

TEMPLATE-BASED STATISTICAL SHAPE MODELLING ON DEFORMATION SPACE

Girum G. Demisse, Djamila Aouada, Björn Ottersten

Interdisciplinary Center for Security, Reliability and Trust
University of Luxembourg, 4, rue Alphonse Weicker, L-2721, Luxembourg
{girum.demisse, djamila.aouada, bjorn.ottersten}@uni.lu

ABSTRACT

A statistical model for shapes in \mathbb{R}^2 or \mathbb{R}^3 is proposed. Shape modelling is a difficult problem mainly due to the non-linear nature of its space. Our approach considers curves as shape contours, and models their deformations with respect to a deformable template shape. Contours are uniformly sampled into a discrete sequence of points. Hence, the deformation of a shape is formulated as an action of transformation matrices on each of these points. A parametrized stochastic model based on Markov process is proposed to model shape variability in the deformation space. The model's parameters are estimated from a labeled training dataset. Moreover, a similarity metric based on the Mahalanobis distance is proposed. Subsequently, the model has been successfully tested for shape recognition, synthesis, and retrieval.

Index Terms— Statistical shape modelling, curves modelling, deformable template, Markov process.

1. INTRODUCTION

Modelling the inter and intra variation of shapes is a difficult problem mainly due to the non-linear nature of the shape space. Despite the difficulty, several approaches have been developed for representing and modelling shapes [1, 2, 3, 4, 5]. Methods based on feature extraction are the popular choice for shape recognition. A variety of descriptors have been designed to extract features; similarity between shapes is measured in the feature space [6]. Particularly, in [7], a successful feature-based method was introduced leading to a nearest-neighbour like classifier in the feature space. However, the space of all shapes is an infinite dimensional space. Theoretically, there can never be a finite set of features that is unique to a distinct shape [8]. Alternatively, approaches that model the deformation of the shape instead of the shape itself have also been proposed. Earlier in [9], template-based shape modeling was presented as a method for image restoration. However, the deformation of the template shape was not learned from a training dataset, but based on an assumed prior distribution. In [10], the deformation space of human body shapes has been modelled as a direct product of transformation matrices, leading to shape analysis on the Lie algebra

of the deformation space. In this paper, we propose a data driven probabilistic model of curves based on a deformable template; by curve or shape we mean a self-intersection free smooth 1-manifold embedded in \mathbb{R}^2 or \mathbb{R}^3 . Similarly to [9], our approach aims to work on the deformation space rather than the shape space. However, we learn the distribution of the deformations from a labeled training dataset as opposed to assuming a priori. The primary goal of the paper is to introduce a simple, yet reasonably general modelling approach for curved shapes. Indeed, such a model can be used to recognize and synthesize shapes with respect to a template, and measure similarity between shapes of the same class.

The remainder of the paper is organized as follows: Section 2 covers the necessary background. The proposed modelling approach is described in Section 3. In Section 4, its possible applications are detailed. Experimental results are given in Section 5, followed by the conclusion in Section 6.

2. PROBLEM DEFINITION

A given deformable template shape S^* is diffeomorphic to a distinct shape S' if the mapping between the two shapes is smooth with a smooth inverse. The space of diffeomorphisms between S^* and S' is denoted by $\text{Diff}(S^*, S')$. The considered shapes are represented with an equal number n of ordered points $\{p_1, \dots, p_n\}$. Hence, points representing two shapes are assumed to correspond to each other according to their order. Subsequently, the mapping of shape S^* to S' is formulated as a sequence of transformation matrices $\mathbf{x} = \{x_1, \dots, x_n\}$ acting on each point. More accurately, a mapping function \mathbf{x} is defined as $\text{Diff}(S^*, S') \ni \mathbf{x} : S^* \ni p_i^* \rightarrow p_i' \in S'$. If a template shape S^* is deformed to a set of shapes $\mathcal{S} = \{S_1, \dots, S_q\}$, our observation is a set of mapping functions $\mathcal{T} = \{\mathbf{x}_1^*, \dots, \mathbf{x}_q^*\}$, where the superscript denotes the source shape while the subscript denotes the target shape. Moreover, if the shapes in \mathcal{S} are labeled as similar, then learning the *probability density function* (pdf) of our observation \mathbf{x} from the dataset \mathcal{T} should enable us to recognize and synthesize shapes belonging to the same class as \mathcal{S} from the template shape S^* . As a result, we will treat the mapping function \mathbf{x} as a random vector (RV) taking values from $\prod_{i=1}^n \text{SE}(3)_i$ where SE and \prod denote the Special Eu-

clidean group and the direct product, respectively. SE(3) is a Lie group which is also a differentiable manifold, alternatively referred to as M hereafter. A differentiable manifold can further be endowed with a smooth metric g_z at the tangent space $T_z M$ for every $z \in M$. Thus, the realization of a random element is a point on a Riemannian manifold (M, g) . Below we formalize our probability and observation space.

