Learning Shape Prior Models for Object Matching

Tingting Jiang, Frederic Jurie and Cordelia Schmid
LEAR team, INRIA, Grenoble, France
{tingting.jiang, frederic.jurie, cordelia.schmid}@inrialpes.fr

Abstract

The aim of this work is to learn a shape prior model for an object class and to improve shape matching with the learned shape prior. Given images of example instances, we can learn a mean shape of the object class as well as the variations of non-affine and affine transformations separately based on the thin plate spline (TPS) parameterization. Unlike previous methods, for learning, we represent shapes by vector fields instead of features which makes our learning approach general. During shape matching, we inject the shape prior knowledge and make the matching result consistent with the training examples. This is achieved by an extension of the TPS-RPM algorithm which finds a closed form solution for the TPS transformation coherent with the learned transformations. We test our approach by using it to learn shape prior models for all the five object classes in the ETHZ Shape Classes. The results show that the learning accuracy is better than previous work and the learned shape prior models are helpful for object matching in real applications such as object classification.

1. Introduction

Many object categories can be accurately represented by their shapes. Shape is a very powerful description of object appearance for detection methods [16, 21] with high precision. In this work, we are interested in learning a class-specific shape model from a collection of real images because it (a) makes the model more adapted to real images and (b) can be applied on a large number of categories without requiring a human definition of shapes.

Many recent papers propose methods for learning a model made of edge features [13, 15, 20]. However, edge based models and shapes are not the same because a simple collection of edge features can only provide a local perspective of the shape while the global perspective is missing. For this reason, the arrangement of the edge features, such as the pairwise interactions between edge features [13] or the relative positions of edge features with respect to the centroid of the shape [15], is exploited to improve the edge based models. An edge based model can be learned from real images and refined to obtain a shape [9]. One advantage of our learning approach over [9, 15, 16, 21] is that we propose to learn directly the shape from images, so the constraints or flexibility given by the model can be used during learning, avoiding the complex strategies of [20] or [13].

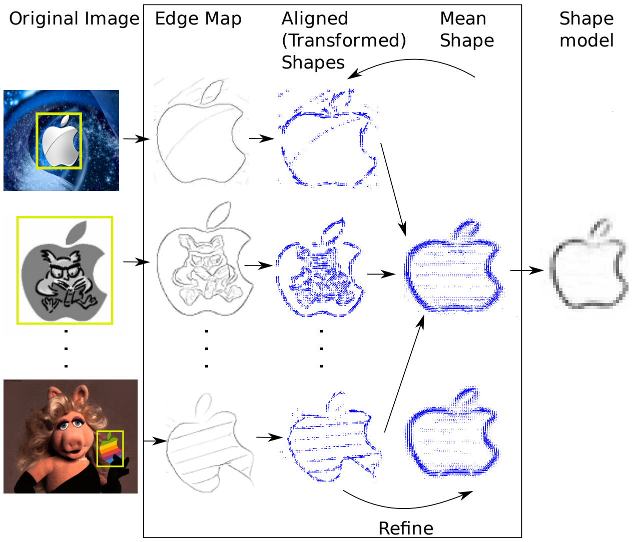

Figure 1. Illustration of the learning process. The first column shows the original training images. The objects are annotated with yellow bounding boxes. The second column shows the edge maps inside the bounding boxes generated by [14]. The third column displays the aligned shapes represented by vector fields after similarity (or non-similarity) transformations. The fourth column shows the inferred mean shape. There are two iterative processes between columns 3 and 4. One is alignment by similarity transformations and the other is the refinement by non-similarity transformations. The last column shows the final mean shape.

As proposed by [3], one reasonable approach to learn the shape model from examples is to first compute the "mean shape" (first order statistics). The mean shape is formulated as the optimal solution to minimize a cost function. This cost function can be the dissimilarity measure between the mean shape and all training shapes. Different cost functions

978-1-4244-3991-1/09/$25.00 ©2009 IEEE

lead to different optimization frameworks. However, it is difficult to learn shape models from real images because of (a) scene clutter and (b) intra-class variations of the shape. The combination of these two issues makes the complexity combinatorial. Due to the large number of possible shapes and images, it is not tractable to try each possible hypothesis to find the optimal solution. For this reason, almost all the methods for learning shape models have been experimented on clean training data (images without clutter) [7, 17, 18]. Contrary to these approaches, the method we propose is very robust to clutter.

Therefore, to find the mean shape, there are two important questions to answer: (1) how to design the cost function and (2) how to find tractable optimization schemes when the mean shape is formulated as the solution to minimize the designed cost function. The answer to the first question clearly depends on the representation of the training images. Using edge points [2, 3, 4, 5, 7, 17, 18] is often the preferred choice but points alone are not very informative (the optimization can be slow and have many local minima). Using more complex features is possible (for example, PAS [9], fragments [15] or shape context [1]), but they are vulnerable to edge clutter and are often object specific. Furthermore, according to [21], complex features often only postpone the complexity problem. To deal with clutter in the training images, we use orientation plus edge points because this combination is local (clutter has almost no effect), generic and the orientation is helpful to remove clutter. To our knowledge, oriented edge points have not been used in the shape learning and matching literature despite their simplicity and the robustness. "Edgelet features" [19] are similar to what we propose, but they are only defined for short lines and segments. One of our contributions is to reformulate the definition given by [3] in order to include a shape distance using edge orientation. As for the second question, instead of looking for the global minimum of the cost function, we compute a local minimum with an extension of the well-known TPS-RPM method [4] which allows to find correspondence and transformations between shapes represented by point sets. As it is used in [1,4,5,7], it alternates between a phase where correspondences are produced, assuming the transformation is known, and the other phase where the transformation is computed assuming the correspondences are known. In contrast with all the previous approaches, we propose a closed form solution of the alignment problem. When the initialization is approximately correct, our approach generally converges to an acceptable solution even if the number of outliers is large.

