Contour Guided Hierarchical Model for Shape Matching

Yuanqi Su, Yuehu Liu, Bonan Cuan, Nanning Zheng
Xi'an Jiaotong University, Xi'an, Shaanxi Province, China, 710049
yuanqisu,liuyh@mail.xjtu.edu.cn,cuanbonan@gmail.com,nnzheng@mail.xjtu.edu.cn

Abstract

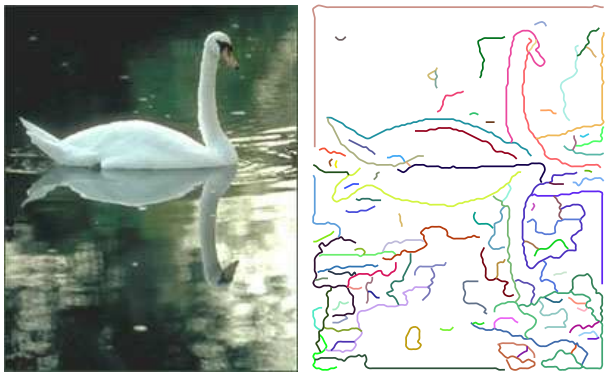
For its simplicity and effectiveness, star model is popular in shape matching. However, it suffers from the loose geometric connections among parts. In the paper, we present a novel algorithm that reconsiders these connections and reduces the global matching to a set of interrelated local matching. For the purpose, we divide the shape template into overlapped parts and model the matching through a part-based layered structure that uses the latent variable to constrain parts' deformation.

As for inference, each part is used for localizing candidates by the partial matching. Thanks to the contour fragments, the partial matching can be solved via modified dynamic programming. The overlapped regions among parts of the template are then explored to make the candidates of parts meet at their shared points. The process is fulfilled via a refined procedure based on iterative dynamic programming. Results on ETHZ shape and Inria Horse datasets demonstrate the benefits of the proposed algorithm.

1. Introduction

Matching a given shape template to an image is the core part in many visual tasks, such as object detection [25, 15, 16, 20], pose estimation [14], etc. Shape matching is challenging especially in cluttered scene due to occlusion, background noise, and the non-rigid deformation of the object. Thus, how to cope with noises and non-rigid deformations is vital for developing successful matching algorithm.

To suppress noises, recent methods [15, 20, 16] have proposed to model the contour fragment instead of the edge points, because the former usually groups edge points of the same object together to assist analysis. On the other side, due to imperfect edge detection results and/or grouping rules, object boundary sometimes breaks into several fragments, such as that shown in Fig.1(b) where the boundary of the swan is scattered in 5 segments, which is difficult to match with a complete shape template. In addition, a fragment may contain edge points of several objects, which further complicates the analysis. Alternatively, [18] proposed to work from the model side by deforming the shape template to fit the image. The limitation of [18] is that it negelects the connections among parts.



(a) (b)

Figure 1. (a). The input image from the ETHZ dataset[7] and (b). the extracted contour fragments.

In this paper, we consider the matching problem from the image side where we attempt to search for the object boundary that resembles a given shape template. In this way, shape matching becomes as how to select a subset of contour fragments to best explain the given template. Since selecting the contour fragments involves a combinatorial optimization[15, 20] process that is usually NP-hard, we propose to locate multiple parts of the object boundary and then assemble them into a closed boundary. As object boundary is usually broken, localizing local parts is obviously easier than detecting the global one. Hence we divide the shape template into parts, and the partial template could help to pick an involved contour fragments for the object. In case the chosen contour fragments cannot guarantee to be the object boundary, we let the adjacent partial templates to share certain regions. These shared regions are shown to be effective in bridging the gap among the chosen fragments, and finally linking them into the boundary.

The rest of the paper is organized as follows. After briefly reviewing related work in Sec.2, we formulate the

shape matching based on the layered deformation model in Sec.3. In the subsequent Sec.4 and 5, the approximate and refined way are derived respectively for inferencing the optimal match. In Sec.6 experimental evaluations and analysis are reported followed by conclusions drawn in Sec.7.

2. Related Work

Existing shape matching literature could be roughly grouped by part description, part structure, and part heuristics. These lines are summarised in the following paragraphs respectively.

For part description, popular choices include the shape context [1], distance measure [25, 14, 18], local shape descriptors based on adjacent segments [8, 7], etc. Among these methods, the Chamfer distance [2] measures the Euclidean distance from each point in the template to its nearest edge point in the image. To handle background noise, Shotton *et al*. [25] considered the tangent orientation, and proposed the oriented chamfer distance that contains both the Euclidean distance and orientation difference. Ming-Yu *et al*. [14] proposed the fast directional chamfer matching (FDCM) algorithm that slides the shape template over the entire image domain. Instead of giving the approximate distance as in [25], [14] could achieve exact solution with the aid of dynamic programming. Since neither CM nor FDCM possesses capability to handle complex deformations, Nguyen [18] introduced a part based chamfer matching that allows each part to deform along its norm direction and used a chain structure to model the connections among parts.

Different from these distance measures, this paper presents a distance measure that enforces the order of points underlying the contour. Since contour is a point sequence, order is an indispensable factor [24] that fits into the representation of contour fragments and suppresses noises. To handle the complex deformation of parts, we further introduce a layered model based distance measure with latent variable to allow deformation of each part within a linear space.

For part structure, the chain structure is a natural choice for describing the connections among parts because the contour is a point sequence. There are many proposed methods [24, 23, 26, 18] that represent the shape by a chain of parts and use dynamic programming for inference. In addition, Coughlan and Huiying [3] viewed each contour point as a part, and used a fully connected network for defining the connections among part. Leordeanu *et al*. [13] adopted the same structure as [3], with inference substituted by the spectral matching. Felzenszwalb[4] and Lu *et al*. [15] proposed a hierarchical structure to represent the detected contour fragments with rules defined on the structure. Besides the chain structure, star model is perhaps the most popular one and is adopted by many recent work [12, 11, 19, 25, 20]. It works by localizing the candidates of parts from the image and letting the candidates to vote for the object poses.

