surfaces. The issue of surface matching not only has theoretical importance but also has wide applications in the real world.

Unlike the matching of 2D images, the matching of 3D free-form surfaces is a much more difficult problem due to the shape and topology complexity of 3D surfaces. In general, there is no simple representation such as a matrix for 2D images that can be used to compare 3D surfaces. Issues such as surface sampling resolution, occlusion and high dimension of the pose space further complicate the problem.

In this thesis, a novel representation called Harmonic Shape Images is proposed for representing the shape of 3D free-form surfaces that are originally represented by triangular meshes. This representation is created based on the mathematical theory on harmonic maps. Given a 3D surface patch with disc topology and a selected 2D planar domain, a harmonic map is constructed by a two-step process which includes boundary mapping and interior mapping. Under this mapping, there is one to one correspondence between the points on the 3D surface patch and the resultant harmonic image. Using this correspondence relationship, *Harmonic Shape Images* are created by associating shape descriptors computed at each point of the surface patch at the corresponding point in the harmonic image. As a result, Harmonic Shape Images are 2D shape representations of the 3D surface patch.

Harmonic Shape Images have some properties that are important for surface matching. They are unique; their existence is guaranteed for any valid surface patches. More importantly, those images preserve both the shape and the continuity of the underlying surfaces. Furthermore, Harmonic Shape Images are not designed specifically for representing surface shape. Instead, they provide a general framework to represent surface attributes such as surface normal, color, texture and material. Harmonic Shape Images are discriminative and stable, and they are robust with respect to surface sampling resolution and occlusion. Extensive experiments have been conducted to analyze and demonstrate the properties of Harmonic Shape Images.

The usefulness of Harmonic Shape Images in surface matching is demonstrated using application examples in the real world such as face modeling and mesh watermarking. In addition, the experimental results on recognizing objects in scenes with occlusion demonstrate its successful application in object recognition. Experiments to measure the accuracy of surface registration using Harmonic Shape Images are also described.

Acknowledgements

First and foremost, I am most indebted to my thesis advisor, Prof. Martial Hebert. Martial gave me invaluable guidance along my road to a Ph.D. in robotics. His broad and in-depth knowledge of computer vision and his enthusiasm for exploring unknown research fields have a great influence on me. I will always admire his knowledge and dedication to work.

I am grateful to Dr. John Webb, who was my first advisor when I enrolled in the Robotics Institute of Carnegie Mellon University. I appreciated the advice and help he gave me during my early years of study toward my Ph.D. degree.

I would like to acknowledge the members on my Ph.D. thesis committee: Prof. Takeo Kanade, Prof. Katsushi Ikeuchi and Prof. Michael Erdmann. I appreciate their comments and suggestions on my research over the years.

I would like to acknowledge Dr. Hughs Hoppe at Microsoft Research and Dr. Shizhuo Yin at Pennsylvania State University for providing range images for my research.

I would like to thank my supervisors when I was a summer intern at the Austin Product Center, Schlumberger LTD in 1998: Dr. Peter Highnam and Dr. Richard Hammersley. Peter gave me the chance to conduct interesting research in computer graphics. Richard worked with me closely and helped me understand many issues in geoscience.

I would also like to thank the people who provided help in one way or the other for my research. Dr. Andrew Johnson at Jet Propulsion Laboratory developed some software for range image processing during his Ph.D. study in the Robotics Institute. The software is still being used in our research group. Dr. Sing Bing Kang and Dr. Harry Shum at Microsoft Research, Dr. Mark Wheeler at Cyra Technologies, and Dr. Yoichi Sato at Tokyo University provided kind help with my research during my early years at Carnegie Mellon University. Owen Carmichael and Daniel Huber provided help on facilitating the experiments conducted for my research. My special thanks go to Marie Elm who reviewed all my publications and my Ph.D. thesis.

Last but not least, with all my heart, I thank my beloved husband, Qinghui Zhou, for his love, understanding and support. I am very proud that we encouraged each other to achieve our career goals.

Table of Contents

List of Figures

Chapter 1

Chapter 2

Chapter 3

Chapter 4

Chapter 5

Chapter 6

List of Tables

Chapter 1

Chapter 2

Chapter 3

Chapter 6

Appendix A

Chapter 1

Introduction

Surface matching is the process of determining whether two surfaces are equivalent in terms of shape. The research topic of this thesis focuses on how to effectively represent free-form surfaces in three-dimensional space and how to use that representation to perform surface matching. The main contribution of the work described in this thesis is the development of a novel geometric representation for 3D free-form surfaces named Harmonic Shape Images and the application of this representation in surface matching.

In the early years of computer vision, the shape information of 3D objects was obtained using camera images which are 2D projections of the 3D world. Because of the lack of depth information about the objects in the scene, the proposed approaches suffer from difficulties especially when there are significant lighting variations, or when the objects in the scene have complex shapes, or when the scene is highly cluttered.

In recent years, due to the advances in 3D sensing technology and shape recovery algorithms, digitized 3D surface data have become widely available. Devices such as contact probes, laser range finders, stereo vision systems, Computed Tomography systems and Magnetic Resonance Imaging systems can all provide digitized 3D surface data in different application domains. Wide range of applications is another motivation for the research on surface matching using 3D surface data.

Applications of surface matching can be classified into two categories. The first category is surface registration with the goal of aligning surface data sets in different coordinate systems into the same coordinate system. Application examples in this category include: industrial inspection, which determines whether a manufactured part is consistent with a pre-stored CAD model; surface modeling, which aligns and integrates surface data sets from multiple views of a 3D object into a complete 3D mesh model; and mesh watermarking, which protects the copyright of published mesh models of 3D objects. The second category is object recognition with the goal of locating and/or recognizing an object in a cluttered scene. Robot navigation is one of the application examples in this category.

In this chapter, the problem statement of this thesis is first defined, followed by a discussion on the difficulties of developing appropriate surface representation schemes for surface matching. Then the previous research work in the field of surface representation and object recognition is reviewed. Harmonic Shape Images, a novel representation for surface matching proposed in this thesis, is introduced next. The overview of the thesis is presented at the end of this chapter.

1.1 Problem Definition

Given two free-form surfaces represented by polygonal meshes in the 3D space, the objective of surface matching is two-fold. Firstly, it is necessary to determine whether those two surfaces are similar to each other in shape. Secondly, when there is a match, it is necessary to find the correspondences between the two surfaces.

Several remarks should be made with respect to the above definition. First of all, a free-form surface is defined to be a smooth surface such that the surface normal is well defined and continuous almost everywhere, except at vertices, edges and cusps[25]. Second, according to [31], a polygonal mesh is “a collection of vertices, edges and polygons connected such that each edge is shared by, at most, two polygons. An edge connects two vertices, and a polygon is a closed sequence of edges. An edge can be shared by, at most, two adjacent polygons and a vertex is shared by at least two edges”. Without losing generality, triangular meshes are assumed to be the input format of all free-form surfaces because triangular meshes can be easily obtained from any polygonal meshes. Third, there is no prior knowledge about the positions of the two surfaces in the 3D space. In general, they are in different coordinate systems. The transformation between those two surfaces is assumed to be rigid transformation, i.e., three parameters for rotation plus three parameters for translation. The two surfaces can be aligned into the same coordinate system if they match each other in shape.

1.2 Difficulties of Surface Matching

Consider matching the two surfaces shown in Figure 1.1. Although it is obvious that the two surfaces are from the same object but with different viewing directions, it is difficult to come up with a generic algorithm to solve this problem. Difficulties of matching 3D free-form surfaces include the following.

- Topology

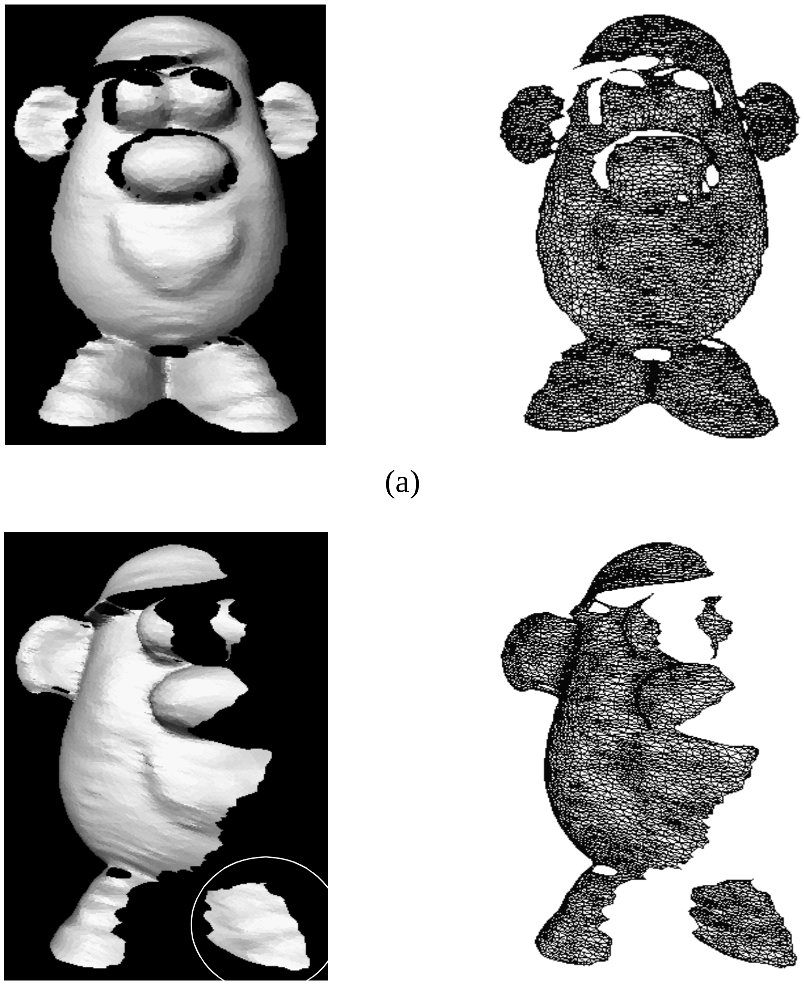
The two surfaces to be matched may have different topologies. For example, the two surfaces in Figure 1.1 are topologically different because they are two different surfaces, although they partially overlap. Occlusion can also result in topological difference. For example, in Figure 1.1, the circled part on S_2 appears to be a disconnected region due to occlusion. In contrast, its correspondence is connected to the entire surface on S_1. The topology issue is difficult to address when trying to conduct global matching between two surfaces.

- Resolution

Generally speaking, the resolutions of different digitized surfaces are different. The resolution problem makes it difficult to establish the correspondences between two surfaces, which in turn, results in the difficulty of comparing the two surfaces. Even if the resolution of two sampled surfaces is the same, in general, the sampling vertices on one surface are not exactly the same as that on the other one.

- Connectivity

For arbitrary triangular meshes, the connectivities among vertices are arbitrary. Even if two surfaces have the same number of vertices, they may still have different connectivities among vertices. This is in contrast to images. An image has a regular *m* by *n* matrix structure. The connectivities are the same for all the pixels (pixels on the boundary have the same connectivity pattern as well). When conducting template matching, the correspondences between two images can be naturally established.



(a)

(b)

Figure 1.1: Examples of 3D free-form surfaces to be matched. (a) Surface S_1; (b) surface S_2.

- Pose

It has been mentioned in the problem definition that there is no prior knowledge about the positions of the two surfaces in 3D space. Therefore, unlike conducting template matching of images, there is no natural coordinate system for aligning two surfaces. Although an exhaustive search strategy could be used to find the transformation in the six-dimensional pose space, it is computationally prohibitive without a good initial estimate of the transformation.

- Occlusion

Either self-occlusion or occlusion due to other objects is a common phenomenon in real scenes. When comparing two 2D images, if occlusion is present in one image, then some robust techniques may be used to discount the corresponding part in another image so that only the non-occluded parts of the two images are taken into account in template matching. Here, it is important to notice that the occlusion does not change any of the remaining parts of the image. Therefore, the comparison result of the two images will not be affected by occlusion as long as the occluded part can be correctly detected and discounted.

In contrast to comparing 2D images, matching 3D free-form surfaces is far more complicated when occlusion is present in the scene. Model-based matching is a common framework for solving the 3D surface matching problem. It requires that an intermediate representation be created for each given surface so that the problem of matching surfaces can be reduced to matching those representations. When there is occlusion in a given surface, it is important for the representation to remain the same as that created for the same surface without occlusion. Otherwise, the comparison of the two representations with and without occlusion will yield different results. Therefore, it is crucial for representations to be invariant to occlusion. However, this requirement is too demanding to be fulfilled in practice. As an alternative, the representation is desired to change gradually as occlusion increases so that, the matching result degrades gracefully. Although a considerable amount of work has been done in developing representations for 3D free-form surfaces, the problem of developing occlusion-robust representation is still open.

1.3 Previous work

A considerable amount of research has been conducted on comparing 3D free-form surfaces. The approaches used to solve the problem can be classified into two categories according to methodology. Approaches in the first category try to create some form of representation for input surfaces and transform the problem of comparing the input surfaces to the simplified problem of comparing their representations. These approaches are used most often in model-based object recognition. In contrast, approaches in the second category work on the input surface data directly without creating any kind of representation. One data set is aligned to the other by looking for the best rigid transformation, using optimization techniques to search the six-dimensional pose space. These approaches are mainly used for surface registration.

According to the manner of representing the shape of an object, existing representations of 3D free-form objects may be regarded as either global or local. Examples of global representations are algebraic polynomials[49][80], spherical representations such as EGI (extended Gaussian Image)[40], SAI (Spherical Attribute Image)[34][36][19][20] and COSMOS (Curvedness-Orientation-Shape Map On Sphere)[25], triangles and crease angle histograms[9], and HOT (High Order Tangent) curves[47]. Although global representations can describe the overall shape of an object, they have difficulties in representing objects of arbitrary topology or arbitrary complexity in shape. For example, the SAI can represent only objects with spherical topology. Moreover, global representations have difficulty in handling clutter and occlusion in the scene.

Many local representations are primitive-based. In [29], model surfaces are approximated by linear primitives such as points, lines and planes. The recognition of objects is carried out by attempting to locate the objects through a hypothesize-and-test process. In [78], super-segments and splashes are proposed to represent 3D curves and surface patches with significant structural changes. A splash is a local Gaussian map describing the distribution of surface normals along a geodesic circle. Since a splash can be represented as a 3D curve, it is approximated by multiple line fitting with differing tolerances. Therefore, the splashes of the models can be encoded and stored in the database. The on-line matching of a scene object to potential model objects consists of indexing into the database using the encoded descriptions of the scene splash to find the best matched model splash. In [16], a three-point-based representation is proposed to register 3D surfaces and recognize objects in cluttered scenes. On the scene object, three points are selected with the requirement that (1) their curvature values can be reliably computed; (2) they are not umbilical points; and (3) the points are spatially separated as much as possible. Then, under the curvature, distance, and direction constraints, different sets of three points on the model surface are found to correspond to three points on the scene objects. The transformations computed using those scene-model correspondences are verified to select the best one. More recently, a local representation called Spin-Images is proposed in [44]. Instead of looking for primitives or feature points at some parts of the object surface with significant structural changes, a Spin-Image is created for every point of the object surface as a 2D description of the local shape at that point. Given an oriented point on the surface and its neighborhood of a certain size, the normal vector and the tangent plane are computed at that point. Then the shape of the neighborhood is described by the relative positions of the vertices in the neighborhood to the central vertex using the distances to the normal and tangent plane. A Spin-Image is a 2D histogram of those distances. Good recognition results in complex scenes using Spin-Images are reported in [44]. However, Spin-Images are not well-understood at a mathematical level and they discard one dimension information of the underlying surfaces, namely, Spin-Images do not preserve the continuity of surfaces. As can be seen from above, although local representations can not provide an overall description of the object shape, they have advantages in handling clutter and occlusion in the scene.

Among 3D surface registration algorithms, Iterative Closest Point (ICP) plays an important role. In [10], the scene surface is registered with the model surface by iteratively finding the closest points on the model surface to the points on the scene surface and refining the transformation in the six-dimensional pose space. Although this approach guarantees finding the local minimum of the registration error, it requires good initial estimate of the transformation in order to find the global minimum. Another limitation of this approach is that it can not handle two surfaces which only partially overlap. An heuristic method was proposed in [100] to overcome the partially overlapping difficulty. A K-D tree structure was also used in [100] to speed up the process of finding the closest point. Unlike the ICP approach, an algorithm is proposed in [14] to increase the accuracy of registration by minimizing the distance from the scene surface to the nearest tangent plane approximating the model surface. In order to reduce computational complexity, control points are selected for registration instead of using the entire data set of the model surface. However, this may not work well on surfaces with no control points selected on some of their parts that have significant structural changes. Moreover, this approach also requires a good initial estimate of the transformation. In [6], surfaces are approximated by constructing a hierarchy of Delaunay triangulations at different resolutions. Starting at a lower resolution, the correspondences between triangles can be established through the attributes of the triangles such as centroids and normals. Based on these correspondences, a rigid transformation can be computed and refined by tracking the best matching pairs of triangles in the triangulation hierarchy. Although this algorithm was proposed for integrating object surfaces from different view points, it can be modified and applied to object recognition as well. In summary, in order for the surface registration algorithms to work well, a good initial estimate of the transformation is usually required.

1.4 The Concept of Harmonic Shape Images

The key concept of Harmonic Shape Images is to compare surfaces by comparing patches on them. A surface patch is defined to be a connected region without holes on a given surface. Harmonic Shape Images are 2D shape representations of surface patches. Using this representation, the problem of 3D surface matching is reduced to 2D image matching. The paradigm in Figure 1.2 illustrates this idea.

Given two surfaces S_1 and S_2, most of the previous approaches conduct point-based matching as shown in Figure 1.2(a). As a result, only point-to-point correspondences can be established when the representations of two points match. In contrast, the approach proposed in this thesis matches patches on two surfaces as shown in Figure 1.2(b). Correspondences between the two patches, i.e., correspondences between every pair of points in the two patches, can be established immediately without any extra cost once the representations of the two patches match each other.

The main advantage in conducting patch-based surface matching is that the complete geometric information, both shape and continuity, of the underlying surfaces is used in the matching process whereas in point-based matching, only the shape information is used and the continuity information is discarded. Making use of surface continuity allows the natural establishment of correspondences between two surfaces after the matching process. This means that the difficult problem of finding correspondences on two surface patches becomes trivial. The matching process does not only provide a matching score, but also the correspondences, i.e., the correspondences are obtained without any extra computation. In contrast, if surface continuity is discarded, then the correspondence between only one pair of points (usually the centers of patches) can be established. In order to find more pairs of correspondences, more matching needs to be conducted and heuristic-guided post-processing has to be done to organize the matched points into mutually consistent correspondences.

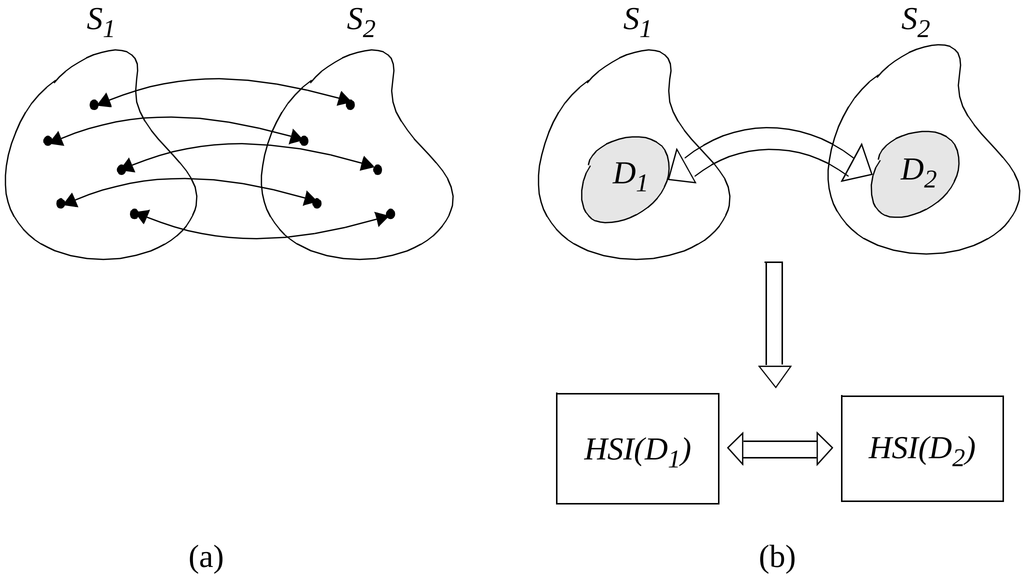


Figure 1.2: Illustration of point-based surface matching and patch-based surface matching.

This thesis describes patch-based surface matching conducted by matching Harmonic Shape Images of those patches as shown in Figure 1.2(b). Therefore, creating those images is the core of the patch-based matching. The main challenges for creating such a representation are as follows. The first one is that the representation needs to preserve both shape and continuity of the underlying surface patch. Previous approaches have shown that it is relatively easy to preserve shape only but not both. The second challenge is how to make the representation robust with respect to occlusion.

The following section introduces the generation of Harmonic Shape Images, discusses how to address the above two challenges, then summarizes the properties of Harmonic Shape Images.

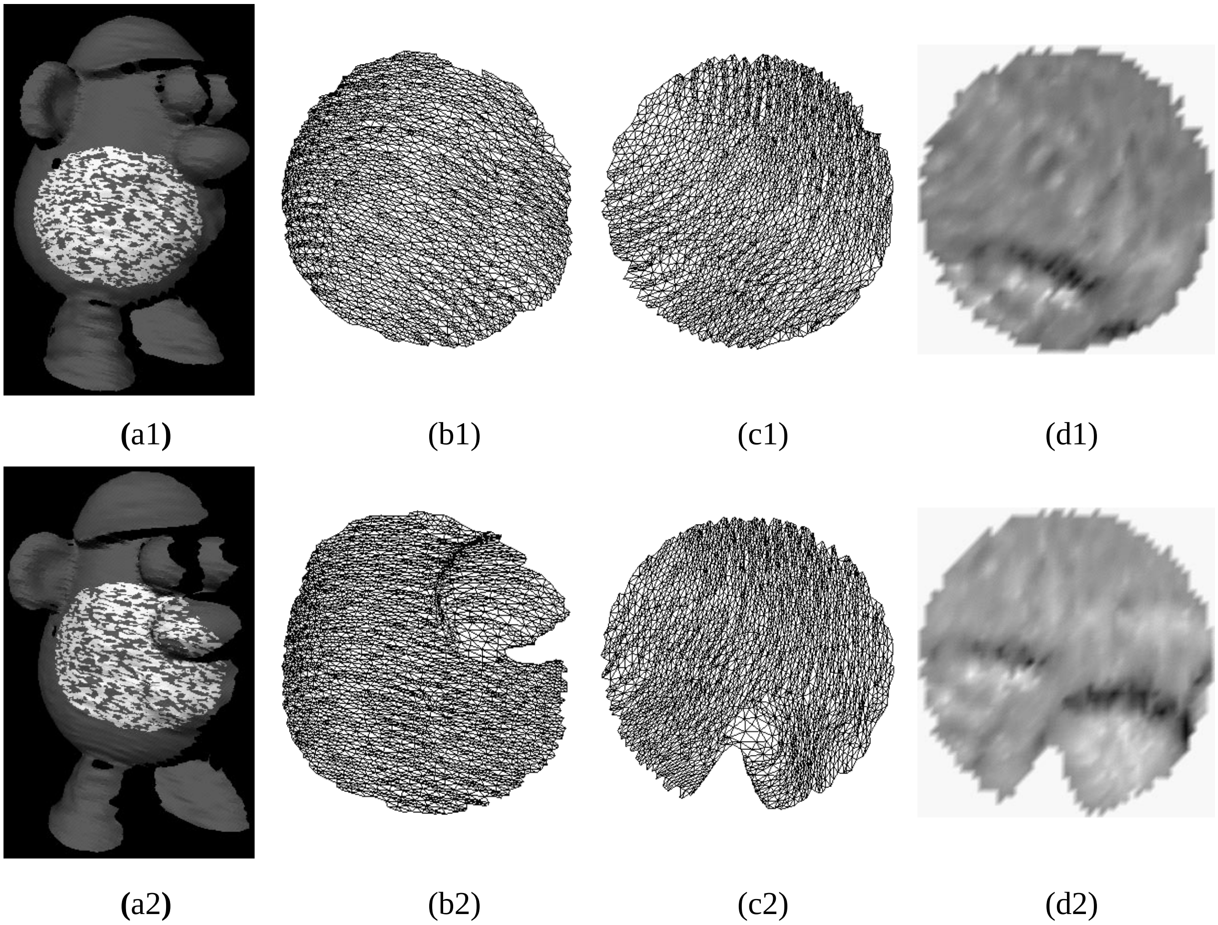
Generation of Harmonic Shape Images

Given a 3D surface *S* as shown in Figure 1.3(a1), let *v* denote an arbitrary vertex on *S*. Let *D(v, R)* denote the surface patch which has the central vertex *v* and radius *R*. *R* is measured by distance along the surface. *D(v, R)* is assumed to be a connected region without holes. *D(v, R)* consists of all the vertices in *S* whose surface distances are less than, or equal to, *R*. The overlaid region in Figure 1.3(a1) is an example of *D(v, R)*. Its amplified version is shown in Figure 1.3(b1). The unit disc *P* on a 2D plane is selected to be the target domain. *D(v, R)* is mapped onto *P* by minimizing an energy functional. The resultant image *HI(D(v, R))* is called the harmonic image of *D(v, R)* as shown in Figure 1.3(c1).

As can be seen in Figure 1.3(a1) and (c1), for every vertex on the original surface patch *D(v, R)*, one, and only one, vertex corresponds to it in the harmonic image *HI(D(v, R))*. Furthermore, the connectivities among the vertices in *HI(D(v, R))* are the same as that of *D(v, R)*. This means that the continuity of *D(v, R)* is preserved on the harmonic image *HI(D(v, R))*.

The preservation of the shape of *D(v, R)* is shown more clearly on the Harmonic Shape Image *HSI(D(v, R))*(Figure 1.3(d1)) which is generated by associating shape descriptor at every vertex on the harmonic image(c1). The shape descriptor is computed at every vertex on the original surface patch(b1). On *HSI(D(v, R))*, high intensity values represent high curvature values and low intensity values represent low curvature values. The reason for Harmonic Shape Images' ability to preserve the shape of the underlying surface patches lies in the energy functional which is used to construct the mapping between a surface patch *D(v, R)* and the 2D target domain *P.* This energy functional is defined to be the shape distortion when mapping *D(v, R)* onto *P.* Therefore, by minimizing the functional, the shape of *D(v, R)* is maximally preserved on *P.*

Another surface patch is shown in Figure 1.3(a2) and (b2). Its harmonic image and Harmonic Shape Image are shown in (c2) and (d2), respectively. In this case, there is occlusion in the surface patch(Figure 1.3(b2)). The occlusion is captured by its harmonic image and Harmonic Shape Image(Figure 1.3(c2), (d2)). The latter's ability to handle occlusion comes from the way the boundary mapping is constructed when mapping the boundary of *D(v, R)* onto the boundary of *P*; because of the boundary mapping, the images remain approximately the same in the presence of occlusion.



(a1) (b1) (c1) (d1)

(a2) (b2) (c2) (d2)

Figure 1.3: Examples of surface patches and Harmonic Shape Images. (a1), (a2) Surface patches on a given surface; (b1), (b2) the surface patches in wireframe; (c1), (c2) their harmonic images; (d1), (d2) their Harmonic Shape Images.

From the above generation process, it can be seen that the only requirement imposed on creating Harmonic Shape Images is that the underlying surface patch is connected and without holes. This requirement is called the topology constraint. The relaxation of this constraint will be discussed in Chapter 5.

Properties of Harmonic Shape Images

Harmonic Shape Images are a patch-based representation for 3D free-form surfaces. For a given surface, a surface patch with radius *R* can be generated for every vertex on that surface. This means that a Harmonic Shape Image can then be generated for each surface patch as long as that surface patch satisfies the topology constraint. The size of the a surface patch is defined by the radius *R*. When *R* increases, the area of the surface patch increases and its Harmonic Shape Image consists of more information of the underlying surface. Another point that needs to be noticed about the patch representation is that every vertex on a given surface is treated equally. No feature points or special points need to be selected in order to create Harmonic Shape Images. This is in contrast to previous feature-based (or primitive-based) representations in which the robustness of feature extraction is a difficult issue to cope with.

Harmonic Shape Images are defined on a simple domain which is a unit disc. This simplifies the 3D surface matching problem to a 2D image matching problem. Furthermore, because the unit disc is specified as the domain for any given surface patches, the Harmonic Shape Images of all those patches are defined in the same coordinate system regardless of the actual positions of those patches. This means that Harmonic Shape Images are pose invariant.

Harmonic Shape Images capture both the shape and the continuity information of the underlying surface patches. It can be seen easily from Figure 1.3 that there is one-to-one correspondence between the vertices on the surface patch and its harmonic image. In fact, the mapping from the surface patch to the disc domain is a well-behaved mapping. It is one-to-one and onto; it is continuous; it is unique; and it is intrinsic to the original surface patch. This property allows the natural establishment of correspondences between two surface patches once their Harmonic Shape Images match.

Finally, Harmonic Shape Images are robust to occlusion and sampling resolution. This property has been briefly discussed earlier in this section and will be discussed in detail in later chapters.

The comparison of Harmonic Shape Images and some surface representations previously proposed is listed in Table 1.1.

1.5 Thesis Overview

This thesis presents a novel representation, Harmonic Shape Images, for 3D free-form surfaces. In addition to describing in detail the generation of Harmonic Shape Images, the properties of this representation are thoroughly investigated by conducting various experiments. A surface matching strategy using Harmonic Shape Images is proposed and applied to surface matching experiments. The usefulness of these images is demonstrated by different applications of surface matching. There are eight chapters in this thesis. The content of each chapter is summarized as follows.

Chapter 2 describes the generation of Harmonic Shape Images in detail. The concepts of 3D surface patch and 2D target domain are first defined followed by the two-step construction of the harmonic map. The interior mapping and boundary mapping are the crucial steps in constructing the harmonic map between a 3D surface patch and the 2D target domain. The generation of Harmonic Shape Images is then discussed as a surface attribute association process on the harmonic maps.

Chapter 3 discusses the matching of Harmonic Shape Images. A shape similarity measure of these images is defined and a statistical approach is presented for quantitatively determining how a Harmonic Shape Image differs from others. At the end of Chapter 3, a 3D free-form surface-matching strategy using Harmonic Shape Images is presented.

In Chapter 4, an experimental approach is employed to analyze and demonstrate some of the important properties of Harmonic Shape Images. Those properties include discriminability, stability, robustness to surface sampling resolution and robustness to occlusion. Two libraries of surface patches are used in those experiments.

Chapter 5 discusses some enhancements for Harmonic Shape Images. The hole interpolation strategy can help relax the topology constraint which has been discussed earlier in this Chapter. The cutting and smoothing techniques can further improve the robustness of Harmonic Shape Images in the presence of occlusion.

In Chapter 6, experiments on surface matching using Harmonic Shape Images are described and the results are presented and analyzed. Application examples of surface matching include face modeling and mesh watermarking. Harmonic Shape Images have also been used to recognize objects in scenes with occlusion. Experiments have been conducted to measure the registration error using Harmonic Shape Images.

Chapter 7 discusses the future research directions for the use of Harmonic Shape Images. Some extensions can be added to the images to improve the matching process. Applications of the images in Shape analysis are discussed.

Chapter 8 concludes this thesis by summarizing the contributions of the thesis.

Table 1.1 Comparison of some surface representations

Comparison Items Representations	Object domain	Type of representation	Mapping	Completeness of representation
SAI	Objects with spherical topology	Global	Spherical mapping of surface curvature at all points	Complete, both surface shape and continuity are represented
Splashes	Objects with-Free-form surfaces	Local	Gaussian map of surface normals along a geodesic circle	Partial
COSMOS	Objects with-Free-form surfaces	Global	Spherical mapping of orientation of CSMPs	Partial for non-convex objects
Spin-Images	Objects with-Free-form surfaces	Local	2D histogram of distances to the reference tangent plane and surface normal at all points	Partial, surface continuity is not represented
HSI	Objects with-Free-form surfaces	Local	Harmonic map of the underlying surface onto a unit disc. surface curvature is stored on the map for all points.	Complete, both surface shape and the continuity are represented

Chapter 2

Generating Harmonic Shape Images

The name of Harmonic Shape Images comes from the fact that the mathematical tool, which is called harmonic maps, is used in its generation process. The idea of using harmonic maps to develop a surface representation is partly inspired by the work in the computer graphics field done by Eck et al.[26] at the University of Washington.

Given a triangular mesh S with arbitrary topology, the goal of the work in [26] is to create a *remesh S'* which has the same shape as S but the connectivities among its vertices satisfy a subdivision requirement[26]. As part of the remeshing process, S is partitioned into connected surface patches without holes and each such patch is mapped onto an equilateral triangle using harmonic maps. The maps on those equlateral triangles are then used to resample S with specified connectivities among the resampled points. It can be seen that the use of harmonic maps in [26] is to perform a certain form of resampling on 3D triangular meshes.

In this thesis, the use of harmonic maps is to conduct surface matching which is different from the goal in [26]. In order to compare two surfaces, an intermediate representation, Harmonic Shape Images, is created using harmonic maps. The concept of Harmonic Shape Images has been introduced in Chapter 1. In this chapter, the generation of Harmonic Shape Images will be discussed in detail. The background of Harmonic Maps is briefly reviewed first. The source and target manifolds in creating Harmonic Shape Images are then defined. The core steps of the generation process, the interior mapping and the boundary mapping, are explained next in detail followed by the discussion on how to select the parameters when generating Harmonic Shape Images. At the end of this chapter, different schemes for approximating the curvature at each vertex of the surface mesh are presented.

2.1 Harmonic Maps

The theory of Harmonic Maps studies the mapping between two manifolds from an energy point of view. Formally, let *(M, g)* and *(N, h)* be two smooth manifolds of dimensions m and n respectively, and let ϕ: $(M, g) \to (N, h)$ be a smooth map. Let (x^i), $i = 1, \ldots, m$ and (y^α) $\alpha = 1, \ldots, n$ be local coordinates around x and $\phi(x)$, respectively. Take (x^i) and (y^α) of M and N at corresponding points under the map ϕ whose tangent vectors of the coordinate curves are $\partial / \partial x^i$ and $\partial / \partial y^\alpha$, respectively. Then the energy density of ϕ is defined as[94]

$$e(\phi) = \frac{1}{2} \sum_{i,j=1}^{m} g^{ij} \langle \phi_* \frac{\partial}{\partial x^i}, \phi_* \frac{\partial}{\partial x^j} \rangle_N = \frac{1}{2} \sum_{i,j=1}^{m} g^{ij} \sum_{\alpha,\beta=1}^{n} \frac{\partial \phi^\alpha}{\partial x^i} \frac{\partial \phi^\beta}{\partial x^j} h_{\alpha\beta}(\phi) \qquad (2.1)$$

In (2.1), g_{ij} and $h_{\alpha\beta}$ are the components of the metric tensors in the local coordinates on M and N, respectively. The energy of ϕ in local coordinates is given by the number[94]

$$E(\phi) = \sum_M e(\phi) v_g \qquad (2.2)$$

If ϕ is of class C^2, $E(\phi) < \infty$, and ϕ is an extremum of the energy, then ϕ is called a harmonic map and satisfies the corresponding Euler-Lagrange equation. In the special case in which M is a surface D of disc topology and N is a convex region P in $\boldsymbol{E}^2$, the following problem has a unique solution[27]: Fix a homeomorphism b between the boundary of D and the boundary of P. Then there is a unique harmonic map ϕ: $D \rightarrow P$ that agrees with b on the boundary of D and minimizes the energy functional of D. In addition to the energy minimizing property, the harmonic map ϕ has the following properties: it is infinitely differentiable; it is an embedding of D into P; it is intrinsic to the underlying surface.

2.2 Definition of Surface Patch

The concept of surface patch has been explained briefly in Section 1.4. In this section, surface patch is defined formally as follows. Given a 3D free-form surface S represented by triangular mesh, an arbitrary vertex v on S and a radius R measured by surface distance, a surface patch $D(v, R)$ of S is defined to be the neighborhood of v, which includes all the vertices whose surface distances to v are less than or equal to R. $D(v, R)$ is said to be valid when the topology constraint, which requires $D(v, R)$ be connected and without holes, is satisfied. Three surface patches are shown in Figure 2.1(a), (b) and (c), respectively. Their wireframe versions are shown in Figure 2.1(d), (e) and (f). The patch in Figure 2.1(a) is valid while the patches in Figure 2.1(b) and (c) are invalid because they both have holes.

The implementation of generating a surface patch mainly involves the computation of the shortest path from a given vertex to any other vertex on the underlying triangular mesh. The triangular mesh can be considered as a bi-directional connected graph which has positive cost for all of its edges. Then the problem of computing the shortest path from a given vertex to any other vertex can be solved using the Single Source Dijkastra's algorithm[1].

Two implementation issues that need to be discussed about generating surface patches are how to deal with dangling triangles and how to use the radius margin.

Dangling Triangles

After a surface patch is generated, a post-processing step needs to be done in order to clean up the patch boundary. The purpose of the cleanup is to delete the dangling triangles on the patch boundary. For example, the surface patch shown in Figure 2.2(a) has two dangling triangles (shaded). The reason for performing this operation is as follows. When constructing the boundary mapping which will be explained in the following section, the boundary vertices need to be ordered in either a clock-wise or counter-clock-wise manner. In Figure 2.2(a), starting from vertex v_1, it is not uniquely determined whether the next boundary vertex should be v_2 or v_4 because there is a loop caused by the dangling triangle (v_2, v_3, v_4). Here, a triangle is considered to be dangling if not all of its three vertices have degrees greater than or equal to three. In order to uniquely determine the ordering of the boundary vertices and avoid boundary pathologies(Figure 2.2(b)), dangling triangles on surface patches such as the ones in Figure 2.2(a), (b) need to be pruned.

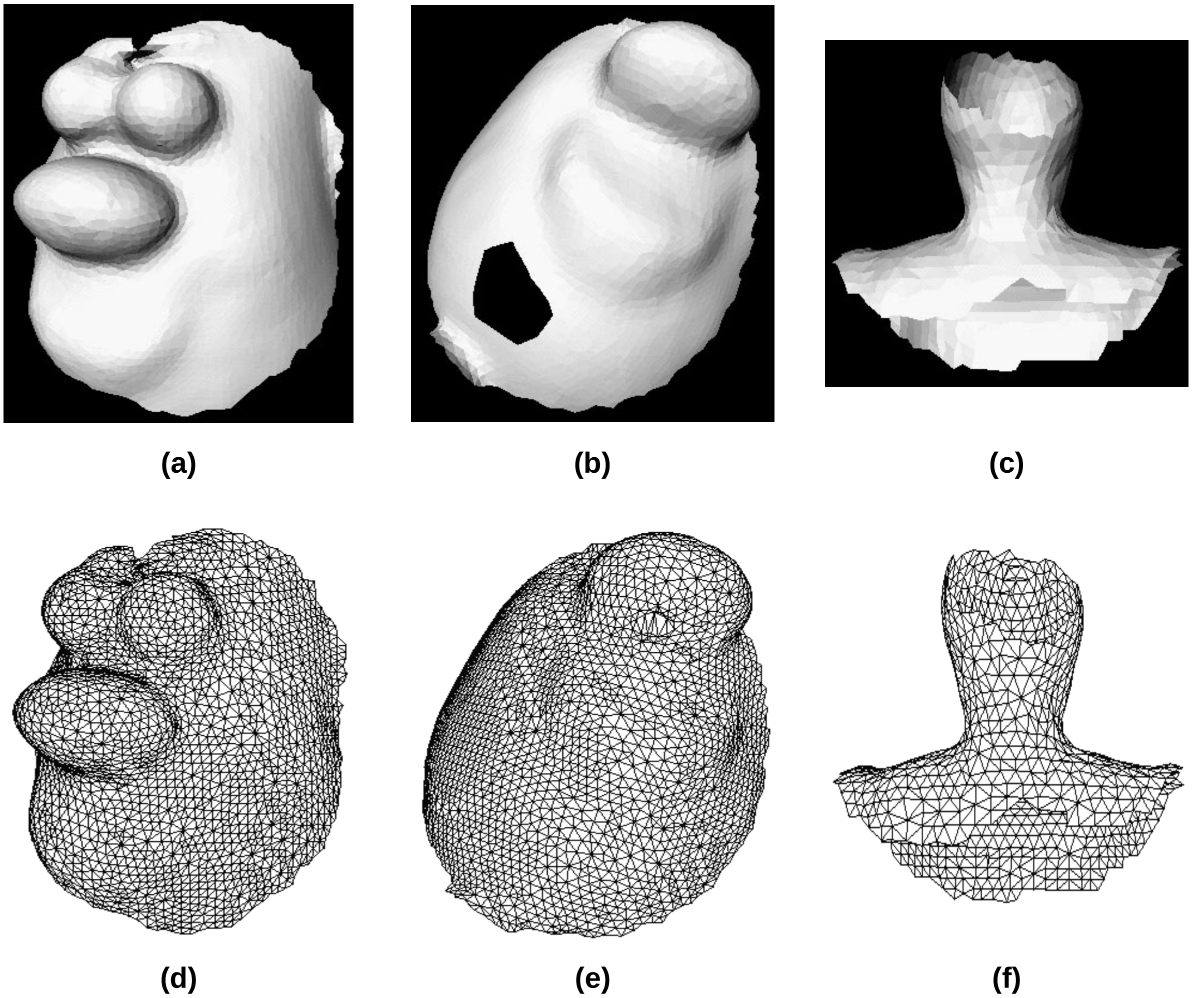


Figure 2.1: Example of surface patches. (a) A valid surface patch; (b), (c) Invalid surface patches; (d), (e), (f) Surface patches of (a), (b) and (c) in wireframe form, respectively.

Radius Margin

Another issue with generating surface patches is the radius margin. In general, the shortest distance from a vertex u to the central vertex v can not be exactly equal to the specified radius R. If the vertex u is not included in the surface patch centered at v because the shortest distance from u to v is slightly greater than the specified radius R, then some information may be lost due to the irregular sampling of the mesh as shown in Figure 2.3. In Figure 2.3(a) and (c), the two surfaces have the same shape but different sampling. If a surface patch is to be generated at vertex v using radius R on the two surfaces shown in (a) and (c), respectively, then the resultant surface patches are shown in (b) and (d), respectively. It can be seen that the vertex u is not included in the surface patch in (b) because the shortest distance from u to v is slightly greater than R. In contrast, u is included in the surface patch shown in (d) because the shortest distance from u to v is equal to R. The difference in the shortest distance from u to v is not caused by the shape of the surface. Rather, it is caused by the sampling noise. If a radius margin, which is equal to a small percentage of R, is added to the actual radius, then the vertex u will be included in the

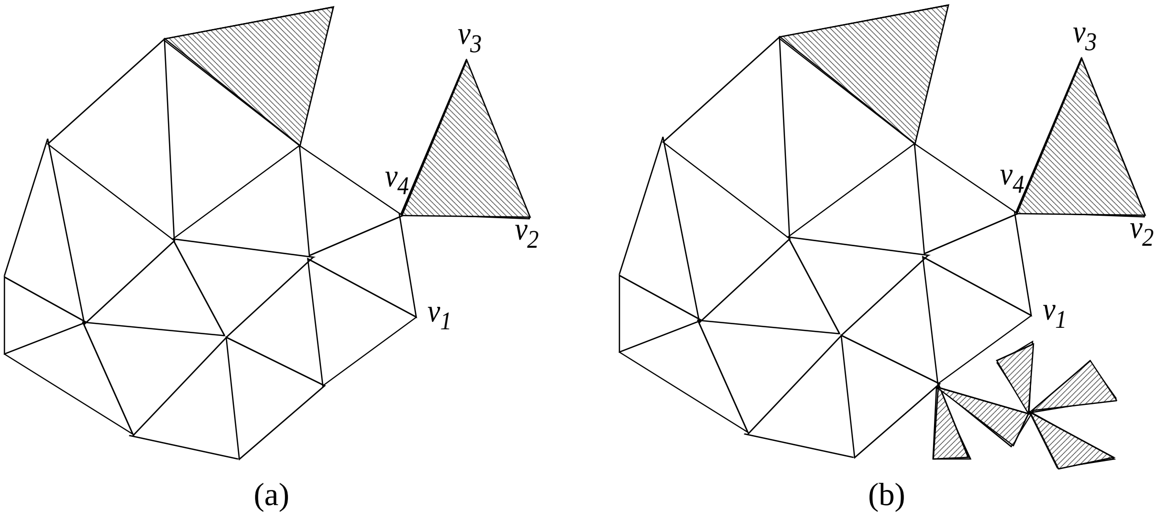


Figure 2.2: Surface patches with dangling triangles on the boundary.

surface patch in (b) and the resultant surface patch will be more similar to the one in (d). As a summary, using radius margin is a heuristic to overcome sampling noise when generating surface patches so that the surface patches to be compared would be as similar as possible if they were actually the same patch. In all the experiments that have been conducted, the radius margin is set to be 5%.

It should be noted that the radius margin is only a computationally inexpensive heuristic to reduce the influence of the surface sampling noise. It is not a permanent fix for the problem. In fact, interpolation on the edges along with triangulation is a better solution, especially when the mesh resolution varies widely across the mesh. Figure 2.4 illustrates the idea of the interpolation and triangulation strategy. In Figure 2.4(a), the vertices marked by the gray dots are in the surface patch while the vertex v_1, marked by the white dot, is not. Interpolation is performed on the edge connecting v_1 and v_5. This edge is on the shortest path from v_1 to the center of the surface patch. u_1 is the resultant vertex of the interpolation. Local triangulation is performed after the interpolation and four new triangles are created. The two lightly shaded triangles along with the heavily shaded triangles are in the surface patch.

When using the interpolation and triangulation strategy, the condition under which an edge should be split is that both vertices that the edge faces should be in the surface patch. If this condition is not satisfied, as shown in Figure 2.4(b), then no interpolation is performed along that edge. In Figure 2.4(b), because v_2 is not in the surface patch, the edge connecting v_1 and v_5 is not split. In contrast, interpolation is performed along the edge connecting v_2 and v_6 and vertex u_2 is created. The resultant surface patch contains the two lightly shaded triangles along with the heavily shaded ones.

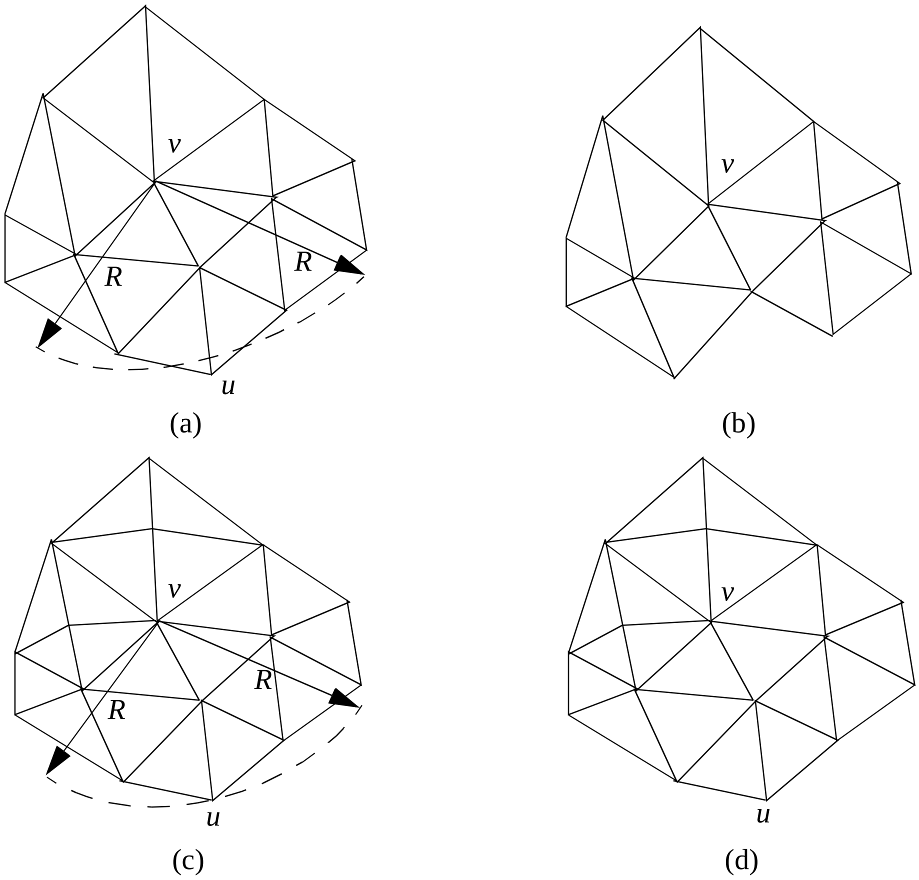


Figure 2.3: Illustration of radius margin. (a), (c) Surfaces that have the same shape but different sampling; (b) the surface patch generated using the center vertex *v* and radius *R* on the surface in (a). Vertex *u* is not included in the patch because the shortest distance from *u* to *v* is slightly greater than *R*. If using the 5% radius margin, then vertex *u* will be included in the patch. The resultant patch is similar to the one in (d). (d) The surface patch generated using the center vertex *v* and radius *R* on the surface in (b). Vertex *u* is included in the patch because the shortest distance from *u* to *v* is equal to *R*.

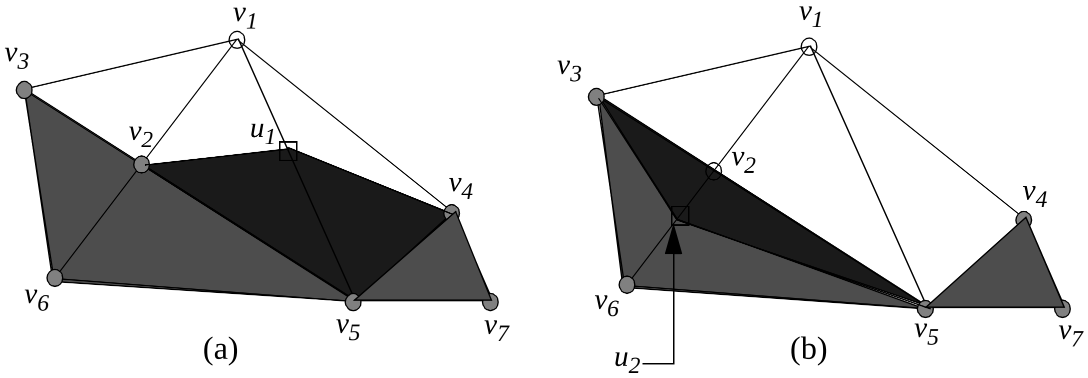


Figure 2.4: Illustration of the interpolation and triangulation strategy.

2.3 Interior Mapping

The theory of harmonic maps has been briefly reviewed in Section 2.1. It is clear that the solution to harmonic maps is the solution to a partial differential equation. Because we deal with discrete surfaces in practice and the computation cost for solving partial differential equations is high, it is more appropriate and practical if some approximation approaches can be used to compute harmonic maps. In [26], an approximation method is proposed; the method consists of two steps: interior mapping and boundary mapping. In our approach, we use an interior mapping which is similar to that in [26], but a different boundary mapping. In this section, assuming that the boundary mapping is already known, the construction of the interior mapping is discussed in detail. The boundary mapping will be discussed in Section 2.4.

Defining Energy Functional

Let *D(v, R)* be a 3D surface patch with central vertex *v* and radius *R* measured by surface distance. Let *P* be a unit disc in a two-dimensional plane. Let ∂D and ∂P be the boundary of *D* and *P*, respectively. Let $v_i, i = 1, \ldots, n$, be the interior vertices of *D*. The interior mapping ϕ maps $v_i, i = 1, \ldots, n$ onto the interior of the unit disc *P* with a given boundary mapping *b*: $\partial D \to \partial P$. ϕ is obtained by minimizing the following energy functional[26]:

$$E(\phi) = \frac{1}{2} \sum_{\{i,j\} \in Edges(D)} k_{ij} \|\phi(i) - \phi(j)\|^2 \tag{2.3}$$

In (2.3), for the simplicity of notation, $\phi(i)$ and $\phi(j)$ are used to denote $\phi(v_i)$ and $\phi(v_j)$ which are the images of the vertices v_i and v_j on *P* under the mapping ϕ. The values of $\phi(i)$ and $\phi(j)$ define the mapping ϕ. k_{ij} serve as spring constants which will be discussed shortly.

The intuition of the energy functional is as follows. An instance of the functional $E(\phi)$ can be interpreted as the energy of a spring system by associating each edge in *D* with a spring. Then the mapping problem from *D* to *P* can be considered as adjusting the lengths of those springs when flattening them down onto *P*. If the energy of *D* is zero, then the energy increases when the mesh is flattened down to *P* because all the springs are deformed. Different ways of adjusting the spring lengths correspond to different mappings ϕ. The best ϕ minimizes the energy functional $E(\phi)$.

The minimum of the energy functional $E(\phi)$ can be found by solving a sparse linear least-square system of (2.3) for the values $\phi(i)$. Taking the partial derivative of $E(\phi)$ with respect to $\phi(i)$, $i = 1, \ldots, n$ and making it equal to zero yield the following equations:

$$\frac{\partial E(\phi)}{\partial \phi(i)} = k_{ij}(\phi(i) - \phi(j)) + k_{ik}(\phi(i) - \phi(k)) + k_{il}(\phi(i) - \phi(l)) + \ldots, i = 1, \ldots, n \tag{2.4}$$

$$= \sum_{\{i,j\} \in 1-Ringof\{i\}} k_{ij}(\phi(i) - \phi(j)) = 0, i = 1, \ldots, n \tag{2.5}$$

In (2.5), *1-Ring of {i}*, illustrated in Figure 2.5, refers to the polygon constructed by the immediate neighboring vertices of the vertex v_i. Equation (2.5) can be rewritten as

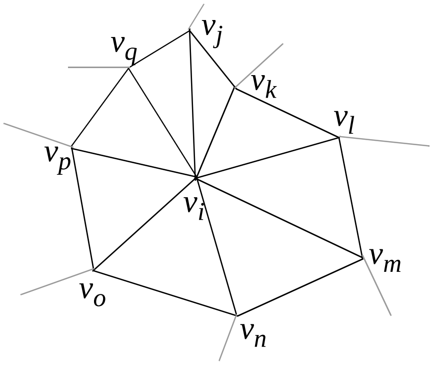


Figure 2.5: Illustration of the 1-Ring of a vertex on a triangular mesh.

$$A_{nxn}X_{nx2} = b_{nx2} \tag{2.6}$$

In (2.6), $X_{nx2} = [\phi(1), \ldots, \phi(n)]^T$, $\phi(i) = [\phi_x(i), \phi_y(i)]$. X_{nx2} denotes the unknown coordinates of the interior vertices of *D* when mapped onto *P* under ϕ. Because A_{nxn} contains the connectivity information of the surface patch *D(v, R)*, it is a sparse matrix that has the following structure (Figure 2.6). If *D* is considered to be a bi-directional graph, then A_{nxn} can be interpreted

$$A_{nx\,n} = \begin{array}{ccccccccc|c} 1 & \ldots & m & \ldots & i & \ldots & j & \ldots & n & \\ * & & & & & & & & & 1 \\ & \ddots & & & & & & & & \vdots \\ & & * & & * & & & & & m \\ & & & \ddots & & & & & & \vdots \\ & & * & & * & & * & & & i \\ & & & & & \ddots & & & & \vdots \\ & & & & * & & * & & & j \\ & & & & & & & \ddots & & \vdots \\ & & & & & & & & * & n \end{array}$$

Figure 2.6: Structure of the matrix A_{nxn}.

as the adjacency matrix of the graph. All the diagonal entries of A_{nxn} are non-zero. For an arbitrary row *i* in A_{nxn}, if vertex v_i is connected to vertices v_j and v_m, *i.e.*, v_j and v_m are in the *1-*

Ring of $\{v_i\}$, then only the *j*th and *m*th entries in row *i* are non-zero. Similarly, the *i*th entry in row *m* and row *j* are also non-zero. Therefore, in addition to the usual properties that matrix A_{nxn} has in least-square problems, A_{nxn} is sparse in this particular case. The boundary condition is accommodated in the matrix b_{nx2}. In X_{nx2}, if a vertex v_i is connected to boundary vertices, then its corresponding entry *i* in b_{nx2} is weighted by the coordinates of those boundary vertices. Otherwise, the entries in b_{nx2} are zero.

Solving the Sparse Linear System

In our current implementation, (2.6) is solved using the conjugate gradient descent algorithm. In addition to the usual steps in implementing that algorithm, special attention is paid to the storage of the matrix A_{nxn} in (2.6) because A_{nxn} is a sparse matrix. Storing A_{nxn} as a *n* by *n* matrix would waste a great deal of memory. In our current implementation, the *indexed storage* strategy introduced in [64] is adopted to store A_{nxn} as two arrays with one being integer array and the other being double array. The index storage strategy is illustrated by the following example. Suppose that

$$A = \begin{bmatrix} 3 & 0 & 1 & 0 & 0 \\ 0 & 4 & 0 & 0 & 0 \\ 0 & 7 & 5 & 9 & 0 \\ 0 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 6 & 5 \end{bmatrix} \tag{2.7}$$

In order to represent a sparse matrix A_{nxn} like the one in (2.7) using the row-indexed scheme, two one-dimensional arrays need to be set up. One is an integer array called *ija* and the other is a double array called *sa*. The following rules are applied to created the two arrays:

- The first *N* locations of *sa* store *A*'s diagonal matrix elements in order.
- Each of the first *N* locations of *ija* stores the index of the array *sa* that contains the first off-diagonal element of the corresponding row of the matrix. If there are no off-diagonal elements for that row, it is one greater than the index in *sa* of the most recently stored element of a previous row.
- Location 1 of *ija* is always equal to *N*+2 which can be read to determine *N*.
- Location *N*+1 of *ija* is one greater than the index in *sa* of the last off-diagonal element of the last row. It can be read to determine the number of nonzero elements in the matrix, or the number of elements in the arrays *sa* and *ija*. Location *N*+1 of *sa* is not used and can be set arbitrarily.
- Entries in *sa* at locations $\geq N + 2$ contain A_{nxn}'s off-diagonal values, ordered by rows and, within each row, ordered by columns.
- Entries in *ija* at locations $\geq N + 2$ contain the column number of the corresponding elements in *sa*.

Using the above rules to construct the indexed storage arrays for the matrix A in (2.7) yields the result shown in Figure 2.7. In that figure, it should be noticed that, according to the above storage rules, the value of *N* (namely 5 in this case) is *ija*[1]-2, and the length of each array is *ija*[*ija*[1]-1]-1, namely 11. The first five elements in *sa* are the diagonal values of A in order. The 6th value is *x*, which represents an arbitrary value. The elements numbered from 7 to 11 in *sa* are the off-diagonal entries of each row in *A*. For each row *i* in *A*, the off-diagonal elements are stored in *sa*[*k*] where *k* loops from *ija*[*i*] to *ija*[*i*+1]-1. For example, let us find the off-diagonal elements in the first row of A in (2.7). From the index in *ija*[1] and *ija*[2], it can be determined that there is one off-diagonal element in that row. The value of this element is located in *sa*[*ija*[1]], *i.e.*, *sa*[7], which is equal to 1.0. The column index of this element in the original matrix *A* is located in *ija*[*ija*[1]], *i.e.*, *ija*[7], which is 3. Checking the original matrix *A* will verify the results we just obtained.

index *k*	1	2	3	4	5	6	7	8	9	10	11
ija[*k*]	7	8	8	10	11	12	3	2	4	5	4
sa[*k*]	3.0	4.0	5.0	0.0	5.0	x	1.0	7.0	9.0	2.0	6.0

Figure 2.7: The row-indexed storage arrays for the matrix in (2.7).

Defining Spring Constants

There are different ways of assigning the spring constants in (2.3). One is to define the spring constant as in (2.8)[26]

$$k_{ij} = ctg\theta(e_{mi}, e_{mj}) + ctg\theta(e_{li}, e_{lj}) \tag{2.8}$$

in which $\theta(e_{mi}, e_{mj})$ and $\theta(e_{li}, e_{lj})$ are defined in Figure 2.8.

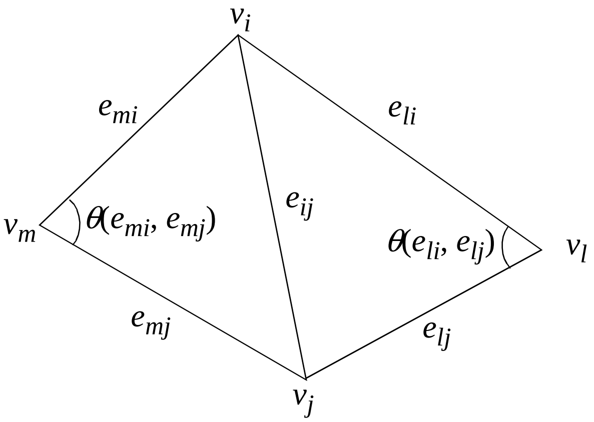


Figure 2.8: Definition of spring constants using angles.

If e_{ij} is associated with only one triangle, then there will be only one term on the right-hand side of (2.8). The intuition behind this definition is that long edges subtending to big angles are given relatively small spring constants compared with short edges which subtend to small angles. Recall the energy functional defined in (2.3); this definition of the spring constant means that long edges will remain long in the harmonic image while short edges will remain short. The reason is that the springs associated with long edges have smaller spring constants compared with the springs associated with short edges. Figure 2.9 shows an example of the interior mapping.: a hemisphere represented by triangular mesh. It should be noted that most of the vertices are distributed evenly on the mesh except at the parts indicated by arrows. In other words, most of the edges have equal length on the mesh. The harmonic image of the mesh is shown in Figure 2.9(b). The short edges indicated in Figure 2.9(a) by arrows are still short in (b) while other edges in (a) remain long in (b) and they have approximated equal length locally. There is obvious boundary distortion on the harmonic image due to the discrete nature of the triangular mesh. The distortion affects the interior mapping; therefore, most of the edges in (b) do not have equal length as they do in (a). Figure 2.10 shows the harmonic image of a surface patch from a real object.

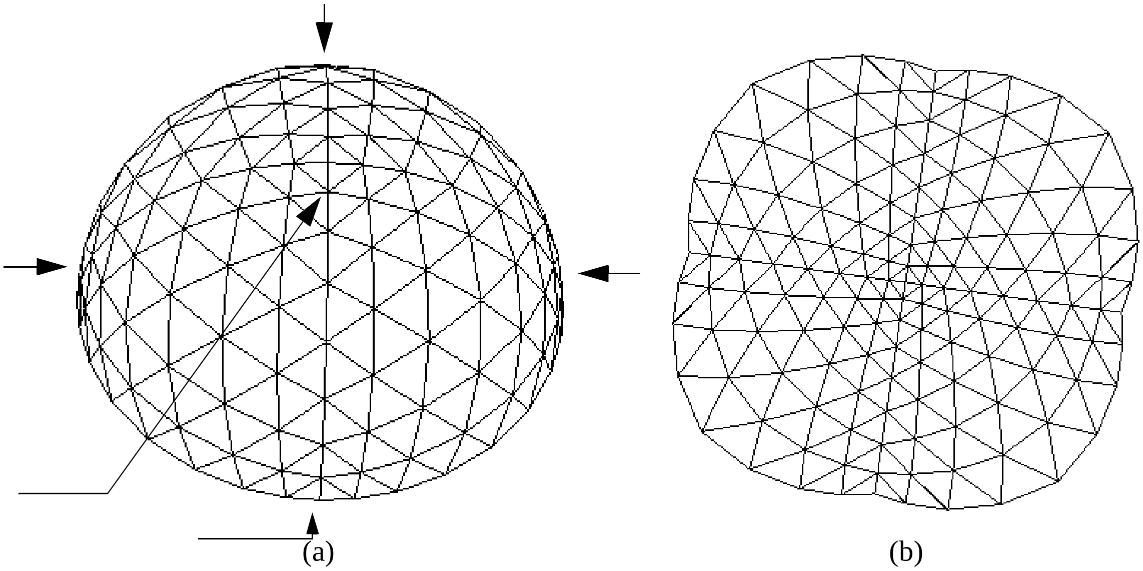
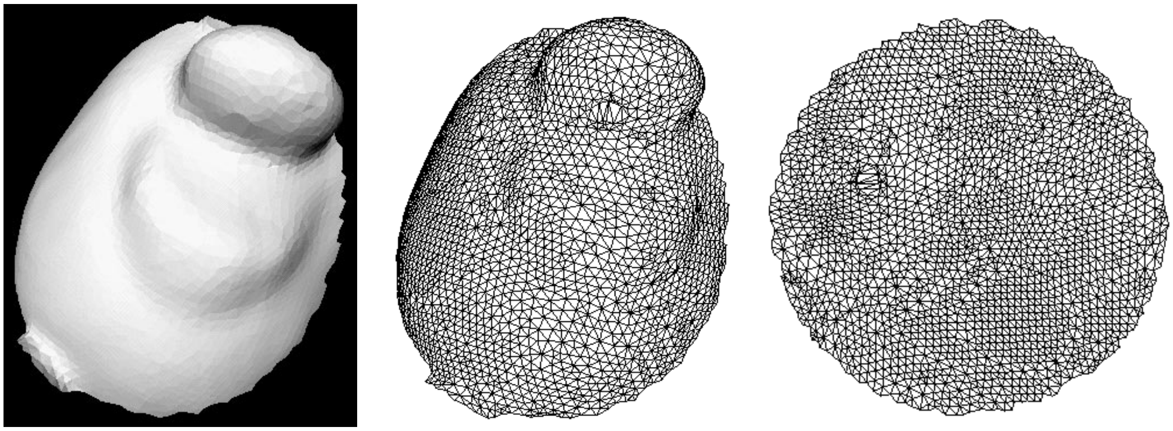


Figure 2.9: Example of interior mapping. (a) A hemisphere represented by triangular mesh; (b) the harmonic image of (a).

From the above discussion, it can be seen that the energy functional in (2.3) tries to preserve the ratio of edge lengths on the original surface patch *D(v, R)* by defining the spring constants as shown in (2.8). Given a specific sampling of a surface (a surface can be sampled in different ways), the ratio of edge lengths is closely related to the shape of the surface. Therefore, by preserving the ratio of edge lengths, the interior mapping ϕ preserves the shape of the *D(v, R)* when mapping it onto the unit disc *P*. Of course there is distortion when mapping *D(v, R)* onto *P*. The distortion is minimized by ϕ.

In order to preserve the ratio of edge lengths, another way to define the spring constant is to use the inverse of the edge length as shown in (2.9).

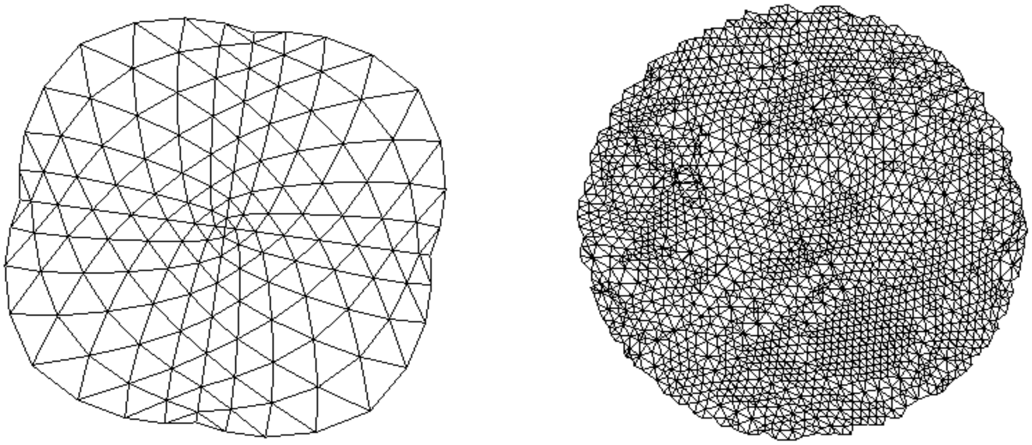


(a) (b) (c)

Figure 2.10: An example of the interior mapping using a real object. (a) Original surface patch (rendered); (b) original surface patch shown in wireframe; (c) the harmonic image of (b).

$$k_{ij} = \frac{1}{\left\| e_{v_i v_j} \right\|} \tag{2.9}$$

Similar to (2.8), the springs associated with longs edges have small spring constants while the springs associated with short edges have large spring constants. Using the definition in (2.9), the interior mappings for the surfaces shown in Figure 2.9(a) and Figure 2.10(a) are computed and shown in Figure 2.11.



