Support Vector Shape: A Classifier-Based Shape Representation

Hien Van Nguyen, *Member*, *IEEE*, and Fatih Porikli, *Senior Member*, *IEEE*

Abstract—We introduce a novel implicit representation for 2D and 3D shapes based on Support Vector Machine (SVM) theory. Each shape is represented by an analytic decision function obtained by training SVM, with a Radial Basis Function (RBF) kernel so that the interior shape points are given higher values. This empowers support vector shape (SVS) with multifold advantages. First, the representation uses a sparse subset of feature points determined by the support vectors, which significantly improves the discriminative power against noise, fragmentation, and other artifacts that often come with the data. Second, the use of the RBF kernel provides scale, rotation, and translation invariant features, and allows any shape to be represented accurately regardless of its complexity. Finally, the decision function can be used to select reliable feature points. These features are described using gradients computed from highly consistent decision functions instead from conventional edges. Our experiments demonstrate promising results.

Index Terms—Shape matching, 2D and 3D representation, support vector machines

1 Introduction

The shape of an object represents the geometrical information that is independent of the transformational (scaling, rotation, articulation, etc.) effects. Understanding shape is essential in many computer vision applications, from recognition of people and their actions in video surveillance to design and inspection in industrial manufacturing [6], [28].

Recent psychophysical findings [54] suggest that the perceptual representation of a shape is primarily based on qualitative properties whose topological structures remain relatively stable over transformational conditions. Other empirical studies [29], [44] have shown that the neural processing of shape in the brain is broadly distributed throughout the ventral (*what*) pathway that is involved in object recognition and the dorsal (*where*) pathway that is involved in spatial localization. In other words, an adequate mathematical representation of shape needs to be invariant to viewpoint changes and articulated object motion and discriminative enough to enable detection and classification.

Two main approaches dominate previous work on shape representation: Global approaches model an object as a whole segment, while part-based approaches advocate segmentation of shape into constituent regions. The drawback of a purely global approach is the exclusion of articulation and sensitivity to occlusion. The drawback of a purely part-based approach is that a consistent partitioning is generally not possible in the face of numerous combinations of possibilities and object shape variations. Besides, segmentation itself is ill-posed, except under controlled environments or in restricted application domains.

Global models cover a wide range of methods. Prominent global shape representations include variational [11] and level set approaches [45], [39]. These approaches have been applied for scene segmentation [13] and tracking [46], [48]. Brookstein initiated the use of thin-plate splines [8] to analyze deformable shape, which were then improved by [9], [10]. These methods are landmark based and suffer from inconsistency in landmark selection. Wang et al. [59] fit a parametric model to a shape using mixture of Gaussian densities. This method requires a clustering process to estimate cluster centers and therefore has the same drawback as other landmark-based approaches. Gorelick et al. [22] assign every internal point of the silhouette a value proportional to mean time of random walk from the point to the boundary. This can be achieved by solving a Poisson equation. Other popular methods are statistical moments [41], eigenshapes [23], curvature scale space [1], elastic matching [3], parametric curves (polylines), image signatures, etc. Zernike moments are a class of orthogonal moments that are invariant to rotation and translation. Eigenshapes decompose a distance matrix of boundary points into an ordered set of eigenvectors and finds the modes of these eigenvectors. Elastic matching evaluates the similarity as a sum of local deformations needed to change one shape into another. Scale space representation successively smooths to the contour while decreasing the number of curvature zero crossings. In general, global models need additional mechanisms to compensate for articulated motion and nonrigid deformation.

In comparison, part-based approaches describe shapes in terms of their part structure. Parts are defined to be nearly convex shapes separated from the rest of the object at concavity extrema [31], [32], [33], [37]. It is possible to build a discriminative classifier from a collection of parts [24] or a bag of feature to solve correspondence [5]. These methods

- *H. Van Nguyen is with the Department of Electrical and Computer Engineering, University of Maryland, College Park, MD 20740. E-mail: hien@umd.edu.*
- *F. Porikli is with the Mitsubishi Electric Research Laboratories (MERL), 201 Broadway 8th floor, Cambridge, MA 02472. E-mail: fatih.porikli@gmail.com.*

Manuscript received 16 Dec. 2011; revised 24 June 2012; accepted 22 Aug. 2012; published online 28 Aug. 2012.
Recommended for acceptance by K. Siddiqi.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-2011-12-0899.
Digital Object Identifier no. 10.1109/TPAMI.2012.186.

often require a multitude of training samples, prior knowledge on the number of parts, and precise formulation of articulation. Other part-based methods try to learn the part structure by considering the shape interior. For instance, shock graphs [51] are defined as the cyclic tree of singularities of a curve evolution. The inner distance [35], geodesic distance [25], and random walk [21] also consider the interior of the shape to build descriptors. Given a pair of points, the inner distance is determined by finding their closest points on the shape skeleton, then measuring the distance along the skeleton. The geodesic distance is the length of the shortest path on the surface, while shock graphs benefit from the skeleton's robustness to articulation, they suffer from boundary noise. The inner and geodesic distances are robust to disturbances along boundaries, yet they are highly sensitive to occlusions and fragmentations. Recently, Flach and Schlesinger [18] proposed Gibbs Random Fields to model shapes as spatial compositions of simple parts.

Pioneering work on the spin image [27] describes the relative spatial distribution of shape points around a set of feature points. It considers a cylindrical support region and accumulates a histogram of points. The shape context (SC) [4], [19] is similar to the spin image except that the support region is a sphere. Since both generate sparse matrices, the distance computation is sensitive to the shape structure. In [7], each shape is indexed based on a variety of features, such as inner distance, euclidean distance, contour distance, etc., that characterize pairwise geometric relationships between interest points on the shape. Shapes in the database are ordered according to their similarity to the query shape and similar shapes are retrieved using a scheme which does not involve shapewise alignment.

The above methods provide satisfactory results under ideal conditions with strong priors and clean segmentation masks. Their representation capacity substantially degrades when the shape boundary is noisy (part-based methods, shock graphs), the shape has internal crevices, branching offshoots, and excessive discontinuities (inner distance, spin images, shape context), and nonconforming articulations (global models). Besides, they would not necessarily extend to higher dimensions or generalize over most shape classes.

Quite differently from existing approaches, we propose a novel implicit shape representation based on Support Vector Machine (SVM) theory. Each shape is represented by a classification decision function obtained by training SVM with interior and exterior shape points providing positive and negative training samples, respectively. The radial basic function (RBF) kernel is used with SVM to make our representation rotation and translation invariant. The decision boundary is a hypersurface on the high-dimensional feature space that separates the positive labeled points, *shape*, from the negative labeled points, *its surroundings*. Our shape representation is not just the shape boundary or the decision function boundary but the function itself. Instead of using the edge or surface gradients on a discrete grid, we use the *gradient* of the classification decision function, which is an analytic function that is defined everywhere in the data space. Furthermore, the use of the RBF kernel enables SVM to model any complicated shape due to the infinite dimensional nature of the associated Hilbert space. Several other advantages are explained in the following section.

To summarize the main contributions, this paper

- proposes a novel method to represent 2D and 3D shapes using support vector classifiers,
- provides an in-depth theoretical analysis for a better understanding of this representation, and
- presents a variety of detailed experimental results to evaluate its performance on challenging datasets.

Section 2 introduces the SVS and explains in detail the SVS representation, choice of classification algorithms, and their stability and robustness properties. Sections 3 and 4 present alternative ways of selecting feature points and constructing local descriptors. Experimental results on several benchmark datasets are presented in Section 5 for 2D shapes and Section 6 for 3D range images.

2 Support Vector Shape

We define the SVS representation to be the decision function of a classifier. The name SVS comes from the fact that the decision function is parameterized by a set of support vectors that is learned from Support Vector Machine. Throughout the paper, the terms "SVS" and "decision function" are used interchangeably. This representation facilitates the extraction of feature points that correspond to salient components of the shape, which are then described using local statistics of the decision function around each point.

To the best of our knowledge, we are the first to consider the shape representation as a classification problem. This classifier-based representation offers several advantages. First, it is general enough to be applied to 2D shapes and 3D volumes. Second, the classification function depends only on a sparse subset of key points, which makes the representation robust against noise, missing data, and other artifacts, including interior fragmentations. Finally, the descriptors are also more discriminative and stable against transformation and disturbances than edge-based descriptors [27], [61], [35] since they are extracted from the dense gradient field of the decision function but not from the original data.

Basically, SVS involves the following tasks:

1. learn a decision function from a given shape,
2. select feature points using the gradient, i.e., the first derivative, of the decision function. (Section 3),
3. compute local descriptors (Section 4),

which are explained in the following sections.

As will be clear, SVS enables selecting a small set of salient features for shape matching and retrieval. These features are described using local statistics of the decision function around each point. For instance, a 2D variant of SVS features picks the high gradient points of the decision function as the feature points and uses the local histogram of oriented gradients (HOG) computed on the decision function as the descriptors.