For part heuristics, since star model often suffers from its loose connection among parts, there are some methods that utilize the bottom-up features to guide the search of parts. *E.g*., Ferrari *et al*. [8, 7] explored the connectivity among contour fragments to use their spatial adjacency to make partial matching robust to noises. Praveen and Shi [20] chose another heuristic which requires a contour fragment to either belong to an instance or be completely irrelevant. These strategies effectively suppress noises and improve the matching accuracy.

In the paper, we tackle the problem of shape template matching from cluttered image using a star-like model. Object part in our model is a piece of contour fragments. Different from the classical star model, we further introduce the hard constraints on the adjacent parts to enforce their shared points to meet each other. The next section elaborates our proposed model.

3. The Layered Deformation Model

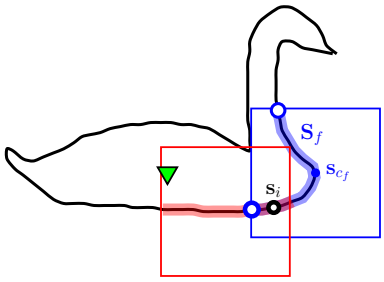


Figure 2. Representation of shape. The blue and red segments are two parts overlapping with each other and the green triangle indicates the object center.

As shown in Fig.2, we represent shape by a set of contours; each contour is a point sequence. For simplicity, our discussion focuses on the shape with a single sequence, and the obtained conclusions can be straightly extended to that with multiple ones. We denote the shape by $\mathbf{S} := \{\mathbf{s}_i | \, i = 1, \cdots, |S|\}$ where $|\cdot|$ gives the number of the points. Throughout the paper, we use an *bold uppercase* letter for the shape, the corresponding *bold lowercase* letter for its point and $|\mathbf{S}|$ for the length.

In the image side, a set of contour fragments are extracted. In our general case for shape matching, the contour fragments $\mathbf{U}$ usually correspond to multiple objects occluding each other. To distinguish the fragments in $\mathbf{U}$, we introduce a vector $\mathbf{b}$ that stores the index range. The range for cth fragment is $[b_c, b_{c+1})$. We give an example of fragments in Fig.1(b), where fragments are discriminated by color.

Considering that boundary of an object is usually broken into several parts as shown in Fig.1(b), we split $\mathbf{S}$ into a set

of parts and let the adjacent parts overlap with each other as shown in Fig.2. Parts of the shape are generated by cropping the $\mathbf{S}$ using a moving square centered on it. Each part itself is a point sequence, denoted by $\mathbf{S}_f$ where the subscript f is the index set. Center of the part is located on $\mathbf{s}_{c_f}$. In practice, we use 0.4 for the ratio of the overlapping region throughout the paper. It means that when two parts overlap, they share 40% points.

3.1. Match based on Two-Stage Deformation

Matching the shape template $\mathbf{S}$ to the image $\mathbf{U}$ involves determining a similarity transform and a one-to-one correspondence. The one-to-one correspondence picks a point from $\mathbf{U}$ for each point in shape template $\mathbf{S}$. The selected points give a shape instance $\mathbf{Y}$. The similarity transform is then used for eliminating the difference in scale, rotation and displacement between the shape template $\mathbf{S}$ and its instance $\mathbf{Y}$ in image. It also corresponds to the pose of shape instance in the image. All the possible similarity transforms construct the parameter space.

In practice, the parameter space is usually discretized by sampling a set of scaling factors, the rotation angles and the displacement. For sake of brevity, we neglect the influence of scale and rotation, and assume that both factors have been optimized by enumerating the possible combinations of them. Then the only factor left is the displacement. Given a displacement vector $\mathbf{o}$, there is a potential instance with its center on $\mathbf{o}$.

Searching for the one-to-one correspondence is interlaced with the optimization of the similarity transform. To model their relations, a layered structure in Fig.3(a) is used. It models the alignment between the boundary $\mathbf{Y}$ and the shape template $\mathbf{S}$ and evaluates their difference.

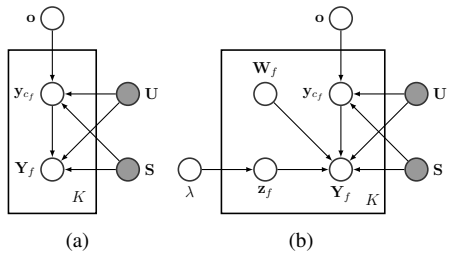


Figure 3. (a). The shape matching model without part variation, (b).The shape matching model with part variation.

The structure in Fig.3(a) gives a part-based way for alignment since registration of a part is much easier than that of the whole shape. On the other hand, it is observed that points belonging to the same part usually share the similar geometric properties. Considering the characteristics, we introduce a two-layered structure to describe a part's deformation. The layered structure allows each part of the boundary to displace with respect to its center $\mathbf{o}$, and shape of each part to deform with respective to the part's center.

In Fig.4, we use a toy example to illustrate the aligning process. The square in Fig.4(a) gives the template; its four corners correspond to the centers of four parts. When registering the square to the shape in Fig.4(b), the proposed model aligns their centers first, then finds the new locations for the four corners, and finally registers the displaced parts of the template. The coarse-to-fine way could effectively handle deformation since it considers the deformation from multiple scales.

