Symbolic Signatures for Deformable Shapes

Salvador Ruiz-Correa, Linda G. Shapiro, *Fellow*, *IEEE*, Marina Meila,
Gabriel Berson, Michael L. Cunningham, and Raymond W. Sze

Abstract—Recognizing classes of objects from their shape is an unsolved problem in machine vision that entails the ability of a computer system to represent and generalize complex geometrical information on the basis of a finite amount of prior data. A practical approach to this problem is particularly difficult to implement, not only because the shape variability of relevant object classes is generally large, but also because standard sensing devices used to capture the real world only provide a partial view of a scene, so there is partial information pertaining to the objects of interest. In this work, we develop an algorithmic framework for recognizing classes of *deformable shapes* from range data. The basic idea of our *component-based approach* is to generalize existing surface representations that have proven effective in recognizing specific 3D objects to the problem of object classes using our newly introduced *symbolic-signature representation* that is robust to deformations, as opposed to a numeric representation that is often tied to a specific shape. Based on this approach, we present a system that is capable of recognizing and classifying a variety of object shape classes from range data. We demonstrate our system in a series of large-scale experiments that were motivated by specific applications in scene analysis and medical diagnosis.

Index Terms—Three-dimensional object recognition and classification, deformable shapes, range data, numeric and symbolic signatures, Mercer kernel, scene analysis, craniosynostosis, craniofacial malformations.

1 Introduction

Object recognition from shape has always been an important topic in computer vision research, but only with the advent of inexpensive range data acquisition devices, such as laser scanners and stereo cameras, has it become possible to develop recognition systems for applications in a variety of fields that include, among others, scene analysis for robotics and autonomous navigation, industrial machine vision, object retrieval in distributed database systems, and, more recently, shape-based analysis for medical diagnosis. However, most of the approaches developed to date have concentrated on designing algorithms for recognizing a small set of specific objects and very little attention has been paid to the more general problem of identifying objects that belong to a particular *shape class*.

Recognizing classes of 3D object shapes is a hard problem, not only because the amount of shape variability of the classes that occur in practice is large, but also because the nature of the input data is rather complex. Real range scenes may contain multiple shapes and the class members have to be identified among varying amounts of clutter. Range scenes also contain sensor noise and occlusion, so there is only partial information relative to the objects of interest.

Most 3D object recognition research in computer vision has heavily used the *alignment-verification* methodology for recognizing and locating specific objects [27]. This approach finds sets of point correspondences between a 3D model and a 2D or 3D scene using attributes of these points, such as *numeric signatures*. A numeric signature at a given point on a surface is a descriptor that encodes the geometric properties measured in a neighborhood of the point. For instance, the *spin images* of Johnson and Hebert [19], are created by constructing a pose-invariant 2D coordinate system at an oriented point (3D point with normal vector) on the surface, and accumulating the coordinates (α, β) of contributing points in a square matrix of dimension $N_n = N^2$. This is illustrated in Fig. 1. Contributing points are those that are within a specific distance of P and for which the surface normal forms an angle less than a specified size with the surface normal $\mathbf{N}$ of P. This angle is called the *support angle*, ϕ, and is used to limit the effect of self occlusion and clutter. As shown in Fig. 1, the coordinate α is the distance from P to the projection of Q onto the tangent plane T_P at point P; β is the distance from Q to this plane. The *width* of the spin image is the dimension N. The remarkable property of spin-image representation is that it combines the descriptive nature of global object characteristics with the robustness to partial views and clutter of local shape descriptions. Through adjustment of their generation parameters such as the support angle and width, spin images can be transformed from global to local representations. In fact, the spin image generation process can be visualized as a sheet spinning around the oriented point basis, accumulating space as it sweeps through space

- S. Ruiz-Correa and R.W. Sze are with the Department of Radiology, University of Washington and Children's Hospital and Regional Medical Center, 4800 Sand Point Way NW R-5438, Seattle, WA 98105. E-mail: sruiz@u.washington.edu, raymond.sze@seattlechildrens.org.
- L.G. Shapiro is with the Department of Computer Science and Engineering and the Department of Electrical Engineering, University of Washington, Box 352350, Seattle, WA 98195-2350. E-mail: shapiro@cs.washington.edu.
- M. Meila is with the Department of Statistics, University of Washington, Box 354322, Seattle, WA 98195-4322. E-mail: mmp@stat.washington.edu.
- M.L. Cunningham is with the Children's Craniofacial Center and the Pediatrics Division of Genetics and Developmental Medicine, University of Washington School of Medicine, 1959 NE Pacific Street, HSB RR-537A, Seattle, WA 98195-6320. E-mail: mcunning@u.washington.edu.
- G. Berson is with Children's Hospital and Regional Medical Center, 4800 Sand Point Way NE, Mailstop M2-9, Seattle, WA 98105. E-mail: gberson@u.washington.edu.

Manuscript received 10 Aug. 2004; revised 19 May 2005; accepted 19 May 2005; published online 11 Nov. 2005.
Recommended for acceptance by C. Kambhamettu.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-0418-0804.

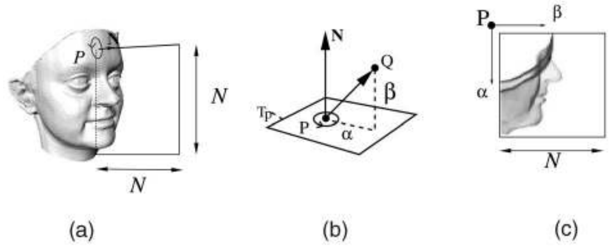


Fig. 1. Spin image construction. The spin image for point P is constructed by accumulating in a 2D histogram the coordinates α and β of a set of contributing points on a surface mesh. The image width is $N = 100$ pixels and the support angle is $\phi = 60^o$. Key: oriented point (P), tangent plane at P (T_P), and normal vector ($\mathbf{N}$).

(Fig. 1a). Using this analogy, it is possible to define the sweep volume as the portion of space swept out by the spinning sheet [19]. The spin image for the oriented point P in Fig. 1a encodes local and some global shape information of a face model, as can be appreciated in Fig. 1c. In this work, we use spin images as our numeric signatures.

The correspondences are used to solve a set of linear equations that determine the parameters of a potential rigid transformation that aligns the model to the scene. The model is projected by applying the transformation, and a verification technique determines if this is a good match by analyzing how well the model overlays the scene. This type of verification is dependent on a specific model for which there is an exact instance in the scene.

The alignment-verification technique has been very successful in industrial machine vision, but, in general, it fails to extend well to the problem of identifying classes of similar shapes. There are three main reasons for this failure: 1) Numeric shape representations are not robust to deformations, 2) there are no exact correspondences between the model and the scene, and 3) instances of a shape class do not necessarily align.

In this work, our focus is the development and testing of a *component-based methodology* that overcomes the limitations described above and is capable of identifying classes of deformable shapes that contain a significant amount of intraclass variability. Our approach consists of five key elements. We use *numeric signatures* to encode the surface geometry of an object shape and a *constellation of components*, formed by groups of equivalent numeric signatures, to quantify the geometry of a shape class. The spatial configuration of the components is encoded into a set of *symbolic signatures*, whose similarity across deformations is quantified using a novel *kernel function*. Finally, an architecture that consists of a series of *classification stages* learns components and their spatial arrangement from a set of exemplar shapes or a *training set*.

All the shapes considered in this work are represented as oriented surface meshes of fixed *resolution*. A surface mesh is a piecewise linear surface consisting of vertices, edges, and faces. Surface meshes are often used to approximate the boundary of an object and can be considered a dense version of a landmark-based shape descriptor whose resolution is defined as the median length of its edges. We also assume that, for a given shape class, the vertices of the meshes in the *training set* are in full correspondence.

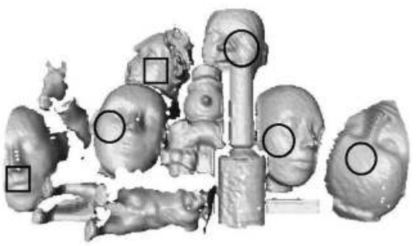


Fig. 2. Recognizing shape class members in a range scene. True positive recognition is indicated by a circle. A false negative decision is indicated by a square.

Finding full correspondences is a difficult problem that we approached using a morphable models algorithm developed in [2], [32] that is summarized in Section 3.

1.1 Problem Statement

We wish to use our component-based approach to develop a practical system that is capable of recognizing and classifying deformable shapes from range data. In a recognition task, a large part of the scene is clutter and we don't know which of the object classes are in the scene, while, in a classification task, all the objects in the scene belong to a known class. These tasks can be formally stated as follows:

Problem 1 (Recognition). We are given a training set of L surface meshes that form a random sample of a shape class $\mathcal{C}$. Using the given meshes, we wish to construct an algorithm that determines whether or not members of the shape class are present in a cluttered range scene containing occlusions (Fig. 2).

Problem 2 (Classification). We are given a random sample of surface meshes $\mathcal{S}$ of two shape classes $\mathcal{C}^{+1}$ and $\mathcal{C}^{-1}$, labeled by either $+1$ or -1, respectively. Using the given meshes and labels, we wish to construct an algorithm that can predict the label of a new surface mesh (Fig. 3). We also consider the problem of classifying shapes in a range scene containing more than one class member.

1.2 Contributions

In this work, we present a computational framework for recognizing classes of deformable shapes from range data. The contributions of our component-based approach include

1. a novel symbolic-signature representation of deformable shape classes that is robust to intraclass variability and missing information,
2. a region-growing algorithm for learning shape class components from numeric signatures,

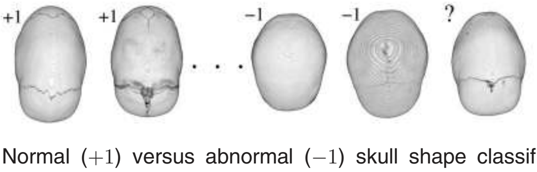


Fig. 3. Normal ($+1$) versus abnormal (-1) skull shape classification problem. The label of the last skull is unknown.

3. a novel kernel function for quantifying symbolic signature similarities,
4. a novel architecture of classifiers for abstracting the geometry of a shape class, and
5. a validation of our methodology in a set of large scale recognition and classification experiments.

The paper is organized as follows: Related work on 3D object shape recognition and classification is summarized in Section 2. Section 3 introduces our component-based methodology for recognizing and classifying object shapes. This section contains the development of the symbolic signature representation, a proposed kernel function for symbolic descriptors, and a novel classification architecture. The results of 6,170 large scale recognition/classification experiments on a variety of shape classes aimed at validating our approach are presented in Section 4. The paper concludes with a discussion of the lessons learned from the presented experiments and future research directions enabled by this work. The results of preliminary experiments related to the methodology presented in this work can be found in [29], [28].