2.1 Learning a Decision Function

Let $S = \{\mathbf{x}_n\}_{n=1}^{N}$ be a set of points representing a shape[1] and $\bar{S}$ be the set of points not in S, i.e., outside. We wish to learn a classifier

$$f(\mathbf{x}) = \begin{cases} \geq 0, & \mathbf{x} \in S \\ < 0, & \mathbf{x} \in \bar{S}. \end{cases} \quad (1)$$

Two classifiers, $f(\mathbf{x})$ and $g(\mathbf{x})$, are said to represent the same shape if

$$\text{sign}[f(\mathbf{x})] = \text{sign}[g(\mathbf{x})] = \begin{cases} \geq 0, & \mathbf{x} \in S \\ < 0, & \mathbf{x} \in \bar{S}. \end{cases} \quad (2)$$

The following theorem states that if the binary shapes generated by taking the zero-level crossings of these classifiers are equivalent (as in 2) to each other, then the decision functions are equivalent with a constant factor.

Theorem 1 (Curtis and Oppenheim [15]). *Let $f(\mathbf{x})$ and $g(\mathbf{x})$ be real, 2D, band-limited, and irreducible functions. If $f(\mathbf{x})$ and $g(\mathbf{x})$ take on both positive and negative values in a closed-bounded region $D \subset \mathbb{R}^2$ and $\text{sign}[f(\mathbf{x})] = \text{sign}[g(\mathbf{x})]$ for all $\mathbf{x}$ in D, then $f(\mathbf{x}) = \kappa g(\mathbf{x})$, where κ is a real positive constant.*

Here, *irreducible function* means that its Fourier transform is not factorable. The band-limited condition implies the Fourier transform will exist and it will have a compact region of support for finite-energy signals (interested readers are referred to [53] for further discussions on band-limitedness condition).

The above theorem suggests that the gradient orientation is equivalent for such two functions satisfying the above constraints. In other words, two decision functions representing the same shape will be consistent and exhibit invariance properties in terms of their gradient orientations if these functions are real, 2D, band-limited, irreducible, and have almost identical responses.

One such example is the radial basic function that support vector machine [58] operates on. It is real and band limited: It has the form of the sum of finite number of weighted exponential; thus, its Fourier transform is a sum of weighted exponentials, which has finite energy and a compact region of support especially when insignificant coefficients of the radial kernels are disregarded. A theorem for the irreducibility of the general class of functions including the RBF kernel is given below, with the proof in Section 7.

Theorem 2. *A function of the following form:*

$$f(\mathbf{x}) = \sum_{i=1}^{m} \alpha_i g(\mathbf{x} - \mathbf{x}_i^*), \quad m \geq 5, \quad (3)$$

is irreducible if the Fourier transform of $g(\mathbf{x})$ does not have any zero (either real or complex).

For an ideal shape classifier as defined in (1), the decision function is positive for the shape regions, zero on the shape boundary $\mathcal{B}$, and negative otherwise. In other words, the gradient of the decision function along the tangent space of a point on the boundary $\mathcal{B}$ is zero (see Fig. 1). This means that the gradient of the decision function must be perpendicular to the tangent plane; thus, the gradient itself coincides with the normal vector of the shape.

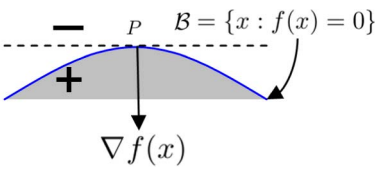


Fig. 1. Illustration of SVS decision boundary. The dotted line is the tangent space.

Proposition 1. *Gradient $\nabla f(x)$ at a shape boundary point is a close approximation for the normal vector at that point.*

This property is very desirable for SVS since the normal direction is an essential input for the construction of many descriptors [4], [16], [27]. It is especially useful for computing descriptors from 3D point clouds where the knowledge about points orientations is missing.

Moving from the boundary $\mathcal{B}$ to the interior of shape, the gradients $\nabla f(x)$ result from combining effects of local edge segments. This effectively creates fusions of local topologies that enhance the discriminative power of local descriptors (Section 4).

We are interested in representing a shape by a continuous function that has the mathematical form of (3) and at the same time satisfies (1) as much as possible. Notice that the class of functions in (3) contains the decision function of SVM with the RBF kernel [50], [49]. Therefore, we employ SVM to learn our parametric shape representation. To our advantage, the SVM decision function is analytic, i.e., it is in the form of weighted sum of kernel responses; thus, its gradient can be efficiently computed at a point in the space. Furthermore, the RBF kernel functions can effectively map data $\mathbf{x}_n$ to an ∞-dimensional space where S and $\bar{S}$ would be linearly separable; thus, even for intricate boundaries, a high classification performance, and therefore accurate shape representation, is guaranteed.

SVMs construct a hyperplane in a high (or infinite)-dimensional feature space between a set of labeled input vectors $\mathbf{x}$ that can be either $+1$ for shape pixels or -1 for nonshape pixels by definition for binary SVMs. The decision boundary is defined in terms of a typically small subset of training examples, called support vectors, that result in a maximum margin separation between these classes. The decision function of SVM is given as

$$f(\mathbf{x}) = \sum_{i=1}^{m} \alpha_i [\Phi(\mathbf{x}).\Phi(\mathbf{x}_i^*)], \quad (4)$$

where $\mathbf{x}_i^*$ are support vectors, α_i are the corresponding weights of the support vectors, m is the number of nonzero support vectors, and Φ is a mapping function in some dot product space $\mathcal{H}$. By defining a similarity measure k in $\mathcal{H}$ as

$$k(\mathbf{x}, \mathbf{x}_i^*) = \Phi(\mathbf{x}).\Phi(\mathbf{x}_i^*), \quad (5)$$

every dot product in the decision function is replaced by a kernel function. This allows efficient computations without having to venture into the high-dimensional feature

1. We denote vectors in bold letters.

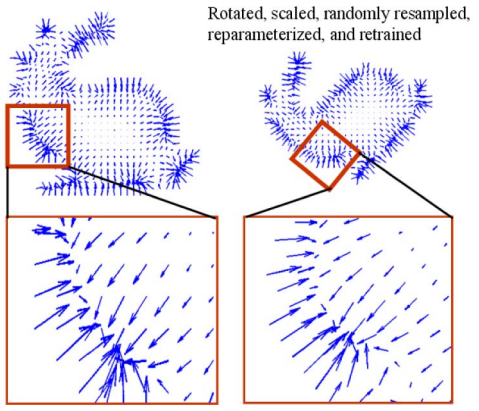


Fig. 2. The orientation of $\nabla f\mathbf{x}$ remains stable even if the shape transforms. A different set of parameters and points is used to train the SVS on the right.

space $\mathcal{H}$. The transformation may be nonlinear; thus, though the classifier is a hyperplane in $\mathcal{H}$, it may be nonlinear in the original input space. If the kernel used is a Gaussian, the corresponding feature space is a Hilbert space of infinite dimension:

$$\Phi(\mathbf{x}).\Phi(\mathbf{x}_i^*) = \exp\big(-\gamma\|\mathbf{x}-\mathbf{x}_i^*\|^2\big), \tag{6}$$

where γ stands for the Gaussian kernel width. By using the RBF, it is always possible to find a decision function that perfectly represents a shape. Such a decision function has the form:

$$f(\mathbf{x}) = \sum_{i=1}^{m} \alpha_i \exp\Big(-\gamma\|\mathbf{x}-\mathbf{x}_i^*\|^2\Big) \tag{7}$$

and the final classification is made by $l(\mathbf{x}) = \text{sign}[f(\mathbf{x})]$. It is worth noting that the set of support vectors is usually small in comparison with the entire training set.

Since the decision function in (7) only depends on the distance between points, the SVS representation is robust to translation and rotation.[2] This can be observed from Fig. 2 which shows that the gradient directions computed from SVS decision functions of transformed versions of the same shape are almost the same.

Furthermore, the γ multiplier in the kernel function is inversely proportional to the squared scale change. For a given shape with unknown scale change from its original, the mean of pairwise distances between all points can be used to normalize the shape. This scale normalization technique has been used in shape context [19] and has been proven to be effective. In addition, our analysis demonstrates that small variations of γ would not perturb the best possible classification accuracy.

For training, we select a random subset of internal points to be positive training samples from a given shape. Another random subset of points surrounding the shape is chosen to be negative samples. Random selection is preferred just for computational efficiency. The input to the classifier is the coordinates and the corresponding inside/outside labels of the training points.

2. Note that for linear and polynomial kernels such an invariance does not apply as they impose inner products of point coordinates.

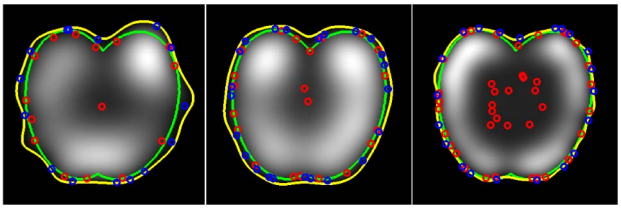


Fig. 3. Support vectors (red circles $\alpha_i > 0$, blue circles $\alpha_i \leq 0$), the decision function boundaries $f(\mathbf{x}) = 0$ (yellow), and the original shape (green) for support vector numbers 25, 44, and 61. Classifier accuracies are 93, 97.2, and 99.8 percent, respectively. The original shape contains $\sim$ 54K points ($\sim$ 1K points on the boundary), yet only a small fraction (e.g., 61) of points are needed to encode the shape.