According to the structure, we can deduce the posterior for the object center $\mathbf{o}$ and the boundary $\mathbf{Y}$ given the $\mathbf{U}$ and $\mathbf{S}$. It takes the form by,

$$p(\mathbf{Y},\mathbf{o}|\mathbf{U},\mathbf{S}) \propto \tag{1a}$$

$$p(\mathbf{o})\prod_f p(\mathbf{y}_{c_f}|\mathbf{o},\mathbf{U},\mathbf{S})p(\mathbf{Y}_f|\mathbf{y}_{c_f},\mathbf{U},\mathbf{S}) \propto \tag{1b}$$

$$\prod_f p(\mathbf{y}_{c_f}|\mathbf{o},\mathbf{U},\mathbf{S})p(\mathbf{Y}_f|\mathbf{y}_{c_f},\mathbf{U},\mathbf{S}) \propto \tag{1c}$$

$$\prod_f \exp\left(-\phi_{c_f}(\mathbf{y}_{c_f},\mathbf{o})\right)\exp\left(-\phi_f(\mathbf{Y}_f)\right) \tag{1d}$$

where, $p(\mathbf{o})$ is assumed to be uniformly distributed on the image domain, thus can be omitted in Eq.1c; ϕ_{c_f} and ϕ_f in Eq.1d are the energy for displacing the center and registering the part respectively.

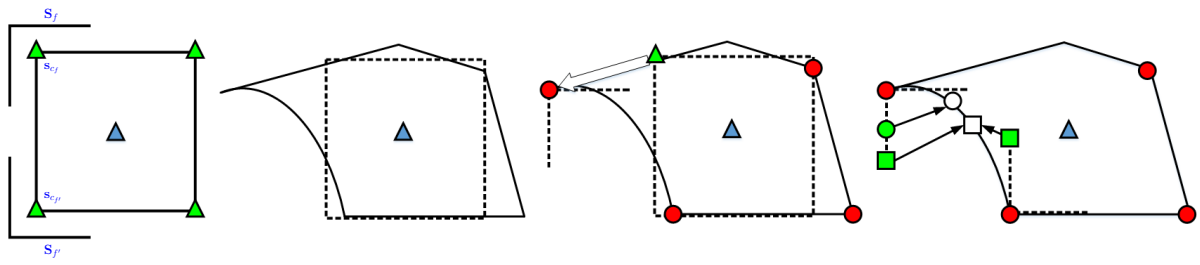
It is worth noting that matches of our parts are interconnected with each other. Their counterparts $\{\mathbf{Y}_f\}$ are the restriction of the same boundary $\mathbf{Y}$ on the respective set f. Given a point $\mathbf{s}_i$ shared by two parts $\mathbf{S}_f$ and $\mathbf{S}_{f'}$, the point in both parts should be deformed to the same point as shown in Fig.4(d). Thus through the shared points, we rejoin the connections among parts, making our model different from the star model.

The similarity between part $\mathbf{Y}_f$ and $\mathbf{S}_f$ are determined by two potential functions. The function ϕ_{c_f} measures the energy for displacing center of the part as shown in Fig.4(c). Its form is given by,

$$\phi_{c_f}(\mathbf{y}_{c_f},\mathbf{o}) = \frac{|f|}{2}\|\mathbf{y}_{c_f} - \mathbf{o} - \mathbf{s}_{c_f}\|_{\Sigma_{c_f}} \tag{2}$$

where $\|\cdot\|_{\Sigma_{c_f}}$ is the Mahalanobis distance: $(\cdot)^T\Sigma_{c_f}^{-1}(\cdot)$. Without loss of generality, we assume that center of the shape template $\mathbf{S}$ is aligned with the origin.

After the operation, we measure the energy for registering two centered point sequences. Energy for the second operation uses the similar form as chamfer distance[2]. It measures the shape difference by calculating the total distance between them as shown in Eq.3. Because scale and rotation have been excluded via the enumeration, unlike chamfer distance the distance ϕ_f does not suffer from the



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

Figure 4. A toy example for registration with the proposed layered structure. (a) gives the shape template that is to be aligned with the shape in (b) where centers of both shapes are aligned. (c) gives the new locations for the four corners of the square and the displaced parts with respected to the new locations and (d) illustrates the registration of the displaced parts of the template.

scale.

$$\phi_f(\mathbf{Y}_f) = \frac{1}{2} \sum_{i \in f \setminus \{c_f\}} \left[\frac{1}{\rho} d_{fi}(\mathbf{y}_i) + \delta_{\mathbf{U}}(\mathbf{y}_i) \right] \quad (3)$$

where

$$d_{fi}(\mathbf{y}_i) = \left\| (\mathbf{y}_i - \mathbf{y}_{c_f}) - (\mathbf{s}_i - \mathbf{s}_{c_f}) \right\| \quad (4)$$

The distance measure considers the influence of the missing correspondence, and $\delta_{\mathbf{U}}$ is an indicator function which equals to 0 when $\mathbf{y}_{c_f} \in \mathbf{U}$, and otherwise 1. It penalizes the case that $\mathbf{s}_i$ misses its counterpart in $\mathbf{U}$. That is to say, it requires that if the match point for $\mathbf{s}_i$ exists, then $\mathbf{s}_i$ should be a point in $\mathbf{U}$. The constraint forces the $\mathbf{Y}$ to follow the boundary of $\mathbf{U}$.

3.2. Shape Match with Part Variation

We introduce a latent model for the part, allowing shape of each part to deform within a linear space spanned by a set of shape bases. The latent model is shown in Fig.3(b) with its mathematical form in Eq.5.

$$p(\mathbf{Y}, \{\mathbf{z}_f\}, \mathbf{o} | \mathbf{U}, \mathbf{S}) \propto \quad (5a)$$

$$\prod_f p(\mathbf{y}_{c_f} | \mathbf{o}, \mathbf{U}, \mathbf{S}) p(\mathbf{Y}_f | \mathbf{y}_{c_f}, \mathbf{z}_f, \mathbf{W}_f, \mathbf{U}, \mathbf{S}) p(\mathbf{z}_f | \lambda) \quad (5b)$$

$$\propto \prod_f e^{-\phi_{c_f}(\mathbf{y}_{c_f}, \mathbf{o})} e^{-\phi_f(\mathbf{Y}_f, \mathbf{z}_f)} \exp\left(-\frac{\|\mathbf{z}_f\|^2}{2\lambda} \right) \quad (5c)$$