(a) (b)

Figure 2.11: Harmonic images computed using the inverse of edge length as spring constant. (a) Harmonic image of the hemisphere shown in Figure 2.9(a); (b) harmonic image of the surface shown in Figure 2.10(a).

According to the experiments that have been conducted, there is little difference between the harmonic images resulted from the two approaches to defining the spring constants. For example, the harmonic images in Figure 2.9(b) and Figure 2.11(a) are almost the same. So are the images in Figure 2.10(c) and Figure 2.11(b). For the second case, for each vertex on the two harmonic images, the length of the difference vector is plotted in Figure 2.13. The curve shows that little variation in postion for each vertex on the harmonic images results from using different definitons of the spring constant. The unit of the vertical axis in Figure 2.13 is the radius of the 2D unit disc.

When the ratio of edge length varies significantly across the surface patch (Figure 2.12(b)), the difference between the harmonic images obtained using different definitions of spring constants is more visible (Figure 2.12(c) and (d)). The length of the difference vector for each vertex is plotted in Figure 2.14. However, even in this case, there is little effect on the Harmonic Shape Images which are created based on harmonic maps. Harmonic Shape Images will be discussed in Section 2.5.

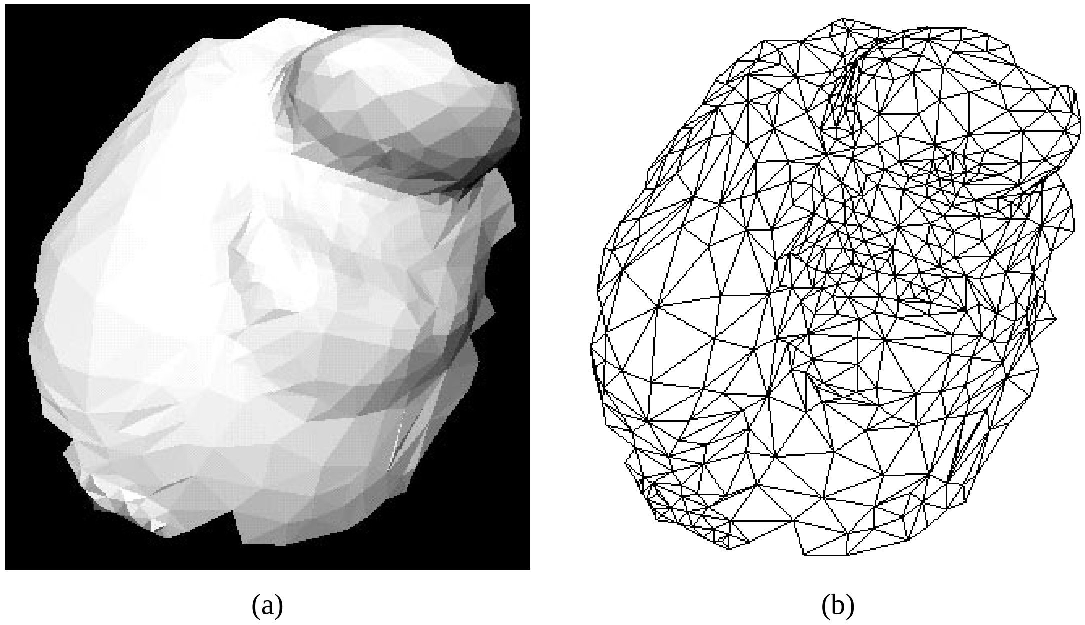
As a summary, based on the experiments we have conducted, no evidence has been found to show that one definition is better than the other. The angle definition is used in all experiments.

2.4 Boundary Mapping

As discussed in Section 2.3, the unit disc *P* is selected to be the target domain of the harmonic mapping. In this section, the construction of boundary mapping, which maps the boundary of the given surface patch *D(v, R)* onto the boundary of *P* is discussed.

The construction of the boundary mapping is illustrated in Figure 2.15. First of all, let us define the vertices and vectors in Figure 2.15. *O* is the central vertex of the surface patch *D(O, R)* and *O'* is the center of the unit disc *P*. v_i, *i*=1,…,5 are the boundary vertices of *D(O, R)*. For some boundary vertices, e.g., v_i^r, *i*=1,…,4, the surface distance between any of them and the central vertex *O* is equal to *R*; these are called *radius boundary vertices*. For other boundary vertices, e.g., v_1^o, the surface distance is less than *R*; these are called *occluded boundary vertices*. Radius boundary vertices are determined by the radius of the surface patch, while occluded boundary vertices are determined either by self occlusion or by occlusion by other objects. The vector from the central vertex *O* to a radius vertex v_i^r is called a *radius vector,* while the vector from *O* to an occluded boundary vertex v_j^o is called an *occlusion vector.*

Now let us define the angles in Figure 2.15. Angles a_i, *i*=1,…,4 are the angles between two adjacent radius vectors v_i^rO and v_{i+1}^rO. Angles b_j, *j*=1, 2, are the angles between two adjacent occlusion vectors, or one occlusion vector and one adjacent radius vector, in an occlusion range. An occlusion range is a consecutive sequence of occlusion boundary vertices except for the first and last ones. For example, (v_4^r, v_1^o, v_1^r) is an occlusion sequence. The sum of b_j over an occlusion sequence is the angle a_i formed by the first and last radius vectors, for example, the sum of b_j over (v_4^r, v_1^o, v_1^r) is a_1.



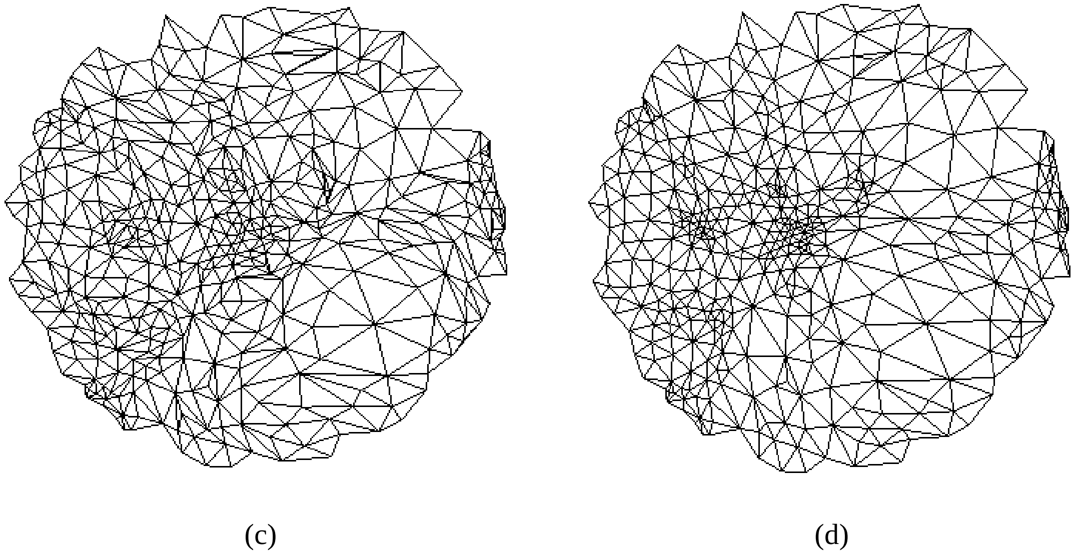


Figure 2.12: An example mesh with large variation of the ratio of edge lengths. (a) Rendered surface; (b) surface in wireframe; (c) harmonic image using the angle definition; (d) harmonic image using the inverse of edge length definition.

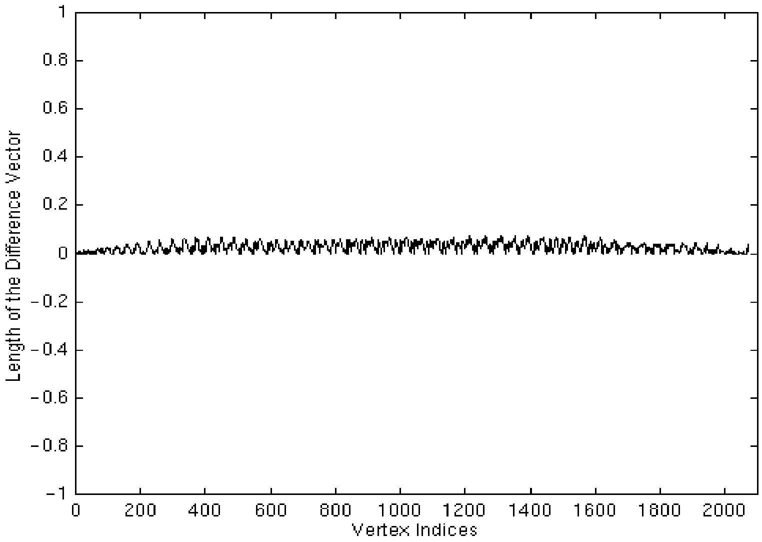


Figure 2.13: The position difference for each vertex on the two harmonic images obtained using different definitions of the spring constant. The original surface patch is shown in Figure 2.10(a). Harmonic images are defined on the 2D unit disc. The unit of the vertical axis is the radius of the 2D unit disc.

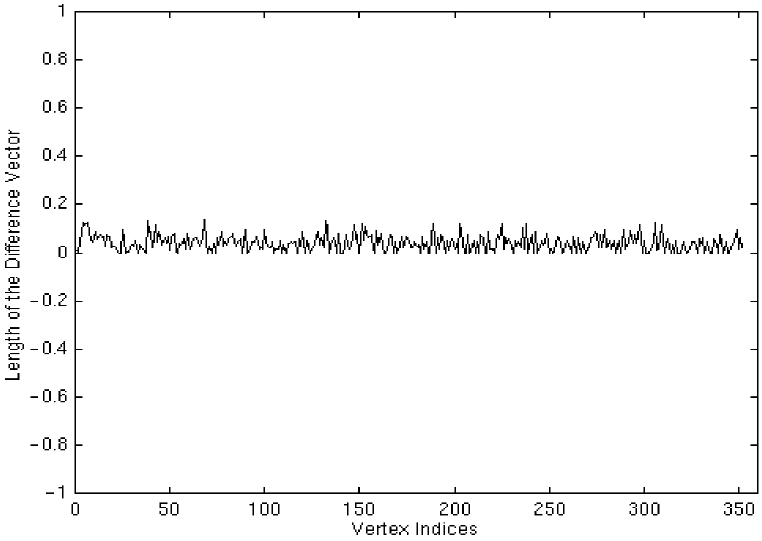


Figure 2.14: The position difference for each vertex on the two harmonic images obtained using different definitions of the spring constant. The original surface patch is shown in Figure 2.12(a). Harmonic images are defined on the 2D unit disc. The unit of the vertical axis is the radius of the 2D unit disc.

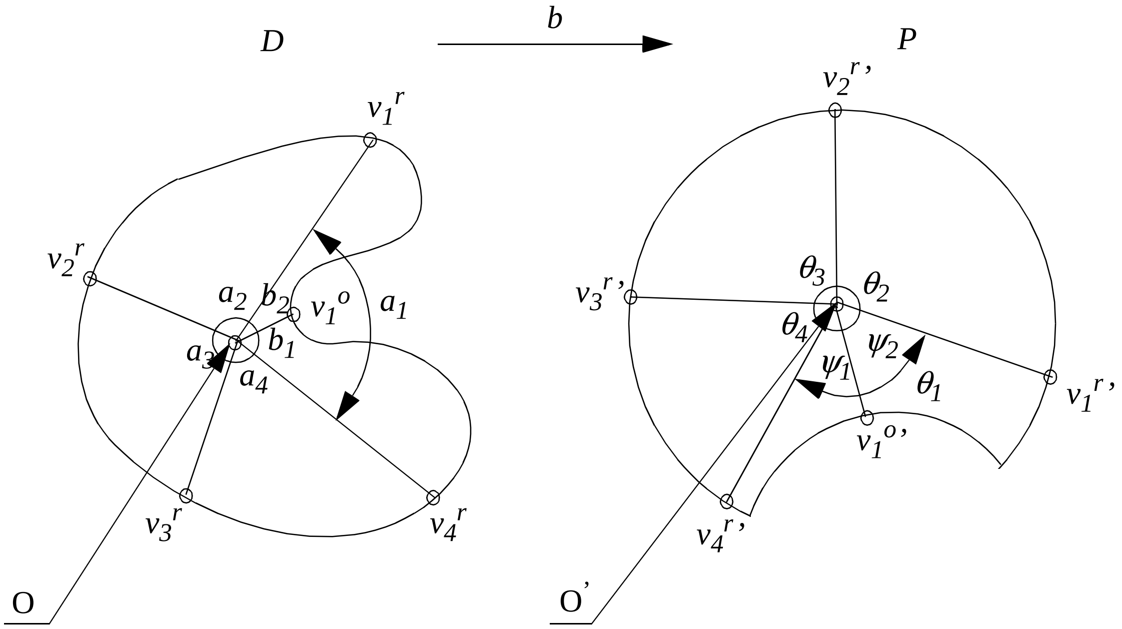


Figure 2.15: Illustration of the boundary mapping.

The construction of the boundary mapping consists of two steps. At the first step, the radius boundary vertices are mapped onto the boundary of the unit disc *P*, which is a unit circle. In Figure 2.15, v_i^r, i=1,…,4, are mapped to $v_i^{r'}$, i=1,…,4, respectively. Once the angles a_i are determined, the positions of $v_i^{r'}$ are determined. θ_i is computed as follows:

$$\theta_i = \frac{a_i}{\sum_{k=1}^{n} a_k} 2\pi \qquad (2.10)$$

At the second step, the occlusion boundary vertices in each occlusion range are mapped onto the interior of the unit disc *P*. For example, in Figure 2.15, v_1^o, which is in the occlusion range (v_4^r, v_1^o, v_1^r), is mapped onto $v_1^{o'}$. Once the angles ψ_j and the radii r_j are determined, the position of v_1^o is determined. ψ_j are computed as follows:

$$\psi_j = \frac{b_j}{\sum_{m=1}^{n} b_m} \qquad (2.11)$$

in which *n* is the number of angles within the occlusion range. The sum in the denominator is equal to the angle formed by the first and last radius vectors of the occlusion range. In Figure 2.15, it is equal to a_1. r_j is defined to be

$$r_j = \frac{dist(v_j^o, O)}{R} \tag{2.12}$$

in which $dist(v_j^o, O)$ is the surface distance between the occlusion boundary vertex v_j^o and the central vertex O. R is the radius of the surface patch $D(O, R)$.

The intuition behind the boundary mapping is that the ratio of the boundary edge lengths is preserved as much as possible on the boundary of *P*. This is consistent with the way the interior mapping is constructed. Therefore, the harmonic mapping minimizes the shape distorsion when mapping the surface patch *D(v, R)* onto the 2D unit disc *P*.

In the following sections, we discuss ordering the boundary vertices and selecting a starting boundary vertex.

Ordering Boundary Vertices

The boundary vertices of *D* need to be ordered in either a clock-wise or counter-clock-wise manner before constructing the boundary mapping. Similarly, when mapping vertices onto the boundary of *P*, either clock-wise or counter-clock-wise order needs to be determined. To facilitate matching, the two orders must remain consistent for all surface patches. In our implementation, a vector is assumed to be known as the reference vector for a given surface patch. The viewing direction along which the range image of the surface patch is taken can be used as the reference vector. The order of the boundary vertices is determined according to this vector.

Selecting Starting Boundary Vertex

When constructing the boundary mapping, if the starting vertex on the boundary of *D* is always mapped to the same vertex on the boundary of *P*, then different starting vertices will result in different boundary mappings. This will, in turn, result in different interior mappings. For example, a surface patch is shown in Figure 2.16(a) and (b). With different starting vertices indicated by the arrow, two different harmonic images are obtained (Figure 2.16(c) and (d)).

A closer examination on the harmonic images in Figure 2.16(c) and (d) shows that one image is the rotated version of the other. In fact, the harmonic images with different starting vertices are different by a planar rotation because neither the angles θ_i in (2.10) nor the angles ψ_i in (2.11) change with respect to different starting vertices. Nor does the radius r_j in (2.12) change. Therefore, the starting vertex can be selected randomly. The rotation difference will be found later by the matching process which will be discussed in the next chapter.

2.5 Generation of Harmonic Shape Images

In Sections 2.2, 2.3 and 2.4, we have discussed defining a surface patch and obtaining its harmonic image. In this section, we will explain how to construct its Harmonic Shape Image from its harmonic image.

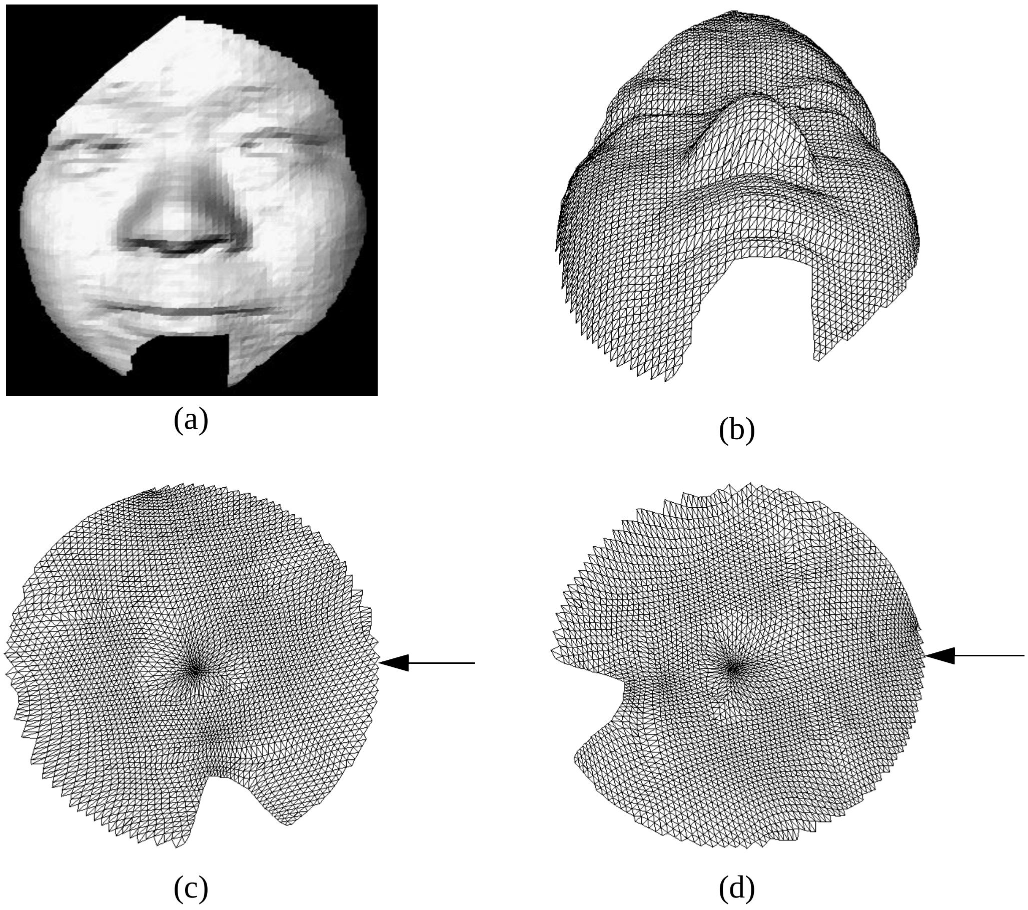


Figure 2.16: The effect of starting vertex on harmonic images. (a) A surface patch (rendered); (b) the wireframe version of (a). It is displayed from a different viewing direction from that in (a) in order to show the shape of the surface patch clearly. (c) The harmonic image of (a) with the starting vertex indicated by the arrow; (d) the harmonic image of (a) with a different starting vertex indicated by the arrow.

Recall from previous sections that one important property of harmonic images is the one-to-one correspondence between the vertices on the surface patch *D(v, R)* and those on its harmonic image *HI(D(v, R))*. This means that, all distribution functions defined on a surface patch can be recorded accordingly on its harmonic image. In other words, harmonic images provide a general framework for representing distribution functions defined on surfaces, e.g., surface normal, shape, color, texture, material. To describe the shape of free-form surfaces, the shape information, e.g., curvature, is stored at each vertex of the harmonic image *HI(D(v, R))*. The resultant image is called a Harmonic Shape Image, which is denoted as *HSI(D(v, R))*. If texture information is stored on *HI(D(v, R))*, the resultant image is called a Harmonic Texture Image.

Curvature is the shape descriptor of surfaces. However, there are two difficulties in computing curvature in practice. One is that we deal with discrete surfaces. The other is that computing curvature involves computing the second order derivatives, which are sensitive to noise. One way to address these issues is that smooth surfaces can be fit to the triangular mesh locally so that the curvature can be computed. This approach may be computationally expensive because of non-linear minimization. Another approach is to approximate the curvature distribution function using *discrete curvature*. In the following, three different approaches to computing discrete curvature are discussed.

Simplex Angle

The first discrete curvature, Simplex Angle proposed in [19], is used to describe the shape at each vertex in a mesh with the appropriate regularities. A topology constraint on the mesh requires that each vertex has three neighboring vertices. Another requirement is that the projection of each vertex onto the place formed by its three neighboring vertices should coincide with the center of the triangle formed by those three vertices. Suppose P is a vertex on a given mesh M and P_1, P_2, P_3 are its neighboring vertices. The Simplex Angle at P is defined in Figure 2.17. In Figure 2.17(a),

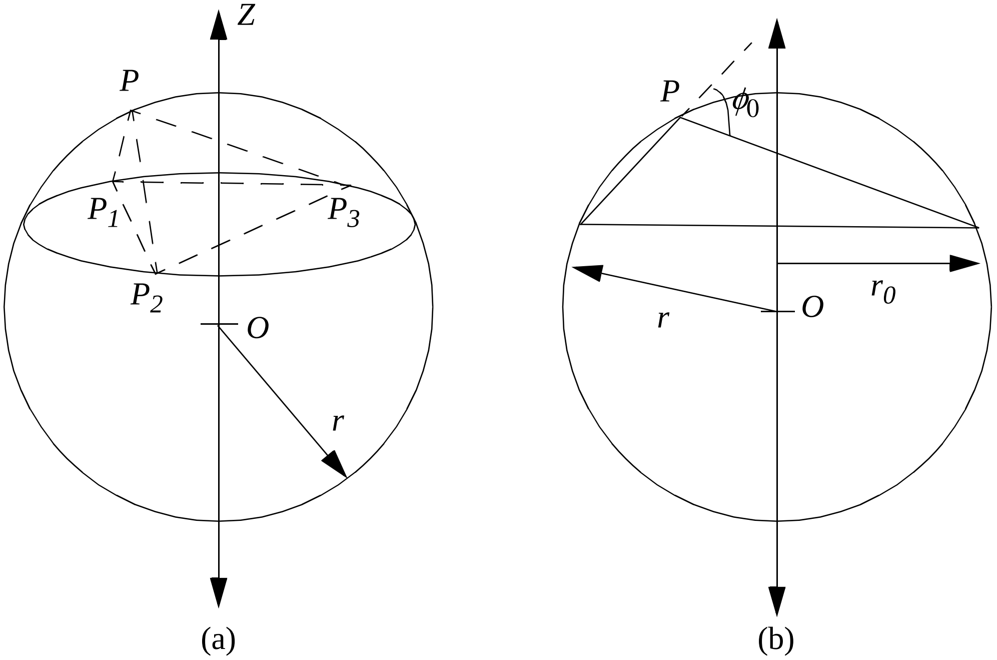


Figure 2.17: Definition of Simplex Angle.

O is the center of the sphere circumscribed to the tetrahedron (P, P_1, P_2, P_3). Z is the line passing through O and through the center of the circle circumscribed to (P_1, P_2, P_3). Consider the cross section of the surface by the plane Π containing Z and P. The intersection of Π with the tetrahedron is a triangle. One vertex of the triangle is P, and the edge opposite to P is in the plane formed by (P_1, P_2, P_3) (Figure 2.17(b)). The angle, ϕ_0, between the two edges of the triangle intersecting at P is defined to be the Simplex Angle at P.

The Simplex Angle varies between $-\pi$ and π and is equal to zero for a flat surface. It is large in absolute value if P is far from the plane of its three neighbors. Suppose that the normal to the plane is pointing toward the outside of the surface; the Simplex Angle is negative if the surface is locally concave and positive if the surface is locally convex (Figure 2.18).

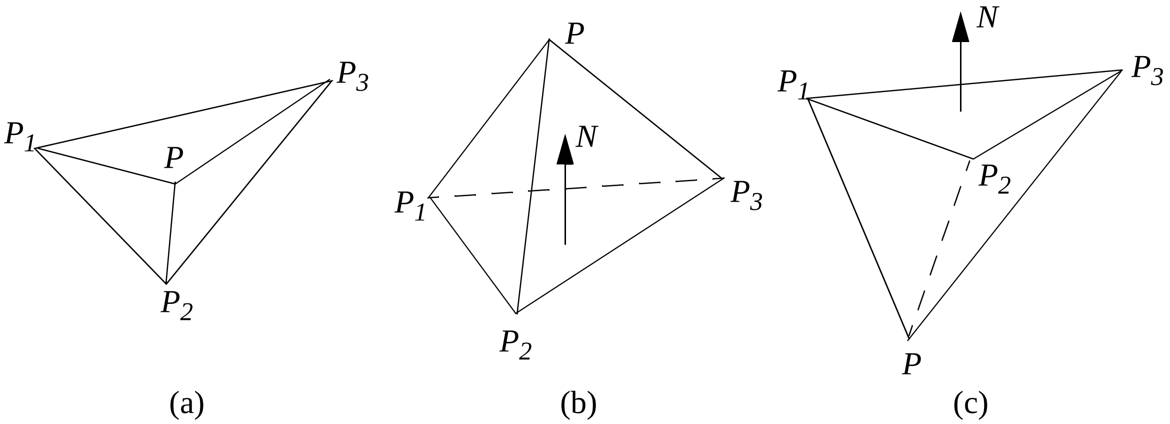


Figure 2.18: Typical values of Simplex Angles. (a) Flat surface: ϕ_0=0; (b) convex peak: ϕ_0 close to π; (c) concave peak: ϕ_0 close to -π.

One issue that should be noted when computing Simplex Angle is that, for arbitrary triangular meshes, the connectivities among vertices are arbitrary, i.e., a vertex can have any number of neighboring vertices. However, in order to compute Simplex Angle, each vertex must have three neighboring vertices. In order to address the connectivity issue, it is necessary to compute Simplex Angle for the dual mesh of the given mesh. The dual mesh is obtained by taking the center of each triangle on the given mesh and connecting the centers of neighboring triangles as shown in Figure 2.19. The simplex angle for each vertex on the dual mesh angle can be computed as long as it has three neighboring vertices. In fact, it is the simplex angle for each face on the original mesh. The Simplex Angle at each vertex of the original mesh can be taken as the average of the Simplex Angles of the faces intersecting at that vertex.

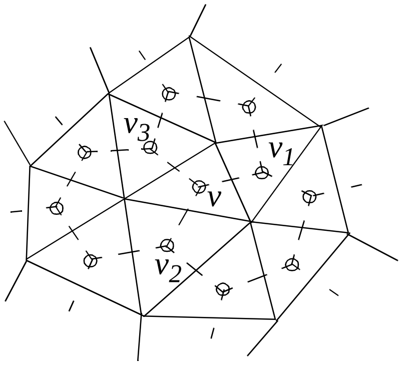


Figure 2.19: The dual mesh of a given triangular mesh. The solid lines represent the original triangular mesh. The dashed lines represent the dual mesh.

Complete Angle

The second method for computing discrete curvature is to use the complete angles[1] at each vertex of the triangular mesh(Figure 2.20). The complete angle at a vertex *v* is defined to be

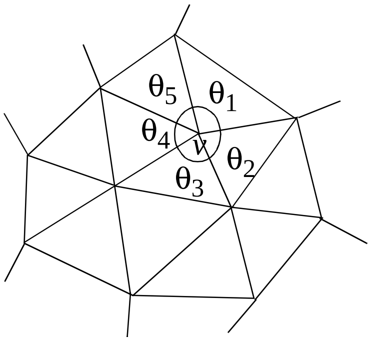


Figure 2.20: Definition of discrete curvature using angles.

$$\theta = \sum_{i=0}^{'''} \theta_i \tag{2.13}$$

in which *m* is the number of neighboring vertices around *v*. The discrete curvature *c* is defined to be

$$c = 2\pi - \theta \tag{2.14}$$

The definition in (2.14) does not give the sign of the discrete curvature at *v*, i.e., it describes how much the surface is curved at *v*, but does not indicate whether the surface is locally convex or concave. Similar to computing the Simplex Angle at *v*, the sign of *c* can be determined by taking the average of the Simplex Angles of the faces intersecting at *v*.

Attention should be paid to the boundary vertices when computing the discrete curvature using (2.14). Because the angles are not complete for the boundary vertices, their curvature values should be labelled invalid.

Weighted Dot Product of Normals

The third method for computing discrete curvature is to use area-weighted dot product of normals as illustrated in Figure 2.21. This discrete curvature is defined for each face of a given triangular mesh. For a face *f* with surface normal *N*, its local neighborhood with a certain size is defined first. Then for each triangle in this neighborhood, compute the dot product of its normal N_i with *N* and weight the dot product by its area A_i. The curvature of *f* is defined to be

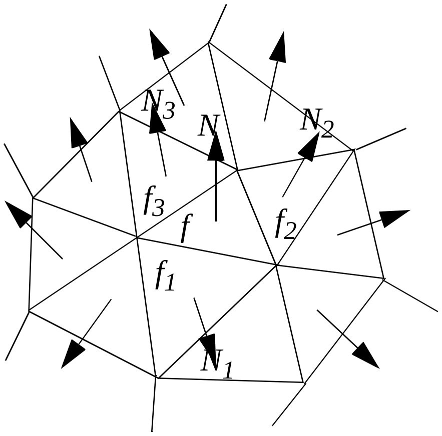


Figure 2.21: Illustration of defining discrete curvature using area-weighted dot product of normals.

$$c = \frac{\sum_{i=0}^{m} A_i (N_i \bullet N)}{\sum_{i=0}^{m} A_i} \tag{2.15}$$

This definition can be interpreted using the concept of normal curvature. By taking the weighted average of the dot product of the normals, the definition in (2.15) measures the change of the surface normal in all directions. Using the area of each triangle as the weight takes into account the sampling resolution. Similar to the definition in (2.14), this definition does not give the sign of the curvature. The sign can be determined using the same method as that used to compute the Simplex Angle. After the curvature value is computed for each face on a given mesh, the vertex curvature values can be computed by taking the average of the values of the faces intersecting at that vertex. Again, triangle area can be used to weight the average.

Using the above three definitions for discrete curvature, the Harmonic Shape Images for the surface patch in Figure 2.16(a) are computed and shown in Figure 2.22 The three images in Figure 2.22 reflect the same shape features. Similar to the result found in [21], the Simplex Angle is better than the Complete Angle at differentiating shapes with small variation. This can also be seen in Figure 2.22 where the shape variation in (a) has more clarity than that in (b) on the cheek and forehead of the face. In the experiments that will be discussed in later chapters, Simplex Angle is used to generate Harmonic Shape Images for shape comparison.

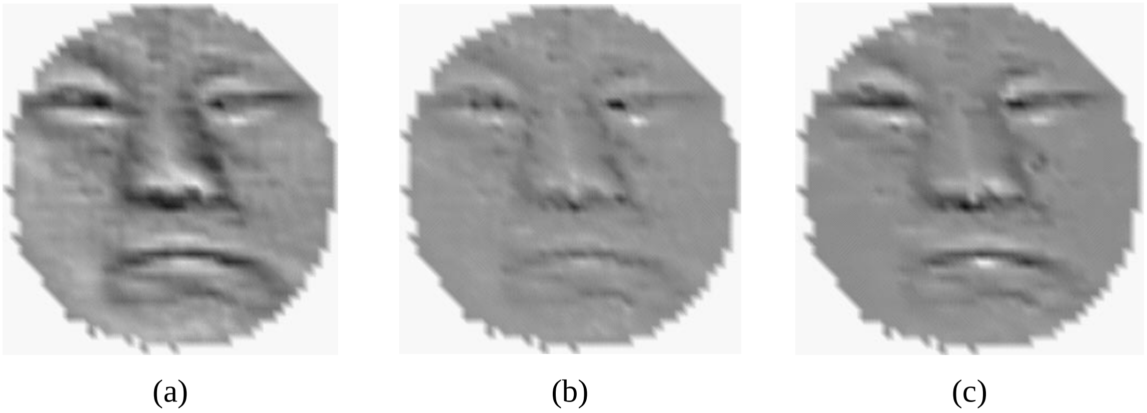


Figure 2.22: Harmonic Shape Image of the surface patch in Figure 2.16(a) using (a) Simplex Angle; (b) complete angle; (c) weighted dot product of normals.

2.6 Complexity Analysis

Suppose there are n_t triangles formed by n vertices on the input surface D. α is the ratio between the number of boundary vertices and n. n_e is the number of edges in D. n_r denotes the number of resampling vertices in the 2-D domain P. In general, n_r is about the same as n. Table 2.1 lists the main functions in generating Harmonic Shape Images along with their computation complexities.

Table 2.1: Functions and their computation complexity

Function	complexity
ConstructEdgeListOnSurface	$O(n_e \log(n_e))$
ConstructBoundaryMapping	$O(n_e)+O(\alpha n \log(\alpha n))$
ConstructInteriorMapping	$O((1-\alpha)n \log((1-\alpha)n)$
ResampleInputSurface	$O(n_r)$
ComputeShapeInvariant	$O(n_r)$

Chapter 3

Matching Harmonic Shape Images

It has been discussed in Chapter 2 that Harmonic Shape Images are shape representations for 3D free-form surfaces. The comparison of 3D surfaces can be performed by comparing their Harmonic Shape Images. In this chapter, we discuss how to match Harmonic Shape Images and how to use this technique to match surfaces. Related issues such as shape similarity and the coarse-to-fine matching strategy are also discussed.

3.1 Matching Harmonic Shape Images

From the generation process of Harmonic Shape Images, it can be seen that they are no different from general 2D images except that they are not in the *m*-by-*n*-pixel format. Both horizontal and vertical scanning of HSIs can result in such a format. This implementation issue will be discussed later in this chapter. For now, Harmonic Shape Images are considered to be the same as general 2D images. The comparison of two Harmonic Shape Images can be performed using the normalized correlation(3.1).

$$R(HSI_1,HSI_2) = \frac{N\sum p_i q_i - \sum p_i \sum q_i}{\sqrt{\left(N\sum p_i^2 - \left(\sum p_i\right)^2\right)\left(N\sum q_i^2 - \left(\sum q_i\right)^2\right)}} \tag{3.1}$$

In (3.1), HSI_1, HSI_2 are two Harmonic Shape Images, p_i and q_i are corresponding pixels in HSI_1 and HSI_2, respectively. N is the number of correspondences.

It has been discussed in Chapter 2 that there may be a planar rotation difference between two Harmonic Shape Images due to different choices of the starting vertex for the boundary mapping. This rotation difference needs to be found by the matching process. This means that the correlation coefficient between two surface patches is a function of the rotation angle θ between their harmonic images.

$$R(HSI_1,HSI_2) = \max_{\theta}(R(\theta)) \tag{3.2}$$

The most direct approach works as follows: Given two surface patches S_1 and S_2, compute the their harmonic images HS_1, HS_2, respectively. Compute the Harmonic Shape Image HSI_1 of HS_1 and fix it. Then rotate HS_2 starting at 0 degree incrementally increase the angle until the angle reaches 2π. At each rotation angle, compute the Harmonic Shape Image HSI_2^r from HS_2^r and compute the normalized correlation coefficient $R(\theta)$ between HSI_1 and HSI_2^r. The maximum $R(\theta)$ is the correlation coefficient between S_1 and S_2.

Figure 3.1 shows an example of matching Harmonic Shape Images. The surface patch shown in Figure 3.1(b) is rotated about the z-axis for 45 degrees and translated along the x-axis and z-axis for 10 and 20 units, respectively. Both the original mesh S_1 and the transformed mesh S_2 are shown in Figure 3.1(a). The Harmonic Shape Image of S_1, HSI_1, is computed and shown in Figure

3.1(b). The harmonic image of S_2, HI_2, is computed(Figure 3.1(c)). The comparison process starts by rotating HI_2 at 0 degree and computing the Harmonic Shape Image $HSI_2^{r=0}$(Figure 3.1(d)). The correlation coefficient between HSI_1 and $HSI_2^{r=0}$ is computed and stored. The comparison process repeats rotating HI_2, computing HSI_2^r and correlating HSI_1 and HSI_2^r until r is equal to 2π. The correlation curve as a function of the rotation angle is shown in Figure 3.2. As expected, the maximum correlation coefficient is 1.0 because S_1 and S_2 are essentially the same surface patch and they have the same Harmonic Shape Image. This result also justifies one of the invariance properties of Harmonic Shape Images -- they are invariant to the pose of the underlying surface patch in 3D space.

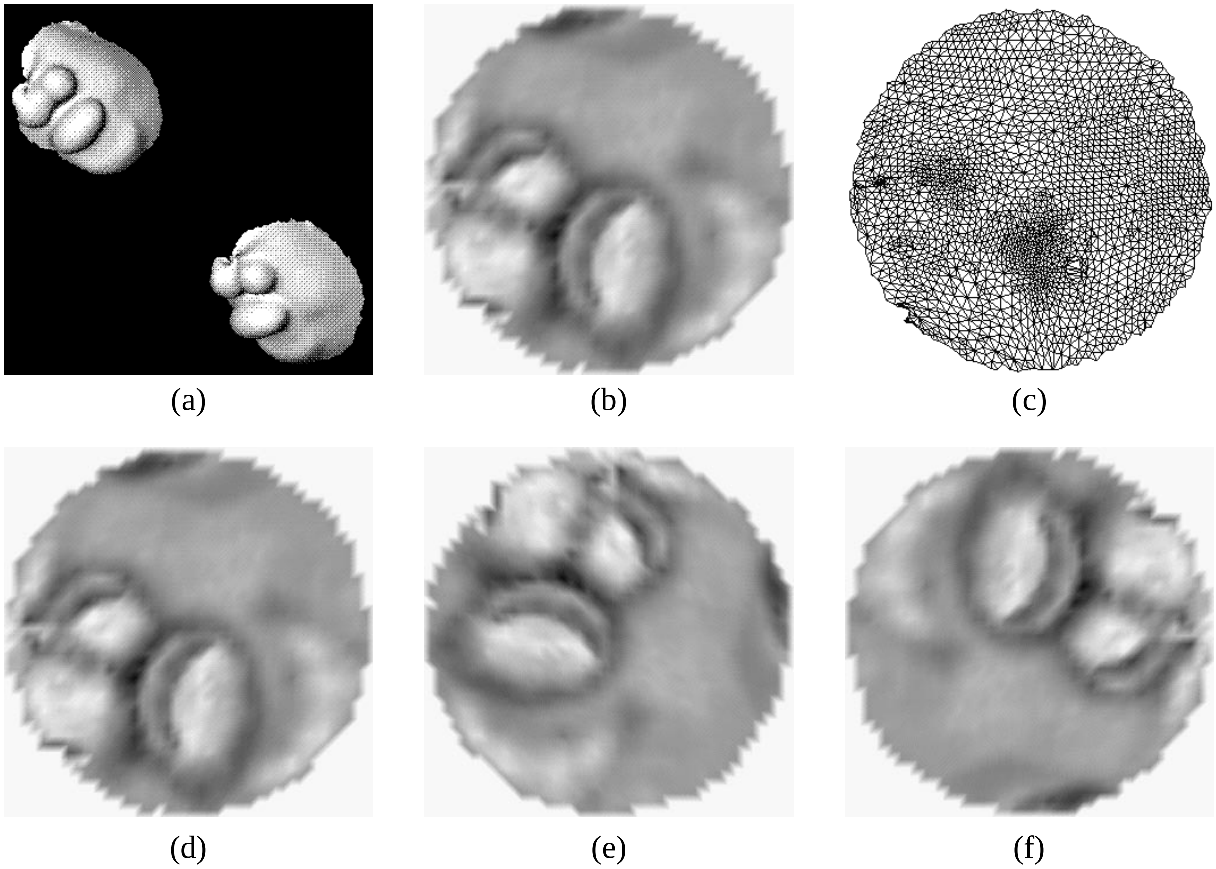


Figure 3.1: Example of comparing Harmonic Shape Images. (a) Two surface patches of the same shape but having different positions in 3D space; (b) (c) their Harmonic Shape Images.

As discussed in Chapter 2, surface continuity is preserved in Harmonic Shape Images. This property makes the correspondence problem trivial which means that no extra computation is needed in order to establish the correspondences between the two surface patches after their Harmonic Shape Images are matched. The reason is as follows. The correlation-based image matching establishes the correspondences between the two Harmonic Shape Images naturally. Because each vertex on the Harmonic Shape Image has its correspondence on the original surface patch, the correspondences between the two Harmonic Shape Images are also the correspondences

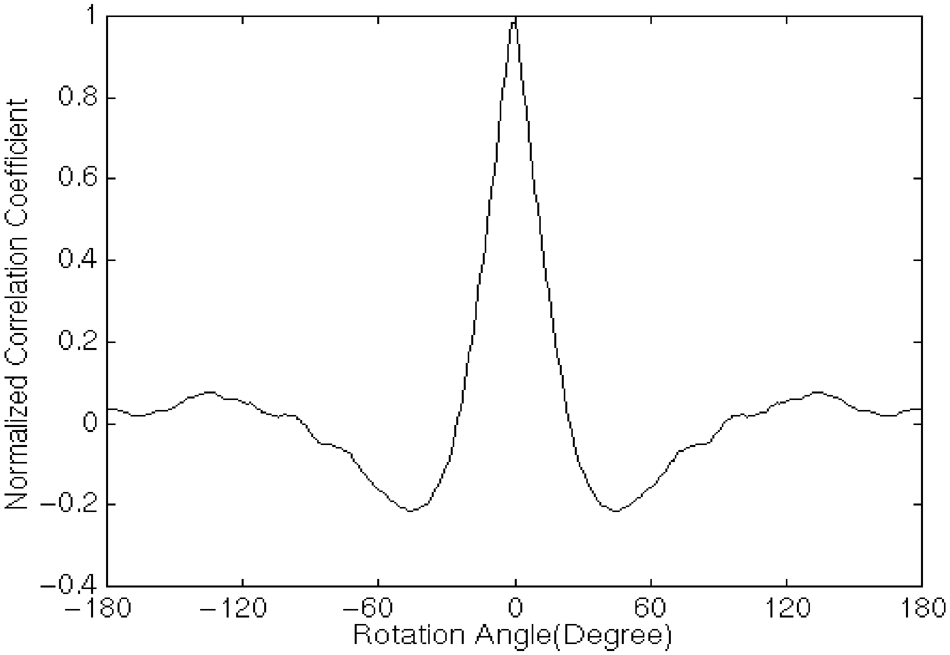


Figure 3.2: The correlation coefficient between two Harmonic Shape Images as a function of the rotation angle.

between the vertices on the two surface patches. For the above example, some of the correspondences between the two surface patches are shown in Figure 3.3. Using those correspondences, the transformation between the two surface patches can be computed[29]. The rotation angles for the above example is found to be

$$\alpha = 0.0003, \beta = 0.0002, \gamma = 90.0007 \tag{3.3}$$

The translation vector is found to be

$$trans = [10.00005699, 0.00023227, 19.99974986] \tag{3.4}$$

A comparison with the true values $\alpha = 0$, $\beta = 0$, $\gamma = 90.0$ and T=[10.0 0.0 20.0] shows that the result is accurate.

3.2 Shape Similarity Measure

The comparison of two Harmonic Shape Images has been discused in the previous section and a comparison example has been shown. Suppose that we have another surface patch S_3 shown in Figure 3.4(a) and we compare it to S_1 in the above example, we get the correlation coefficient $R(HSI_1, HSI_3)$ = 0.177536. Now we ask which surface patch, S_2 or S_3, is more similar to S_1 in shape. In this case, since $R(HSI_1, HSI_2)$ = 1.0, which is much greater than $R(HSI_1, HSI_3)$, the answer is easy. However, if we have a large number of surface patches and we need to find out which one is most similar to a specified surface patch, using a simple threshold would not be an

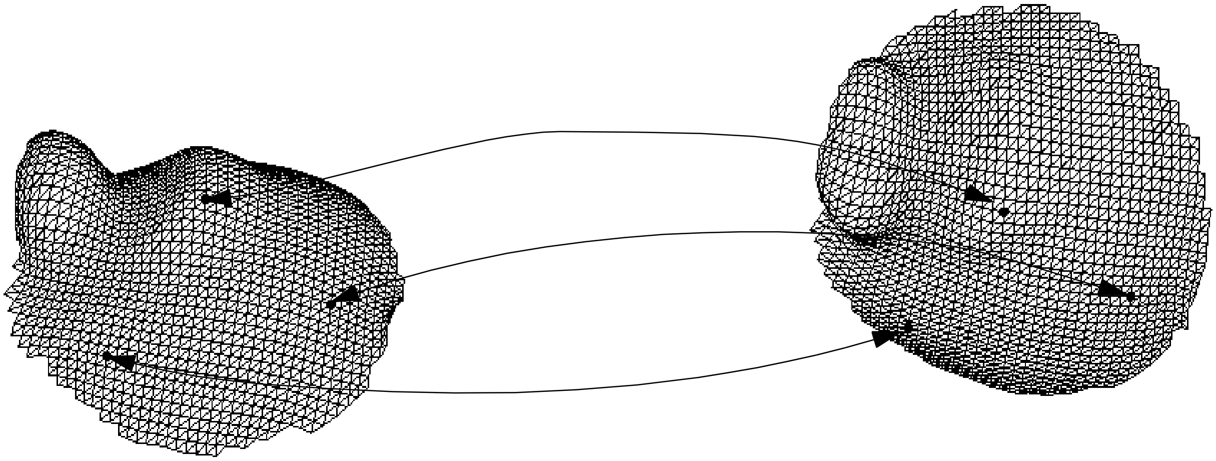
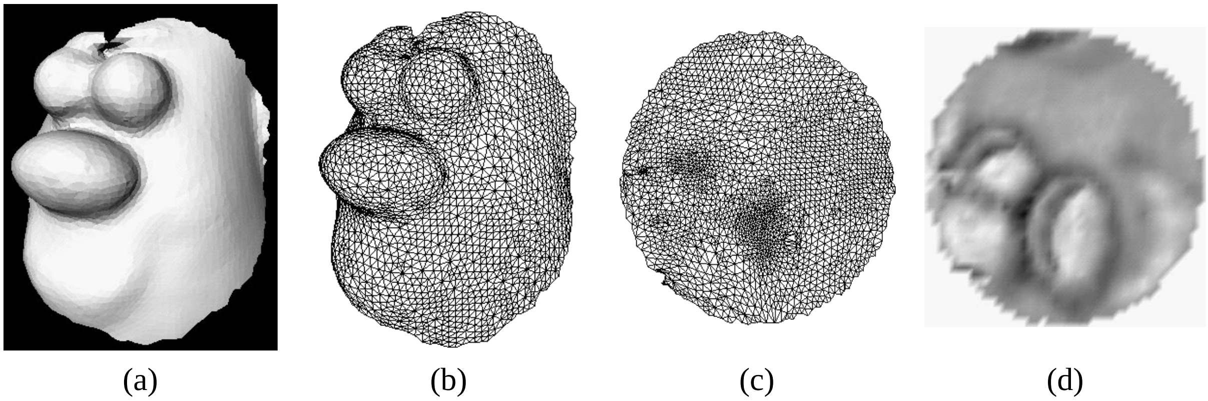


Figure 3.3: Correspondences between the resampled meshes.

adequate answer. In this case, we would need a more sophisticated measurement in order to determine how similar the most promising surface patch is to the specified patch. This measurement is called shape similarity measure. It is defined using the normalized correlation coefficient *R* between two Harmonic Shape Images as follows[44]:



(a) (b) (c) (d)

Figure 3.4: An example surface patch. (a) Rendered version; (b) wireframe version; (c) harmonic image; (d) Harmonic Shape Image.

$$C(S_1, S_2) = \sum \ln \frac{1 + R(HSI_1, HSI_2)}{1 - R(HSI_1, HSI_2)} \sum \qquad (3.5)$$

The above shape similarity measure is a heuristic loss function that will return a high value for two highly correlated Harmonic Shape Images. The change of variables, a standard statistical technique[22] performed by the hyperbolic arctangent function on the right hand side of (3.5), transforms the correlation coefficient into a distribution that has better statistical properties, namely, the variance of the distribution is independent of *R*. In this case, the variance of the transformed correlation coefficient becomes *1/(N-3),* in which *N* is the number of correspondences in the two Harmonic Shape Images[22].

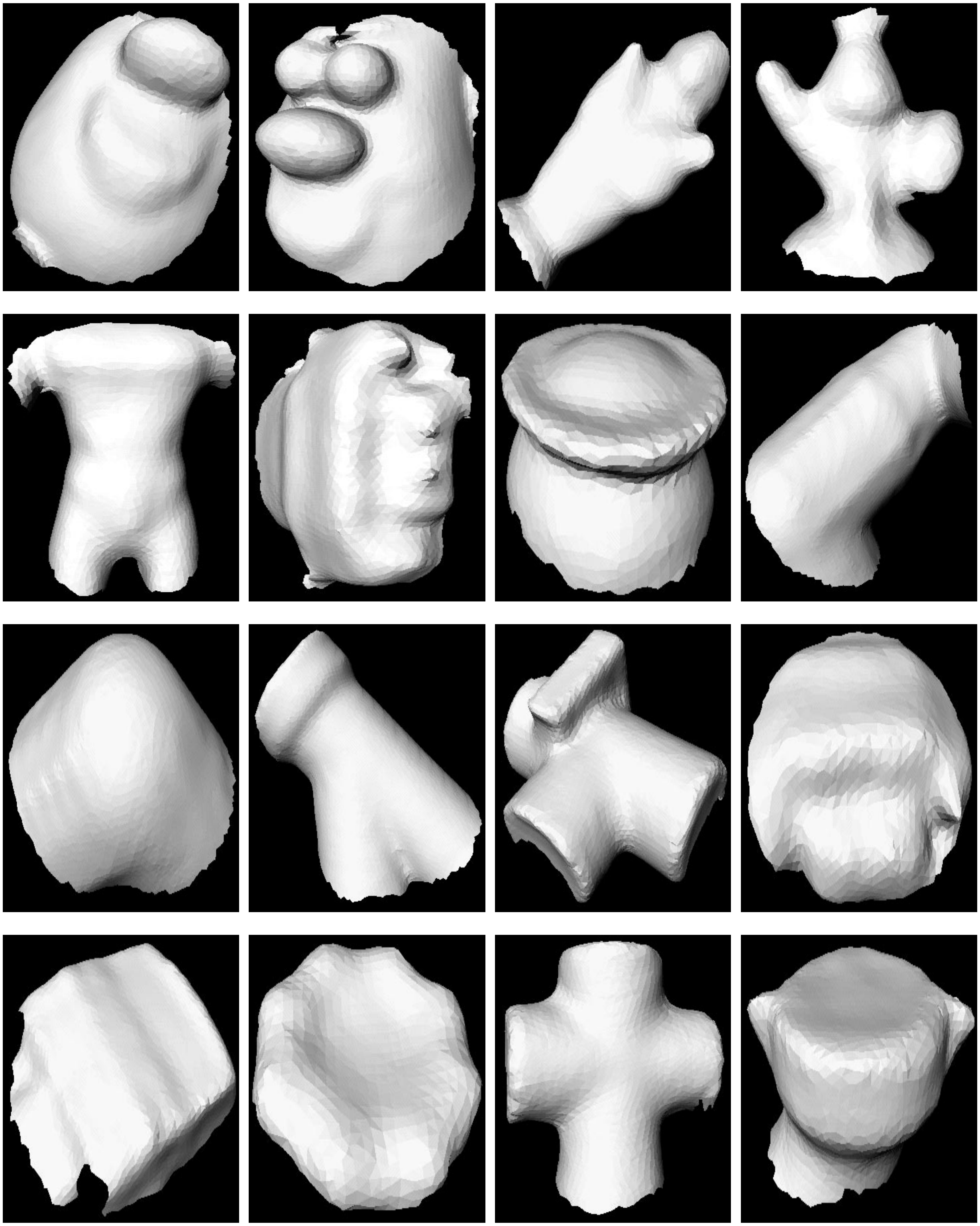


Figure 3.5: Library of surface patches.

In order to explain how to use the shape similarity measure to determine a good match between two surface patches, the following experiment is performed. Sixteen surface patches are created from real objects and stored in a library (Figure 3.5). The patches are numbered from 1 to 16 from top to bottom and from left to right in Figure 3.5. The first one is selected to be the target patch, and all the sixteen patches are compared with it by comparing their Harmonic Shape Images to that of the first patch. The values of shape similarity measure are listed in Table 3.1. The histogram of those values are shown in Figure 3.6

Table 3.1: Shape Similarity Measure

1	2	3	4
3.8002	0.2142	0.1463	0.1223
5	6	7	8
0.1515	0.1899	0.2062	0.1629
9	10	11	12
0.3252	0.2045	0.1873	0.2320
13	14	15	16
0.1199	0.3033	0.1592	0.1893

Figure 3.6 shows that the shape similarity measure of the first patch to itself is much greater than that of the other patches. Obviously, it is the best match. However, this does not tell us how good this match is or how well it is distinguished from the other matches. Therefore, instead of using a simple threshold, a statistical method[22] is used to automatically detect the best match and determine how good that match is. According to this method, good matches correspond to the outliers of a given histogram. The outliers are defined as follows. Suppose there are *N* values in the histogram. At the first step, the values are sorted and arranged from the lowest to the highest. At the second step, the median of the largest *N/2* values is found and recorded as the upper fourth. The median of the smallest *N/2* values is found and recorded as the lower fourth. At the third step, the fourth spread, f_s, is computed as the difference between the upper fourth and the lower fourth. At the fourth step, the values that are greater than $1.5f_s$ from the upper fourth are determined to be moderate outliers and the values that are greater than $3f_s$ from the upper fourth are determined to be extreme outliers. In Figure 3.6, the cut-off value for extreme outliers is indicated by the line segment. Because the shape similarity value of patch one to itself is above that cut-off, it is considered to be well distinguished from other values. This means that the shape of surface patch one is well distinguished from the shape of any other patch in the library.

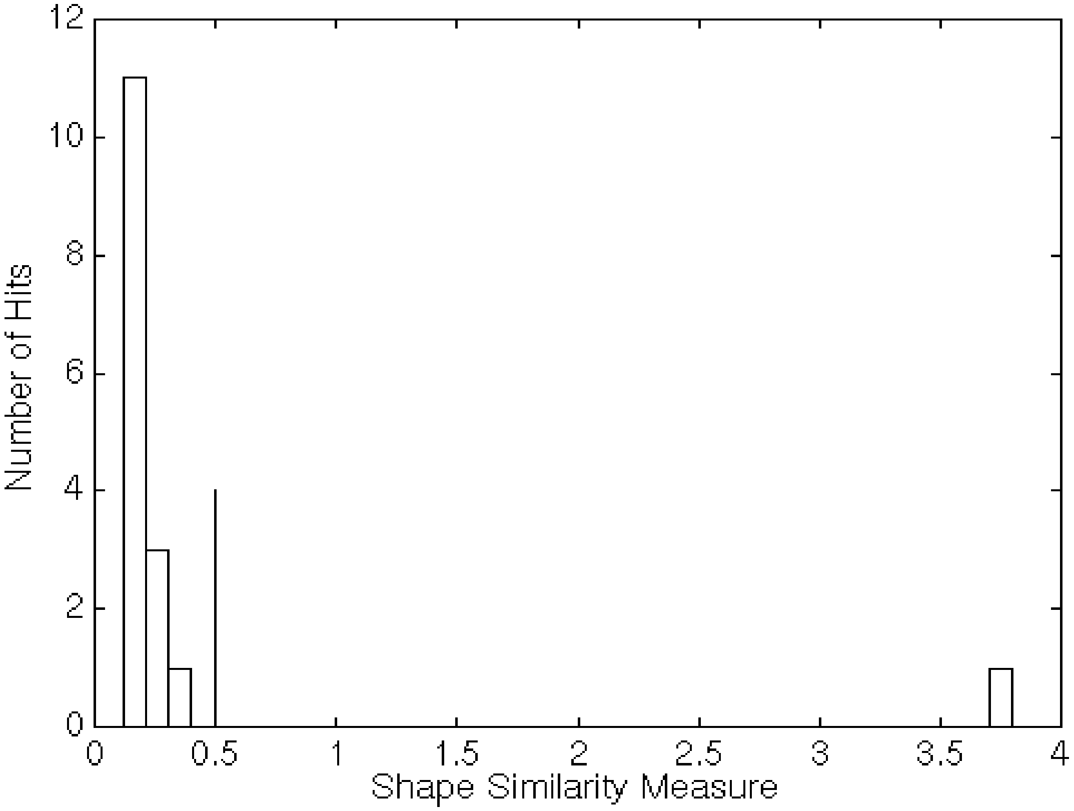


Figure 3.6: Histogram of the shape similarity measure.

3.3 Matching Surfaces by Matching Harmonic Shape Images

In the previous sections, we discussed how to match Harmonic Shape Images, how to define shape similarity measure and how to automatically determine a good match based on the histogram of the shape similarity measures. Remember that Harmonic Shape Images are only partial representations of given surfaces because they are based on surface patches; matching Harmonic Shape Images is only an intermediate step in matching surfaces. In this section, we will discuss how to match two surfaces by matching Harmonic Shape Images.

Given two surfaces, the matching process is illustrated in Figure 3.7. At the first step, a surface patch $D_2(v, R)$ on surface S_2 is randomly selected and its Harmonic Shape Image $HSI(D_2)$ is computed. At the second step, surface patches are created on S_1 using some randomly selected vertices as centers and the specified radius R. The Harmonic Shape Images of those surface patches are computed and stored in a stack. At the third step, a search is conducted by matching $HSI(D_2)$ to every Harmonic Shape Image in the stack of S_1. If a match is found, then at step four, the corre-

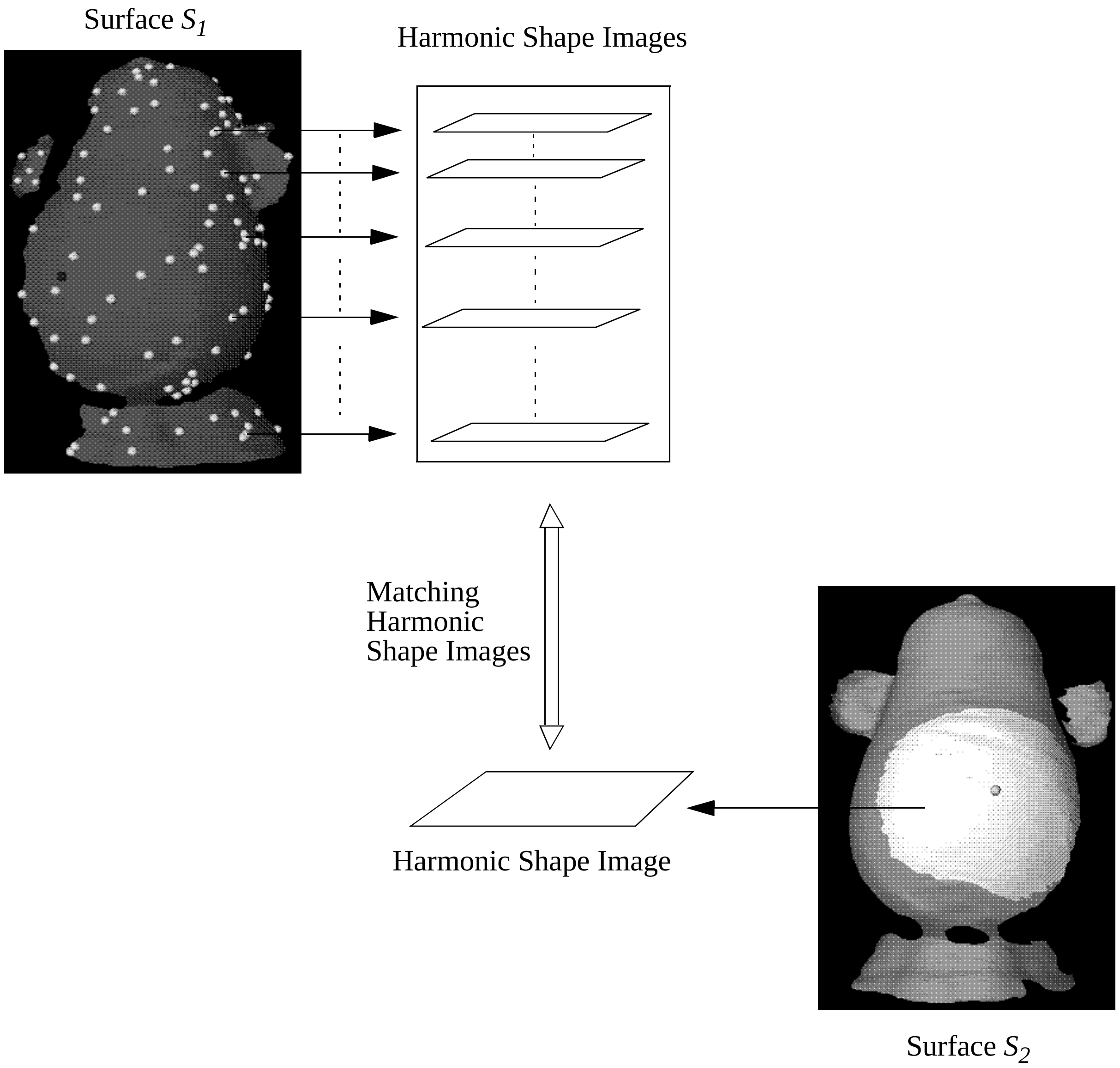


Figure 3.7: Illustration of surface matching by matching Harmonic Shape Images.

spondences between the two matched surface patches are established and then used to compute the transformation between the two surface patches. At step five, the two surfaces are transformed to the same coordinate system using the transformation found in step four. A consistency check is performed to determine whether the transformation is correct.

It needs to be emphasized that using Harmonic Shape Images, matching two surfaces is performed by matching surface patches. Those patches are regions of the two surfaces instead of a small neighborhood. All the vertices in those regions participate in the matching process explicitly. This means that all the vertices in one region can naturally find their correspondences in the other region after the matching. This matching process is different from those approaches using point-based representations which only match one pair of points on the two surfaces.

Two issues that need to be discussed about the above surface matching strategy are choosing the radius of surface patches and using a hierarchical approach to speed up the matching process.

Choosing the Radius of Surface Patches

The two main applications of surface matching are surface registration and object recognition. The strategies for choosing the radius of surface patches are different in the two cases. In the case of surface registration, in general, there are two surfaces, S_1 and S_2, to be registered at one time. S_1 and S_2 partially overlap. The area of S_1 is denoted as A_1 and the area of S_2 is denoted A_2. Assume $A_1 \le A_2$. It is assumed that the area of the overlapping part is at least 40% of the A_1. The area of the surface patches used for matching is at least 25% of A_1. If a 2D disc model is used to approximate the surface patches, then the radius can be estimated by $R = \sqrt{0.25A_1 / \pi}$.

In the case of object recognition, model surface patches are pre-created and stored in the model library. During run-time, scene surface patches are created and compared with the model surface patches. Because all the models are known before recognition and the scene surfaces are partial surfaces of the models, the optimal radius of patches for matching may be determined before recognition.

Given an object surface as a model, some observations can be made for choosing the radius of surface patches on the model. If the radius is too small, then the surface patch will be small and it will not have enough features to distinguish it from other patches on the model. In other words, if a small surface patch with few features is selected, then there may be so many qualified matches on the model that it is impossible to decide which one is the best match according to the statistical method we have discussed in the previous section. On the other hand, if the radius is too large, then there may be no matched patches at all on the model other than the patch itself. This is not good for recognition because the sampling on the scene surfaces is not the same as that on the model surfaces. As a result, there will not be any matched patches on the scene surfaces. The following example illustrates how to choose the appropriate radius for surface patches on a model surface.

Figure 3.8 shows an object with closed surface. Here, a closed surface is a triangular mesh in which every edge is associated with two triangles. A vertex, v_c, is selected as the central vertex to generate a series of target surface patches, $D_t^i(v_c, R_i)$, $i=1...6$. The radius, R_i, of this surface gradually increases so that the area of the corresponding surface patch increases from 5% to 30% of the entire area of the closed surface. Table 3.2 lists the radius with respect to each area percentage.

Table 3.2: Radius of Target Surface Patch with Respect to Area Percentage

Area Percentage	5%	10%	15%	20%	25%	30%
Radius	2.5	3.8	4.8	5.8	7.0	8.0

For each radius R_i, a library of model surface patches is created according to the following: Every vertex v_j, $j= 1...n$, is selected as the central vertex and a surface patch with radius R_i, $D_m^i(v_j, R_i)$, is created. Then the target model patch, $D_t^i(v_c, R_i)$, is compared with each patch in the model library

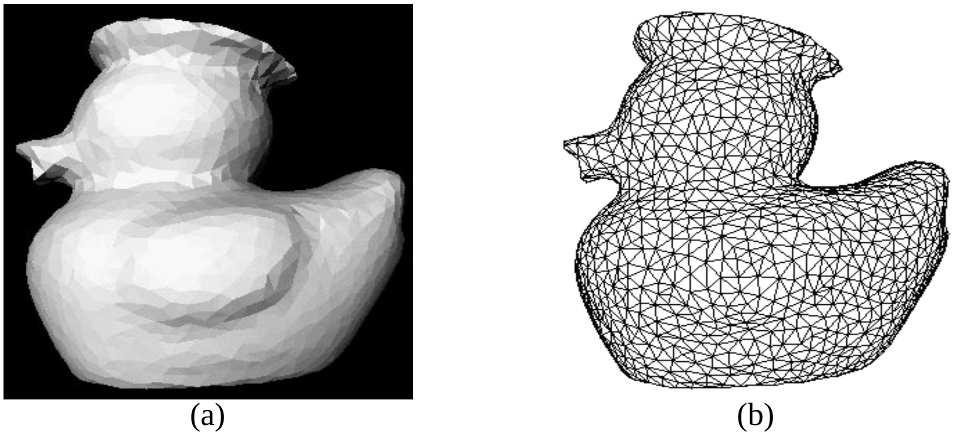


Figure 3.8: An object with closed surface. (a) Rendered version; (b) Wireframe version.

by comparing their Harmonic Shape Images. The shape similarity measure is computed for each pair comparison and the histogram of all the shape similarity values is drawn. The statistic method that has been discussed in Section 3.2 is then used to find the qualified matches of the target surface patch among the model patches in the library. In the following, we present the comparison result for each radius R_i.

Figure 3.9(a) and (b) show the target surface patch $D_t^1(v_c, R_1)$ with $R_1=2.5$. Figure 3.9(c) shows its Harmonic Shape Image. The histogram of the shape similarity values for the pairwise patch comparison is shown in Figure 3.9(d). Here, it should be mentioned that a perfect match is obtained when the target patch $D_t^1(v_c, R_1)$ is compared to the model patch $D_m^1(v_c, R_1)$ because the two patches are the same. The similarity value for this comparison is not defined according to (3.5). Therefore, the histogram in Figure 3.9(d) does not include this value. This is the case for the remaining histograms which will be presented later in this section.

The vertical line segment in Figure 3.9(d) indicates the cut-off value for the qualified matches. There are no qualified matches in this case because the target surface patch does not have enough features to distinguish itself from the model surface patches in the library. In fact, the shape similarity values of the top 50 matches in the library are not high enough to qualify for the automatic selection of the statistical method. Figure 3.9(e) and (f) show the central vertices of those 50 top matches. It can seen that the central vertices are at different locations of the model surface instead of concentrating at the central vertex, v_c, of the target surface patch. This result shows that the matching of the target surface patch is not successful.