For object classes with large intra-class variance, learning only a mean shape is not enough because it is also important to learn possible deformations from the mean shape (second order statistics). The advantage of using multiple training images to build the model is that it can handle more clutter and learn priors on shape deformations. Two shapes are similar if one can be transformed into the other by rotation and translation as well as non-rigid transformation with respect to priors (otherwise all shapes are similar). Various shape priors [1,2,5,6,7,12,17,18] have been proposed in recent years in different contexts. Some of them consider the amount of transformation allowed or "smoothness" of non-rigid transformations [1, 18] while other methods apply statistical analysis on training shapes based on a specific shape representation [2, 7, 17]. The most widely used method is probably the Active Shape Model (ASM) [6] where the shape prior is learned by the Principal Component Analysis (PCA) on the training shapes represented by point sets and the shape matching in a test image is iteratively refined with respect to the learned transformations. In the same spirit, we propose a novel shape prior which can treat non-affine transformations and affine transformations independently based on the TPS parameterization [8]. In contrast with all these previous methods we compute the deformation priors not on point sets but directly in the transformation space. It has the advantage to decouple affine and non-affine priors. Moreover, this new shape prior model enables us to extend the TPS-RPM method and to find a closed form solution for the TPS transformation during shape matching.

We demonstrate that our learned shape models can improve recognition results on the realistic ETHZ dataset for both object classification and boundary localization, using training data annotated with bounding boxes.

The rest of the paper is organized as follows. Section 2 gives a short review of the TPS-RPM method since it will be used later. In section 3 we detail the learning process of the shape prior model which will be applied to shape matching in section 4. The experimental results are presented in section 5 and final conclusions are given in section 6.

2. TPS-RPM

Given two point sets $Z = \{z_k\}_{k=1}^{K}$ and $X = \{x_l\}_{l=1}^{L}$ in 2-D, the TPS-RPM by Chui and Rangarajan [4] matches Z and X by a nonrigid TPS transformation f represented by $\{w, d\}$ where w represents the $K \times 3$ nonrigid transformation matrix and d denotes the 3×3 affine transformation matrix. Using homogeneous coordinates where $z_k = (1, z_{kx}, z_{ky})$, the transformation is

$$f(z_k, d, w) = z_k \cdot d + \phi(z_k) \cdot w \tag{1}$$

where $\phi(z_k)$ is a $1 \times K$ vector representing the TPS kernel. The matching algorithm alternates between updating the correspondence $M = \{m_{kl}\}$ for each pair of z_k and x_l and updating the transformation f. In each iteration, after the correspondence M is updated, the point within X corresponding to z_k is estimated as

$$y_k = \Sigma_{l=1}^{L} m_{kl} x_l. \tag{2}$$

Then a mapping function $f(z_k)$ is fitted between $\{y_k\}$ and $\{z_k\}$ by minimizing the following objective function

$$E_{TPS}(f) = \Sigma_{k=1}^{K}||y_k - f(z_k)||^2 + \lambda \int\int [(\frac{\partial^2 f}{\partial x^2})^2 + .. 2(\frac{\partial^2 f}{\partial x \partial y})^2 + (\frac{\partial^2 f}{\partial y^2})^2]\mathrm{d}x\mathrm{d}y. \quad (3)$$

We substitue Eqn. (1) into Eqn. (3) and obtain

$$E_{TPS}(f) = ||Y - Z_M d + \Phi w||^2 + \lambda \cdot trace(w^T \Phi w) \quad (4)$$

where Y and Z_M are concatenated versions of the point coordinates y_k and z_k, and Φ is a $K \times K$ matrix formed by $\phi(z_k)$. The second term is a constraint on the transformation based on smoothness.

To find the best least-squares solutions for the pair $\{w, d\}$, a QR decomposition is used to separate the affine and non-affine transformations,

$$Z_M = [Q_1 Q_2] \begin{pmatrix} R & 0 \\ 0 & 0 \end{pmatrix} \quad (5)$$

where Q_1 and Q_2 are orthonormal matrices. R is upper triangular. Setting $w = Q_2\gamma$ can separate the warping into affine and non-affine subspaces. The separation is very important and will be used to construct our shape prior model in section 3.5. With the QR decomposition, Eqn. (4) can be rewritten as

$$E_{TPS}(\gamma, d) = ||Q_2^T Y - Q_2^T \Phi Q_2 \gamma||^2 + \lambda \gamma^T Q_2^T \Phi Q_2 \gamma + ||Q_1^T Y - Rd - Q_1^T \Phi Q_2 \gamma||^2. \quad (6)$$

The final solution for w and d are given as

$$\hat{w} = Q_2(Q_2^T \Phi Q_2 + \lambda I_{(K-3)})^{-1} Q_2^T Y,$$
$$\hat{d} = R^{-1}(Q_1^T Z - \Phi \hat{w})$$

where I denotes the identity matrix.