In the equation, ϕ_{c_f} keeps the same, and ϕ_f is modified to incorporate the latent variable and takes the form in Eq.6.

$$\phi_f(\mathbf{Y}_f, \mathbf{z}_f) = \frac{1}{2} \sum_{i \in f \setminus \{c_f\}} \left[\frac{1}{\rho} d_{fi} + \delta_{\mathbf{U}}(\mathbf{y}_i) \right] \quad (6)$$

The distance d_{fi} changes to,

$$d_{fi}(\mathbf{y}_i, \mathbf{z}_f) = \left\| (\mathbf{y}_i - \mathbf{y}_{c_f}) - (\mathbf{s}_i + \mathbf{W}_{fi}\mathbf{z}_f - \mathbf{s}_{c_f}) \right\| \quad (7)$$

The matrix $\mathbf{W}_{fi}$ contains a set of shape bases for the ith point. Size of the matrix is $2 \times L$, and L is the length of the latent variable $\mathbf{z}_f$.

For each cropped part $\mathbf{S}_f^{(0)}$, we randomly generated a set of affine transforms and operate them on the part. After that, a set of samples $\{\mathbf{S}_f^{(i)}\}$ are on hand, which are further analyzed with the principle component analysis (PCA), resulting in the mean shape $\mathbf{S}_f$ and a set of shape basis $\mathbf{W}_f$.

4. Approximate Match via Iterative DP

To search for optimal matching is now reduced to maximum a posterior estimation that can be further transformed to an energy-minimization in Eq.8.

$$\arg\min_{\mathbf{Y}, \{\mathbf{z}_f\}} \sum_f E_f(\mathbf{Y}_f, \mathbf{z}_f, \mathbf{o}) \quad (8)$$

The definition for E_f is given by Eq.9

$$E_f(\mathbf{Y}_f, \mathbf{z}_f, \mathbf{o}) = \phi_{c_f}(\mathbf{y}_{c_f}, \mathbf{o}) + \phi_f(\mathbf{Y}_f, \mathbf{z}_f) + \frac{\|\mathbf{z}_f\|^2}{2\lambda} \quad (9)$$

4.1. Pose Estimation through Voting

Due to noises and the descriptive capbability of the model, it is very difficult to find a global solution $\mathbf{Y}$ for the problem in Eq.8. To approach an approximate solution, we neglect the connections among parts of $\mathbf{Y}$ and deal with each part independently. The relaxation reduces the original problem to a set of subproblems in Eq.10.

$$\min_{\mathbf{Y}_f^f, \mathbf{z}_f} E_f(\mathbf{Y}_f^f, \mathbf{z}_f, \mathbf{o}) \quad (10)$$

It is worth mentioning that the part $\mathbf{Y}_f^f$ involved in Eq.10 is different from $\mathbf{Y}_f$. The new notation is not the restriction of whole shape $\mathbf{Y}$ on the subset f; it does not require that $\mathbf{y}_i^f$ must equal to $\mathbf{y}_i^{f'}$ for a shared point indexed by i.

The graph structure underlying E_f is then explored for the problem in Eq.10. It tells that $\mathbf{Y}_f^f$ is independent of

the pose parameter conditioned on the center. Given the assumption that the center point of a part is pinned on $\mathbf{u}_j$, the contour fragment can be optimized ignoring the influence of the pose. The conclusion leads to the decomposed minimization in Eq.11.

$$\min_j \phi_{c_f}(\mathbf{u}_j, \mathbf{o}) + \min_{\mathbf{z}_f, \mathbf{Y}_f^f : \mathbf{y}_{c_f}^f = \mathbf{u}_j} \phi_f(\mathbf{Y}_f^f, \mathbf{z}_f) + \frac{\|\mathbf{z}_f\|^2}{2\lambda} \quad (11)$$

Center point $\mathbf{y}_{c_f}$ in Eq.11 is forced to be on some edge point $\mathbf{u}_j$. For each $\mathbf{u}_j$, an optimal contour fragment centered on it can be optimized, resulting in an optimal value $\phi_{f|\mathbf{u}_j}$. Minimization with respect to j then takes the form as,

$$\min_j \phi_{c_f}(\mathbf{u}_j, \mathbf{o}) + \phi_{f|\mathbf{u}_j} \quad (12)$$

Problem in Eq.12 belongs to the generalized distance transform of sampled function. When Σ_{c_f} in the potential function ϕ_{c_f} is diagonal, fast algorithm supplied in [6] is applicable. For each part f, the minimization outputs a voting map for instance centers. By summing the voting maps of different fs, we reach a map for the object that indicates how possible the object is located on $\mathbf{o}$. By thresholding the map, a set of potential candidates for the object poses are on hand.

4.2. Iterative DP for Contour Fragments

The left piece for the puzzle is the calculation of $\phi_{f|\mathbf{u}_j}$. This involves the optimization w.r.t. $\mathbf{Y}_f^f$ and $\mathbf{z}_f$ under the constraint that center of the fragment is pinned on $\mathbf{u}_j$. Before describing the procedure, we summarize the problem in Eq.13 where ϕ_f is substituted by its definition.

$$\min \frac{\sum_{i \in f \setminus \{c_f\}} \left[\frac{1}{\rho} d_{fi}(\mathbf{y}_i^f, \mathbf{z}_f) + \delta_{\mathbf{U}}(\mathbf{y}_i)\right]}{2} + \frac{\|\mathbf{z}_f\|^2}{2\lambda} \quad (13)$$

Solution for the problem is achieved by an iterative loop alternating between the contour fragment and the latent variable. It comprises: (1) the gradient descent for $\mathbf{z}_f$ given current estimate of $\mathbf{Y}_f^f$; and (2) the dynamic programming for $\mathbf{Y}_f^f$ given the estimate of $\mathbf{z}_f$.