Fig. 3 shows support vectors and decision boundaries for a sample shape. It can be noticed that support vectors are not required to lie on shape edges. This is because the kernel is mapping data to the high-dimensional space, where nonedge points might happen to lie on the decision boundary of the learning algorithms. The number of support vectors typically varies from 0.1 to 3 percent of the total number of points.

2.2 ν-SVM and One-Class SVM

We employ ν-SVM [50] and one-class SVM [49] for learning the decision function as its parameters have a natural interpretation for shapes. Given a set of labeled samples (x_i, y_i), the learning problem of ν-SVM is formulated as the minimization of

$$\underset{\mathbf{w},\xi,\rho}{\arg\min} \ \frac{1}{2}\|\mathbf{w}\|^2 - \nu\rho + \frac{1}{l}\sum_{i=1}^{l}\xi_i \tag{8}$$

subject to

$$y_i.(\mathbf{w}.\Phi(\mathbf{x}_i) + b) \geq \rho - \xi_i, \tag{9}$$

$$\xi_i \geq 0, \quad \rho \geq 0, \tag{10}$$

where Φ is a function that maps the input data to some Hilbert space. The above optimization tries to correctly classify as many data as possible by penalizing the misclassified samples through variable ξ_i. At the same time, the minimization of $\|\mathbf{w}\|$ keeps the model as simple as possible, and the margin is made as large as possible through maximization of variable ρ.

The tradeoff between the model complexity and the training error is controlled by parameter $\nu \in [0, 1]$. It is also the lower and upper bound on the number of examples that are support vectors and lie on the wrong side of the hyperplane, respectively. The larger (smaller) we make ν, the more (less) points are allowed to lie inside the margin, which gives coarser (finer) shape representations. It is possible to use a small value which results in a larger number of support vectors to allow accurate representation of complex shapes, while smaller numbers of support vectors enhance the robustness against corrupted training data.

The formulation and parameters of one-class SVM is similar with ν-SVM. The only difference is that one-class SVM allows learning from data with positive samples only. It separates data from the origin in the feature space instead

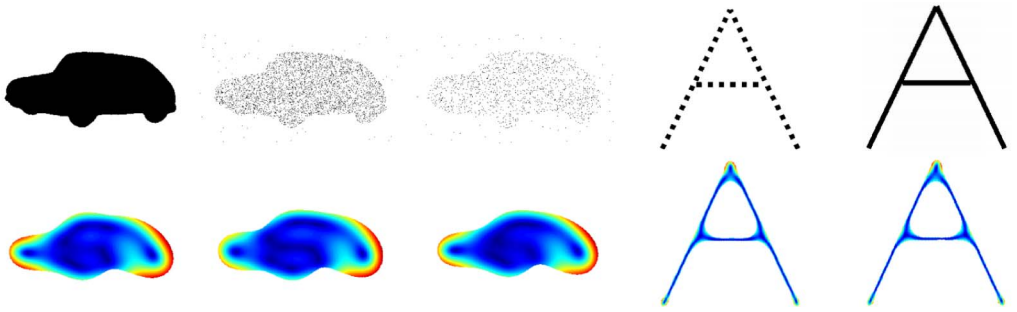


Fig. 4. Comparison of the decision function responses for noise-free and noisy shapes. In the first experiment, a car shape is distorted by randomly removing 20 and 50 percent of points and adding speckle noise to the background. In the second experiment, the letter A is interleaved with blank spaces. The differences of decision function for both cases are hardly noticeable. They are both trained using one-class SVM.

of separating positive samples from negative samples. This becomes extremely useful to deal with missing data due to occlusion and camera sampling error.

The selection of parameters for SVM algorithms (e.g., kernel width γ of (6), error margin ν of (8)) is done automatically by imposing a constraint on the cross-validation accuracy. Specifically, we divide the data points into two subsets, one for training and another one for testing. We then perform cross validation and select the set of parameters that produce a classification accuracy higher than 99 percent. Note that if heavy occlusions are present, we allow the classification accuracy to be lower.

Fig. 4 illustrates the robustness of SVS representation under different noise effects. In particular, we compare the color-coded response of the decision functions for a car shape and an A-letter shape before and after being distorted. As for the car, we randomly remove pixels from the shape and also add noise to the background. The A-letter shape is interwoven with white spaces to create distortion of the shape boundary. For both cases, it can be seen from Fig. 4 that the color-coded responses, and thus the associated decision functions, remain quite stable.

From the computational perspective, SVS complexity is essentially the same with that of SVM algorithms. In general, it is polynomial in the number of input points. In our experiments, it takes about 0.15 seconds to compute SVS for a shape in the MPEG7 dataset [30] using a 2.4 GHz Quad Core machine. The learning process can be significantly sped up using approximate variants of SVM [26], [55]. The query of the decision function is very efficient. It is linear in the number of support vectors, which is a small fraction of the total number of points (e.g., 1 percent). In addition, it can be accelerated by two orders of magnitude [12] using statistical approximations.

3 SVS Feature Points

Shape matching algorithms using SVS representation are comprised of two constituents: feature (interest) points and their descriptors. This section discusses possible ways of selecting the feature points. All these methods are based on the previously explained decision function $f(x)$.

The feature points are desired to be stable under local and global shape perturbations (including affine transformations and articulated motion) as well as noise and other artifacts. Such feature points should be reliably computed with high degree of reproducibility. In addition, the local descriptors computed at these feature points have to be discriminative enough for reliable shape matching and registration; therefore, structure around the feature points should be rich in terms of local information content. In what follows, we will elaborate on different possibilities of selecting good feature points for SVS representation.

3.1 Gradient-Based Feature Points

A corollary of Theorem 1 is that the gradient orientation is stable while gradient magnitude differs only by a constant factor. The gradient orientation is given by

$$\frac{\nabla f(\mathbf{x})}{\|\nabla f(\mathbf{x})\|}, \quad \text{where} \tag{11}$$

$$\nabla f(\mathbf{x}) = 2\gamma \sum_{i=1}^{m} \alpha_i \exp\big(-\gamma \|\mathbf{x} - \mathbf{x}_i^*\|^2\big)\big(\mathbf{x}_i^* - \mathbf{x}\big). \tag{12}$$

To evaluate the stability of the gradient orientation, we randomly choose a set of 500 points on each of 70 SVSs created from different shapes in the MPEG7 database [30] and examine their gradients as the training parameters (γ, ν) vary (γ by $6\times$ and ν by $10\times$ with respect to the smallest value of each parameter). Note that each different parameterization may generate a different decision function in magnitude; however, we are interested in how the *direction* of the gradient of the decision function changes. Therefore, to account for the multiplication factor of the decision function, we normalize the decision function values by their mean value yielding relative magnitudes.

Fig. 5 shows how the standard deviation of gradient direction changes with respect to the relative gradient magnitudes for 500 points from one of the 70 SVSs. One can easily notice that the gradients vary with respect to the training parameters. This is because large variation of the training parameters (γ, ν) results in different classification functions that do not strictly satisfy the condition $\text{sign}[f(\mathbf{x})] = \text{sign}[g(\mathbf{x})]$ as in Theorem 1.

However, the variation exhibits a strong dependency on gradient magnitude. The higher the gradient magnitude gets, the smaller the standard deviation is. For points with gradient magnitude of more than two, the standard deviation, which directly corresponds to direction changes, is as small as 4 degree. Note that in practice the variation should be smaller than what we see in Fig. 5 since the

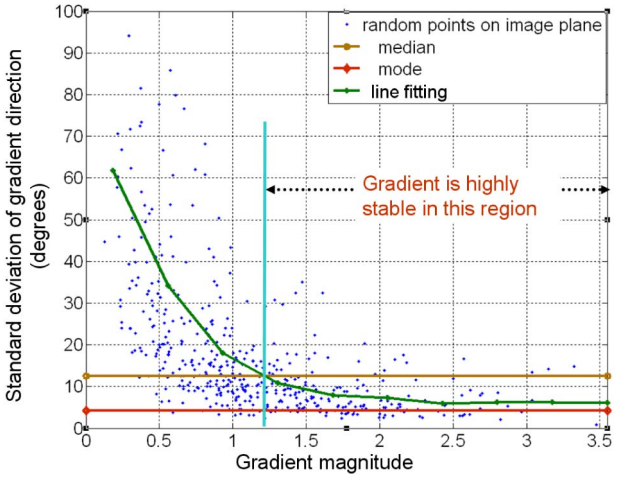


Fig. 5. The relative gradient magnitude at a point is computed by dividing its gradient magnitude by the mean of gradient magnitude across the entire shape.

constraint of high classification accuracy implicitly requires a consistent set of parameters (γ, ν).