Definition 1. Let (Ω, \mathcal{F}, m) be a probability measure space and $(M_i, \mathcal{B}_i)_{i=\{1, \dots, n\}}$ be measurable spaces where M_i is a Riemannian manifold, and \mathcal{B}_i is Borel-sigma algebra. An RV $\mathbf{x} = [x_1, \dots, x_n]$ is a set of random elements defined on the same probability space, taking values from the set of measurable spaces: $x_i : \Omega \rightarrow M_i$.

A random element¹ x induces a probability distribution P on the observation space (M, \mathcal{B}) ; $P = m(x^{-1}(a))$, $a \in \mathcal{B}$. Alternatively, the induced distribution can be expressed as $P(x \in R) = \int_R p d\omega$, where $p(\cdot)$ is the pdf, and $\omega(\cdot)$ is the Riemannian measure [11]. Thus, given a template shape S^* and a set of shapes \mathcal{S} , labeled as similar, the problem of shape modelling is defined as estimating the pdf of an RV $\mathbf{x} = [x_1, \dots, x_n]$, where each random element $x_{i=\{1, \dots, n\}}$ takes a value from SE(3). Given a shape label l , the distribution of \mathbf{x} is described as follows

$$p(\mathbf{x} | l) = p(x_1, \dots, x_n | l). \quad (1)$$

3. PROPOSED CURVE MODELLING

The conditioned joint distribution in (1) can best be estimated if we consider the spatial relationship between the random elements induced by points ordering [12]. In that regard, the joint distribution can be seen as a Markov process. The distribution of the RV \mathbf{x} is, thus, reduced to the following simpler equation,

$$p(x_1, \dots, x_n | l) = \prod_{i=1}^n p(x_{i+1} | x_i). \quad (2)$$

The Markov chain is formulated as $Q_{i+1} = Q_*^{i+1} \times Q_i$, where $Q_{i+1}, Q_i \in \text{SE}(3)$ are the *realizations* of two consecutive random elements. The state difference Q_*^{i+1} can be expressed, without loss of generality, as $Q_*^{i+1} = Q_{i+1} \times Q_i^{-1}$. In general, every time a template shape S^* is deformed to a distinct shape S' , the realization of the RV \mathbf{x} can be factorized as $\mathbf{x} = [Q_1, Q_*^2 \times Q_1, Q_*^3 \times Q_*^2 \times Q_1, \dots]$. In fact, our actual observation is $[Q_1, Q_*^2, \dots, Q_*^n]$. Intuitively, the transformation matrix Q_*^{i+1} can be seen as the deviation between Q_i and Q_{i+1} . As a consequence, by taking $p(Q_1) = 1$, we have,

$$p(x_1, \dots, x_n | l) = \prod_{i=2}^n p(Q_*^i | Q_{i-1}). \quad (3)$$

¹ Note that random elements take values from SE(3) and not \mathbb{R} .

In essence, the thrust of the proposed probabilistic model is on the deviations between two consecutive random elements. In this work, we have chosen to estimate each term of (3) with a unimodal density function, namely, multivariate Gaussian distribution (MGD). MGD is characterized by two parameters; mean μ_i and covariance matrix Σ_i for $i = 2, \dots, n$.

3.1. Computing the mean in SE(3)

The notion of distance is a central mathematical ingredient in estimating the mean and the covariance of a distribution. However, the observation space SE(3) is non-linear and the usual Euclidean distance does not hold as a distance metric. Alternatively, the distance between two points on a Riemannian manifold is defined as the length of the shortest and straightest curve connecting them, which is called a geodesic distance [13]. Furthermore, if the topology of the manifold is compact, the geodesic can be extended to \mathbb{R} . Although SE(3) is not a compact group, it is a semi-direct product of SO(3)², which is compact, and \mathbb{R}^3 , i.e., it can be decomposed into a rotation and a translation, respectively. In effect $Q_i \in \text{SE}(3)$ can be expressed as follows

$$Q_i = \begin{pmatrix} H_i & v_i \\ 0 & 1 \end{pmatrix}, \text{ s.t., } H_i \in \text{SO}(3), v_i \in \mathbb{R}^3.$$