2 Related Work

There are several methods reported in the literature that address the problem of recognizing or classifying 3D object shape instances. Nevertheless, to the best of our knowledge, none of them has addressed the recognition/classification problem using range data, where, as mentioned before, there is only partial information about the object shapes to be recognized/classified in the scene. Most existing methods attempt to recognize/classify a single presegmented instance, as opposed to our methodology, which can deal with multiple and incomplete instances. This section summarizes some recent work on shape class recognition and classification. The interested reader is referred to [9], [14], [21], [23] for references on these topics.

Recent investigations on 3D object class recognition have focused on developing shape representations and similarity measures for comparing shapes represented as polygonal meshes. For instance, Osada et al. developed in [25] a shape representation to recognize similar objects for applications in 3D object retrieval. The so-called shape distribution encodes the shape information of a complete 3D object as a probability distribution sampled from a shape function measuring geometric properties. The primary motivation of this approach is to reduce the shape matching problem to the comparison of probability distributions using an L_p norm. The underlying assumptions are that objects belonging to the same class will have similar shape distributions and that there is enough 3D shape information for the probability distributions to converge. Funkhouser et al. [12] extended the work on shape distribution by developing a representation based on a spherical harmonics expansion of the points of a polygonal surface mesh rasterized into a voxel grid. The grid is aligned to the center of mass of the object. Query objects represented as harmonics descriptors are matched to the database using a nearest neighbor classifier. In [21], Kazhdan et al. introduced a novel descriptor that represents a measure of symmetry for an arbitrary 3D model for all planes through the model's center of mass. This descriptor is insensitive to noise, is stable under point sampling, and improves the performance of the descriptors described above. The main advantage of these shape representations is that they can operate on degenerate meshes. They lack robustness to scene clutter and occlusion due to their global character.

Research on shape classification has concentrated on developing shape representations that are suitable for generative probabilistic modeling and statistical analysis. Common approaches include active shape models (ASM) [6], [8], statistical deformation models (SDM) [26], and morphometrics [3], [22]. However, in only a few instances are full 3D models considered and the classification scheme is restricted to a linear or quadratic classifier that operates on data obtained by performing Principal Component Analysis (PCA) on the original data set [1], [24]. A notable exception to this trend is the work of Golland [13], which develops a discriminative framework for classifying and quantifying shape that is based on the so-called *discriminative direction* for kernel classifiers. The discriminative direction corresponds to the differences between classes implicitly represented by a classification function. It is worth mentioning that none of the approaches mentioned in this section is suitable for applications with range data since they assume complete presegmented models that are aligned to the center of mass of the objects.

3 Materials and Methods

As pointed out in Section 1, although object recognition based on the alignment-verification technique has been very successful in industrial machine vision applications, its use in the context of shape classes is limited for three main reasons: 1) Numeric signature representations are not robust to deformations. For instance, spin images do not vary smoothly as a function of the orientation of normal vectors across an object [18]. As a consequence, instances of the same shape class with different normal vector distributions may have very different spin images on regions that are semantically equivalent. 2) There are not exact correspondences between any of the models in the training set and the shapes in the scene and 3) instances of a shape class do not necessarily align.

The *component-based methodology* developed in this section addresses these problems. We use numeric signatures to encode the geometry of specific objects and groups of equivalent classes of numeric signatures, or *shape class components*, to encode the geometry of a shape class. That is, our approach learns components from examples in order to account for shape variations across class instances so that explicit modeling for handling deformations is not required.

The spatial configuration of a group of components (or *constellation*) is encoded into a set of symbolic descriptors (or *symbolic signatures*) whose variability across deformation space is learned by a series of classification stages. These stages utilize a novel *Mercer kernel* for quantifying symbolic similarities. More specifically, the conceptual role of our proposed kernel is to embed purely symbolic patterns into *Hilbert space* in which a similarity measure between patterns is defined as the inner product of the space. The nice

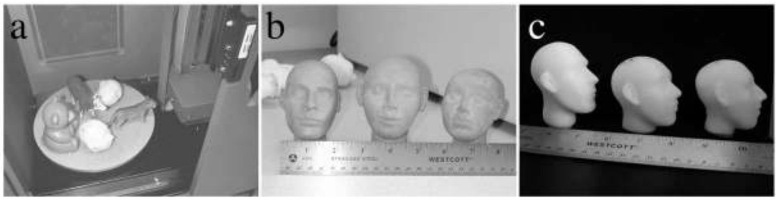


Fig. 4. (a) Laser scanner used for acquiring data. (b) Artistic models made by a sculptor. (c) Real models of virtual examples that were produced with a 3D printer.

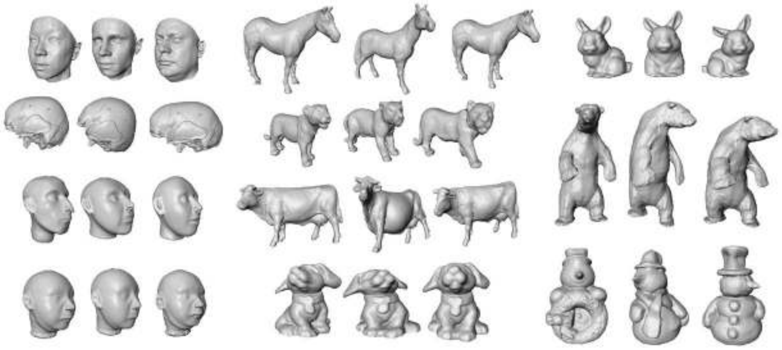


Fig. 5. Some of the shape classes in our database.

property of the Mercer kernels is that explicit knowledge of the embedding and inner product functions is not necessary. The kernel itself provides both implicitly [34].

This section is divided in four parts that present:

1. a description of our database,
2. a preview of our component-based methodology,
3. training and testing algorithms, and
4. implementation.

3.1 Object Database

We used a Roldand-LPX250 laser scanner to acquire our data. The scanner has a maximum resolution of 0.008 inches for plane scanning mode. The scanner and some of the objects used in the experiments are shown in Fig. 4. Raw data consisted of clouds of points that were further processed to obtain smooth and uniformly sampled triangular meshes of 0.9-1.0 (mm) resolution. Real scenes used in the experiments described in Section 4 were made of handmade clay toys, animals, scale models of human heads sculpted by an artist, real skull models, and other common house and office objects chosen at random. Fig. 5 shows several examples of each of the shape classes of our database. Some recognition experiments also included real human faces from a database provided by the University of South Florida [2], which consists of 100 dense surface meshes (70,000 vertices and 160,000 polygons each).

3.2 Component-Based Methodology: A Preview

For the sake of simplicity, we consider the classification task defined in Section 1 for which we are given L surface meshes representing instances and noninstances of a shape class. Each surface mesh is labeled by $y \in \{\pm 1\}$. The problem is to use the given meshes and the labels to predict the label of a new surface mesh. We summarize the key points of our approach by presenting an architecture that consists of two stages (Fig. 6a). The first stage identifies shape class components by using two classification blocks: a bank of component detectors and a multiway classifier that assigns component labels (Fig. 6b). The input is a surface mesh and the output is a *labeled surface mesh*, defined as a mesh for which each vertex has an associated symbolic label referencing the component in which it lies. The labeled mesh of Fig. 6b has four component labels numbered from 5 to 8.

The second stage verifies the spatial configuration of the components detected in the previous stage. It also consists of two classification blocks, a set of symbolic signature detectors, and a multiway classifier that assigns symbolic labels (Fig. 6c). The input is a labeled surface mesh and the output is a class label. In Fig. 6c, the bank of detectors identifies a pattern associated with components 5, 6, and 8. The multiway classifier determines that this symbolic pattern corresponds to the class of abnormal heads (-1). Symbolic patterns associated with components 1-4 (not shown in the figure) correspond to normal heads.

3.3 Training Algorithms

3.3.1 Learning Shape Class Components

The pseudocode of our training algorithm for learning shape class components is shown in Fig. 7. The process starts with the acquisition of surface mesh models ($MeshModels$) from L real objects ($L \sim 15$). These meshes are processed by the functions listed in Fig. 7 as follows.

—computePointCorrespondences: This function implements a morphable surface models algorithm developed by Shelton that finds dense correspondences between surfaces of arbitrary topology by minimizing an energy function [32]. The algorithm does not require user intervention. The input to this function is the set of surface meshes and the output is a set of correspondences ($Correspondences$) for the vertices of all surface meshes. Correspondences are computed so that the vertices have the same semantic across mesh models. For instance, consider a collection of human

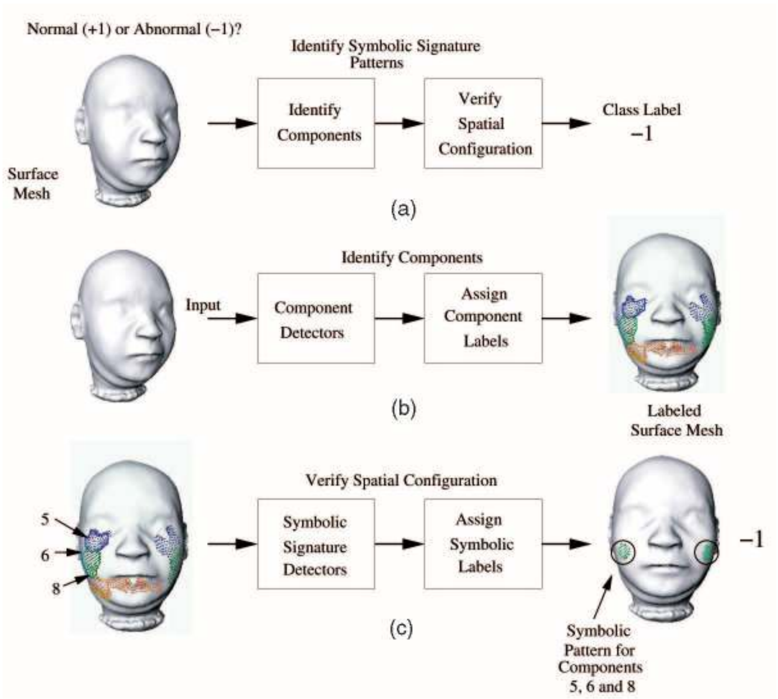

Normal (+1) or Abnormal (−1)?
Identify Symbolic Signature Patterns
Identify Components
Verify Spatial Configuration
Class Label
−1
Surface Mesh
(a)
Identify Components
Input
Component Detectors
Assign Component Labels
(b)
Labeled Surface Mesh
Verify Spatial Configuration
5
6
8
Symbolic Signature Detectors
Assign Symbolic Labels
−1
(c)
Symbolic Pattern for Components 5, 6 and 8

Fig. 6. (a) Our proposed architecture for a binary classification problem consists of two main stages. (b) The first stage identify components. (c) The second verifies their spatial relationships.

```
Input: A set of $L$ of $RealObjectShapes$ and $UserInput$ that is required to select seed points.
Algorithm:
$MeshModels = \mathrm{scan}(RealObjectShapes)$;
$Correspondences = \mathrm{computePointCorrespondences}(MeshModels)$;
$SeedPoints = \mathrm{selectSeedPoints}(MeshModels, Correspondences, UserInput)$;
$TrainingSet = \mathrm{generateVirtualExamples}(MeshModels, Correspondences)$;
$[TS1, TS2] = \mathrm{splitTrainingSetInHalf}(TrainingSet)$;
$\mathrm{bank_n} = \mathrm{computeBank_nOfComponentDetectors}(TS1, SeedPoints)$;
$LabeledTS1 = \mathrm{detectComponents}(TS1, \mathrm{bank_n})$;
$LabeledTS2 = \mathrm{detectComponents}(TS2, \mathrm{bank_n})$;
$\mathrm{multiway_n} = \mathrm{computeComponentsClassifier}(LabeledTS1, LabeledTS2)$;
Output: a bank $\mathrm{bank_n}$ of component detectors, $\mathrm{multiway_n}$ classifier and two sets of labeled surface meshes $LabeledTS1$ and $LabeledTS2$.
```

Fig. 7. Training algorithm for learning a shape class components.

skull mesh models. If a vertex is on the tip of the nasal bone in a particular mesh, then the corresponding vertices must lay on the tip of the nasal bone for all the meshes in the collection.