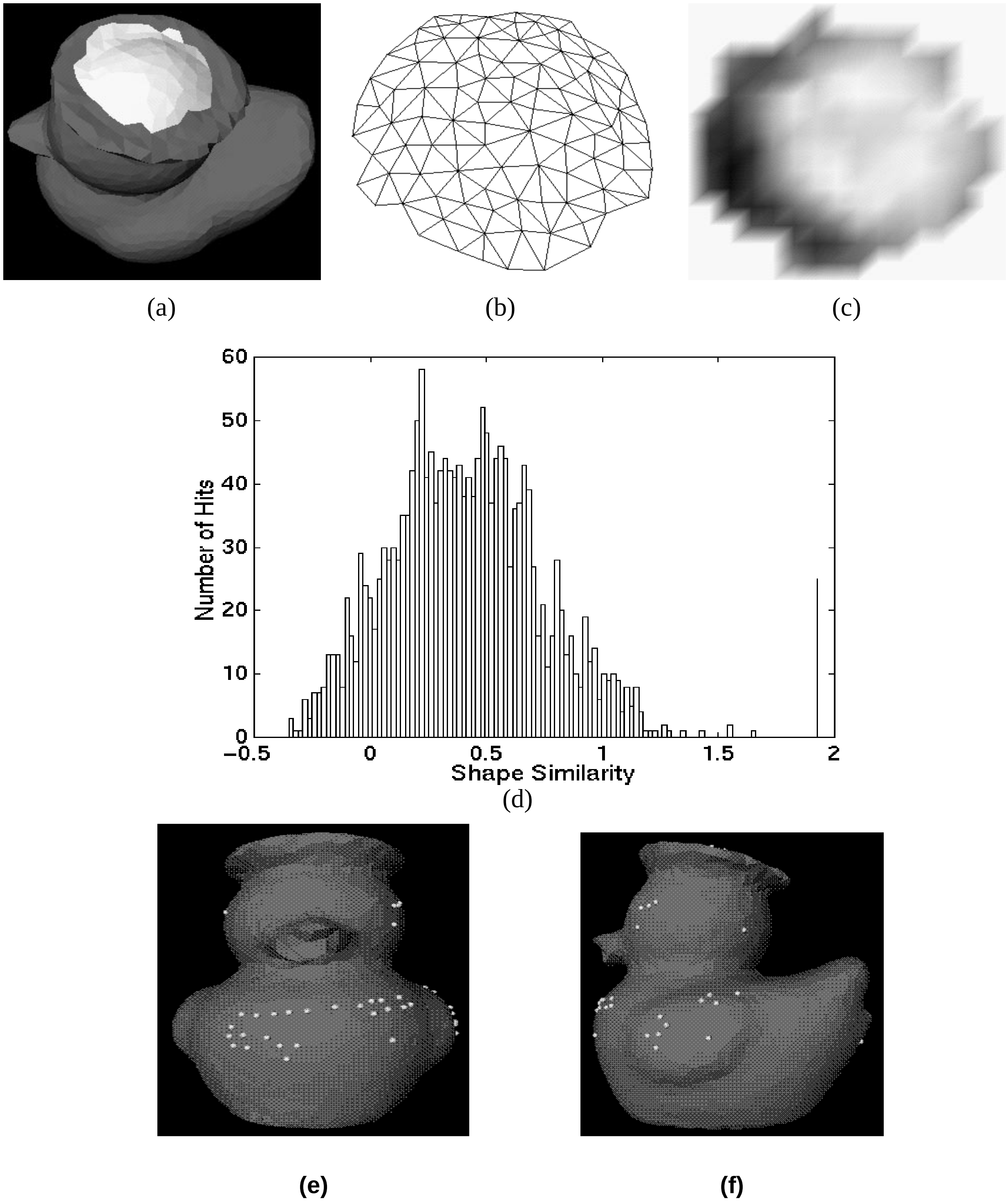


Figure 3.9: Target surface patch with radius *R_i = 2.5*. (a) Rendered version; (b) wireframe version; (c) Harmonic Shape Image; (d) the histogram of the shape similarity values for the pairwise patch comparison between the target surface patch and the model patches; (e), (f) central vertices of the top 50 model matches in the library.

Now let us look at the case in which $R_2 = 3.8$. Figure 3.10(a), (b) and (c) show the target surface patch $D_t^2(v_c, R_2)$ and its Harmonic Shape Image, respectively. After comparing the target surface patch to the model patches in the library, the resultant histogram of the shape similarity values is shown in Figure 3.10(d). It can be seen that quite a few qualified matches are identified. In fact, the number of qualified matches is 50. Figure 3.10(e) shows the central vertices of those 50 surface patches. They are all around the central vertex, v_c, of the target surface patch. This fact shows that the target surface patch has been well identified among the model patches in the library.

Compared with previous case in which $R_i = 2.5$, the target surface patch with $R_i = 3.8$ has more features that can be seen from its Harmonic Shape Image. This definitely helps identify similar model patches in the library. However, considering the number 50, there seem to be too many qualified matches. In the following, we will continue increasing the radius of the target surface patch to determine whether the number of qualified matches decreases. This will demonstrate how the radius of surface patches affects the matching result.

Figure 3.11(a), (b) and (c) show the target surface patch $D_t^3(v_c, R_3)$ with $R_3 = 4.8$ and its Harmonic Shape Image. The histogram of matching results is shown in Figure 3.11(d). The central vertices of the qualified patches are shown in Figure 3.11(e). In this case, the number of qualified model matches is still 50 and the central vertices of those surface patches are around the central vertex, v_c, of the target surface patch as well. Therefore, increasing the radius from 3.8 to 4.8 does not help to better identify the best matches in the model library because no new features are added to the target surface patch with this increase.

In the cases in which $R_4 = 5.8$, $R_5 = 7.0$ and $R_6 = 8.0$, the corresponding target surface patches, their Harmonic Shape Images and the histograms of the pairwise patch comparison results are shown in Figure 3.12. The numbers of qualified matches in those three cases are 41, 18 and 3, respectively. Therefore, increasing the radius of the target surface patch does help identify it from the model patch library. However, it may not be realistic to use surface patches with large radii in the application of object recognition because the full surface patches may not be seen from restricted viewing angles even without being occluded by other objects in the scene. For example, if the viewing direction is as shown in Figure 3.12(a1), (b1) and (c1), then the target surface patches with radii greater than 4.8 will be seen only partially. In fact, the resultant patches will have too much occlusion to be usable in recognition.

From the above example and discussion, it can be seen that the best value of the radius in order to maintain both discrimination and stability is 4.8. In practice, a general guideline can be made about selecting the radius of a surface patch in the application of object recognition. The lower bound of the radius of a surface patch can be determined by conducting the above experiments. The upper bound can be determined using the viewing directions.

Hierarchical Matching Strategy

A hierarchical strategy can be used to speed up the matching process. Earlier in this section, we discussed a direct search approach to finding the best match on the second surface when given a surface patch on the first surface. Due to the stability of Harmonic Shape Images (this will be discussed in detail in the next chapter) meaning that neighboring vertices have similar surface patches, the search can be conducted among part of the vertices on the second patch.

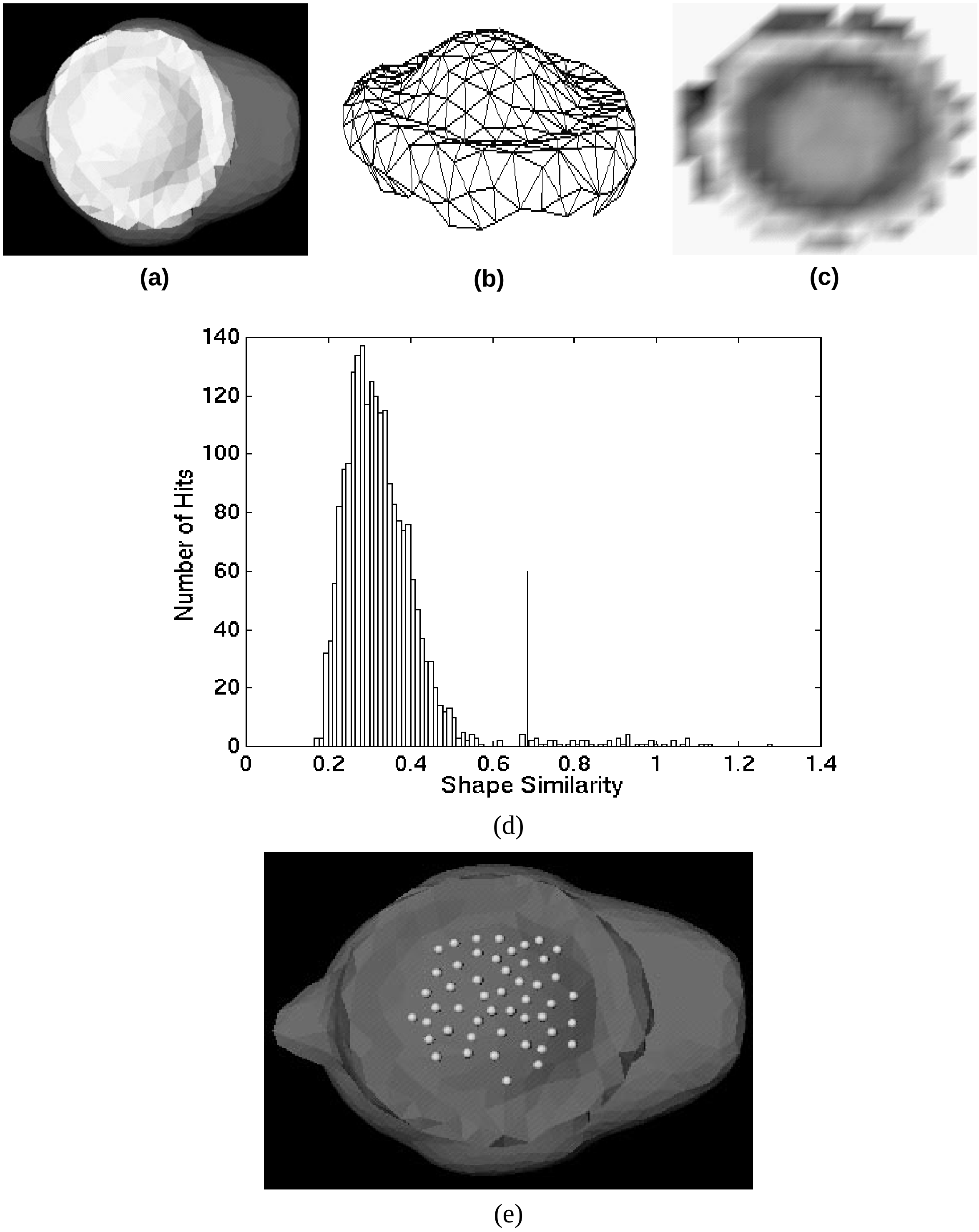
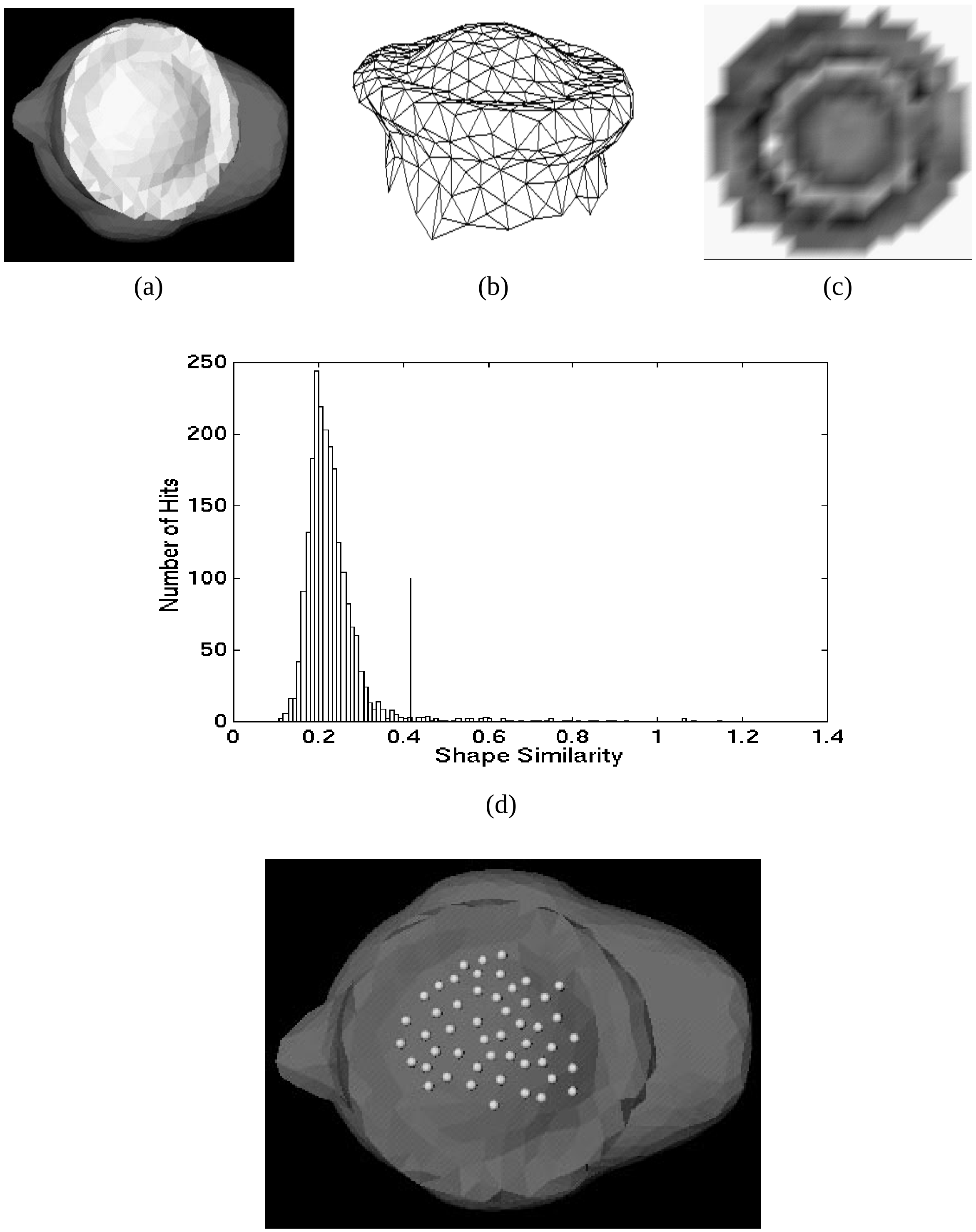


Figure 3.10: The target surface patch with *R_i = 3.8*. (a) Rendered version; (b) wireframe version; (c) Harmonic Shape Image; (d) histogram for the matching results; (e) the central vertices of the matched surface patches.



(a)

(b)

(c)

(d)

(e)

Figure 3.11: The target surface patch with *R_i = 4.8*. (a) Rendered version; (b) wireframe version; (c) Harmonic Shape Image; (d) histogram for the matching results; (e) the central vertices of the matched surface patches.

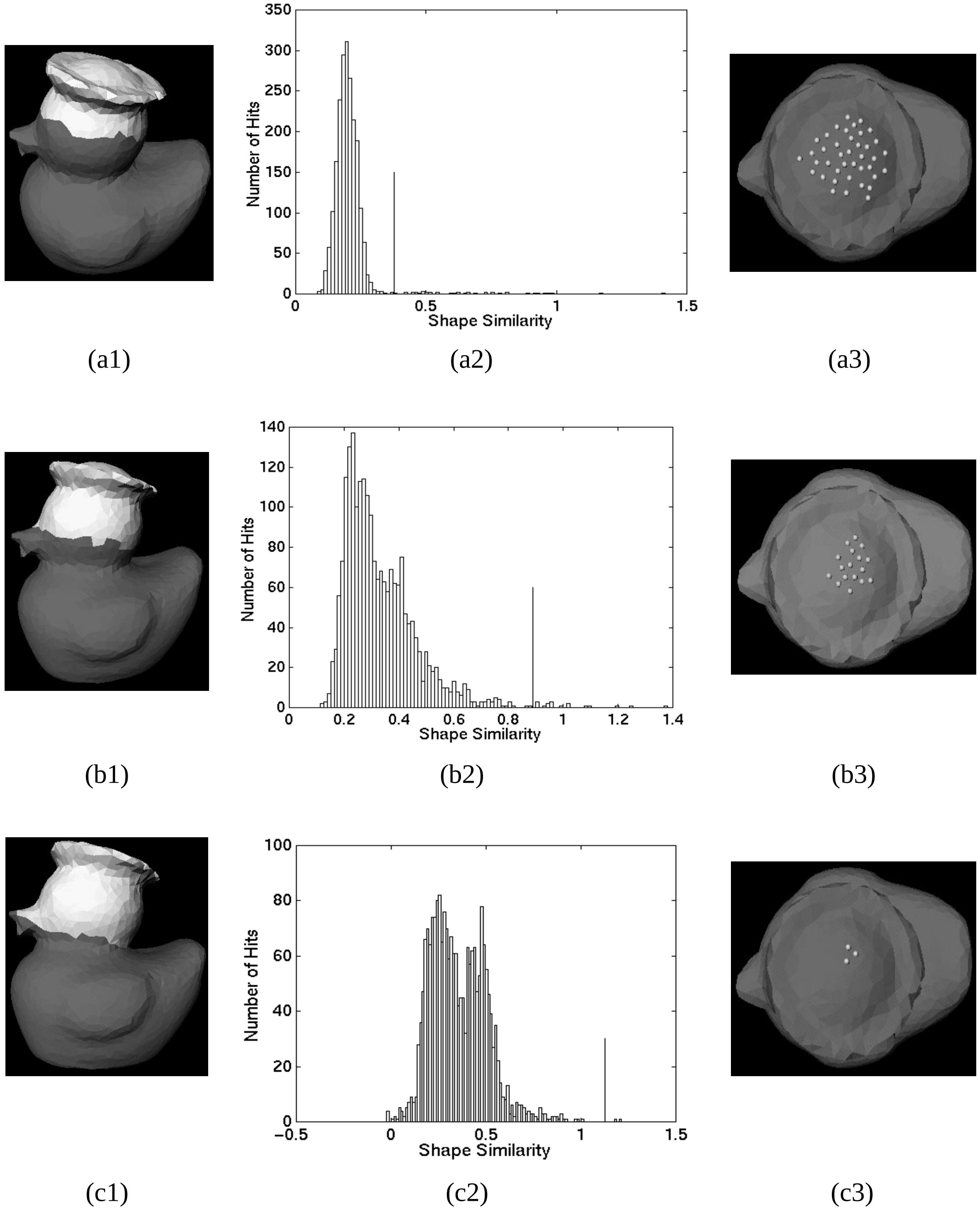
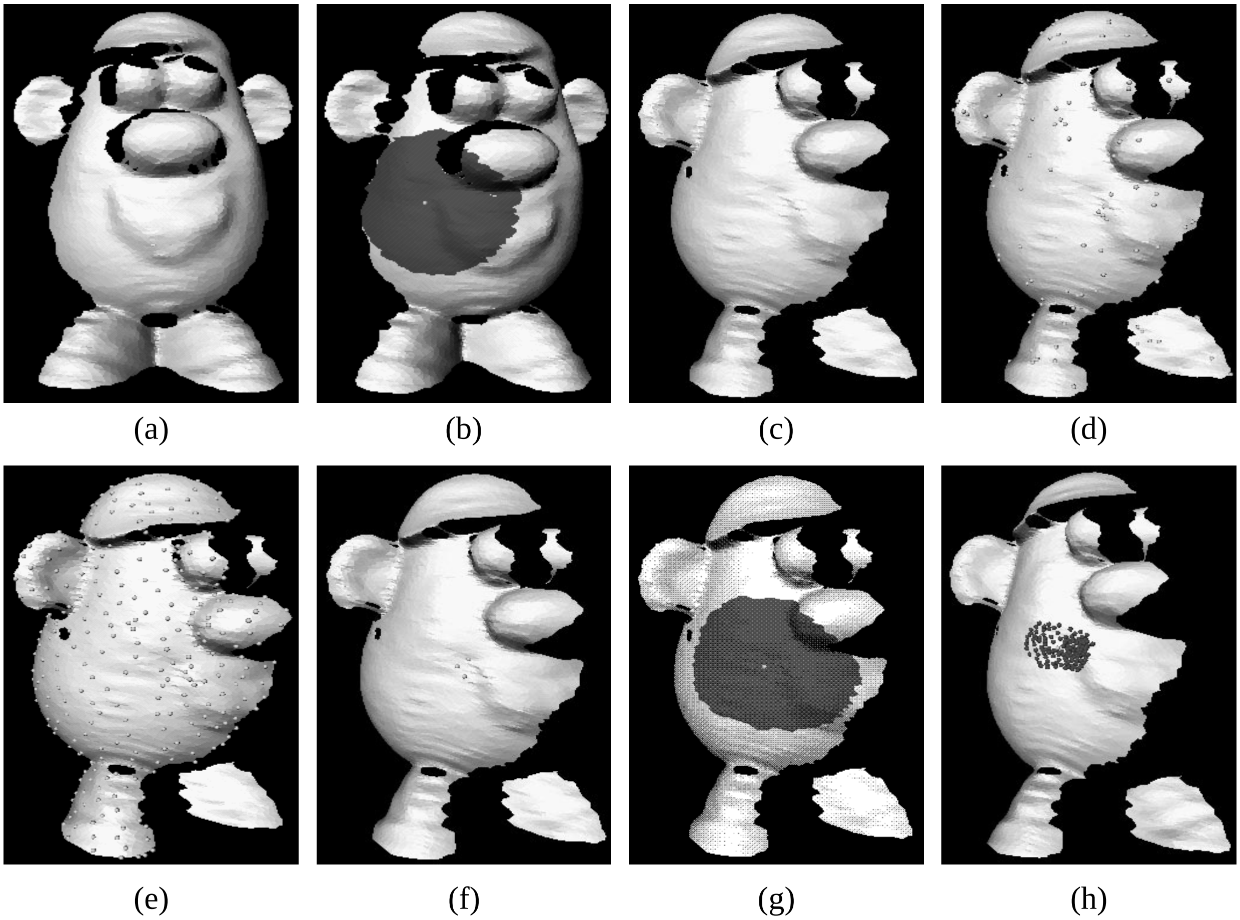


Figure 3.12: Matching results for the target surface patches with different values of *Ri.* (a) 5.8; (b) 7.0; (c) 8.0.

This is referred to as the coarse-level search. When the best match is found at the coarse level, it is further refined in the neighborhood of the best match at a finer resolution. This process is repeated until the best match is found on the second surface. The hierarchical approach is illustrated using the following example.

Figure 3.13(a) and (c) show two surfaces S_1 and S_2 to be registered. The target surface patch $D_t^{S1}(v, R)$ is overlaid on S_1 and shown in Figure 3.13(b) with its central vertex *v* highlightened. In order to apply the hierarchical search strategy, *m* vertices are randomly selected on S_2 as central vertices to generate *m* surface patches that will be compared with the target patch $D_t^{S1}(v, R)$. In this case, *m* = 100. The way to randomly select the central vertices is to use the pseudo random number generator to generate *m* integers that are in the range of [0, $n(S_2)$] with *n* being the number of vertices on S_2. Those *m* central vertices are displayed in Figure 3.13(d). Ideally, the *m* vertices will distribute uniformly on S_2 so that they construct a coarse resampling of S_2. It can be seen from Figure 3.13(d) that they are not distributed uniformly as expected. The reason for this is that the indices of the vertices are not necessarily related to their positions in the 3D space.



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 3.13: Illustration of the coarse-to-fine surface matching strategy.

In order to overcome the above resampling problem, a control strategy should be used when selecting the central vertices so that they distribute as uniformly as possible on the surface. Before explaining how the control strategy works, let us define the *n*th-step neighbor of a vertex *v* on a triangular mesh.

- The *1st*-step neighbors of *v*, $u \in S_1(v)$, are the vertices that are connected to *v* by edges
- The *2nd*-step neighbors of *v*, $w \in S_2(v)$, are the vertices that satisfy $w \in S_1(u), u \in S_1(v) \wedge w \notin S_1(v)$
- The *nth*-step neighbors of *v*, $w \in S_n(v)$, are the vertices that satisfy $w \in S_1(u), u \in S_{n-1}(v) \wedge w \notin S_{n-1}(v)$

The *nth*-step neighbors of *v* can be interpreted as the *nth*-layer of vertices that are around *v*. The control strategy works as follows. At the first step, the first vertex is selected randomly. Then at the second step, a user-specified separation distance *d*, defined by the order of *nth*-step neighbors, is used to determine which vertices are selected next. The rule is that the neighboring vertices of *v* lower than the order of *d* are not selected. And the neighboring vertices of *v* with order *d* are selected. At the third step, with each of the newly selected vertices, repeat the second step. The process stops when there are no more vertices to be selected. The pseudo code for implementing the control strategy is the following.

```
// SelectedVerts: the array that stores the selected vertices
// ToBeAddedVerts: the array that stores the vertices to be
// added to SelectedVerts
// Processed: a vertex is either labelled unselected or selected
// or in ToBeAddedVerts
void SelectSamplingVers(int StartVert, int SeparationDistance,
IntArray SelectedVerts)
{
      ToBeAddedVerts.push_back(StartVert);
      while ( ToBeAddedVerts is not empty )
      {
            CurrentVert = pop_front(ToBeAddedVerts);
            SelectedVerts.push_back(CurrentVert);
            GetNeighborsOfOrderD(CurrentVert, d, ToBeAdded);
      }
}
void GetNeighborsOfOrderD(int CurrentVert, int SeparationDis-
tance, IntArray ToBeAdded)
{
      IntArray Queue;
      int i, i1, i2, i3;

      i1 = 0; // head pointer to the current layer
      i2 = 1; // tail pointer to the current layer
      i3 = 1; // moving pointer for adding new vertices

      Queue.push_back(CurrentVert);
```

```
    for ( i = 0; i < d; i ++ )
    {
        while ( i1 != i2 )
        {
            GetFirstStepNeighbor(Queue[i1]);
            if ( !Processed(FirstStepNeighbor(Queue[i]) )
            {
                Label FirstStepNeighbor(Queue[i]) unselected;
                Queue.push_back(FirstStepNeighbor(Queue[i]));
                Move Pointer i3;
            }
            i1 ++;
        }
        i2 = i3; // update tail pointer for the next layer
    }
    // Store the dth-step vertices into ToBeAddedVerts
    for ( i = i1; i < i2; i ++ )
    {
        ToBeAddedVerts.push_back(Queue[i]);
        Label Queue[i] selected;
    }
}
```

Using the above control strategy, with the separation distance set to 4.0, the 273 selected vertices on the surface in Figure 3.13(c) are shown in Figure 3.13(e). It can be seen that the distribution of the selected vertices greatly improves compared with the case shown in Figure 3.13(d).

After the coarse-level surface patches on S_2 are generated using the selected vertices, they are matched to the target surface patch on S_1 (Figure 3.13(b)). Three qualified patches are found; their central vertices of those three patches are shown in Figure 3.13(f). Figure 3.13(g) shows the best matched patch of the coarse-level matching overlaid on S_2.

The next step matching is conducted at a finer resolution. The vertices that are within the neighborhood of the central vertex of the best matched patch at the coarse level are used as the central vertices to generate surface patches for the fine-level matching. The size of the neighborhood is a user-specified parameter. In our implementation, the size of the neighborhood is set to half of the size of the target surface patch. Figure 3.13(h) shows the central vertices of the surface patches to be used for fine-level matching.

3.4 Resampling Harmonic Shape Images

As an implementation issue, it has been mentioned in Section 3.1 that a horizontal and vertical scanning of Harmonic Shape Images needs to be done in order to request them using the usual *m*-by-*n*-pixel format. This step is called resampling Harmonic Shape Images.

A unit grid is created and overlaid on the Harmonic Shape Images. Then the curvature value for each of the points on the grid is determined by interpolating the Harmonic Shape Image(Figure 3.14). Figure 3.14(a) shows a grid with equal spacing overlaid on a Harmonic Shape Image. For an arbitrary point $u(i, j)$ on the grid, its curvature value $c(i, j)$ is interpolated using

$$c(i,j) = \alpha c(v_0) + \beta c(v_1) + \gamma c(v_2) \qquad (3.6)$$

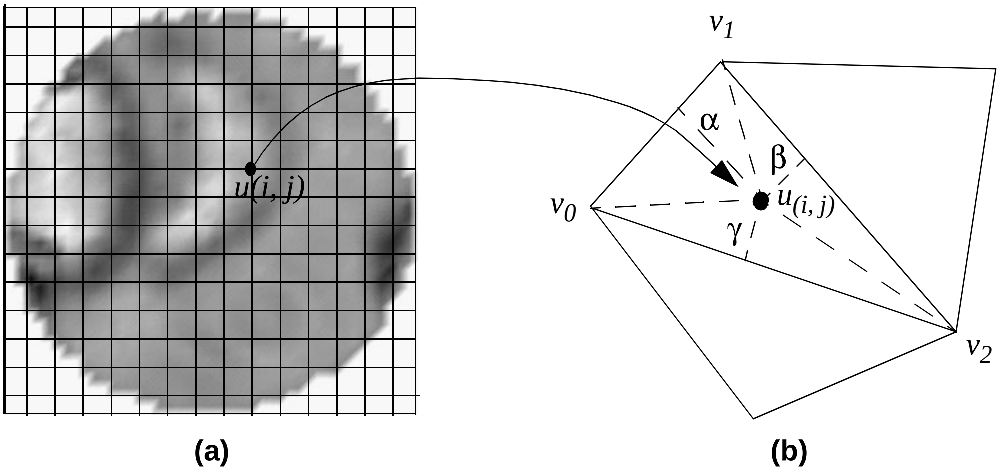


Figure 3.14: Illustration of the resampling strategy. (a) A unit grid overlaid on a Harmonic Shape Image; (b) the interpolation on the Harmonic Shape Image.

in which v_0, v_1, v_2 are the vertices of the triangle in the Harmonic Shape Image that *u(i, j)* falls in(Figure 3.14(b)). α, β and γ are the barycentric coordinates of *u(i, j)* in the triangle (v_0, v_1, v_2) and they satisfy the constraint

$$\alpha + \beta + \gamma = 1, \alpha, \beta, \gamma \geq 0 \qquad (3.7)$$

Resampling Resolution

There are a few issues that need to be discussed with respect to the resampling process. In order to not lose the shape information of the original surface, the resolution of the resampling grid should not be lower than that of the triangular mesh. Suppose that the size of the resampled Harmonic Shape Image is *NxN, N* is determined according to the following

$$N = \lceil \sqrt{n} \rceil \qquad (3.8)$$

in which n_v is the number of vertices on the triangular mesh and the function *ceil(x)* means to obtain the nearest integer which is not greater than $\sqrt{n_v}$. When there are two surface patches, with N_m and N_v vertices, respectively to be compared, N is selected to be the larger one between N_m and N_v.

Locating Resampling Points

The second issue is how to efficiently locate the resampling points on the Harmonic Shape Images. As discussed earlier in this section, the curvature value of a resampling point is interpolated using its bary-centric coordinates in the triangle in which it falls. So there is the issue of locating the triangle in which the resampling vertex falls. The linear search approach is computationally expensive. In order to speed up the point-locating process, the following divide-and-conquer algorithm is used in our implementation.

The unit grid is first divided into *mxm* bins. *m* is selected as

$$m = \frac{\sqrt{n_t}}{T} \tag{3.9}$$

in which n_t is the number of triangles in the Harmonic Shape Image and *T* is a constant that adjusts the number of bins. In our implementation, *T* is set to 10. After the bins are set up, the triangles are stored into those bins. This is done by first storing the vertices into corresponding bins. At the next step, for each triangle $T_i = (v_j, v_k, v_l)$, the bin indices for each of its vertices are denoted as $bin(v_j) = (bin_x(v_j), bin_y(v_j))$ as shown in Figure 3.15. Then the minimum and maximum of bin indices of the three vertices are found for *binx* and *biny* and are denoted as *min(binx)*, *max(binx)*, *min(biny)*, *max(biny)*. Then the bins that the triangle T_i is in can be determined as $bin(i, j) | min(binx) \le i \le max(binx), min(biny) \le j \le max(biny)$. This means that in Figure 3.15, all the bins are considered as containing the triangle T_i. Obviously, this strategy is conservative in that it takes into account some bins that do not contain T_i, e.g., the bins in Figure 3.15 labelled with question marks. Because it takes much more effort to determine the exact the number of bins that the triangle is in, the cost of taking extra bins into account is ignored.

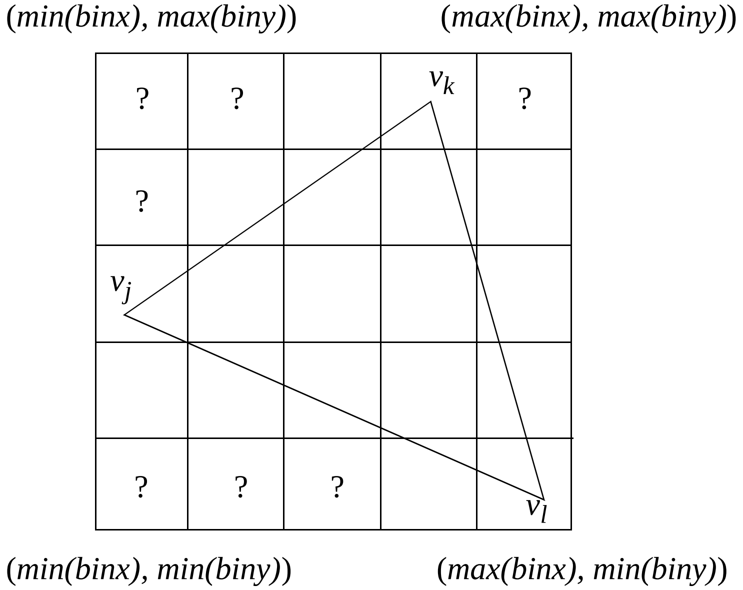


Figure 3.15: Locating resampling points using the divide and conquer strategy.

After all the triangles are stored into the bins, each resampling vertex will be located in a specific triangle using this information. First of all, it is easy to locate the bin that the resampling vertex falls in using its coordinates. Second, since there may be a number of triangles in that bin, a linear search is needed to locate the right triangle. The criterion for locating the right triangle is to compute the bary-centric coordinates for the resampling vertex in each triangle in that bin. If the bary-centric coordinates satisfy (3.7), then the resampling vertex falls into that triangle. The bary-centric coordinates can be obtained by solving the following equations. Let (x_i, y_i), i=*0, 1, 2* denote the coordinates of vertices v_i, i=*0, 1, 2,* on the Harmonic Shape Image. Let (x, y) denote the coordinates of the resampling point u. Then the following equations hold

$$\left\{\begin{aligned} \alpha x_0 + \beta x_1 + \gamma x_2 &= x \\ \alpha y_0 + \beta y_1 + \gamma y_2 &= y \\ \alpha + \beta + \gamma &= 1 \end{aligned}\right. \tag{3.10}$$

(3.10) can be written in matrix form as (3.11)

$$\begin{bmatrix} x_0 & x_1 & x_2 \\ y_0 & y_1 & y_2 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{3.11}$$

The unknown vector $[\alpha, \beta, \gamma]^T$ can be solved as follows:

$$\begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix} = \begin{bmatrix} x_0 & x_1 & x_2 \\ y_0 & y_1 & y_2 \\ 1 & 1 & 1 \end{bmatrix}^{-1} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{3.12}$$

According to the way the interior mapping is constructed, there should not be any degenerated triangles. This means that the matrix in (3.12) always has full rank. In our implementation, rank checking is still performed in case of any numerical problems caused by skinny triangles.

Invalid Resampling Points

Because a unit grid is overlaid on the Harmonic Shape Image which is defined on a unit disc, some resampling points are outside that image(Figure 3.14(a)). Those resampling points are labelled invalid so that they do not participate in the correlation-based image matching. Invalid resampling points can be caused by occlusion as well. In this case, those invalid resampling points are inside the unit circle. Their curvature values are labeled invalid on the Harmonic Shape Image.

Validity of the Resampling Grid

The resampling grid shown in Figure 3.14(a) has equal spacing among all points. This kind of resampling scheme is called parametric uniform resampling. No matter what resampling scheme is used, its goal is to resample the original surface patches as uniformly as possible so that they can be compared in a meaningful way. For example, there are three surface patches shown in Figure 3.16(a1), (b1), (c1). Their harmonic images are shown in (a2), (b2) and (c2), respectively. The

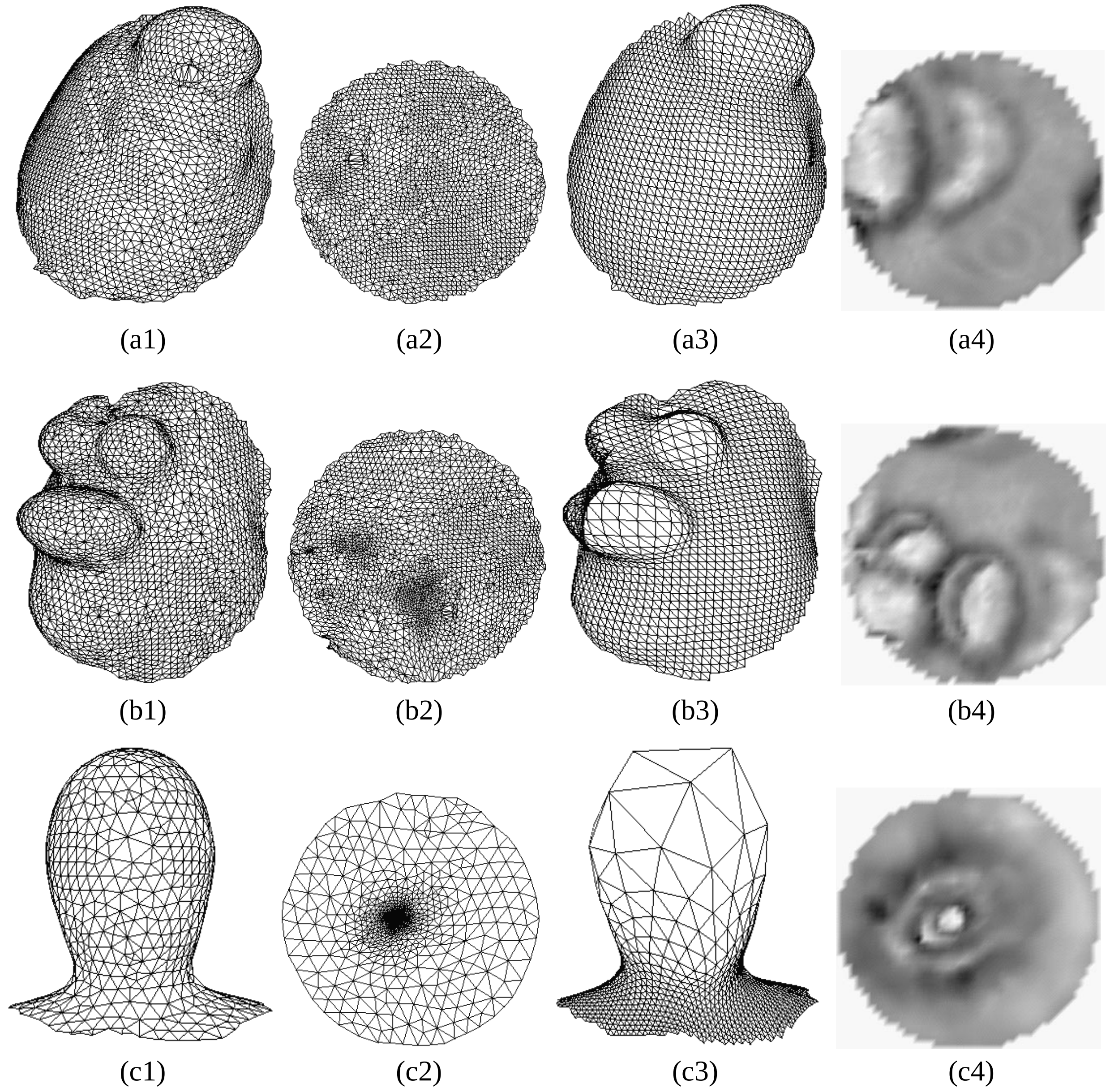


Figure 3.16: Examples of surface patches and their resampled versions. (a1), (b1, (c1) Three surface patches in wireframe; (a2), (b2), (c2) their harmonic images; (a3), (b3), (c3) the resampled meshes; (a4), (b4), (c4) their Harmonic Shape Images.

resampled meshes for those three surfaces shown in (a3), (b3) and (c3) are obtained by using the same interpolation strategy that has been used to generate Harmonic Shape Images. The only difference is that, instead of interpolating the curvature distribution function on the harmonic image, the coordinate function is interpolated.

As can be seen In Figure 3.16(a3), (b3), the reconstructed meshes are reasonably uniform. However, in Figure 3.16(b3), there is significant difference between the densities of resampling points on different parts of the reconstructed surface. In this case, the Harmonic Shape Image in Figure 3.16(b4) is not a good representation of the original surface. Therefore, comparing the surface patches in Figure 3.16(a1) and (c1) by comparing their Harmonic Shape Images in Figure 3.16(a4) and (c4) is not meaningful because Figure 3.16(c4) does not represent the shape of Figure 3.16(c1) correctly.

The above discussion suggests that, before conducting resampling, it is necessary to check whether the uniform resampling grid is appropriate for the underlying harmonic image. In the following, we will first discuss why it is not appropriate for some Harmonic Shape Images and then we will present a method to check its validity.

Recall that resampling is performed on the harmonic image of a given surface patch. If the uniform resampling strategy is used, then the resampling result depends solely on the distribution of vertices on the harmonic image. Let us look at the harmonic images in Figure 3.16 as an example. The distribution of vertices in the harmonic images shown in Figure 3.16(a2) and (b2) is reasonably uniform. Therefore, the two resampled surfaces in Figure 3.16(a3) and (b3) are good enough to represent the original surfaces in Figure 3.16(a1) and (b1). In contrast, there is a high concentration of vertices in the harmonic image shown in Figure 3.16(c2). When uniform resampling is performed on it, the resampled mesh shown in (c3) is not a good reconstruction of the original surface patch. The reason for this problem is that, although the high concentration of vertices takes up only a small area in the harmonic image, their correspondences in the original surface patch actually take up a much larger area of the entire surface. In other words, there is much higher area compression for those triangles than for others in the original surface when being mapped onto the harmonic image. The area compression depends only on the shape of the underlying surface patch. In general, the more a surface patch is curved, the more area compression there is on its harmonic shape image.

After determining that the area compression is intrinsic to the shape of the surface patch, we have two approaches to solving the resampling problem. We perform a pre-processing step to determine whether uniform resampling will generate good resampled meshes. At this step, a 1D histogram called Area Histogram is created for each harmonic image and compared to a pre-stored histogram. If the newly generated Area Histogram is not consistent with the pre-stored one, then it is determined to be abnormal and no resampling will be performed on that harmonic image.

The area histogram is generated as follows. Given a unit disc and a certain number of concentric circles on that disc(Figure 3.17(a)), if the mass density of the disc is uniform everywhere, then we can compute the area (it is proportional to the mass) of each disc with increasing radius along the radial direction. The curve called the area histogram of the disc, which characterizes the relationship between radius and disc area, is denoted as a solid line in Figure 3.18. Now let us perform the same computation on a harmonic image with two modifications(Figure 3.17(b)). The first is that the harmonic image is discrete. The area of each disc with different radius is computed as the sum of the area of the triangles in that disc. The second modification is that the area of each triangle in the harmonic image is not its actual area. Rather, it is the ratio between the area of its correspon-

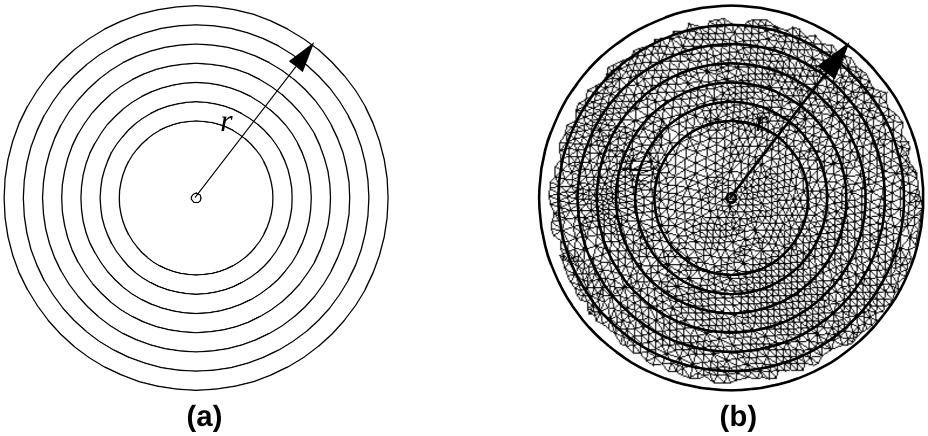


Figure 3.17: Illustration of area histogram.

dence triangle in the original mesh and the entire area of the original mesh. If the degree of area compression does not vary much across the harmonic image, then the area histogram of the harmonic image will be similar to that of the unit disc. Figure 3.18 shows the area histograms of some surface patches in Figure 3.16 and Figure 3.19.

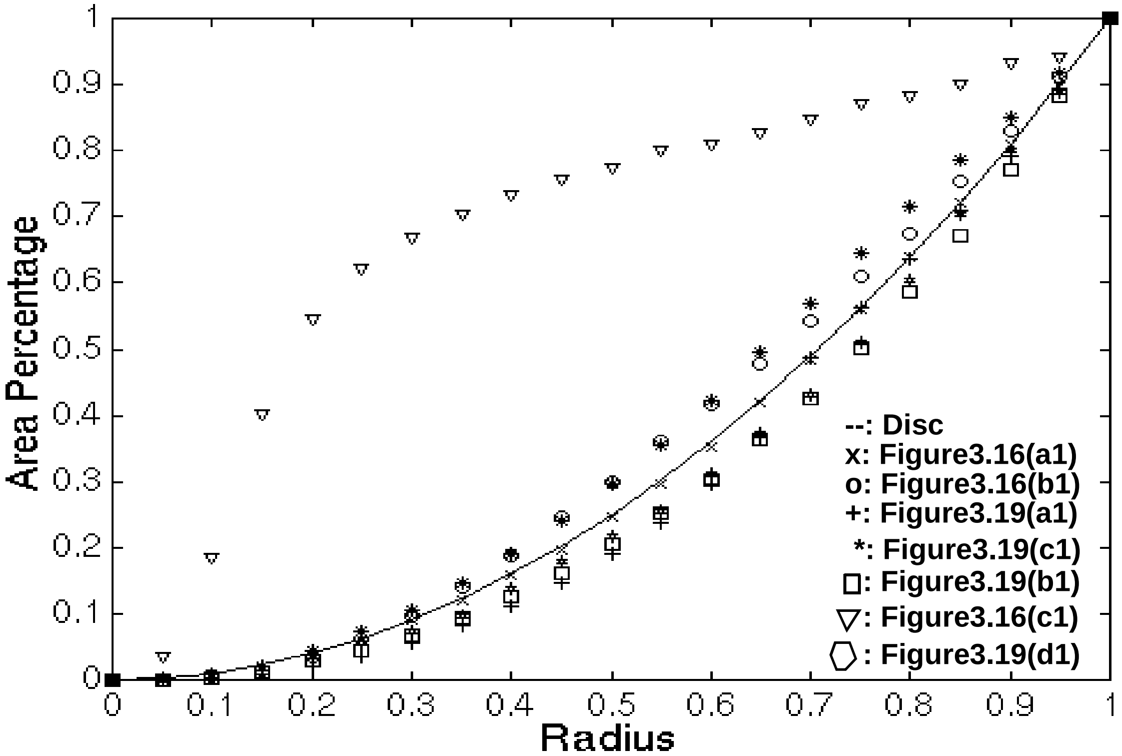


Figure 3.18: Area histograms of the surface patches in Figure 3.16 and Figure 3.19.

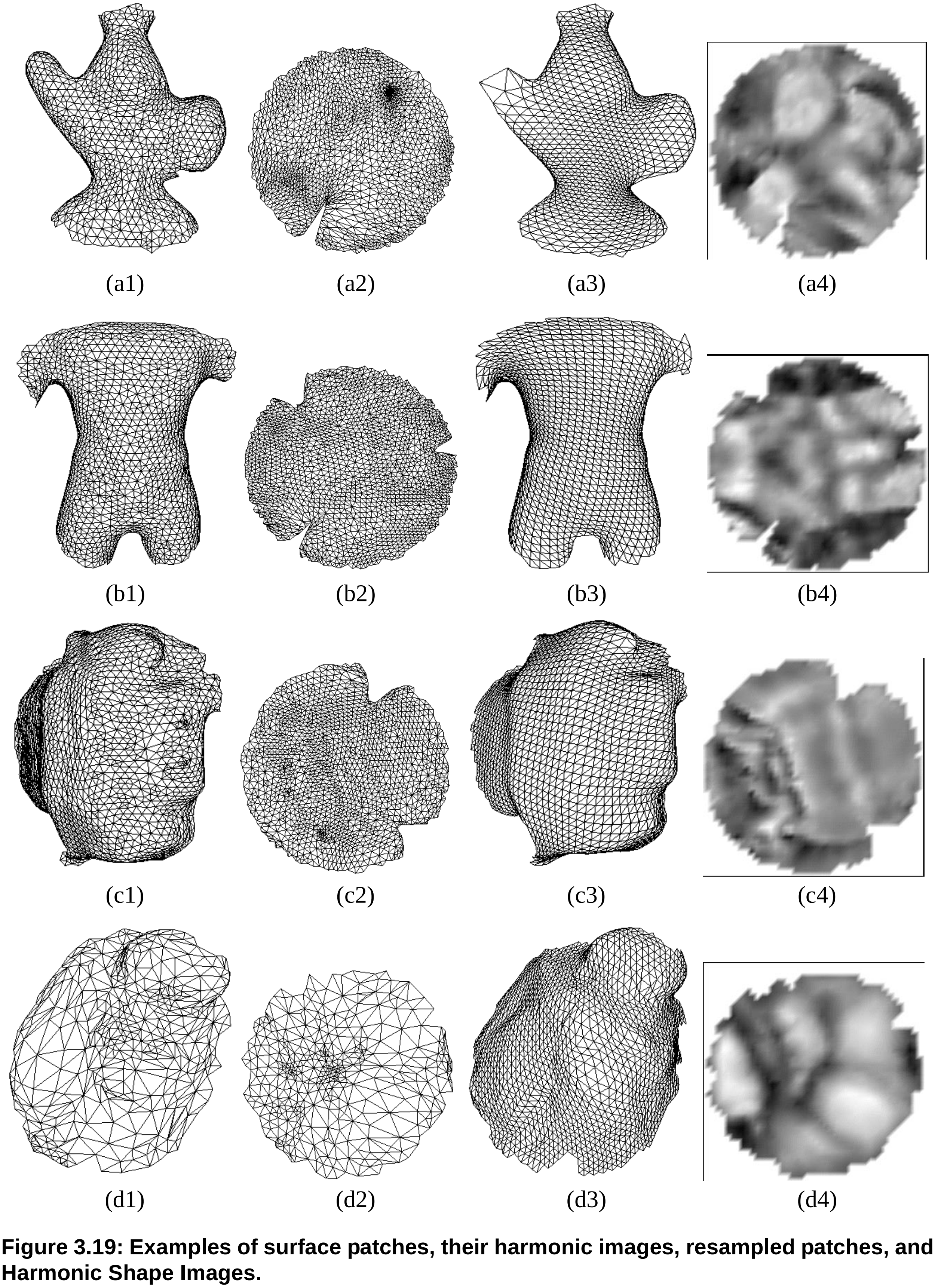


Figure 3.19: Examples of surface patches, their harmonic images, resampled patches, and Harmonic Shape Images.

Figure 3.18 shows that all the area histograms, except for the one for the surface patch in Figure 3.16(c1), are consistent with the area histogram of the disc. And their resampled meshes are reasonably uniform. In our implementation, the area histogram of the disc is pre-stored as the standard so that the area histograms of harmonic images can be compared with it. The result of the comparison determines whether uniform resampling can be performed on those harmonic images.

It should be pointed out that the distribution of vertices on the original surface patch does not affect the resampling strategy. Nor does it affect the area histogram. The surface patch in Figure 3.19(d1) is an example of this. The patch has the same shape as the surface patch in Figure 3.16(a1) but with different distribution of vertices. Its area histogram is very close to that of the surface patch in Figure 3.16(a1) (Figure 3.18). Using the uniform resampling strategy, the distribution of vertices on the resampled mesh (Figure 3.19(d3) is comparable to that in Figure 3.16(a3). As a summary, area histogram is a reliable way to detect whether uniform resampling is appropriate for resampling Harmonic Shape Images.

Area histogram can also be used to pre-eliminate unlikely matches from the search space because it is only determined by the shape of the underlying surface patch. In other words, the histogram is intrinsic to the shape of the surface patch. Therefore, if two surface patches have completely different area histograms, e.g., the surface patches in Figure 3.16(a1) and (c1), then they have different shapes. This pre-elimination can save the time for correlating the Harmonic Shape Images of those two surface patches.

Area histogram is a more formal way to check the validity of the uniform resampling grid. Another quick checkup can be performed by computing the ratio between the area and the circumference of the surface patch. When the ratio is high, it means that a large area is enclosed by a small circumference. In this case, the area compression is usually high on its harmonic image.

Chapter 4

Experimental Analysis of Harmonic Shape Images

The concept of Harmonic Shape Images -- how to generate and match Harmonic Shape Images, has been discussed in previous chapters. In this chapter, some properties of Harmonic Shape Images under different sources of noise are investigated. Due to the fact that it is difficult to obtain an analytical function of the Harmonic Shape Images with respect to noise such as surface resolution and occlusion, an experimental approach is employed to demonstrating the performances of the images.

In this chapter, four properties of Harmonic Shape Images are investigated: discriminability, stability, robustness to surface resolution and robustness to occlusion. In the following, these four properties are briefly introduced, followed by the introduction of the experimental setup.

Discriminability: Discriminability describes how discriminative a representation's ability to discern surfaces of different shapes. This is the basic requirement for all representations.

Stability: Stability measures the capability of representting surfaces of similar shapes using similar representations. This means that the representation does not change significantly when there are small changes on the underlying surface. Stability is an important property when conducting surface matching, e.g., it enables the use of the coarse-to-fine matching strategy. Detailed explanations are given in the following sections.

Robustness to surface resolution: In practice, the surfaces we deal with are discretek, i.e., no two surfaces to be compared have exactly the same sampling resolution. Without the property of being robust to surface resolution, the representations for surfaces of the same shape but of different sampling will be different. Obviously this is not what we expected from a qualified representation for surface matching.

Robustness to occlusion: Occlusion occurs quite often in practice either because of self occlusion or occlusion by other objects. In its presence, we should still be able to conduct surface matching using the non-occluded parts of the surfaces. This requires that the representation for the non-occluded parts of the surfaces remain approximately the same as that when there is no occlusion. Without the robustness to occlusion, surface matching would fail immediately when there is a small percentage of occlusion on the surfaces to be compared.

The basic setup of the experiments for testing the four properties are essentially the same. In the following, three key components of the setup are explained, including patch library, matching procedure and display format.

Patch Library: One or more libraries of surface patches are used in each experiment. Altogether there are three different kinds of libraries. The first consists of ten patches generated by using different parametric functions. The second one consists of sixteen patches extracted from real objects of different shapes. The third one consists of patches generated from the same real object. In some experiments, extensions are added to the second library by adding new surface patches.

Matching Procedure: A Harmonic Shape Image for each surface patch in a library is computed and stored. Then every patch is compared with every other one by comparing their Harmonic Shape Images, and the shape similarity value is computed. Suppose there are *n* patches in a library; there are n^2 shape similarity values including self-comparison of every patch.

Display Format: There are two display formats for the comparision result in each experiment. The first is a three-dimensional height field with the *X* and *Y* axes representing the indices of the surface patches in the library and the *Z* axis representing the shape similarity value for every pair of surface patches. For example, if there are *n* patches in the library, then the range of indices for the *X* and *Y* axes is $i \in [1, n]$ and $j \in [1, n]$, respectively. The shape similarity value for the patches (*i, j*) is represented by *z*(*i, j*). This display format is referred as the shape similarity matrix. The matrix is symmetric, i.e., $z(i, j) = z(j, i)$ because the shape similarity measure is symmetric. This display format provides an overall view of the comparison results among the surface patches in the library. The second display format is a histogram of the shape similartiy values for one patch as compared with all the patches in the library. The intepretation of the histogram has been discussed in Chapter 3. This display format provides an inside look at how different a surface patch is when compared to others.

In the following sections, each of the four properties of Harmonic Shape Images will be analyzed using the experimental approach that has been discussed above.

4.1 Discriminability

One of the important properties of shape representation is its ability to discriminate surfaces of different shapes. This is referred as the discriminability of a shape representation. In this section, the discriminability of Harmonic Shape Images is investigated under two different scenarios. In the first scenario, we study the discriminability of Harmonic Shape Images within a collection of surface patches of different shapes. The aim is to check whether any two surface patches can be distinguished according to their Harmonic Shape Images. In the second scenario, we study the discriminability on the surface of an individual object. This scenario is quite common in the application of object recognition. The aim of this experiment is to check whether a surface patch on the object surface can be distinguished effectively from other patches on the object surface according to their Harmonic Shape Images.

Two libraries of surface patches are created for conducting the first experiment. One library consists of 10 patches extracted from 10 different parametric surfaces. The 10 surfaces patches and their Harmonic Shape Images are shown in Figure 4.1. All the patches have the same radius. Among those patches, every pair is compared by matching their Harmonic Shape Images. The comparison result is summarized in a matrix form as shown in Figure 4.2. Figure 4.2(a) shows the normalized correlation coefficient for the comparison of every pair of surface patches. The *X*-axis and *Y*-axis represent the indices of surface patches from 1 to 10. The *Z*-axis represents the normalized correlation coefficients, i.e., *z(i, j)* is the correlation coefficient of surface patches *i* and *j*. Figure 4.2(a) shows that, the correlation coefficients on the diagonal are all equal to 1.0, while the correlation coefficients elsewhere are much lower than that. This simply shows the fact that every surface patch is completely correlated to itself in shape but not so to other patches.

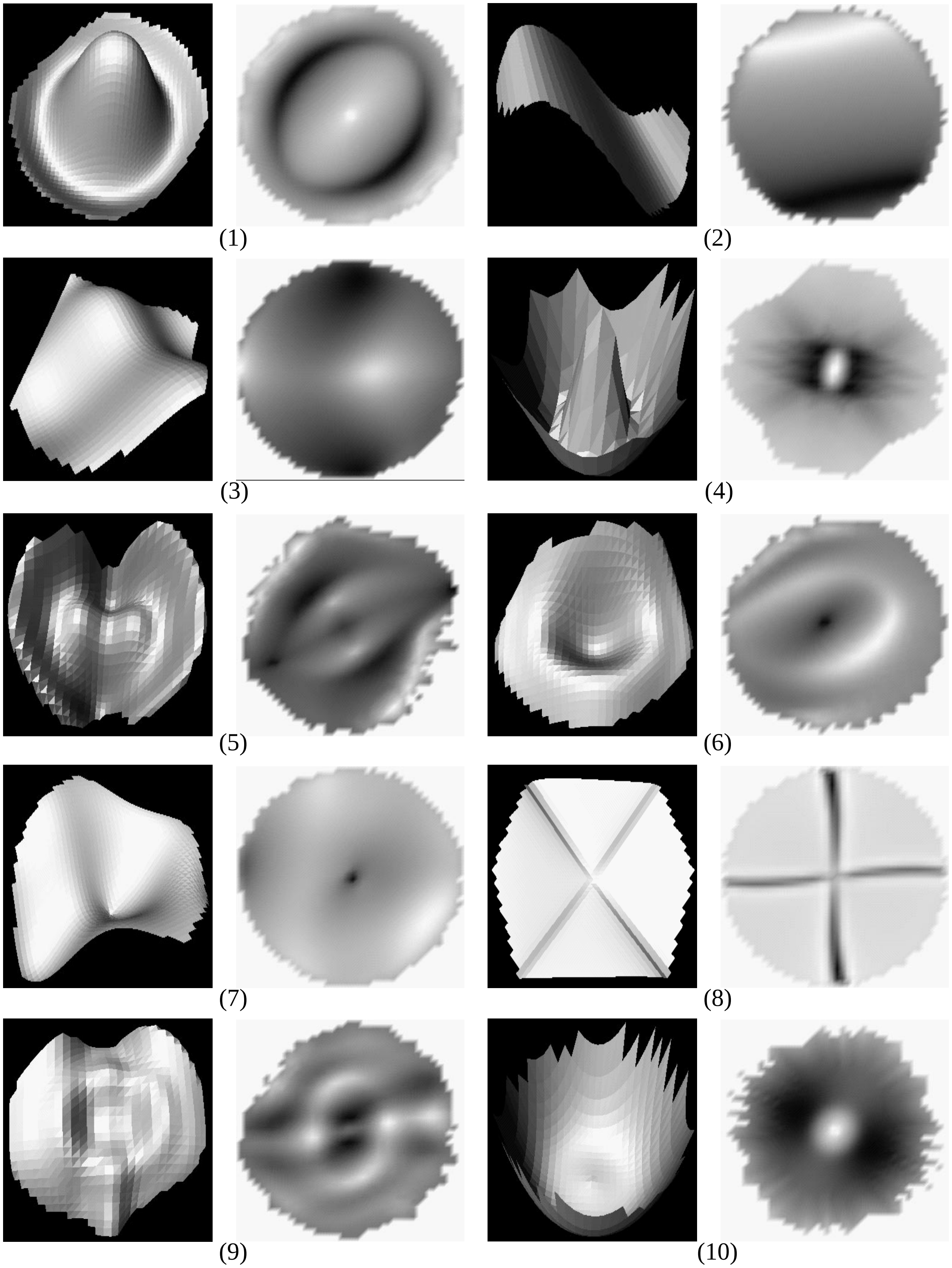
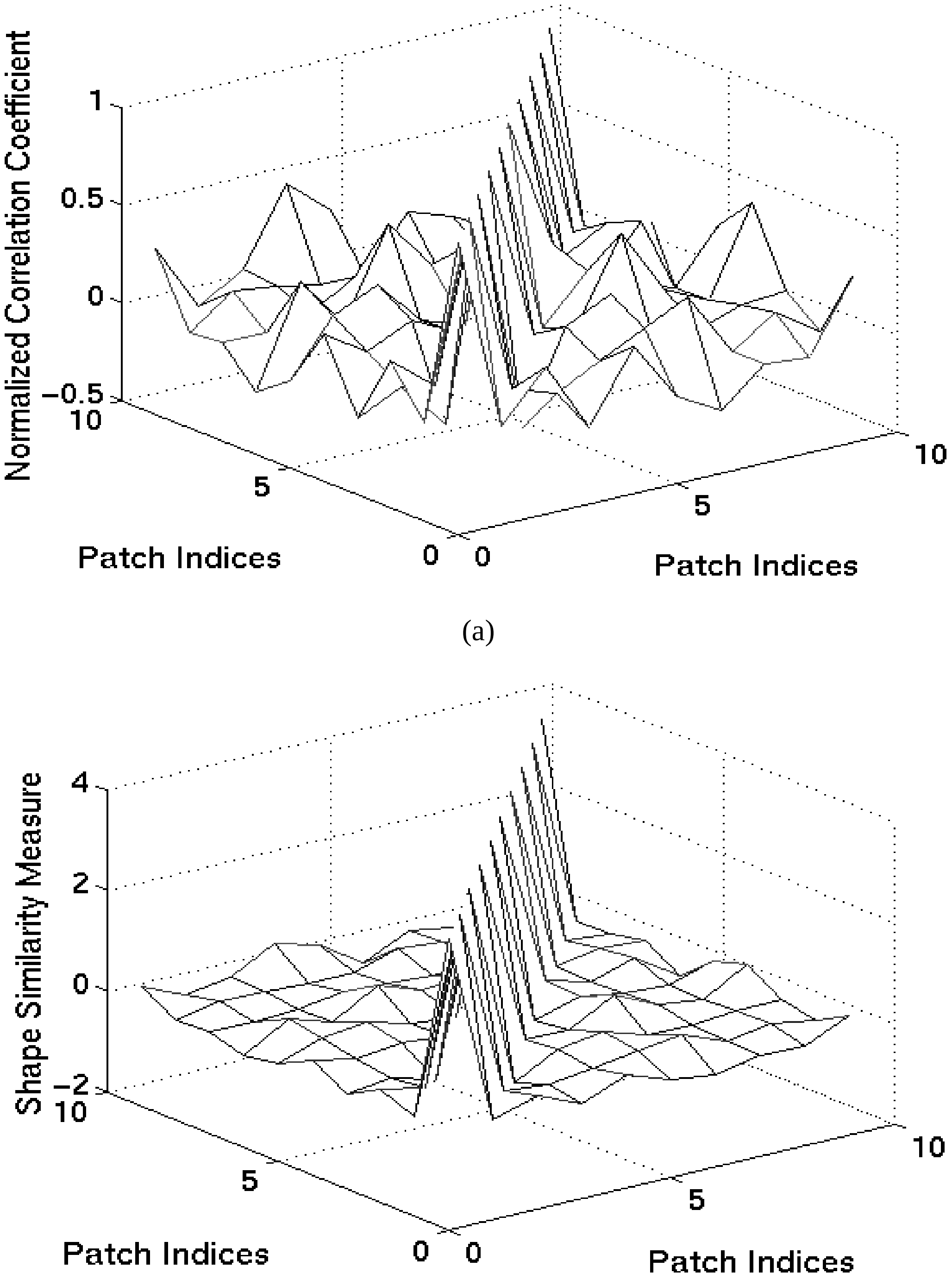


Figure 4.1: The library of 10 surface patches extracted from 10 parametric surfaces.



(a)

(b)

Figure 4.2: Pair comparison results of the surface patches in the first library. (a) The resultant matrix of the normalized correlation coefficients; (b) the resultant matrix of the shape similarity measure.

In order to quantify the difference between one surface patch and all the other patches, the shape similarity measure is computed for every pair comparison of surface patches. Figure 4.2(b) shows the resultant shape similarity matrix. The matrix in Figure 4.2(b) is similar to the matrix in Figure 4.2(a) and the shape difference between any patch and other patches is more obvious. Figure 4.3 and Figure 4.4 show the histogram of every row or column of the shape similarity matrix in Figure 4.2(b). Each histogram shows how different a surface patch is compared to all the patches in the library. The cut-off value for the qualified match is indicated by a line segment in every histogram. It can been seen that every surface patch is the only qualified match to itself, and that this match is well above the threshold of the qualified match.