Given $\mathbf{Y}_f^f$, latent variable $\mathbf{z}_f$ is optimized by the steepest descent. The partial derivative w.r.t. $\mathbf{z}_f$ is calculated by Eq.14.

$$\frac{\partial}{\partial \mathbf{z}_f} = \frac{1}{2\rho} \sum_{i \in f \setminus \{c_f\}} \frac{\partial d_{fi}(\mathbf{y}_i^f, \mathbf{z}_f)}{\partial \mathbf{z}_f} + \frac{\mathbf{z}_f}{\lambda} \quad (14)$$

With the gradient, gradient descent iterates until a local optimum is found.

When latent variable is fixed, the optimization defined in Eq.13 reduces to the problem in Eq.15.

$$\min_{\mathbf{Y}_f^f : \mathbf{y}_{c_f}^f = \mathbf{u}_j} \frac{\sum_{i \in f \setminus \{c_f\}} \left[\frac{1}{\rho} d_{fi}(\mathbf{y}_i^f, \mathbf{z}_f) + \delta_{\mathbf{U}}(\mathbf{y}_i)\right]}{2} \quad (15)$$

If we neglect the fact that contour fragment is a point sequence, the optimization minimizes each i independently for a point $\mathbf{y}_i^f$; it searches for a point from the contour fragment that minimizes the distance $d_{fi}(\mathbf{y}_i^f, \mathbf{z}_f)$.

The optimization breaks the order of the contour fragment because it cannot guarantee that selected points $\{\mathbf{y}_i^f\}$ make a point sequence. The goal of the optimization here is to find partial boundary of the object, which is obviously a point sequence. Thus order is an indispensable factor. Since the fragment holding $\mathbf{u}_j$ is also a point sequence, the problem can be reduced to find a subsequence from the fragment which has minimum distance to the centered $\mathbf{S}_f$. The optimization can be fulfilled by dynamic programming (DP) proposed by Scott and Nowak[24]. Their DP can handle the case that some points in the sequence may loose their matches. DP then starts from $\mathbf{u}_j$, and outputs an optimal point sequence. For accuracy, the process can jump to the fragments adjacent to the one holding $\mathbf{u}_j$.

For initialization, we set $\mathbf{z}_f$ to $\mathbf{0}$ and start with the optimization of $\mathbf{Y}_f^f$. The refinement then alternates between the latent variable $\mathbf{z}_f$ and the part $\mathbf{Y}_f^f$. Our experiments show that after $2-3$ epochs, the process converges and we choose 2 for all parts.

5. The Matching Refinement

The way that optimizes each part independently neglects the connections among parts; the localized parts often lead to the inconsistent case as shown in Fig.5. Given a point $\mathbf{s}_i$ shared by two parts: f and f', the case shows that the matched points $\mathbf{y}_i^f$ and $\mathbf{y}_i^{f'}$ cannot meet each other.

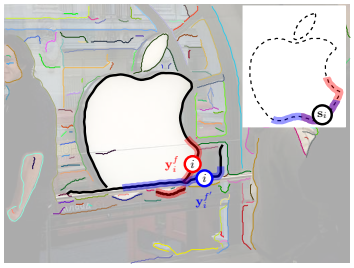


Figure 5. The defects led by the independent optimization of parts.

To resolve the inconsistency, we explicitly require that part $\mathbf{Y}_f^f$ must equal to the restriction of some shape $\mathbf{Y}$ on the set f. The matching problem with added constraints then takes the form as,

$$\begin{aligned} \min \quad & \textstyle\sum_f E_f(\mathbf{Y}_f^f, \mathbf{z}_f, \mathbf{o}) \\ \text{s.t.} \quad & \mathbf{Y}_f^f = \mathbf{Y}_f, \ \forall f \end{aligned} \quad (16)$$

The Dual decomposition (DD) [10, 22] introduces a set of the Lagrangian multipliers, each for a constraint. It then

gives the Lagrangian dual function as,

$$g(\{\lambda_f\}) = \min_{\{\mathbf{Y}_f^f, \mathbf{Y}\}} \sum_f \left[E_f(\mathbf{Y}_f^f, \mathbf{z}_f, \mathbf{o}) + (\mathbf{Y}_f^f - \mathbf{Y}_f)^T \lambda_f \right] \tag{17}$$

where λ_f is the vector for the Lagrange multipliers for the constraints involving $\mathbf{Y}_f^f$. After eleminating the $\mathbf{Y}$, the original problem is reduced to a set of slaves and a single master as shown in Eq.18.

$$\begin{cases} g_f(\{\lambda_f\}) = \min_{\mathbf{Y}_f^f} E_f(\mathbf{Y}_f^f, \mathbf{z}_f, \mathbf{o}) + (\mathbf{Y}_f^f)^T \lambda_f \\ \max_{\{\lambda_f\} \in \Lambda} \sum_f g_f(\{\lambda_f\}) \end{cases} \tag{18}$$

In our shape matching, each slave corresponds to a partial matching problem, and the master adjusts the multipliers in the constrained space Λ to restrain the slaves, making the localized parts meet each other at their shared points. To achieve the same effects, we propose a modified slave in Eq.19.

$$\min_{\mathbf{Y}_f^f, \mathbf{z}_f} E_f(\mathbf{Y}_f^f, \mathbf{z}_f, \mathbf{o}) + \frac{\gamma_t}{2} \sum_{i \in f} \sum_{f' \in F_i \setminus \{f\}} \|\mathbf{y}_i^f - \mathbf{y}_i^{f'}\| \tag{19}$$

Given a shared point $\mathbf{y}_i$, F_i is the set of all the segments containing the point. γ_t can be viewed as multiplier, controlling the influence of the added term. The slave makes its part to be consistent with its adjacent fragments at the shared points. It uses the shared points to rejoin the different parts. During the optimization, an iterative process is then implemented where parts of boundary $\mathbf{Y}$ are sequential refined with the value of γ_t gradually tuned.