3. Learning a shape prior model

3.1. Shape representation and mean shape

Our shape representation is based on a vector field $\vec{V}(x, y)$. As stated in the introduction, the advantage of using vectors instead of points to represent shape is that the orientation information can make the representation more robust to clutter. An image is first preprocessed by the Berkeley edge detector [14]. Then the shape in the image is represented by an oriented edge map $(\psi(x, y), s(x, y))$ where $\psi(x, y) \in [0, \pi)$ denotes the orientation of the point (x, y) and $s(x, y) \in [0, 1]$ denotes the edge strength. If point (x, y) is an edge point, its orientation $\psi(x, y)$ is decided by the orientation of the edge if there is. If it is not an edge point, the orientation is decided by that of its closest oriented edge point. If it is a singular edge point without any neighboring edge points, the orientation is resolved by neighboring non-edge points whose orientations have been decided by other oriented edge points.

For ease of statistical analysis on circular data, the orientation $\psi(x, y)$ is scaled by two to fit the range $[0, 2\pi)$ [11] and the shape representation for an image can be written as a vector field $\vec{V}(x, y) = \{V_X(x, y), V_Y(x, y)\}$ where

$$V_X(x, y) = s(x, y) \cdot \cos(2\psi(x, y)), \quad (7)$$
$$V_Y(x, y) = s(x, y) \cdot \sin(2\psi(x, y)). \quad (8)$$

The distance between two shapes is defined as the distance between their vector fields $\vec{V}_1$ and $\vec{V}_2$, i.e.,

$$D(\vec{V}_1, \vec{V}_2) = \int_x \int_y ||\vec{V}_1(x, y) - \vec{V}_2(x, y)||^2 \mathrm{d}x\,\mathrm{d}y. \quad (9)$$

Given a set of training images with bounding boxes which contain instances of the object. Let $\vec{V}_i$ denotes the oriented edge map within the ith bounding box ($i = 1, \ldots, N$). We define their mean shape $\vec{V}_0$ as the shape which minimizes the following energy function

$$E_1 = \Sigma_{i=1}^{N}[(D(\vec{V}_0, T_i(\vec{V}_i)) + g(T_i)] \quad (10)$$

where each T_i is a transformation of an image, which can be nonrigid. $T_i(\vec{V}_i)$ represents the vector field of the transformed shape by applying T_i on $\vec{V}_i$. The second term $g(T_i)$ is a constraint on transformation T_i.

The goal of our shape learning process is to find the mean shape as well as the associated transformations $\{T_i\}$. To achieve this, we divide the learning process into two stages. First align the training shapes by similarity transformations only. Second consider the non-similarity transformations with the TPS parameterization.

3.2. Shape alignment by similarity transformations

To find the mean shape and best similarity transformations to minimize the energy function E_1, we use an algorithm which alternates between two steps:

- Given a mean shape $\vec{V}_0$, find the best similarity transformation S_i between each training shape $\vec{V}_i$ and $\vec{V}_0$.
- Given the transformed shapes $\{S_i(\vec{V}_i)\}$, find $\vec{V}_0$.

The first step is searching in a 4-D parameter space because a similarity transformation includes 2-D translation, rotation and scaling. We search the optimal solution by the gradient descent method. The second step can be done by taking the average of the transformed shapes as the mean shape, i.e.,

$$\vec{V}_0(x, y) = \frac{1}{N}\Sigma_{i=1}^{N} S_i(\vec{V}_i(x, y)). \quad (11)$$

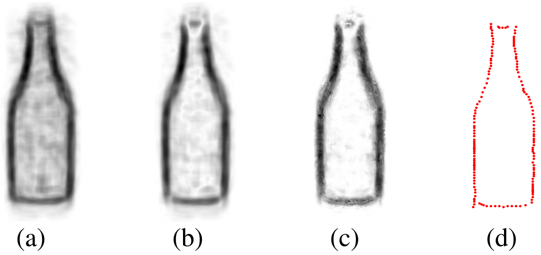

Figure 2. (a)-(c) Different mean shape estimations (a) by edge points without orientation; (b) by the average of vector fields from aligned training shapes; (c) cleaned up by the statistics of the vectors based on (b); (d) Extracted point set from (c).

The above alignment process starts with the mean shape as the average of the initial training shapes $\{\vec{V}_i\}$ and stops when the improvement on E_1 is below a threshold.

Given that the training shapes are annotated with bounding boxes, it is easy to normalize the training bounding boxes with respect to scaling and translation [9]. However, removing rotation differences remains important to align the training shapes. The above alignment process aims to remove all three differences.