—selectSeedPoints: This function implements a graphical interface that allows the user to manually select points on a set of surface meshes whose vertices are in full correspondence. The output is a list of seed points ($SeedPoints$) whose meaning will be explained below.

—generateVirtualExamples: This function allows the user to generate virtual examples by applying deformations on a set of surface meshes [2]. The input is a set of mesh model correspondences. The output is a set of deformed surface meshes, called $TrainingSet$, whose vertices are also in full correspondence. Global deformations are generated by using morphing operators whose parameters are randomly chosen from given ranges. The operators included eight parametric deformations (taper, twist, stretch, push, bend, spherify, scale, and mirror) and each deformed model included at least five different deformations applied in a random sequence. Global deformations are also generated using convex combinations. Local deformations are produced by constructing elastic models of the objects and applying small random force fields using a multiresolution approach [5]. In the case of the human head class, shape changes are controlled to keep the following anthropometric landmark distances within normal averages: head width and length, forehead height, nasal length, lower facial height, facial height, and facial width [11]. Real plastic models of virtual examples were created with a 3D printer (Fig. 4c).

—splitTrainingSetInHalf: The input to this function is a set of surface meshes $TrainingSet$. The outputs are two sets of surface meshes called $TS1$ and $TS2$. The sets have the same size, their vertices are in full correspondence, and their elements were randomly chosen without replacement

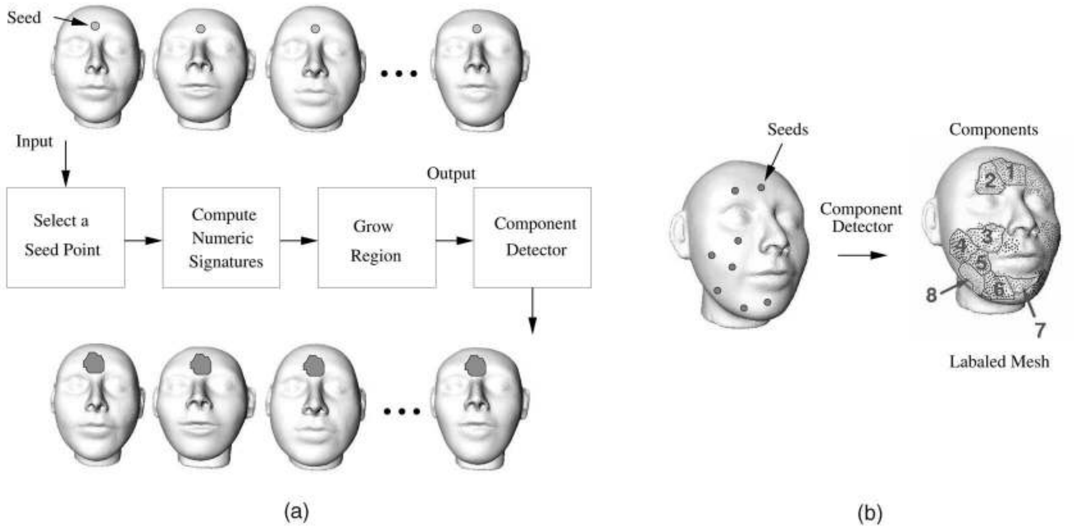


Fig. 8. (a) A component detector is computed using the numeric signatures of seed points selected on a training set (top row). The signatures are fed to a region growing algorithm that expands the component around the seeds. (b) Extracted components for eight seed points are shownon the right side.

from $TrainingSet$. These sets are required to separately train and test the classification functions described below.

—$\mathrm{computeBank_nOfComponentDetectors}$: The input to this function is a set of surface meshes $TS1$ and a set of seed points used to grow the components. The output is a bank of component detectors $\{\mathrm{bank}_\mathrm{n}^1, \cdots, \mathrm{bank}_\mathrm{n}^{\mathrm{N_C}}\}$, where N_C is the number of components (number of seed points). Each component detector $\mathrm{bank}_\mathrm{n}^\mathrm{i}$ is a classifier that can recognize the component that is grown about $SeedPoint_i$. A region growing algorithm both grows the region and learns its classifier as follows: Given a set of surface meshes $TS1$, whose vertices are in full correspondence, and a seed point s, the numeric signatures are computed at the corresponding seed points of every training instance in $TS1$, as shown in Fig. 8. This set of signatures is the training set T_s for the selected seed point s. Then, using the training set T_s, a component detector is trained to learn a component about s.

Region Growing. The component detector (a ν-SVM) actually grows a region about s using the shape information from the numeric signatures in the training sample. The growing phase is as follows: The performance of the component detector for point s can be quantified by calculating a bound on the expected probability of error E on the training set as $E = \#SV_s/\#P$, where $\#SV_s$ is the number of support vectors in the component detector for s, and $\#P$ is the number of mesh points that belong to the region. Using the classifier for point s, an iterative component growing operation expands the component about s. Initially, the component consists only of point s. An iteration of the procedure consists of the following steps:

1. Select a point that is an immediate neighbor of one of the points in the component and is not yet in the component.
2. Retrain the classifier with the current component plus the new point.
3. Compute the error E' for this classifier.
4. If the new error E' is lower than the previous error E, add the new point to the component and set $E = E'$.
5. This continues until no more neighbors can be added to the component.

Fig. 8a summarizes the region growing algorithm. The training instances in the top row are marked with a seed point on the leftmost sample and the corresponding seed points on the other instances. After region growing around the seed, the growth components are shown in the bottom row. Note that the user only selects seed points on one of the training instances. The corresponding seeds on other instances are automatically known since the points of all meshes are in full correspondence.

A component detector is implemented as a one class ν-SVM [30]. This classifier partitions the space of numeric signatures into regions of typical and outlier elements. A component is thus defined as the set of all numeric signatures that are classified as typical by the component detector. Fig. 8b shows a component extraction example. Eight seed points were selected by hand and the components were grown one at a time on the training set. The resulting component detectors were applied to an independent test set and the output is shown in Fig. 9. Each surface mesh in the figure has a total of eight surface regions that belong to each of the learned components. Regions belonging to the same component appear in both sides of the faces. This is due to the fact that the numeric signatures being used are invariant to mirror symmetry and our approach preserves this invariance approximately. The approach we have followed to construct our component detectors is related to the one used by Heisele et al. [16].

Selection of Seed Points. Seed points were selected one at a time following an iterative process. First, the (normalized) mean curvature of several shape models was estimated using the methods described in [33]. The first seed point was located on a region that, on average, had low curvature and its corresponding component detector estimated. The next point was manually located close to

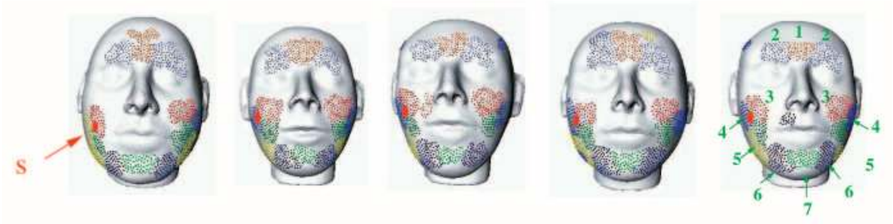
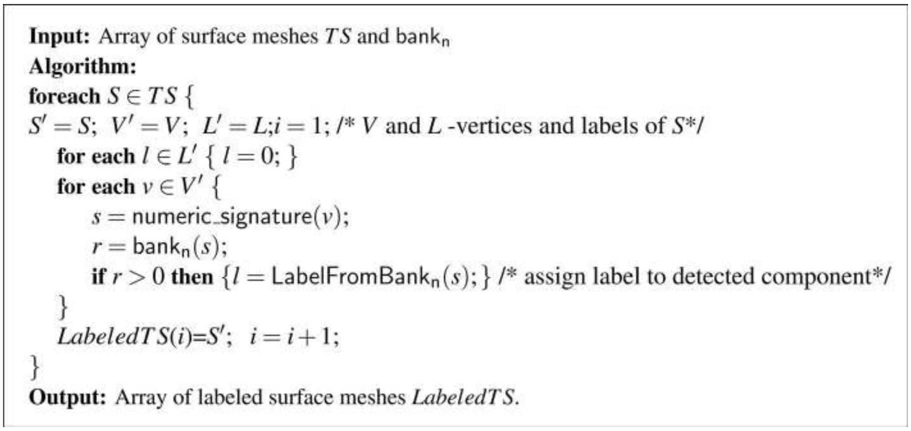


Fig. 9. Output of a set of eight component detectors applied to a testing set of surface meshes representing human head shapes.



```
Input: Array of surface meshes $TS$ and bank$_n$
Algorithm:
foreach $S \in TS$ {
$S' = S$;  $V' = V$;  $L' = L$;$i = 1$; /* $V$ and $L$ -vertices and labels of $S$*/
    for each $l \in L'$ { $l = 0$; }
    for each $v \in V'$ {
        $s = $ numeric_signature($v$);
        $r = $ bank$_n$($s$);
        if $r > 0$ then {$l = $ LabelFromBank$_n$($s$); } /* assign label to detected component*/
    }
    $LabeledTS(i)$=$S'$;  $i = i + 1$;
}
Output: Array of labeled surface meshes $LabeledTS$.
```

Fig. 10. Pseudocode for function detectComponents.

the first seed point. It was rejected as the second seed point whenever its corresponding component detector produced a region that overlapped that of the first seed point. In that case, a different location for the second seed point was manually selected. This operation was repeated until no overlap occurred. A similar procedure was followed to select other seed points. The total number of components per class was iteratively selected on the basis of the recognition/classification rates. Initially, a small number of components per shape class (two to three) was selected and a pilot study performed. The number of components was increased and the study repeated whenever recognition/classification rates were low. This procedure continued until the system achieved true positive and true negative rates close to 80 percent.