5.1. Matching Refinement for Parts

After some rearrangement of the modified slave, we can follow the same procedure invented in the approximate stage to find the optimal part. This is fulfilled by absorbing functions of the multipliers into the potential functions. To show, we rewrite the modified potential function $\tilde{\phi}_{c_f}$ and $\tilde{\phi}_f$ in Eq.20,

$$\begin{cases} \tilde{\phi}_{c_f}(\mathbf{y}_{c_f}^f, \mathbf{o}) = \phi_{c_f}(\mathbf{y}_{c_f}^f, \mathbf{o}) + \\ \qquad \frac{\gamma_t}{2} \sum_{f' \in F_{c_f} \setminus \{f\}} \|\mathbf{y}_{c_f}^f - \mathbf{y}_{c_f}^{f'}\| \\ \tilde{\phi}_f(\mathbf{Y}_f^f, \mathbf{z}_f) = \phi_f(\mathbf{Y}_f^f, \mathbf{z}_f) + \\ \qquad \sum_{i \in f \setminus \{c_f\}} \frac{\gamma_t}{2} \sum_{f' \in F_i \setminus \{f\}} \|\mathbf{y}_i^f - \mathbf{y}_i^{f'}\| \end{cases} \tag{20}$$

For the modified potential $\tilde{\phi}_f$, the added term is absorbed in its distance metric, resulting in the modified distance in Eq.21.

$$\tilde{d}_{fi}(\mathbf{y}_i, \mathbf{z}_f) = d_{fi}(\mathbf{y}_i, \mathbf{z}_f) + \frac{\gamma_t}{2} \sum_{f' \in F_i \setminus \{f\}} \|\mathbf{y}_i^f - \mathbf{y}_i^{f'}\| \tag{21}$$

With the modified distance, the iterative DP is also applicable. However, since the estimated parts $\{\mathbf{Y}_f^f\}$ change with the pose, the modified potential function $\tilde{\phi}_f$ becomes pose dependent while the original ϕ_f is independent of the pose. Thus, the optimization of the modified potential $\tilde{\phi}_f$ is pose specific, subsequently the refinement can only be implemented pose by pose.

In one epoch for the refinement, all the parts are updated in turn. Experiments show that the procedure usually converges in 8 to 10 epochs. In the experiments, we chose 8 for all refinements.

6. Experimental Evaluation

6.1. Experimental Setup

The experiments were conducted on the ETHZ shape dataset[7] and the INRIA horses[7] dataset, and we followed the same experimental setup in [14, 18] that used a single template to detect and localize all its instances. The shape template is preprocessed such that each has a diagonal length of 256. The two datasets have six classes, and all come with the edge map by Berkeley edge detector [17]; to make the comparison fair, we used the same detector which finds the salient boundaries of the object while suppressing the noises.

Parameters for the matching were determined empirically; in fact, our method works for a wide range of parameters. Throughout the experiments, we make ρ equal to 20 and the $\Sigma_{c_f} = \text{diag}([200^2, 200^2])$. Parts are generated by cropping the shape template with the moving squares centered on the contour. In experiments, we test three different sizes: 48, 64, and 96 for the square.

6.2. Comparison

We compared our algorithm against oriented chamfer matching (OCM) [25], methods by Ferrari *et al.* [8, 7], the fast direction chamfer distance matching (FDCM)[14], and chamfer template matching (CTM)[18]. All the involved algorithms adopted 0.2 Intersection over Union (IoU) to determine an instance. Given an instance, we have its one-to-one correspondence from the shape template to edge map and thus get a set of edge points, i.e. $\mathbf{Y}$. The bounding box for the instance is then generated by calculating the bounding rectangle of these selected points.

We show the false positive per image (FPPI) vs. detection rate (DR) in Fig.6. The proposed method achieved the best performance in 4 out of 6 classes. For the bottles class, the proposed algorithm is slightly worse to the more recent CTM[18]. For the INRIA horses class, the detection accuracy of the proposed algorithm is almost same as the method by [7] that uses half of the samples for training. It slightly outperforms the method by [7] when FPPI is less than 0.5 and is slightly worse when FPPI goes beyond 0.5. In Fig.7, we give some localized results; among the localized false positives, some are very similar to our shape template.

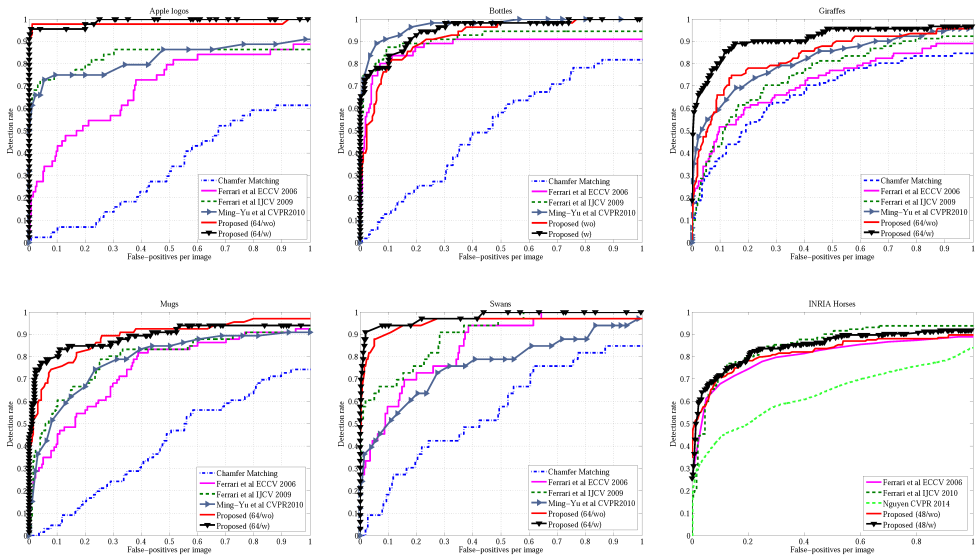


Figure 6. The FPPI vs DR curves for ETHZ dataset. All methods for comparison used the 0.2 overlapping ratio. 64 is the size of the square for generating parts; w stands for the match with part variation and wo for match without the part variation.