In [14, 15], a geodesic curve between two rotation matrices is given as follows

$$\gamma(t) = H_1(H_1^{-1}H_2)^t, \quad (4)$$

where $H_1, H_2 \in \text{SO}(3)$ and $t \in [0, 1]$; note that (4) is not always unique. Subsequently, the following equation is given as a natural expression of distance.

$$d_H(H_1, H_2) = \int_0^1 \sqrt{\langle \dot{\gamma}(t), \dot{\gamma}(t) \rangle} dt, \quad (5)$$

where $\dot{\gamma}(t) \in T_{\gamma(t)}M$, and $\langle \cdot, \cdot \rangle$ denotes the inner product which is also the Riemannian metric. Moreover, $\langle \cdot, \cdot \rangle$ is left-translation invariant. The following equation shows this property,

$$\langle H_1, H_2 \rangle_{T_{\gamma(t)}M} = \langle \gamma(t)^{-1}H_1, \gamma(t)^{-1}H_2 \rangle_{T_I M}, \quad (6)$$

where I is used to denote the identity.

As a result, the distance between H_1 and H_2 can be written using the associated Frobenius norm $\| \cdot \|_F$ as follows,

$$d_H(H_1, H_2) = \| \log(H_1^T H_2) \|_F. \quad (7)$$

In [16], the Karcher mean is given as a generalization of the arithmetic mean in a metric space. It is defined as a value that minimizes the variation of a dataset $\{H_1, \dots, H_q\}$; the formulation is described as

$$f_H(\bar{H}) = \frac{1}{q} \sum_{i=1}^q \| \log(H_i^T \bar{H}) \|_F^2. \quad (8)$$

²SO(3) denotes special orthogonal group.

Subsequently, the gradient of (8) is given in [14] as

$$\nabla f_H(\bar{H}) = \sum_{i=1}^q \bar{H} \log(H_i^T \bar{H}). \quad (9)$$

As a result, a geodesic curve $\varphi(\cdot)$ between two matrices $Q_1, Q_2 \in \text{SE}(3)$ can be defined, similar to (4), as follows,

$$\varphi(t) = \begin{pmatrix} H_1(H_1^{-1}H_2)^t & v_1 + (v_2 - v_1)t \\ 0 & 1 \end{pmatrix}. \quad (10)$$

The above curve parametrization is proven to be geodesic in [17]. Thus, the distance between Q_1 and Q_2 is

$$d_Q(Q_1, Q_2) = (\|\log(H_1^T H_2)\|_F^2 + \|v_2 - v_1\|_F^2)^{1/2}. \quad (11)$$

Equivalently to (8), given a dataset $\{Q_1, \dots, Q_q\}$, the Karcher mean $\bar{Q} = \begin{pmatrix} \bar{H} & \bar{v} \\ 0 & 1 \end{pmatrix}$ is defined as a value that minimizes the following equation.

$$f_Q(\bar{Q}) = \frac{1}{q} \sum_{i=1}^q \|\log(H_i^T \bar{H})\|_F^2 + \frac{1}{q} \sum_{i=1}^q \|\bar{v} - v_i\|_F^2. \quad (12)$$

Since \mathbb{R}^3 is a linear space, minimizing the second term of (12) for \bar{v} is equivalent to finding the arithmetic mean of $\{v_i\}_{i=1}^q$. In [18], several optimization approaches for matrix manifolds are given. In our case, Algorithm 1 has proven to be sufficient for computing the solution \bar{H} . The initial guess \bar{H}_0 is computed recursively as in [19]. Hence, the means μ_i of the distribution of the random element x_i (see Definition 1) are estimated from q observations as in the estimation of \bar{Q} with the minimization of (12).