The approach just described was used to select and locate the seed points for the components of all the shape classes considered in the experimental validation of Section 4. No more than three iterations were needed to specify seed points and compute two to eight component detectors. On average, the processing time with a training set consisting of 200 meshes was roughly 1.75 hours.

—detectComponents: The inputs to this function are an array of surface meshes and a $\mathrm{bank_n}$ of component detectors. The output is an array of labeled surface meshes. The pseudocode of this function is shown in Fig. 10.

—computeComponentsClassifier: The inputs to this function are two sets of labeled surface meshes, $LabeledTS1$ and $LabeledTS2$. The output is a $\mathrm{multiway_n}$ classifier that is trained and tested with the numeric signatures associated with the components of the meshes in $LabeledTS1$ and $LabeledTS2$, respectively. The role of the $\mathrm{multiway_n}$ classifier is to verify the labels assigned by the $\mathrm{bank_n}$ of detectors. In practice, we found that the use of the $\mathrm{bank_n}$ requires a subsequent processing with a $\mathrm{multiway_n}$ classifier in order to achieve good recognition and classification performance at testing time. A detailed description of the joint use of the $\mathrm{bank_n}$ of detectors and $\mathrm{multiway_n}$ classifier will be given in Section 3.4.

3.3.2 Learning Symbolic Signatures

Symbolic signatures were motivated by the observation that the algorithm described in the previous section generated regular patterns on the occurrence and geometric configuration of the learned components. An example of this behavior, which was consistently observed in the experiments described in Section 4, is depicted in Fig. 9.

Symbolic surface signatures are shape descriptors that encode the geometric configuration of a constellation of components. They are somewhat related to numeric surface signatures in that they also start with a *critical point* P on the surface mesh and consider a set of contributing points Q, which are still defined in terms of the distance from P and support angle (refer to Section 1 for a definition of this parameter). The main difference is that they are derived from a labeled surface mesh (shown in Fig. 11a); each vertex of the mesh has an associated symbolic label referencing a *component* in which it lies. For symbolic surface signature construction, the vector $\overline{PQ}$ in Fig. 11b is projected to the tangent plane at P, where a set of orthogonal axes γ and δ have been defined. The direction of the $\gamma - \delta$ axes is arbitrarily defined since no curvature information was used to specify preferred directions. This ambiguity is resolved by the methods described below. The discretized version of the γ and δ coordinates of $\overline{PQ}$ are used to index a square matrix of dimension N^2 and the indexed position of the matrix is set to the component label of Q. Note that it is possible for multiple points Q that have different labels to

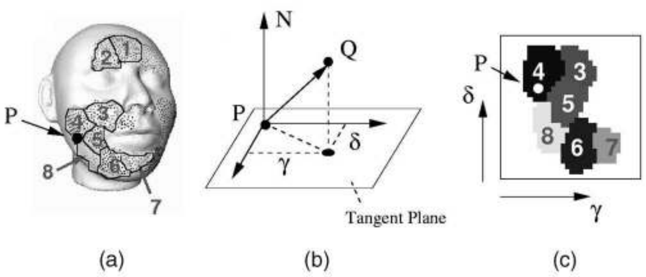


Fig. 11. (a) Labeled surface mesh. (b) Coordinate system used to construct the symbolic signature for point P. (c) Symbolic signature for P.

project into the same bin. In this case, the label that appeared most frequently is assigned to the bin. Empty bins, that is, bins that do not store any component label are set to "0." The resultant matrix is the symbolic surface signature at point P. This signature captures the relationships among the labeled regions on the mesh. It is shown as a labeled image in Fig. 11c. The computational complexity for constructing a symbolic signature with $N_s = N^2$ elements is $O(N_s)$.

Symbolic signatures for point s for the deformed range images in Fig. 9 are shown in Fig. 12. Although these shapes belong to a class that contains a significant amount of intraclass variability, the symbolic signatures remain consistent across deformation space. The actual plastic models corresponding to the heads 2 and 4 (from left to right) in Fig. 9 are shown in Fig. 4c in order to illustrate shape differences.

The pseudocode of our training algorithm for learning the symbolic signatures is shown in Fig. 13. The process starts with the manual selection of the critical points. These are used as the base P for computing the symbolic signatures during the training phase (Fig. 11b). This computation is performed on the points of the labeled meshes in $LabeledTS1$ and $LabeledTS2$.

—selectCriticalPoints: This function implements a graphical interface for selecting critical points with user interaction.

—$\mathrm{computeBank_sOfSymbolicSignatureDetectors}$: The input to this function are two sets of labeled surface meshes $LabeledTS1$ and $LabeledTS2$ and a set of $CriticalPoints$. The output is a bank of symbolic signature detectors $\{\mathrm{bank}_s^1, \cdots, \mathrm{bank}_s^{N_s}\}$, where N_S is the number of critical points. The $i\mathrm{th\text{-}detector}$ is trained with symbolic signatures associated with the $i\mathrm{th\text{-}critical}$ point of the surface meshes in the set $LabeledTS1$. The testing is performed with the signatures associated with the surface meshes of the set $LabeledTS2$.

—computeSymbolicSignatureClassifier: The inputs to this function are two sets of labeled surface meshes, $LabeledTS1$ and $LabeledTS2$, and a set of $CriticalPoints$. The output is $\mathrm{multiway_s}$, a classifier that is trained and tested with the numeric signatures associated with the components of the meshes in $LabeledTS1$ and $LabeledTS2$, respectively. The role of the $\mathrm{multiway_s}$ classifier is to verify the labels assigned by the $\mathrm{bank_s}$ of detectors. In practice, we found that the use of the $\mathrm{bank_s}$ requires a subsequent processing with a $\mathrm{multiway_s}$ classifier in order to achieve good recognition and classification performance at testing time. A detailed description of the joint use of the $\mathrm{bank_s}$ of detectors and $\mathrm{multiway_s}$ classifier will be given in Section 3.4.

Selection of Critical Points. Critical points are used as the base points to construct symbolic signatures. A symbolic signature is generated by projecting the labels of the vertices of a surface mesh onto a tangent plane defined at the base point (Fig. 11). We selected the location of the critical points in order to project the maximum number of different labels to the tangent plane. That is, critical points were selected such that the symbolic signatures encoded the maximum possible number of distinct class labels. We found that symbolic signatures constructed in this way provided information that improved the performance of the classification stages. The entropy of symbolic signatures with a diversity of class labels is higher than the entropy of symbolic signatures encoding few labels.

3.3.3 A Mercel Kernel for Symbolic Signatures

Our kernel function for measuring similarities between symbolic patterns was motivated by the development of

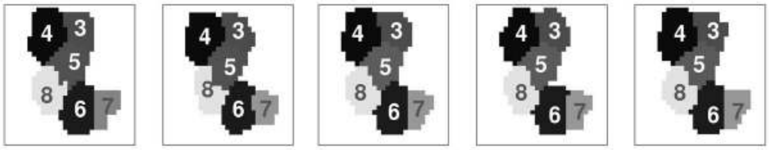
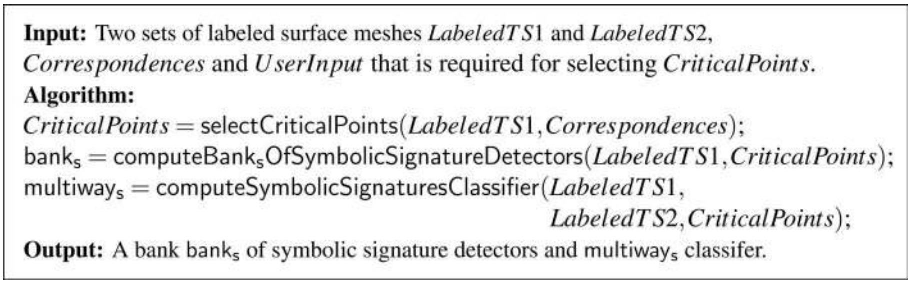


Fig. 12. Symbolic signatures for critical point P in Fig. 11.



```
Input: Two sets of labeled surface meshes $LabeledTS1$ and $LabeledTS2$,
$Correspondences$ and $UserInput$ that is required for selecting $CriticalPoints$.
Algorithm:
$CriticalPoints$ = selectCriticalPoints($LabeledTS1, Correspondences$);
$\mathrm{bank_s}$ = computeBank$_s$OfSymbolicSignatureDetectors($LabeledTS1, CriticalPoints$);
$\mathrm{multiway_s}$ = computeSymbolicSignaturesClassifier($LabeledTS1$,
                                                   $LabeledTS2, CriticalPoints$);
Output: A bank $\mathrm{bank_s}$ of symbolic signature detectors and $\mathrm{multiway_s}$ classifer.
```

Fig. 13. Training algorithm for learning symbolic signatures.

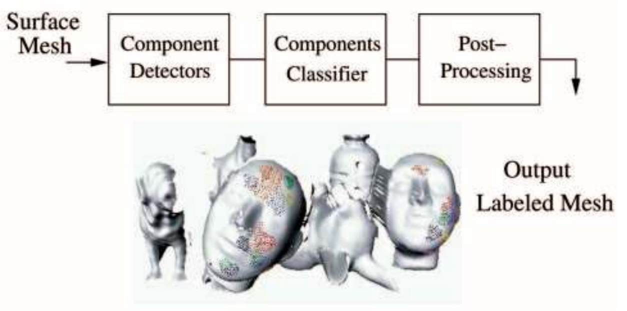


Fig. 14. Classification stage I structure in a recongition task. In this example, $N_C = 8$.

kernels for string matching [31]. The function is rather simple, but it is powerful enough to produce high recognition and classification rates (Section 4). It is constructed by embedding symbolic signatures into a Hilbert space by means of a Mercer kernel as follows. Let A and B be two symbolic signatures of dimension N^2 storing component labels. The symmetric mapping defined as $< A, B >= \frac{1}{N^2}\sum_{ij}\delta_{a_{ij}b_{ij}}$, where δ is the Kroneker delta and whose range is the interval $[0, 1]$, can be interpreted as the cosine of angle θ_{AB} between two unit vectors on the unit sphere lying within a single quadrant. The quantity $cos\theta$ is the correlation coefficient between A and B, and the angle $\theta_{AB} = cos^{-1}(< A, B >)$ is the geodesic distance. Our kernel function is defined as $k_\sigma(A, B) = \exp(-\theta^2_{AB}/\sigma^2)$, a Gaussian kernel for symbolic signatures. The parameter σ is defined as the width of the function and determines the zone of influence of one symbolic pattern on to another.