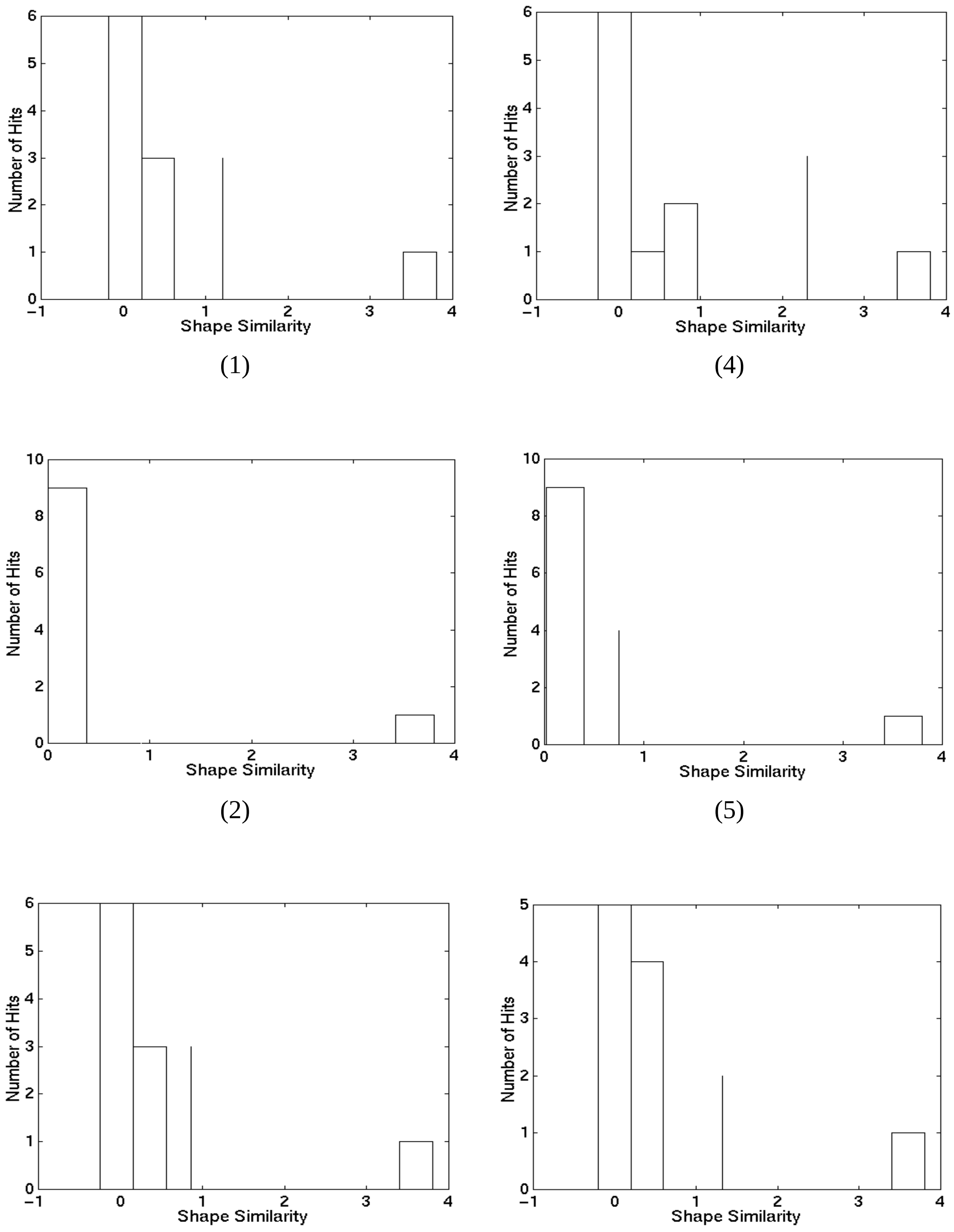
The second library consists of 16 patches extracted from real objects. The patches and their Harmonic Shape Images are shown in Figure 4.5 and Figure 4.6, respectively. The pair comparison results are shown in Figure 4.7 , Figure 4.8, Figure 4.9 and Figure 4.10. This example that each surface patch can be effectively discriminated from other patches in the library by using Harmonic Shape Images.

The next experiment is conducted among the surface patches extracted from the same object. The object is shown in Figure 4.11(a) with the target surface patch overlaid on it. The target patch has radius $R = 2.0$. Its wireframe and Harmonic Shape Images are shown in Figure 4.11(b) and (c), respectively. There are 2137 vertices on the object. Every vertex is used as the central vertex to generate a surface patch with $R = 2.0$. The matching is conducted between the target patch and each of the 2137 patches. The histogram of the shape similarity values from the matching is shown in Figure 4.12. It can be seen that a number of surface patches are above the qualified match threshold. The central vertices of those patches are shown in Figure 4.11(d).

From the above result, it can be seen that the target patch can be distingushed from 99.16% of the total number of patches on that object. This shows that the Harmonic Shape Image of the target surface patch is discriminative enough. However, a question may be asked why the target patch cannot be identified as uniquely as that in the previous two experiments. The answer is that the surface patches in the previous two libraries came from different objects and those patches are different from one another in shape. There is no relationship of any kind between any two patches. Therefore, a surface patch can always be identified uniquely. However, in this example, all the surface patches come from the same smooth surface. There are surface patches of similar shape because the central vertices of those patches are neighboring vertices and all the surface patches have the same radius. Under such circumstances, a target patch should only be matched to those patches that have similar shape. It should still be discriminated from other patches that have different shapes. The results shown in Figure 4.11(d) and Figure 4.12 demonstrate that the result is just as what we expected. The central vertices of the qualified patches are neighbors of the central vertex of the target patch. Figure 4.13 shows two qualified patches and their Harmonic Shape Images.

4.2 Stability

Stability is another important property of Harmonic Shape Images. Unlike discriminability which measures the capability of discriminating different shapes, stability measures the capability of identifying similar shapes. This property is extremely useful in the following two scenarios. The first one is that the surfaces we are dealing with in practice are discrete; therefore,

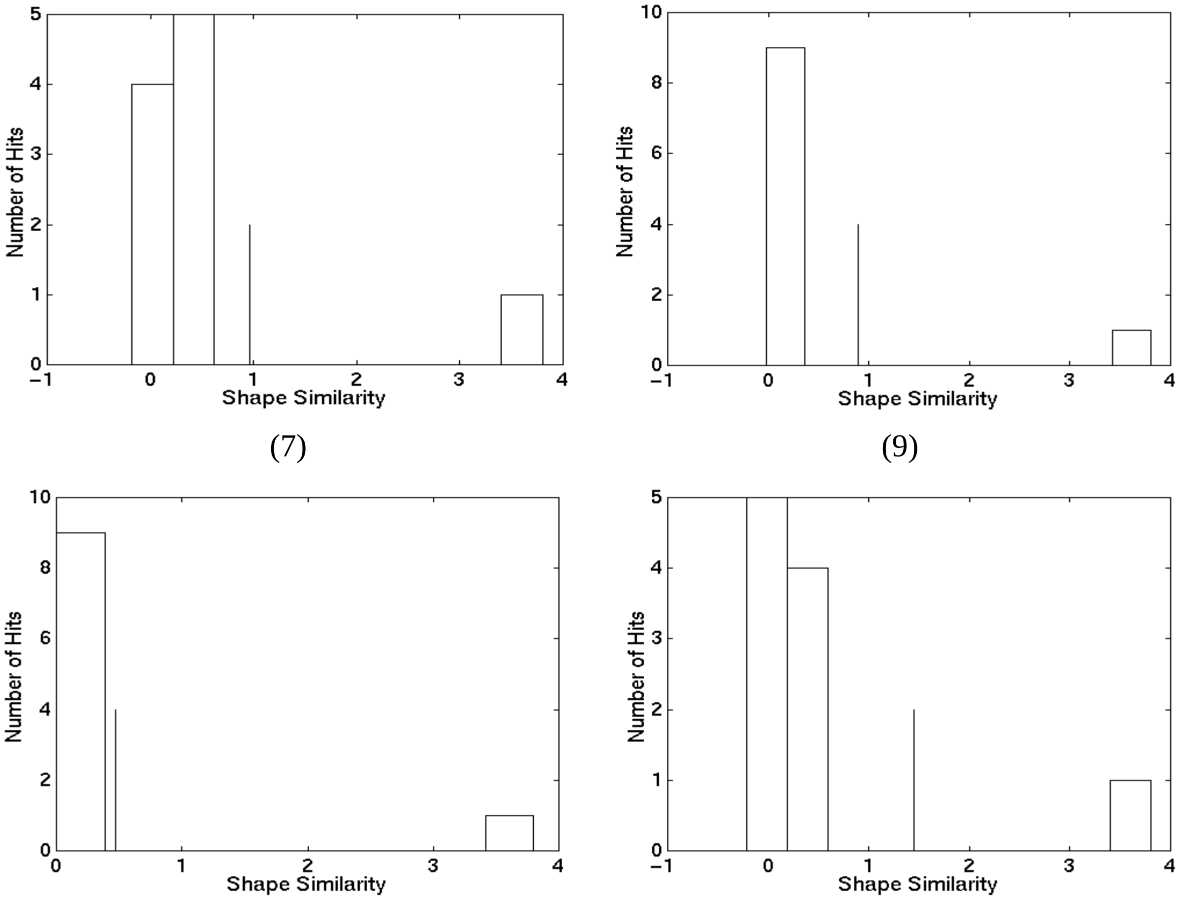


(1) (4)

(2) (5)

(3) (6)

Figure 4.3: Histogram of the shape similarity values for surface patch 1 through 6 to other patches.



(7)

(9)

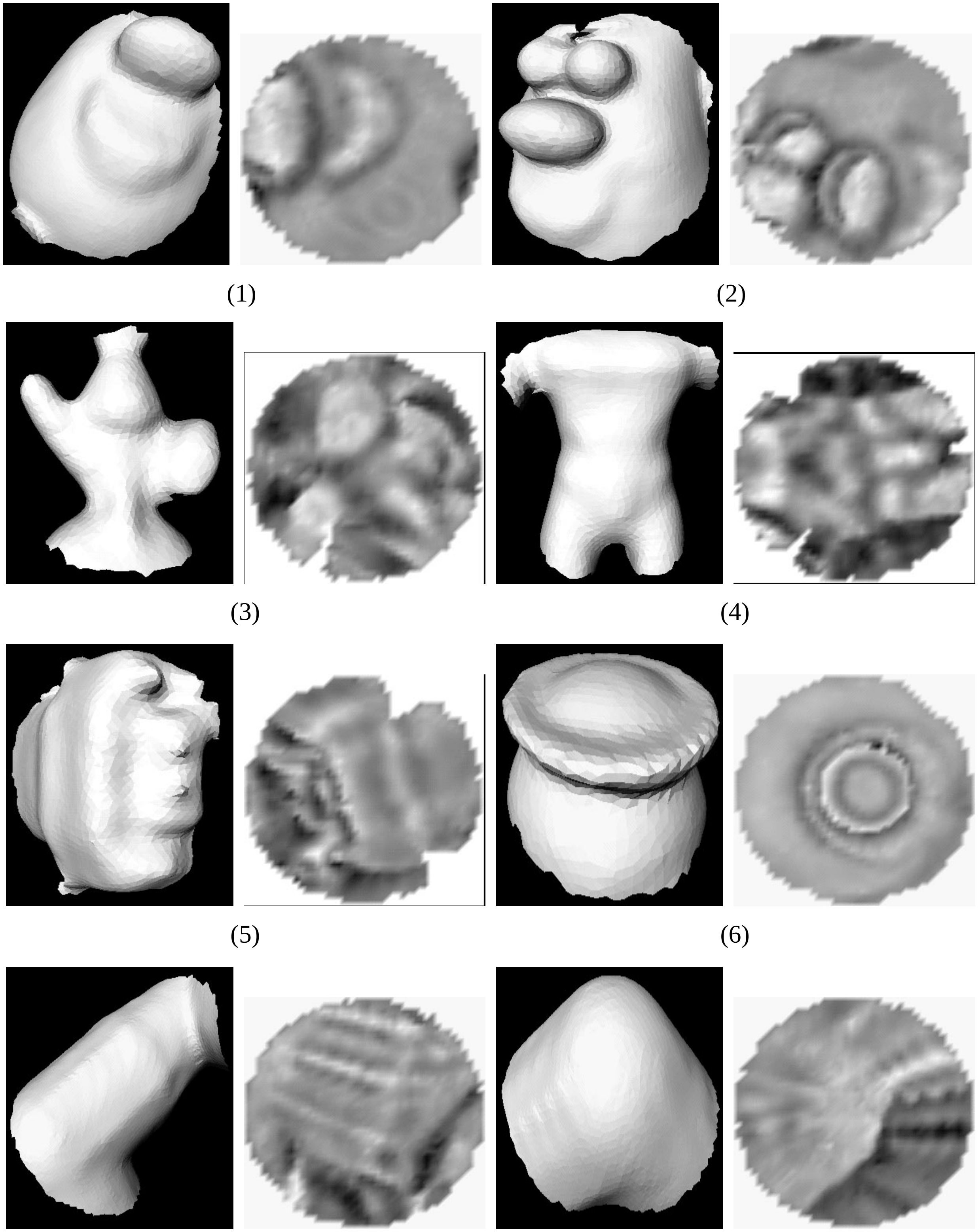
(8)

(10)

Figure 4.4: Histogram of the shape similarity value for every surface patch 7 through 10 to other patches.

it is almost impossible to have two surface patches be exactly the same due to the differences in sampling resolution. In this case, given a target patch on one surface, we cannot expect to find its match on the second surface with exactly the same central vertex as that of the target patch. In fact, there may be patches on the second surface that have central vertices that are very close to the central vertex of the target patch. Using Harmonic Shape Images, those patches should be identified correctly.

The second scenario is that, given one target patch on the first surface, we can sub-sample the second surface in order to speed up the matching process if we have the knowledge that neighboring vertices on a surface have similar patches and Harmonic Shape Images. In fact, we have already seen the application of this property in Chapter 3, in which the coarse-to-fine matching strategy was discussed. That strategy is based on the stability property of Harmonic Shape Images.



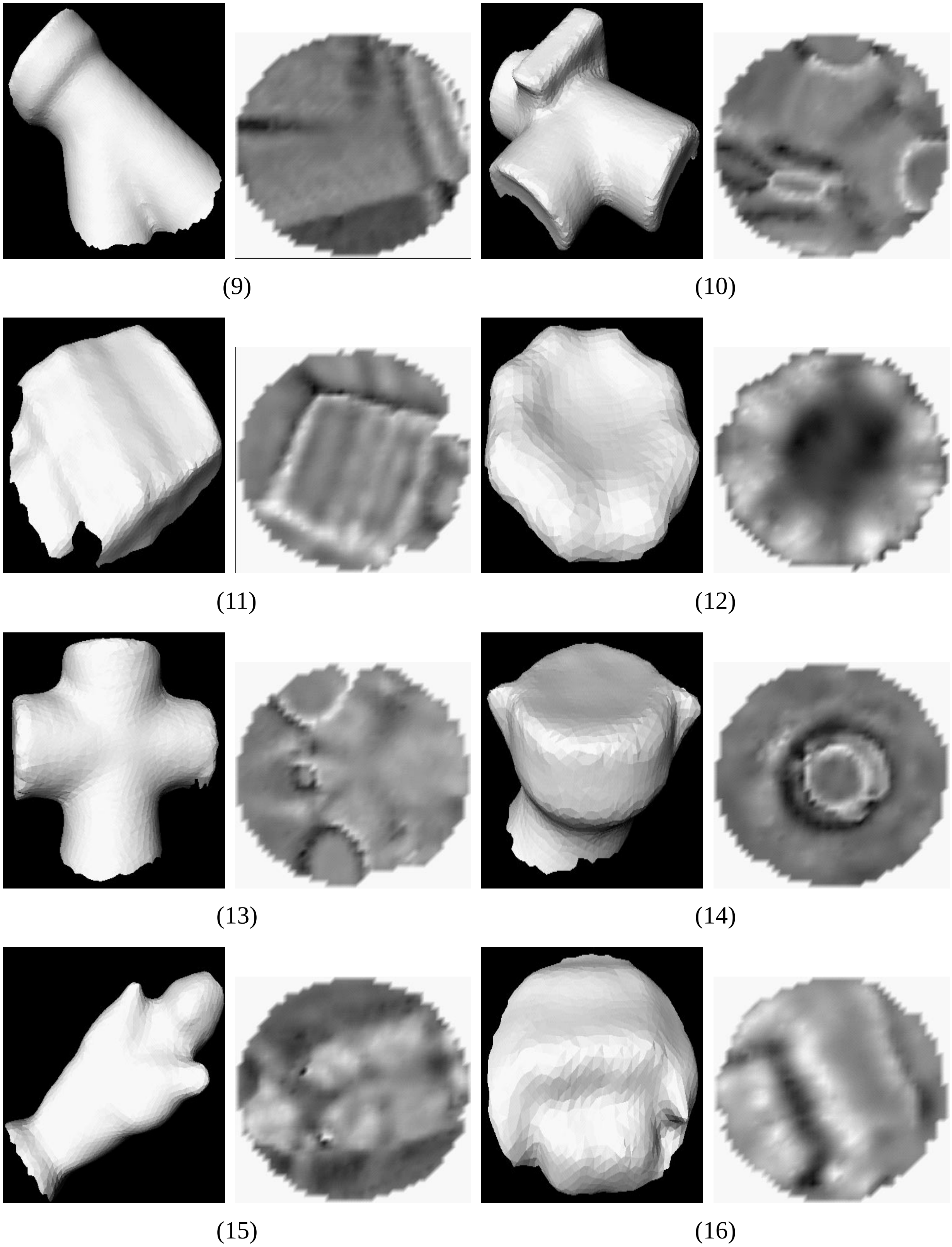
(1) (2)

(3) (4)

(5) (6)

(7) (8)

Figure 4.5: The patches 1 through 8 and their Harmonic Shape Images in the second library.



(9)

(10)

(11)

(12)

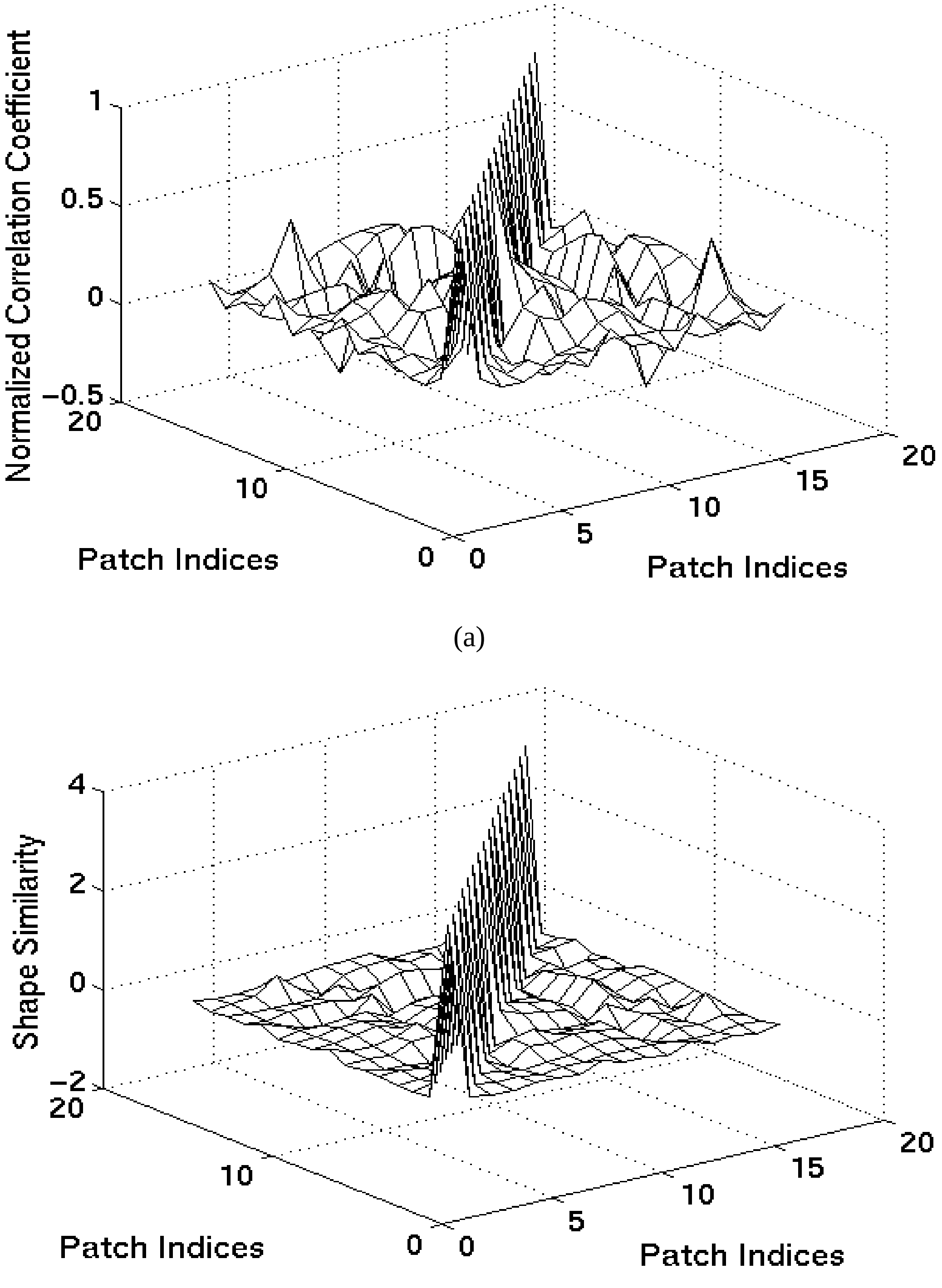
(13)

(14)

(15)

(16)

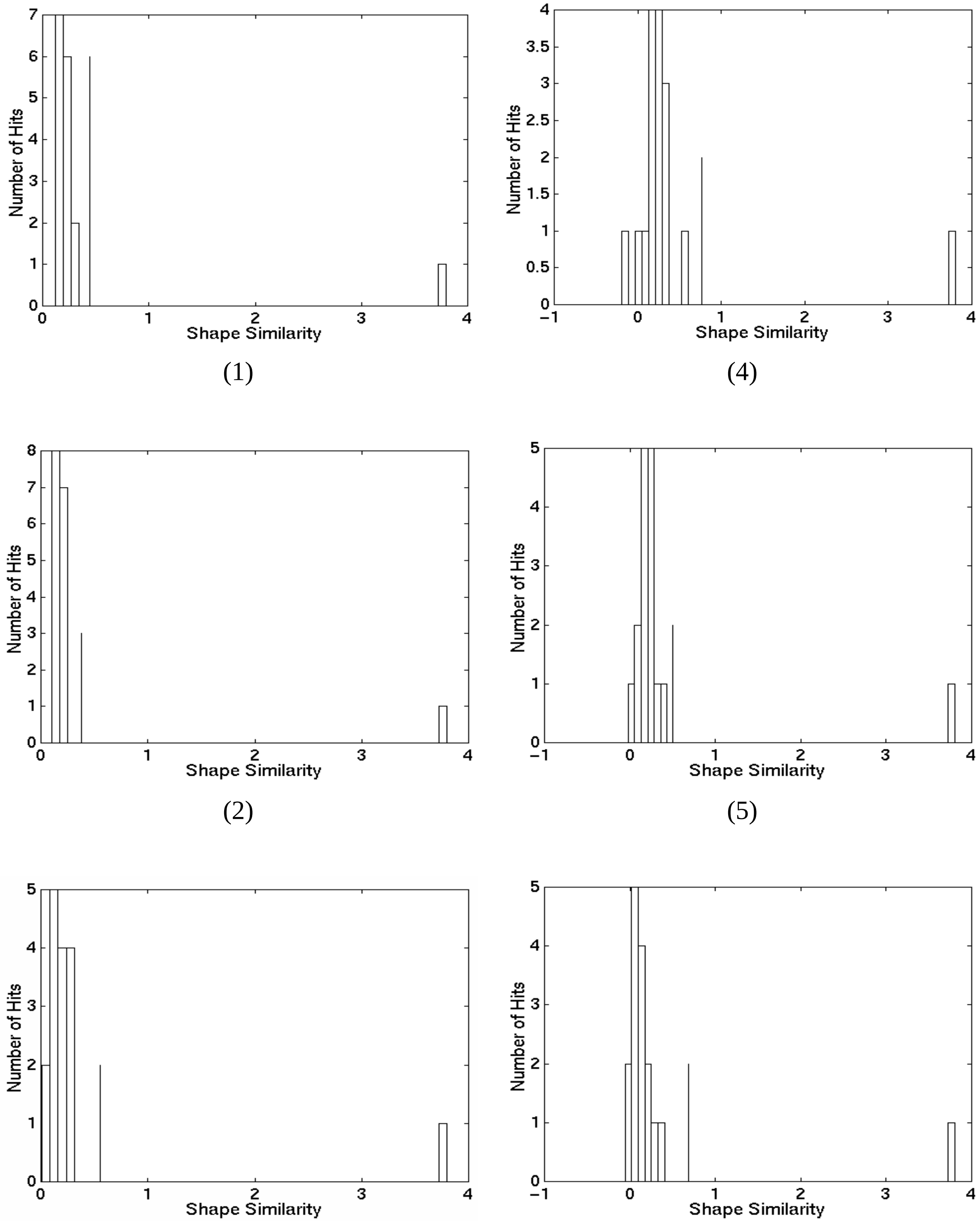
Figure 4.6: The patches 9 through 16 and their Harmonic Shape Images in the second library.



(a)

(b)

Figure 4.7: Pair comparison results of the surface patches in the second library. (a) The resultant matrix of the normalized correlation coefficients; (b) the resultant matrix of the shape similarity measure.



(1)

(4)

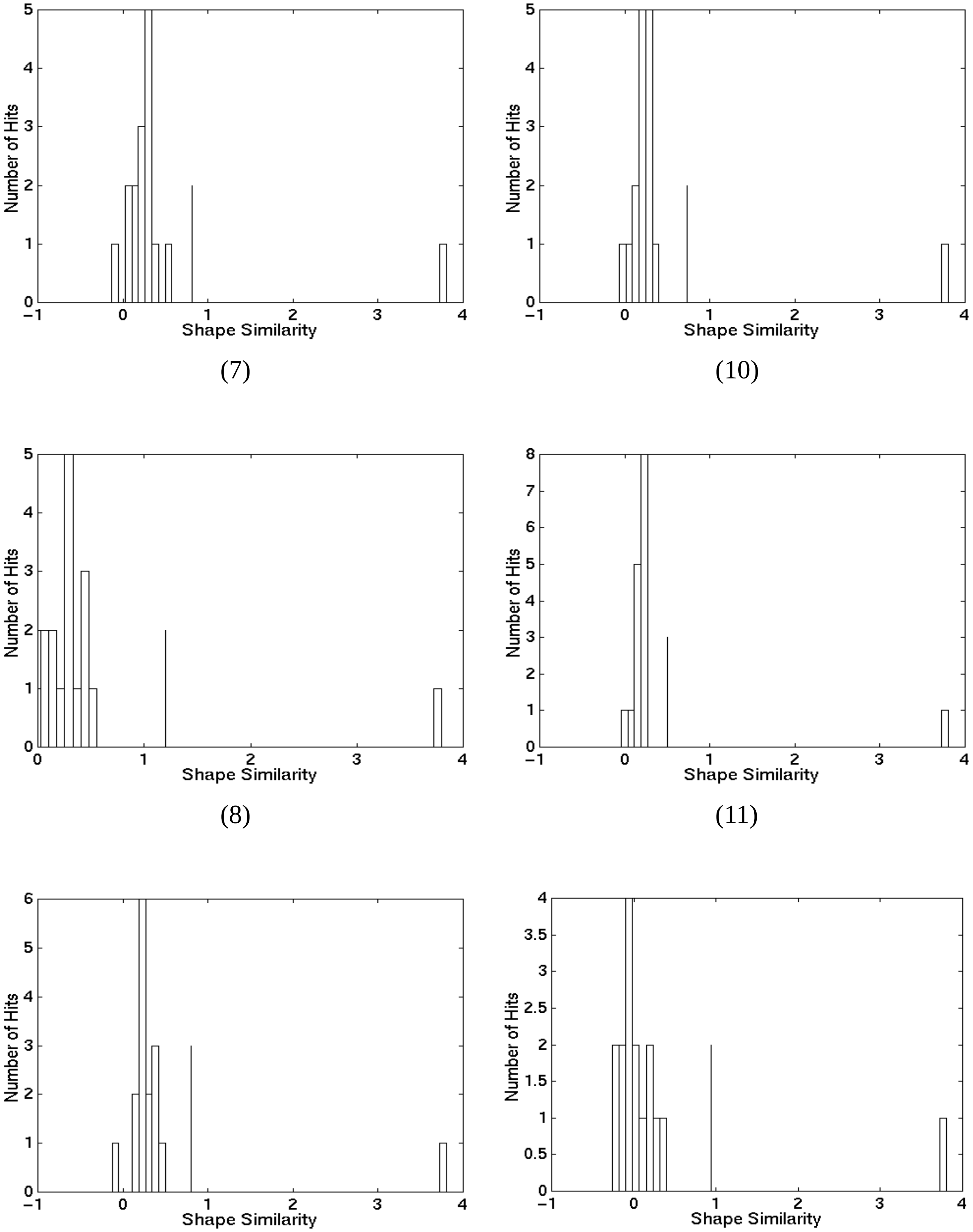
(2)

(5)

(3)

(6)

Figure 4.8: Histogram of the shape similarity values for surface patches 1 through 6 to other patches in the second library.



(7)

(10)

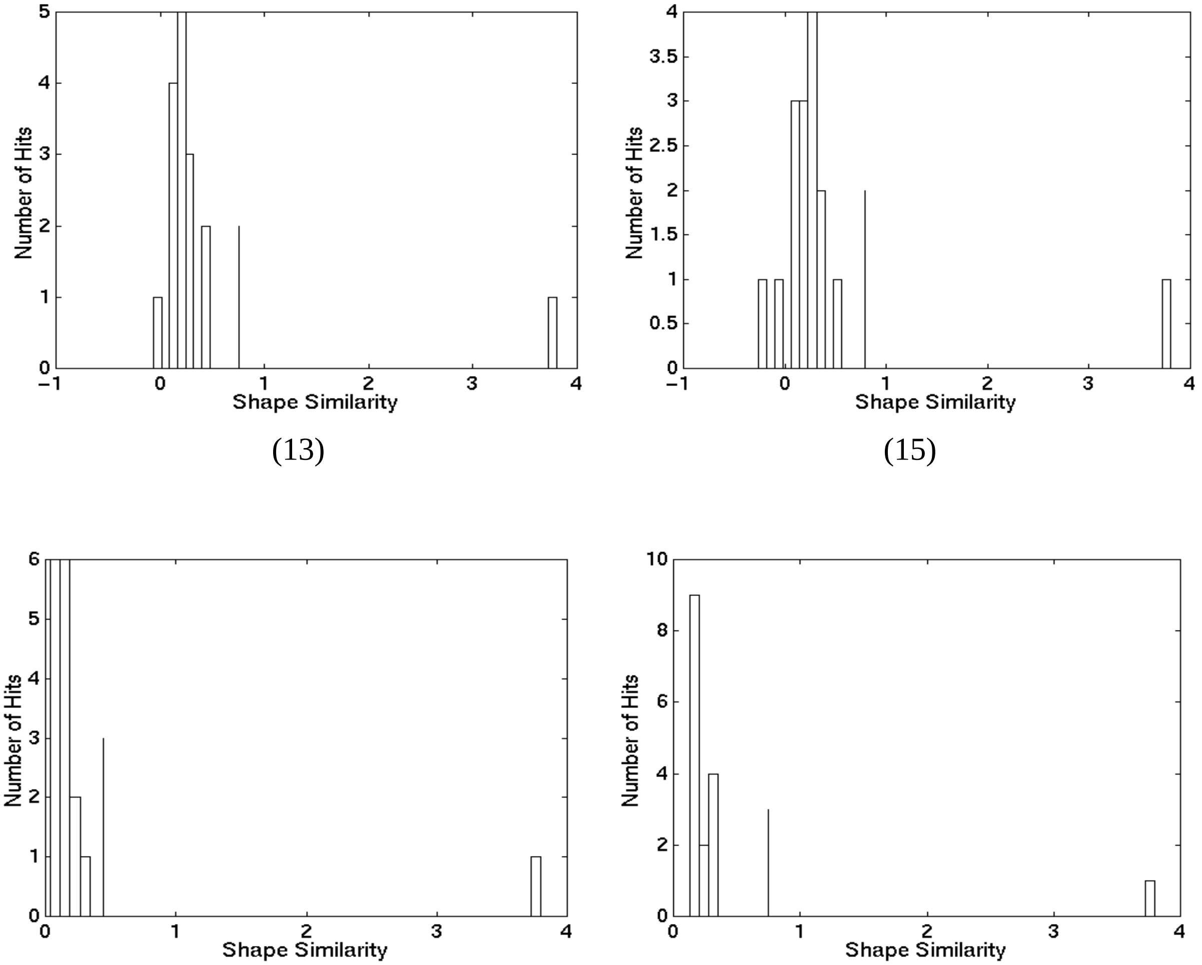
(8)

(11)

(9)

(12)

Figure 4.9: Histogram of the shape similarity values for patches 7 through 12 to other patches in the second library.



(13) (15)

(14) (16)

Figure 4.10: Histogram of the shape similarity values for surface patches 13 through 16 to other patches in the second library.

The example shown in Figure 4.11, Figure 4.12 and Figure 4.13 is also an example illustrating the stability property. The vertices shown in Figure 4.11(d) are the central vertices of the surface patches that matched the target patch. All are in the neighborhood of the central vertex of the target patch. Two more examples are shown in Figure 4.14 to Figure 4.19.

In the example shown in Figure 4.14 to Figure 4.16, the number of vertices on the object surface is 5950. Since the resolution is high, only one-third of the vertices are selected for conducting the experiment; those vertices are selected according to their indices -- every one in three is selected. This is just a simple way to subsample the original surface. The result in Figure 4.14(d) shows that the central vertices of the qualified patches are in the neighborhood of the central vertex of the target patch. This demonstrates the stability of Harmonic Shape Images. Two qualified patches and their Harmonic Shape Images are shown in Figure 4.16. They serve as a visual comparison to the target patch in Figure 4.14.

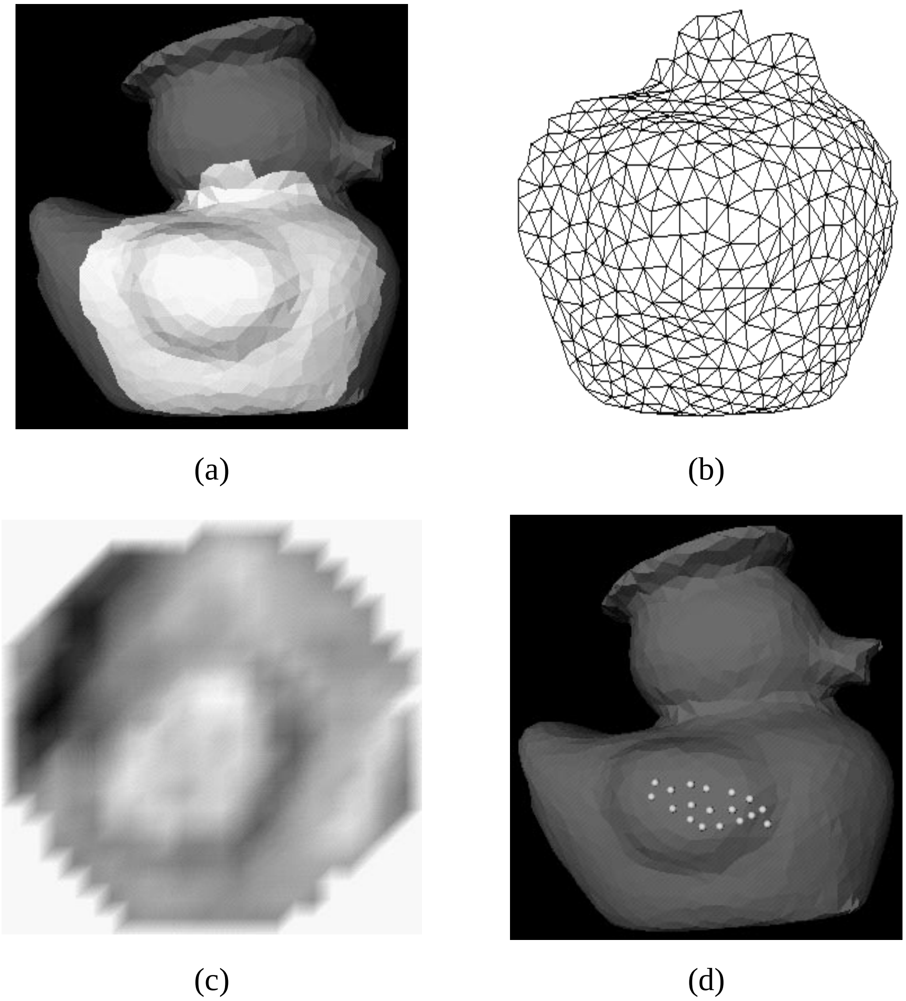


Figure 4.11: Identify one surface patch from all the surface patches on one object. (a) The target surface patch overlaid on the object; (b) the wireframe version of the target surface patch; (c) the Harmonic Shape Image of the target surface patch; (d) the central vertices of the qualified surface patches that match the target patch.

The example object surface shown in Figure 4.17 has 3684 vertices. Half of those vertices are selected for conducting the stability experiment. The central vertices of the qualified matches are shown in Figure 4.17(d). Figure 4.18 shows the histogram of the shape similarity values of the matching result. Figure 4.19 shows that the examples of the matched patches are quite similar to the target patch in Figure 4.17.

4.3 Robustness to Resolution

It has been discussed in Chapter 2 that Harmonic Shape Images do not depend on any specific sampling strategy, e.g., uniform sampling. For a given surface, as long as the sampling rate is high enough such that the shape of the surface can be sufficiently represented, its Harmonic Shape Image is also accurate enough for surface matching. It should be noted that the comparison of

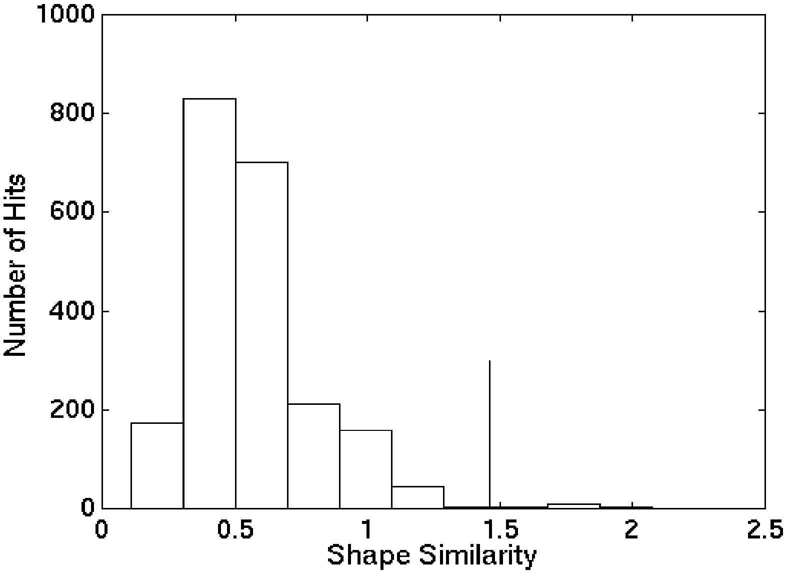


Figure 4.12: Histogram of the shape similarity values from the matching experiment.

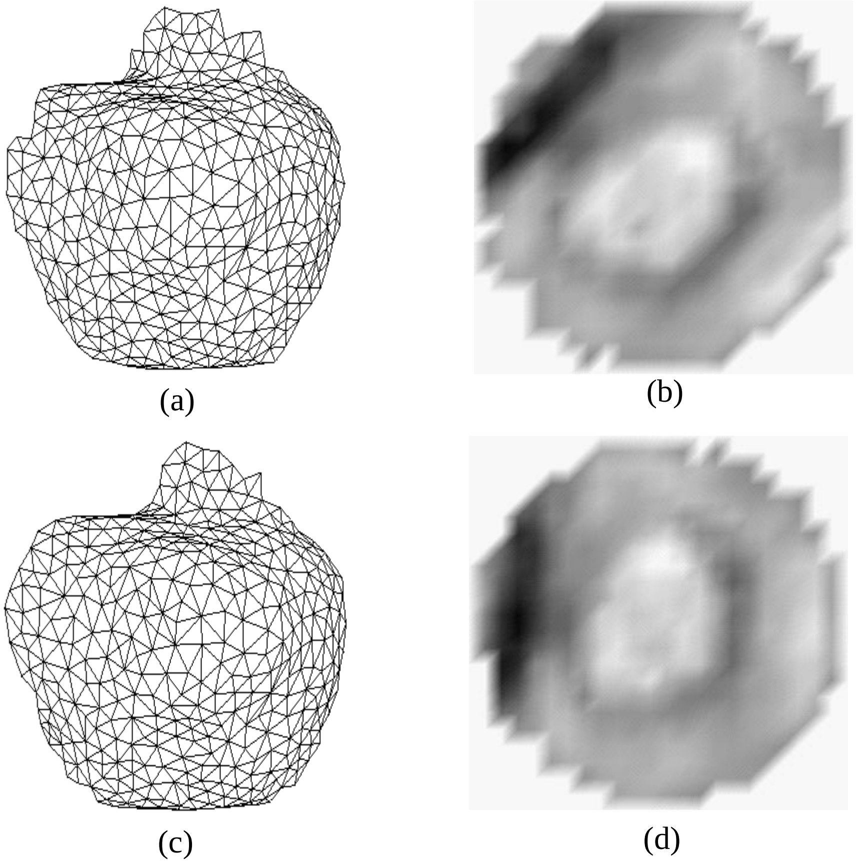

(a)
(b)
(c)
(d)

Figure 4.13: Examples of the qualified patches of the target patch in Figure 4.11.

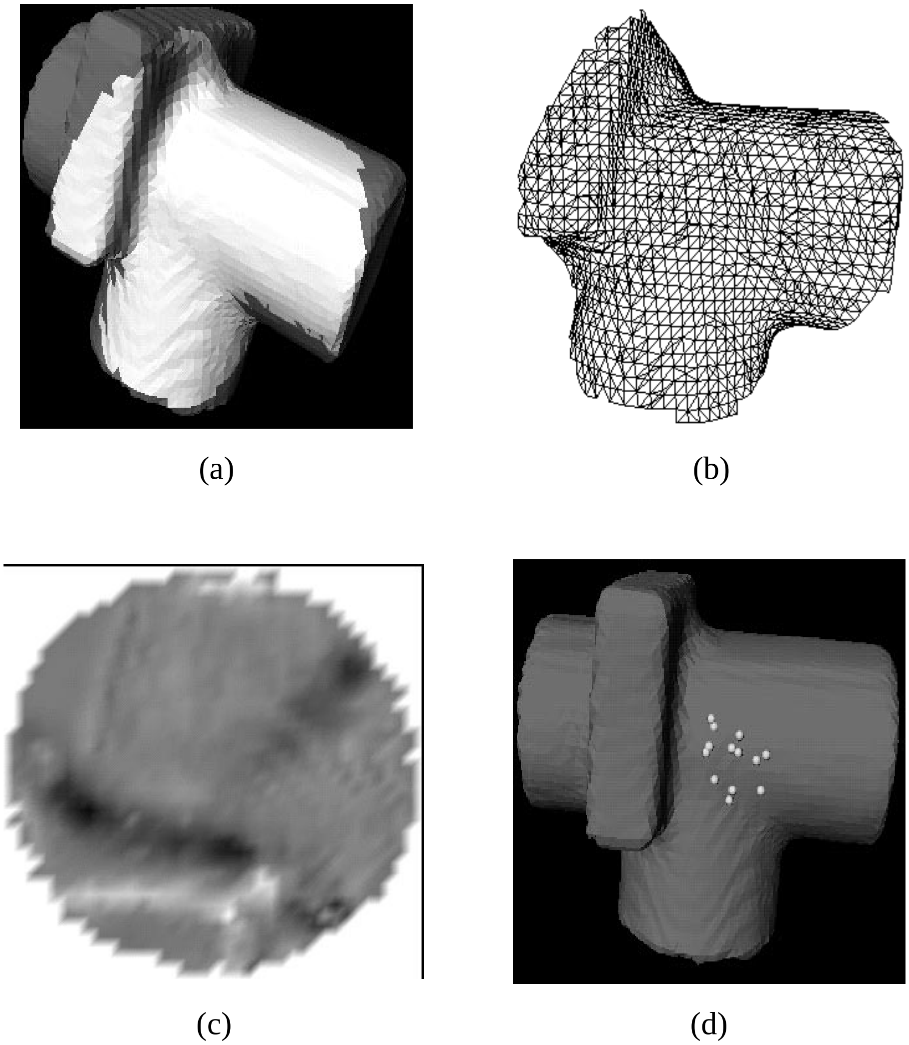


Figure 4.14: Identify one surface patch from all the surface patches in one object. (a) The target surface patch overlaid on the object; (b) the wireframe version of the target surface patch; (c) the Harmonic Shape Image of the target surface patch; (d) the central vertices of the qualified surface patches that match the target patch.

Harmonic Shape Images does not require that the two surface patches have the same sampling frequency. In practice, it is rare for discrete surfaces to have exactly the same sampling frequency. Although different resolutions may introduce noise in creating Harmonic Shape Images, the following experiments show that Harmonic Shape Images are robust to this kind of noise.

The first experiment is conducted using the surface patches in the second patch library shown in Figure 4.5 and Figure 4.6. Different versions of the first patch(Figure 4.5(1)) in the library are generated using a mesh decimation program[33]. The new meshes have different sampling resolution from one another. Although they are supposed to have the same shape as that of the original one, the shape gradually changes as the sampling resolution decreases. The new meshes are shown in Figure 4.20 and Figure 4.21 along with their Harmonic Shape Images. These meshes are added to the patch library.

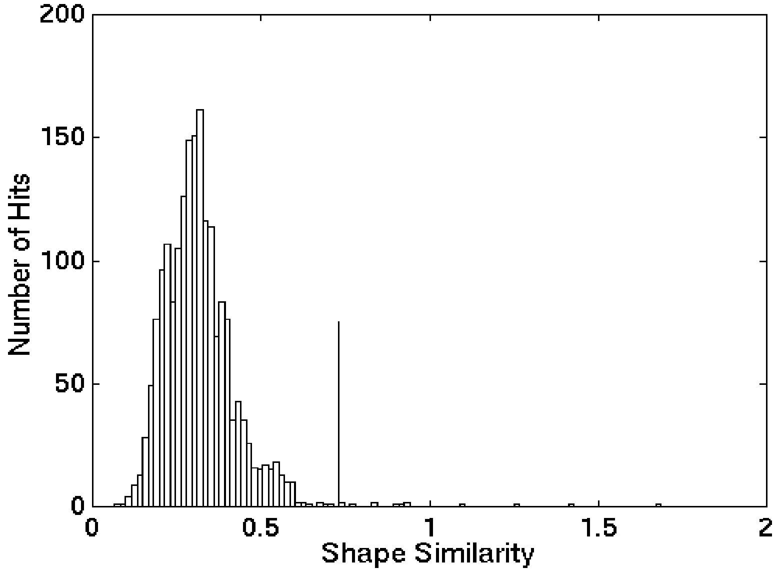


Figure 4.15: Histogram of the shape similarity values for the matching experiment in Figure 4.14.

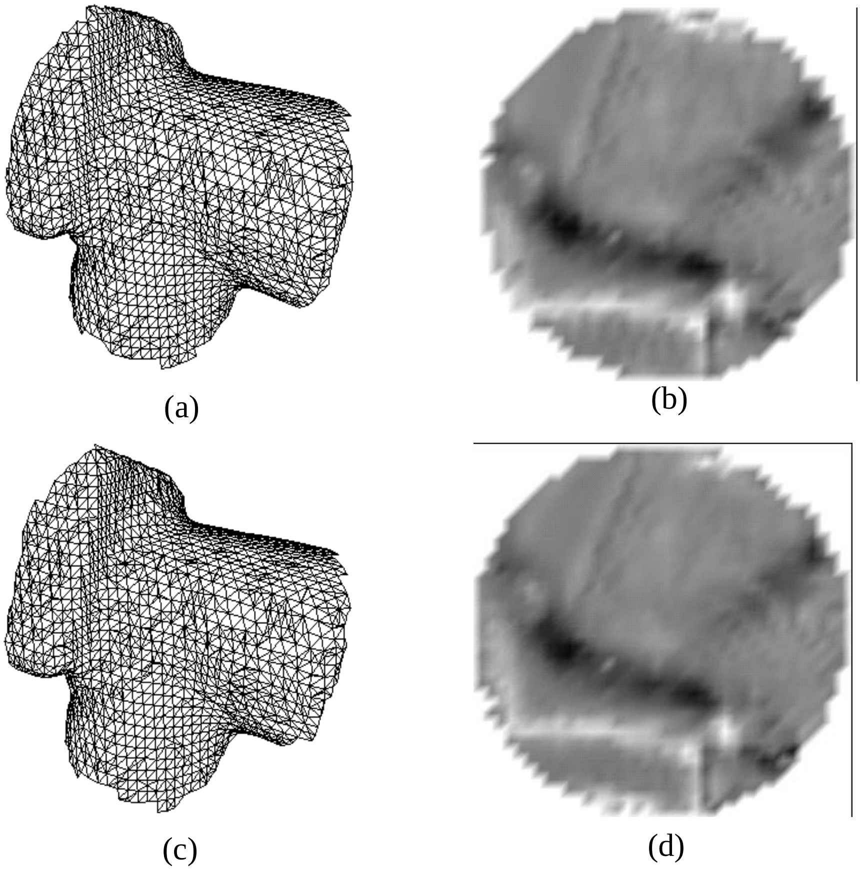
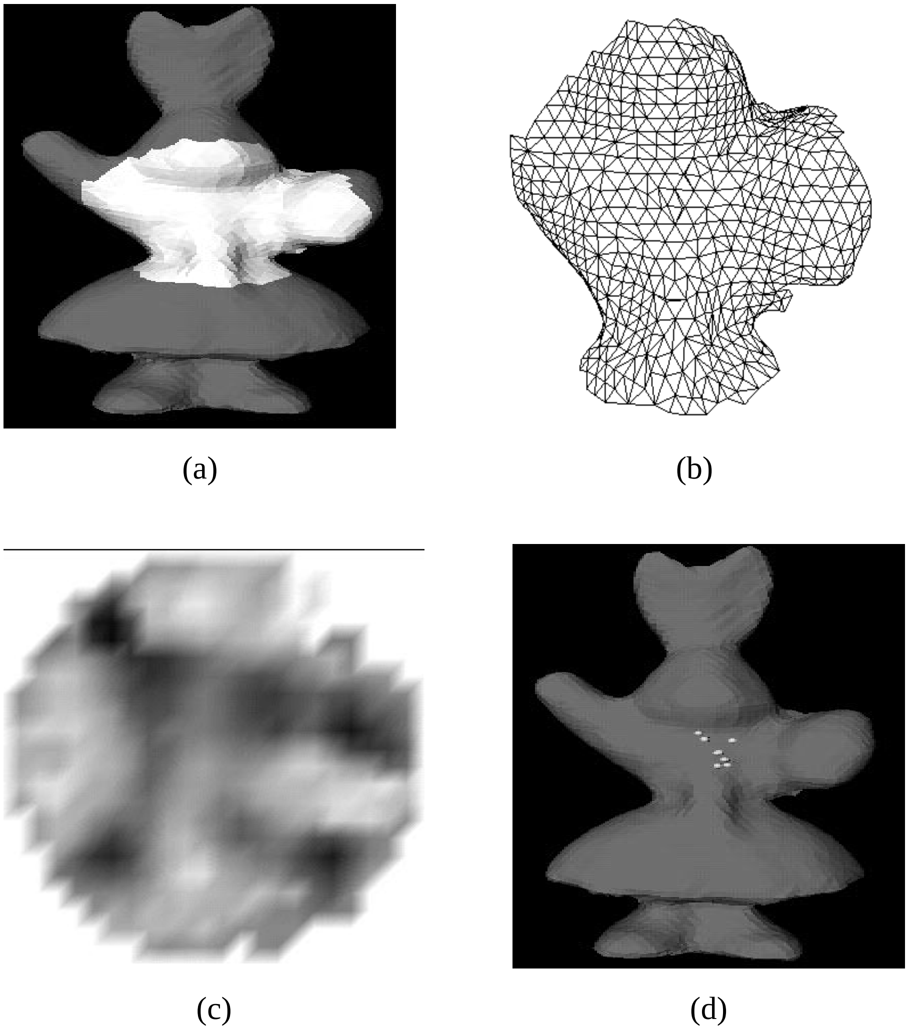


Figure 4.16: Examples of the qualified patches of the target patch in Figure 4.14(b).



(a) (b)

(c) (d)

Figure 4.17: Identify one surface patch from all the surface patches in one object. (a) The target surface patch overlaid on the object; (b) the wireframe version of the target surface patch; (c) the Harmonic Shape Image of the target surface patch; (d) the central vertices of the qualified surface patches that match the target patch.

As in the previous experiments, pair comparison is performed for every pair of surface patches in the extended library. The expected result is that the new patches are identified to be similar to the original patch and different from other patches in the library. Figure 4.22(a) and (b) show the comparison result. The matrix of the normalized correlation coefficients is shown in Figure 4.22(a). As before, the correlation coefficients on the diagonal are self-comparison for each patch in the library, therefore having the value of 1.0. In addition to the diagonal, there are high correlation values between the patches indexed from 17 to 23 to the first patch. The patches indexed from 17 to 23 are the patches of the same shape as patch one but are of different resolutions. The correlation values for the pair comparison among those patches 17 to 23 are also high as shown in Figure 4.22(a) by the block in the back of the figure. This distribution of correlation values is exactly as we had expected. The matrix of shape similarity values in Figure 4.22(b) also verifies this result.

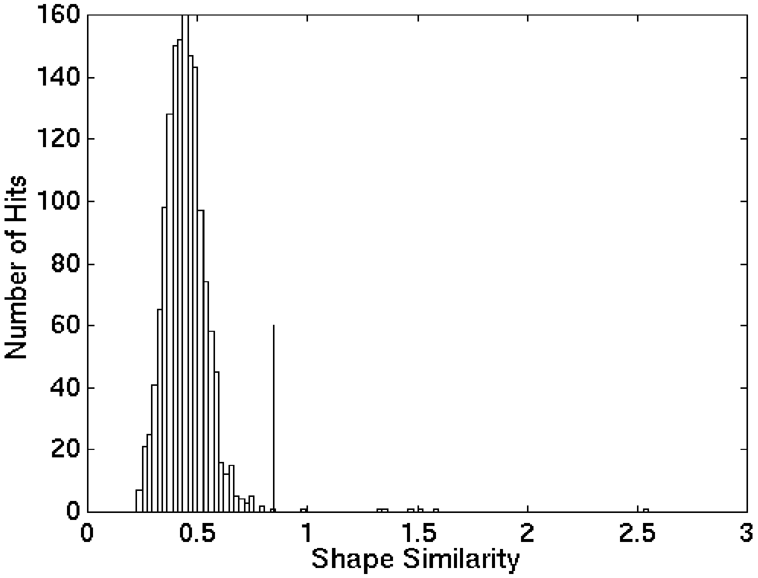


Figure 4.18: Histogram of the shape similarity values for the matching experiment in Figure 4.17.

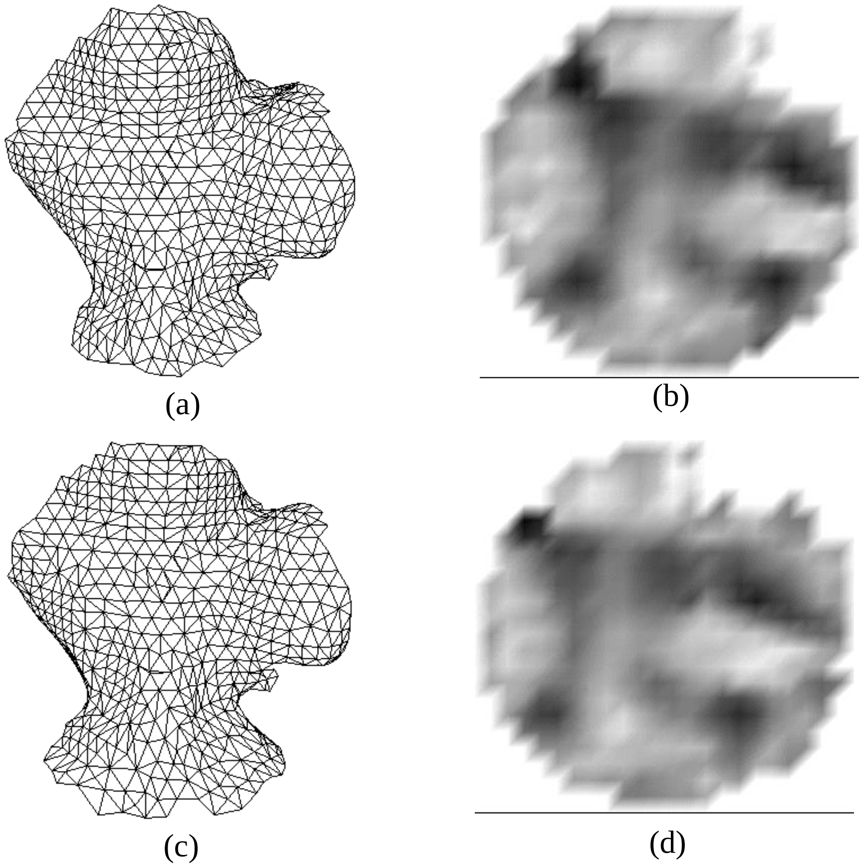


Figure 4.19: Examples of the qualified patches of the target patch in Figure 4.17(b).

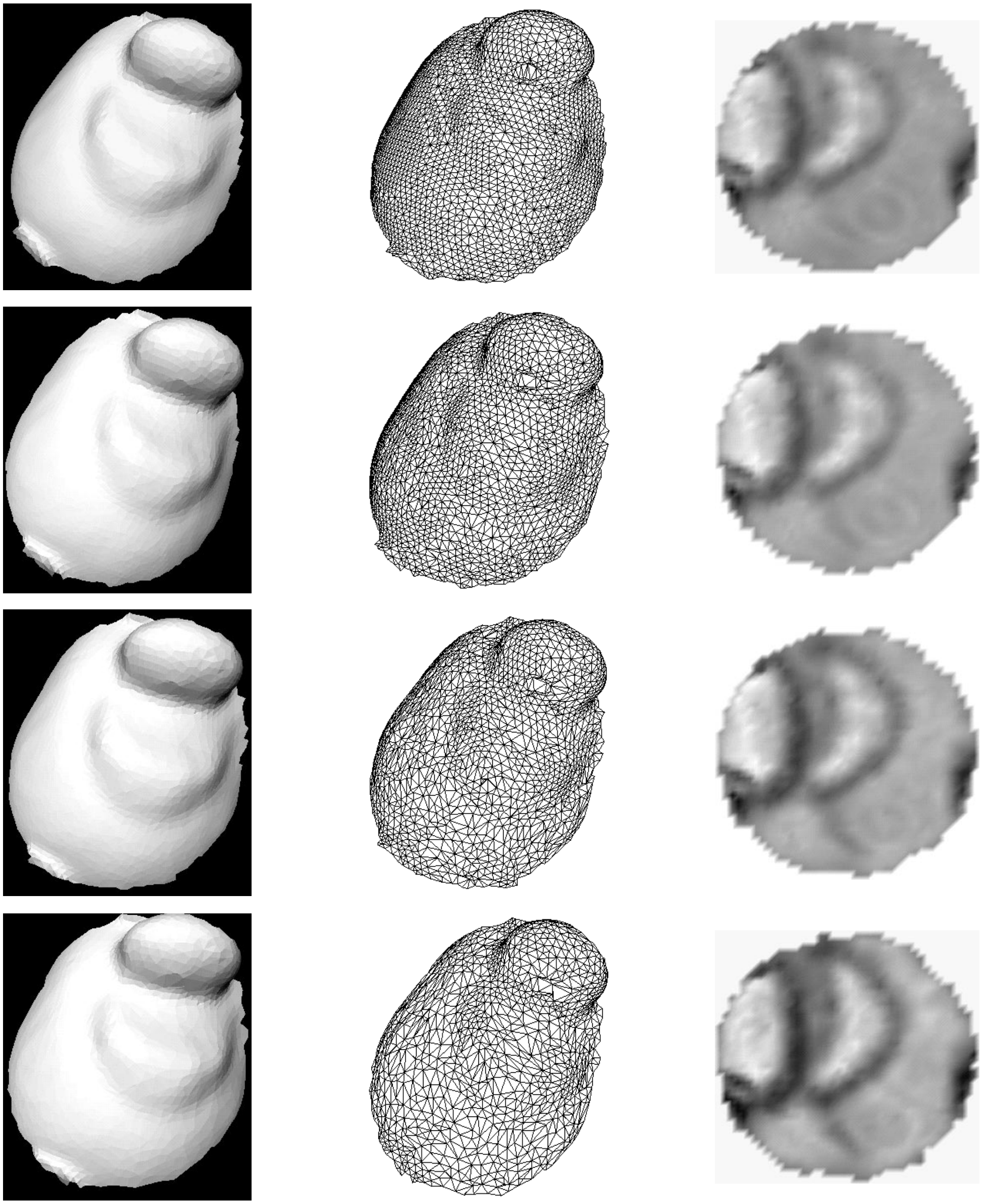


Figure 4.20: Surface patches of the same shape but of different resolutions. Left column: rendered surfaces; middle column: surfaces in wireframe; right column: Harmonic Shape Images: From top to bottom, the number of vertices are 2073, 1689, 1344 and 1028.

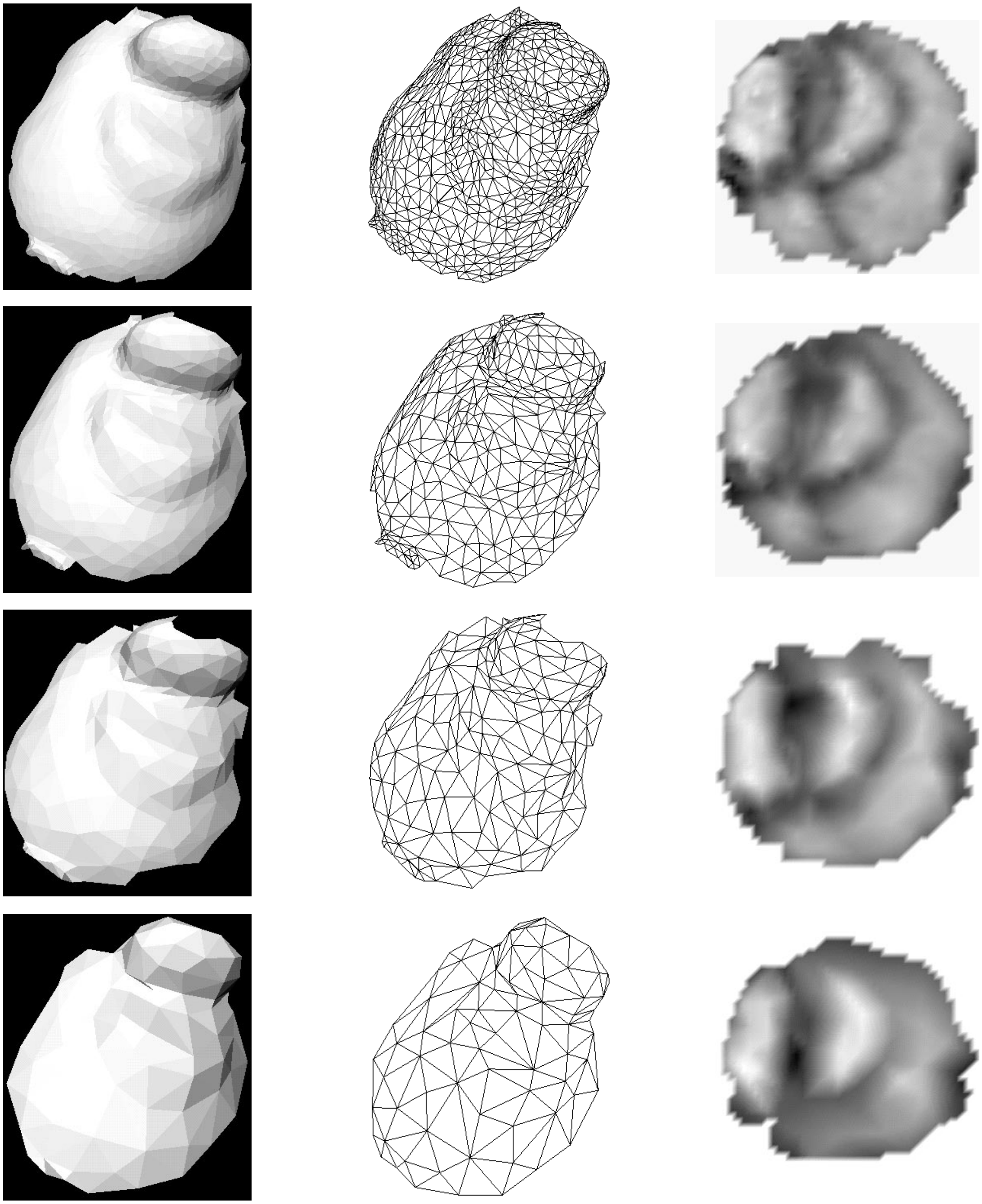
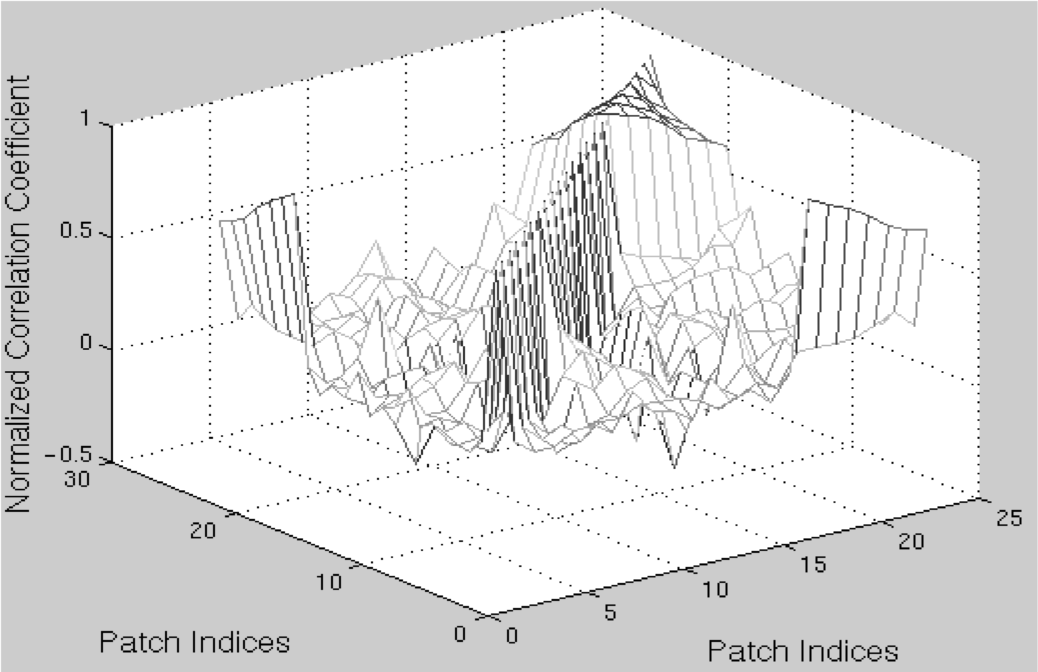
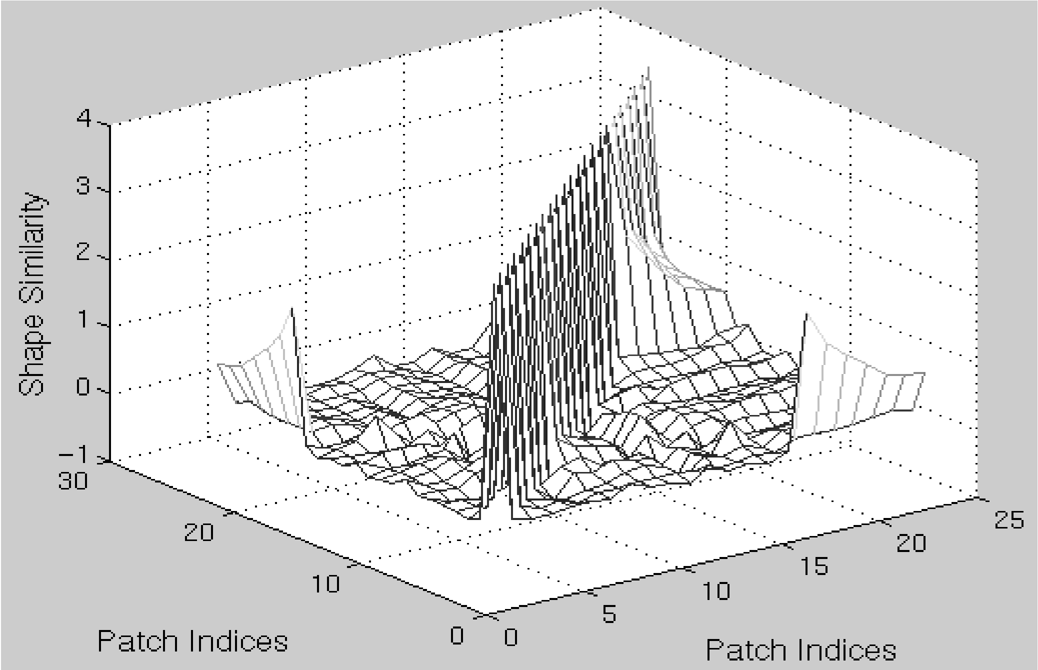


Figure 4.21: Surface patches of the same shape but of different resolutions. Left column: rendered surfaces; middle column: surfaces in wireframe; right column: Harmonic Shape Images: From top to bottom, the number of vertices are 686, 324, 169 and 82.