3.3. Mean shape generalization

Once the training shapes are aligned by similarity transformations $\{S_i\}$, we can generate an initial mean shape $\vec{V}_0$ based on the statistics of the vectors $S_i(\vec{V}_i(x, y))$ for each point (x, y). The heuristic is that if a point is an edge point in the mean shape, the training shapes are expected to agree on both (a) the edge strength and (b) the edge orientation. The second condition can successfully remove the noisy points because many of them can agree only on edge strength but not on orientation. Specifically, for each point (x, y), we view the corresponding vectors $\{S_i(\vec{V}_i(x, y))\}_{i=1}^N$ as observations of a variable $\vec{V}_0(x, y)$ assumed to be following a 2-D Gaussian model. Then we take the expected mean value as $\vec{V}_0(x, y)$.

After this clean-up process, only edge points with a consensus view from training shapes remain in the cleaned mean shape. Fig. 2 (a)-(c) show different mean shape estimations from edge points without orientations and with orientations (before and after clean-up).

3.4. Shape refinement by TPS transformations

To consider the non-similarity transformations, we extract a set of model points from the mean shape generated in section 3.3 (Fig. 2 (d)) and match the point set back to the aligned training shapes $\{S_i(\vec{V}_i)\}$, using an extension of the TPS-RPM [4]. We extend the original objective function Eqn. (3) by considering the difference between vectors of matched points besides the Euclidean distance and find the TPS transformation between shape model points and edge points in the test image. Similar to the shape alignment, the shape refinement also alternates between two steps:

- Given a mean shape, find a TPS transformation f_i between each aligned training shape $S_i(\vec{V}_i)$ and the extracted point set from the mean shape.
- Given the transformed shapes $f_i(S_i(\vec{V}_i))$, find the mean shape by the method in section 3.3.

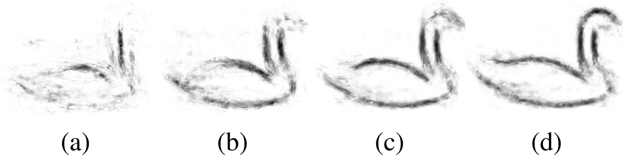

Figure 3. Improved mean shapes by shape refinement over 4 iterations for swans.

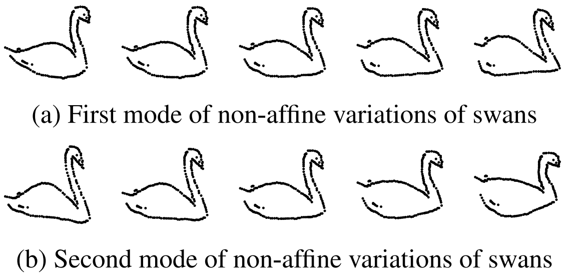
(a) First mode of non-affine variations of swans
(b) Second mode of non-affine variations of swans

Figure 4. Modes of variations for non-affine transformations. The middle column is the mean shape.

The improvement on the mean shape found by the shape refinement is usually significant during the first few iterations (Fig. 3) and then becomes small later. We observe that the number of refinement iterations needed depends on the intra-class variation. Object classes with large intra-class variance require more iterations than those with small variance. In practice, we refine the mean shape 2–5 times.

3.5. Learning a shape prior model

After shape refinement, we can examine the learned TPS transformation parameters $\{w_i, d_i\}$ for each training shape. Set $w_i = Q_2\gamma_i$ and let $\vec{\gamma}_i$ be the vectorization of the last two columns of γ_i. Applying PCA on $\{\vec{\gamma}_i\}$ enables us to capture the variety of non-affine deformations in a low dimensional space ($n_{pc} \leq 15$) where n_{pc} denotes the number of principal components. Each $\vec{\gamma}_i$ can be approximated by

$$\vec{\gamma}_i = \sum_{j=1}^{n_{pc}} \alpha_{i,j}\Gamma_j + \vec{\gamma}_0 \qquad (12)$$

where Γ_j denotes the jth principal component of $\vec{\gamma}$ with coefficient $\alpha_{i,j}$ and $\vec{\gamma}_0$ is the mean of $\{\vec{\gamma}_i\}$. Each non-affine transformation is represented by $\alpha_i = \{\alpha_{i,j}\}$. We can also apply PCA on affine transformation parameters $\{d_i\}$ and have similar results but the number of principal components is smaller (≤ 6) than that of non-affine transformations. In particular, if we ignore translation, there are at most 4 principal components for affine transformation parameters. Specifically, let μ denote the 6-D affine transformation parameter vector, $\mu = d(:, 2 : 3)$. And let θ denote the 4-D affine transformation parameter vector without translation, $\theta = d(2 : 3, 2 : 3)$. Notice that $\theta = H\mu$ where $H = [0\,0\,I_4]_{4\times 6}$. Let β denote the coefficients of principal components of θ which are represented by Θ. Then each θ can be approximated as

$$\theta = \sum_{j=1}^{4} \beta_{i,j}\Theta_j + \theta_0 \tag{13}$$

where θ_0 is the mean of $\{\theta_i\}$. The covariance matrix of α and β are represented by Σ_α and Σ_β respectively. From the above, we construct a shape prior model which includes the mean shape, and variations of affine and non-affine transformations from the mean shape. Fig. 4 shows the first two modes of variations for non-affine transformations within object class "swans".