Since the gradient orientation is stable especially for higher gradient magnitude points, we choose a small subset of such points for matching. We apply an iterative search method that finds the maximum gradient magnitude point on $\nabla f(\mathbf{x})$ until it selects 100 points and build a list by ordering them according to their angles from the center of the shape in a circular fashion. The starting (0 degrees) orientation of circular sweep is set with respect to a dominant gradient direction of $\nabla f(\mathbf{x})$.

Severe occlusions or distortions might lead to change of the dominant gradient directions. In such a scenario, we allow the generation of multiple sets of descriptors corresponding to different tentative orientations. During the shape matching phase, the distance between two shapes is the smallest matching cost among all the orientations. Fig. 6 shows the selected feature points for a sample shape.

3.2 Support Vector-Based Feature Points

The support vectors are sufficient to construct the decision function and its gradient; thus, they are good candidates for feature points.

One can ask whether the set of support vectors enable reliable shape matching. The answer to this question largely depends on the problem at hand. For nonarticulated transformations, the support vectors remain stable. In Fig. 7, we show that support vector locations are quite

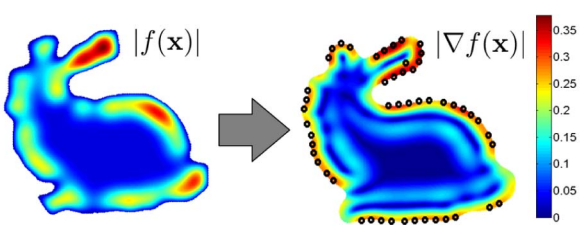


Fig. 6. Decision function responses (left) and the gradient magnitude of the decision function (right) for the SVS representing the Stanford Bunny. Points with higher values in $\nabla(f\mathbf{x})$ seem to be good candidates to be selected as discriminative feature points. Black circles are the 51 feature points (102 support vectors).

Fig. 7. Stability of support vectors with respect to variation of RBF kernel width (γ) and articulation. **Top**: Support vectors when γ varies. $\gamma = 0.0027$ in the first shape and $\gamma = 0.0054$ in the second shape. Sixty-two out of 78 support vectors in the first shape appear in the second shape. The overall positioning of the support vectors is very similar. **Bottom**: Support vectors with shape articulation. Blue circles indicate SVs that do not have correspondences on the other shape. Eighty-three out of 114 SVs in the first shape appear in the second shape.

stable when the kernel width varies $2\times$. Constraining the classification accuracy to be sufficiently high (e.g., 99 percent) and preventing the kernel width from changing too much would produce similar support vectors. Yet, if the kernel width changes too much, e.g., $20\times$, support vectors change significantly, as in Fig. 3. Besides, the support vectors are sensitive to shape articulations due to the topology changes, as illustrated in Fig. 7 bottom pair.

3.3 Curvature-Based Feature Points

It is possible to select points with high curvature on SVS by looking at the Hessian matrix of a decision function. More specifically, for the 2D case, we first solve the eigenvalues of the Hessian matrix, which is proportional to curvatures:

$$\mathbf{H} = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} \\ \frac{\partial^2 f}{\partial x_1 \partial x_2} & \frac{\partial^2 f}{\partial x_2^2} \end{bmatrix} = \mathbf{Q} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \mathbf{Q}^{-1}. \quad (13)$$

Points with high curvatures associate with large eigenvalues of both dimensions (λ_1, λ_2). We randomly sample 200 points to compute curvatures. Feature points are chosen where the first eigenvalue λ_1 and the second eigenvalue λ_2 are larger than the median of λ_1 and λ_2, respectively. To mitigate noise effects, the decision function can be smoothed at different scales before computing curvatures. Gaussian smoothing of the decision function (7) is a mixture of Gaussian functions, which can be computed efficiently with a closed-form expression.

An advantage of this selection scheme is that local descriptors are highly discriminative. For instance, SIFT can be computed to find pointwise correspondences for aligning two similar shapes. However, this point selection method is not appropriate for shape matching. For example, choosing only those points around corners makes it

impossible to differentiate a rectangle from a square. This disadvantage arises for shapes characterized mainly by their dimensions and shapes characterized by the arrangement of similar local structures. Another disadvantage of this method is the difficulty of computing curvatures when generalizing to higher dimension.

3.4 Entropy-Based Feature Points

Another possibility is selecting a small subset of points whose local gradient orientation have high entropy. To compute this entropy, we first create a histogram of gradients over the local window of size 0.25×0.25 (with respect to mean pairwise distances). The entropy is then computed as follows:

$$-\sum_{i=1}^{n} p_i log_2(p_i), \tag{14}$$

where p_i is the weight of the ith bin and n is the number of bins in the histogram. High entropy is equivalent to high variation of gradient orientations. Therefore, it is a good indication of complex local topologies of SVS, and thus more discriminative local descriptors. This strategy has similar philosophy as the high-curvature selection method. However, it does not involve the computation of principal curvatures, which can be difficult for higher dimensional cases.

4 SVS Descriptors

SVS facilitates the computation of a set of descriptors extracted from the decision function. Below, we give only a few examples of possible descriptors for 2D and 3D data.

4.1 Two-Dimensional Descriptors

We compute a local histogram of oriented gradients descriptor around each of the points *on the decision function gradient* ∇f but not on the conventional edge gradient; thus, our HOG_f is significantly different from the existing descriptors.

For a given feature point, a 4×4 array of eight-bin orientation histograms is constructed within a local window. The size of the window is set to be 0.25×0.25 (relative with respect to mean pairwise distance). Our experiments indicated that this size provides satisfactory results for both very coarse and fine shapes. A histogram is populated for each subblock by aggregating the gradient magnitude of the decision function into the corresponding orientation bin of the gradient direction of the decision function.

Since gradients with larger magnitudes are more stable, the contribution of each gradient to the histogram is set to be proportional to its magnitude. We impose a Gaussian kernel to weight gradients based on their relative distances with respect to the feature point. This spatial weighting puts more emphasis on the gradients that are closer to the center and helps improve the discriminative power of the local descriptors.

To prevent problems due to the coarse binning issues, the value of each gradient point is interpolated into adjacent histogram bins. Finally, the mean of gradients for the local window is taken to be the orientation of the descriptor and the histogram bins are reoriented accordingly with respect to this mean orientation to achieve the rotation invariance of the descriptor. The histograms are then concatenated into a 128-dimensional vector.

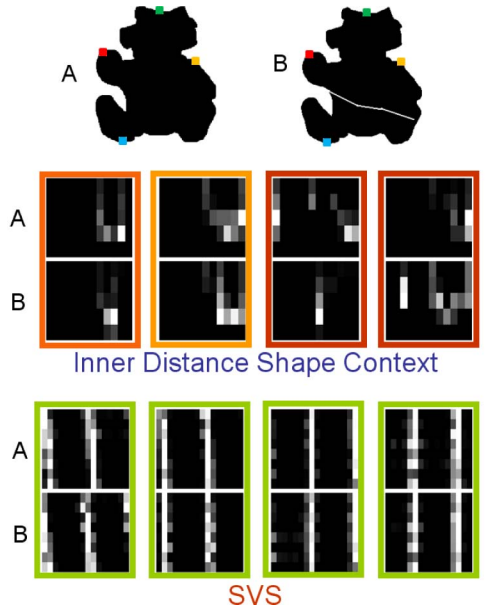


Fig. 8. Comparison of IDSC with the SVS computed at four locations indicated by square dots on the shapes. Descriptors are displayed from left to right corresponding to the following order of dots' colors: red, yellow, green, and cyan. The second image has a crack in the shape. IDSC changes drastically (it highly depends on the boundary), while the SVS remains robust generating almost the identical descriptors. Box color indicates the severity of the mismatch: green (< 20 percent), orange ($20 \rightarrow 50$ percent), red (> 50 percent).

Fig. 8 shows a comparison of the Inner Distance Shape Context descriptors (IDSC) [35] with the SVS descriptors for a pair of images where one contains irregularity in the shape (assuming that, even after some morphology, such artifacts remain). The inner distance responses change drastically, while the descriptors computed from SVS stay very robust. Fig. 9 demonstrates the strength of the SVS descriptors even for very noisy data (note that fitting an outer shell, morphological filtering, etc., would not help for this shape as the noise is dispersed into the background). In this case, we compare with Shape Context since IDSC will not work due to the disconnected interior. To the best of our knowledge, there is no other representation that can give such a robustness that SVS provides for heavily noisy and disconnected data. For all of the experiments presented in Figs. 8 and 9, we use one-class SVM to learn the decision functions.

4.2 Three-Dimensional Descriptors

Concentric Ring Signature (CORS) [2] is used as the descriptor for our 3D feature points. CORS is constructed in a circular fashion around a center point. Each point is projected onto a fitted plane. The orthogonal distances are indexed by rotation angles and radial distances. The advantage of CORS over other 3D descriptors, e.g., the spin image proposed by Johnson and Hebert [27], is that it is more compact and discriminative. Another advantage is that CORS computation can be accelerated if points are oriented. Since SVS provides each point cloud with a stable orientation of the decision function, it allows the efficient construction of CORS descriptors.