(a)



(b)

Figure 4.22: Pair comparison results of the surface patches in the second library after adding the surface patches with different resolution. (a) The result matrix of normalized correlation coefficients; (b) the result matrix of shape similarity measure.

In order to show clearly how similar the new patches are to the original patch and how different they are from other patches, histograms of the shape similarity values in Figure 4.22(b) are shown in Figure 4.23. There, the number of qualified matches is 79. This includes the self comparison(23), the new patches to the original patch and vice versa(7x2=14) and the pair comparison among the new patches(7x6=42). Figure 4.24 and Figure 4.25, show the histogram for every new patch compared to the 16 patches in the library is shown. It can been seen that the new patches 1 to 6 can be determined effectively to be similar to the first patch. However, as shown in Figure 4.25(7), the shape similarity value between new patch 7 and patch 1 is a moderate outlier among the shape similarity values between new patch 7 to all the 16 patches. This shows that, although patch 7 can still be determined to be similar to patch 1, the confidence level decreases when compared with the other 6 new patches. The reason for this is that the sampling resolution of new patch 7 is very low compared to patch 1. Therefore, there is significant change in both the boundary and the shape of new patch 7. The curve of normalized correlation coefficients as a function of the sampling resolution is shown in Figure 4.26. Here, the sampling resolution is defined as the ratio between the number of vertices on each of the new patches and the number of vertices on surface patch 1. As the sampling resolution decreases, the normalized correlation coefficient decreases gracefully. This demonstrates the robustness of Harmonic Shape Images with respect to sampling resolution.

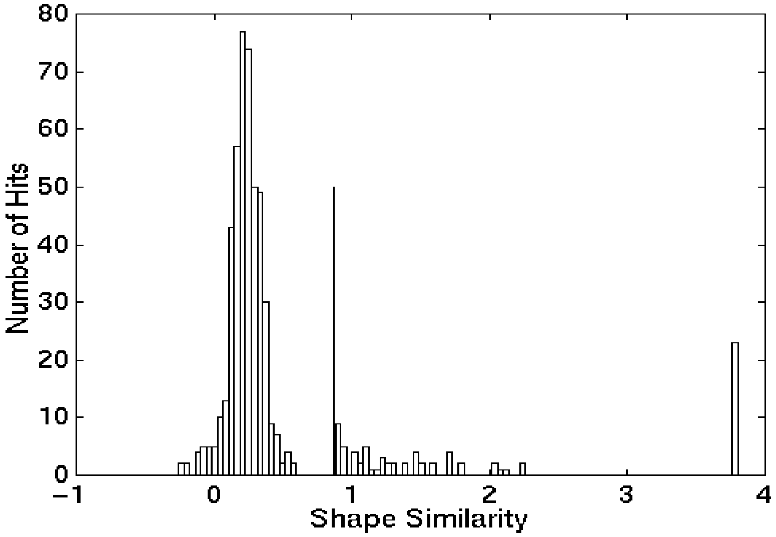
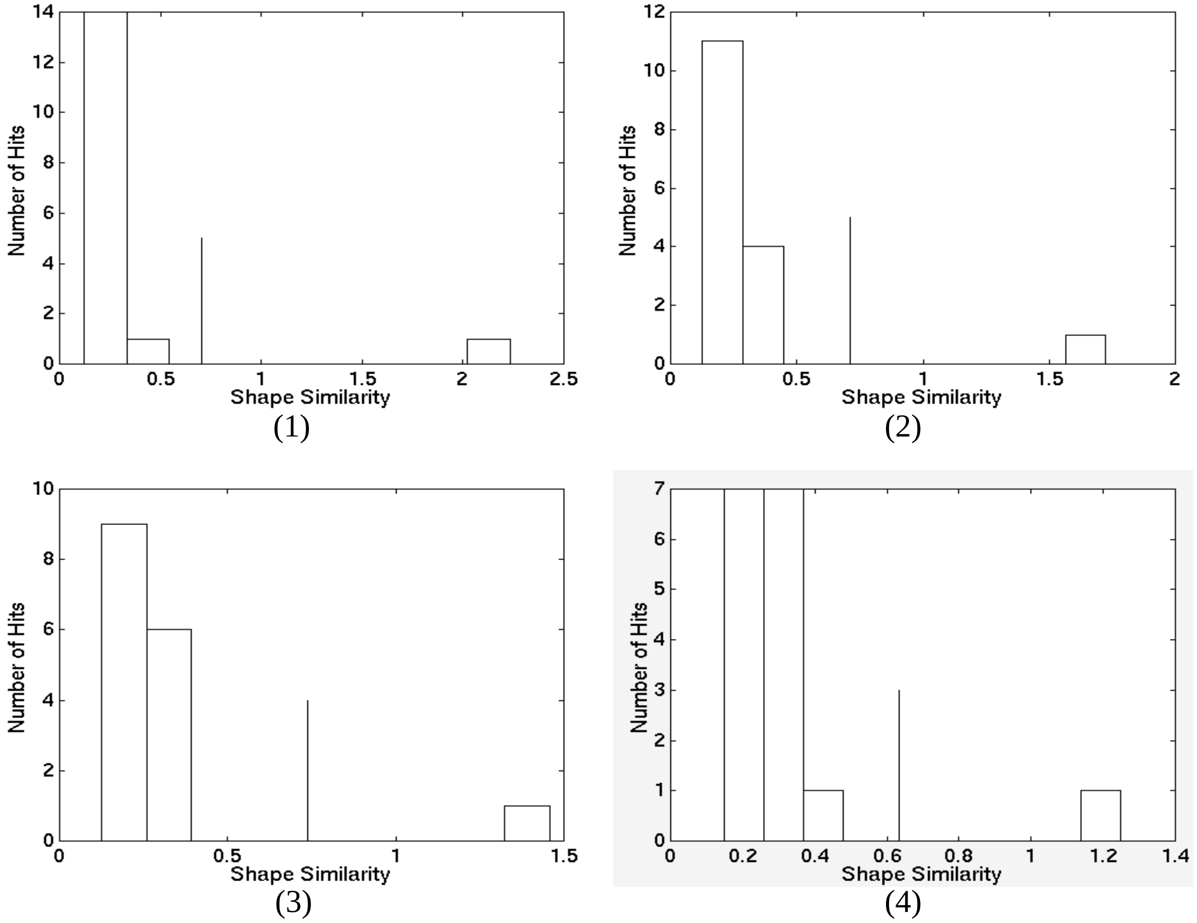


Figure 4.23: The histogram of shape similarity values for the pair comparison in the patch library with new patches of different resolutions added.

The second experiment is conducted in the same way as was the first experiment. Seven new patches of the same shape as that of patch 2 in the library, but of different sampling resolutions, are generated and added to the 16-patch library. The new patches and their Harmonic Shape Images are shown in Figure 4.27 and Figure 4.28. The pair comparison results are shown in Figure 4.28, Figure 4.29, Figure 4.30, Figure 4.31 and Figure 4.32.



(1)

(2)

(3)

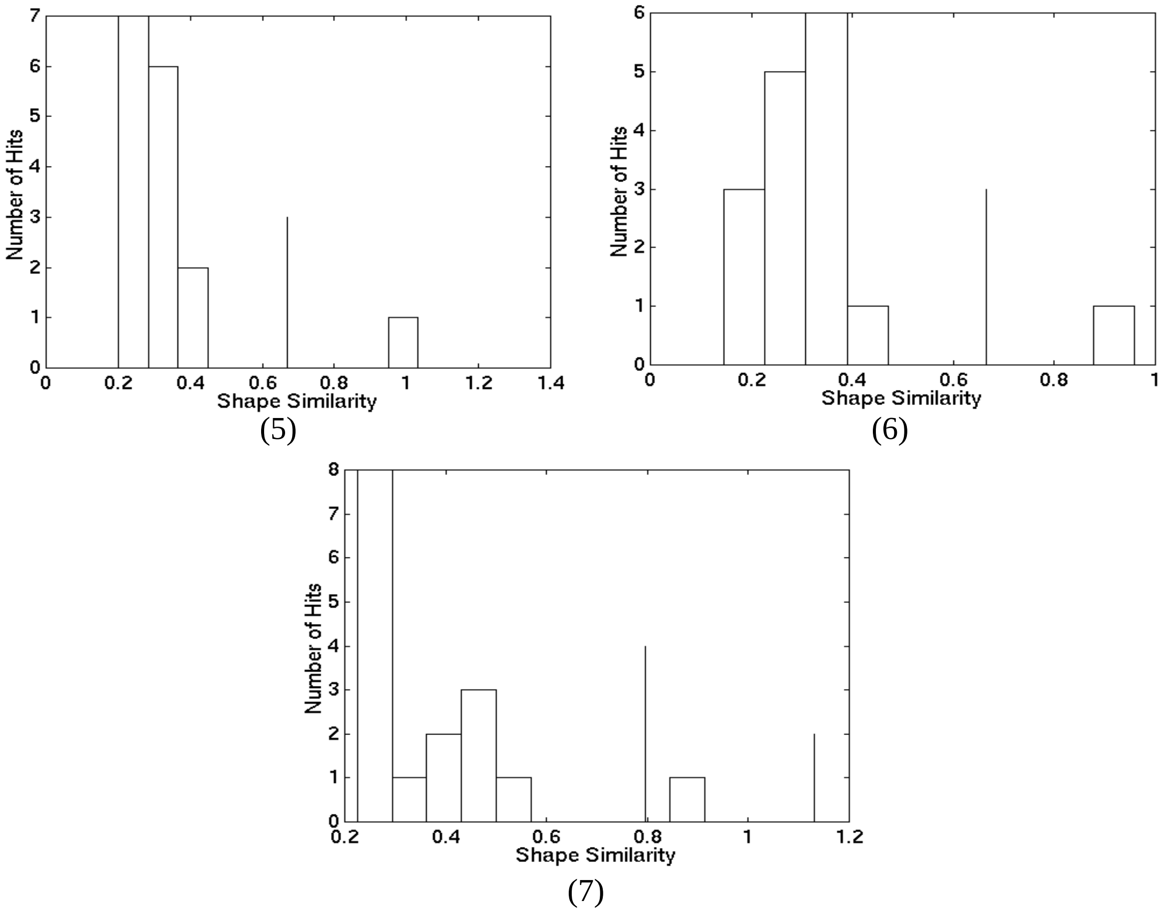
(4)

Figure 4.24: Histograms of the shape similarity values for the comparison between each of the new patches 1 to 4 to the 16 patches in the library.

4.4 Robustness to Occlusion

One important property for surface representation is its robustness to occlusion, i.e., correct matching result should still be obtained even when the surfaces being compared are not complete. In this section, we will first explain why Harmonic Shape Images are robust to occlusion and then, using real data, demonstrate their robustness.

The reason why occlusion is difficult to handle in surface matching is as follows. Suppose S_1 is a given surface and a shape representation *Rep*(S_1) is created. If there is occlusion on S_1, the non-occluded part of S_1 is denoted as S_2 and its representation is *Rep*(S_2). In order to represent the shape of the underlying surface, the relationship among the points on the surface must be used in some way to create a representation. This means that, when some points are missing due to occlusion, the resulted representation for other non-missing points will become different compared with the case that those missing points are present on the surface. Using the above notation, *Rep*(S_2) will not be a subset of *Rep*(S_1) in the presence of occlusion. Because of this inherent problem, for 3D free-form surfaces, it is impossible for a representation to



(5)

(6)

(7)

Figure 4.25: Histograms of the shape similarity values for the comparison between each of the new patches 5 to 7 to the 16 patches in the library.

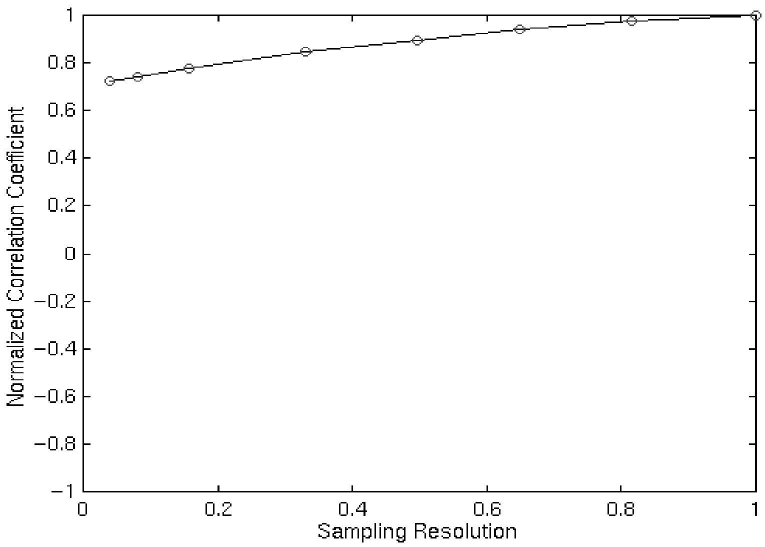


Figure 4.26: The normalized correlation coefficient as a function of the surface sampling resolution.

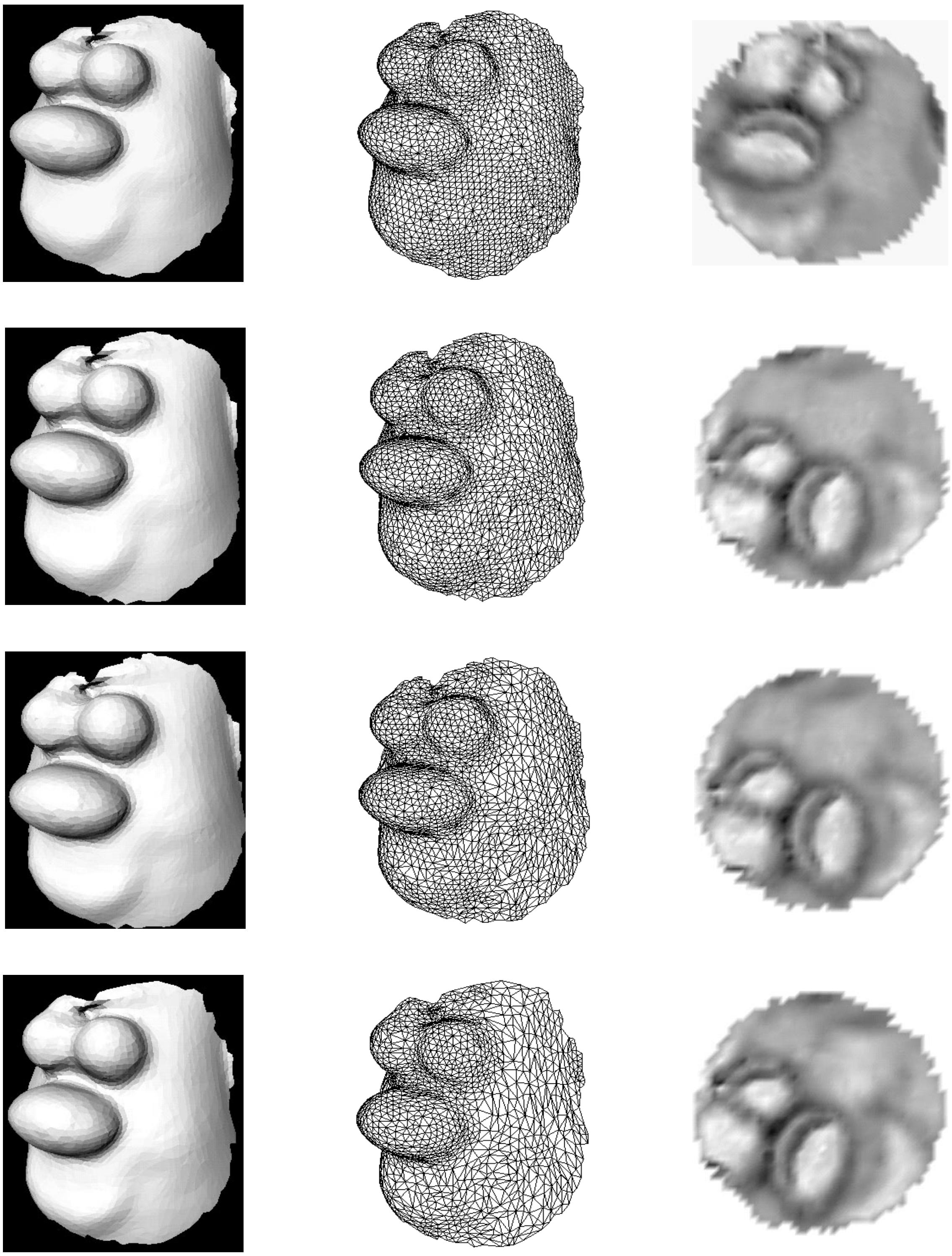


Figure 4.27: Surface patches of the same shape but of different resolutions. Left column: rendered surfaces; middle column: surfaces in wireframe; right column: Harmonic Shape Images: From top to bottom, the number of vertices are 2283, 2018, 1780 and 1451.

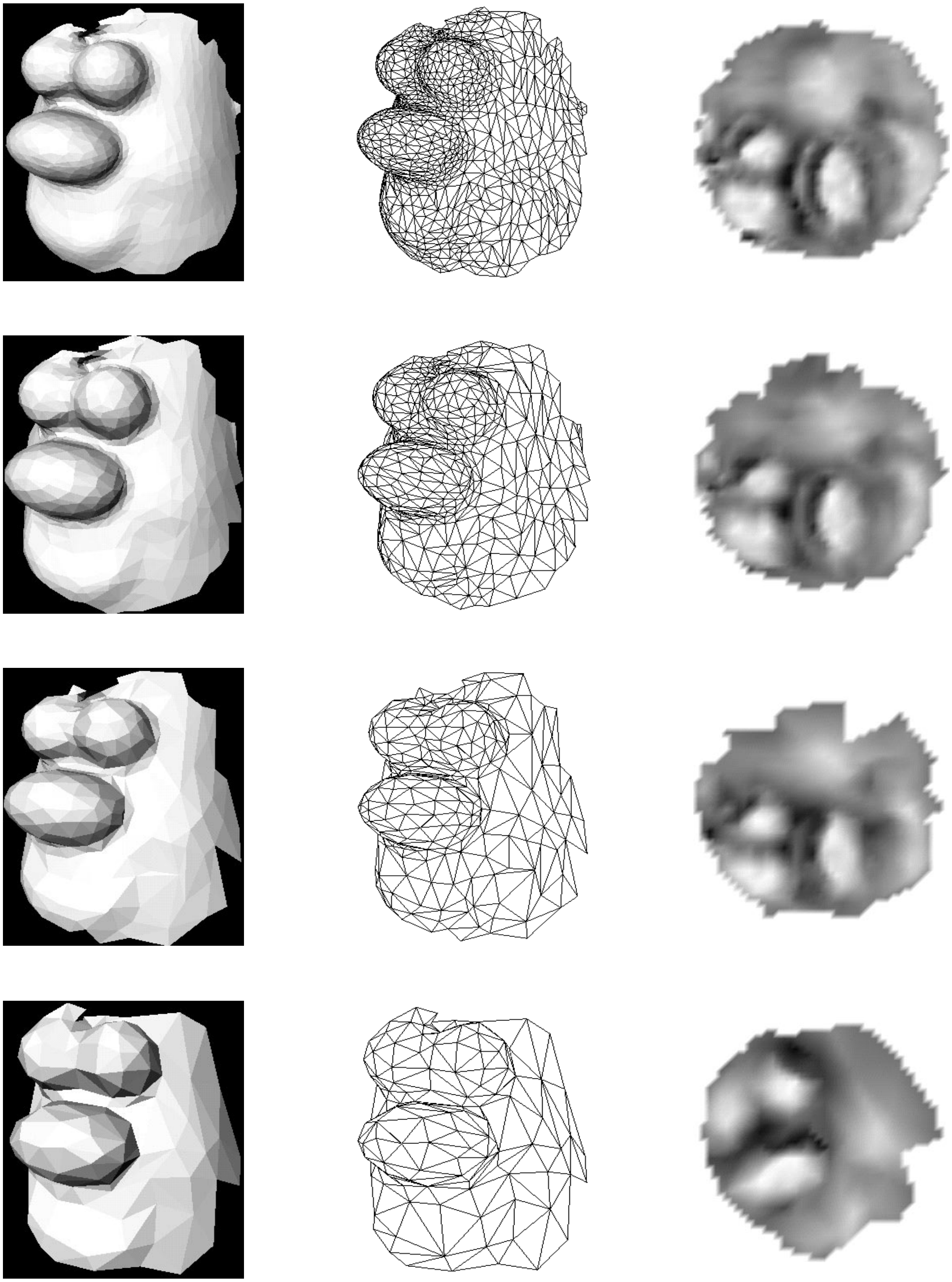


Figure 4.28: Surface patches of the same shape but of different resolutions. Left column: rendered surfaces; middle column: surfaces in wireframe; right column: Harmonic Shape Images: From top to bottom, the number of vertices are 960, 491, 243 and 122.

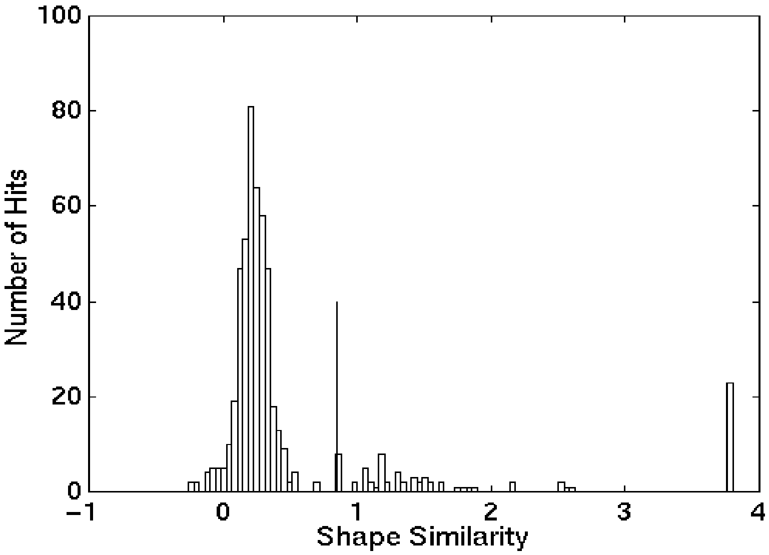
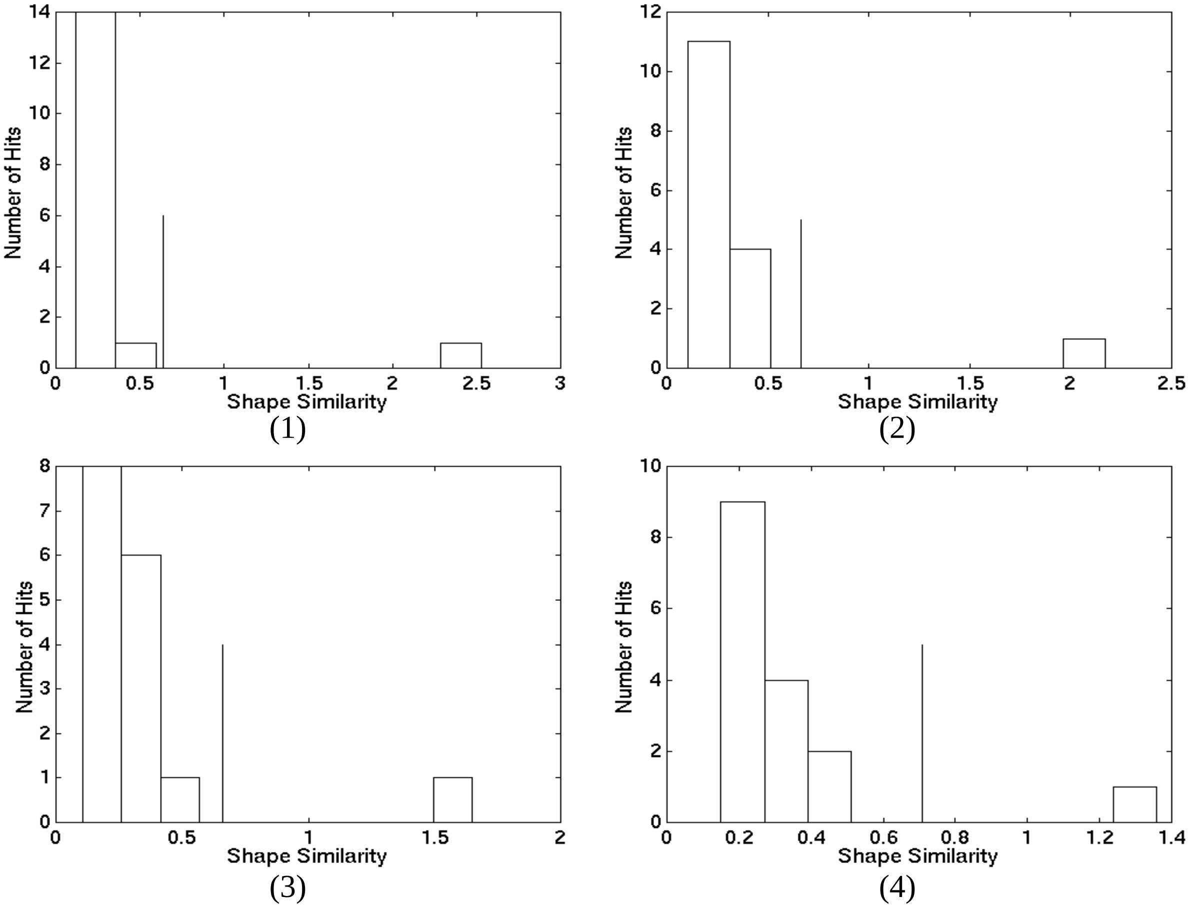


Figure 4.29: The histogram of the shape similarity values for the pair comparison in the patch library with new patches of different resolutions added.



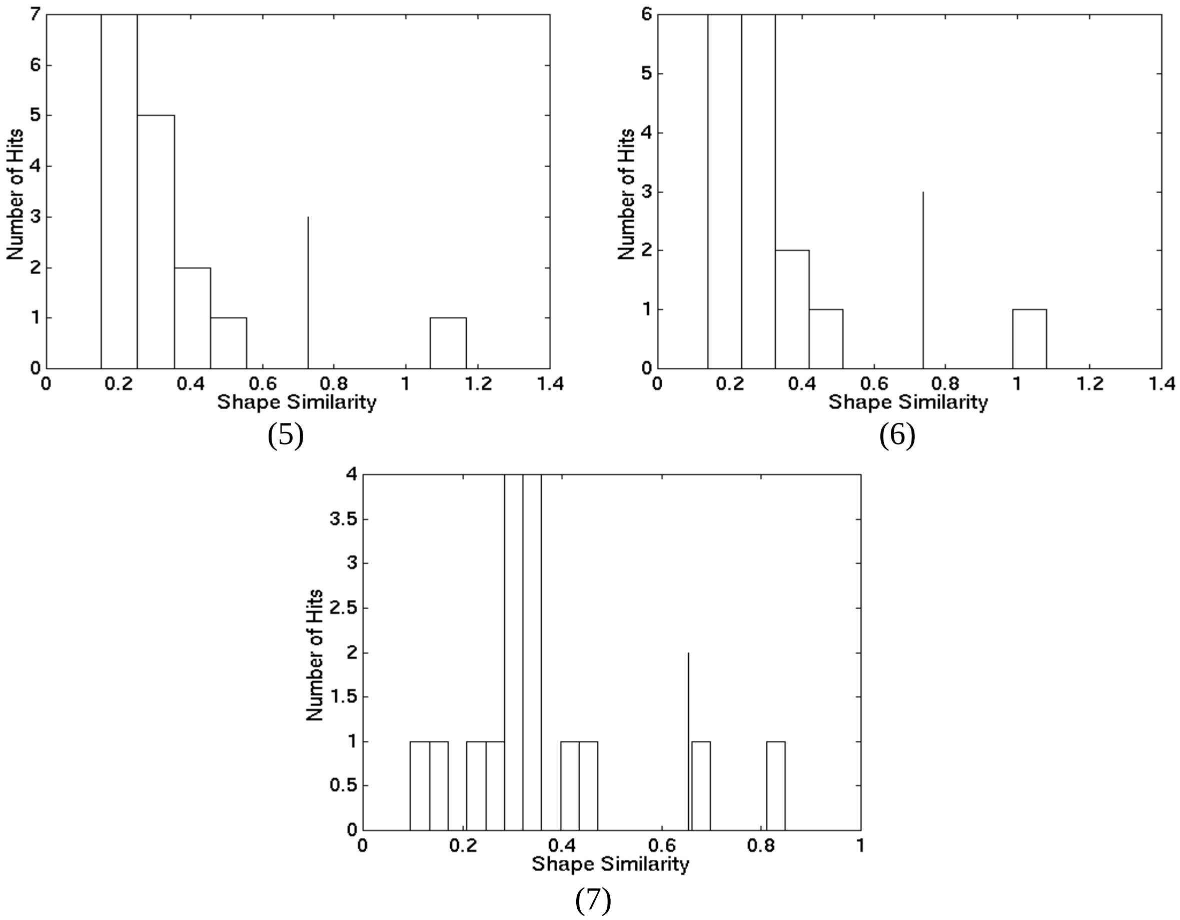
(1)

(2)

(3)

(4)

Figure 4.30: Histograms of the shape similarity values for the comparison between each of the new patches 1 to 4 to the 16 patches in the library.



(5)

(6)

(7)

Figure 4.31: Histograms of the shape similarity values for the comparison between each of the new patches 5 to 7 to the 16 patches in the library.

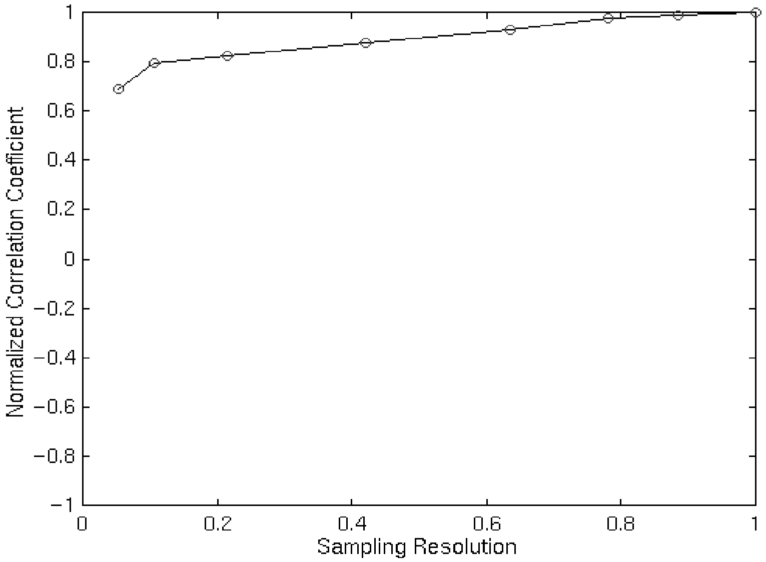


Figure 4.32: The normalized correlation coefficient as a function of surface sampling resolution.

remain exactly the same under occlusion. The best thing that can be done is to make the change in representation gradual as occlusion increases. Therefore, the matching performed using the representation degrades gracefully without any sudden failure due to a small amount of occlusion. This characteristic is called the robustness with respect to occlusion.

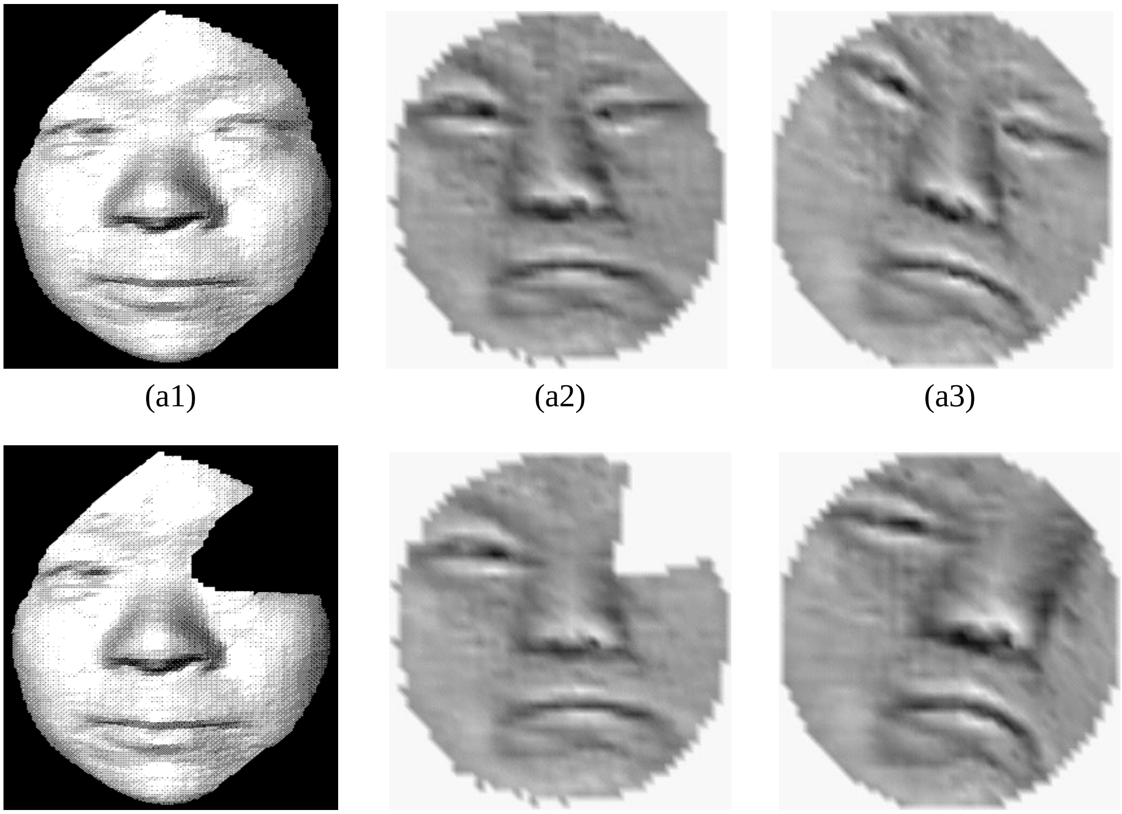
The reason for the robustness of Harmonic Shape Images with respect to occlusion lies in the way in which the boundary mapping is constructed. Recall that the construction of harmonic maps consists of two steps: boundary mapping and interior mapping. The boundary mapping is used as a constraint for computing the interior mapping. Therefore, the change in boundary mapping affects the resultant harmonic image.

In order to handle occlusion that may occur on the boundary of the underlying surface patches, special attention is paid when constructing the boundary mapping. Recall from Section 2.4 that the boundary vertices are classified into radius boundary vertices and occlusion boundary vertices. It is the radius boundary vertices that determine the angles a_i, which then determine the overall boundary mapping. Based on this, the occlusion boundary vertices are then mapped to the interior of the unit disc instead of to its boundary according to the ratio of their surface distances to the central vertex and the radius of the surface patch (this ratio is referred as distance ratio for the rest of the section). It is important to map the occlusion boundary vertices onto the interior of the unit disc because those vertices are, in fact, interior vertices of the surface patch when there is no occlusion. If, instead, those occlusion vertices are mapped onto the boundary of the unit disc, there will be severe distortion in the boundary mapping, which in turn, will result in the distortion on the interior mapping.

By mapping the occlusion boundary vertices onto the interior of the unit disc using the distance ratio, the effect of occlusion is limited within the occlusion range; therefore, it does not propagate much outside of the occlusion range. This means that, as long as there are enough radius boundary vertices present in the surface patch, the overall harmonic image will remain approximately the same in spite of the occlusion. Figure 4.33 is an example to illustrate how the boundary mapping handles occlusion.

In Figure 4.33, a complete surface patch is shown in (a1). Its Harmonic Shape Images generated with and without using the distance ratio are shown in (a2) and (a3), respectively. Since there is no occlusion, there is little difference between those two images. The surface patch shown in Figure 4.33(b1) is the patch in (a1) with occlusion. It can be seen that, using the distance ratio, its Harmonic Shape Image (b2) is very similar to the non-occluded one(a2). In addition, the occlusion is present in the Harmonic Shape Image as well. In contrast, without using the distance ratio, the occlusion cannot be represented in the Harmonic Shape Image(b3). Compared with the non-occluded version in (a3), the distortion is significant.

In order to verify the robustness of Harmonic Shape Images, the following experiment is conducted. Among the surface patches in the patch library shown in Figure 4.5 and Figure 4.6, the first patch (Figure 4.5(1)) is selected. Different parts of that patch are removed manually to simulate occlusion, resulting in seven new patches. These new patches are added to the patch library. As before, pair comparison among all the patches in the library is conducted. The expected result is that the patches with occlusion should be identified as having the same shape as that of the first patch. Figure 4.34 and Figure 4.35 show the patches with occlusion and their Harmonic Shape



(a1) (a2) (a3)

(b1) (b2) (b3)

Figure 4.33: Illustration of how to handle occlusion using the boundary mapping. (a1) A complete surface patch; (a2), (a3) Harmonic Shape Images of (a1) obtained with/without using the distance ratio; (b1) the surface patch in (a1) with occlusion; (b2), (b3) harmonic Images of (b1) obtained with/without using the distance ratio.

Images. The results of pair comparison are shown in Figure 4.36. Figure 4.36(a) shows that, in addition to the high correlation values on the diagonal, the correlation values for patches 17 to 23 compared with the first patch are also high. The patches numbered from 17 to 23 are the patches with occlusion. The correlation values for the pair comparison among the occlusion patches are high as well. This result shows that the Harmonic Shape Images of the patches with occlusion still remain similar to the Harmonic Shape Image of the first patch. Figure 4.36(b) shows the matrix of the shape similarity values for the pair comparison among all the patches in the library. The matrix has a similar distribution to that in Figure 4.36(a). Figure 4.36 shows the change of the normalized correlation coefficient as the boundary occlusion increases. As discussed above, the more occlusion there is on the boundary, the more change there will be on Harmonic Shape Images. The curve in Figure 4.36 verifies this observation. It should be noted that, in this example, the correlation coefficient still remains very high when the percentage of boundary occlusion increases to 50%. The reason for this is that the most significant features in the surface patch remain almost intact in all the occlusion patches, and there is small variation in shape in other parts of the surface patch. Occlusion occurs mainly in those parts of the patch with small shape variation. Figure 4.38 and Figure 4.39 show the histogram of each occlusion patch compared with the 16 patches in the library. In all the 7 cases, the first patch is identified as the qualified match.

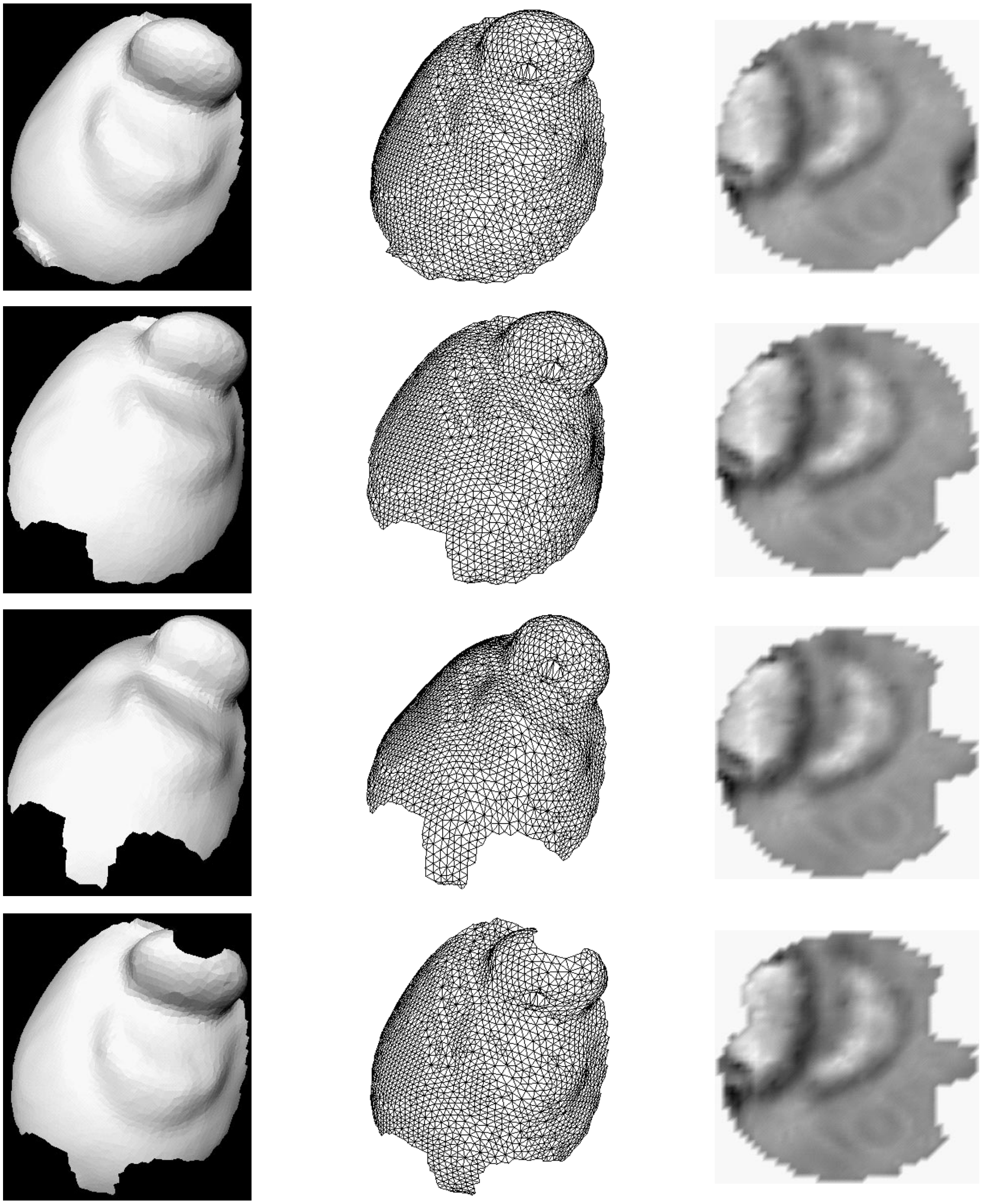


Figure 4.34: Patches with occlusion and their Harmonic Shape Images.

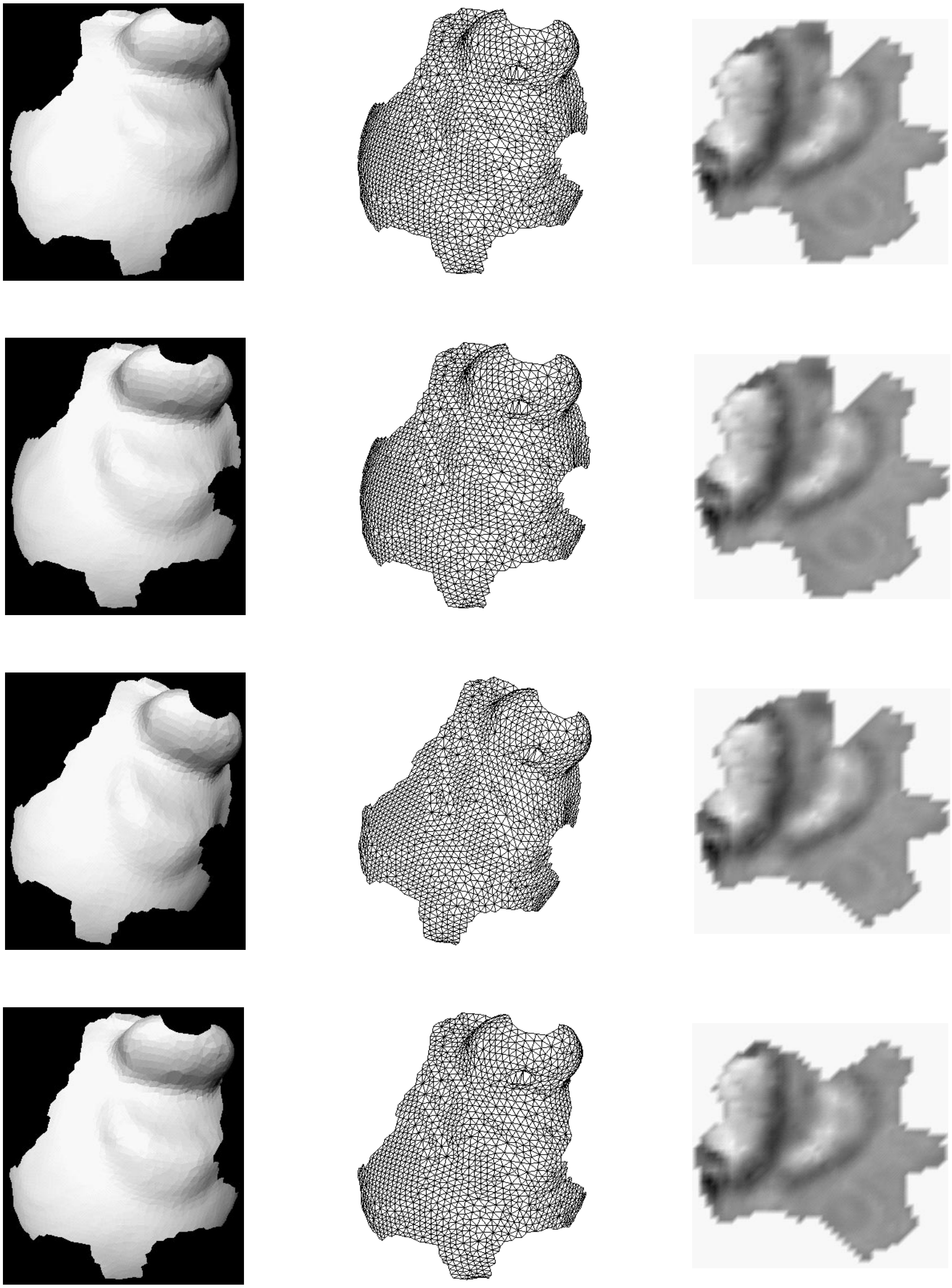
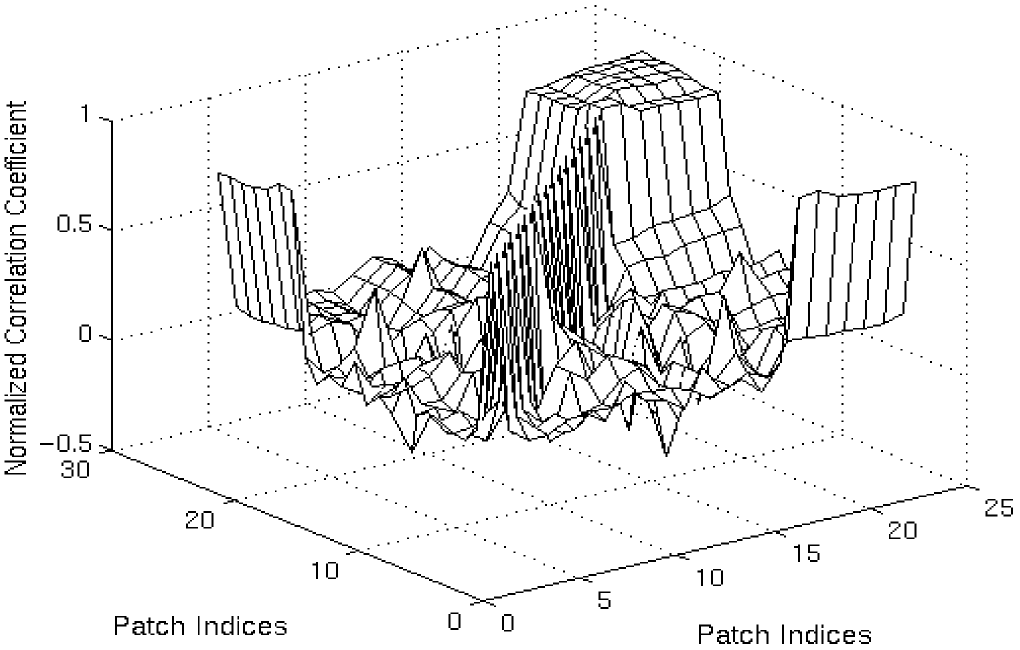
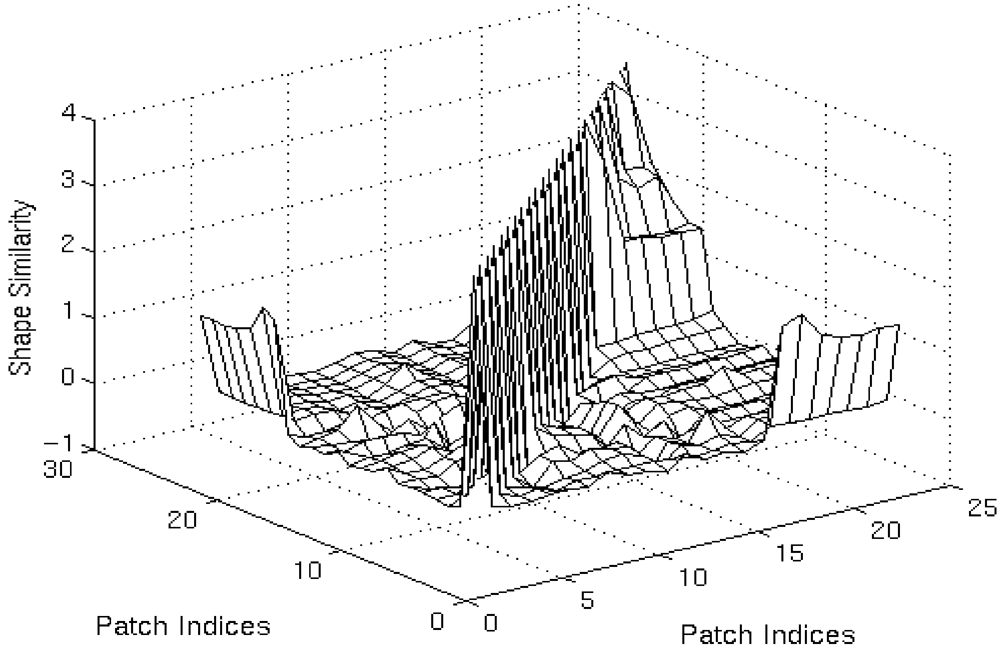


Figure 4.35: Patches with occlusion and their Harmonic Shape Images.



(a)



(b)

Figure 4.36: The results of pair comparison in the library with occlusion patches. (a) The matrix of the normalized correlation coefficients; (b) the matrix of the shape similarity values.

In the second example, the second surface patch in the library Figure 4.5(2) is selected. Seven new occlusion patches are generated by removing different parts of that patch(Figure 4.40, Figure 4.41). These new patches are added to the library. Now the number of patches in the library is 23. Similarly to the first experiment, pair comparison among the 23 patches in the library is conducted and the results are shown in Figure 4.42. The results are similar to those in the first experiment. Figure 4.43 shows the normalized correlation coefficient as a function of the boundary occlusion. The normalized correlation coefficient remains fairly stable as the percentage of boundary occlusion increases to 60%. The reason why the absolute value of the correlation coefficient is a slightly lower than that in the first experiment is that parts of the most significant features of the patch are occluded; therefore, the result is more corruption in the Harmonic Shape Images,a natural occurance since, the more features are missing, the more difficult it is to recognize an object. The histograms of the shape similarity values for the occlusion patches compared with the 16 patches are shown in Figure 4.44 and Figure 4.45. In all 7 cases, the second patch is identified to be the qualified match.

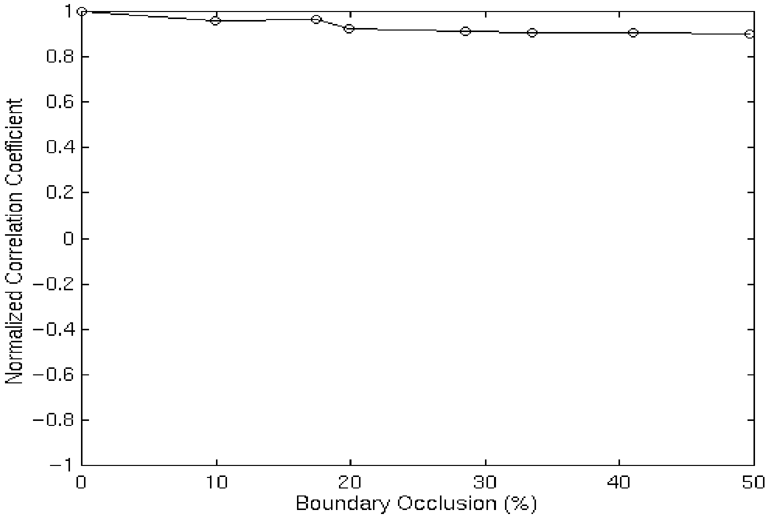
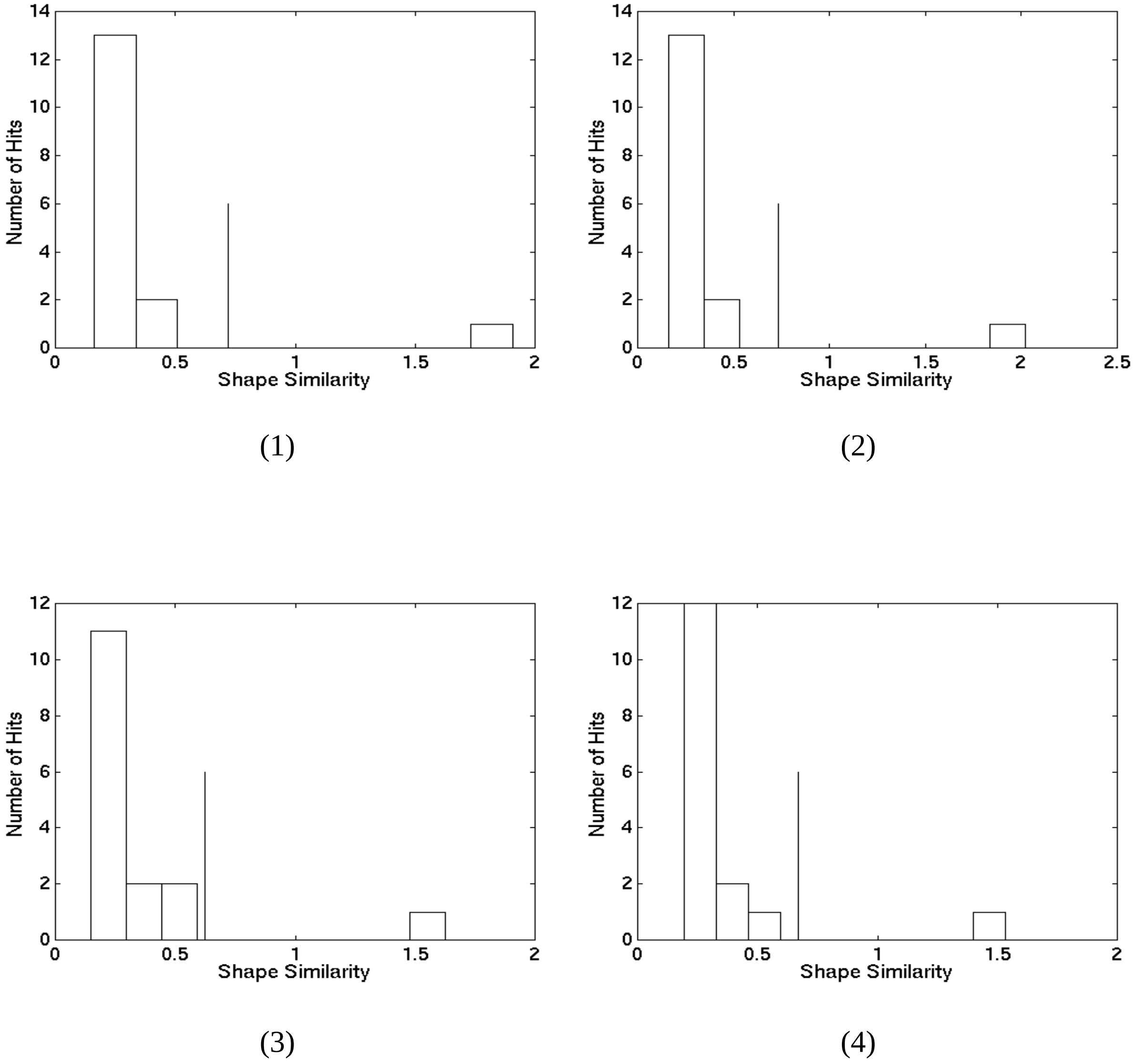


Figure 4.37: The curve of the normalized correlation coefficient as a function of the percentage of occlusion boundary. As the occlusion increases, the normalized correlation coefficient decreases gracefully.



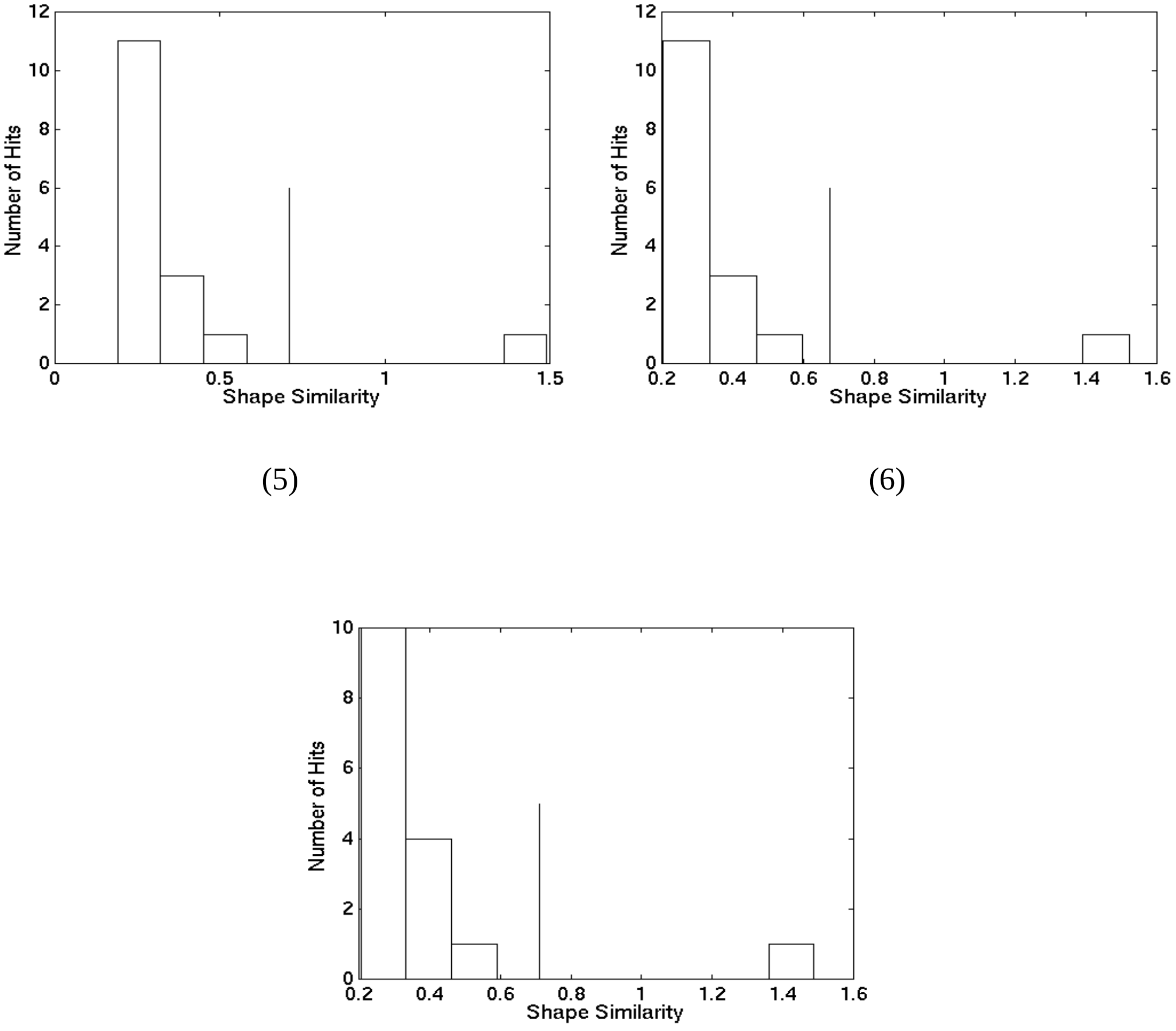
(1)

(2)

(3)

(4)

Figure 4.38: Histogram of the shape similarity values for the occlusion patches 1 through 4 compared with the 16 patches in the library.



(5)

(6)

(7)

Figure 4.39: Histogram of the shape similarity values for the occlusion patches 5 through 7 compared to the 16 patches in the library.

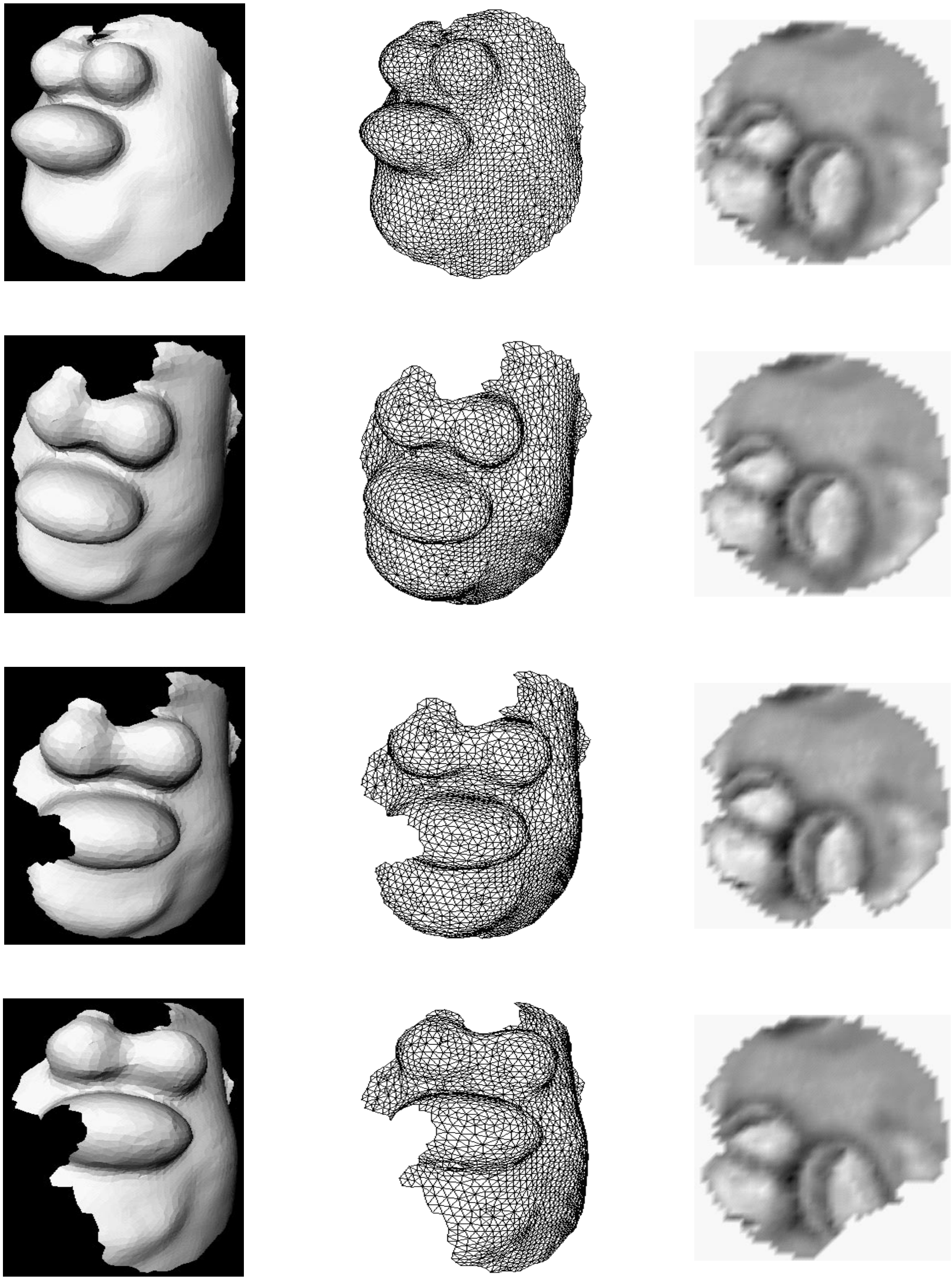


Figure 4.40: Occluded patches and their Harmonic Shape Images.