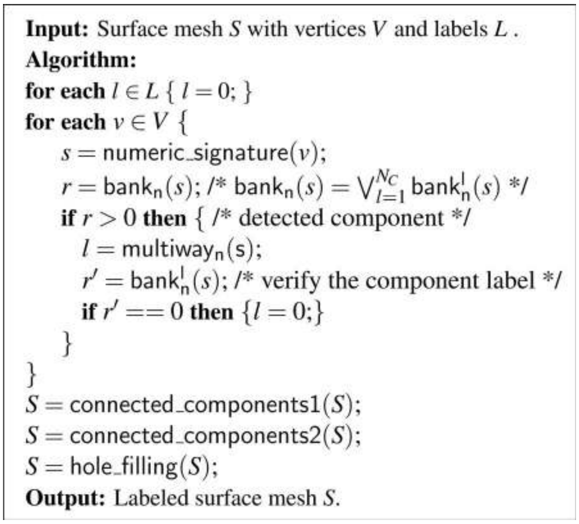
3.4 Testing Algorithms

Our testing algorithms are implemented as an architecture consisting of a cascade of two modules. The first module extracts shape class components; the second verifies their corresponding geometric configuration in order to identify symbolic patterns associated with particular shapes. A nice feature of these modules is that they have the same internal structure. Their only difference is that they operate on different input spaces. It is worth recalling that we are addressing two kinds of tasks: a recognition task and a binary classification task (see Problems 1 and 2 in Section 1).

3.4.1 Classification Stage I (for Numeric Signatures)

The first module operates on a surface mesh (Fig. 14). It consists of a bank $\{\mathrm{bank}^1_\mathrm{n}, \cdots, \mathrm{bank}^{\mathrm{N_C}}_\mathrm{n}\}$ of detectors and a multiway classifier $\mathrm{multiway_n}$. The bank detects components of the input mesh based on the numeric signatures of the vertices. Recall that the binary output of the lth component detector is defined as 1 if the signature lies inside its corresponding typical region and 0 otherwise. The multiway classifier assigns component labels to the numeric signatures that are accepted by the bank of detectors. The operation of this stage can be summarized by the pseudocode of Fig. 15, where $\bigvee$ represents the "or" operator. The functions used by the algorithm are defined as follows:

—connected_components1: This function applies a connected components algorithm to a labeled surface mesh S, where each connected component consists of groups of labeled points connected by a path on the mesh. Components with less than f_a (~ 16) points are filtered out, that is, their corresponding vertices are set to label "0."



```
Input: Surface mesh $S$ with vertices $V$ and labels $L$ .
Algorithm:
for each $l \in L$ { $l = 0$; }
for each $v \in V$ {
    $s = \mathsf{numeric\_signature}(v)$;
    $r = \mathsf{bank_n}(s)$; /* $\mathsf{bank_n}(s) = \bigvee_{l=1}^{N_C} \mathsf{bank^l_n}(s)$ */
    if $r > 0$ then { /* detected component */
      $l = \mathsf{multiway_n(s)}$;
      $r' = \mathsf{bank^l_n}(s)$; /* verify the component label */
      if $r' == 0$ then {$l = 0$;}
    }
}
$S = \mathsf{connected\_components1}(S)$;
$S = \mathsf{connected\_components2}(S)$;
$S = \mathsf{hole\_filling}(S)$;
Output: Labeled surface mesh $S$.
```

Fig. 15. Classification stage I algorithm.

—connected_components2: This function applies a connected components algorithm to a surface mesh S, where each connected component consists of groups of points with the same label. Components with less than f_b (~ 5) points are filtered out.

—fill_holes: This function relabels a surface mesh S applying the following algorithm: For each vertex of the labeled mesh, compute a histogram of labels corresponding to its immediate neighbors, including the label of the current vertex. Relabel each vertex of the mesh using the label that has the highest frequency according to the histogram of the current vertex.

The worst-case complexity of the component detection procedure in a range scene with V mesh vertices is $O(N_n N_C V)$. Our multiway classifier uses a *pairwise* classification scheme, therefore, the worst-case complexity for assigning labels is $O(N_n N_C^2 V)$. A *one-versus-all* scheme reduces the complexity to $O(N_n N_C V)$, but we have not determined if this configuration decreases the recognition/classification performance significantly.

3.4.2 Classification Stage II (for Symbolic Signatures)

The second module operates on a labeled surface mesh (Fig. 16). It consists of a bank $\{\mathrm{bank}^1_\mathrm{s}, \cdots, \mathrm{bank}^{\mathrm{Ns}}_\mathrm{s}\}$ of symbolic signature detectors and a multiway classifier $\mathrm{multiway_s}$. The bank identifies symbolic patterns on the input mesh based on the symbolic signatures of the vertices and the multiway classifier assigns labels to the patterns found. The input to this stage is the labeled surface mesh S from the previous stage. The output is a class label. The operation of this stage can be summarized by the pseudocode of Fig. 17.

The output class label is computed from the labels assigned to S' by classification stage II. This label identifies symbolic patterns associated with constellations of components. For instance, in the binary classification task illustrated in Fig. 18b, the symbolic patterns of components 1, 2, and 3 are related to the class of normal human head shapes. The

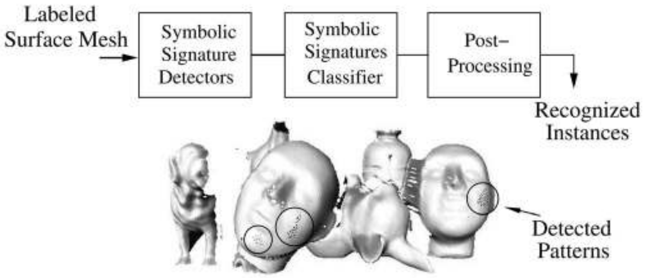


Fig. 16. Classification stage II structure used in a recognition task. In this example, $N_S = 2$, that is, there are two symbolic patterns that identify human head shape class members.

symbolic patterns of components 5, 6, and 7 correspond to abnormal human heads. Therefore, the labels of the vertices of S' are set to $+1$ (normal head) where evidence of the symbolic pattern of components 1, 2 and 3 was found and the output class label is set to $+1$; the labels of S' are set to -1 (abnormal) where evidence of the symbolic pattern of components 5, 6, and 7 was detected and the output class label is set to -1 (Fig. 18c). The case of a recognition task is similar except that all symbolic patterns correspond to the same shape class. For instance, in Fig. 16, there are ($N_S = 2$) patterns that identify human head class members.

3.5 Implementation

We used one class ν-support vector machines (ν-SVMs) or novelty detectors as our component detectors and ν-SVM binary classifiers for constructing multiway classifiers. ν-SVMs were introduced in [30] as an approach to control the number of margin errors and support vectors. More specifically, the parameter ν of a ν-SVM has a rather intuitive interpretation as an upper bound on the fraction of margin errors and a lower bound on the fraction of support vectors. Our SVMs were trained using Gaussian kernels for numeric signature classifiers and our proposed Gaussian kernel for symbolic signature classifiers. We trained all of our classifiers by using standard cross-validation techniques for model (ν and σ) selection [7], [31]. The interested reader is referred to [4] and [31] for an introductory tutorial to SVMs and training techniques, respectively.

Since symbolic surface signatures are defined up to a rotation, we use the virtual support vectors method for training all the classifiers involved with symbolic signatures [31]. The method consists of training a component detector on the signatures to calculate the support vectors. Once the support vectors are obtained, new virtual support vectors are extracted from the labeled surface mesh in order to include the desired rotational invariance, that is, a number r of rotated versions of each support vector is generated by rotating the $\delta - \gamma$ coordinate system used to construct each symbolic signature (see Fig. 11). Finally, the novelty detector used by the algorithm is trained with the enlarged data set consisting of the original training data and the set of virtual support vectors.

4 Experimental Results

This section describes the results of 6,170 large scale experiments aimed at validating our approach for recognizing/classifying 3D shape classes from range data. The experiments focus on 1) measuring the conditions under which our approach generalizes on unseen 3D data and 2) quantifying its sensitivity (true positive) and specificity (true negative) rates as a function of scene clutter and occlusion.

We applied the methods described in [28] to measure the scene occlusion and clutter of an object model. These quantities are respectively defined as:

$$\text{occlusion} = 100 - \frac{\text{model surface patch area on the scene}}{\text{total model surface area}}, \tag{1}$$

```
Input: Labeled surface mesh $S$ from classification stage I output with vertices $V$ and labels $L$.
Algorithm:
$S' = S;\ V' = V;\ L' = L$
for each $l \in L'$ $\{ l = 0; \}$
for each $v \in V'$ $\{$
    $s = \text{symbolic\_signature}(v);$
    $r = \text{bank}_s(s);$ /* $\text{bank}_s(s) = \bigvee_{l=1}^{N_S} \text{bank}_s^l(s)$ */
    if $r > 0$ then $\{$ /* detected symbolic pattern */
        $l = \text{multiway}_s(s);$
        $r' = \text{bank}_s^l(s);$ /* verify pattern label */
        if $r' == 0$ then $\{l = 0;\}$
    $\}$
$\}$
$S' = \text{connected\_components1}(S');$
$S' = \text{connected\_components2}(S');$
$S' = \text{hole\_filling}(S');$
$class\_label = \text{class\_label}(S');$
Output: $class\_label$.
```

Fig. 17. Classification stage II algorithm.

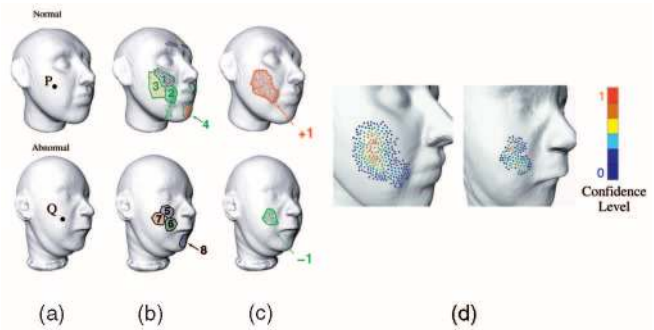


Fig. 18. Classification example of normal versus abnormal human heads. (a) Normal (top) and abnormal (bottom) human head. (b) Components 1, 2, 3, and 4 are associated with normal human head shapes (top). Component 5, 6, 7, and 8 are associated with abnormal heads (bottom). (c) The symbolic patterns associated with normal heads was found on the red region (top), while the symbolic patterns associated with abnormal heads were located on the green region (bottom). (d) Red points indicate the highest confidence level regarding the presence of the symbolic patterns found.

$$\text{clutter} = 100 - \frac{\text{clutter points in model surface patch}}{\text{total points in model surface}}, \quad (2)$$

where the clutter points in the model surface patch are those whose numeric signatures have been corrupted during their generation process due to the presence of cluttering objects in the range scene [18] that are within sweep volume of the signatures. The generation process was described in Section 1 and illustrated in Fig. 1.