Algorithm 1: Estimation of the Karcher mean in $\text{SO}(3)$

Data: $\{H_1 \dots H_q\} \subset \text{SO}(3)$
Result: $\bar{H} \in \text{SO}(3)$
 initialization: $\bar{H} = \bar{H}_0$, thresh = val₁, step = val₂,
 $\nabla f_H = \sum_{i=1}^q \bar{H} \log(H_i^T \bar{H})$;
while $\|\nabla f_H\|_F > \text{thresh}$ **do**
 $\bar{H} = \bar{H} \exp(-\text{step} \times \nabla f_H)$;
 $\nabla f_H = \sum_{i=1}^q \bar{H} \log(H_i^T \bar{H})$;
end

3.2. Computing the covariance matrix in $\text{SE}(3)$

The covariance matrix Σ_i of the distribution of x_i can be calculated at the tangent space of its mean μ_i . Alternatively, the covariance matrix can be seen as a bilinear form at $T_{\mu_i}M$. If $\{Q_1, \dots, Q_q\}$ are q observations of x_i , then considering the curve $\varphi(\cdot)$ defined in (10), with $\varphi(0) = \mu_i$ and $\varphi(1) = Q_j$, the image $\dot{\varphi}(1)_j$ of Q_j on the tangent space $T_{\mu_i}M$ is,

$$T_{\mu_i}M \ni \dot{\varphi}(1)_j = \begin{pmatrix} \log(\bar{H}^T H_j) & H_j^T (v_j - \bar{v}) \\ 0 & 0 \end{pmatrix}. \quad (13)$$

Matrices in $\text{SE}(3)$ are highly redundant with only 6 degrees of freedom. Consequently, we use principal component analysis (PCA) at $T_{\mu_i}M$ to learn the most varying directions. The principal directions \mathbf{P}_i are used to reduce the dimensionality of the data to a smaller orthonormal basis; they are estimated from $\{\dot{\varphi}(1)_1, \dots, \dot{\varphi}(1)_q\}$. The projection of each vectorized $\dot{\varphi}(1)_j$ on the principal directions is denoted by $\dot{\varphi}(1)_j^r$. Subsequently, the covariance matrix is computed similarly to computing the covariance for a centred dataset in Euclidean space. Using $\mathbb{E}(\cdot)$ as the expectation operator, the estimated covariance matrix of the i^{th} random element from all the q observations $\dot{\varphi}(1)^R = [\dot{\varphi}(1)_1^r, \dots, \dot{\varphi}(1)_q^r]$ is defined as,

$$\Sigma_i = \mathbb{E}[(\dot{\varphi}(1)^R)(\dot{\varphi}(1)^R)^T]. \quad (14)$$

In summary, given a training dataset of RVs $\mathcal{T} = \{\mathbf{x}_1, \dots, \mathbf{x}_q\}$, we estimate the mean μ_i , the principal directions \mathbf{P}_i , and the covariance matrix Σ_i , of every random element x_i from the training dataset \mathcal{T} . Hence, each class model is described by $(\mu_i, \Sigma_i, \mathbf{P}_i)_{i=1, \dots, n}$.

4. SHAPE RECOGNITION, SYNTHESIS AND SIMILARITY

The problem definition (1) describes the likelihood of an RV \mathbf{x} given the label l . Consequently, the Maximum a posteriori (MAP) decision framework can be described as,

$$\hat{l} = \arg \max_{l \in \mathcal{L}} P(l | \mathbf{x}) \approx p(\mathbf{x} | l), \quad (15)$$

where \hat{l} is the estimated label, and \mathcal{L} is the set of all possible labels; in compliance with frequentist statistics we do not assume any priori. Conversely, the likelihood function can be used to synthesize shapes. Algorithm 2 summarizes the shape synthesis procedure; sample(\cdot, \cdot) is a function that samples from a centred MGD.

Algorithm 2: Sample random shape generator

Data: $(\mu_i, \Sigma_i, \mathbf{P}_i)_{i=1 \dots n}$
Result: $\mathbf{x} = \{x_2 \dots x_n\}$
 initialization: $x_1 = Q_1, i = 2, M = \vec{0}$;
while $i \leq n$ **do**
 $Q_*^i = \mu_i \exp(\mathbf{P}_i^{-1} \times \text{sample}(M, \Sigma_i))$;
 $x_i = Q_*^i \times x_{i-1}$;
end

The proposed probabilistic model can also be used to measure similarity between shapes of the same class, and therefore can be deployed for automatic shape retrieval. To that end, we use the arithmetic mean of the Mahalanobis distance between each pair of matching random elements as a similarity measure. The smaller the distance, the more similar the shapes are. Let S_1 and S_2 be two shapes from the same

	Apple	Octopus	Bell	Bone	Bottle	Brick	Comma	Hat	Car	Flatfish	Chopper	Cellphone	Children	Bullseye Score
Ours	100%	21%	38%	91%	65%	94%	99%	27%	100%	52%	55%	98%	100%	72%
SC[7]	84%	21%	66%	69%	63%	85%	83%	29%	95%	40%	40%	98%	100%	67%
IDSC[20]	100%	63%	99%	46%	96%	90%	32%	46%	87%	25%	84%	98%	100%	74%

Table 1. All shapes are represented with 40 equally spaced points.