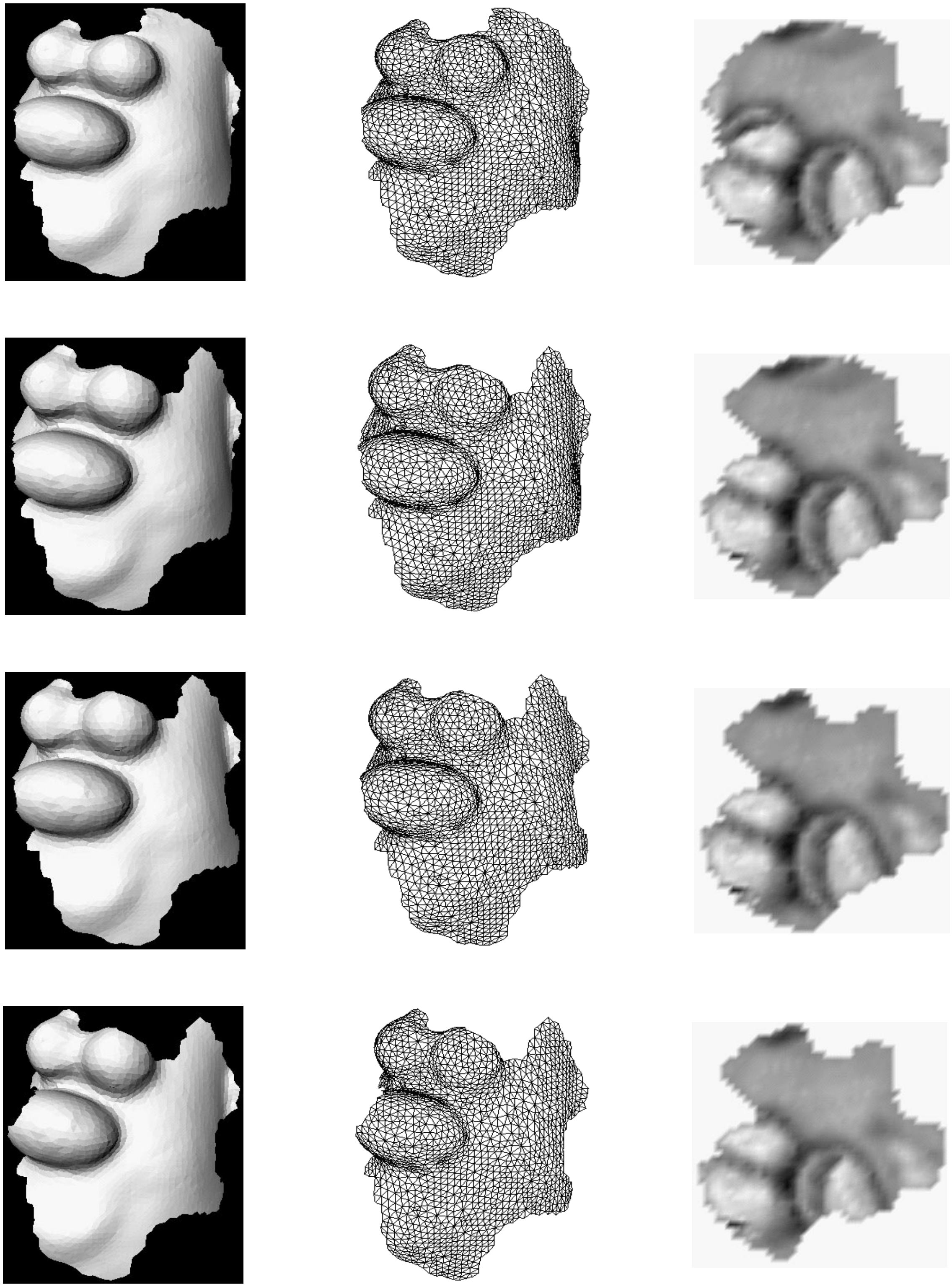
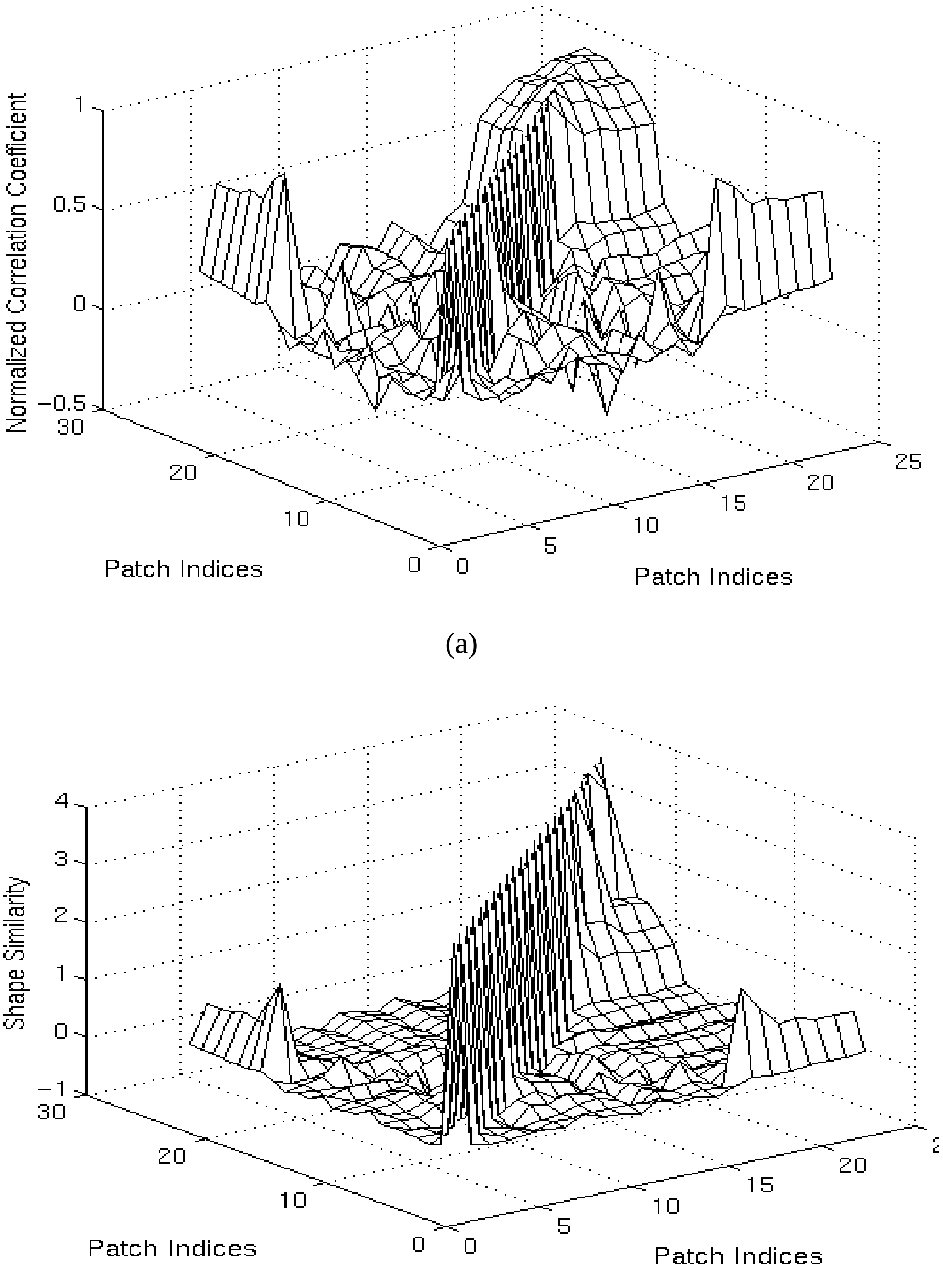


Figure 4.41: Occluded patches and their Harmonic Shape Images.



(a)

(b)

Figure 4.42: The results of pair comparison in the library with occlusion patches. (a) The matrix of the normalized correlation coefficients; (b) the matrix of the shape similarity values.

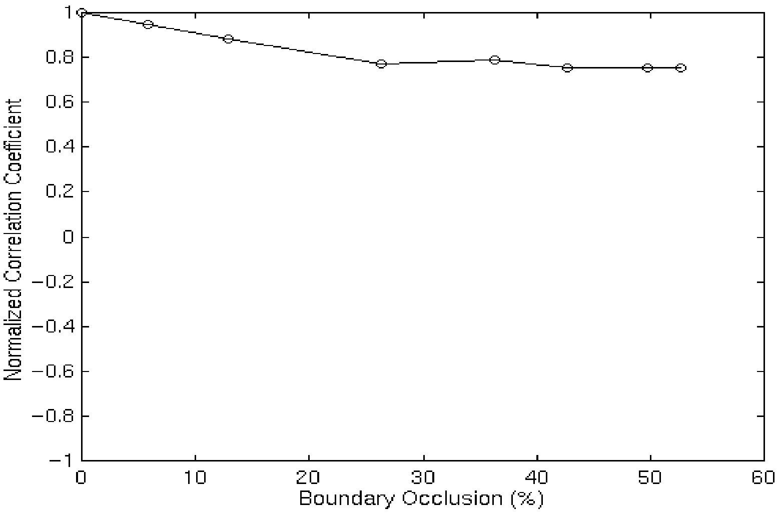


Figure 4.43: The curve of the normalized correlation coefficient as a function of the percentage of occlusion boundary. As the occlusion increases, the normalized correlation coefficient decreases gracefully.

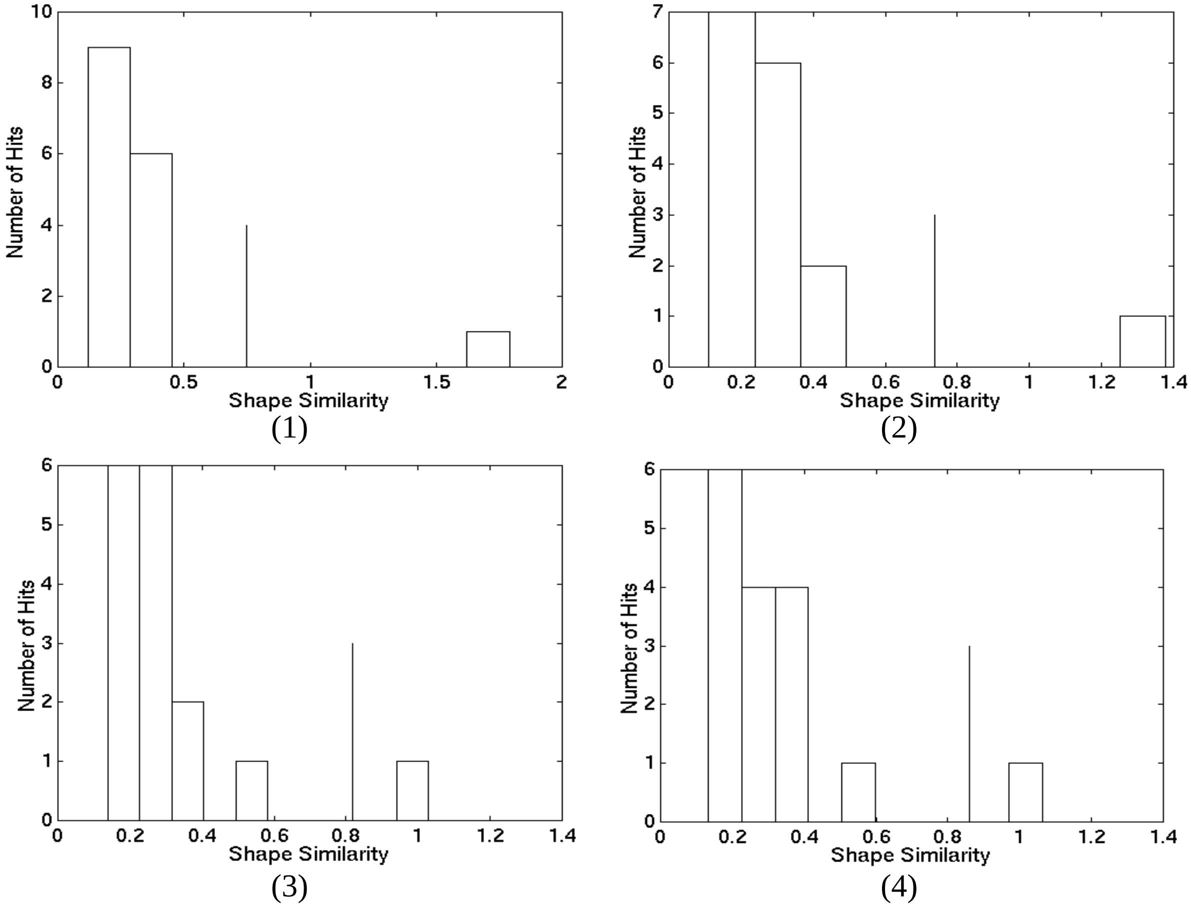
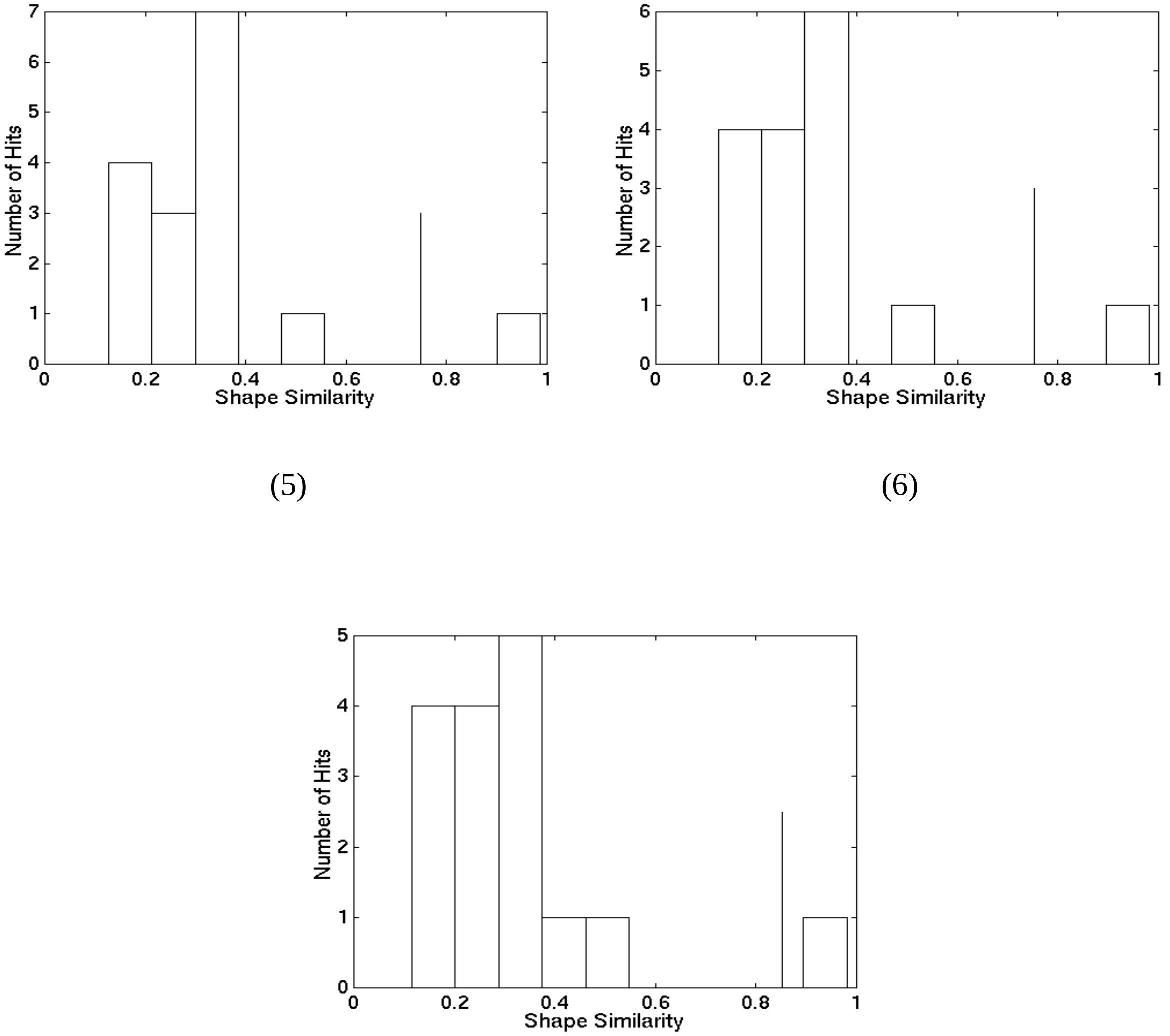


Figure 4.44: Histogram of the shape similarity values for the occlusion patches 1 through 4 compared with the 16 patches in the library.



(5)

(6)

(7)

Figure 4.45: Histogram of the shape similarity values for the occlusion patches 5 through 7 compared to the 16 patches in the library.

Chapter 5

Enhancements of Harmonic Shape Images

The generation, matching and experimental analysis of Harmonic Shape Images have been discussed in previous chapters. In this chapter, some enhancements of Harmonic Shape Images will be discussed; these include dealing with missing data and improving the matching result of Harmonic Shape Images in the presence of noise.

5.1 Dealing with Missing Data

We have seen in Chapter 2 that, in order to compute the Harmonic Shape Image of a given surface patch, that patch needs to be a connected region without holes. In practice, this topology requirement may not be satisfied because of the topology of the underlying surfaces, the occlusion caused by other surfaces in the scene or missing data due to sensor errors. In other words, some surface patches may have holes. In order to compute Harmonic Shape Images for those surfaces, an interpolation strategy is proposed and then explained in detail in this section. In the following, the effects of missing data on harmonic images is discussed first followed by the explanation of how to triangulate a surface patch that has missing data. Then Harmonic Shape Images obtained using the triangulation strategy are presented.

Effects of missing data on harmonic images

The idea of the interpolation strategy is to triangulate the holes resulting from missing data. Before presenting the interpolation algorithm, let us look at how missing data affect the computation of harmonic image for a given surface patch. This also explains why such an interpolation algorithm is necessary.

Multiple Boundaries: Two surface patches of the same shape are shown in Figure 5.1. The difference between the two surface patches is that the one in Figure 5.1(a) does not have holes while the one in Figure 5.1(b) does. Recall the procedure of computing the harmonic image for a given surface patch. The procedure consists of the following steps:

- Locate the boundary vertices of the surface patch
- Construct the boundary mapping
- Compute the spring constant for each interior edge of the surface patch
- Construct the interior mapping

Since we have made the assumption about the surface patches to be orientable two-dimensional manifold, the boundary vertices of the surface patch in Figure 5.1(a) can be easily detected by checking the number of triangles with which every edge is associated. Those boundary vertices are then re-ordered to construct a closed boundary for the surface patch. However, if a surface patch has holes (Figure 5.1(b)), then an extra step needs to be taken in order to determine which closed boundary is the outside boundary of the surface patch. This is the first effect of missing data on the computation of harmonic images.

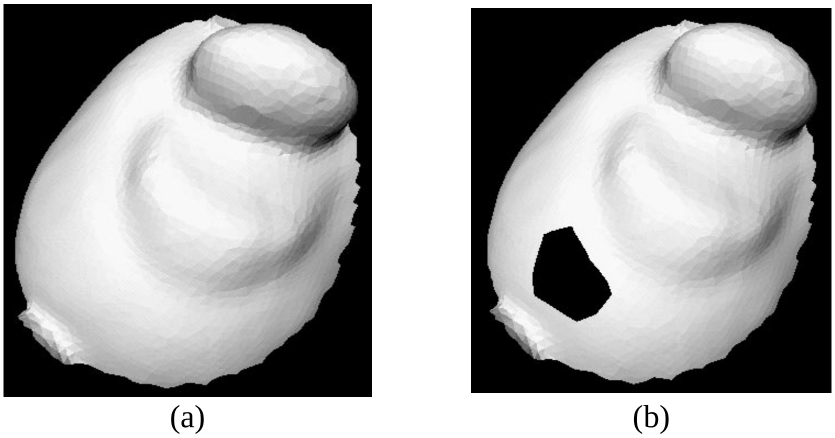


Figure 5.1: Two surface patches of the same shape with (a) and without (b) holes.

Under the assumption that the holes are relatively small with respect to the entire surface patch and the boundaries of the holes are simple curves in the three-dimensional space, i.e., there are no twists, the outside boundary of a surface patch with holes can be determined by checking the shortest path from the central vertex to each of the boundary vertices. In general, the inside boundaries block the shortest paths between the central vertex and some of the outside boundary vertices; the opposite is not true, however. Therefore, none of the inside boundary vertices has the outside boundary vertices on its shortest path. Based on this fact, the outside boundary is the one whose vertices have no other boundary vertices on its shortest path to the central vertex.

Change of Spring Constants: Once the outside boundary has been determined, the boundary mapping can be computed. Using the boundary mapping as a constraint, the interior vertices are mapped onto the interior of the 2D target domain by minimizing the energy of a spring system. This spring system is constructed by associating a spring with each interior edge of the surface patch. Computing the spring constant of each interior edge is an important step in constructing the spring system. Recall from previous sections that the spring constant k_{ij} of an edge e_{ij} is defined as

$$k_{ij} = ctg\theta(e_{mi},e_{mj}) + ctg\theta(e_{li},e_{lj}) \tag{5.1}$$

in which the angles θ are defined in Figure 5.2:

According to (5.1), the spring constant of an interior edge has two terms because the edge is associated with two triangles. When the surface patch has holes, the spring constants of the interior edges that are on the hole boundaries have only one term. Although the spring constants of those edges are different from that without holes, the construction of the energy functional of the interior mapping is not affected. However, as can be seen from Figure 5.3, the resultant harmonic image of the interior vertices is affected.

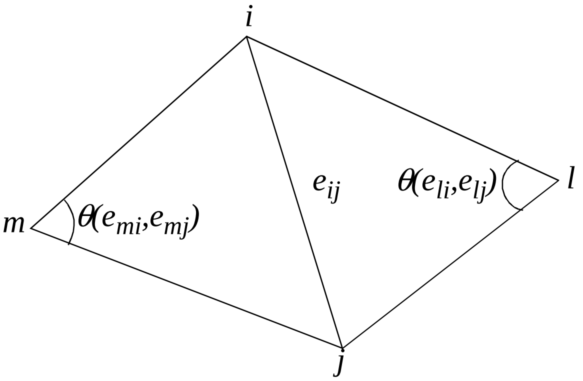


Figure 5.2: Illustration of the definition of spring constant for an edge.

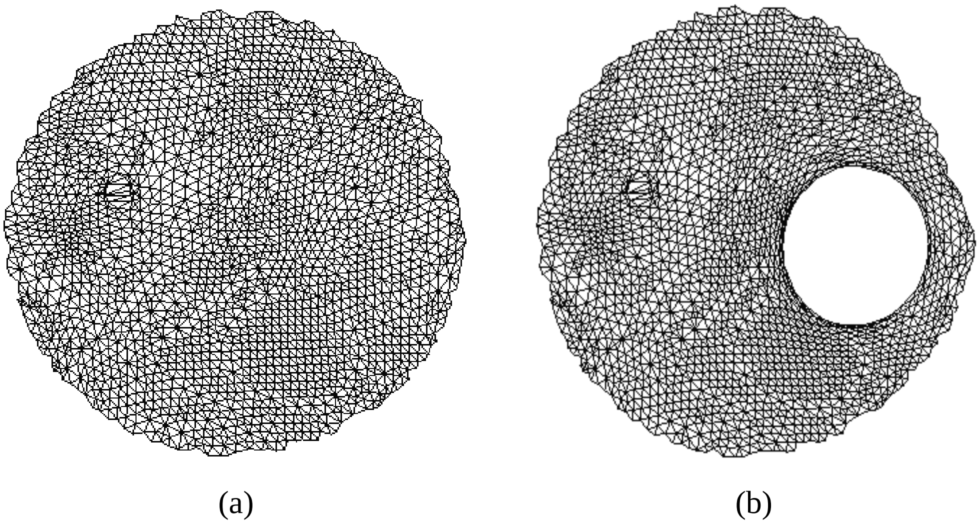


Figure 5.3: Comparison of the harmonic image of the same surface patch with (a) and without (b) holes.

Area Percentage and Boundary Shape of Hole: Figure 5.3(a) and (b) are the harmonic images of the surface patches shown in Figure 5.1(a) and (b), respectively. Under the harmonic mapping, the image of the hole boundary in Figure 5.1(b) is a circular curve in Figure 5.3(b) which means that, the shape of the hole boundary is changed in the harmonic image. The ratio of hole area to the area of the surface patch changes as well. The area percentage of the hole is higher in Figure 5.3(b) than that in Figure 5.1(b). The changes in the shape of the hole boundary and in the area percentage of the hole are the effects of holes on the computation of harmonic images. Because there are no surface points constraining the hole boundary from inside the hole, the boundary vertices behave relatively freely during the energy-minimization process and end up being circular in order to achieve the energy minimum.

Because of the above changes in the harmonic image, the positions of the vertices that are close to the hole boundaries also change in the harmonic image. This can be seen from the overlaid image of Figure 5.3(a) and (b) shown in Figure 5.4; the vertices marked by circles and boxes are the same vertices on the surface patches shown in Figure 5.1(a) and (b). The black circles indicate the positions of those vertices on the harmonic image when there is no hole in the surface patch. In contrast, the black boxes indicate the positions of the same vertices on the harmonic image when there is a hole in the surface patch. Figure 5.4 shows that for the vertices that are close to the hole boundary, there is significant difference between their positions on the two harmonic images. This difference gradually decreases as the vertices are further and further away from the hole boundary.

Positions of Boundary Vertices When comparing the boundary images in Figure 5.3(a) and (b), we notice that the two boundary images differ in those vertices that have a hole boundary on their shortest paths to the central vertex. Recall that the position of each boundary vertex in the harmonic image is determined by the angle between two consecutive radius vectors and the length of the shortest path between that vertex and the central vertex. While hole boundaries do not change the angle between two radius vectors, they do change the lengths of the shortest paths between the central vertex and some of the outside boundary vertices. This is another effect that holes have on the computation of harmonic images.

The goal of the strategy for handling missing data is to minimize the above effects on harmonic images resulting from missing data. The strategy is discussed in detail in the following section.

Triangulation of The Hole

Since there are no data in the hole area, the most direct solution is to fit a plane to the hole and triangulate it. The complete hole interpolation algorithm consists of the following steps:

- Fit a plane to the hole
- Project the vertices on the hole boundary onto the plane
- Triangulate the polygon on the plane formed by the vertices on the hole boundary
- Project the vertices in the planar triangulation to the 3D space
- Update the surface patch

The first step is straightforward. Steps two to five are described in detail in the following.

Projection of Hole Boundary Vertices Denote the plane fit to the hole as *P*. Figure 5.5 illustrates how to project the vertices on the hole boundary onto the fitting plane *P*. In Figure 5.5, v_i is a vertex on the hole boundary and v_i' is its projection on the best-fit plane *P*. *N* is the normal of *P* and v_{mean} is defined in (5.3). v_i' can be computed using (5.2) and (5.3). An example of hole boundary and its projection onto the fit plane is shown in Figure 5.6.

$$v_i' = v_i - v_{mean} - \alpha N \tag{5.2}$$

$$\alpha = (v_i - v_{mean}) \cdot N, v_{mean} = \frac{1}{n} \sum_{i=0}^{h} h_i \tag{5.3}$$

The fit plane is then rotated so that it is parallel to the X-Y plane. The result is that, the z values of the vertices on the hole boundary become zero. This transformation can be expressed in (5.4), in which *R* represents the rotation matrix from the fit plane to the X-Y plane.

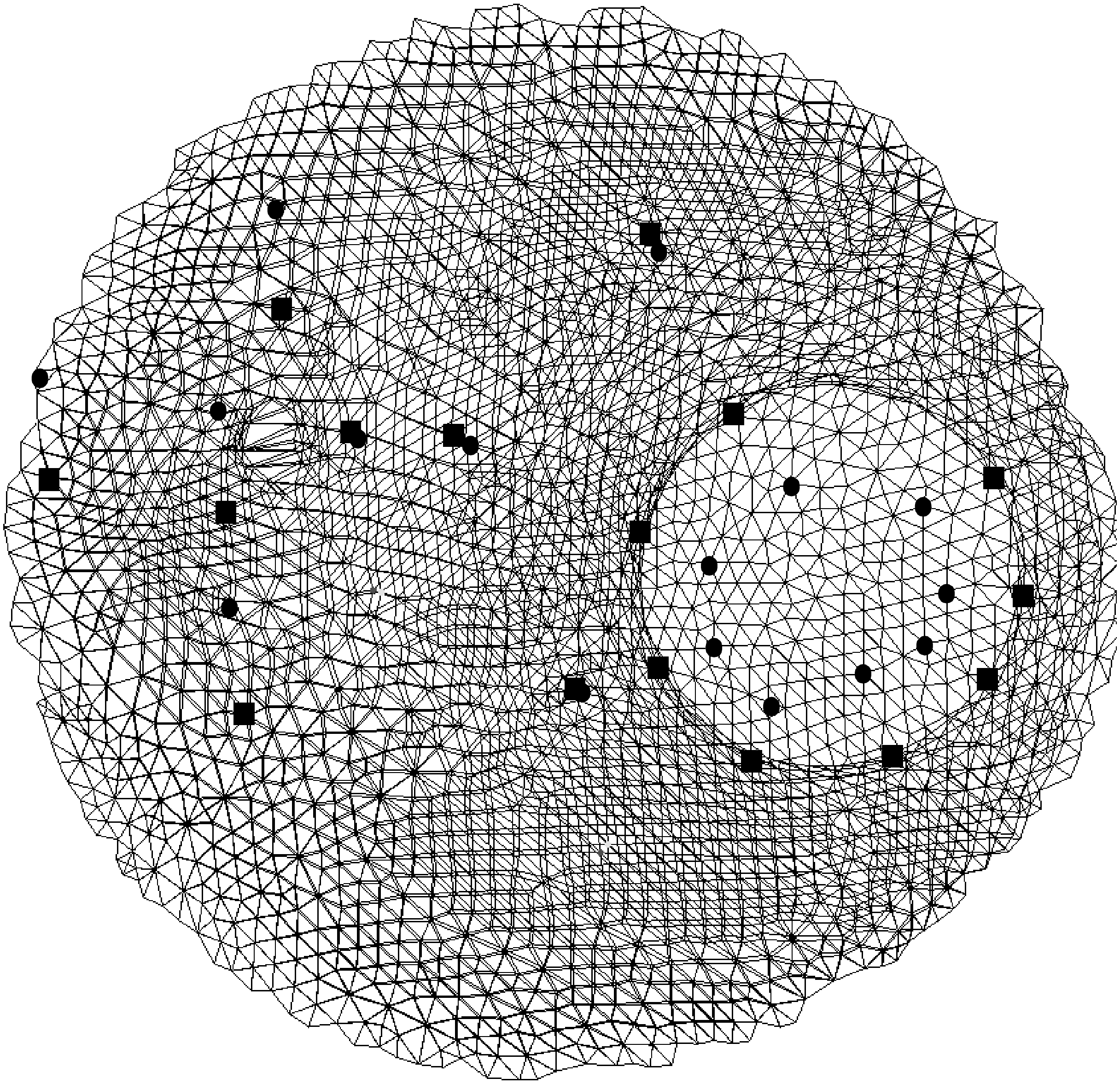


Figure 5.4: Comparison of the harmonic images of the surface patches shown in Figure 5.1(a) and (b). The black circles indicate the positions of the vertices on the harmonic image when there is no hole in the surface patch. The black boxes indicate the positions of the same vertices when there is a hole.

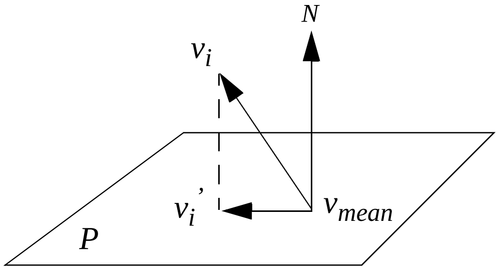


Figure 5.5: Projection of vertices on the hole boundary onto the best-fit plane.

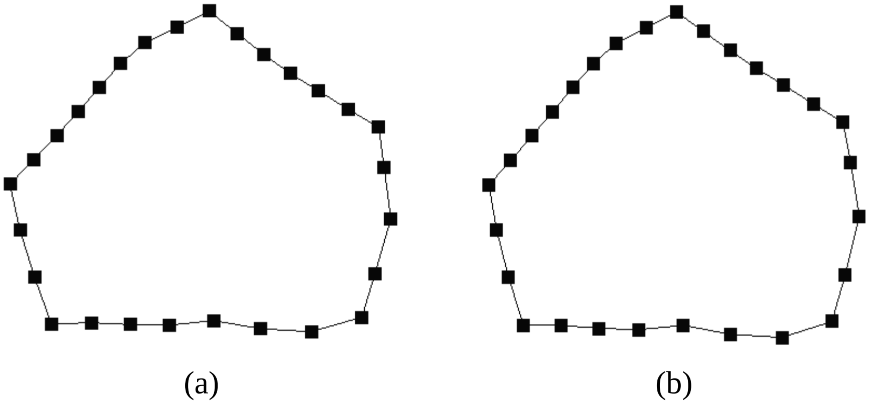


Figure 5.6: An example of hole boundary (a) and its projection (b) onto the best-fit plane.

$$v_i'' = Rv_i' \tag{5.4}$$

Planar Triangulation After the vertices on the hole boundary are projected onto the fit plane, a program called Triangle is used to triangulate those vertices.

The program Triangle is a C program for two-dimensional mesh generation and construction of Delaunay triangulations, constrained Delaunay triangulations and Voronoï diagrams. It is developed by J. Shewchuk at Carnegie Mellon University[70]. Figure 5.7 shows an application example of the Triangle program. A Planar Straight Line Graph(PSLG) is shown in Figure 5.7(a). Triangulating such a PSLG requires that the line segments in the original PSLG be the same line segments in the final triangulation. In fact, this is the same requirement in our case when triangulating the holes because the hole boundary which consists of a sequence of straight line segments actually constructs a PSLG. Using the *Constrained Delaunay triangulation* function in the Triangle program, the PSLG in Figure 5.7(a) is triangulated as shown in Figure 5.7(b).

One issue with performing a constrained Delaunay triangulation is that some skinny triangles with large aspect ratios may appear in the triangulation(Figure 5.7(b)). In order to address this issue, Triangle has another function called *conforming Delaunay triangulation*. This triangulation of a PSLG is a true Delaunay triangulation in which each PSLG segment may have been subdivided into several edges by the insertion of additional points, called Steiner points. These points are necessary to allow the segments to exist in the mesh while maintaining the Delaunay property. Steiner points are also inserted to meet constraints on the minimum angle and maximum triangle area. The second function of Steiner points is more important to our application because it can avoid triangles with large aspect ratios. The result of triangulating Figure 5.7(a) using Conforming Delaunay triangulation is shown in Figure 5.7(c). Quite a few steiner points are added to the edges of the original PSLG. The resultant triangles in Figure 5.7(c) have much better aspect ratios than those in Figure 5.7(b).

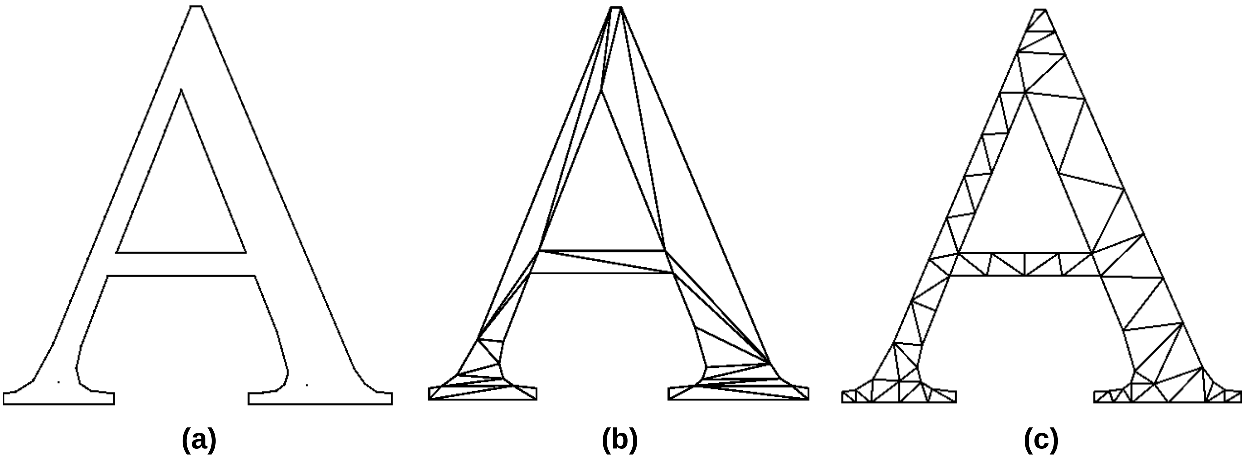


Figure 5.7: An application example of the Triangle program. (a) A Planar Straight Line Graph(PSLG); (b) the constrained triangulation of (a).

Using the *conforming Delaunay triangulation* function of Triangle, the hole boundary shown in Figure 5.6(b) is triangulated and shown in Figure 5.8. The vertices in the interior of the hole are new; there were created by Triangle in order to ensure quality triangles with good aspect ratios.

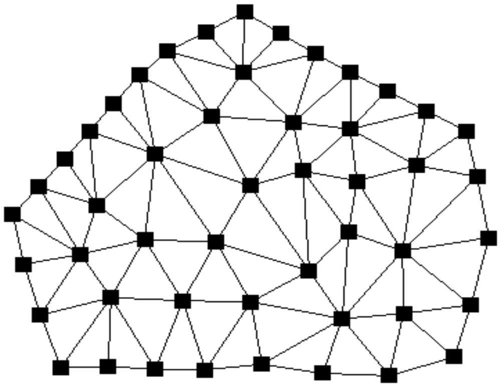


Figure 5.8: Triangulating Figure 5.6(a) using the conforming Delaunay triangulation function of Triangle.

Back Projection of Hole Boundary Vertices From the discussion in the previous section, we can see that new vertices may be created by the Triangle program. Since those new vertices are on the X-Y plane, they need to be projected back into the 3D space so that they can interpolate the original surface patch. Based on the equations (5.2) and (5.4), the positions of the new vertices on the fit plane can be computed using (5.5).

$$v_{i,\,new} = R^{-1} v''_{i,\,new} + v_{mean} \tag{5.5}$$

Update of The Surface Patch After triangulating the holes in a given surface patch, new vertices, edges and triangles are created. Therefore, the original surface patch needs to be updated by performing the following steps:

- Add the new vertices, edges and triangles to their respective lists
- Update the topology information such as vertex-triangle, triangle-edge, edge-triangle, triangle-triangle, etc.
- Perform local retriangulation if necessary

It is straightforward to perform the first step by adding the new vertices, edges and triangles to the existing arrays.

At the second step, updating the topology information means to update the connectivity information among the primitives of the surface patch. The primitives of a surface patch are vertices, edges and triangles. The triangles with which a vertex is associated are denoted as vertex-triangles. Similarly, the triangles with which an edge is associated are denoted as edge-triangles. Triangle-edge records the edges that construct a triangle. Triangle-triangle records the neighboring triangles of a triangle. When updating the topology information, special attention should be paid to the vertices, edges and triangles that are on the hole boundary because old and new primitives interact on the boundary.

The third step is necessary only when original boundary edges are broken up into multiple line segments by the *conforming Delaunay triangulation* function of the Triangle program. As explained earlier in this section, in order to ensure quality triangles with good aspect ratios, the *conforming Delaunay triangulation* function may create new vertices. If some of the new vertices are on the edges of the hole boundary, then those edges will no longer exist. They are broken up into new edges. This case is illustrated in Figure 5.9.

In Figure 5.9, v_i, i=0...5 are vertices on the surface patch. The edges [v_0, v_1], [v_1, v_2], [v_2, v_3] and [v_3, v_0] construct the boundary of the hole. The vertices u_i, i=0...3 are the new vertices created by the Triangle program. Since the new vertices are on the edges of the hole boundary, the original edges [v_0, v_1] and [v_2, v_3] are replaced by new edges [v_0, u_0], [u_0, u_1], [u_1, v_1], [v_2, u_2], [u_2, u_3] and [u_3, v_3]. Other new edges are indicated by dash lines in Figure 5.9.

Although the hole is triangulated, the surface mesh is not a valid triangular mesh because of the new vertices on the hole boundary. In this case, local re-triangulation needs to be done in order to maintain the original surface patch as a valid triangular mesh. In Figure 5.9, the dot line segments indicate the new edges created by local re-triangulation. Similar to step 2, the topology information of the surface patch needs to be updated after local re-triangulation.

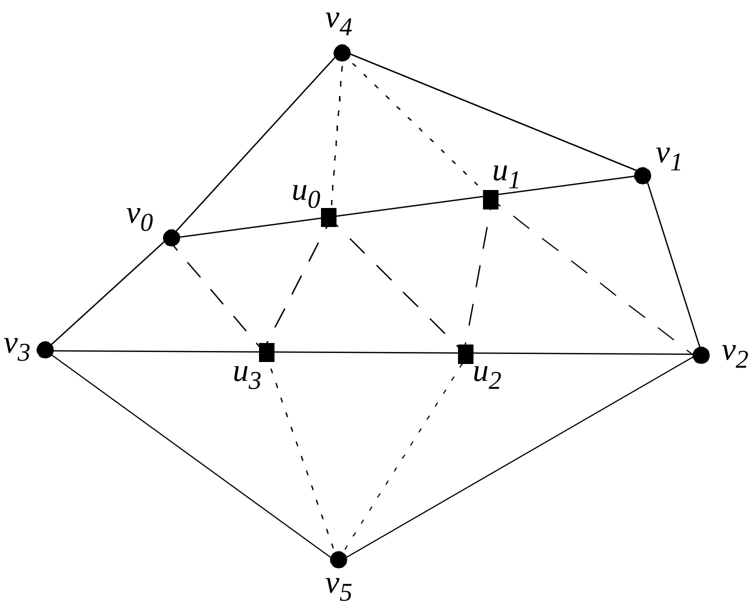


Figure 5.9: An example of broken edges on the hole boundary.

Harmonic Image with Triangulation

After using the triangulation strategy to process a given surface patch with holes, the harmonic image of that surface patch can be computed. Figure 5.10(a) and (b) show the harmonic images of the surface patches in Figure 5.1(a) and (b); respectively. The triangulated area is indicated by the arrow. Compared with Figure 5.3(b), the boundary mapping in Figure 5.10(b) is greatly improved. Figure 5.11 shows the difference between the boundary mappings in Figure 5.10(a) and (b) more clearly.

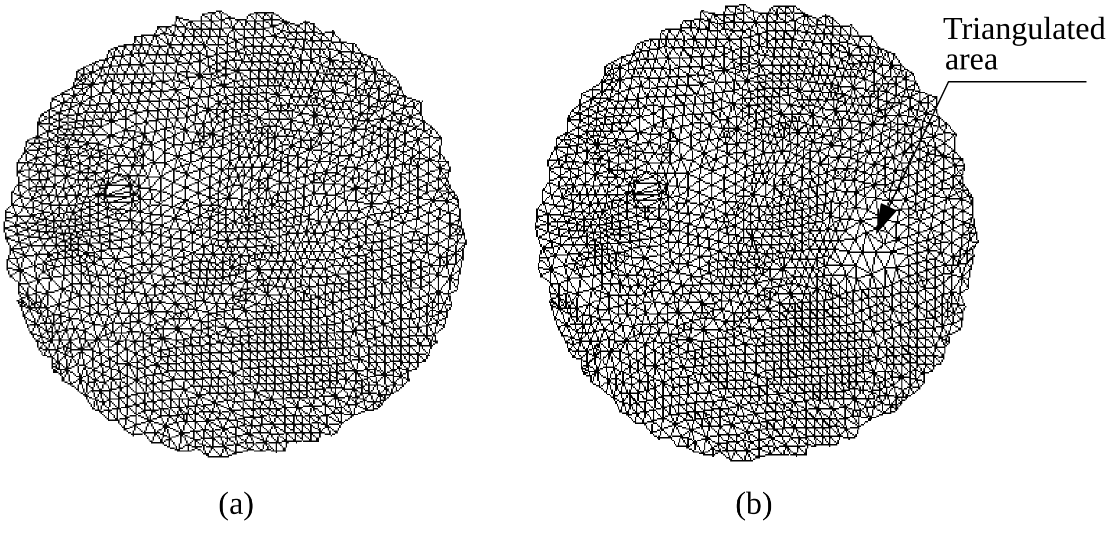


Figure 5.10: Harmonic images of the surface patches shown in Figure 5.1(a) and (b), respectively. (b) is obtained using the triangulation strategy.

Using the hole triangulation strategy, the surface distance from the central vertex to each of the boundary vertices in Figure 5.1(a) is computed. All the values are stored in a vector *d*. The same distances are computed for the surface patch in Figure 5.1(b) and stored in d_h^t. The differences between *d* and d_h^t are computed and displayed by the solid line in Figure 5.11. Without using the triangulation strategy, the same distances are computed again for the patch in Figure 5.1(b) and stored in d_h. The differences between *d* and d_h is displayed by the dashed line in Figure 5.11. It can be seen that the differences are zero for both curves except for the vertices in the index range of [0, 10) and (145, 160] because those vertices have vertices on the hole boundary on their shortest paths to the central vertex. For those boundary vertices, the differences are much smaller when using the hole triangulation strategy compared to not using it.

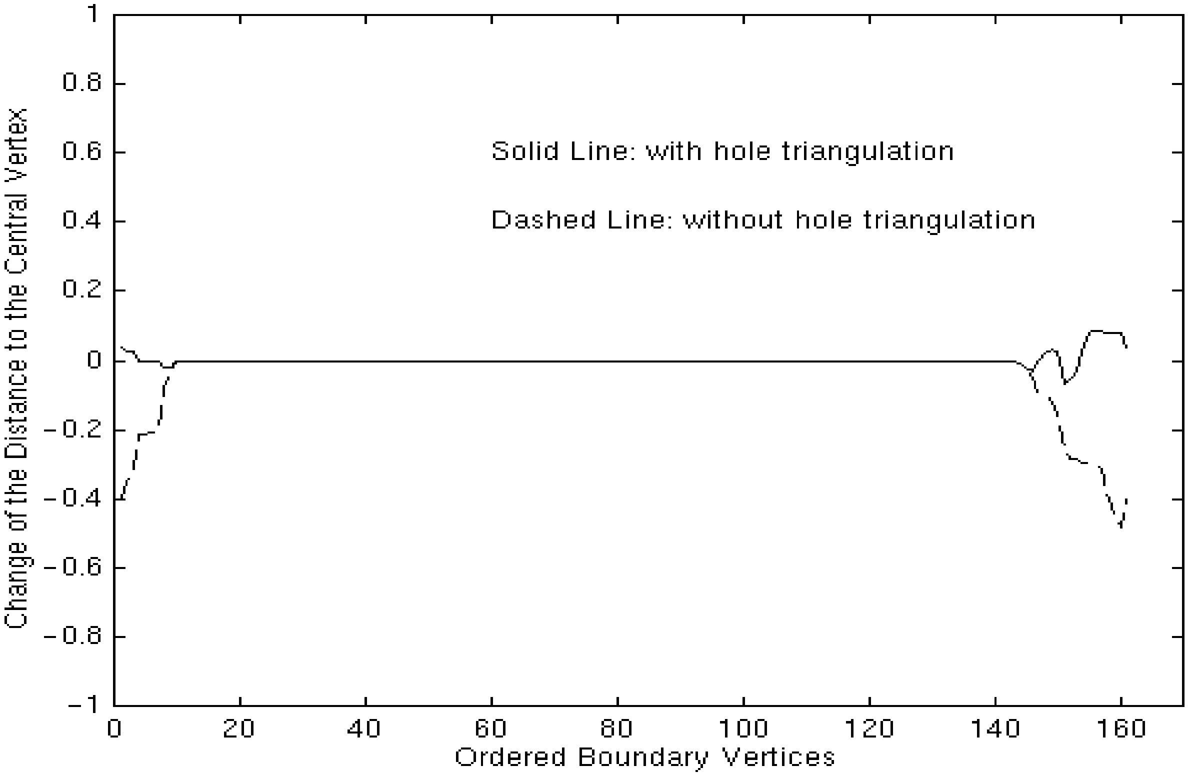
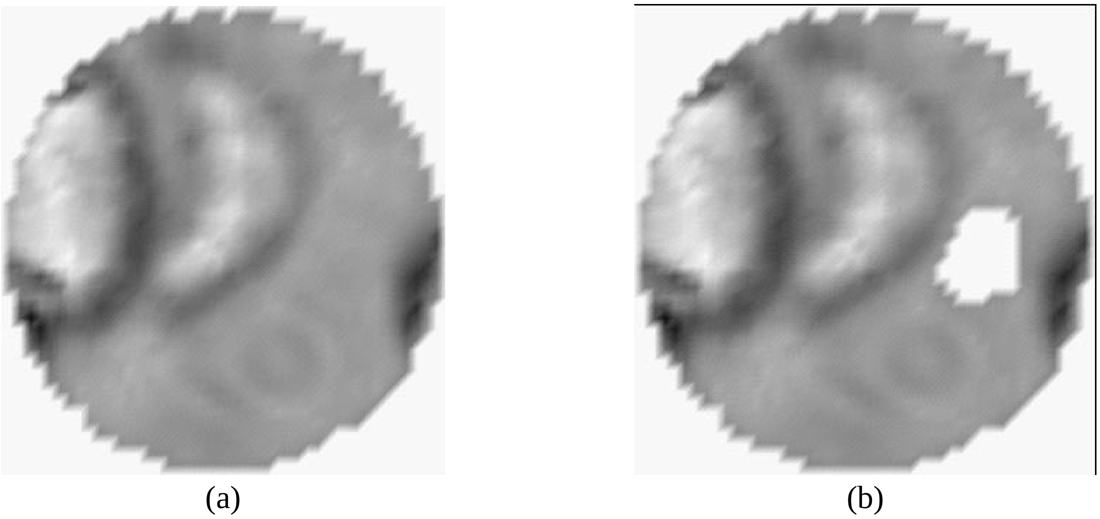


Figure 5.11: Comparison of boundary mapping with and without using the hole triangulation strategy. The unit of the vertical axis is the radius of the 2D unit disc.

It should be noticed that the change of boundary mapping depends on the shape of the underlying surface patch. Because the triangulation is actually a first-order approximation of the missing surface, it approximates the surface well when the missing surface has low curvature.

Based on the harmonic images, the Harmonic Shape Images of the surface patches in Figure 5.1(a) and (b) are computed and displayed in Figure 5.12(a) and (b), respectively. The interpolated vertices are colored white to indicate that they are invalid for shape comparison. More examples are shown in Figure 5.13 to illustrate the effectiveness of the triangulation strategy. The correlation coefficients between the surface patch in Figure 5.1(a) and each of the patches with missing data in Figure 5.1(b) and Figure 5.13 are computed; and the results are 0.996446, 0.993301 and 0.992507, respectively. This quantitatively demonstrates how well Harmonic Shape Images can handle missing data.


(a)

(b)

Figure 5.12: The Harmonic Shape Images of the surface patches in Figure 5.1(a) and (b), respectively. (b) is computed using the triangulation strategy.

5.2 Cutting And Smoothing

In Section 4.3 and 4.4, it has been discussed and demonstrated that Harmonic Shape Images are robust with respect to sampling resolution and occlusion. In order to further enhance their robustness so that the matching result will be more accurate, cutting and smoothing are used to pre-process the Harmonic Shape Images before matching them.

Recall the construction of boundary mapping for generating Harmonic Shape Images in Chapter 2. Given a surface patch, its boundary vertices can be classified into radius boundary vertices and occlusion boundary vertices according to the surface distance from each boundary vertex to the central vertex of the surface patch. This means that the occlusion parts of the surface patch can be exactly located before comparison. Considering the fact that most distortion on Harmonic Shape Images appears around the occlusion boundary, discounting the corresponding pixels in those areas when comparing the images should help obtain the correct matching result. In our implementation, this technique is referred as cutting. Smoothing is also applied to the Harmonic Shape Images before matching in order to decrease the distortion caused by resolution and occlusion. In our implementation, Gaussian smoothing is used with $\sigma = 0.25$. The implementation of cutting and smoothing is further explained using the example in Figure 4.40.

Figure 5.14(a), (b) and (c) show the surface patch and its Harmonic Shape Image on the third row of Figure 4.40. After applying cutting and smoothing on (c), the resultant Harmonic Shape Image is shown in Figure 5.14(d). The normalized correlation coefficient between the occluded patch and the original patch (shown in the first row of Figure 4.40) increases from 0.759951 to 0.854042. Using the cutting and smoothing techniques, the second experiment on occlusion (Fig-

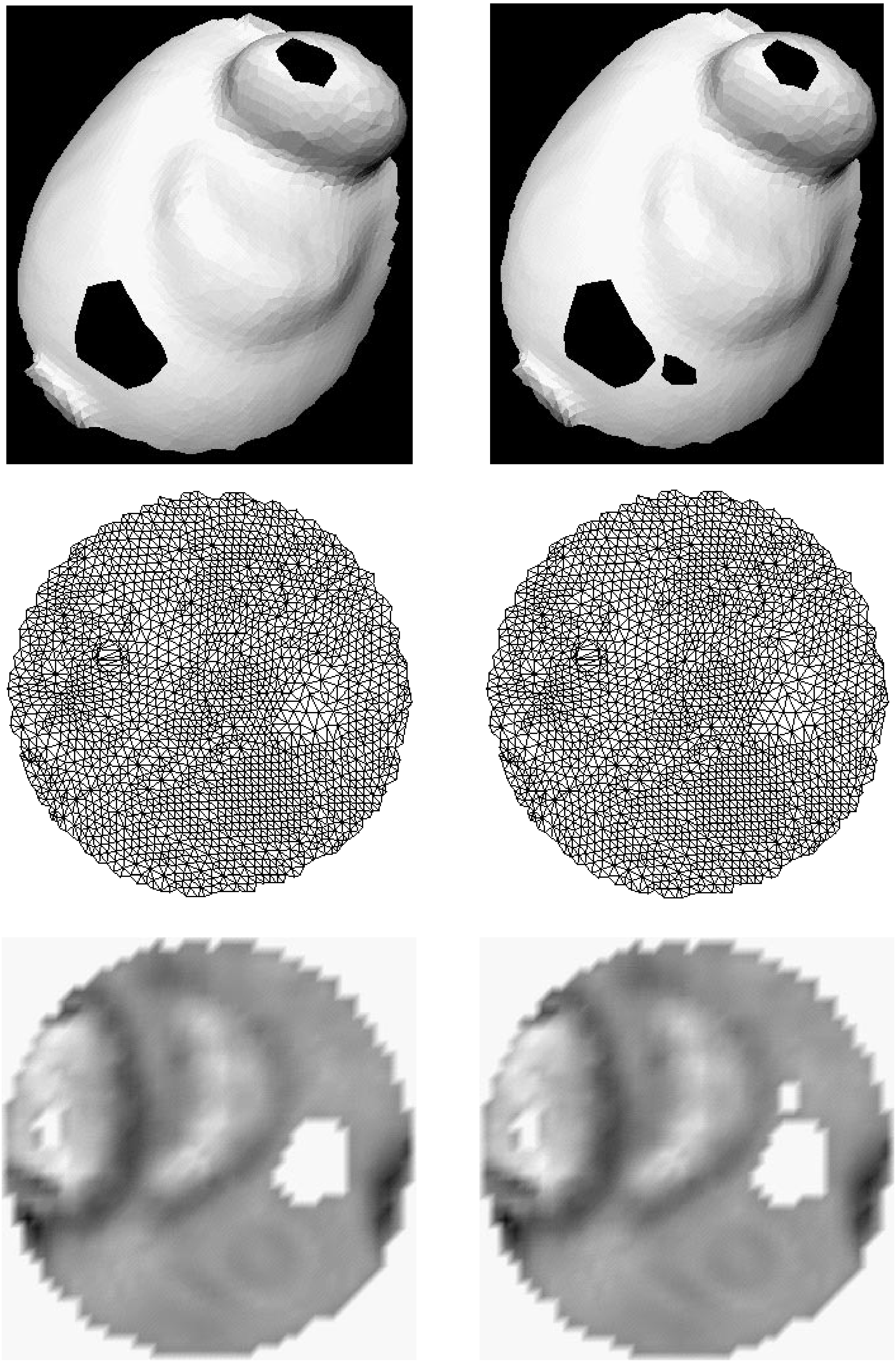


Figure 5.13: Examples of using the triangulation strategy to compute the Harmonic Shape Images of surface patches with holes.

ure 4.40 and Figure 4.41) is repeated and the results are presented as follows. Figure 5.15 shows the curves of normalized correlation coefficient as a function of the occlusion boundary with and without applying cutting and smoothing. Clearly, the comparison of the two curves shows that, with the help of cutting and smoothing, the correlation coefficient improves as the boundary occlusion increases.

Although smoothing can help decrease the distortion caused by resolution and occlusion, the smoothing parameter σ cannot be specified too large because large smoothing eliminates features of surface patches and causes different patches to appear similar. To verify that cutting and smoothing do not introduce confusion in surface matching, the occluded patches in Figure 4.40 and Figure 4.41 are compared with all the 16 patches in the second patch library. As before, pair comparison among all the patches is performed and the matrices for the normalized correlation coefficients and shape similarity are shown in Figure 5.16(a) and (b), respectively. The histogram of the shape similarity values for each occluded patch compared with the 16 patches is shown in Figure 5.17. These results further demonstrate how the occluded patches can be distinguished from other patches with the help of cutting and smoothing.

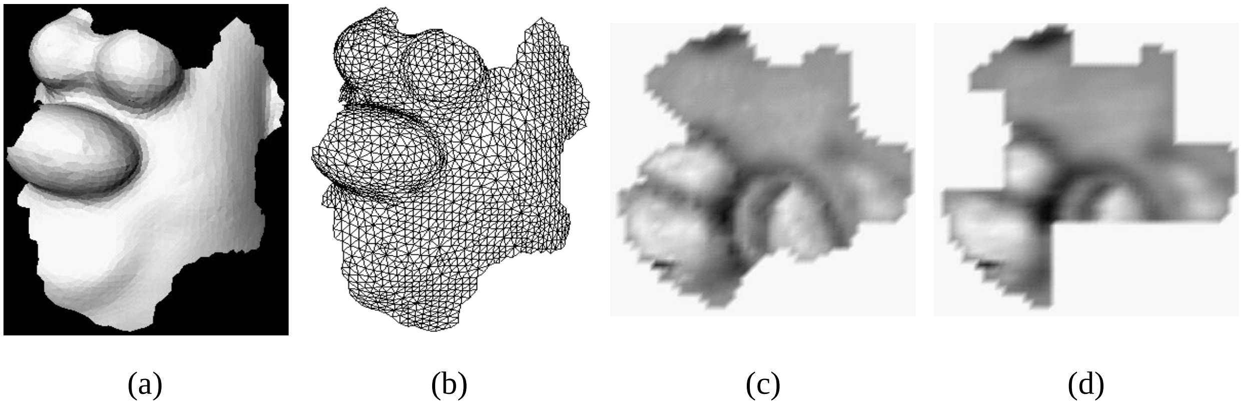


Figure 5.14: The effect of cutting and smoothing on Harmonic Shape Images. (a) surface patch with occlusion; (b) wireframe version; (c) the Harmonic Shape Image of (a); (d) the resultant Harmonic Shape Image of (c) with cutting and smoothing with σ = 0.25;

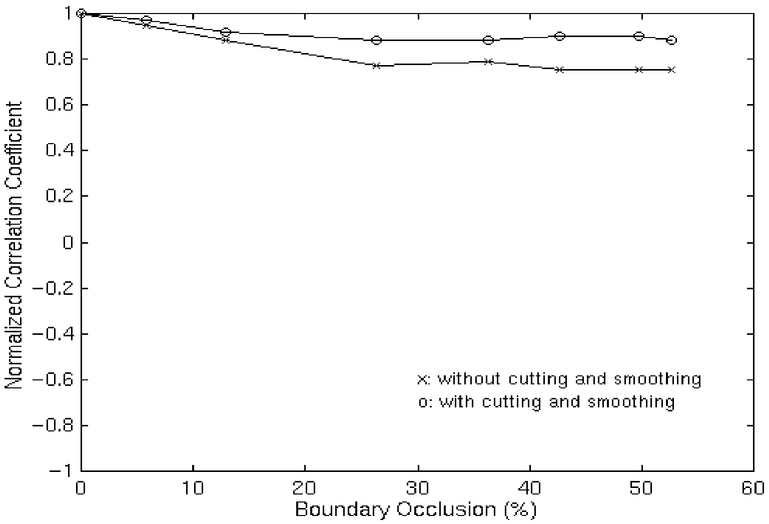
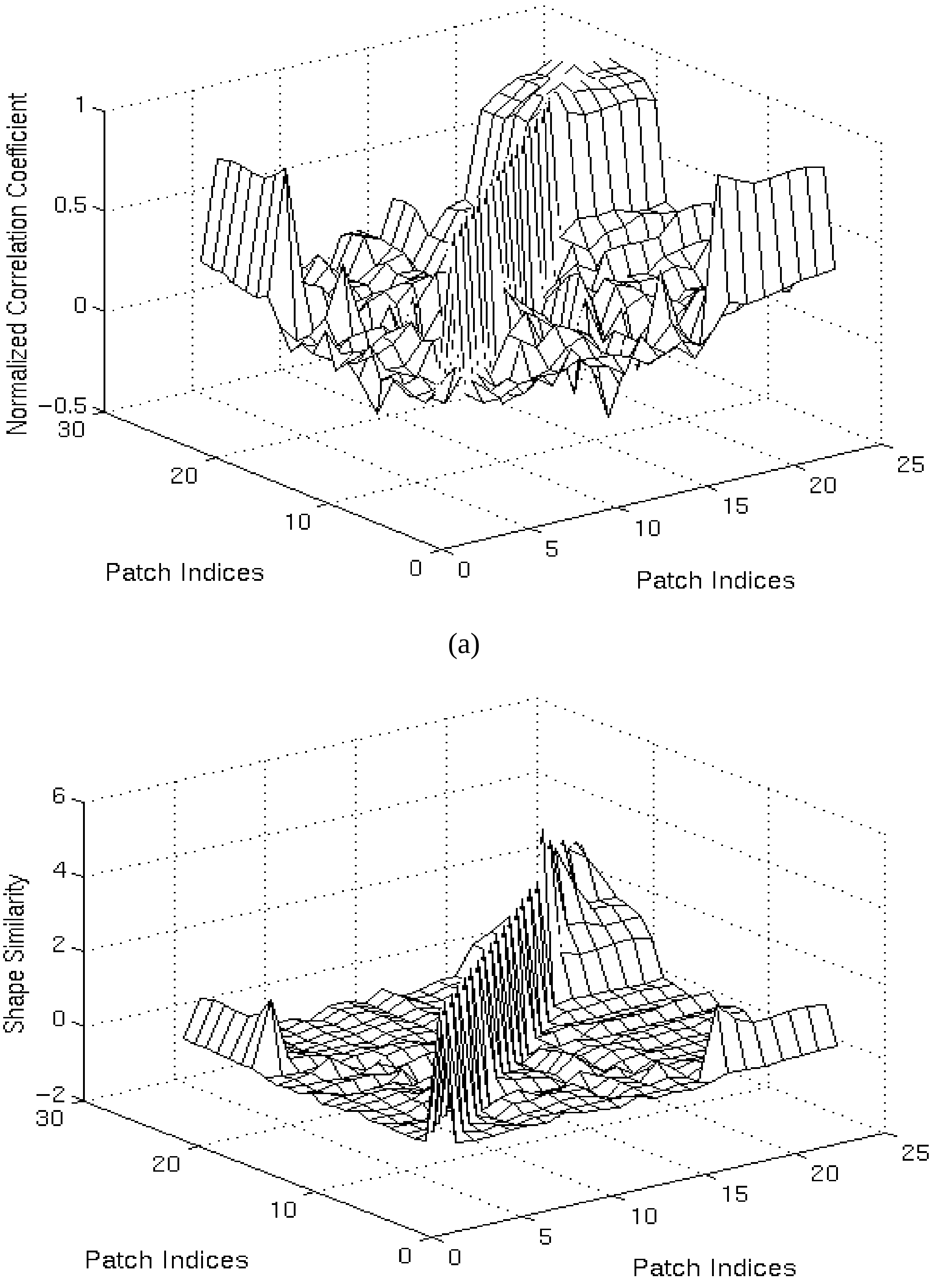


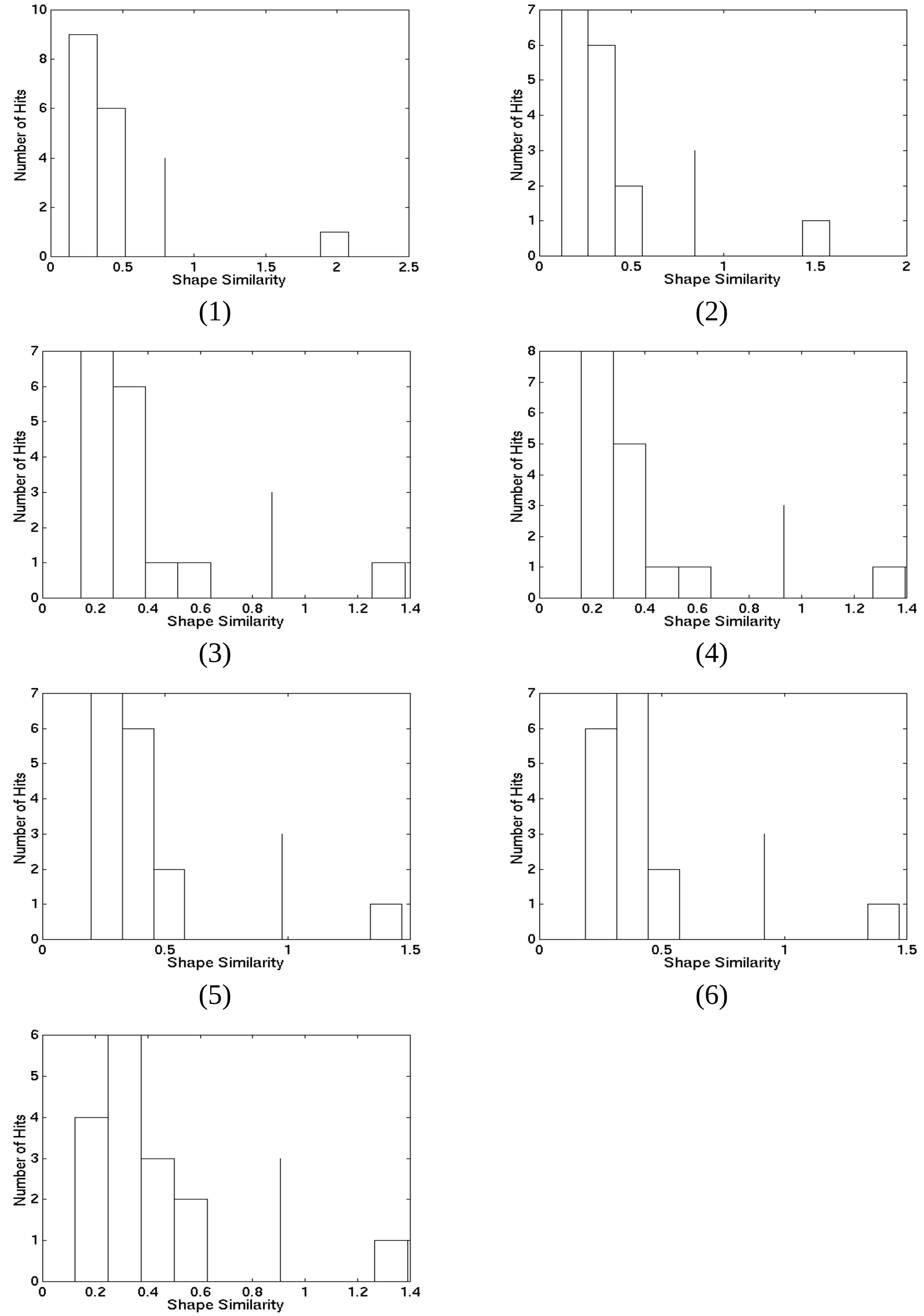
Figure 5.15: The effect of cutting and smoothing with respect to occlusion. The normalized correlation coefficient improves as boundary occlusion increases.



(a)

(b)

Figure 5.16: The results of pair comparison in the library with occlusion patches. (a) The matrix of normalized correlation coefficients; (b) the matrix of shape similarity values.



(1)

(2)

(3)

(4)

(5)

(6)

(7)

Figure 5.17: Histogram of shape similarity values for the occlusion patch 1 through 7 compared with the 16 patches in the library.

Chapter 6

Additional Experiments

In this chapter, the results of additional experiments on surface matching using Harmonic Shape Images are presented. The experiments include surface registration for face modeling and mesh watermarking, accuracy test for surface registration and object recognition in scenes with occlusion. These additional experiments are presented to further demonstrate the usefulness of Harmonic Shape Images in different applications of surface matching. The algorithms and strategies in those examples can be incorporated into a surface matching system in the future, although building a complete system is not addressed at this time.

6.1 Surface Registration for Face Modeling

Face modeling has received more and more attention in the vision and graphics fields due to its important role in the medical area, e.g., plastic surgery. Systems that can capture accurate 3D geometry and high quality color surface data along with software that can provide accurate face models can aid the surgeons in various ways. For example, it is possible to measure distance and area on 3D models; it is practical to track recovery, growth and degenerative conditions by direct comparison of 3D models; it is possible to objectively compare the effects of alternative treatments by comparing the 3D models.

In the experiments of face modeling, the range images are provided by the research group led by Professor Shizhuo Yin in the Electrical Engineering Department of Pennsylvania State University. Typically, each data set consists of three images of either a human face or the face of a mask: one front view along with two side views. Each of the two side views has about 40% overlap with the front view. There is no prior knowledge about the transformation among any of the range images. An example face data set is shown in Figure 6.1.

The example set shown in Figure 6.2[1] is used to illustrate the registration process. In Figure 6.2, The front and left range images are first registered. Some of the statistics of the meshes are listed in Table 6.1. Because the meshes are of high resolution, a hierachical approach is used to perform the registration (discussed in Chapter 3) for experimental purposes. A number of vertices are selected randomly on both the front and the left images so that those vertices are distributed as evenly as possible. Some of the selected vertices are shown in Figure 6.3. The number of randomly selected vertices on the front and left images are denoted as n_{front} and n_{left}. Surface patches are created using those vertices as central vertices; the radius of the surface patches is selected to be 40.0. The radius is determined based on the following mesh information. The area of the surface patch is selected to be 20% of the overlapping area of the two images. The area of the overlapping part of the two images is estimated using the average of the areas of the two meshes multiplied by the estimated overlapping percentage, about 40% in this case. Assuming that the surface patch is a disc, the radius can be calculated once the area of the disc is known.

[1] The range images were taken from a mask instead of a human face in this example.