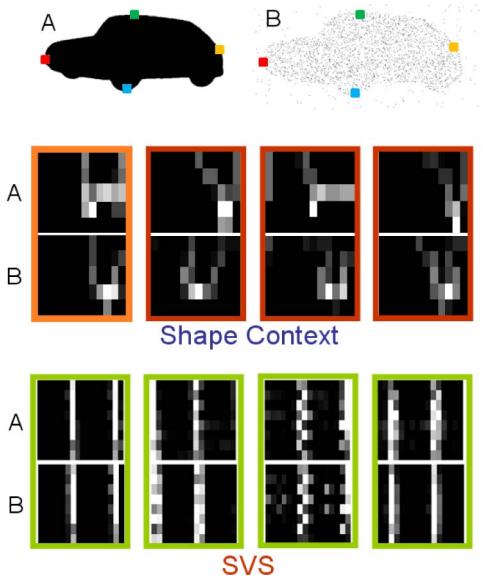


Fig. 9. Comparison of shape context with the SVS computed at four locations, indicated by square dots on the shapes. Descriptors are displayed from left to right corresponding to the following order of dots' colors: red, yellow, green, and cyan. Box color indicates the severity of the mismatch: green (< 20 percent), orange ($20 \rightarrow 50$ percent), red (> 50 percent).

5 Experiments in 2D

For our experiments, we need to establish the distance between two shapes. Given two SVS decision functions $f_A(\mathbf{x})$ and $f_B(\mathbf{x})$, for shapes A and B, respectively, we compute a distance score between their descriptors.

As we explained before, since the feature points are already ordered with respect to the dominant gradient direction, initial alignment for the shapes with similar overall structures is almost accurate. Next, we use the local descriptors for comparison of two shapes. The advantage of using local descriptors is that they are robust against occlusion and shape articulation.

Let two sets of the descriptors for two shapes A and B be $\Lambda^A : \{\lambda_1^A, \lambda_2^A, ..\lambda_t^A\}$ and $\Lambda^B : \{\lambda_1^B, \lambda_2^B, ..\lambda_s^B\}$, where t does not have to be equal to s assuming $t \geq s$. The correspondence is established through a mapping function h such that $h : \Lambda^B \rightarrow \Lambda^A$. If a descriptor λ_i^B is matched to another λ_j^A, then $h(i) = j$. We define a cost function as

$$\boldsymbol{E}(h) = \sum_{1 \leq i \leq s} \epsilon(h(i), i), \tag{15}$$

where the descriptor distance is computed using χ^2 statistic:

$$\epsilon(h(i), i) = \sum_{1 \leq k \leq 128} \frac{\left[\lambda_{h(i)}^A(k) - \lambda_i^B(k)\right]^2}{\lambda_{h(i)}^A(k) + \lambda_i^B(k)}. \tag{16}$$

This cost represents the overall distance for the corresponding pairing of the descriptors. Note that the mapping h is neither one-to-one nor overlapping, but keeps the ordering of the descriptors.

To minimize $\boldsymbol{E}$, we use dynamic programming to find the solution to (16). It is worth noting that the start points and the end points of two sequences are already roughly aligned for the dynamic programming algorithms to converge to the correct solution. Under certain conditions, the initial alignment provided by the ordered lists may not be valid. To overcome this, we find and compensate for the angle that maximizes the correlation between two "global" histogram of oriented gradients, which are defined as HOG_f yet computed for the entire shapes, of two given shapes before minimizing the above cost function.

TABLE 1
Comparison of MPEG7 Classification Results Using Four Different Feature Point Selection Methods in Section 3 Based on Gradients, Support Vectors, Curvatures, and Entropies

Method	Gradient	SV	Curvature	Entropy	Edges
Accuracy (%)	91.07	70.13	72.52	62.25	30.25

The last column shows the baseline classification result when gradients are computed from edges instead of SVS.

We run the first set of experiments on the entire MPEG7 shape benchmark dataset [30], which has 70 classes and 20 shapes for each class, a total of 1,400 images. The performance is measured by the standard Bullseye test. For each shape, the retrieval accuracy is measured by counting how many of 20 correct shapes are in the top 40 matches.

For the gradient-based method, we apply an iterative search method that finds the maximum gradient magnitude point on $\nabla f(\mathbf{x})$ until it selects 100 features. In the entropy-based feature selection method, 200 points are randomly selected in each shape where HOG_f descriptors and their entropy are computed. Points associated with entropy larger than 1.5 times the median of overall entropy value are selected for matching. In the curvature-based method, we also randomly sample 200 points to compute curvatures. Feature points are chosen where the first eigenvalue λ_1 and the second eigenvalue λ_2 are larger than the median of λ_1 and λ_2, respectively. For the support vector-based method, we randomly select half of the support vectors set as feature points for shape matching (we observed that using only the most significant support vectors in fact deteriorates the performance).

Table 1 gives the correct retrieval percentages for four feature point selection methods (gradient based, support vector based, curvature based, and entropy based) presented in Section 3. Results indicate that the gradient-based method produces more consistent feature points and better matching performances. We also compare our performances with HOG computed on the edges gradients (30.25 percent) to verify that SVSs improve the discriminative power of our descriptors. Edge gradients are commonly used to extract feature points for 2D shape representation.

As in Table 1, the best overall accuracy on the MPEG7 dataset using the Bulleyes test is 91.07 percent, which is based on the gradient-based feature selection method. This is better than the performances of SC [4] (76.51 percent), IDSC [35] (85.40 percent), the post-refined version of [7] (86.48 percent), and shape tree [17] (87.70 percent), as summarized in Table 2. The best performance, 93.67 percent, is reported in [20]; however, their part-based algorithm is highly sensitive to segmentation errors especially for noisy data. A matching scheme [60], which takes into account the influence of the other shapes while computing the similarity of a pair of shapes, has reported an accuracy of 93.32 percent. SVS can be effectively used as

TABLE 2
Comparison on MPEG7 Data Set

Algorithm	Accuracy (%)
Gopalan [20]	93.67
IDSC + LCDP [60]	93.32
Mixture of Gaussian + tSL [38]	89.1
Shape-tree [17]	87.7
IDSC + DP + EMD [36]	86.56
Biswas [7]	86.48
Hierarchical Proscrustes [40]	86.35
IDSC + DP [35]	85.4
Shape L'Âne Rouge [47]	85.25
Generative Models [56]	80.03
Curve Edit [52]	78.14
SC+TPS [4]	76.51
Visual Parts [30]	76.45
CSS [43]	75.44
SVS + DP	**91.07**

TABLE 3
MPEG7 Classification Results versus the Number of Features Selected Using the Gradient-Based Selection Strategy

♯ Features	10	50	100	200	300
Accuracy (%)	15.93	68.83	91.07	90.38	89.92

TABLE 4
Random Distortion Results on the Partial MPEG7 Dataset

Method	IDSC	Denoised IDSC	SVS-BoW	SVS-DP
Accuracy	57.0%	85.9%	81.4%	93.7%

the shape representation in [60]. Fig. 10 shows sample retrieval results in descending order of matching scores using the gradient based features.

We also investigate the tradeoff between the model complexity and the classification performance by varying the number of feature points from $10 \rightarrow 300$. The results are summarized in the Table 3. In general, more feature points would yield better accuracies. However, it can be noticed that the classification accuracy suffers slightly when going beyond 100 features. This is because a large number of features inevitably leads to less-discriminative interest points being selected, therefore negatively interfering with the shape matching algorithm.

In the second experiment, we pick a subset of six shape classes (eight samples per class) from the MPEG7 database and add random distortions into them. The performance is measured by counting how many of eight correct shapes appear in the first 16 matches for each shape. Instead of using ν-SVM algorithm like in the first two experiments, we use one-class SVM for this experiment since it produces better retrieval results. In addition to classification using SVS and IDSC, we zero threshold our SVS representations to reconstruct the original binary shapes. These binary shapes are then used as the input for computing IDSC. We call this scheme *denoised IDSC*.

The overall performance of IDSC is 57.0 percent, while the SVS method, using gradient-based feature points, gives 93.7 percent, as given in Table 4. Besides, Table 4 shows that denoised IDSC (85.9 percent) performs significantly better than IDSC. The result implies that our shape representation provides some denoising effects over random distortions. We also compare the performance of SVS when using the bag-of-word approach (81.4 percent) instead of the dynamic programming technique. Fig. 11 shows retrieval results for both IDSC and SVS methods in descending order of the matching scores. IDSC's performance dramatically degrades on this database because artifacts within a shape severely change the inner distances. Shape context and other edge-based methods will also have the same issue. In contrast, the SVS method invariably gives accurate retrieval results thanks to classifier-based representation.