The recognition state of our algorithm was defined as:

1. true positive (TP) whenever a class member present in the scene is assigned its class label;
2. false positive (FP) whenever a class label is assigned to an object present in the scene that is not a class member;
3. true negative (TN) whenever class members are not present in the scene and no labels are assigned; and
4. false negative (FN) whenever a class label is not assigned to a scene object that is a member of the shape class.

The classification state was defined in a similar manner. We define sensitivity and specificity as the TP and TN rates, respectively. Note that $\mathrm{TP} = 100 - \mathrm{FN}$ and $\mathrm{TN} = 100 - \mathrm{FP}$.

We used spin images as our numeric signatures (see Section 1). In all the experiments, the support angle was set to 60°. This angle allows the spin-image representation to encode global aspects of shape while maintaining robustness against clutter [18]. The support angle for the symbolic signatures was set to 30°.

4.1 Recognition

In the recognition experiments, we developed three tasks that considered range scenes containing: 1) a single class member with no clutter and low levels of occlusion, 2) multiple shape class members and multiple clutter objects, and 3) human head models and multiple clutter objects (detection of human faces). The parameters settings used in Tasks 1-3 are summarized in Table 1. Note that the training and testing set sizes, as well as the number of component and symbolic signature detectors shown in the table, are specified per shape class.

4.1.1 Task 1 (Single Object)

This task considered real range scenes containing a single object with no clutter and low levels of self-occlusion (< 30 percent). The objects were randomly selected from a set containing known shape class members (snowmen, rabbits, floppy-eared dogs, cats, cows, bears, horses, human heads, and faces) and other unrelated shapes (a total of 50 objects). These experiments were run in order to estimate true positive (TP) and true negative (TN) rates. We note that, in the case of the human face class, all the scenes were synthetically generated as a convex combination of the original UFS data that consists of 100 faces.

The results of 2,360 experiments suggest that our approach is capable of recognizing a variety of shape classes with high sensitivity and specificity. In all cases but one, the TP and TN rates are above 80 percent (see Table 2). These results are encouraging considering the large inter and intraclass variability of the shape classes tested. The average TP and TN rates across classes (excluding human faces) are 90.27 percent and 88.15 percent, respectively. Typical qualitative results are shown in Fig. 19. We note that the recognition performance for the human faces class was poor. This is likely due to the fact that our training set is small (100 meshes) and the shape class variability is rather large. The set of faces includes people of different age, gender, race,

TABLE 1
Parameter Setting per Class for Tasks 1-3

Parameter settings	Task 1	Task 2	Task 3
Training set size (per class)*	400	400	400
Testing set size (per class)	200	200	250
No. experiments	2360	1200	1010
No. shape classes	9	3	1
No. cluttering objects out of 50†	0	1-5	0-4
No. component detectors per class (N_C)	3	3	8
No. symbolic signature detectors per class (N_S)	1	1	2
Numeric (symbolic) signature size N_n (N_s)	40^2, (20^2)	40^2, (20^2)	70^2 (12^2)
No. virtual rotations for symbolic signatures	10	10	12

** For the class of human faces, we used 100 meshes for training and 100 meshes for testing. † For each experiment, clutter objects were randomly selected from a set of 50 real 3D objects.*

TABLE 2
Percent True Positive (TP) and True Negative (TN) Rates for Recognition Tasks 1-3

Task	SC	TP	TN	O	C	Task	SC	TP	TN	O	C
1	snowmen	89.14	85.17	< 30	0	2	snowmen	91.00	69.00	< 45	< 45
	rabbits	90.72	89.60	< 30	0		rabbit	90.20	72.40	< 45	< 45
	dogs	91.26	90.00	< 30	0		dog	89.60	65.40	< 45	< 45
	cats	92.04	87.33	< 30	0		snowman	87.50	72.00	< 75	< 70
	cows	88.15	81.70	< 30	0		rabbit	84.3	76	< 75	< 70
	bears	87.03	86.40	< 30	0		dog	89.60	77.9	< 75	< 70
	horses	93.11	91.55	< 30	0	3	heads	96.00	92.00	< 35	< 40
	heads	90.07	93.44	< 30	0		heads	87.00	84.7	< 40	< 40
	faces	76.00	43.39	< 30	0		heads	86.30	81.01	< 40	<45

Key: shape class (SC), % occlusion (O), and % clutter (C).

and facial gesture. We hypothesize that a larger training set could improve the performance dramatically.

4.1.2 Tasks 2 (Multiple Objects)

This task considered 510 real range scenes containing multiple shape class members (snowmen, rabbits, and dogs) and multiple clutter objects. The scenes were created by placing, without any systematic method, two to six objects in the 3D scene by hand. Each scene always contained at least one member of the learned classes. Typical range scenes of Task 2 are shown in the left column of Fig. 20. The ellipses indicate the region of the scene where the class members were found. Squares indicate false positive or false negative recognition.

The true positive rates for this task are high considering that, in practice, the objects in a scene are not as packed as they are in scenes in our database (Table 2). The average TP rates are 90.26 percent and 87.13 percent for levels of occlusion and clutter lower than 45 percent and 75 percent, respectively. However, the true negative rates are low (> 65 percent). On average, for levels of clutter and occlusion below 45 percent (75 percent), the algorithm assigns erroneous labels 31 percent (25 percent) of the time. These results indicate that if a small number of components is used to identify each shape class (3 in this case), it is possible to achieve high sensitivity levels at the cost of low specificity since there may be a number of scene objects that share similar components but belong to distinctive shape classes. The results of Task 3 revealed that a successful solution to this issue in a practical application is simply to increase the number of components and symbolic signature detectors to represent each shape class.

4.1.3 Task 3 (Detection of Human Faces in Complex Scenes)

This task considered real range scenes containing human head models and multiple clutter objects. One hundred sixty range scenes were created by following the procedure described in Task 2. One hundred thirty contained at least one shape class member and other object shapes randomly selected from a set of 50 models. Thirty scenes contained only nonclass member objects. All scenes are real; no synthetic scenes were generated in this task. The heads in this scenes were physically made of head data provided by a sculptor. The heads shape class was represented by using eight components and two symbolic signature detectors. Typical range scenes for this task are illustrated in the right column of Fig. 20.

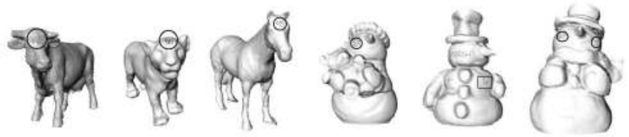


Fig. 19. Qualitative recognition results for Task 1. Ellipses indicate true positive recognition. Squares indicate false positive recognition.

It is clear from Fig. 20 that the models are closely packed, a condition that creates cluttered scenes with occlusions (see (1) and (2) for a definition of clutter and occlusion). We also note that the shape class components in Task 3 were grown from seed points located on the right cheek (Fig. 9). However, the algorithm is able to correctly detect the partially occluded faces (Fig. 20), due to the fact that the algorithm is approximately invariant to mirror transformations. Currently, the components of a shape class must be visible in the scene so that the classification stage II can identify the corresponding symbolic signature. Shape classes with multiple signatures can be recognized if at least one signature is visible.

The results of 1,010 experiments showed that our proposed algorithm can achieve high sensitivity and specificity. For instance, in Table 2, the TP and TN rates are above 92 percent for levels of clutter and occlusion levels below 35 percent and 40 percent, respectively. Both rates decrease as the levels of clutter and occlusion in the scene increase, but remain above 81 percent. The behavior of our algorithm is remarkable considering the complexity of the scenes. These results suggest that the sensitivity-specificity trade-off can be greatly improved by increasing the number of component and symbolic signature detectors that are used to represent a shape class. Finally, Fig. 21 shows the TP rate as a function of scene occlusion and clutter. The vertical bars bound the average values, shown as filled circles. Both charts indicate that the recognition performance decreases for levels of clutter and occlusion above 45 percent. The bounding bars were estimated using a bootstraping algorithm. The bootstrap is a resampling method for assigning nonparametric measures of accuracy [10].

4.2 Classification

We developed two binary classification tasks that considered scenes containing: 1) a single class member with no occlusion and clutter and 2) multiple class members with

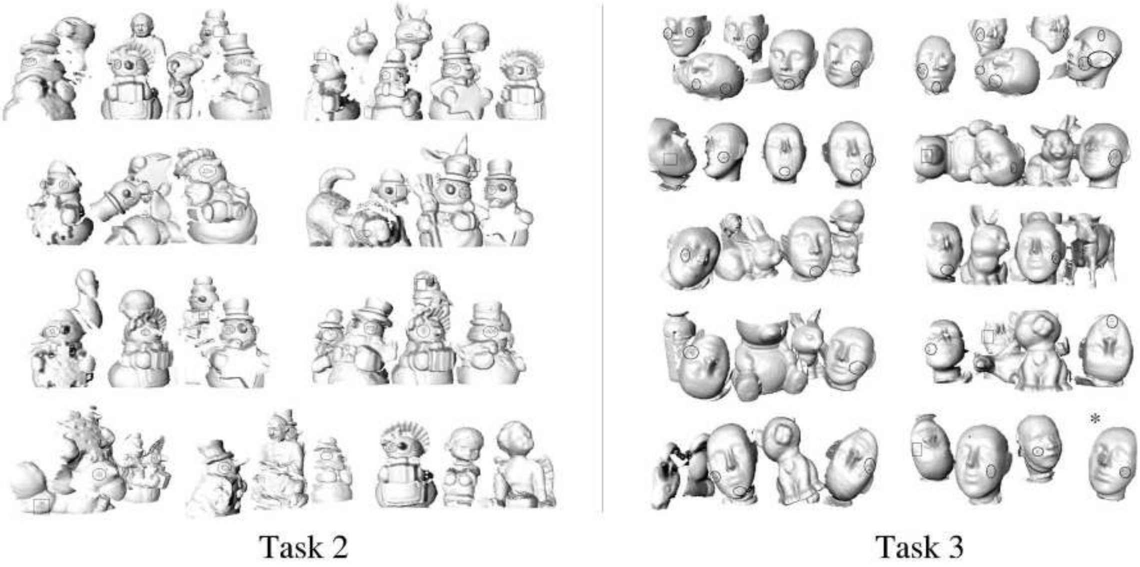


Fig. 20. Qualitative recognition results for Tasks 2 and 3. Ellipses indicate true positive recognition. Squares indicate false positive recognition.