4. TPS-RPM with shape prior

Based on the shape prior model, each variation of the shape can be represented by a pair of parameters (w, d) and further by (α, μ) based on PCA analysis. For a test image window represented by $\vec{V}$, the probability that there is a shape represented by (α, μ) is given by

$$P(\alpha, \mu|\vec{V}) = \frac{P(\vec{V}|\alpha, \mu)P(\alpha, \mu)}{P(\vec{V})} \tag{14}$$

and the best estimation of α and μ is given by

$$\begin{aligned}
& (\alpha^*, \mu^*) \\
= \;& \underset{\alpha,\mu}{\operatorname{argmax}} \log P(\alpha, \mu|\vec{V}) \\
= \;& \underset{\alpha,\mu}{\operatorname{argmax}} [\log P(\vec{V}|\alpha, \mu) + \log P(\alpha, \mu)] \\
= \;& \underset{\alpha,\mu(\beta)}{\operatorname{argmax}} [\log P(\vec{V}|\alpha, \mu) + \log P(\alpha, \beta)] \\
= \;& \underset{\alpha,\mu(\beta)}{\operatorname{argmax}} [\log P(\vec{V}|\alpha, \mu) + \log P(\alpha) + \log P(\beta)]
\end{aligned}$$

The first term reflects the influence from the data and the other two are regularization terms based on the non-affine prior and affine prior.

Notice that the original TPS-RPM objective function Eqn. (3) (also the variation in [18]) has a regularization term for transformation f based on a "smoothness" constraint. However, with the training shape examples, we can improve the TPS-RPM matching with a more accurate shape prior model based on the PCA on non-affine and affine transformation parameters in section 3.5. Therefore, we design a new energy function for shape matching as follows:

$$E_2 = E_{euc} + \lambda_1 E_{vec} + \lambda_2 E_{na} + \lambda_3 E_a \tag{15}$$

where E_{euc} and E_{vec} denote the Euclidean distance and vector distance between matched points respectively. E_{na} and E_a denote the amount of non-affine transformation and affine transformation (without translation) respectively. They are measured based on the shape prior model. Specifically, following [4], the function can be written as

$$\begin{aligned}
E_2 \;=\;& ||Ay - B\vec{\gamma}||^2 + ||Q_1^T Y - Rd - Q_1^T \Phi Q_2 \gamma||^2 \,.. \\
& + \lambda_1 E_{vec} + \lambda_2 \alpha^T \Sigma_\alpha^{-1} \alpha + \lambda_3 \beta^T \Sigma_\beta^{-1} \beta
\end{aligned} \tag{16}$$

where
$A = \begin{pmatrix} Q_2^T & 0 \\ 0 & Q_2^T \end{pmatrix}, B = \begin{pmatrix} Q_2^T \Phi Q_2 & 0 \\ 0 & Q_2^T \Phi Q_2 \end{pmatrix}$
and y is a vectorization of the coordinates of matched points in the normalized test image or $y = Y(:, 2 : 3)$.

To get the best α, assume that the items $||Q_1^T Y - Rd - Q_1^T \Phi Q_2 \gamma||^2$ and $\lambda_1 E_{vec}$ are approximately zero if λ_1 and λ_3 are very small. The part of the energy function which is dependent on α is

$$\begin{aligned}
E_2(\alpha) &= ||Ay - B\vec{\gamma}||^2 + \lambda_2 \alpha^T \Sigma_\alpha^{-1} \alpha \\
&= ||Ay - B(\Gamma\alpha + \vec{\gamma}_0)||^2 + \lambda_2 \alpha^T \Sigma_\alpha^{-1} \alpha.
\end{aligned}$$

Let $\frac{\partial E_2(\alpha)}{\partial \alpha} = 0$, and we have

$$\alpha^* = (\Gamma^T B^T B \cdot \Gamma + \lambda_2 \Sigma_\alpha^{-1})^{-1} \Gamma^T B^T (Ay - B\vec{\gamma}_0). \tag{17}$$

With α^* , the part of the energy function dependent on μ is

$$\begin{aligned}
& E_2(\mu) \\
= \;& ||C - Rd||^2 + \lambda_3 \beta^T \Sigma_\beta^{-1} \beta \\
= \;& ||\vec{C} - G\mu||^2 + \lambda_3 \beta^T \Sigma_\beta^{-1} \beta \\
= \;& ||\vec{C} - G\mu||^2 + \lambda_3 (H\mu - \theta_0)^T \Theta \Sigma_\beta^{-1} \Theta^T (H\mu - \theta_0).
\end{aligned}$$

where $G = \begin{pmatrix} R & 0 \\ 0 & R \end{pmatrix}, C = Q_1^T (Y - \Phi Q_2 \gamma^*)$, γ^* is computed from α^* by Eqn. (12) and $\vec{C}_{6\times 1}$ is the vectorization of last two columns of C.