In the third set of experiments, we test our matching algorithm with the gradient-based feature selection strategy on the articulation database reported in [35]. This database includes 40 images from eight objects with different articulations, as shown in Fig. 12. This consists of highly similar shapes like types of scissors with only minor differences but significant articulations. The recognition result is evaluated for each shape by choosing the four most

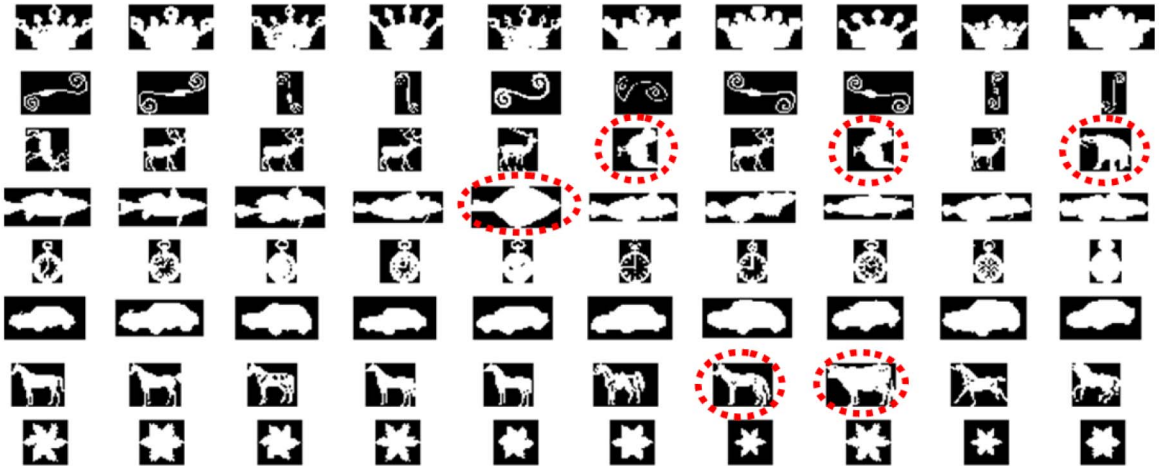


Fig. 10. SVS results: Red circles show the incorrect matches. Note that none are in top four rankings.

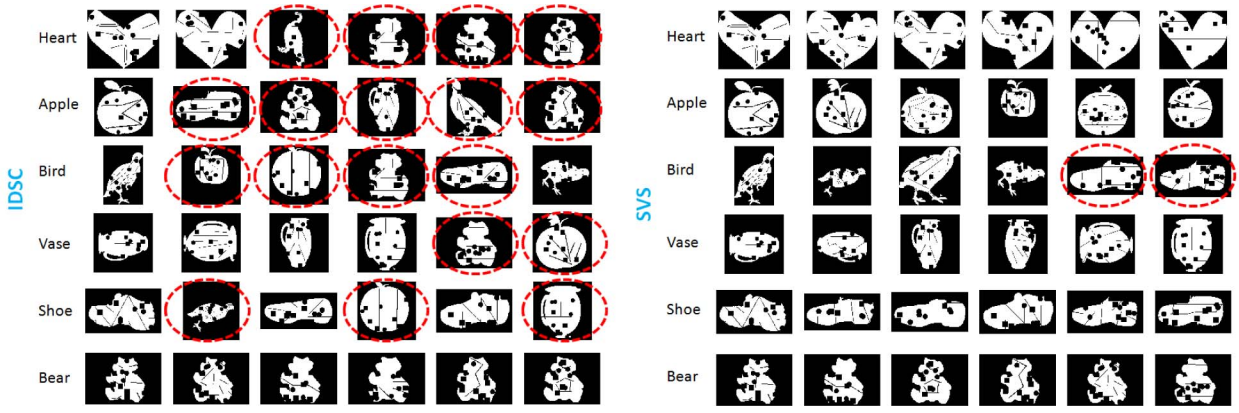


Fig. 11. Comparison of the retrieval results for the noisy dataset. The accuracy of IDSC is 57.0 percent and SVS is 93.7 percent. Red circles show incorrect matches.

similar matches and then sorting them according to their matching scores. Table 5 summarizes the matching results. We compare against the ℓ_2 distance on a bag of features, the shape context, the inner distance with shape context, the multiple feature indexing [7], and the layered graph matching [34]. In this experiment, the SVS uses the gradient-based feature point selection. It is apparent that our method handles articulation as well as (even better for later matches) IDSC, and much more accurately than the SC thanks to the robust nature of the SVS feature point selection and the locality of its descriptor.

6 Experiments in 3D

For many recognition and registration tasks, it is desirable for features to be selected consistently at similar locations on both target and model shapes. In the first experiment, we examine the repeatability of our 3D/2.5D feature points selection using the 3D Stanford shapes [14], [57]. The database contains both full 3D shapes and their 2.5D range images. Repeatability is the percentage of features which are detected on the target shapes and correctly matched to their associated locations on the 3D model shapes.

Similarly to 2D, we use one-class SVM to learn a decision function for 3D/2.5D data using (x, y, z) coordinates of points. Features points are selected where entropy of local HOG_f is larger than a predefined threshold. In construction of HOG_f, $\nabla f(\mathbf{x})$ is computed at uniformly sampled points within the local sphere. The sphere's radius is set to be five times the scanner resolution.

After selecting a subset of discriminative points, local descriptors are computed for recognition and registration tasks. We use CORS as the local descriptors since it is compact, discriminative, and easy to compute. In addition, a consistent orientation ∇f at each point enables efficient approximation of CORS and reduction of computational cost. Readers are referred to [2] for details on the construction and approximation of CORS descriptors.

Fig. 13 shows the repeatability scores obtained by our method. Fig. 14 shows sample features matching between two range scan images of the Buddha shape.

In the second experiment, we perform 3D objects recognition and registration using the publicly available database [42], which contains five objects and 50 scenes. Our goal is to detect if there is any objects of interest in a scene and register the full 3D shapes to any detected objects. An object is said to be correctly detected if the resulting errors of the translation and pose estimations compared to

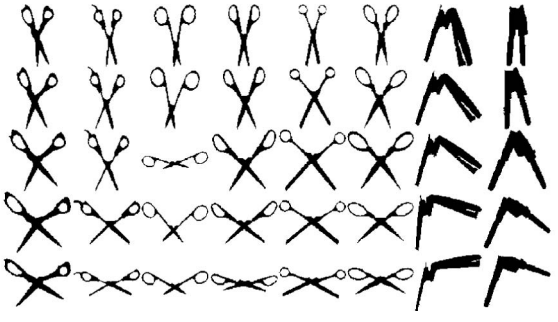


Fig. 12. Samples from the articulation database. Note that each **column** corresponds to a different object.

TABLE 5
Matching Results on the Articulation Dataset

Descriptor type	1^{st} Match	2^{nd} Match	3^{rd} Match	4^{th} Match
ℓ_2 (baseline)	25/40	15/40	12/40	10/40
SC [4]	20/40	10/40	11/40	5/40
IDSC [35]	40/40	34/40	35/40	27/40
Biswas [7]	40/40	38/40	33/40	20/40
Lin [34]	40/40	38/40	36/40	33/40
SVS HOG_f	**40/40**	**38/40**	**35/40**	**31/40**

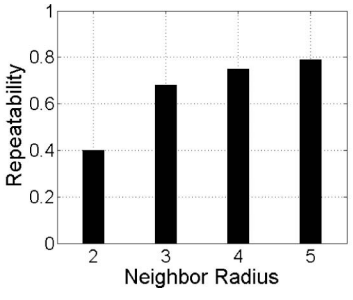


Fig. 13. Repeatability scores for feature selection using HOG_f-entropy. Two feature locations are similar if they are within the radius of five times the scanner resolution.

Fig. 14. Correspondences between two views of the Buddha range scan images.

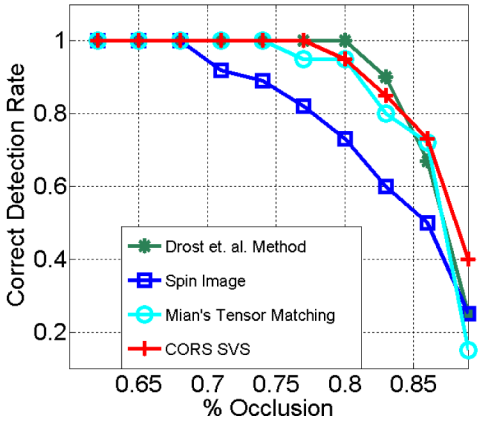


Fig. 15. Comparison of the detection rates versus the percentage of occlusions: for the spin image [27], tensor matching [42], Drost method [16], and SVS-CORS.

the ground truth are smaller than 1/10 of the object's diameter and 12 degrees, respectively. These criteria are the same as that of Drost et al. [16], therefore allowing the comparison to their methods.

Similarly to the first experiment, we use one-class SVM to learn SVSs for 3D point clouds using (x, y, z) coordinates of points. We adopt a bag-of-feature approach. In this framework, CORS descriptors are computed from SVSs at keypoint locations. Approximate k-NN is used to provide tentative correspondences between 2.5D scenes and the 3D objects. We use RANSAC to estimate the transformation matrices. The full 3D shapes are then transformed into the scene's coordinates.