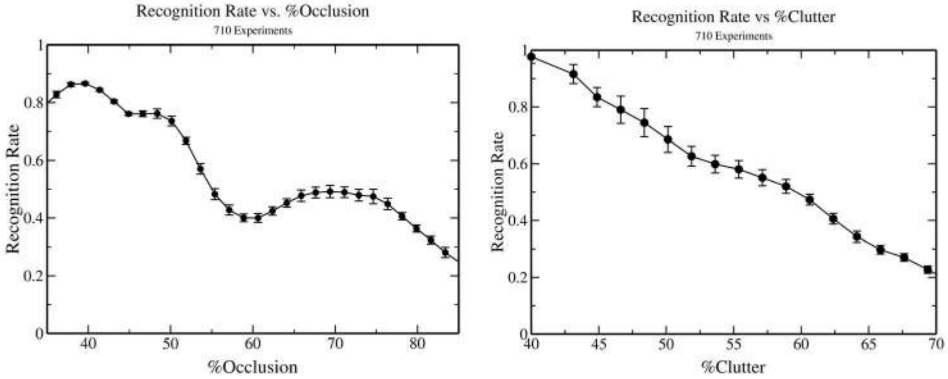


Fig. 21. Quantitative recognition results for Task 3.

occlusion and low levels of clutter. The parameter settings for these tasks are shown in Table 3.

Task 4.2.1 (Single Object). For this task, we used six subclasses of human head shapes and two classes of human skull shapes. We performed classification of normal versus abnormal heads (skulls), a task that occurs in medical settings. The head abnormalities considered are related to two genetic syndromes that produce severe craniofacial deformities. The head samples were obtained from models with craniofacial features based upon either Greig cephalopolysyndactyly (Abnormal 1) or trisomy 9 mosaic (Abnormal 2) syndromes [20]. Our goal was to measure the sensitivity and specificity of our classifier in discriminating two well-defined classes, for which a very fine distinction exists (Fig. 22).

We attempted to discriminate between test samples that were: 1) either 100 percent normal or 100 percent abnormal and 2) either 100 percent abnormal or had a varying degree of abnormality (Abnormals 3-5 are shown in Fig. 22). The samples in 2) were convex combinations of normal and abnormal heads. The (normal, abnormal) weights used to compute the convex combinations are: Abnormal 3 (65 percent, 25 percent), Abnormal 4 (50 percent, 50 percent), Abnormal 5 (25 percent, 75 percent) and Abnormal 6 (15 percent, 85 percent). The degree of collusion between the resulting classes made the discrimination process more difficult. Our rationale was to drive a realistic task to its limits in order to study the discrimination capabilities of our approach. It is worth mentioning that all the models constructed as convex combinations were validated by an expert geneticist and an expert radiologist, both coauthors of this work.

Our data shows that the classification algorithm has high values of sensitivity and specificity for discriminating

TABLE 3
Parameter Settings per Class for Tasks 4 and 5

Parameter settings	Task 4	Task 5
Training set size (per class)*	400	400
Testing set size (per class)	200	200
No. experiments	1400	200
No. shape classes	8	2
No. component detectors per class (N_C)	3	4
No. symbolic signature detectors per class (N_S)	1	2
Numeric (symbolic) signature size N_n (N_s) *	50^2 (12^2)	50^2 (20^2)
No. virtual rotations for symbolic signatures	10	10

** For the class of human skulls, numeric signature size (N_C) was set to* 100^2.

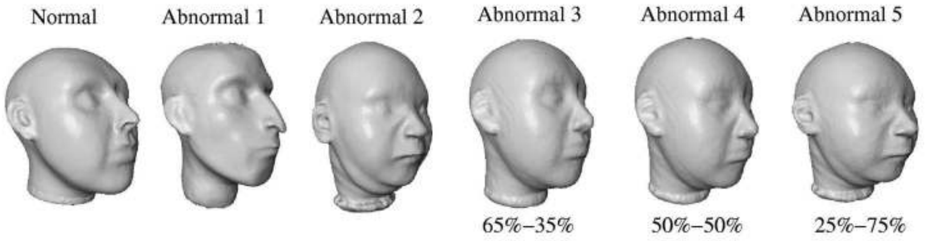


Fig. 22. Detecting syndromes from craniofacial deformations. From left to right: normal human head shape, abnormal head with the characteristics of a patient with Greig cephalopolysyndactyly syndrome, abnormal head model with the characteristics of a patient with the trisomy 9 mosaic syndrome. The last three models are human head models that are a convex combination of Normal and Abnormal 2.

TABLE 4
Percent True Positive (TP) and True Negative (TN) Rates for Recognition Tasks 4 and 5

Task	SC	TP	TN	O	C
4	N 1 vs. A 1	98	91	0	0
	N 2 vs. A 2	100	88.1	0	0
	N 2 vs. A 3	98	89.7	0	0
	N 2 vs. A 4	97	92	0	0
4	N 2 vs. A 5	92.00	93	0	0
	N 2 vs. A 6	48.20	51.7	0	0
	NS vs. AS	85.97	86.3	0	0
5	N 2 vs. A 1	88.03	82.7	<45	<15

Key: shape class (SC), % occlusion (O), % clutter (C), normal head (N), abnormal head (A), normal skull (NS), and abnormal skull (AS).

TABLE 5
Worst-Case Complexity for All Processing Stages

Processing Stages	Computational complexity	Typical Values	T (seconds per scene point)
bank_n	$O(N_n N_C V)$	$N_n = 70^2, N_C = 16$ and $V = 20,000$	0.029 ± 0.0098
multiway_n	$O(N_n N_C^2 V)$		
bank_s	$O(N_s N_S V)$	$N_s = 12^2, N_S = 4$ and $\varepsilon \sim .15$	0.028 ± 0.0069
multiway_s	$O(N_s N_S^2 V \varepsilon)$		

The parameter settings of Task 3 are reported as typical values in the table. Key: Numeric signature size (N_n), total number of components class components (N_C), average number of scene points (V), symbolic signature size (N_s), total number of symbolic signature detectors (N_S), average proportion of scene points that were not filtered out by classification stage I (ε), and average running time per range scene point (T).

normal from abnormal heads (Table 4) . It also discriminates very well between heads that are 100 percent abnormal and heads with varying degrees of abnormality, including meshes that have only 25 percent of normal features. The algorithm breaks down when the test sample has only 15 percent of normal features.

This task also includes classification experiments of normal and abnormal skull shapes. The abnormality considered is associated with *craniosynostosis*, a serious and common disease of children cause by premature fusion of the sutures of the skull. The resulting abnormal skull growth can lead to severe deformity, increased intracranial pressure, vision, hearing, and breathing problems. Fig. 3 shows a normal and a synostotic skull shapes. The sensitivity and specificity values are also high (above 85 percent); see Table 4.

4.2.2 Task 5 (Multiple objects). For this task, we considered real range scenes containing two or more members of the subclasses of 100 percent normal or 100 percent abnormal head shapes. The levels of occlusion and clutter were below 45 percent and 15 percent, respectively, and the number of components per class was set to 4. Moderate sensitivity and specificity rates were obtained in these experiments (88.03 percent and 82.7 percent, respectively).

Processing times and computational complexity. For the sake of space, we report the running time for the experiments of Task 3, which had the longest execution times. The numbers are summarized in Table 5. We note that the range scenes for these experiments contained 20,000 points on average. The recognition process took 12 minutes per scene on a Linux server running with a 2.8GHz processor. We were able to train and test our algorithm with surface meshes composed of $2 - 3 \times 10^4$ points. The processing time with meshes containing more than 8×10^4 points was impractical.

We end this section by emphasizing that all our experiments considered range scenes from real object shapes containing a significant degree of intra and interclass variation. For instance, our data set of human heads included several distinctive head models produced by a sculptor (Figs. 4b and 4c). The shape variation of the snowmen set is another pertinent example (Figs. 19 and 20). We acknowledge that our training sets were enlarged using a variety of deformation techniques. However, the deformations produced are large and very significant. Our point is that, to the best of our knowledge, our proposed algorithm is the first to be able to recognize/classify real range data with clutter and occlusion into learned deformable shapes. We suggest that this advance should be beneficial to a number of practical applications. Obviously, the shape variability of everyday objects is enormous and a comprehensive solution to the problem is yet to be discovered.

5 Conclusions

In this paper, we have presented a novel symbolic-signature approach to the difficult problem of recognizing and classifying raw data into learned deformable shapes. We have shown that our symbolic descriptors and classification architecture can be used to represent a variety of objects classes so that the expected sensitivity to large shape deformations and consequent intraclass variability issues are properly addressed. We validated our approach with a set of 6,170 large-scale experiments with real range data containing varying levels of scene clutter and occlusion and obtained encouraging results. We have also suggested that our approach is practical and can be effectively used in industrial and medical applications. However, there are a number of important issues that need to be investigated in the future, as listed below:

1. The region growing algorithm for computing class components uses seed points that are manually selected by the user. We have described an iterative algorithm that accomplishes this task and have shown that it is practical and works for the classes of objects considered in the experiments. During the training process of our experimental validation, we did not observed much changes in recognition/classification performance as a function of the location of the seed points as long as the average (normalized) curvature of the region on which the points were selected was low. However, we acknowledge that it is necessary to carefully study the sensitivity in recognition/classification performance as the number and location of the seeds points change. We hypothesize that components computed on regions of high curvature are less reliable and more difficult to learn than components computed on regions of low curvature. The numeric signatures used in this work require the estimation of surface normal vectors. It is well-known that the estimates of normal vectors on regions of high curvature are noisy and produce numeric signatures with higher variance than those estimated on regions of low curvature.
2. In our experiments, we observed that few components can provide high sensitivity at the expense of specificity and that a simple solution to this is to increase the number of components per class. However, this could become an issue in the case of a large number of classes as the complexity of the $\mathrm{multiway_n}$ and $\mathrm{multiway_s}$ classifiers is proportional to the square of the number of component detectors and symbolic signature detectors, respectively (Table 5). A possible solution to decrease complexity could be to use a one-versus-all classification scheme. However, the recognition/classification performance of this scheme has not been determined in the context of our algorithm.
3. The extension of our approach to a multiclass setting requires the addition of an output stage that looks up the class shapes labels assigned to the learned symbolic signature patterns. However, experiments are needed to determine the extensibility to a large number of classes due to the complexity reasons mentioned in point 2.
4. More extensive testing is required to provide a full characterization as a function of inter and intraclass variability. Although how to quantitatively assess such variability is an open question at present, it is clear that large data sets are required to learn a shape class with high variance, such as the USF face set.
5. We used three different image widths for both the numeric and symbolic signatures described in the experiments. The results are comparable in all cases, but more extensive testing is required for a complete characterization of the methodology as a function of the image width and support angle of the numeric and symbolic signatures.
6. We made extensive use of off-the-shelf ν-SVM classifiers, but are currently investigating the use of Maximum-Entropy [17] and Bayes point classifiers [15], which allow us to utilize prior knowledge that could potentially improve the performance of our algorithms for cases in which the amount of intraclass variability is very large. We are also investigating techniques to simultaneously train and test the bank of detectors and the multiway classifiers.