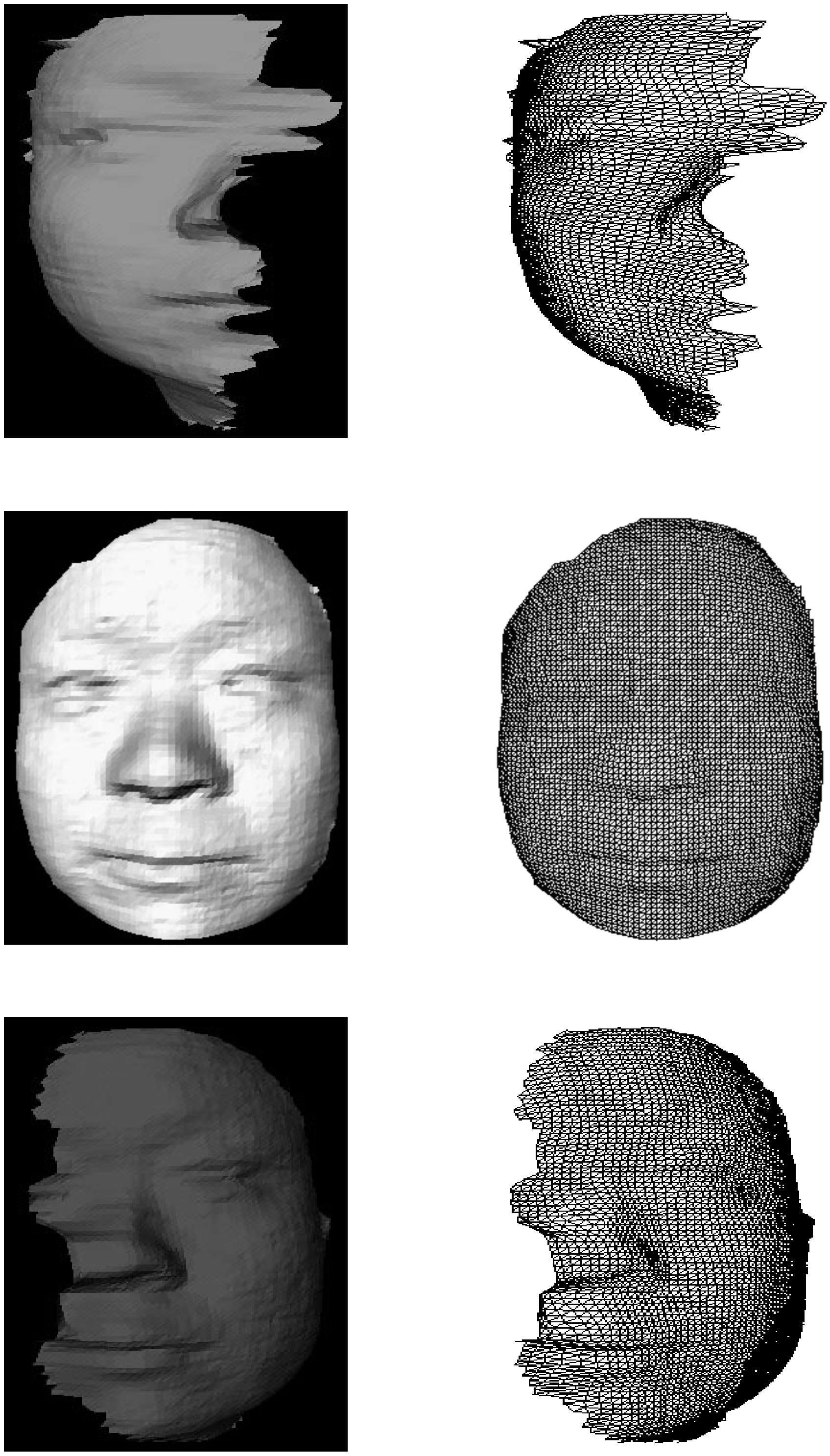


Figure 6.1: An example set of the face range images. First row: the rendered view of the right, front and left images; second row: the wireframe of the right, front and left images.

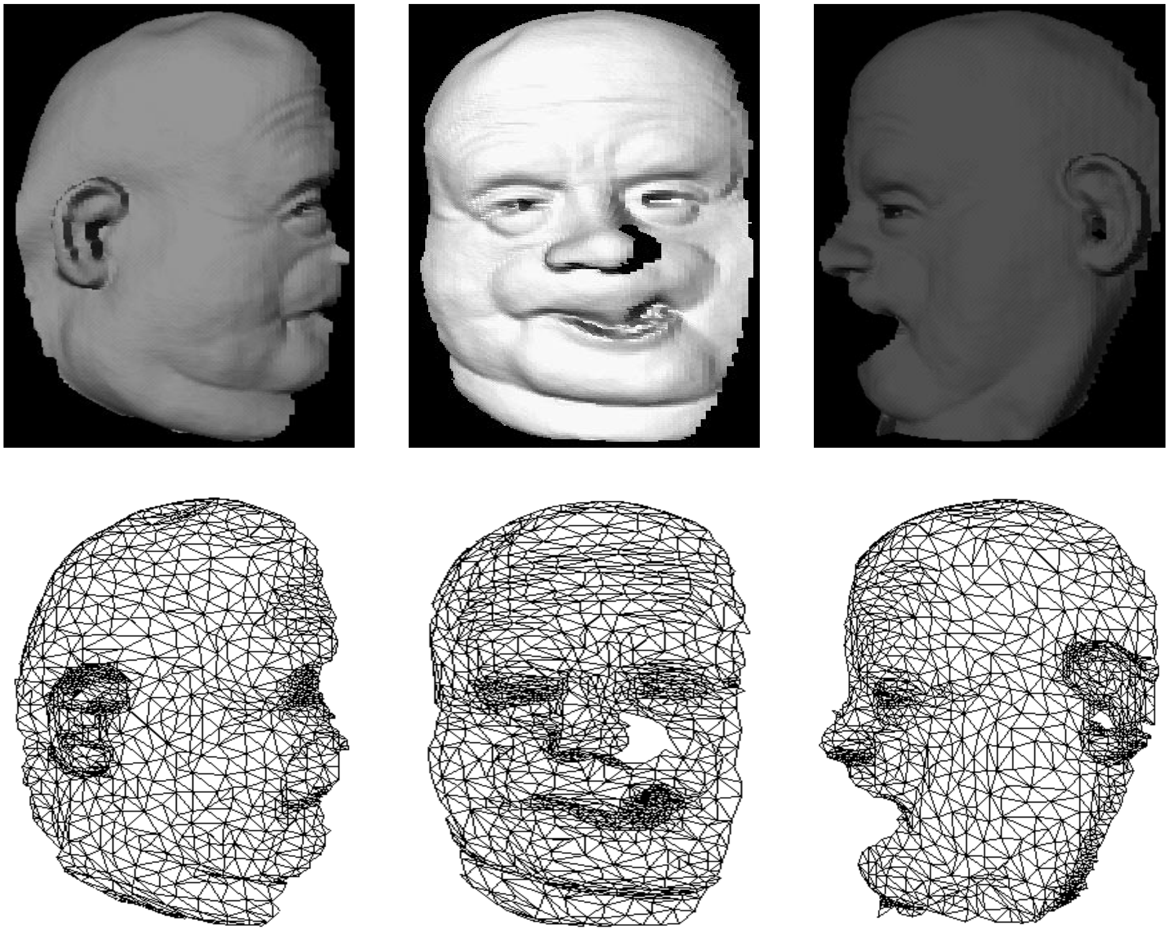


Figure 6.2: An example set of the face range images. First row: the rendered views of the right, front and left images; second row: the wireframe versions of the right, front and left images. The wireframe versions are meshes after simplification, which is for display purposes only. The original meshes are of high resolution. (See Table 6.1 for mesh statistics.) The wireframe versions shown in later figures are all simplified meshes.

Table 6.1: Mesh Statistics

Mesh	Right	Front	Left
Num. of vertices	11081	10227	11103
Num. of triangles	21709	19960	21719
Area (cm^2)	50202.3	48468.5	50658.6

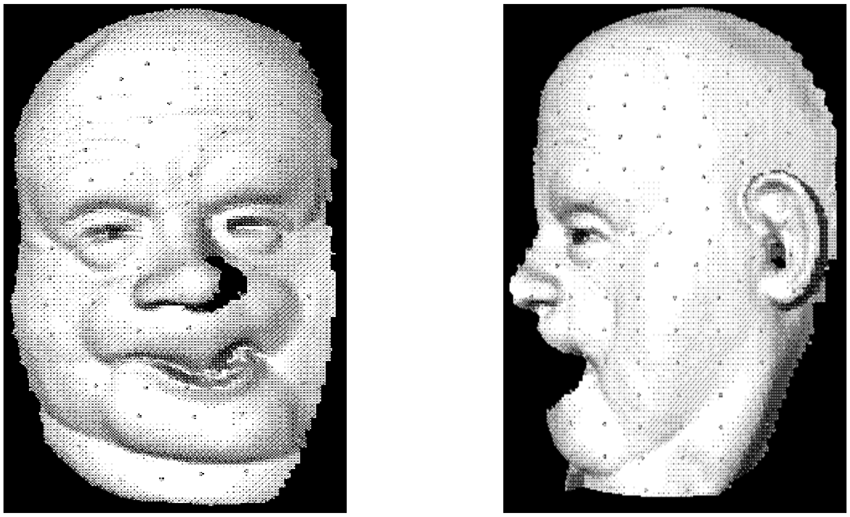


Figure 6.3: Randomly selected vertices on the front and left images.

After surface patches are created on both meshes, a validity checking step is performed. A surface patch is considered to be valid when satisfying the following requirements.

- Topology

  If the surface patch has holes in it, then those holes are triangulated. If none of the estimated area of the holes exceeds 5% of the total area of the surface patch, then the topology requirement is considered to be satisfied.

- Occlusion

  If the angle for any occlusion range is larger than $\alpha * \pi$, then the patch is considered to have too much occlusion, therefore not satisfying the occlusion requirement. α is a parameter determined by the application. In general, in the case of surface registration, the patches that are on the boundary of meshes tend to have more occlusion. Accordingly, α is set to be 0.375, making the threshold of the angle for the occlusion range equal to 67.5 degrees.

After the validity check, the number of valid surface patches on the front and left meshes is n'_{front} and n'_{left}. The Harmonic Shape Images are created for each of the valid patches. The comparison is performed by matching each Harmonic Shape Image on the front mesh to all the Harmonic Shape Images on the left mesh. Alltogether, there are n'_{front} x n'_{left} number of comparisons to be performed.

Recall that the issue of surface matching using Harmonic Shape Images has been discussed in Chapter 3. When given two surfaces to be matched, *m* surface patches were randomly selected on the first surface while only one surface patch was selected on the second surface. The process of searching for the best-matched patch for the patch on the second surface was explained in detail. In practice, considering the fact that, on the first surface, there may not be a match for a randomly selected patch on the second surface. In this case, more surface patches need to be selected on the second surface. That is what has been done in the above face modeling example. There, the principle of searching for the best-matched patch for any one patch is the same as that explained in Chapter 3. The difference is that, because there may be multiple good matches, one more step needs to be taken in order to determine which match is the best one. This extra step is explained below.

After the comparison of Harmonic Shape Images on both meshes, the result for each patch on the front mesh compared with all the patches on the left mesh is analyzed using the fourth-spread statistical method which was discussed in Chapter 3. The following are some examples of the resultant histograms of the comparison results.

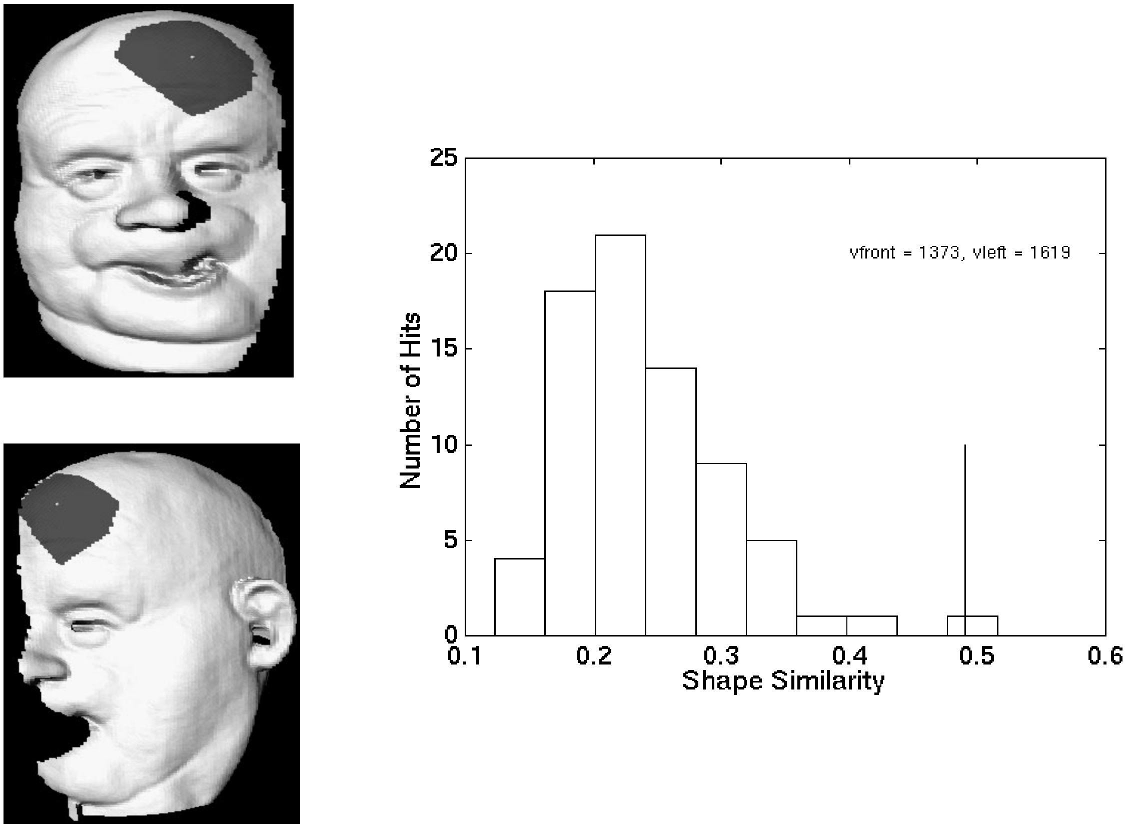


Figure 6.4: An example of patch comparison results between the front mesh and the left mesh. The Harmonic Shape Image of the surface patch on the front mesh is compared to the Harmonic Shape Images of all the selected valid patches on the left mesh. The histogram of the shape similarity values is shown on the right. One qualified match on the left mesh is found which is shown by overlapping it on the left mesh. It should be noticed that, in this case, the shape similarity value of the best-matched patch is just above the threshold.

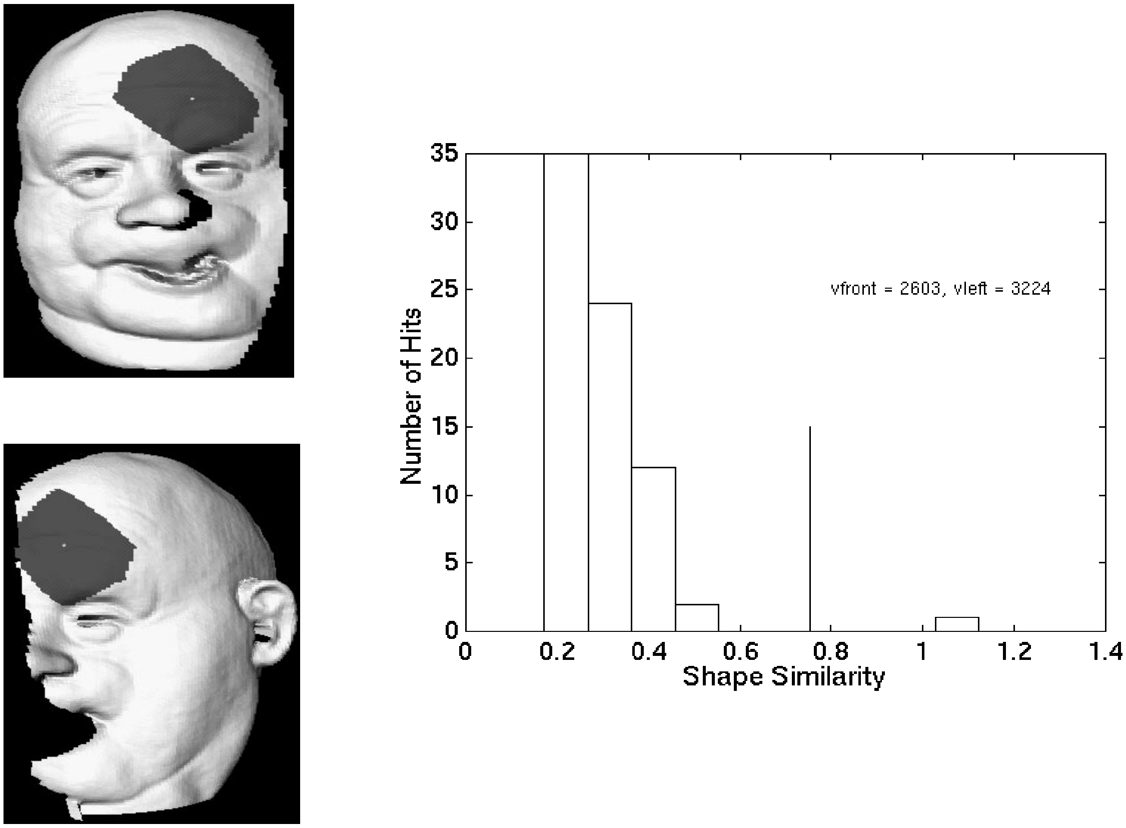


Figure 6.5: An example of patch comparison results between the front mesh and the left mesh. The Harmonic Shape Image of the surface patch on the front mesh is compared to the Harmonic Shape Images of all the selected valid patches on the left mesh. The histogram of the shape similarity values is shown on the right. One qualified match on the left mesh is found which is shown by overlapping on the left mesh. In this case, the shape similarity value of the best-matched patch is well above the threshold.

In Figure 6.4, the histogram of the shape similarity values shows that there is one qualified match (above the threshold) on the left mesh for the patch on the front mesh. The two matched patches are shown by overlapping them on the front and left meshes, respectively. The shape similarity value of the qualified match is just above the threshold. In Figure 6.5, the histogram shows a qualified match whose shape similarity value is well above the threshold. This fact shows that the surface patch on the front mesh is well distinguished from the patches on the left mesh. In Figure 6.6, no qualified matches are found according to the histogram because all the shape similarity values are below the threshold.

The examples in Figure 6.4, Figure 6.5 and Figure 6.6 are representatives of the n'_{front} x n'_{left} patch comparison results between the front mesh and the left mesh. Two issues should be brought up regarding the comparison results. Let us define the surface patches on the front mesh to be scene patches and those on the left mesh to be model patches. The first issue is to select the patch which should be used to perform the search on the left mesh. The second issue is that more than one scene patches found their qualified matches among the models patches, which pair of matched patches should be used to compute the transformation between the two meshes. The discussion on the above two issues is as follows.

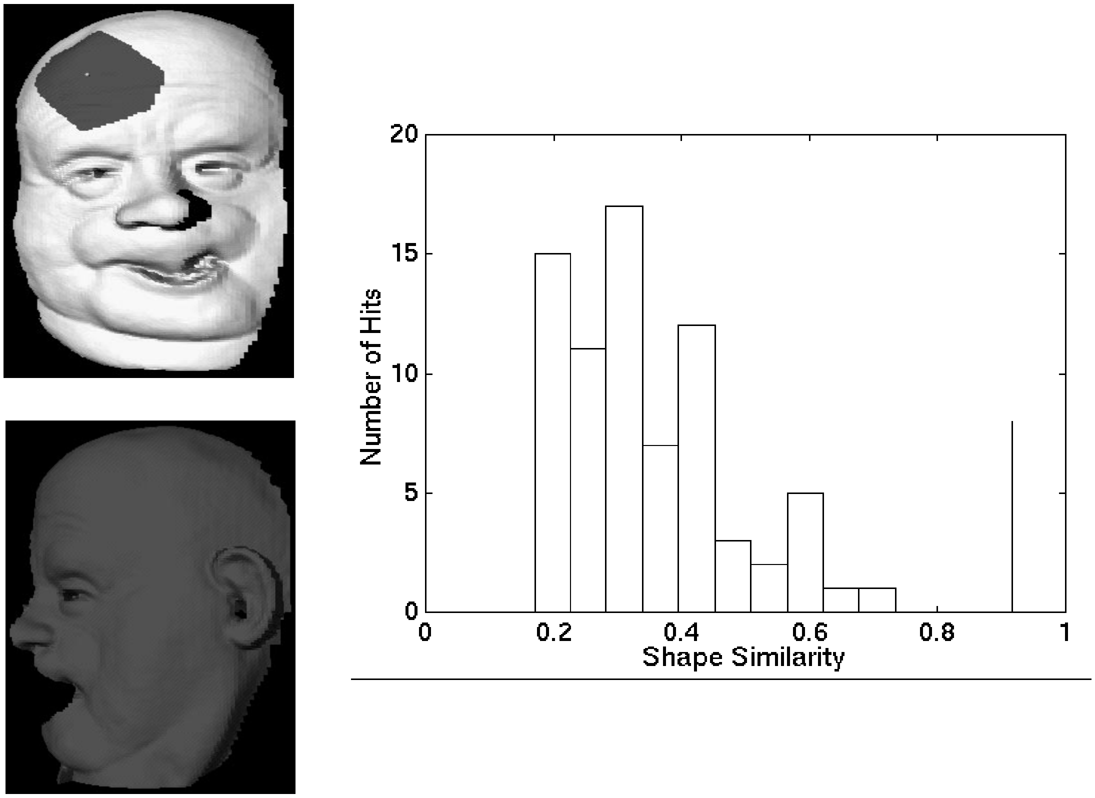


Figure 6.6: An example of patch comparison results between the front mesh and the left mesh. The Harmonic Shape Image of the surface patch on the front mesh is compared with the Harmonic Shape Images of all the selected valid patches on the left mesh. The histogram of the shape similarity values is shown on the right. In this case, no qualified match is found because all the shape similarity values are below the threshold.

Selecting Scene Patches

It is true that there are many possible scene patches on the front mesh. In fact, every vertex on the front mesh can be used as a central vertex to generate a scene patch using the specified radius. However, not every one of these scene patches can find its qualified match on the left mesh, because meshes only partly overlap. In general, there are two types of scene patches that may be able to find their matches among the model patchesl The ones in the overlapping region are the registration-correct matches for which we are looking. Others outside the overlapping region are registration-incorrect matches caused by local shape similarity of the two surfaces, e.g., symmetry. Because there is no prior knowledge about the overlapping region, it is impossible to select a scene patch in that region searching for its best match among the model patches. The solution to this problem is to use a number of randomly selected scene patches. The patches in the overlapping region have equal chance to be selected as other patches.

Selecting the Best Match for Computing Transformation

The second question is that, among all the qualified matches, which pair of patches should be used to compute the transformation? The solution to this problem is to define a discriminability value *d(p, Ω)* between a surface patch and a set of surface patches according to the histogram of the shape similarity values. *d(p, Ω)* is defined in (6.1).

$$d(p, \Omega) = max(C(p, q)|q \in \Omega) - Threshold(C(p, q)|q \in \Omega) \tag{6.1}$$

In (6.1), *p* denotes a surface patch, Ω denotes the set of surface patches being compared to *p*. *C(p, q)* denotes the shape similarity value between the Harmonic Shape Images of *p* and *q*. The first term on the right-hand side is the maximum of the shape similarity value; the second term is the threshold for the qualified matches. When the discriminability value *d(p, Ω)* is positive and large, the best match *q* in Ω of *p* can be well identified from Ω. When *d(p, Ω)* is positive and small, the best match *q* in Ω of *p* can be identified but with less discriminability than the case in which *d(p, Ω)* is positive and large. When *d(p, Ω)* is negative, no qualified match is found. Based on the above definition, the pair of matched surface patches with the largest discriminability value should be used to compute the transformation between the two meshes.

The matches in the above example with positive discriminability values are shown in Figure 6.7. Some of the matched surface patches along with their Harmonic Shape Images are shown in Figure 6.8. The one on the last row is a mis-match. Among all the matched surface patches, five of them are correct matches. Their discriminability values are shown in circles in Figure 6.7. Some analysis that has been done regarding the matching result is presented below.

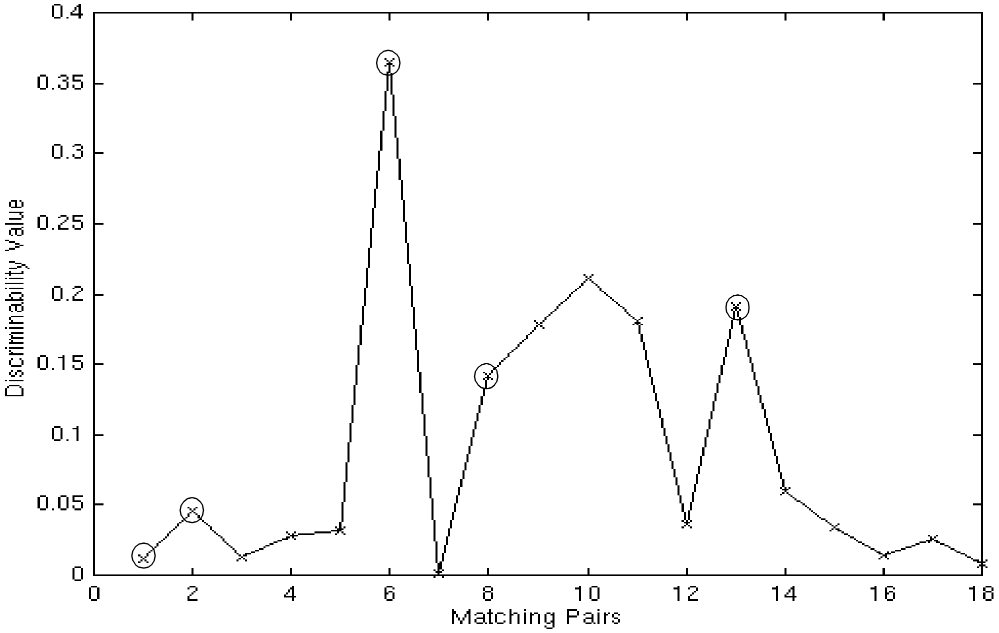


Figure 6.7: The matches on the two surfaces in Figure 6.3 with positive discriminability values.

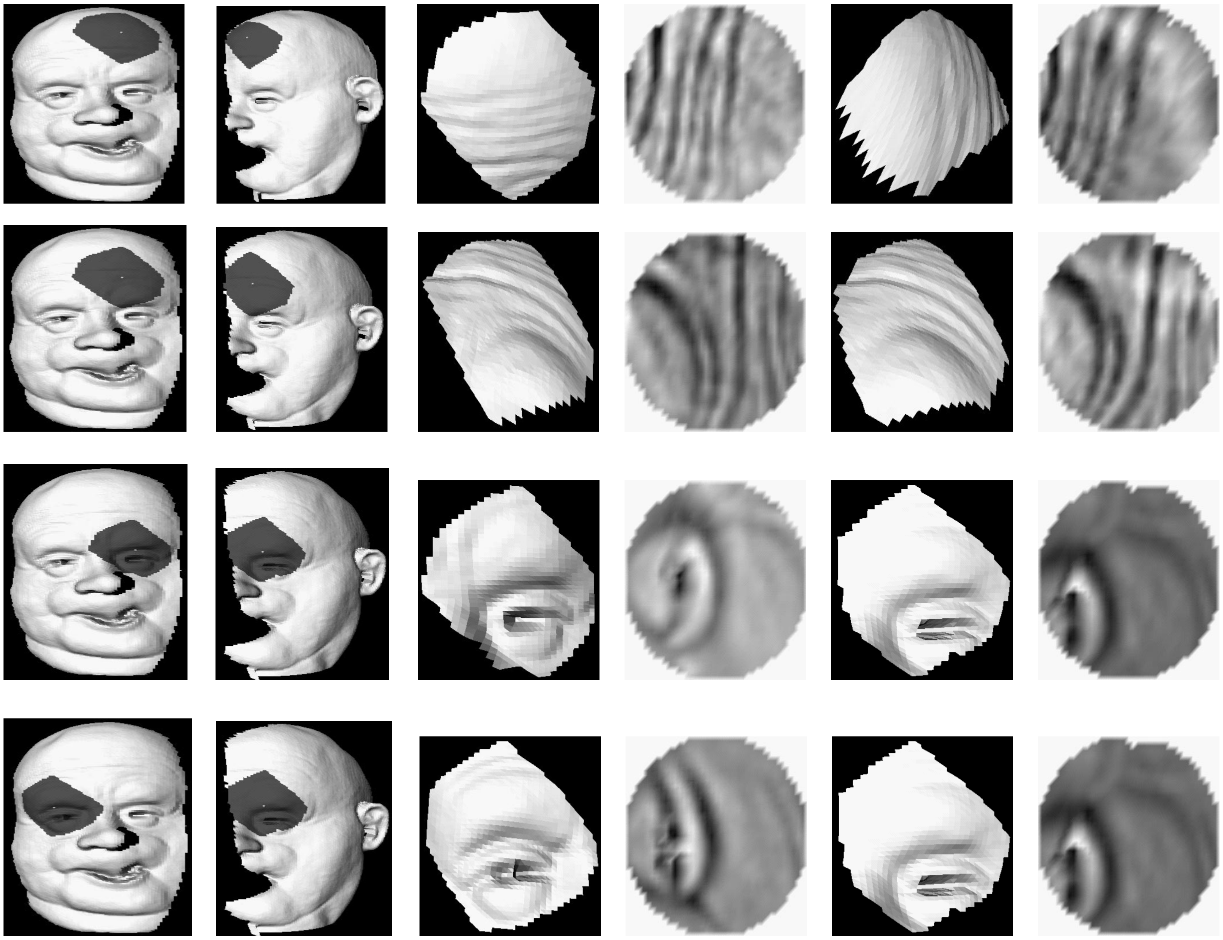


Figure 6.8: Matched surface patches on the two surfaces in Figure 6.3 with positive discriminability values.

The Match with the Largest Discriminability Value

The matched patches shown on the second row of Figure 6.8 constitute a correct match. Its discriminability value is the largest in Figure 6.7. Therefore, the transformation between the two surfaces should be computed using the correspondences on the two matched surface patches.

Patches with More Features vs. Patches with Fewer Features

Figure 6.8 shows that the matched patches are rich in shape variation. This justifies the observation that patches with more features are more easily identified among a group of surface patches. This also shows that Harmonic Shape Images represent those shape variations well.

Interpreting Mis-matches

There is a mis-match on the last row of Figure 6.8. It should be noted that the mis-matches are not correct in the sense of using them to compute the transformation between the two surfaces. In the sense of matching Harmonic Shape Images, the mis-matches *are* correct because they are similar in shape. This is the case for the mis-matches in Figure 6.8. Mis-matches occur because surface patches are only part of the entire surface. There may be multiple patches of the same shape on the surface. For example, the front mesh in the above example has symmetry in the area of the eyes. Therefore, patches around the two eyes on the front mesh are matched to the patches around the eye on the left mesh.

At the conclusion of the matching process, the two best-matched surface patches on the second row of Figure 6.8 are used to compute the transformation between the two surfaces. Using the correspondences on the two patches, the following result is obtained. The angles are *X*-*Y*-*Z* fixed angles.

$$\alpha = -89.541844, \beta = 29.977152, \gamma = 87.96621$$

$$t_x = 102.444318, t_y = 11.359146, t_z = -117.445421$$

The two registered meshes are shown in Figure 6.9.

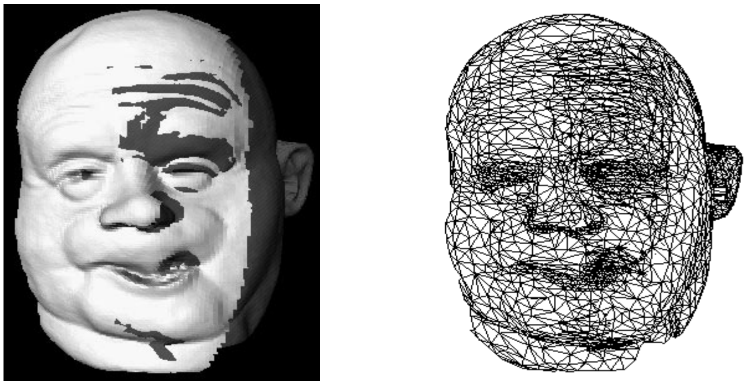


Figure 6.9: Registered meshes of the front mesh and the left mesh.

Since the meshes in the above example are of high resolution, the search performed above is only a coarse search, namely, only n'_{front} and n'_{left} vertices are selected on the two meshes, respectively. A fine search can then be conducted in the neighborhood of the vertex which is the central vertex of the best-matched patch on the left mesh. This neighborhood is shown in Figure 6.10(a). The scene patch on the front mesh is shown in Figure 6.10(b). It is the scene patch that has the highest discriminability value during the coarse search. After the fine search, the best-matched patch on the left surface is shown in Figure 6.10(c). Using the correspondences between the two matched patches, the transformation result is as follows.

$$\alpha' = -86.018154, \beta' = 28.3786, \gamma' = 88.6423$$

$$t_x^{'} = 102.677337, t_{y'} = 9.943254, t_{z'} = -117.079115$$

The registered meshes after the fine search is shown in Figure 6.11.

Using the same principle, the right mesh (Figure 6.2(a)) and the front mesh are registered as shown in Figure 6.12. As can be seen from Figure 6.12(a) and (b), the registration result after the coarse search is not particularly good. However, it provides a good initial guess for performing the fine search. The registration result after the fine search improves significantly (Figure 6.12(c), (d)).

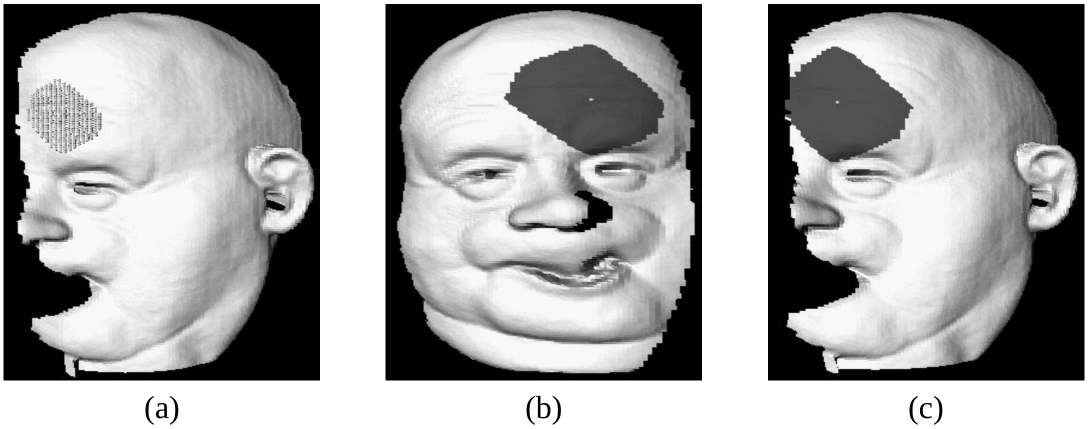


Figure 6.10: Fine search on the left surface. (a) The central vertices of the patches among which the fine search is conducted; (b) the patch on the front surface to be matched; (c) the best match of (b) on the left surface after the fine search.

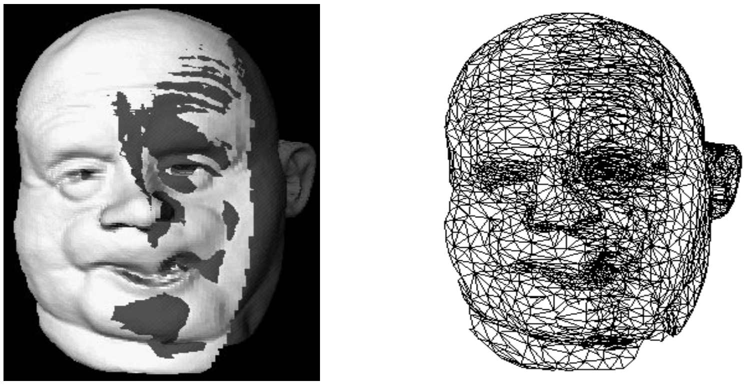


Figure 6.11: The registered meshes after the fine search.

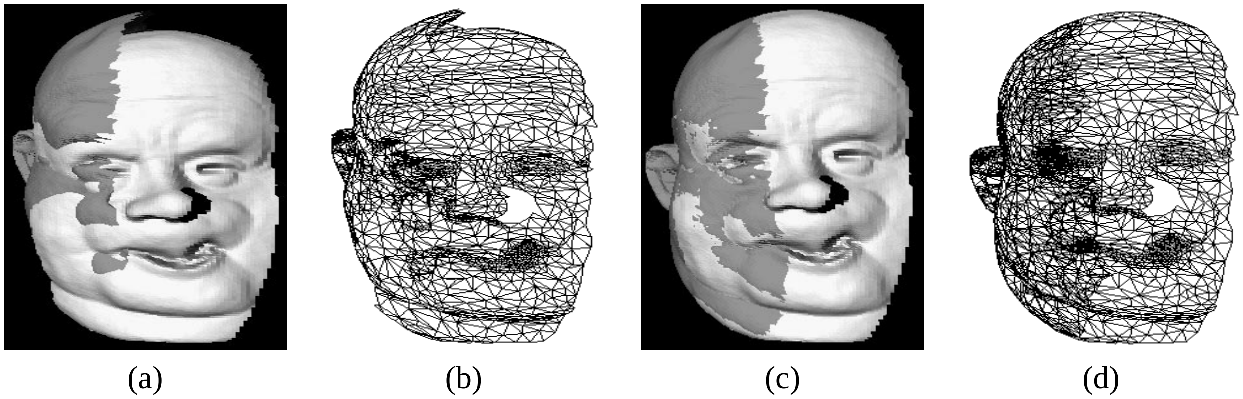


Figure 6.12: The registered meshes of the front mesh and the right mesh. (a), (b) registration result after the coarse search; (c), (d) registration result after the fine search.

Using a mesh integration program described in [46], the three registered meshes (Figure 6.13(a)) are integrated into a single mesh (Figure 6.13(b), (c), (d)).

The registration and integration results for the meshes in Figure 6.1 are shown in Figure 6.14.

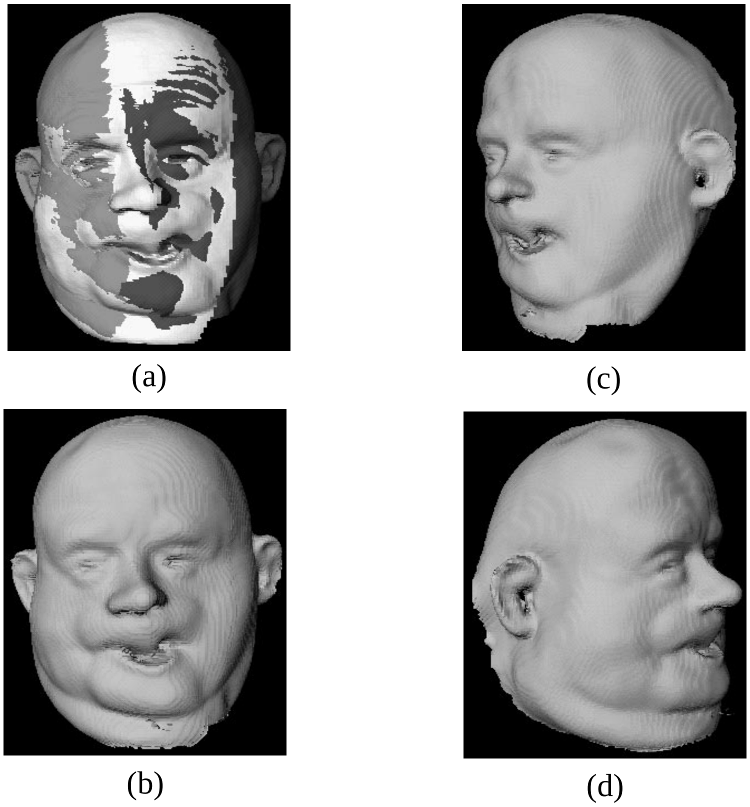

(a) (c) (b) (d)

Figure 6.13: Mesh integration. (a) The three registered meshes to be integrated; (b), (c) and (d) the front, left and right views of the integrated mesh.

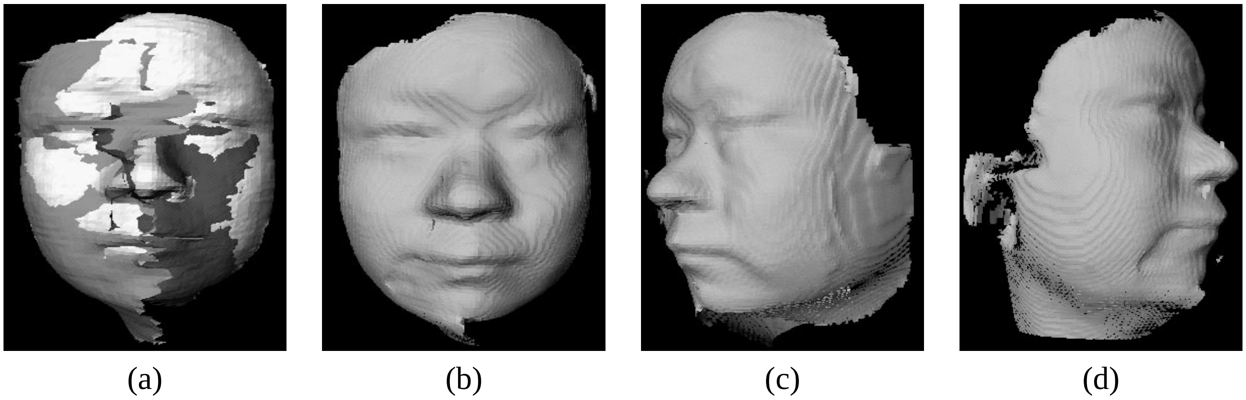

(a) (b) (c) (d)

Figure 6.14: Mesh integration. (a) The three registered meshes to be integrated; (b), (c) and (d) the front, left and right views of the integrated mesh.

6.2 Registration in mesh watermarking

An approach was proposed by Dr. H. Hoppe in [62] to watermarking triangular meshes. Watermarking provides a mechanism for copyright protection of digital media by embedding information identifying the owner in the data. The piece of embedded information is called watermark. Robust watermarks are able to survive a variety of attacks, including transformation, scaling, cropping, smoothing, simplification and resampling. The watermarking process proposed in [62] is as follows.

A watermark vector *w* is selected using the information inherent to the original mesh *M*. Then the watermark, *w*, is inserted into *M* using some basis functions defined on *M*. The watermarked mesh is denoted as $M_{watermark}$. As a result of this insertion, the 3D position of each vertex on *M* has gone through a different linear transformation in the following form.

$$\begin{bmatrix} \\ v_x' \\ \\ \end{bmatrix} = \begin{bmatrix} v_x \end{bmatrix} + \varepsilon \begin{bmatrix} \Phi \end{bmatrix} \begin{bmatrix} h_1 d_{1x} & & \\ & \ldots\ \ldots & \\ & & h_m d_{mx} \end{bmatrix} \begin{bmatrix} w \end{bmatrix} \tag{6.2}$$

In (6.2), v_x' are the X coordinates of the vertices on the watermarked mesh $M_{watermark}$. v_x are the coordinates of the vertices on the original mesh *M*. ε is a user-provided global parameter that controls the energy of the watermark. Φ is a *nxm* matrix which has scalar basis functions ϕ^i_j as columns. d_i is a global displacement vector associated with vertices v_j. The matrix hd_x is a diagonal matrix which represents the weighted (by h_i) X coordinate of the displacement d_i. *w* is the watermark vector. Equation (6.2) shows that the effect of the watermark coefficient w_i is to perturb the each vertex v_j by a vector proportional to $w_i\phi^i_j d_i$. Meanwhile, the connectivity among the vertices does not change. Of course, the transformation needs to be small enough so that the appearance of *M* does not change visually. An example original mesh and its watermarked version are shown in Figure 6.15 (All the mesh models shown in this subsection are provided by Dr. H. Hoppe of Microsoft Research).

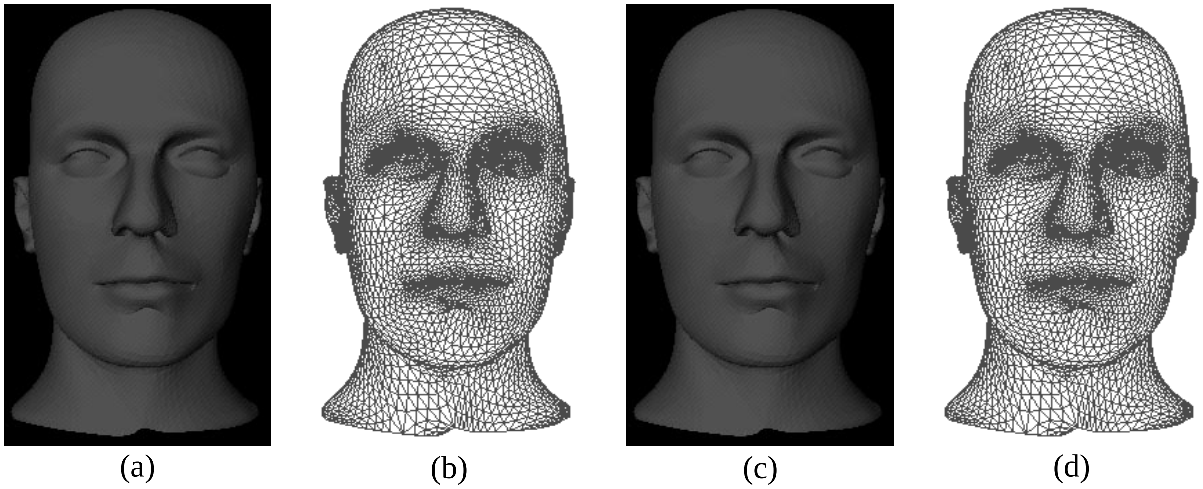


Figure 6.15: An example mesh (a), (b) and its watermarked version (c), (d).

When a suspect mesh $M_{suspect}$ is obtained, it is first registered to the original mesh M in the same coordinate frame. Then $M_{suspect}$ is resampled so that the resultant mesh $M^r{}_{suspect}$ has the same connectivity as that of the original mesh M. At the next step, the difference between the 3D coordinates of each pair of the corresponding points on $M^r{}_{suspect}$ and M is taken. The watermark vector *w'* is extracted from the above residuals. At the final step, a statistical method is used to determine whether *w* and *w'* are the same.

It can be seen from the above watermarking algorithm that the registration of the suspect mesh $M_{suspect}$ to the original mesh M is crucial to the extraction of the watermark from the suspect mesh. If the registration is not accurate enough, then the extracted watermark $w^{'}$ will not be accurate enough in order to match the true watermark *w*. In [62], the registration algorithm proposed in [14] which required good initial estimate of the underlying transformation was used. Some user intervention was also necessary as reported in [62].

Because the registration problem in watermarking triangular meshes is actually a special case of 3D surface matching, Harmonic Shape Images can be used to find the global transformation, followed by refinement using the ICP algorithms. An example is shown as follows.

An original mesh is shown in Figure 6.16(a) and (b). Figure 6.16(c) is an suspect mesh. Compared with the original mesh, the suspect mesh is simplified, cropped and transformed using a rigid transformation. Some of the randomly selected vertices on the original mesh and the suspect mesh are overlaid on both meshes (Figure 6.16(d), (e)). Using these vertices as centers and specifying the radius equal to 50.0, surface patches along with their Harmonic Shape Images are computed. Among all these, the best-matched patches are shown in Figure 6.16(f) and (g). Using that pair of patches, the transformation between the original and the suspect meshes is first computed and then refined using ICP. The suspect mesh is aligned with the original mesh as shown in Figure 6.16(h) and (i).

The most meaningful way to check the registration accuracy in mesh watermarking is to extract the watermark from the aligned suspect mesh and compare the extracted watermark with the original watermark. The comparison result can tell us whether the registration is sufficiently accurate. Unfortunately, we do not have the software to perform the watermark extraction. Therefore, the following method is used to check the registration accuracy.

Suppose that rigid transformation is the last operation that is applied to the original mesh to obtain a suspect mesh. Without applying rigid transformation, the suspect mesh stays aligned with the original mesh. Although this alignment may not be the best one since the suspect mesh has already been changed from the original mesh to a different one, the alignment can be used as a starting point for the ICP algorithm to find a better alignment $T^{'}$ and a smaller alignment error $E^{'}$. For the suspect meshes which have gone through rigid transformations, the registration error E_T should be comparable to $E_T{}^{'}$. For the example shown in Figure 6.16, $E^{'}$ = 0.261163, and $E_T{}^{'}$ = 0.241831; they are fairly close.

Figure 6.17 shows another registration example for mesh watermarking. The original mesh is shown in Figure 6.17(a) and (b). Figure 6.17(c) shows the suspect mesh which is simplified, cropped and transformed using a rigid transformation. Some of the randomly selected vertices on the original mesh and the suspect mesh are overlaid on both meshes (Figure 6.17(d), (e)). Using these vertices as centers and specifying the radius equal to 1.5, surface patches are generated and

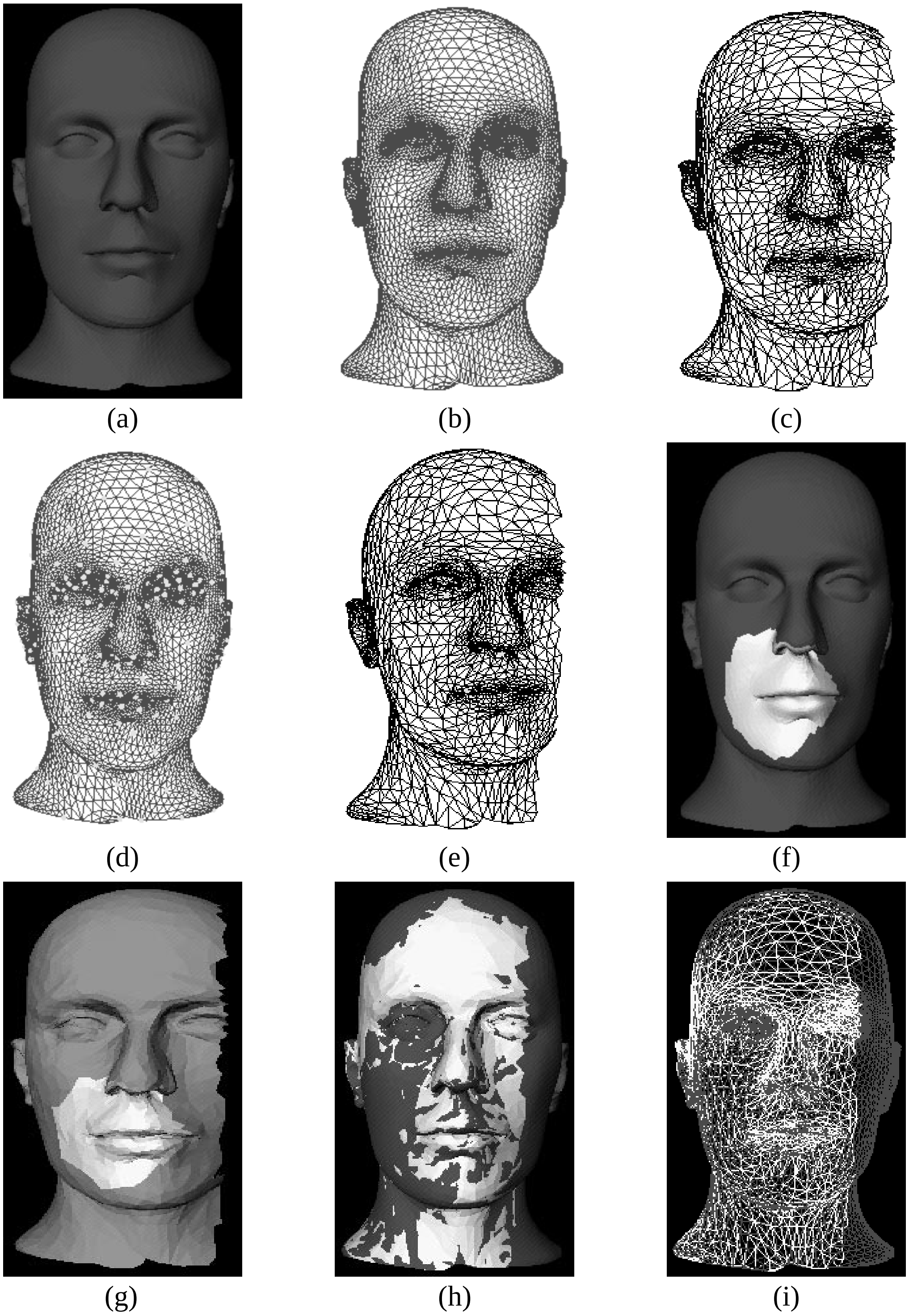


Figure 6.16: Harmonic Shape Images used in registration of the mesh watermarking process. (a), (b) Original mesh; (c) suspect mesh; (d) some of the randomly selected vertices on the original mesh; (e) some of the randomly selected vertices on the suspect mesh; (f), (g) the best matched patches overlaid on the original and the suspect meshes; (h), (i) the aligned suspect mesh with the original mesh.

their Harmonic Shape Images are then computed as well. Among all these patches, the best-matched patches are shown in Figure 6.17(f) and (g). Using that pair of patches, the transformation between the original and the suspect meshes is first computed and then refined using ICP. The suspect mesh is aligned with the original mesh as shown in Figure 6.17(h) and (i). The registration error for this example is 0.044202, as compared with the reference error 0.044516. Again, the two errors are close.

6.3 Accuracy Test of Surface Registration Using Harmonic Shape Images

One important issue with surface registration is the registration result’s degree of accuracy. Due to the difficulties in obtaining ground truth, even if the measurement of the registration accuracy can be conducted, the confidence of that accuracy still depends on how accurate the ground truth is. In order to conduct the best accuracy test using the hardware equipment available in our lab, the following experiments were designed.

An object with free-form surfaces is selected and markers are attached to the object. For example, Figure 6.18 shows a white bunny with red dots painted on it. Two range images along with texture images can be taken of the object from different viewing directions as shown in Figure 6.18. The markers are present in both range images (texture images as well) because the two range images overlap. Therefore, the fact that those markers are the same physical points in 3D space can be used as ground truth to measure the registration accuracy as long as those markers can be identified in the two range images.

The two texture images taken by the sensor are used to locate the markers in the range images. The process is as follows. At the first step, the corresponding red markers are manually picked in the two texture images. The texture coordinates of those markers are recorded. Then, by searching for the same texture coordinates as the markers among the vertices in the two range images, the corresponding pair of points in 3D space can be located. Those pairs of 3D points are used as ground truth to test the registration accuracy.

A few practical issues about the above operation need to be discussed. These issues are directly related to how accurate the ground truth is. First, each red marker in the texture image maps to a few pixels instead of to only one pixel. Second, the resolution of the texture image is much higher than that of the range image. For example, the resolution of the texture images in Figure 6.18 is 400x400, and the number of vertices in the range images in Figure 6.18 is fewer than 4500 in both cases. What the second issue means is that, for any particular pixel in the texture image, there may not be a corresponding vertex in the range image. In this case, the fact that the size of each marker is a few pixels can help us find a corresponding vertex in the range image for each marker in the texture image. Figure 6.19 further illustrates this idea.

In Figure 6.19, $N(c, m)$ is a small neighborhood in a texture image that represents a marker. c, selected mannually, is the center of the neighborhood. The size of $N(c, m)$ is $(2m+1)$x$(2m+1)$ in which m is determined according to the actual size of the marker. For each pixel in $N(c, m)$, its correspondence is searched through the texture coordinates of all the vertices in the range image. As a result, a few vertices may be found as shown in Figure 6.19. Those vertices are in a small neighborhood denoted by $N^{'}$. After $N^{'}$ is located, the average of all the vertices in $N^{'}$ is used as the correspondence of the marker in the texture image.

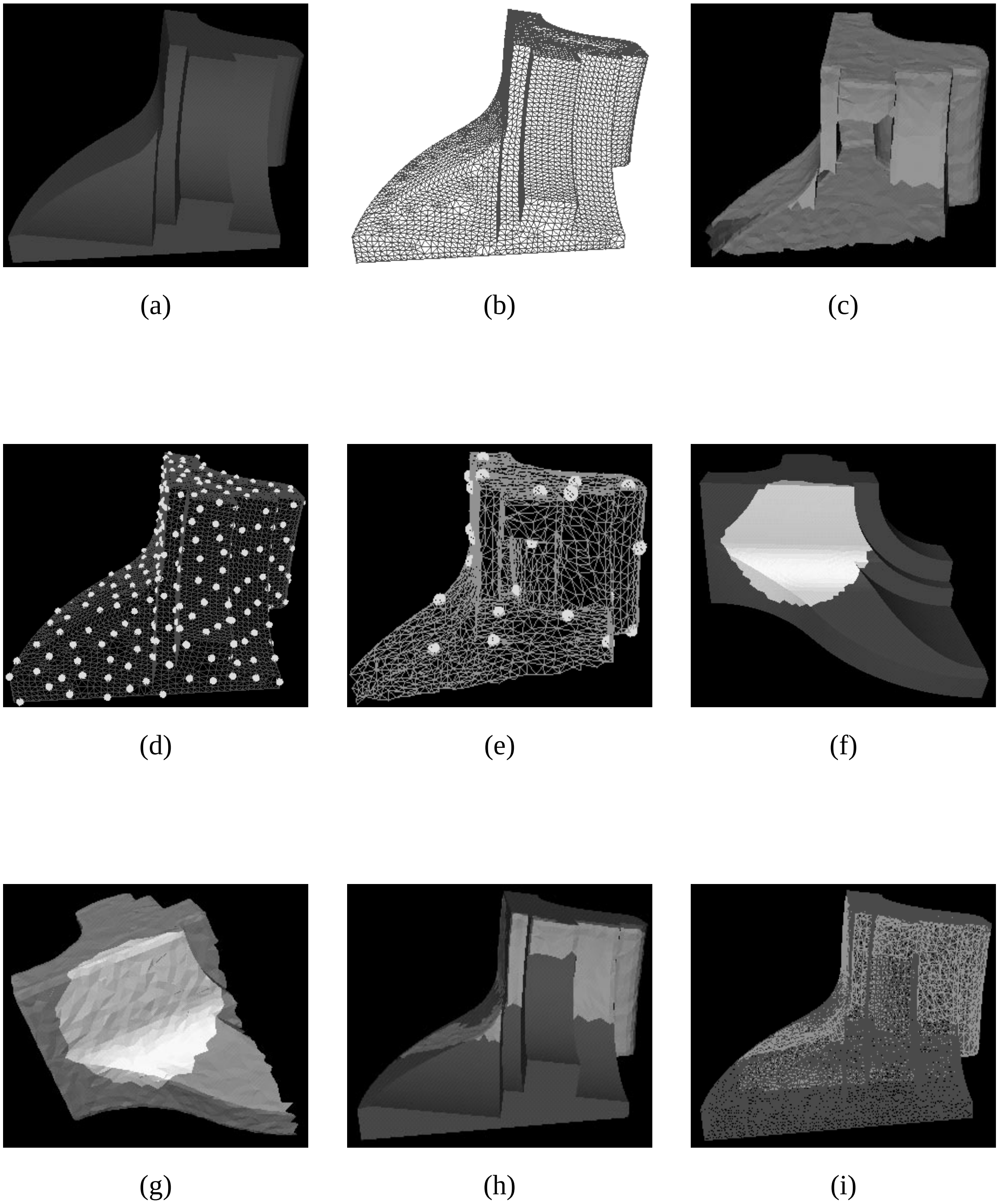


Figure 6.17: Harmonic Shape Images used in registration of the mesh watermarking process. (a), (b) original mesh; (c) suspect mesh; (d) some of the randomly selected vertices on the original mesh; (e) some of the randomly selected vertices on the suspect mesh; (f), (g) the best matched patches overlaid on the original and the suspect meshes; (h), (i) the aligned suspect mesh with the original mesh.

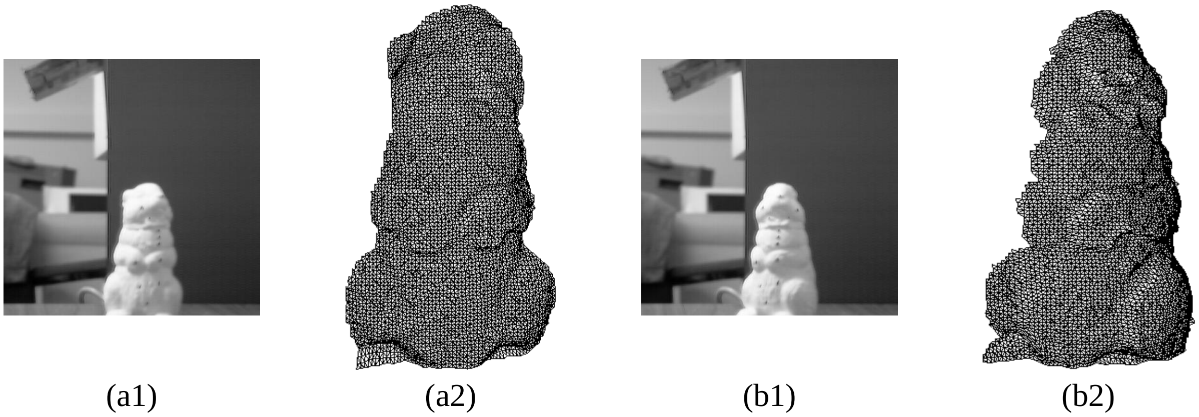


Figure 6.18: The texture images of the same object taken from two different viewing directions. The red dots on the object are used as markers for obtaining group truth.

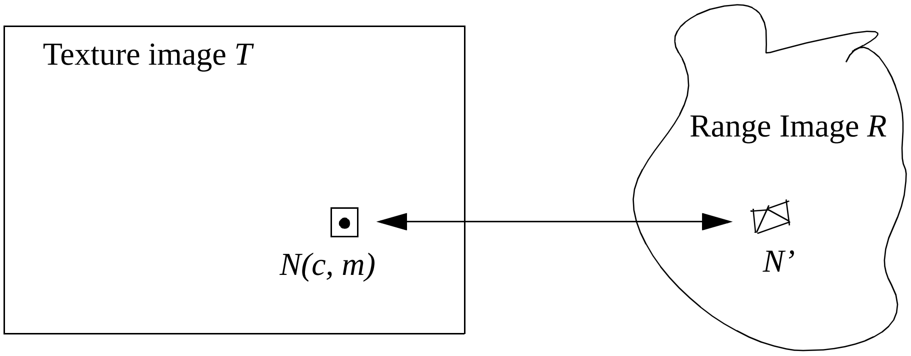


Figure 6.19: Illustration of how to find correspondences in a range image for the markers in the texture image.

Compensating for the Center Location Error

The above description shows that the manual location of the center of each marker is crucial for finding the correct correspondence in the range image. However, this step could introduce some error even when the center is only one pixel off. To compensate for the possible center location error, the following scheme is applied. Assume the upper bound of the center location error is p pixels. Given a manually selected center c, generate multiple centers $c(i, j)$ using every pixel in the $(2p+1)$x$(2p+1)$ window centered at c. Then for each center $c(i, j)$, get its $(2m+1)$x$(2m+1)$ neighborhood and find its correspondence vertex $v_{c(i, j)}$ in the range image. After the two range images are registered, compute an error vector for each pair of the ground truth points. Since multiple pairs of ground truth points have been generated for each marker, there are multiple error vectors for each marker. Among those error vectors, the one with the smallest norm is selected to be the error vector. This gives an upper bound of the registration accuracy for that marker given the upper bound of the center location error in the texture image.

According to the strategy discussed above, the result of the registration accuracy test for the example in Figure 6.18 is presented below. Table 6.2 shows the manually selected centers for the 9 markers in the texture images shown in Figure 6.18. Considering the fact that the number of pixels is about 28,000 (for the object only in the texture image) and the number of vertices in the range image is about 4,500, on average, there are about 6 pixels in the texture image corresponding to one vertex in the range image. Therefore, the size of *N(c, m)* is selected to be 9 with *m=1*. The corresponding vertices found for each marker are shown in Figure 6.20.

Table 6.2: The texture coordinates of the manually selected centers for the markers

Marker Index	Coordinates in Texture Image T_1	Coordinates in Texture Image T_2
1	(216, 232)	(190, 232)
2	(248, 214)	(217, 215)
3	(228, 248)	(201, 248)
4	(243, 268)	(215, 267)
5	(244, 277)	(214, 279)
6	(243, 289)	(213, 289)
7	(212, 355)	(186, 354)
8	(244, 351)	(213, 350)
9	(224, 382)	(195, 380)

Assuming the upper bound for center location error on the texture image is *p=1* pixel, the registration error for each marker is shown in Table 6.3. Each value is the norm of the error vector between the corresponding markers in the two range images. The unit of the measurements is the millimeter. The average registration error is 0.3986. The resolution of a range image is defined to be the mean of the edge lengths. For the two range images in Figure 6.18, the resolution is about 2mm. Therefore, the relative registration error is one-fifth of the mesh resolution.

Table 6.3: Registration error for each marker

Marker	1	2	3	4	5	6	7	8	9
Error	0.3177	0.4194	0.0909	0.3471	0.6252	0.1797	0.4811	0.7983	0.3280

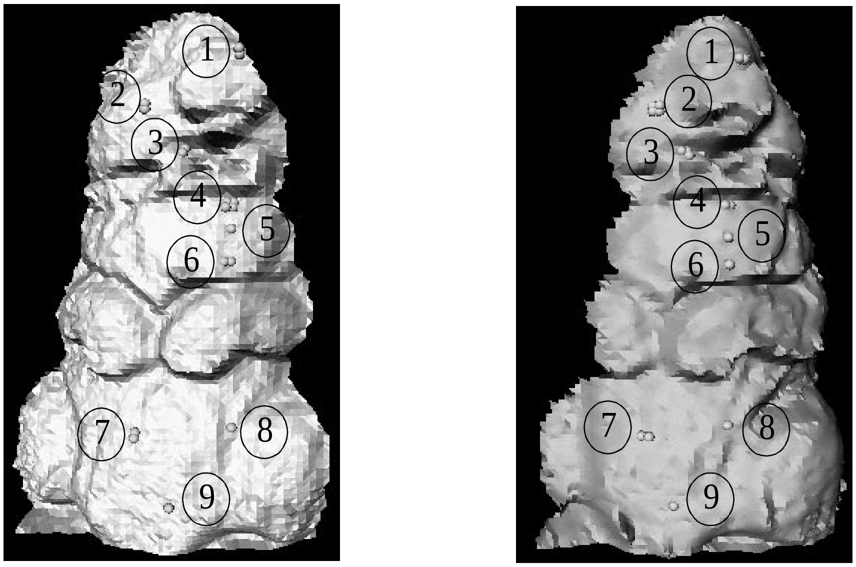


Figure 6.20:Corresponding markers on the two range images which were taken from different viewing directions.

6.4 Object Recognition in Scenes with Occlusion

In this section, Harmonic Shape Images are used to recognize objects in scenes with occlusion. The goal of the experiments is to demonstrate the usefulness of Harmonic Shape Images in this application. The recognition strategy in this section is for experimental purposes although it could be incorporated into a complete recognition system in the future. In the following, the construction of the object library will be introduced first, followed by the explanation of the hypothesize-and-test recognition strategy. Then the experimental results will be presented along with an analysis of those results.

Object Library

The object library consists of four objects as shown in Figure 6.21, where only the front part of each object is used to create the models. This is because there is not much shape variation in other parts of each object. Some vertices on each object are randomly selected as central vertices to generate model surface patches. Those selected vertices are distributed as uniformly as possible on each object. It is not necessary to create surface patches for all the vertices on each object because the sampling resolution is high and surface patches created by neighboring vertices are similar in shape. Some of the central vertices of the model surface patches are overlaid on each object and shown in Figure 6.21(a3), (b3), (c3) and (d3). The radius is selected to be 25.0. The area of each model surface patch is about 10 to 15 percent of the area of each object. Harmonic Shape Images are generated for each of the model surface patches and are stored in the object library.