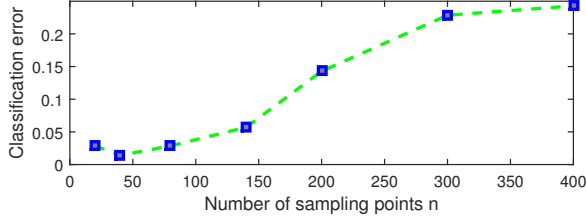


Fig. 1. Classification error versus number of points representing the shape for the fighter jets experiment.

class, and let us consider $r_i^1 = \hat{\varphi}(1)_i^r$, and $r_i^2 = \hat{\varphi}(1)_i^t$ as derived from the realizations of random elements observed from S_1 and S_2 deforming, respectively, see Section 3.2. Then, the proposed similarity metric $h(\cdot, \cdot)$ is given as follows,

$$h(S_1, S_2) = \frac{1}{n} \sum_{i=1}^n ((r_i^1 - r_i^2)^T \Sigma_i^{-1} (r_i^1 - r_i^2))^{1/2}. \quad (16)$$

5. EXPERIMENTAL RESULTS

Classification: The described approach is tested for classification on the *fighter jet* dataset [21]. It contains contours of seven fighter jets each with 30 example images; the inner class variation is achieved by scaling and rotating the images. We estimate the model $(\mu_i, \Sigma_i, \mathbf{P}_i)_{i=1 \dots n}$ for each class with 20 training images and reserve the 10 remaining ones for testing. Each shape is represented with $n = 40$ points. In this particular experiment, the template shape S^* is a straight line represented with 40 ordered points. The number of points, however, is a free parameter to be tuned. Overall, 70 images, 10 from each class, are reserved for testing. Subsequently, the multi-class classification is done using (15). The model’s error is measured as: $\text{error} = 1 - \frac{1}{k} \sum_{i=1}^k \mathbb{1}_l(\hat{l})$, where $\mathbb{1}_l$ is a label indicator function and $k = 70$. The obtained overall error is 0.01428, which is due to misclassifying a single image. However, one must note the relationship between models quality (error rate) and the number of points used to represent the shapes when aiming for high performance, see Fig. 1.

Retrieval: Furthermore, the similarity metric is tested on 13 classes of shapes from MPEG-7 dataset [22]. Again, we train



Fig. 2. Synthetic shapes of "comma" and "stef", both shapes are from MPEG-7 dataset [22]. Each shape is randomly generated with the left most shape being the mean shape.

a model for each of the 13 classes on the first 10 images and compare the left out 10 images with every other image, excluding the training dataset; the comparison is done as described in (16). Subsequently, the number of shapes from the same class among the 10 most similar shapes are counted. Thus, the highest score a class can achieve is 100. Consequently, we compute the bull’s-eye score [23] where the highest possible value is $10 \times 10 \times 13$. Although, our method is a supervised learning skim we compare our result with [7, 20] on the same dataset excluding the training data, see Table. 1. Moreover, it must be noted that we have not done cross-validation due to the small size of the training dataset. Hence, it is probable that the model overfits the dataset. Furthermore, the quality of the preprocessing stage, to satisfy the point correspondence assumption, contributes to the performance significantly. In general, our result can be improved by further investigating the effects of *bias-variance* dilemma. **Synthesis:** Synthetic shape generation is shown in Fig. 2.

6. CONCLUSION

A deformable template-based modelling approach for curves is presented. The central idea is the modelling of deformations with a stochastic process–Markov process is used to simplify the estimation of the distribution. MGD is used to model the deviation between consecutive Markov states. Subsequently, parameters of MGD are estimated in SE(3). We have shown the application of the model for recognition, synthesis, and as a similarity measure. In future work, we will address the dangling free parameter, number of points. Although, subject to deeper investigation, we believe it is task specific and should be estimated as part of the model’s parameter. Additionally, the approach assumes the ordered correspondence of the representing points.