Acknowledgments

This work supported in part by The Laurel Foundation Center for Craniofacial Research, The Marsha Solan-Glazer Craniofacial Endowment, and a grant from NICDR R01-DE-13813. The authors gratefully acknowledge the suggestions made by the reviewers.

References

[1] R. Andersen, F.L. Bookstein, K. Conradsen, B.K. Ersboll, J.L. Marsh, and S. Kreiborg, "Surface-Bounded Growth Modeling Applied to Human Mandibles," *IEEE Trans. Medical Imaging,* vol. 19, no. 11, pp. 1053-1063, 2000.

[2] B. Blantz and T. Vetter, "A Morphable Model for the Synthesis of 3-F Faces," *Proc. 1999 ACM SIGGRAPH,* pp. 187-194, 1999.

[3] F.L. Bookstein, "Shape and the Information in Medical Images: A Decade of the Morphometric Synthesis," *Computer Vision and Image Understanding,* vol. 66, no. 2, pp. 99-118, 1997.

[4] C.J.C. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition," *Data Mining and Knowledge Discovery,* vol. 2, no. 2, pp. 121-167, 1998.

[5] S. Capell, S. Green, B. Curless, T. Duchamp, and Z. Popovic, "A Multiresolution Framework for Dynamic Deformations," *Proc. 2002 ACM SIGGRAPH,* vol. 2, pp. 42-48, 2002.

[6] T.F. Cootes, D. Cooperand, C.J. Taylor, and J. Graham, "Active Shape Models," *Computer Vision and Image Understanding,* vol. 61, no. 1, pp 38-59, 1995.

[7] C. Cortes and V.N. Vapnik, "Support Vector Networks," *Machine Learning,* vol. 20, pp. 273-279, 1996.

[8] R.H. Davies, C.J. Twining, T.F. Cootes, J.C. Waterton, and C.J. Taylor, "A Minimum Description Length Approach to Statistical Shape Modeling," *IEEE Trans. Medical Imaging,* vol. 21, no. 5, pp. 525-537, 2002.

[9] I. Dryden and K. Mardi, *Statistical Shape Analysis.* New York: Wiley, 1998.

[10] B. Efron, *The Jackknife, the Bootstrap, and Other Resampling Plans.* Philadephia: SIAM, 1982.

[11] U.G. Froster-Iskenius and J.E. Allanson, *Handbook of Normal Physical Measurements.* Oxford Medical Publications, 1989.

[12] T. Funkhouser, P. Min, M. Kazhdan, J. Chen, A. Halderman, D. Dobkin, and D. Jacobs, "A Search Engine for 3D Models," *ACM Trans. Graphics,* vol. 22, no. 1, pp. 83-115, 2003.

[13] P. Golland, "Discriminative Direction for Kernel Classifiers," *Advances in Neural Information Processing Systems,* vol. 13, pp 745-752, 2001.

[14] P. Golland, "Statistical Shape Analysis of Anatomical Structures," doctoral dissertation, Massachusetts Inst. of Technology, 2001.

[15] R. Hebrich and T. Graepel, "Bayes Point Machines," *J. Machine Learning Research,* no. 1, pp. 245-279, 2001.

[16] B. Heisele, T. Serre, M. Pontil, T. Vetter, and T. Poggio, "Categorization by Learning and Combining Object Parts," *Advances in Neural Information Processing Systems,* vol. 2, pp. 1239-1245, 2001.

[17] T. Jaakkola, M. Meila, and T. Jebara, "Maximum Entropy Discrimination," *Advances in Neural Information Processing Systems,* no. 11, pp 640-646, 1999.

[18] A.E. Johnson and M. Hebert, "Control of Polygonal Mesh Resolution for 3D Computer Vision," *Graphics, Modeling, and Computer Vision,* 1998.

[19] A.E. Johnson and M. Hebert, "Using Spin Images for Efficient Object Recognition in Cluttered 3D Scenes," *IEEE Trans. Pattern Recognition and Machine Intelligence,* vol. 21, no. 5, pp. 433-449, May 1999.

[20] K.L. Jones, *Smith's Recognizable Patterns of Human Malformation.* W.B. Saunders Company, 1999.

[21] M. Kazhdan, B. Chazelle, D. Dobkin, T. Funkhouser, and S. Rusinkiewicz, "A Reflexive Symmetry Descriptor for 3D Models," *Algorithmica,* vol. 38, no. 2, pp. 201-225, 2003.

[22] S.R. Lale and J.T. Richtsmeier, *An Invariant Approach to Statistical Analysis of Shape.* Chapman and Hall/CRC, 2001.

[23] S. Loncaric, "A Survey of Shape Analysis Techniques," *Pattern Recognition,* vol. 31, no. 8, pp. 938-1001, 1998.

[24] J. Martin, A. Pentland, S. Sclaroff, and R. Kikinis, "Characterization of Neurophatological Shape Deformations," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 20, no. 2, Feb. 1998.

[25] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin, "Matching 3D Models with Shape Distributions," *Proc. Shape Modeling Int'l,* pp. 154-166, 2001.

[26] D. Rueckert, A.F. Frangi, and J.A. Schnabel, "Automatic Construction of 3D Statistical Deformation Models of the Brain Using Nonrigid Registration," *IEEE Trans. Medical Imaging,* vol. 22, no. 8, pp. 1014-1025, 2003.

[27] S. Ruiz-Correa, L.G. Shapiro, and M. Meila, "A New Signature-based Method for Efficient 3D Object Recognition," *IEEE Soc. Conf. Computer Vision and Pattern Recognition,* vol. 1, pp. 769-1776, 2001.

[28] S. Ruiz-Correa, L.G. Shapiro, and M. Meila, "A New Paradigm for Recognizing 3D Object Shapes from Range Data," *Proc. IEEE Int'l Conf. Computer Vision,* vol. 2, pp. 1126-1133, 2003.

[29] S. Ruiz-Correa, L.G. Shapiro, M. Meila, and G. Berson, "Discriminating Deformable Shape Classes," *Advances in Neural Information Processing Systems 16,* S. Thrun, L. Saul, and B. Schölkopf, eds., Cambridge, Mass.: MIT Press, 2004.

[30] B. Scholköpf, J. Platt, A.J. Smola, J. Shawe-Taylor, and R.C. Williamson, "Estimating the Support of a High-Dimensional Distribution," *Neural Computation,* vol. 13, pp. 1443-1471, 2001.

[31] B. Scholköpf and A.J. Smola, *Learning with Kernels.* Cambridge Univ. Press, 2002.

[32] C.R. Shelton, "Morphable Surface Models," *Int'l J. Computer Vision,* vol. 38, no. 1, pp. 75-91, 2000.

[33] G. Taubin, "Estimating the Tensor of Curvature of a Surface from a Polyhedral Approximation," *IEEE CS Proc. Fifth Int'l Conf. Computer Vision,* p. 902, 1995.

[34] V.V. Vapnik, *Statistical Learning Theory.* John Wiley and Sons, 1998.

Salvador Ruiz-Correa received the BS and MS degrees in electrical engineering from the Universidad Nacional Autónoma de México, México, in 1990, and 1994, and the PhD degree in electrical engineering from the University of Washington, Seattle, in 2004. At present, he is a senior fellow in the Department of Radiology at the University of Washington and Children's Hospital and Regional Medical Center. His interests include computer vision, machine learning, and biomedical imaging applications.

Linda G. Shapiro received the BS degree in mathematics from the University of Illinois, Urbana, in 1970 and the MS and PhD degrees in computer science from the University of Iowa, Iowa City, in 1972 and 1974, respectively. She is currently a professor of computer science and engineering, professor of electrical engineering, and adjunct professor of medical education and biomedical informatics at the University of Washington. Her research interests include computer vision, image database systems, pattern recognition, and medical imaging. Dr. Shapiro is a fellow of the IEEE and the IAPR. She is a past chair of the IEEE Computer Society Technical Committee on Pattern Analysis and Machine Intelligence and is currently an editorial board member of *Computer Vision and Image Understanding* and of *Pattern Recognition*. She has served on the program committees of numerous vision and AI workshops and conferences and has coauthored three textbooks, one on data structures and two on computer vision.

Marina Meila received the MS degree in automatic control from the Polytechnic University of Bucharest in 1985 and the PhD degree in computer science and electrical engineering from the Massachusetts Institute of Technology in 1999. She is an assistant professor of statistics at the University of Washington. She held appointments at the Bucharest Research Institute for Computer Technology, the Polytechnic University of Bucharest, and the Robotics Institute of Carnegie Mellon University. Her long term interest is in machine learning and reasoning in uncertainty. She has worked in domains like vision and robotics, but, most of all, on the theoretical and algorithmic aspects of clustering, classification, and learning graphical probability models from data.

Gabriel Berson is an MD graduate of the University of Pennsylvania and the Albert Einstein College of Medicine and completed his residency in pediatrics at Montefiore Medical Center. He is a fellow in medical genetics at the University of Washington. His research interests include dysmorphology and the development of clinical decision support tools for physicians.

Michael L. Cunningham received the BA degree in biology and the MD degree from the University of Vermont in 1983, and 1988, respectively. He completed his pediatric residency training at the University of Washington in 1991, a birth defects fellowship in 1993, and the PhD degree in biological structures in 1996. Now, he is an associate professor of pediatrics, director of the Children's Craniofacial Center at Children's Hospital and Regional Medical Center and the Center for Craniofacial Research. His research interests lie in the genetics of craniosynostosis and the molecular developmental biology of the calvaria.

Raymond W. Sze received the MD degree from Robert Wood Johnson Medical School and completed postgraduate training in radiology at New England Deaconess Medical Center, Stanford Health Services, and Cincinnati Children's Hospital. His research interests include the development of molecular imaging techniques to study animal models of human craniofacial developmental diseases and pediatric brain cancer, quantitative imaging assessment of skull shape in children with developmental skull malformations, and development of imaging techniques to guide treatment decisions in patients with developmental central nervous system malformations or craniofacial malformations.