Table 1. Comparison for the detection rate for $0.3/0.4$ FPPI under 0.5 overlapping ratio.

	Applelogos	Bottles	Giraffes	Mugs	Swans	Mean
Proposed(48/w)	0.977/0.977	0.891/0.909	0.747/0.747	0.924/0.924	**1/1**	0.908/0.912
Proposed(48/wo)	0.977/0.977	0.946/0.946	0.725/0.747	0.879/0.939	**1/1**	0.905/0.922
Proposed(64/w)	**1/1**	0.964/0.982	0.758/0.758	0.849/0.849	0.970/0.970	0.908/0.912
Proposed(64/wo)	0.977/0.977	0.909/0.946	0.571/0.582	0.833/0.864	0.939/0.939	0.846/0.862
Proposed(96/w)	0.932/0.955	**1/1**	0.703/0.736	0.833/0.849	0.939/0.939	0.882/0.896
Proposed(96/wo)	0.886/0.909	0.964/0.964	0.517/0.539	0.788/0.803	0.939/0.939	0.819/0.831
Proposed(best)	**1/1**	**1/1**	0.758/0.758	0.924/0.924	**1/1**	0.936/0.936
Srinivasan[20]	0.95/0.95	**1/1**	0.872/0.896	0.936/0.936	**1/1**	**0.952/0.956**
Wang[27]	0.90/0.90	**1/1**	**0.92/0.92**	**0.94/0.94**	0.94/0.94	0.940/0.940
Maji[16]	0.95/0.95	0.929/0.964	0.896/0.896	0.936/0.967	0.882/0.882	0.919/0.932
Riemenschneider[21]	0.933/0.933	0.970/0.970	0.792/0.819	0.846/0.863	0.926/0.926	0.893/0.905
Fezenswalb[5]	0.95/0.95	**1/1**	0.729/0.729	0.839/0.839	0.588/0.647	0.821/0.833
Ferrari[7]	0.777/0.832	0.798/0.816	0.399/0.445	0.751/0.8	0.632/0.705	0.671/0.72
Gu[9]	0.906/-	0.948/-	0.798/-	0.832/-	0.868/-	0.871/-

For further evaluation, we compare the proposed algorithm against some state-of-art methods through the detection rate at $0.3/0.4$ FPPI. In Tbl.1, we summarized the results. All the involved algorithms adopted the PASCAL standard that uses 0.5 IoU. When best performance of our algorithm is considered, it gets the highest detection accuracy in 3 out of 6 classes. Our overall performance is very close to Wang *et al.*'s method [27] and slightly outperformed by Srinivasan *et al.*'s method [20]. On the other side, both Wang *et al.*'s [27] and Srinivasan *et al.* [20] require a training procedure for improving the descriptive capability of their shape models. Besides, Srinivasan *et al.* [20] have to combine the discriminative model with the descriptive model to further improve the accuracy. To conclude, the proposed algorithm achieves the comparable results with the state-of-the-art methods.

6.3. Influence of the Parameters

In the section, we discuss the influence of parameters from the use of refinement, the part variation, and size of the part and the centers of parts. First, experiments with/without connectivity priors have shown that the prior is critical for resolving the inconsistent case in the approximate stage and help pruning false positives, which leads to clear increase on detection accuracy. As reported in Tbl.2, the average precision (AP) for matching with refinement is significantly higher than that without the refinement.

The use of the part variation also benefits the model's

Figure 7. The localization results. (For left to right row: "Applelogos", "Bottles", "Giraffes", "Mugs", "Swans", "INRIA Horses".) Last row shows some examples of false positives.

Table 2. AP for match with and without refinement (0.5 IoU) recorded by "R" and "NoR" respectively.

	48/w		64/w		96/w	
	NoR	R	NoR	R	NoR	R
Applelogos	0.73	0.91	0.80	0.95	0.73	0.87
Bottles	0.82	0.82	0.77	0.84	0.72	0.92
Giraffes	0.60	0.65	0.56	0.65	0.49	0.62
Mugs	0.52	0.84	0.50	0.75	0.58	0.70
Swans	0.64	0.89	0.71	0.90	0.78	0.87
Horses	0.49	0.79	0.43	0.76	0.34	0.64

adaption to deformations, but it is less significant than the refinement. As shown in Tbl.1, matching with the variation gets higher detection rate than that without the part variation in nearly every case. The 'Bottles' under size 48 is the only exception. For 'Bottles', when the square size equals to 48, majority of the cropped parts are the straight lines. Use of the part variation makes the discriminative capability of parts degenerate. For others, part variation enlarges the deformation of the parts; and meanwhile, does not sacrifice the discriminative capability.

As for the size of the square, it determines the length of the cropped fragments, and subsequently influences the matching performance. Our matching algorithm gets higher detection performs when the size is 48 and 64. With the increase of the size, the cropped fragments has a higher risk that they cannot distribute on a single contour fragment and subsequently decrease the matching accuracy. This is the reason why smaller size gives relatively higher performance. But it does not mean that smaller is better. With the decrease of the size, we also face a risk that parts lose their discriminative capability just like the 'Bottles' discussed before. If the size is too small, parts cannot discriminate the contour fragments belonging to the object from those of the backgrounds. Thus, this parameter should carefully determined though it works in a wide range.