7. REFERENCES

- [1] Michael Kass, Andrew Witkin, and Demetri Terzopoulos, “Snakes: Active contour models,” *International journal of computer vision*, vol. 1, no. 4, pp. 321–331, 1988.
- [2] Alan L Yuille, Peter W Hallinan, and David S Cohen, “Feature extraction from faces using deformable templates,” *International journal of computer vision*, vol. 8, no. 2, pp. 99–111, 1992.
- [3] Benjamin B Kimia, Allen R Tannenbaum, and Steven W Zucker, “Shapes, shocks, and deformations i: the components of two-dimensional shape and the reaction-diffusion space,” *International journal of computer vision*, vol. 15, no. 3, pp. 189–224, 1995.
- [4] David G Kendall, “Shape manifolds, procrustean metrics, and complex projective spaces,” *Bulletin of the London Mathematical Society*, vol. 16, no. 2, pp. 81–121, 1984.
- [5] Fred L Bookstein, *Morphometric tools for landmark data: geometry and biology*, Cambridge University Press, 1997.
- [6] Mingqiang Yang, Kidiyo Kpalma, Joseph Ronsin, et al., “A survey of shape feature extraction techniques,” *Pattern recognition*, pp. 43–90, 2008.
- [7] Serge Belongie, Jitendra Malik, and Jan Puzicha, “Shape matching and object recognition using shape contexts,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, no. 4, pp. 509–522, 2002.
- [8] Eitan Sharon and David Mumford, “2d-shape analysis using conformal mapping,” *International Journal of Computer Vision*, vol. 70, no. 1, pp. 55–75, 2006.
- [9] Yali Amit, Ulf Grenander, and Mauro Piccioni, “Structural image restoration through deformable templates,” *Journal of the American Statistical Association*, vol. 86, no. 414, pp. 376–387, 1991.
- [10] Oren Freifeld and Michael J Black, “Lie bodies: A manifold representation of 3d human shape,” in *Computer Vision–ECCV 2012*, pp. 1–14. Springer, 2012.
- [11] Christopher G Small, *The Statistical Theory of Shape, Springer Series in Statistics*, New York: Springer-Verlag, 1996.
- [12] Julian Besag, “Statistical analysis of non-lattice data,” *The statistician*, pp. 179–195, 1975.
- [13] John M Lee, *Riemannian manifolds: an introduction to curvature*, vol. 176, Springer, 1997.
- [14] Maher Moakher, “Means and averaging in the group of rotations,” *SIAM journal on matrix analysis and applications*, vol. 24, no. 1, pp. 1–16, 2002.
- [15] Rajendra Bhatia, *Positive definite matrices*, Princeton University Press, 2009.
- [16] Hermann Karcher, “Riemannian center of mass and mollifier smoothing,” *Communications on pure and applied mathematics*, vol. 30, no. 5, pp. 509–541, 1977.
- [17] Milos Zefran, Vijay Kumar, and Christopher B Croke, “On the generation of smooth three-dimensional rigid body motions,” *Robotics and Automation, IEEE Transactions on*, vol. 14, no. 4, pp. 576–589, 1998.
- [18] P-A Absil, Robert Mahony, and Rodolphe Sepulchre, *Optimization algorithms on matrix manifolds*, Princeton University Press, 2009.
- [19] Jeffrey Ho, Guang Cheng, Hesamoddin Salehian, and Baba Vemuri, “Recursive karcher expectation estimators and geometric law of large numbers,” in *Proceedings of the Sixteenth International Conference on Artificial Intelligence and Statistics*, 2013, pp. 325–332.
- [20] Haibin Ling and David W Jacobs, “Shape classification using the inner-distance,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 29, no. 2, pp. 286–299, 2007.
- [21] Ninad Thakoor, Jean Gao, and Sungying Jung, “Hidden markov model-based weighted likelihood discriminant for 2-d shape classification,” *Image Processing, IEEE Transactions on*, vol. 16, no. 11, pp. 2707–2719, 2007.
- [22] Longin Jan Latecki, Rolf Lakamper, and T Eckhardt, “Shape descriptors for non-rigid shapes with a single closed contour,” in *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*. IEEE, 2000, vol. 1, pp. 424–429.
- [23] Xiang Bai, Xingwei Yang, Longin Jan Latecki, Wenyu Liu, and Zhuowen Tu, “Learning context-sensitive shape similarity by graph transduction,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 32, no. 5, pp. 861–874, 2010.