Let $\frac{\partial E_2(\mu)}{\partial \mu} = 0$, and we have

$$\begin{aligned}
\mu^* \;=\;& (G^T G + \lambda_3 H^T \Theta \Sigma_\beta^{-1} \Theta^T H)^{-1} \\
& \cdot (\lambda_3 H^T \Theta \Sigma_\beta^{-1} \Theta^T \theta_0 + G^T \vec{C}).
\end{aligned} \tag{18}$$

From the above, we can see that the learned shape prior model can be incorporated into the TPS-RPM process without explicitly referring to the shape prior model like the constrained TPS-RPM proposed by Ferrari *et al.* [9]. They constrain the shape matcher to search only within a "valid" region of shapes which is determined by the principal components of the matched point sets from training examples. This method can avoid implausible shapes outside of the "valid" region. However, the shapes inside the "valid" region are not guaranteed to be consistent with the training shapes. As shown in Fig. 5, we can see the difference between the three different matching methods. Fig. 5 (a) is the matching result from original TPS-RPM [4] using the model points learned by our approach. Fig. 5 (b) displays the matching result from the constrained shape matching method by [9]. You can see that both output shapes are distracted to the clutter inside the bottle due to the reflection. Although the output shape in Fig. 5 (b) is twisted, it is not far from the model shape (within the "valid" region) and therefore can survive. Fig. 5 (c) shows our matching result which is robust to the clutter. More comparisons on shape matching between [9] and our method are shown in Fig. 8.

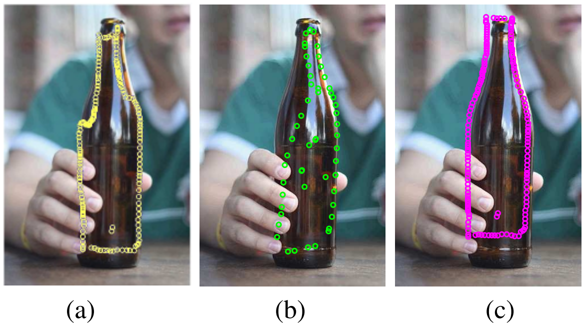

Figure 5. Comparison of different shape matching methods based on the same initialization. (a) Output shape with original TPS-RPM [4] using the mean shape learned by our method. (b) Output shape with the constrained TPS-RPM [9] using their learned shape model. (c) Output shape obtained with our shape matching method using our learned shape prior. The shape models used by [9] and by our method are learned from the same training data.

5. Experiments

We evaluate the proposed shape learning and shape matching methods based on the ETHZ shape classes [9] containing five diverse object classes with 255 images in total. Some classes such as giraffes and swans are very challenging because they have significant intra-class variations. Moreover, some objects are partially occluded or surrounded by background clutter.

5.1. Shape Learning

The experimental setup is same as [9]. For each object class, we learn 5 different shape models by sampling 5 subsets of half of the class images at random. Fig. 6 shows the 25 mean shapes learned from the training data. We evaluate the learning accuracy following the measure proposed in [10]. Let B_t be the ground truth boundary and B_r be the aligned shape output by the shape refinement in section 3.4. *Coverage* is the percentage of points from B_t closer than a threshold t from any point of B_r. *Precision* is the percentage of points from B_r closer than t from any point of B_t. Table 1 shows the comparison between the average learning accuracy of our method over training instances and trials, and the results reported in [10]. Both methods set t as 4% of the diagonal of the bounding box of B_t. Our learned shape models are more accurate in terms of both coverage and precision for the first three classes. As for the other two classes (mugs and swans), our coverage is slightly worse but the precision is better. The reason is that some of our mean shapes learned from these two classes are not complete. For example, the second mean shape for mugs in Fig. 6 leaves out a small piece of the boundary just below the handle because many training images don't contain the edge information there due to shadow.

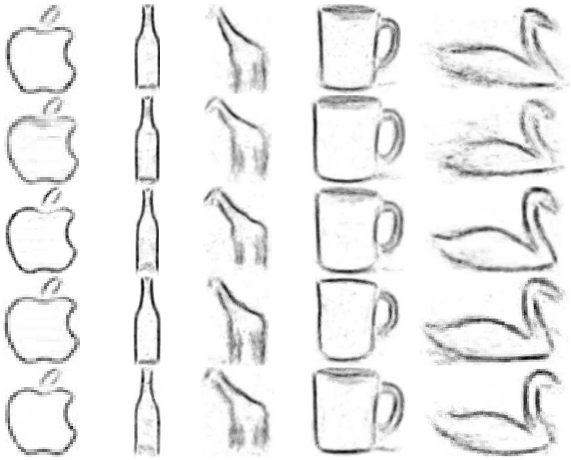

Figure 6. Mean shapes obtained with 5 training sets for each of the 5 object classes in the ETHZ data set.

5.2. Shape Matching

To test the proposed shape matching method with prior described in section 4, we first use the learned shape prior models to localize the object boundaries within ground truth bounding boxes in positive test images (the other images in the same object class excluding the training set) and compare the matching accuracy to the method [9, 10]. Fig 8 shows some example comparisons of the matching results. In general, our shape matching results are more robust to clutter and consistent with training examples. In the matching experiments, we only use the non-affine shape prior because the affine prior knowledge is very limited due to the

	Applelogos	Bottles	Giraffes	Mugs	Swans
Our learning results	**93.0 / 98.1**	**97.6 / 91.4**	**81.0 / 81.8**	92.1 / **93.1**	87.0 / **88.4**
Learning results from [10]	90.2 / 90.6	96.2 / 92.7	70.8 / 74.3	**93.9** / 83.6	**90.0** / 80.0
Our matching results	**95.5 / 98.9**	**89.1 / 90.3**	**77.7 / 79.5**	80.0 / **86.5**	77.8 / **84.2**
Matching results by [10]	92.9 / 95.4	86.8 / 82.1	71.8 / 73.1	**84.4** / 81.4	**82.8** / 76.1