The algorithm acknowledges the presences of the objects within the scenes if the number of overlapping points between the scenes and the 3D shapes are more than a threshold, where two points are said to overlap if their distance is smaller than three times the scanner resolution. The threshold is set to be 15 percent of the number of points in the 3D shapes. A lower threshold triggers too many false alarms and a higher threshold does not allow the detection of many objects in the database with more than 80 percent occlusion. Fig. 15 shows our detection performance in comparison with the state-of-the-art. SVS seems more robust for higher occlusion. Fig. 16 shows sample recognition and registration results for the Chef shape, Parasaurolophus shape, and T-rex shape.

7 Proof of SVM-RBF Irreducibility

The decision function of SVM with RBF kernel has the following form:

$$f(\mathbf{x}) = \sum_{i=1}^{m} \alpha_i \exp\big(-\gamma\|\mathbf{x} - \mathbf{x}_i^*\|^2\big). \tag{17}$$

Fourier transform of the function is as follows:

$$\begin{aligned}\mathcal{F}(\omega) &= \sum_{i=1}^{m} \alpha_i \mathcal{F}\{\exp(-\gamma\|\mathbf{x} - \mathbf{x}_i^*\|^2)\} \\ &= \sum_{i=1}^{m} \alpha_i \mathcal{F}\{\exp(-\gamma\|\mathbf{x}\|^2)\}\exp(-j\omega\mathbf{x}_i^*) \\ &= \sum_{i=1}^{m} \alpha_i \left[\sqrt{\frac{\pi}{\gamma}}\right]^n \exp\left(-\frac{\pi^2}{\gamma}\|\omega\|^2\right)\exp(-j\omega\mathbf{x}_i^*) \\ &= \Phi(\omega)\Psi(\omega),\end{aligned} \tag{18}$$

where n is the dimension of $\mathbf{x}$ and

$$\Phi(\omega) = \left[\sqrt{\frac{\pi}{\gamma}}\right]^n \exp\left(-\frac{\pi^2}{\gamma}\|\omega\|^2\right) \tag{19}$$

$$\Psi(\omega) = \sum_{i=1}^{m} \alpha_i \exp(-j\omega\mathbf{x}_i^*), \tag{20}$$

where n is the dimension of $\mathbf{x}$. The Fourier transform of RBF kernel obviously does not have any zero.

Fig. 16. Detection and registration results of the Chef, Parasaurolophus, and T-rex shapes. Yellow color represents partial scenes and green color represents fully reconstructed 3D models. The average registration error is 0.86, 0.62, and 0.77 for the three scenes, respectively.

From now on, consider $\Phi(\omega)$ as the Fourier transform of an arbitrary kernel function. By assumption, the entire function $\Phi(\omega)$ never vanishes, i.e., it does not have any zero, just like the case of the RBF kernel. Therefore, it remains to show that the function $\Psi(\omega)$ is not factorable to conclude that $f(x)$ is irreducible.

Assume that $\Psi(\omega)$ is reducible. It means that it can be factored into a product of two entire functions. These functions are required to have nonempty zeros set (either real or complex). Then, they must have the following form:

$$\sum_{k=1}^{K} a_k \exp(-j\omega y_k) \sum_{l=1}^{L} b_l \exp(-j\omega z_l) = \sum_{k=1}^{K} \sum_{l=1}^{L} a_k b_l \exp\{-j\omega(y_k + z_l)\}. \tag{21}$$

Note that an exponential function $e^{j\omega\tau}$ never vanishes. This property, together with nonempty zeros set constraint, implicitly requires

$$K \geq 2, \quad L \geq 2. \tag{22}$$

The number of constrains when equating (21) to (20) is $(KL + m)$. The breakdown of constraints is as follows:

- KL constrains to equate the set of exponents $\{y_k + z_l\}$ with the set $\{x_i^*\}$.
- m constrains to equate $\{a_k b_l\}$ with $\{\alpha_i\}$.

The total number of variables is $2(K + L)$, and we require this to be at least equal to the total number of constrains, which gives

$$2(K + L) \geq KL + m. \tag{23}$$

It is easy to verify that this condition does not hold because $K \geq 2$, $L \geq 2$, $m \geq 5$, due to (22) and our initial assumption. Therefore $f(x)$ in (3) is irreducible.

8 Conclusion

We introduced a novel shape representation and explained its application in 2D and 3D. To our observations, this representation is very robust against noise, data acquisition problems, articulation, affine motion, etc. As future work, we plan to investigate alternative descriptors especially tuned for 3D applications.

References

[1] S. Abbasi, F. Mokhtarian, and J. Kittler, "Curvature Scale Space Image in Shape Similarity Retrieval," *ACM Multimedia System,* vol. 7, pp. 467-476, 1999.

[2] H. Van Nguyen and F. Porikli, "Concentric Ring Signature (CORS) for 3D Object Detection Recognition, and Registration," *Elsevier Pattern Recognition,* 2011.

[3] R. Basri, L. Costa, D. Geiger, and D. Jacobs, "Determining the Similarity of Deformable Shapes," *Elsevier J. Vision Research,* vol. 38, pp. 2365-2385, 1998.

[4] S. Belongie, J. Malik, and J. Puzicha, "Shape Matching Object Recognition Using Shape Context," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 24, no. 4, pp. 509-522, Apr. 2002.

[5] A. Berg, T. Berg, and J. Malik, "Multi-Scale Object Detection by Clustering Lines," *Proc. IEEE Int'l Conf. Computer Vision,* 2005.

[6] I. Biederman, "Human Image Understanding: Recent Research and a Theory," *Computer Vision, Graphics, and Image Processing,* vol. 32, pp. 29-73, 1985.

[7] S. Biswas, G. Aggarwal, and R. Chellappa, "An Efficient and Robust Algorithm for Shape Indexing and Retrieval," *IEEE Trans. Multimedia,* vol. 12, no. 5, pp. 372-385, Aug. 2010.

[8] F. Bookstein, "Principal Warps: Thin-Plate Splines and the Decomposition of Deformations," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 11, no. 6, pp. 567-585, June 1989.

[9] H. Chui and A. Rangarajan, "A New Point Matching Algorithm for Non-Rigid Registration," *Computer Vision and Image Understanding,* vol. 89, nos. 2/3, pp. 114-141, Feb. 2003.

[10] H. Chui, A. Rangarajan, J. Zhang, and C.M. Leonard, "Unsupervised Learning of an Atlas from Unlabeled Point-Sets," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 26, no. 2, pp. 160-172, Feb. 2004.

[11] D. Cohen-Steiner, P. Alliez, and M. Desbrun, "Variational Shape Approximation," *ACM Trans. Graphics,* vol. 23, no. 3, pp. 905-914, Aug. 2004.

[12] M. Cossalter, R. Yan, and L. Zheng, "Adaptive Kernel Approximation for Large-Scale Non-Linear SVM Prediction," *Proc. Int'l Conf. Machine Learning,* 2011.

[13] D. Cremers, N.A. Sochen, and C. Schnorr, "A Multiphase Dynamic Labeling Model for Variational Recognition-Driven Image Segmentation," *Int'l J. Computer Vision,* vol. 66, no. 1, pp. 67-81, Jan. 2006.

[14] B. Curless and M. Levoy, "A Volumetric Method for Building Complex Models from Range Images," *Proc. ACM Siggraph,* pp. 303-312, 1996.

[15] S.R. Curtis and A.V. Oppenheim, "Reconstruction of Multidimensional Signals from Zero Crossings," *J. Optical Soc. Am. A,* vol. 4, no. 1, pp. 221-231, 1987.

[16] B. Drost, M. Ulrich, N. Navab, and S. Ilic, "Model Globally, Match Locally: Efficient and Robust 3D Object Recognition," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 998-1005, 2010.

[17] P. Felzenszwalb and J. Schwartz, "Hierarchical Matching of Deformable Shapes," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2007.

[18] B. Flach and D. Schlesinger, "Modelling Composite Shapes by Gibbs Random Fields," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 2177-2182, June 2011.

[19] A. Frome, D. Huber, R. Kolluri, T. Bulow, and J. Malik, "Recognizing Objects in Range Data Using Regional Point Descriptors," *Proc. European Conf. Computer Vision,* 2004.

[20] R. Gopalan, P. Turaga, and R. Chellappa, "Articulation-Invariant Representation of Non-Planar Shapes," *Proc. European Conf. Computer Vision,* 2010.

[21] L. Gorelick, M. Galun, E. Sharon, R. Basri, and A. Brandt, "Shape Representation and Classification Using the Poisson Equation," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2004.

[22] L. Gorelick, M. Galun, E. Sharon, R. Basri, and A. Brandt, "Shape Representation and Classification Using the Poisson Equation," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 28, no. 12, pp. 1991-2005, Dec. 2006.

[23] B. Gunsel and M. Tekalp, "Shape Similarity Matching for Query by Example," *Elsevier J. Pattern Recognition,* vol. 31, no. 7, pp. 931-944, 1998.