For centers of parts, they do affect our detection accuracy. To evaluate their influence, we use the same part size of 48 and generate four sets of centers, denoted by $s1 - s4$. The mean values of the average precision on the ETHZ dataset are 0.824, 0.813, 0.805 and 0.791 respectively. The variation mainly comes from two sources. The first is that the overlapping strategy will make some boundary points be counted more times than the others. Besides, due to the nature of shape, some parts are more salient than the other. Thus, when centers of the parts change, their covered points vary, which subsequently leads to the variation.

Besides, we report the time-complexity of the proposed algorithm. It is runned on a desktop with 4G memory and Intel Core i7 2.93GHz cpu. The proposed algorithm is implemented in Matlab with some codes written in C. For all the **6** classes, we do not consider rotation; while for scaling, we search each image with 10 scales that distribute equally in [0.3,1.8]. For each scale s, the detected contour fragments $\mathbf{U}$ are rescaled with a factor of 1/s to eliminate the scale difference between the template and image. The average processing time of an image is 68s. Specifically, generation of contour fragments uses about 2-3s, the approximation stage takes around 13s, and the refinement is about 53s. When part variation is considered, another 4 seconds is used. The running times under different sizes of the square are similar because while the increase of the size will decrease the number of the parts, it will also increase the number of points in each part, such that the two factors counter affect each other, and the overall running time keeps similar.

7. Conclusion

In the paper, we supply a solution for matching shape in the cluttered images. It works with contour fragments, picks the involved ones with the aid of the shape template, and links them into the boundary. The algorithm noted the fact that an object's boundary is usually scattered in several contour fragments. In response to it, it divides the shape template into overlapped parts. Each part is used for localizing candidates by solving a partial matching problem. Thanks to the contour fragments, the partial matching is robust to noises, but it suffers from the loose connection among the localized candidates. Thus, a refined procedure is utilized to resolve the inconsistency among candidates and makes the partial matching more reasonable, that is verified by the experiments. In the further, we will extend the model to object segmentation, since resolved boundary gives a rough approximation to the object boundary.

Acknowledgements: This work was supported by Natural Science Foundation of China under Grant NO. 61305051 and 61328303.

References

[1] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *PAMI*, 2002. 2

[2] G. Borgefors. Hierarchical chamfer matching: a parametric edge matching algorithm. *PAMI*, 1988. 2, 3

[3] J. Coughlan and H. Shen. Shape matching with belief propagation: Using dynamic quantization to accomodate occlusion and clutter. In *CVPRW*, 2004. 2

[4] P. F. Felzenszwalb. Hierarchical matching of deformable shapes. In *CVPR*, 2007. 2

[5] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *PAMI*, 2010. 7

[6] P. F. Felzenszwalb and D. P. Huttenlocher. Distance transforms of sampled functions. Technical report, Cornell Computing and Information Science, 2004. 5

[7] V. Ferrari, F. Jurie, , and C. Schmid. From images to shape models for object detection. *IJCV*, 2010. 1, 2, 6, 7

[8] V. Ferrari, T. Tuytelaars, and L. Van Gool. Object detection by contour segment networks. In *ECCV*. 2006. 2, 6

[9] C. Gu, J. Lim, P. Arbelaez, and J. Malik. Recognition using regions. In *CVPR*, 2009. 7

[10] N. Komodakis, N. Paragios, and G. Tziritas. Mrf energy minimization and beyond via dual decomposition. *PAMI*, 2011. 5

[11] P. M. Kumar, P. Torr, and A. Zisserman. Extending pictorial structures for object recognition. In *BMVC*, 2004. 2

[12] B. Leibe, A. Leonardis, and B. Schiele. Combined object categorization and segmentation with an implicit shape model. In *ECCV workshop on statistical learning in computer vision*, 2004. 2

[13] M. Leordeanu, M. Hebert, and R. Sukthankar. Beyond local appearance: Category recognition from pairwise interactions of simple features. In *CVPR*, 2007. 2

[14] M.-Y. Liu, O. Tuzel, A. Veeraraghavan, and R. Chellappa. Fast directional chamfer matching. In *CVPR*, 2010. 1, 2, 6

[15] C. Lu, L. Latecki, N. Adluru, X. Yang, and H. Ling. Shape guided contour grouping with particle filters. In *ICCV*, 2009. 1, 2

[16] S. Maji and J. Malik. Object detection using a max-margin hough transform. In *CVPR*, 2009. 1, 7

[17] D. Martin, C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *PAMI*, 2004. 6

[18] D. T. Nguyen. A novel chamfer template matching method using variational mean field. In *CVPR*, 2014. 1, 2, 6

[19] A. Opelt, A. Pinz, and A. Zisserman. A boundary-fragment-model for object detection. In *ECCV*. 2006. 2

[20] Q. Z. Praveen Srinivasan and J. Shi. Many-to-one contour matching for describing and discriminating object shape. In *CVPR*, 2010. 1, 2, 7

[21] H. Riemenschneider, M. Donoser, and H. Bischof. Using Partial Edge Contour Matches for Efficient Object Category Localization. In *ECCV*, 2010. 7

[22] M. Salzmann. Continuous inference in graphical models with polynomial energies. In *CVPR*, 2013. 5

[23] F. Schmidt, D. Farin, and D. Cremers. Fast matching of planar shapes in sub-cubic runtime. In *ICCV*, 2007. 2

[24] C. Scott and R. Nowak. Robust contour matching via the order-preserving assignment problem. *TIP*, 2006. 2, 5

[25] J. Shotton, A. Blake, and R. Cipolla. Multiscale categorical object recognition using contour fragments. *PAMI*, 2008. 1, 2, 6

[26] Y. Su and Y. Liu. A voting scheme for partial object extraction under cluttered environment. *IJPRAI*, 27(2):1–37, 2013. 2

[27] X. Wang, X. Bai, T. Ma, W. Liu, and L. J. Latecki. Fan Shape Model for Object Detection. In *CVPR*, 2012. 7