Table 1. Learning and matching accuracy. The first row are our learning results and the second row are the results from [10]. The third row are our matching results and the last row are the results from [10]. Each entry is the average coverage/precision over trials and training/testing instances.

fact that most images in ETHZ shape classes are well orientated (not rotated). Therefore it is too restrictive to apply the learned affine prior (without rotations) to match test examples (with rotations). Instead of using the learned affine prior we constrain the norm of affine matrix as in [4]. Table 1 shows a comparison with [10]. Our matching results are better for the first three classes. For mugs and swans, our coverage is worse but the precision is higher due to the missing parts of the mean shape as explained in section 5.1.

To test whether the shape prior models can help object classification, we generate negative test examples randomly. For each training set which is half of the class images, we choose the other images in the same object class and half of the class images from each of the other four classes as test images (127 test images). For each test image, we crop 10 regions which include the ground truth windows containing the object as well as random samples from the background. For each learned shape model, there are 1270 test examples in total among which about 20 to 40 examples are positive depending on the object class. We apply both our matching method and the constrained shape matching in [9] on all the test examples.

We score each output shape obtained with our method by a weighted sum of five terms: (1) The number of matched model points. (2) The Euclidean distance between mapped model points and their corresponding test image points. (3) The amount of affine transformation, i.e., the norm of the affine matrix d. (4) The amount of non-affine transformation evaluated by the learned non-affine prior, $\alpha^T \Sigma_\alpha^{-1} \alpha$ in Eqn. (16). (5) The vector response between the mean shape $\vec{V}_0$ and the aligned test shape $T_i(\vec{V}_i)$. The idea is similar to Chamfer matching, but instead of the Chamfer distance, we take the sum of dot products of corresponding vectors, i.e.,

$$VR = \int_x \int_y \vec{V}_0(x, y) \cdot T_i(\vec{V}_i(x, y)) \mathrm{d}x \, \mathrm{d}y. \qquad (19)$$

If the orientations of two corresponding vectors are the same or close, their dot product is positive. If their orientations are opposite, it will be negative. The larger the vector response is, the higher score the matching will get. We do not use the distance between vector fields as in Eqn. (9) in order to avoid the influence of noisy edges. Criteria (1)-(3) are also used in the score function [9]. But their measure for non-rigid transformation follows the "smoothness" criterion, i.e., the term $trace(w^T \Phi w)$ in Eqn. (4) instead of the learned non-affine prior that we used. By checking the consistency of orientations, criterion (5) is designed to remove false positive matches which can fool criteria (1)-(4).

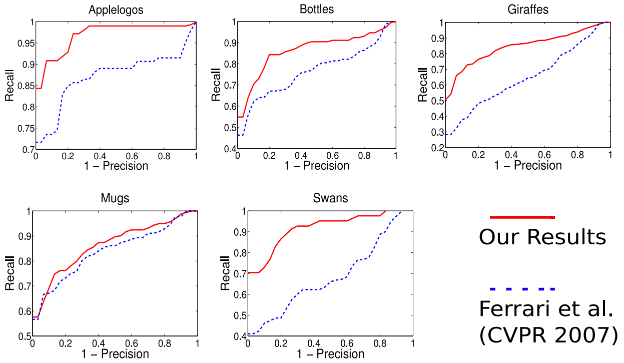

Figure 7. Classification results for 5 object classes.

Fig. 7 evaluates and compares the object classification performance of our approach and [9]. Our approach performs significantly better for all five classes which shows that using the deformation prior learned from training images and considering the orientation consistency in the score function improves the classification accuracy.

6. Conclusion

In this paper, we first presented a novel approach to learning shape prior models from images annotated with bounding boxes. Based on the shape representation of oriented edge points, our learning process is robust to clutter. The shape prior learned by our approach is a prior on shape deformations which can separate the non-affine transformation and affine transformations based on the TPS parameterization. This is very useful to learn the intra-class variability of the shape. Second, we applied the learned shape prior model during shape matching based on TPS-RPM framework and found a closed form solution for TPS transformation. We illustrated our approach on datasets of real images and the experimental results show that our approach can improve both learning accuracy and matching accuracy compared to previous work. The learned shape prior models