[24] G. Heitz, G. Elidan, B. Packer, and D. Koller, "Shape-Based Object Localization for Classification," *Int'l J. Computer Vision,* vol. 84, no. 1, pp. 40-62, 2009.

[25] J. Hua, Z. Lai, M. Dong, X. Gu, and H. Qin, "Geodesic Distance-Weighted Shape Vector Image Diffusion," *IEEE Trans. Visualization and Computer Graphics,* vol. 14, no. 6, pp. 1643-1650, Nov./Dec. 2008.

[26] T. Joachims, "Making Large-Scale SVM Learning Practical," *Advances in Kernel Methods—Support Vector Learning,* B. Schölkopf, C.J.C. Burges, and A.J. Smola, eds., pp. 169-184, MIT Press, 1999.

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

[28] B.B. Kimia and K. Siddiqi, "Parts of Visual Form: Computational Aspects," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 17, no. 5, pp. 544-544, May 1995.

[29] Z. Kourtzi and N. Kanwisher, "Cortical Regions Involved in Perceiving Object Shape," *J. Neuroscience,* vol. 20, pp. 3310-3318, 2000.

[30] L.J. Latecki, R. Lakamper, and U. Eckhardt, "Shape Descriptors for Non-Rigid Shapes with a Single Closed Contour," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 424-429, 2000.

[31] M.D. Levine and K.N. Wu, "3D Part Segmentation Using Simulated Electrical Charge Distributions," *Proc. Int'l Conf. Pattern Recognition,* pp. 14-18, 1996.

[32] J.-M. Lien and N.M. Amato, "Approximate Convex Decomposition of Polygons," *Computational Geometry: Theory and Applications,* vol. 35, pp. 100-123, 2006.

[33] J.-M. Lien and N.M. Amato, "Approximate Convex Decomposition of Polyhedra and Its Applications," *Computer Aided Geometric Design,* vol. 25, no. 7, pp. 503-522, 2008.

[34] L. Lin, X. Liu, and S.-C. Zhu, "Layered Graph Matching with Composite Cluster Sampling," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 32, no. 8, pp. 1426-1442, Aug. 2010.

[35] H. Ling and D. Jacobs, "Shape Classification Using the Inner-Distance," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 29, no. 2, pp. 286-299, Feb. 2007.

[36] H. Ling and K. Okada, "An Efficient Earth Mover's Distance Algorithm for Robust Histogram Comparison," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 29, no. 5, pp. 840-853, May 2007.

[37] H. Liu, W. Liu, and L.J. Latecki, "Convex Shape Decomposition," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 97-104, June 2010.

[38] M. Liu, B.C. Vemuri, S.-I. Amari, and F. Nielsen, "Total Bregman Divergence and Its Applications to Shape Retrieval," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 3463-3468, 2010.

[39] R. Malladi, J.A. Sethian, and B.C. Vemuri, "Shape Modeling with Front Propagation: A Level Set Approach," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 17, no. 2, pp. 158-175, Feb. 1995.

[40] G. McNeill and S. Vijayakumar, "Hierarchical Procrustes Matching for Shape Retrieval," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 885-894, 2006.

[41] Y. Mei and D. Androutsos, "Robust Affine Invariant Shape Image Retrieval Using the ICA Zernike Moment Shape Descriptor," *Proc. Int'l Conf. Image Processing,* 2009.

[42] A.S. Mian, M. Bennamoun, and R. Owens, "Three-Dimensional Model-Based Object Recognition and Segmentation in Cluttered Scenes," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 28, no. 10, pp. 1584-1601, Oct. 2006.

[43] F. Mokhtarian, S. Abbasi, and J. Kittler, "Efficient and Robust Retrieval by Shape Content through Curvature Scale Space," *Proc. Int'l Workshop Image Databases and MultiMedia Search,* 1996.

[44] S. Murray, B. Olshausen, and D. Woods, "Processing Shape, Motion and Three Dimensional Shape-from-Motion in the Human Cortex," *Cerebral Cortex,* vol. 13, pp. 508-516, 2003.

[45] S. Osher and J. Sethian, "Fronts Propagating with Curvature-Dependent Speed: Algorithms Based on Hamilton-Jacobi Formulations," *J. Computational Physics,* vol. 79, pp. 12-49, 1988.

[46] N. Paragios, "A Level Set Approach for Shape-Driven Segmentation and Tracking of the Left Ventricle," *IEEE Trans. Medical Imaging,* vol. 22, no. 6, pp. 773-776, June 2003.

[47] A.M. Peter, A. Rangarajan, and J. Ho, "Shape l'Ane Rouge: Sliding Wavelets for Indexing and Retrieval," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 1-8, 2008.

[48] V.A. Prisacariu and I. Reid, "Nonlinear Shape Manifolds as Shape Priors in Level Set Segmentation and Tracking," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 2185-2192, 2011.

[49] B. Schölkopf, J.C. Platt, J. Shawe-Taylor, A.J. Smola, and R.C. Williamson, "Estimating the Support of a High-Dimensional Distribution," *Neural Computation,* vol. 13, no. 7, pp. 1443-1471, 2001.

[50] B. Schölkopf, J.C. Platt, J. Shawe-Taylor, A.J. Smola, and R.C. Williamson, "Estimating the Support of a High-Dimensional Distribution," *Neural Computation,* vol. 13, no. 7, pp. 1443-1471, 2001.

[51] T. Sebastian, P. Klein, and B. Kimia, "Recognition of Shapes by Editing Their Shock Graphs," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 26, no. 5, pp. 550-571, May 2004.

[52] T.B. Sebastian, P.N. Klein, and B.B. Kimia, "On Aligning Curves," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 25, no. 1, pp. 116-125, Jan. 2003.

[53] D. Slepian, "On Bandwidth," *Proc. IEEE,* vol. 64, no. 3, pp. 292-300, Mar. 1976.

[54] J. Todd, "Visual Perception of 3D Shape," *TRENDS in Cognitive Sciences,* vol. 8, pp. 115-121, 2004.

[55] I.W. Tsang, J.T. Kwok, and P.-M. Cheung, "Core Vector Machines: Fast SVM Training on Very Large Data Sets," *J. Machine Learning Research,* vol. 6, pp. 363-392, 2005.

[56] Z.W. Tu and A.L. Yuille, "Shape Matching and Recognition: Using Generative Models and Informative Features," *Proc. European Conf. Computer Vision,* vol. 3, pp. 195-209, 2004.

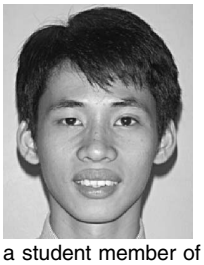
[57] G. Turk and M. Levoy, "Zippered Polygon Meshes from Range Images," *Proc. ACM Siggraph,* pp. 311-318, 1994.

[58] V. Vapnik, *The Nature of Statistical Learning Theory.* Springer Verlag, 1995.

[59] F. Wang, B.C. Vemuri, A. Rangarajan, and S.J. Eisenschenk, "Simultaneous Nonrigid Registration of Multiple Point Sets and Atlas Construction," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 30, no. 11, pp. 2011-2022, Nov. 2008.

[60] X. Yang, S. Koknar-Tezel, and L. Latecki, "Locally Constrained Diffusion Process on Locally Densified Distance Spaces with Applications to Shape Retrieval," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2009.

[61] H. Zhang and J. Malik, "Learning a Discriminative Classifier Using Shape Context Distances," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2003.



Hien Van Nguyen received the BS degree in electrical and computer engineering from the National University of Singapore. He is currently working toward the PhD degree in the Department of Electrical and Computer Engineering, University of Maryland at College Park. He was awarded a full undergraduate scholarship by the Singapore Ministry of Foreign Affairs. His research interests are in computer vision, machine learning, and statistical pattern recognition. He is a student member of the IEEE.

Fatih Porikli received the PhD degree from New York University, New York. He is a Distinguished Research Scientist at Mitsubishi Electric Research Labs (MERL). Before joining MERL in 2000, he developed satellite imaging solutions at Hughes Research Labs and 3D display systems at AT&T Research Labs. His work covers areas including computer vision, machine learning, video surveillance, multimedia processing, structured and manifold-based pattern recognition, biomedical vision, radar signal processing, and online learning, with more than 100 publications and 60 patents. He has mentored more than 40 PhD students and interns. He has received the R&D100 2006 Award in the Scientist of the Year category (select group of winners) in addition to three IEEE Best Paper Awards and five Professional Prizes. He serves as an Associate Editor for *IEEE Signal Processing Magazine*, *SIAM Journal on Imaging Sciences*, *Springer Machine Vision Applications*, *Springer Real-Time Image and Video Processing*, and *EURASIP Journal on Image and Video Processing*. He served as the general chair of the IEEE Advanced Video and Signal-based Surveillance Conference (AVSS) 2010 and has participated on the organizing committee of many IEEE events. He is a senior member of the IEEE and a member of the IEEE Computer Society.