It should be noticed that the objects shown in Figure 6.21(a1), (b1) and (d1) are similar in shape. Compared with using objects of distinct shape in the recognition experiment, this similarity makes the recognition tasks more difficult. The success of the experiment will not only demonstrate the application of Harmonic Shape Images in object recognition, but will also justify its discriminability of different shapes, even for surfaces without much shape variation.

Scene Processing

Figure 6.22 shows a scene with occlusion. As shown in the texture image in Figure 6.22(a), the background of the scene is black. This helps the range finder in our lab increase the depth of the view. The meshes in Figure 6.22(c) and (d) show that, although the object on the left is occluded by the right object, the two objects do not touch each other. Scene processing is a hypothesize-and-test process in which a decision is made about each connected region in a scene on whether it matches any of the objects in the library. In this section, the steps for hypothesis generation are first discussed, followed by discussion of the test process.

Hypothesis Generation The first step in hypothesis generation is to apply smoothing operation on the scene in order to remove sampling noise. Then, some vertices are randomly selected in the scene mesh to generate scene surface patches. Because of the high resolution of the mesh in the scene, it is not necessary to generate a large amount of surface patches. Some of the selected vertices are shown in Figure 6.22(b). Using them as central vertices and specifying the radius equal to 25.0, the scene surface patches are created and their Harmonic Shape Images are generated. At the third step, the scene Harmonic Shape Images are compared with each set of model Harmonic Shape Images in the library.

Recall that Harmonic Shape Images can be created only on connected surfaces. This characteristic means that in the application of object recognition, each connected region in the scene should be examined independently. Matching hypotheses should be generated for each connected region according to its matching scores to the objects in the library. Table 6.4 summarizes the matching statistics for all the connected regions in the scene in Figure 6.22.

.As shown in the last row of Table 6.4, the number of selected scene surface patches is 146. The total number of matched scene patches to each set of model patches is shown as the numerator in the last row of Table 6.4. For each connected region, the number of matched patches to each model and the number of selected patches are listed in the first and second rows. The central vertices of those matched scene and model patches are shown in Figure 6.24.

For each connected region and a model in the library, the generation of the matching hypothesis is determined by the ratio of n_r^m and n_r in which n_r^m is the number of matched surface patches to the model and n_r is the number of selected patches in that region. This ratio is called matching score and denoted as $\lambda(r_i, m_j)$ in which r_i denotes the *ith* region in the scene and m_j denotes the *jth* model in the library. When $\lambda(r_i, m_j)$ is above a threshold λ_{cutoff}, a hypothesis is generated on the matching of region *i* and model *j*.

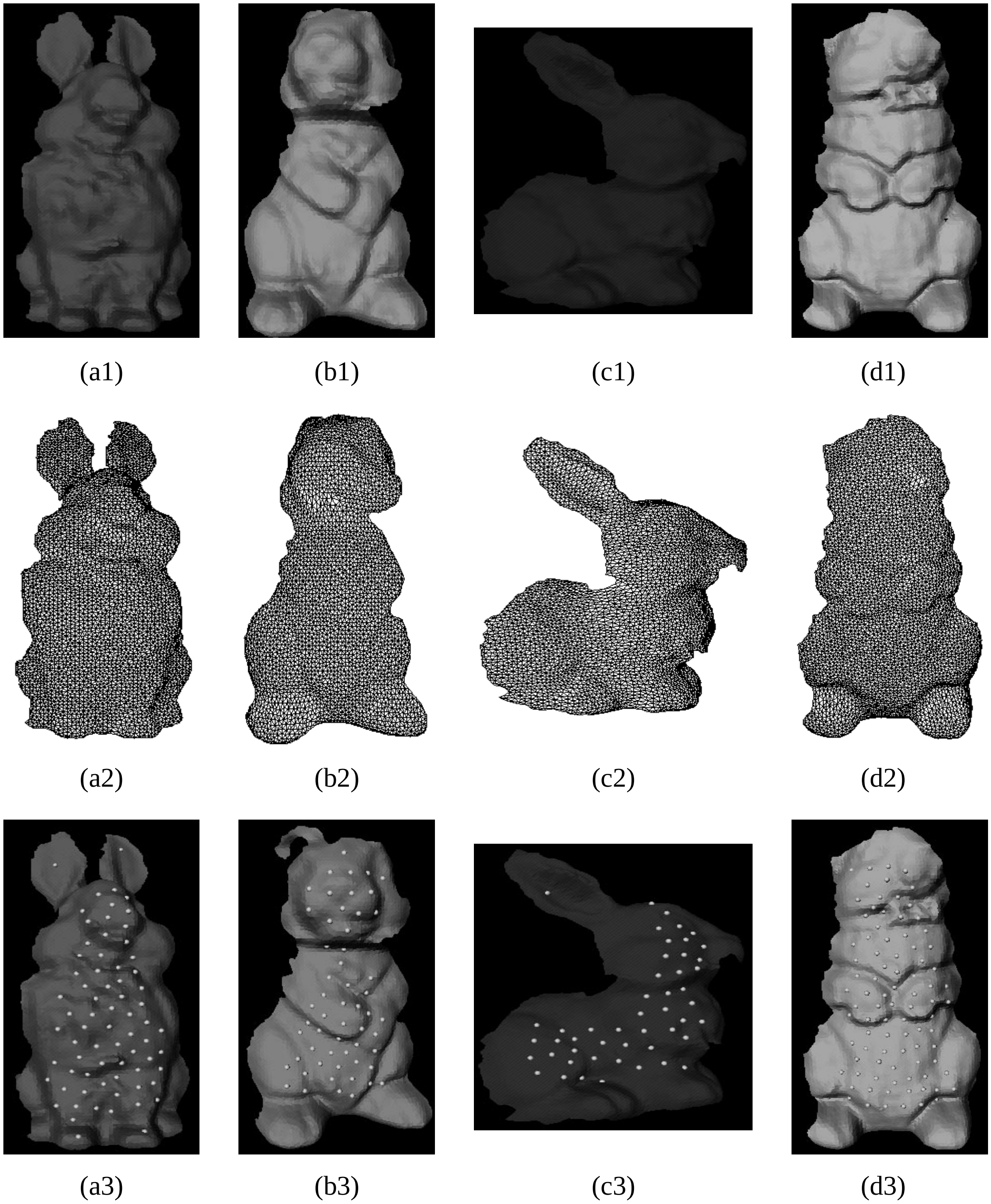
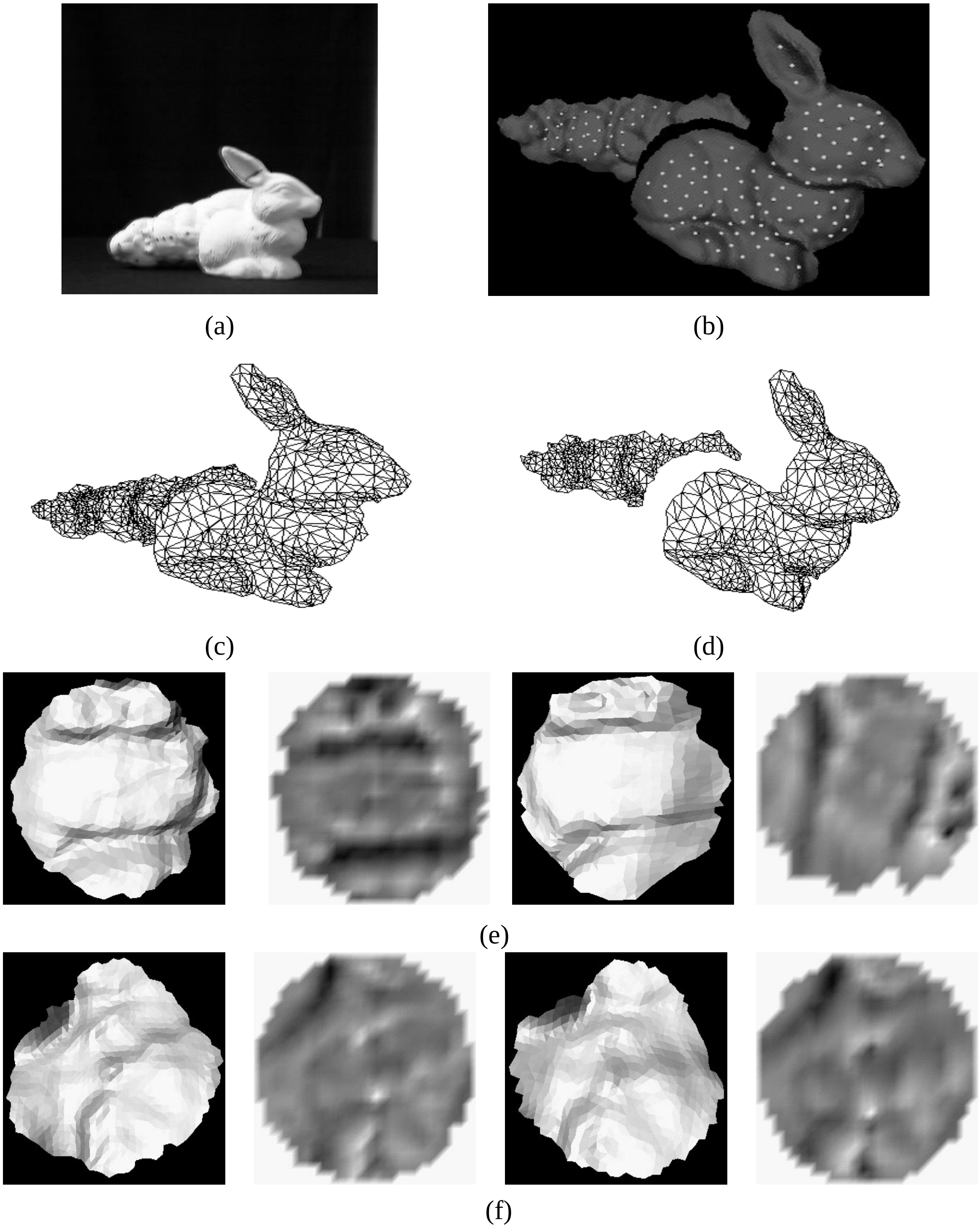


Figure 6.21: Objects in the model library for the recognition experiments.



(a) (b)

(c) (d)

(e)

(f)

Figure 6.22: The first scene for the recognition experiment. (a) Texture image; (b) rendered mesh with the central vertices of the scene surface patches; (c) wireframe of the front view; (d) wireframe of the side view. The wireframe versions are meshes after simplification, which is for display purposes only. The original meshes are of high resolution; (e) best-matched patches on the left region in the scene and model 4; (f) best-matched patches on the right region in the scene and model 3.

Table 6.4: Summary of the Recognition Experiment for Scene 1

Region/Model	Model1	Model2	Model3
Region1	2/37	0/37	1/37
Region2	<u>13/109</u>	5/109	<u>82/109</u>
Total Matches	15/146	5/146	83/146

Selecting λ_{cutoff} The selection of λ_{cutoff} depends on two considerations. The first one comes from the local property of Harmonic Shape Images, which are local representations of surface patches. In general, they can describe the shape of only part of a given surface. Considering the fact that two surfaces of different shapes may be similar at a few regions, using only one pair of matched surface patches is not reliable in determining whether those two surfaces are similar. Therefore, for each connected region and a model in the library, among the selected surface patches in the region there should be a fair amount of matched patches to the model. This consideration determines the lower bound of λ_{cutoff}.

The second consideration when selecting λ_{cutoff} comes from the fact that there may be clutter in the scene. Clutter can result in a connected region with different objects. Under this situation, the matched patches in the region correspond to different models in the library. Therefore, for any one of those models, the matching score of the region, $\lambda(r_i, m_j)$, may not be as high as that when the connected region consists of only one object. This consideration poses the upper bound for λ_{cutoff}.

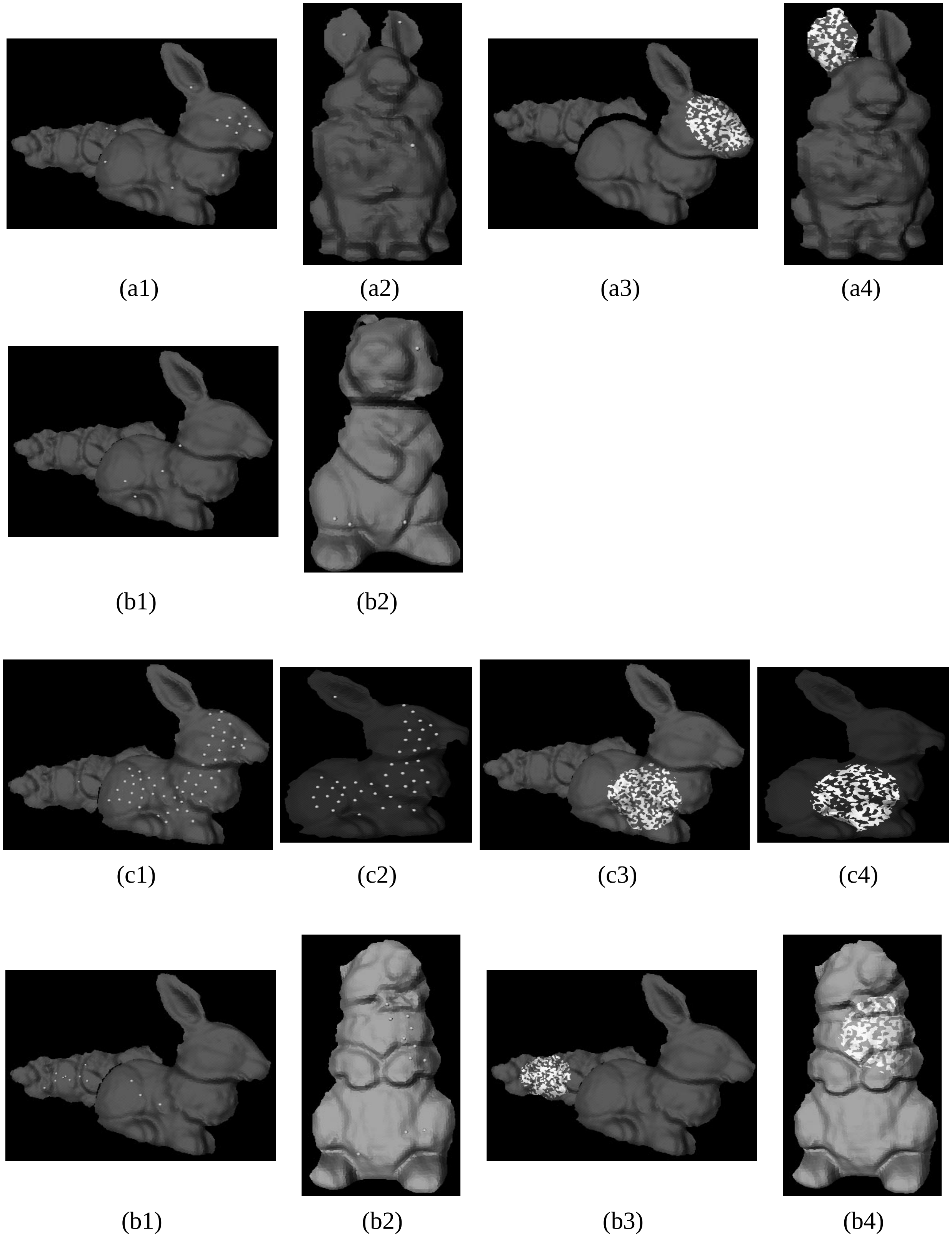
Based on the above two considerations, λ_{cutoff} is set to 10% empirically. In fact, this value is intentionally set low so as not to miss potential matches.

After using λ_{cutoff} equal to 10% to threshold the matching scores in Table 6.4, three hypotheses can be made about the region-model matches (the values greater than λ_{cutoff} are underlined in Table 6.4). Denote the connected region on the left in the scene (Figure 6.22) as region 1 and the one on the right as region 2, the three hypotheses are as follows:

1. Region 1 matches model 4
2. Region 2 matches model 1
3. Region 2 matches model 3

Among the above three hypotheses, the second one is generated because λ_{cutoff} is set low. The verification process would show that this hypothesis will be rejected.

Hypothesis Test The test process consists of four steps. The first is to compute the transformation between the matched region and the model. The transformation is then applied so that both of scene and the model are in the same coordinate system. The second step is to use the Iterative Closest Point (ICP) algorithm to refine the transformation obtained in the first step. At the third step, the hypothesis is accepted or rejected based on the alignment error computed by the ICP



(a1) (a2) (a3) (a4)

(b1) (b2)

(c1) (c2) (c3) (c4)

(b1) (b2) (b3) (b4)

Figure 6.23: Matching results of scene 1 to each object in the library.

algorithm. If the hypothesis is accepted, then at the fourth step, the distance to the model surface is computed for each vertex in the scene region. Each distance value is thresholded in order to determine whether it is on the model surface. This is actually a segmentation step which segments the matched scene object from a cluttered scene.

Following the above four steps, the test result of the scene in Figure 6.22 is shown in Figure 6.24. The alignment results obtained by using Harmonic Shape Images are shown in Figure 6.24(a1), (b1) and (c1). The refined alignment results are shown in (a2), (b2) and (c2). The alignment error after the ICP for each hypothesis is listed in Table 6.5. The alignment error for the second hypothesis is significantly larger than that of the other two. In fact, the threshold for the alignment error is selected to be half of the resolution of the mesh in the scene. In this case, it is 0.867135. Therefore, the first and third hypotheses are accepted and the second one is rejected. Models 1 and 4 are aligned with the scene as shown in Figure 6.25.

Table 6.5: Alignment Error for Each Hypothesis in Scene1

Hypothesis	1	2	3
Error	0.160898	3.02334	0.439101

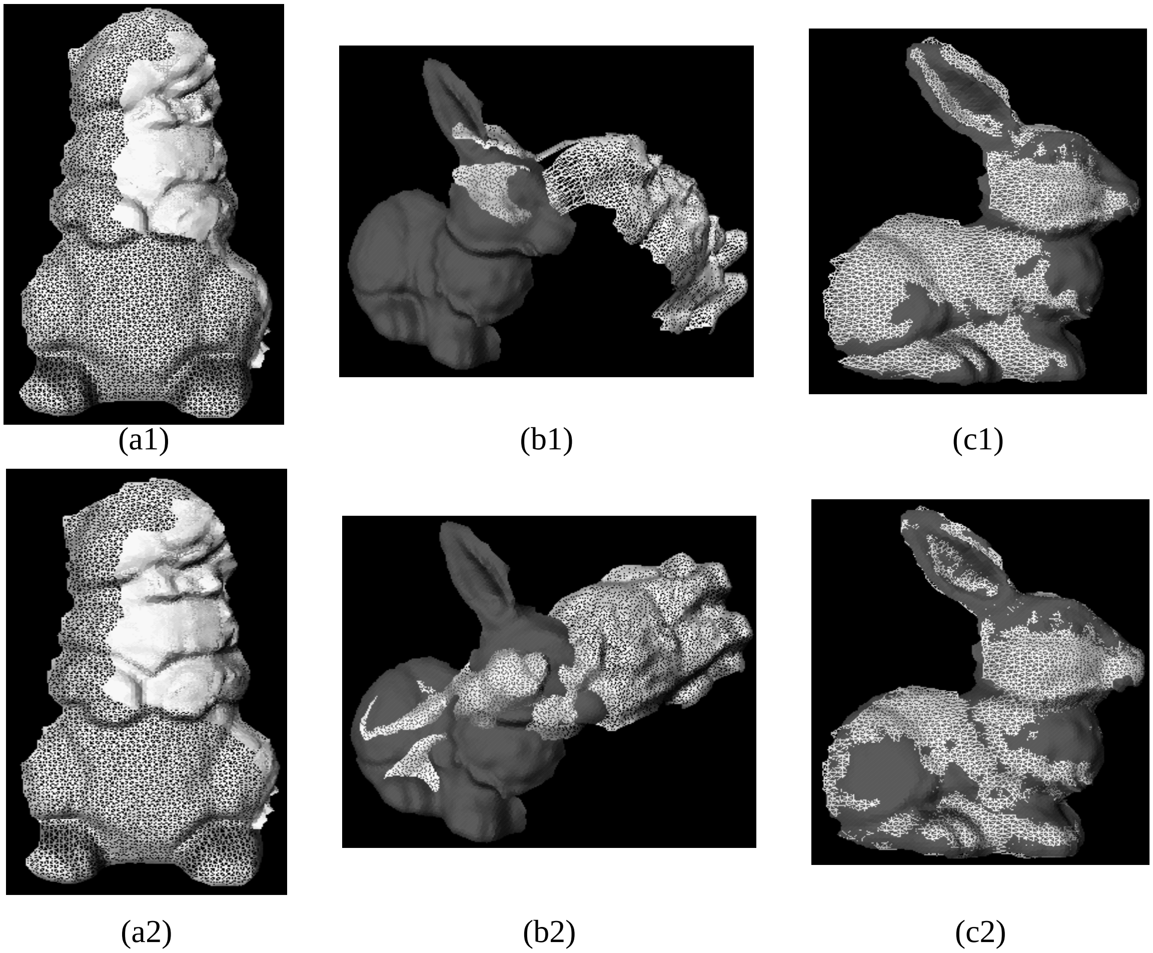
Table 6.6: Matching Scores for Scene2

Region/Model	Model1	Model2	Model3	Model4
Region1	1/57	3/57	2/57	<u>11/57</u>
Region2	3/60	<u>43/60</u>	0/60	2/60
Region3	0/54	2/54	<u>41/54</u>	6/54
Total Matches	4/171	48/171	43/171	19/171

Table 6.7: Alignment Errors for Regions in Scene2

Region	1	2	3
Error	0.783502	0.190562	0.382581

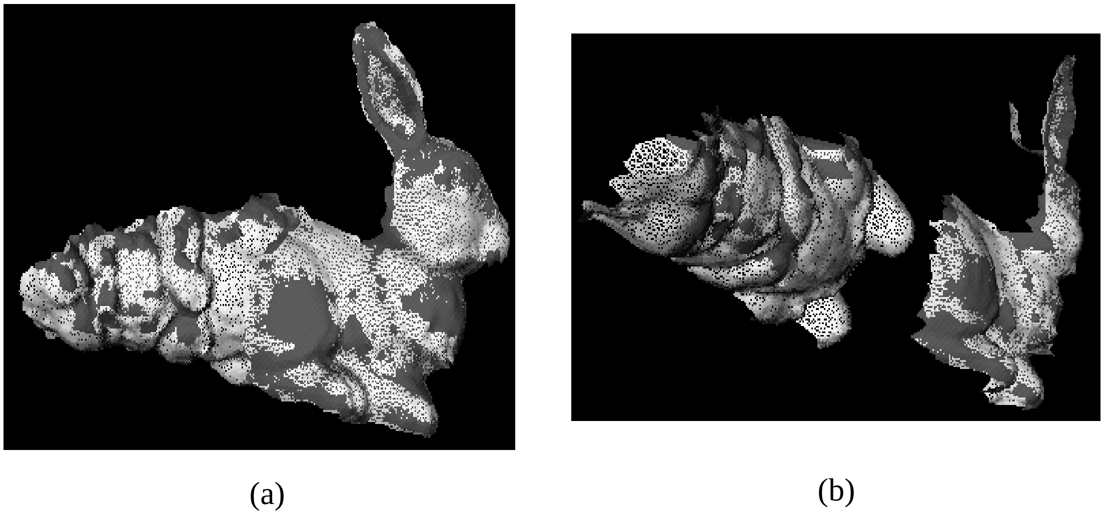
Two other recognition examples are shown in Figure 6.26 and Figure 6.27. The regions in each scene are numbered from left to right sequentially. The matching scores are listed in Table 6.6 and Table 6.8 with the values greater than λ_{cutoff} underlined. Table 6.7 and Table 6.9 list the alignment errors for both scenes. The mesh resolutions for the two scenes are 1.84543 and 1.8381, respectively. In both cases, all the connected regions are recognized correctly and all the alignment errors are less than half of the mesh resolution.



(a1) (b1) (c1)

(a2) (b2) (c2)

Figure 6.24: Verification of the three hypotheses for the scene in Figure 6.22. (a1), (b1), (c1) show the alignment results obtained by using Harmonic Shape Images; (a2), (b2), (c2) show the refined alignment results obtained by using ICP.



(a) (b)

Figure 6.25: Recognition result of the scene in Figure 6.22. The recognized models are aligned with the objects in the scene. (a) Front view; (b) side view.

Table 6.8: Matching Scores for Scene3

Region/Model	Model1	Model2	Model3	Model4
Region1	<u>19/41</u>	0/41	0/41	2/41
Region2	1/82	4/82	11/82	<u>81/82</u>
Region3	1/43	<u>34/43</u>	2/43	1/43
Total Matches	21/166	38/166	13/166	84/166

Table 6.9: Alignment Errors for Regions in Scene3

Region	1	2	3
Error	0.391074	0.15994	0.250005

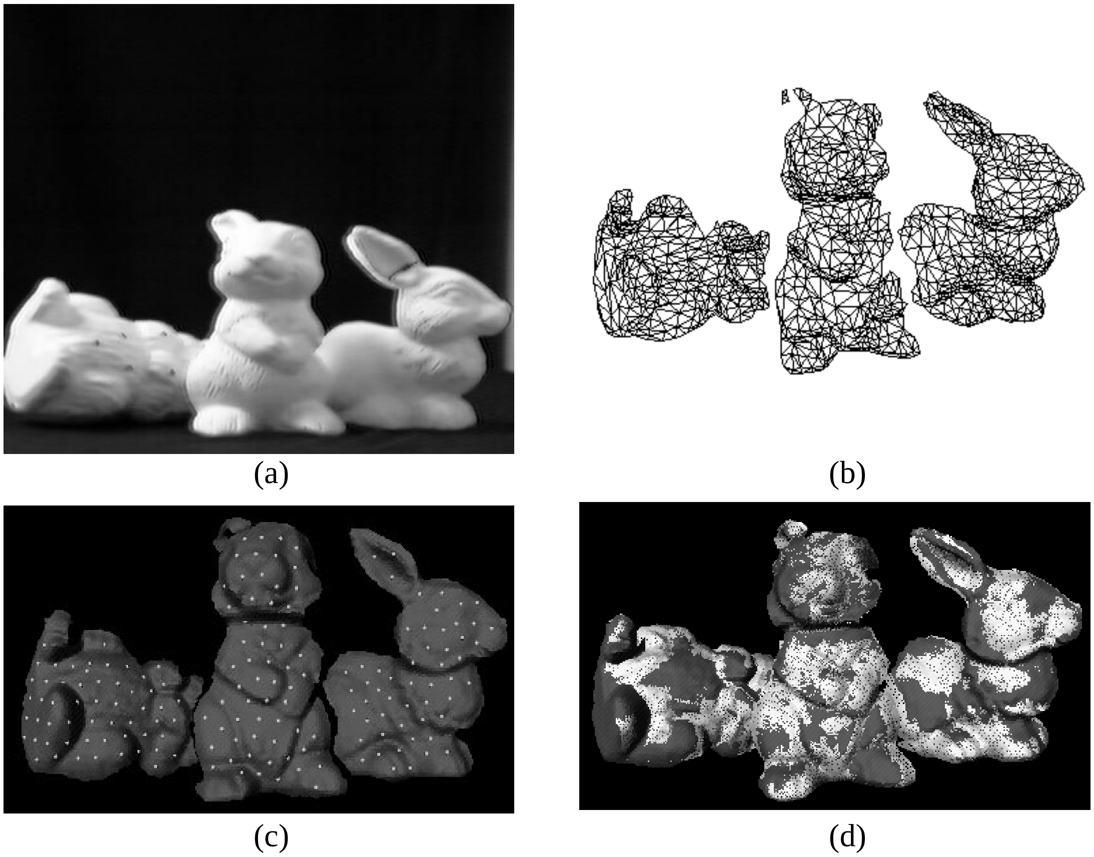
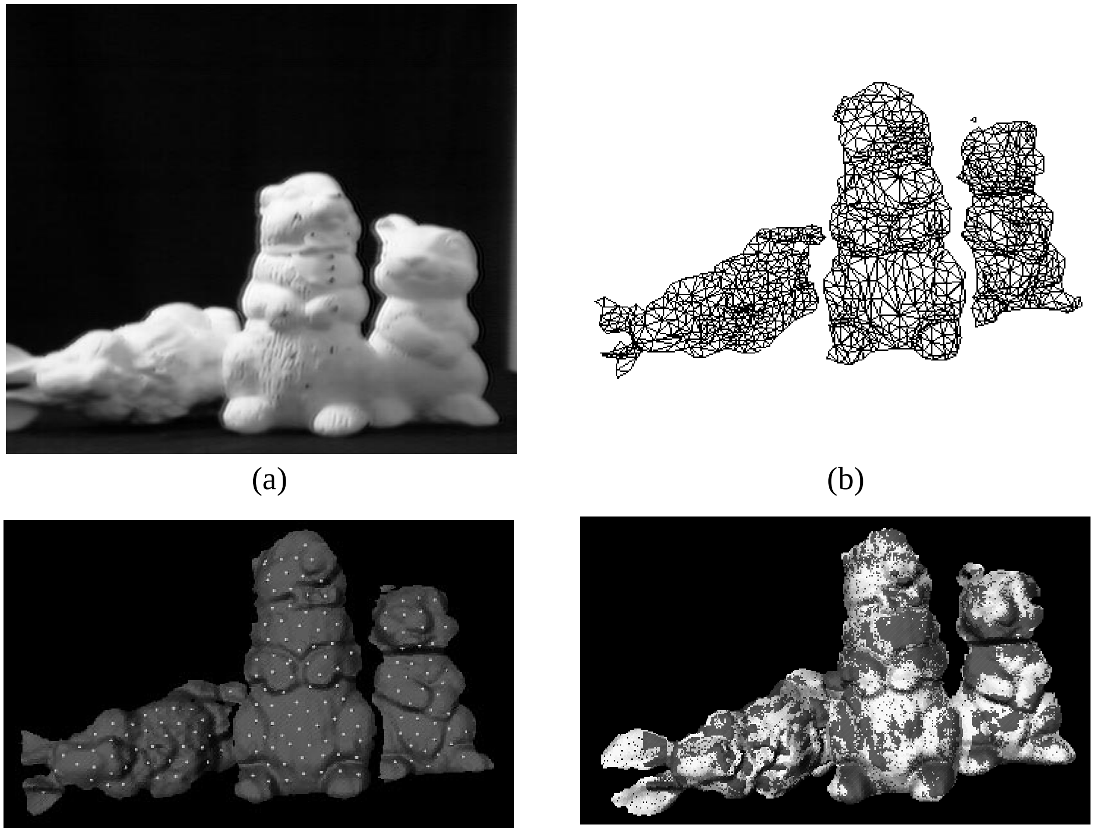


Figure 6.26: Recognition example. (a) The texture image of the scene; (b) the mesh of the scene in wireframe; (c) the central vertices of the selected scene surface patches; (d) the recognized models aligned with their corresponding regions in the scene. The wireframe versions are meshes after simplification, and are for display purposes only. The original meshes are of high resolution.



(a) (b) (c) (d)

Figure 6.27: Recognition example. (a) The texture image of the scene; (b) the mesh of the scene in wireframe; (c) the central vertices of the selected scene surface patches; (d) the recognized models aligned with their corresponding regions in the scene. The wireframe versions are meshes after simplification, and are for display purposes only. The original meshes are of high resolution.

A fourth scene is shown in Figure 6.28. The matching scores and alignment errors are listed in Table 6.10 and Table 6.11, respectively. The first region in the scene does not match any of the models in the library, although it is supposed to match the fourth one. In this case, not being able to match scene and model is due to the lack of features in the first region. In fact, the upper part of model 4 which consists of rich shape information, is occluded by the second object in the scene. Except for the first region, the other three regions are correctly recognized, and all the alignment errors are less than half of the mesh resolution which is 1.78922.

Table 6.10: Matching Scores for Scene4

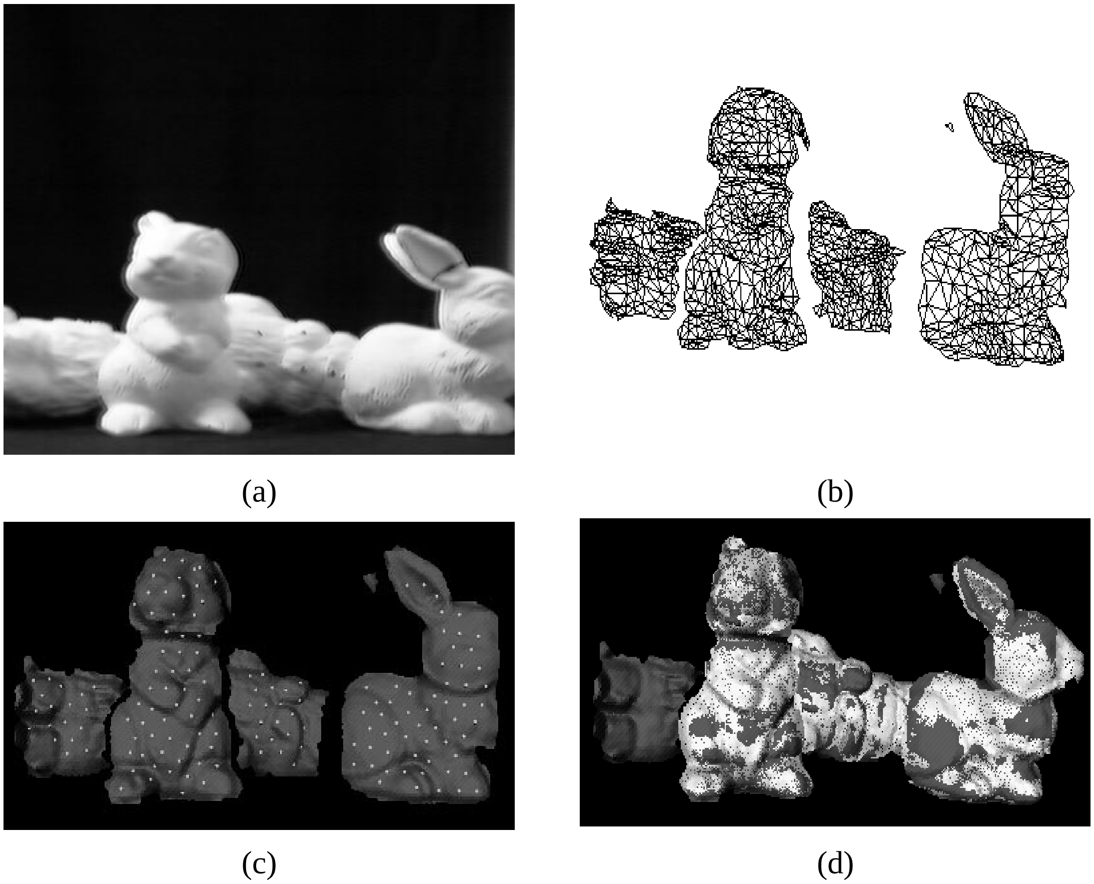
Region/Model	Model1	Model2	Model3	Model4
Region1	0/18	0/18	0/18	0/18
Region2	0/59	42/59	1/59	1/59
Region3	0/22	2/22	0/22	13/22

Table 6.10: Matching Scores for Scene4

Region/Model	Model1	Model2	Model3	Model4
Region4	1/58	4/58	25/58	4/58
Total Matches	1/157	48/157	26/157	18/157

Table 6.11: Alignment Errors for Scene4

Region	1	2	3	4
Error	-	0.12362	0.37545	0.425999



(a)

(b)

(c)

(d)

Figure 6.28: Recognition example. (a) The texture image of the scene; (b) the mesh of the scene in wireframe; (c) the central vertices of the selected scene surface patches; (d) the recognized models aligned with their corresponding regions in the scene. The wireframe versions are meshes after simplification, and are for display purposes only. The original meshes are of high resolution.

A fifth scene is shown in Figure 6.29. The interesting point about this scene is that the second object in the scene does not belong to the object library. The recognition result shows that it was not mis-matched to any of the objects in the library. The other two objects are correctly identified and aligned. The resolution of the mesh in the scene is 1.9706.

Table 6.12: Matching Scores for Scene5

Region/Model	Model1	Model2	Model3	Model4
Region1	1/59	2/59	26/59	1/59
Region2	2/58	1/58	4/58	8/58
Region3	2/77	3/77	1/77	49/77
Total Matches	5/194	6/194	31/194	58/194

Table 6.13: Alignment Errors for Scene5

Region	1	2	3
Error	0.232193	-	0.329847

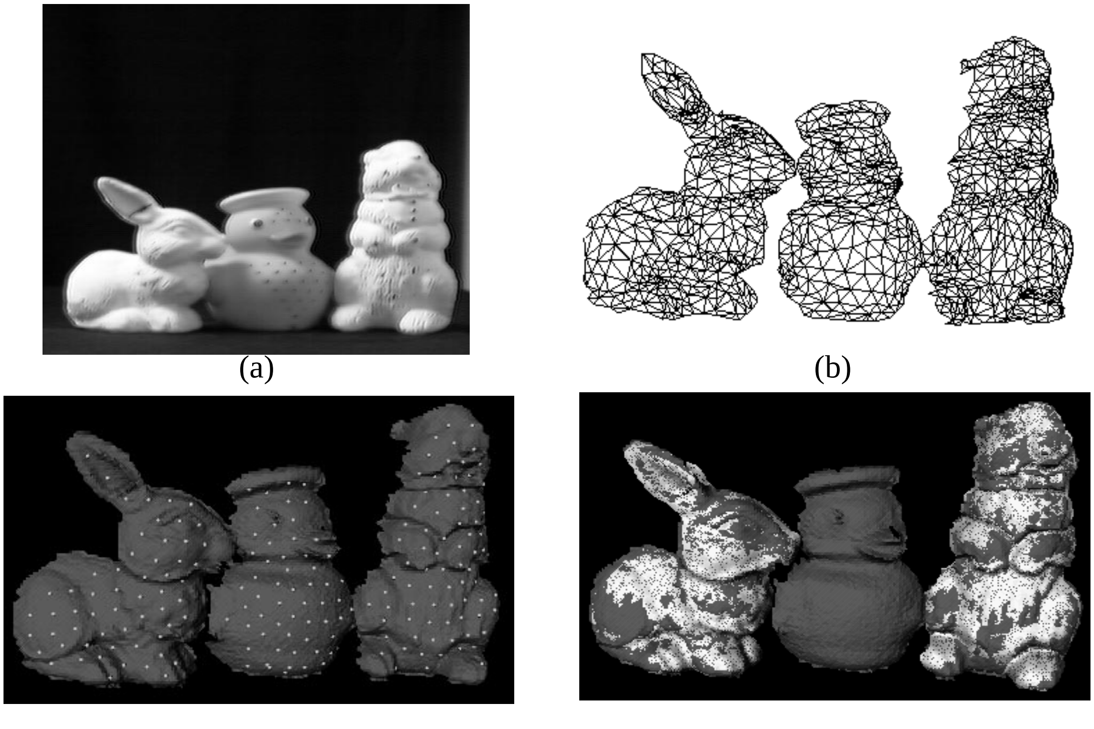


Figure 6.29: Recognition example. (a) The texture image of the scene; (b) the mesh of the scene in wireframe; (c) the central vertices of the selected scene surface patches; (d) the recognized models aligned with their corresponding regions in the scene. The wireframe versions are meshes after simplification, and are for display purposes only. The original meshes are of high resolution.

Chapter 7

Future Work

In this thesis, a new and novel representation for 3D free-form surfaces, which is called Harmonic Shape Images, has been proposed, analyzed and demonstrated by some practical applications. As part of the future work, some extensions of Harmonic Shape Images are first discussed in this chapter. These extensions further demonstrate the novel properties of Harmonic Shape Images and its advantages when being used in surface matching. The second part of the future work focuses on the application of Harmonic Shape Images in shape analysis which is not addressed in this thesis. As a geometric representation which preserves both the shape and continuity of the underlying surfaces, Harmonic Shape Images provide a powerful tool for performing shape analysis.

7.1 Extensions to Harmonic Shape Images

In addition to the properties of Harmonic Shape Images and the matching strategy using those images that have been discussed in the previous chapters, the following algorithmic additions could be used to further improve the efficiency of Harmonic Shape Images in surface matching.

Finding the Rotation

Recall the generation of Harmonic Shape Images in Chapter 2. Because of the one-degree freedom of the boundary mapping, Harmonic Shape Images could be different by a planar rotation for the same surface patch. In the current implementation, the rotation is found by the direct search in the angle space.

Different strategies could be used to speed up the rotation finding process. For example, an estimate of the rotation could be found using subsampled versions of the Harmonic Shape Images. The rotation could be refined at the full resolution of the Harmonic Shape Images. Another strategy would be to use a coarse-to-fine approach to search the angle space.

Cutting the Search Space

Performing correlation-based template matching is not the only way to compare two Harmonic Shape Images. In fact, given a scene Harmonic Shape Image and a set of model Harmonic Shape Images, the goal is to find the best match of the scene image among the model images. This means that eliminating the model images that are not similar to the scene image will definitely speed up the search process.

Due to the fact that Harmonic Shape Images are a geometric representation for 3D free-form surfaces, partial information about the underlying surfaces could be extracted from Harmonic Shape Images. For example, histograms of the normalized shape distribution functions could be constructed from those images. If two Harmonic Shape Images are quite different in the histogram, then they could be eliminated from the search space without further comparison.

Building a Surface Matching System

The experiments conducted in this thesis are for demonstration and verification of the properties and usefulness of Harmonic Shape Images in surface matching. Although the algorithms for generating and matching Harmonic Shape Images are available now, a complete system for surface matching has not yet been built up. More issues need to be addressed when integrating different programs into a complete system. For example, improving the efficiency of the system and automatically setting the thresholds in the matching process.

Harmonic Texture Images

In the current surface matching strategy using Harmonic Shape Images, only shape information is used. However, more surface information such as texture could easily be added to the matching process because Harmonic Shape Images are geometric surface representations which preserve both the shape and the continuity of the underlying surfaces. This representation provides a general framework for representing surface attributes such as shape and texture. Therefore, in addition to creating Harmonic Shape Images, Harmonic Texture Images (HTIs) can be created by storing the texture information on the harmonic images. HTIs are especially useful when comparing surfaces with few shape variations but rich texture variations. Unlike the previous surface representations that were designed only for representing shape, the generality of Harmonic Shape Images enables the easy and natural addition of various information into the surface matching process.

7.2 Shape Analysis and manipulation

As a representation for 3D free-form surfaces with novel properties, Harmonic Shape Images can be used to conduct shape analysis and manipulate shapes. In this section, the future work on applications of Harmonic Shape Images in shape similarity, surface reconstruction and shape synthesis is discussed.

Shape Similarity

Harmonic Shape Images are local surface representations; therefore, a single Harmonic Shape Image can describe only part of the shape of a given surface. This is in contrast to the global representations, which can represent the shape of the entire surface. However, those global representations usually have difficulties with objects of arbitrary topology. In this aspect, defining a statistical shape similarity measure using local representations could provide a way to compare surfaces of arbitrary topologies.

Given two surfaces represented by triangular meshes, the same number of most shape significant surface patches is selected on both of them according to their Harmonic Shape Images. Those selected patches are ranked according to their shape significance. By comparing two sets of Harmonic Shape Images with the same rank, a shape similarity measure can be obtained by combining the pair comparison results of those images. This shape similarity measure is consistent with the visual perception of human in the way that surfaces with similar most significant shape features are considered to be similar.

Surface Reconstruction

Similar to the Spherical Attribute Image (SAI) representation, it is possible to reconstruct the underlying surface given its Harmonic Shape Image and a boundary constraint. This reconstruction property comes from the fact Harmonic Shape Images preserve both the shape and the continuity of the represented surfaces.

Shape Synthesis

Harmonic Shape Images can be used in shape synthesis first in the shape space and then back to the range space. Synthesizing surfaces in the 3D space directly may be difficult due to the issues such as registration, scaling and different sampling resolutions. Harmonic Shape Images provide a way to synthesize shape in the shape space, or in the HSI space. Different Harmonic Shape Images can be morphed together using different combination functions. Then the corresponding surfaces in 3D space can be reconstructed based on the reconstruction property of Harmonic Shape Images.

Chapter 8

Conclusion

In this thesis, a novel geometric representation for 3D free-form surfaces is proposed. Due to the use of the mathematical theory on harmonic maps, the representation is named Harmonic Shape Images. The properties of those images are analyzed and applications of the images in surface matching are demonstrated. The contributions of this thesis are summarized as follows.

- Matching surfaces by matching surface patches

The idea of matching surfaces by matching surface patches is proposed and implemented in this thesis. Harmonic Shape Images, the novel representation for 3D free-form surfaces proposed in this thesis, are the key to the success of patch-based surface matching. Unlike point-based matching, patch-based surface matching uses both the shape and the continuity information of the underlying surfaces. Therefore, the establishment of the correspondences on the two surfaces becomes trivial after patch-based matching is done.

- Harmonic Shape Images are intrinsic to the shape of the underlying surfaces

Due to the energy functional used in construction of the mapping between a 3D surface patch and a 2D planar domain, Harmonic Shape Images are intrinsic to the shape of the underlying surface patches. Therefore, surface patches of different shape have different Harmonic Shape Images, i.e., the images are unique with respect to shape. This is an important property for surface representation to be used to identify surfaces of different shape.

- Harmonic Shape Images preserve the continuity of the underlying surfaces

Due to the way Harmonic Shape Images are constructed, the continuity of the source surface patch is preserved exactly on the Harmonic Shape Image. This means that there is one-to-one correspondence between the points on the surface patch and the points on the Harmonic Shape Image. In fact, the harmonic map is a parameterization of the surface patch onto the 2D planar domain. The property of continuity preservation facilitates the establishment of correspondences after two Harmonic Shape Images are determined to match each other. No heuristic-guided post-processing is needed in order to organize the matched pair of points into mutually consistent correspondences.

- Harmonic Shape Images provide a general framework for representing surface attributes

Harmonic Shape Images are a geometric representation which preserves both the shape and continuity of the underlying surfaces. Therefore, any distribution functions defined on a surface patch can be defined on its Harmonic Shape Image as well. In addition to shape, other distribution functions such as surface normals, colors, texture, material, can all be represented in the same way as Harmonic Shape Images. This general framework makes the use of other surface attributes, e.g., texture, in surface matching easy and natural.

- Harmonic Shape Images are robust to surface sampling resolution and occlusion

Harmonic Shape Images' ability to deal with surface sampling resolution lies in the property that those images are intrinsic to the underlying surfaces. As long as the sampling resolution is high enough to represent the underlying surfaces, the corresponding Harmonic Shape Images are accurate enough as well. Due to the way the boundary mapping is constructed, Harmonic Shape Images are robust to occlusion. Experimental analysis has shown that the matching performance of Harmonic Shape Images decreases gracefully as the sampling resolution decreases and the occlusion increases.

- Harmonic Shape Images are both discriminative and stable

Being an effective surface representation, Harmonic Shape Images are discriminative enough to distinguish among surface patches of different shapes. Meanwhile, their stability ensures that surface patches with similar shapes have similar Harmonic Shape Images. This property is especially useful when applying the coarse-to-fine strategy to speed up the surface matching process. Experimental analysis has been conducted to verify the discriminability and stability of Harmonic Shape Images.

- The use of Harmonic Shape Images has been demonstrated in different applications of surface matching

Harmonic Shape Images have been used in varying applications of surface matching including view registration and object recognition in scenes with occlusion. Those extensive experiments show that, as a novel surface representation, Harmonic Shape Images not only have nice properties, but they can also solve real world surface matching problems.

Bibliography

[1]A. Aho, J.E. Hopcroft and J.D. Ullman, *Data structures and algorithms*, Addison-Wesley, Reading, Mass., 1983.

[2]A. D. Aleksandrov, V.A. Zallgaller, *Intrinsic geometry of surfaces*, Translation of Mathematical Monographs Series, AMS Publisher, 1967.

[3]F. Arman and J.K. Aggarwal, CAD-based vision: object recognition in cluttered range images using recognition strategies, *Computer Vision, Graphics and Image Processing*, Vol.58, No. 1, pp.33-48, 1993.

[4]C.Bajaj, F. Bernardini and G. Xu, Automatic reconstruction of surfaces and scalar fields from 3-D scans. *Proc. Computer Graphics (SIGGRAPH'95)*, pp.109-118, August,1995.

[5]P. Belhumeur, J. Hespanha and D. Kriegman, Eigenfaces vs. Fisherfaces: recognition using class specific linear projection, *Proc. European Conference on Computer Vision (ECCV'96)*, 1996.

[6]R. Bergevin, D. Laurendeau and D. Poussart, Estimating the 3D rigid transfromation between two range views of a complex object, *11th IAPR, Int'l Conf. Patt. Recog.*, pp. *478-482*, The Hague, The Netherlands, Aug. 30 - Sep. 3, 1992.

[7]R. Bergevin, D. Laurendeau and D. Poussart, Registering range views of multipart objects, *Computer Vision and Image Understanding*, Vol.61, No.1, pp. 1-16, 1995.

[8]P.J. Besl, The free-form surface matching problem, *Machine Vision for Three-dimensional Scenes*, H. Freeman, ed., pp. 25-71, Academic Press, 1990.

[9]P.J. Besl, Triangles as a primary representation, *Object Representation in Computer Vision*, M. Hebert, J. Ponce, T. Boult and A. Gross, eds., pp. 191-206, Berlin, Springer-Verlag, 1995.

[10]P.J. Besl and N.D. Mckay, A method for registration of 3-D shapes, *IEEE Transaction Pattern on Pattern Analysis and Machine Intelligence*, 14(2): pp. 239-256, 1992.

[11]J. Boissonnat, Geometric structures for three-dimensional shape representation, ACM Trans. on Graphics, Vol.3, No.4, pp. 266-286, October 1984.

[12]N. Burgess, M. N. Granieri and S. Patarnello. 3-D object classification: application of a constructive algorithm, *International Journal of Neural Systems* (Singapore), 2(4): pp. 275-282, 1991.

[13]T. Cass, Polynomial-time geometric matching for object recognition, *Int'l Jour. Computer Vision*, Vol.21(1/2), pp. 37-61, 1997.

[14]Y. Chen and G. Medioni, Object modeling by registration of multiple range images, *Image Vision Computing*, 10(3): 145-155, 1992.

[15]Y. Chen and G. medioni, Surface description of complex objects from range images, *Proc. IEEE Computer Vision and Pattern Recognition (CVPR'94)*, pp. 153-158, 1994.

[16]C.S. Chua and R. Jarvis, 3D free-form surface registration and object recognition, *Int'l J. of Computer Vision*, Vol. 17, pp. 77-99, 1996.

[17]C.S. Chua and R. Jarvis, Point signatures: a new representation for 3D object recognition, *Int'l J. of Computer Vision*, 25(1), pp.63-85, 1997.

[18]B. Curless and M. Levoy, A volumetric method for building complex models from range images, *Proc. Computer Graphics (SIGGRAPH'96),* August 1996.

[19]H. Dellingette, M. Hebert and K. Ikeuchi, A spherical representation for the recognition of curved objects, *Proc. Computer Vision and Pattern Recognition (CVPR'93),* pp. 103-112, 1993.

[20]H. Dellingette, M. Hebert and K. Ikeuchi, Shape representation and image segmentation using deformable surfaces, *Image and Vision Computing*, Vol.10, No.3, pp. 132-144, 1992.

[21]H. Dellingette, M. Hebert and K. Ikeuchi, Representation and recognition of free-form surfaces, Technical Report, CMU-CS-92-214, Computer Science Department, Carnegie Mellon University, November, 1992.

[22]J. Deovre, *Probability and statistics for engineering and sciences,* Brooks/Cole, Belmont, CA, 1987.

[23]T. D. DeRose, M. Lounsbery, J. Warren, Multi-resolution analysis for surfaces of arbitrary topological type, Technical Report, 93-10-05, Deptment of Computer Science and Engineering, University of Washington, October, 1993.

[24]D. Dion Jr., D. Laurendeau and R. Bergevin, Generalized cylinder extraction in range images, *Proc. Int'l Conf. on Recent Advance in 3-D Digital Imaging and Modeling,* Ottawa, pp. 141-147, May 1997.

[25]C. Dorai, A. Jain, COSMOS - a representation scheme for 3D free-form objects, *IEEE Transaction Pattern on Pattern Analysis and Machine Intelligence,* 19(10): pp. 1115-1130, 1997.

[26]M. Eck, T. DeRose, T. Duchamp, H. Hoppe, M. Lounsbery, and W. Stuetzle, Multi-resolution Analysis of Arbitrary Meshes, Technical Report, 95-01-02, The Computer Science Department, University of Washington, January 1995.

[27]J. Eells and L.H, Sampson, Harmonic mappings of Riemannian manifords, *Amer. J. Math*., 86:109-160, 1964.

[28]O. Faugeras, *Three-dimensional computer vision: a geometric viewpoint*, MIT Press, Cambridge, MA, 1993.

[29]O.D. Faugeras and M. Hebert, The representation, recognition and locating of 3-D objects, *Int'l J. of Robotics Research*, Vol. 5, No. 3, pp. 27-52, Fall 1986.

[30]P. Flynn and A. Jain, BONSAI: 3D object recognition using constrained search, Vol.13, No.10, pp. 1066-1075, October 1991.

[31]J. Foley, A. VanDam, S. Feiner and J. Hughes, *Computer Graphics: Principles and Practice*, Addison-Wesley, New York, 1990.

[32]K. Fukanaga, *Introduction to Statistical Pattern Recognition*, Academic Press, New York, 1972.

[33]M. Garland and P. Heckbert, Surface simplification using quadric error metrics, *Proc. Computer Graphics (SIGGRAPH'97)*, 1997.

[34]M. Hebert, K. Ikeuchi and H. Delingette, A spherical representation for recognition of free-form surfaces, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(7): 681-689, July 1995.

[35]M. Hebert, J. Ponce, T. Boult and A. Gross(Eds.), *Object representation in computer Vision*, Spinger-Verlag, Berlin, 1995.

[36]K. Higuchi, M. Hebert and K. Ikeuchi, Building 3D models from unregistered range images, *CVGIP-Image Understanding*, Vol. 57. No. 4. July 1995.

[37]A. Hilton, A. Stoddart, J. Illingworth and T. Windeatt, Reliable surface reconstruction from multiple range images, *Fourth European Conf. on Computer Vision (ECCV'96)*, pp. 14- 18, April 1996.

[38]H. Hoppe, T. DeRose, T. DuChamp, J. McDonald and W. Stuetzle, Surface reconstruction from unorganized points, *Proc. Computer Graphics (SIGGRAPH'92)*, pp. 71-78, July 1992.

[39]H. Hoppe, Progressive meshes, *Proc. Computer Graphics (SIGGRAPH'96)*, pp. 99-108, 1996.

[40]B.K.P. Horn, Extended Gaussian Image, *Proc. IEEE*, Vol. 72, pp. 1671-1686, 1984.

[41]D. P. Huttenlocher and S. Ullman, Recognizing solid objects by alignment with an image, *Int'l Jour. Computer Vision*, Vol.5, No.2, pp.195-212, 1990.

[42]K. Ikeuchi, T. Shakunaga, M. Wheeler and T. Yamazaki, Invariant histograms and deformable template matching for SAR target recognition, *Proc.Computer Vision and Pattern Recognition (CVPR'96)*, pp. 100-105, 1996.

[43]A. Johnson. and M. Hebert, Control of polygonal mesh resolution for 3D computer vision, Technical Report, CMU-RI-TR-96-20, Robotics Institute, Carnegie Mellon University, 1997.

[44]A. Johnson, Spin-Images: a representation for 3-D surface matching, Ph.D. Thesis, CMU-RI-TR-97-47, Robotics Institute, Carnegie Mellon University, 1997.

[45]A. Johnson and M. Hebert, Surface registration by matching oriented points, *Proc. Int'l Conf. on Recent Advance in 3-D Digital Imaging and Modeling*, pp. 121-128, Ottawa, May 1997.

[46]A. Johnson and S. Kang, Registration and integration of textured 3-D data, *Proc. Int'l Conf. on Recent Advance in 3-D Digital Imaging and Modeling (3DIM'97)*, Ottawa, pp. 234-241, 1997.

[47]T. Joshi, J. Ponce, B. Vijayakumar and D.J. Kriegman, HOT curves for modeling and recognition of smooth curved 3D objects, *Proc. IEEE conf. Computer Vision and pattern recognition*, Seattle, Wash., pp.876-880, June, 1994.

[48]Jurgen Jost, Harmonic maps between surfaces, Lecture *Notes in Mathematics*, Springer-Verlag, Berlin Heidelberg, New York, Tokyo, 1984.

[49]D. Keren, K. Cooper and J. Subrahmonia, Describing complicated objects by implicit polynomials, *IEEE Transaction Pattern on Pattern Analysis and Machine Intelligence,* 16(1): pp. 38-53, 1994.

[50]V. Koivunen and R. Bajcsy, Spline Representations in 3-D vision, in *Object Representation in Computer Vision*, M. Hebert, J. Ponce, T. Boult and A. Gross(Eds.) Springer-Verlag, pp. 177-190, December 1994.

[51]Y. Lamdan and H. Wolfson, Geometric hashing: a general and efficient model-based recognition scheme*, Proc.Second Int'l Conf. Computer Vision (ICCV'88),* pp. 238-249, 1988.

[52]A. Lee, W. Sweldens, P. Schroder, L. Cowsar, D. Dobkin, MAPS: multiresolution adaptive parameterization of surfaces, *Proc.Computer Graphics (SIGGRAPH'87)*, pp. , 1998.

[53]T. Lindeberg, Scale-space for discrete signals, *IEEE Transaction on Pattern Analysis and Machine Intelligence*, Vol. 12, No. 3, March, 1990.

[54]W. Lorensen and H. Cline, Marching cubes: a high resolution 3D surface construction algorithm, *Proc.Computer Graphics (SIGGRAPH'87),* pp. 163-169, 1987.

[55]D. Lowe, Organization of smooth image curves at multiple scales, *Int'l Jour. of Computer Vision,* 3, pp.119-130, 1989.

[56]Minolta Corporation, http://www.minolta3D.com.

[57]H. Murase and S. Nayar, Visual learning and recognition of 3-D objects from appearance, *Int'l Jour. Computer Vision*, Vol.14, pp. 5-24, 1995.

[58]J. Oliensis, Local reproducible smoothing without shrinkage, *IEEE Transaction on Pattern Analysis and Machine Intelligence,* 15(3): pp. 307-312, 1993.

[59]B. O'Neill, *Elementary differential geometry*, Academic Press, Inc., 1996.

[60]A. Pentland and S. Sclaroff, Closed-form solutions for physically based shape modeling and recognition, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol.13, No.7, pp.715-729, 1992.

[61]F. Pipitone and W. Adams, Tripod operators for recognizing objects in range images; rapid rejection of library objects, *Proc.IEEE Robotics and Automation (R&A 1992)*, pp. 1596-1601, 1992.

[62]E. Praun, H. Hoppe and A. Finkelstein, Robust mesh watermarking, *Proc.Computer Graphics (SIGGRAPH'92),* pp., 1999.

[63]F. Preparata and M. Shamos, *Computational geometry: an introduction,* Springer-Verlag, New York, 1985.

[64]W. Press, S. Teukolsky, W. Vetterling and B. Flannery, *Numerical recipes in C: the art of scientific computing,* 2nd Edition, Cambridge University Press, Cambridge, UK, 1992.

[65]N. Raja and A. Jain, Recognizing geons from superquadrics fitted to range data, *Image and Vision Computin*g, Vol.10, No.3, pp.179-190, 1992.

[66]W. Schroeder, J. Zarge and W. Lorensen, Decimation of triangular meshes, *Proc.Computer Graphics (SIGGRAPH'92),* pp.65-70, 1992.

[67]J. Schwartz and M. Sharir, Identification of partially obscured objects in two and three dimensions by matching noisy characteristic curves, *The Int'l. J. Robotics Research,* 6(2): pp. 29-44, 1987.

[68]S. Sclaroff and A. Pentland, Object recognition and categorization using modal matching, *Proc. 2nd CAD-Based Vision Workshop,* pp. 258-265. Champion, Pennsylvania, Feb. 8-11, 1994.

[69]S. Sclaroff and A. Pentland, Model matching for correspondence and recognition, *IEEE Trans. Pattern Analysis and Machine Intelligence,* Vol.17, No.6, pp.545-561, 1995.

[70]J.R. Shewchuk, Triangle: engineering a 2D quality mesh generator and Delaunay triangulator, *Proc. First Workshop on Applied Computational Geometry,* pp.124-133, Pittsburgh, PA, May 1996.

[71]H.Y.Shum, K.Ikeuchi and R.Reddy, Principal component analysis with missing data and its application to polyhedral object modeling, *IEEE Trans. Pattern Analysis and Machine Intelligence,* Vol.17, No.9, 1995.

[72]H. Shum, M. Hebert and K. Ikeuchi, On 3-D shape synthesis, Technical Report, CMU-CS-95-213, The School of Computer Science, Carnegie Mellon University, 1995.

[73]H. Shum, M. Hebert and K. Ikeuchi, On 3D shape similarity, *Proc. Computer Vision and Pattern Recognition (CVPR'96),* pp. 526-531. June 1996.

[74]D. Simon, Fast and accurate shape-based registration, Ph.D. Thesis, The Robotics Institute, Carnegie Mellon University, November 1996.

[75]M. Soucy and D. Laurendeau, Multi-resolution surface modeling from multiple range views, *Proc. IEEE Computer Vision and Pattern Recognition (CVPR'92),* pp.348-353, 1992.

[76]Michael Spivak, *A comprehensive introduction to differential geometry,* Boston, 1970.

[77]R. Sproull, Refinements to nearest neighbor searching in k-dimensional trees, *Algorithmica,* Vol.6, pp.579-589, 1991.

[78]F. Stein and G. Medioni, Structural indexing: efficient 3-D object recognition, *IEEE Transaction Pattern on Pattern Analysis and Machine Intelligence,* 14(2): pp. 125-145, 1992.

[79]R. Szeliski, d. Tonnensen and D. Terzopoulos, Modeling surfaces of arbitrary topology using dynamic particles, *Proc. Computer Vision and Vision Recognition (CVPR'93),* pp.92-97, 1993.

[80]G. Taubin, Estimation of planar curves, surfaces, and nonplanar space curves defined by implicit equations with applications to edge and range image segmentation, *IEEE Transaction Pattern on Pattern Analysis and Machine Intelligence*, 13(11): pp. 1115-1138, 1991.

[81]G. Taubin, A signal processing approach to fair surface design, *Proc. Computer Graphics (SIGGRAPH'95),* pp.351-358, 1995.

[82]D. Terzopoulos and M. Vasilescu, Sampling and reconstruction with adaptive meshes, *Proc. Computer Vision and Pattern Recognition (CVPR'91),* pp.70-75, 1991.

[83]J. Thirion, New feature points based on geometric invariants for 3D image registration, Int'l Jour. Computer Vision, Vol.18, No.2, pp.121-137, 1996.

[84]G. Turk, Re-tiling polygonal surfaces, *Proc. Computer Graphics (SIGGRAPH'92),* pp. 55-64, 1992.

[85]G. Turk and M. Levoy, Zippered polygonal meshes from range images, *Proc. Computer Graphics (SIGGRAPH'94),* pp.311-318, 1994.

[86]M. Turk and A. Pentland, Face Recognition using eigenfaces, *Proc. Computer Vision and Pattern Recognition (CVPR'91),* pp.586-591, 1991.

[87]S. Ullman and R. Basri, Recognition by linear combination of models, *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol.13, No.10, pp.992-1006, 1991.

[88]Hajime Urakawa, *Calculus of variations and harmonic maps*, American Mathematical Society, 1993.

[89]R. Veltkamp, 2D and 3D object reconstruction with the γ-neighborhood graph, Technical Report CS-R9116, CWI Center for Mathematics and Computer Science, 1991.

[90]M. Wheeler and K. Ikeuchi, Iterative smoothed residuals: a low-pass filter for smoothing with controlled shrinkage. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Vol. 18, No.3, pages 334-337, March 1996.

[91]M. Wheeler and K. Ikeuchi, Sensor modeling, probabilistic hypothesis generation, and robust localization for object recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Vol. 17, No.3, pp. 252-265, March, 1995.

[92]A. Witkin and P. Heckbert, Using particles to sample and control implicit surfaces, *Proc. Computer Graphics (SIGGRAPH'94),* pp.269-277, July 1994.

[93]M. Wheeler and K. Ikeuchi. Iterative estimation of rotation and translation using the quaternion. Technical Report CMU-CS-95-215, School of Computer Science, Carnegie Mellon University, December 1995.

[94]Y. Xin, *Geometry of Harmonic Maps,* Birkhauser, 1996.

[95]D. Zhang, M. Hebert, Multi-scale classification of 3D objects, *Proc. Computer Vision and Pattern Recognition (CVPR'97*), pp. 864-869, July, 1997.

[96]D. Zhang, M. Hebert, A. Johnson and Y. Liu, On Generating Multi-resolution Representations of Polygonal Meshes, *Proc. ICCV'98 Workshop on Model-based 3-D Image Analysis*, January 3, 1998, Bombay, India.

[97]D. Zhang, M. Hebert, Harmonic maps and their applications in surface matching, *Proc. Computer Vision and Pattern Recognition (CVPR'99*), pp. , July 1999.

[98]D. Zhang, M. Hebert, Harmonic Shape Images: A representation for 3D free-form surfaces, *Proc. Second International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition (EMMCVPR'99)*, York, England, July, 1999.

[99]D. Zhang, M. Hebert, Experimental analysis of Harmonic Shape Images, *Proc. Second International Conference on 3-D Digital Imaging and Modeling (3DIM'99)*, Ottawa, Canada, 1999.

[100]Z. Zhang, Iterative point matching for registration of free-form curves and surfaces, *IEEE Transaction Pattern on Pattern Analysis and Machine Intelligence*, 13:2, pp. 119-152, 1994.

Appendix A

Laser Range Scanner - VIVID 700

Most of the range images used in this thesis were taken using the non-contact 3D laser scanner, VIVID 700, manufactured by Minolta Corporation. With its high speed image processing unit, VIVID 700 accomplishes each scan in 0.6 second and generates a 200x200 point range image and a 400x400 point color image at the same time.

VIVID 700 operates on a light stripe triangulation range-finder principle. The object's surface is scanned from top to bottom with a projected laser light stripe. The reflected rays are collected and focused onto a CCD. The position of an illuminated surface point relative to the viewpoint is obtained by triangulation. The resolution in x and y is 200X200 range points per scan. By sharing a common optical axis, a color image is obtained with a resolution of 400X400 points without any parallax.

Portability is an important feature of VIVID 700. With the ATA PC-card memory, no on-site computer is required. VIVID 700 can also be remotely controlled from a Silicon Graphics (SGI) workstation (Indy, Indigo2, O2), or a PC-AT-compatible machine (running Windows NT) by using the utility software provided by Minolta. The remote control setup was used for taking the range images in this thesis.

The main functionalities of the software coming with VIVID 700 include: support of different output formats such as DXF, Wavefront, SOFTIMAGE, OpenInventor and ASCII; control of scanning, zooming, manual focusing, laser power setting, camera data reading and focus lock by remote control; support of outputs data in the form of a polygonal mesh, gouraud-shading, color image, or texture-mapping; and data processing such as smoothing, uniform subsampling and character-line mesh fitting. Among all the functionalities, the one that was used most often is generating the output data in the form of polygonal meshes and stored them in the OpenInventor format.

Table A.1 lists the specification of VIVID 700.

Table A.1: Specification of VIVID 700

Type Laser light-stripe	Triangulation rangefinder
Zoom lens	9mm - 46mm, 5 times zoom / 8 steps
Distance to the object	0.6m - 2.5m
Scanned area (x,y)	70mm X 70mm to 1100mm X 1100mm
Resolution (x,y,z)	200 X 200 X 256 points
Scanning time	0.6 seconds
Data transfer time to the computer	Less than 2.0 seconds

Table A.1: Specification of VIVID 700

File size	1.1MB for both range data and color texture map
Interface	SCSI II
Laser power	Class-2
Laser scanning method	Galvanometer-mirror
Ambient light condition	Under 500 lux (normal office lighting conditions)
Power	AC 100 - 240 V
Weight	9kg
Dimensions	210mm(w) X 326(d) X 367(h)
Output file formats	Wavefront (.obj), SOFTIMAGE, DXF, Open inventor, ASCII