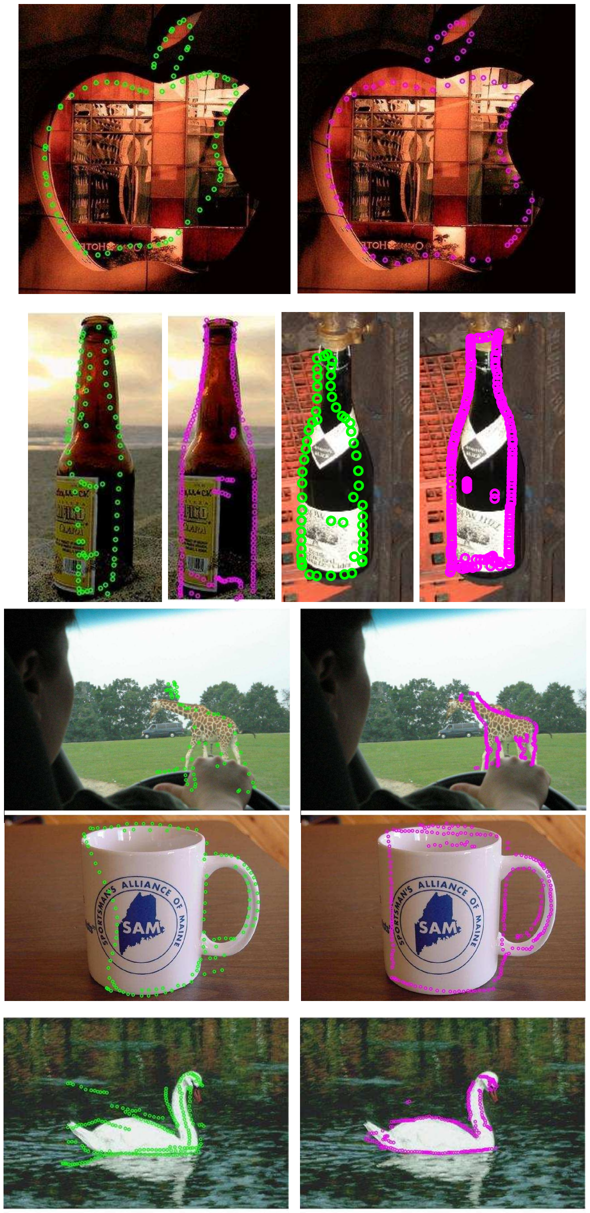

Figure 8. Comparisons between the shape matching results from [9] (green, left) and from our approach (purple, right).

have also been demonstrated to be useful to improve object classification performance.

Acknowledgements Tingting Jiang is funded by a postdoctoral fellowship of INRIA.

References

[1] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *PAMI*, 24(4):509–522, 2002.

[2] A. D. Bue, X. Lladó, and L. de Agapito. Non-rigid face modelling using shape priors. In *AMFG*, volume 3723 of *LNCS*, pages 97–108. Springer, 2005.

[3] G. Charpiat, O. D. Faugeras, and R. Keriven. Shape statistics for image segmentation with prior. In *CVPR*, pages 1–6, 2007.

[4] H. Chui and A. Rangarajan. A new point matching algorithm for non-rigid registration. *CVIU*, 89(2-3):114–141, 2003.

[5] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. *PAMI*, 23(6):681–685, 2001.

[6] T. F. Cootes and C. J. Taylor. Active shape models: Smart snakes. In *BMVC*, pages 267–275, 1992.

[7] D. Cremers and S. Soatto. A pseudo-distance for shape priors in level set segmentation. *2nd IEEE Workshop on Variational, Geometric and Level Set Methods in Computer Vision*, 2003.

[8] J. Duchon. Spline minimizing rotation-invariant semi-norms in sobolev spaces. In *Constructive Theory of Functions of Several Variables*, volume 571 of *Lecture Notes in Mathematics*, pages 85–100, 1977.

[9] V. Ferrari, F. Jurie, and C. Schmid. Accurate object detection with deformable shape models learnt from images. In *CVPR*, pages 1–8, 2007.

[10] V. Ferrari, F. Jurie, and C. Schmid. From images to shape models for object detection. Technical report, INRIA, RR 6600, 2008.

[11] N. I. Fisher. *Statistical Analysis of Circular Data*. Cambridge University Press, Cambridge, England, 1993.

[12] I. Kokkinos and A. L. Yuille. Unsupervised learning of object deformation models. In *ICCV*, pages 1–8, 2007.

[13] M. Leordeanu, M. Hebert, and R. Sukthankar. Beyond local appearance: Category recognition from pairwise interactions of simple features. In *CVPR*, pages 1–8, 2007.

[14] D. R. Martin, C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *PAMI*, 26(5):530–549, 2004.

[15] A. Opelt, A. Pinz, and A. Zisserman. A boundary-fragment-model for object detection. In *ECCV*, pages II: 575–588, 2006.

[16] S. Ravishankar, A. Jain, and A. Mittal. Multi-stage contour based detection of deformable objects. In *ECCV*, pages I: 483–496, 2008.

[17] M. Rousson and N. Paragios. Shape priors for level set representations. In *ECCV*, pages 78–92, 2002.

[18] Z. W. Tu, S. F. Zheng, and A. L. Yuille. Shape matching and registration by data-driven EM. *CVIU*, 109(3):290–304, 2008.

[19] B. Wu and R. Nevatia. Detection and tracking of multiple, partially occluded humans by Bayesian combination of edgelet based part detectors. *IJCV*, 75(2):247–266, 2007.

[20] L. L. Zhu, C. X. Lin, H. Huang, Y. H. Chen, and A. L. Yuille. Unsupervised structure learning: Hierarchical recursive composition, suspicious coincidence and competitive exclusion. In *ECCV*, pages II: 759–773, 2008.

[21] Q. Zhu, L. M. Wang, Y. Wu, and J. B. Shi. Contour context selection for object detection: A set-to-set contour matching approach. In *ECCV*, pages II: 774–787, 2008.