2011 23rd IEEE International Conference on Tools with Artificial Intelligence

Combining Shape and Appearance for Automatic Pedestrian Segmentation

Yanli Li, Zhong Zhou, Wei Wu
State Key Laboratory of Virtual Reality Technology and Systems, Beihang University
School of Computer Science and Engineering, Beihang University
Beijing, P.R.China
{*liyl, zz, wuwei*}*@vrlab.buaa.edu.cn*

***Abstract*—In this paper we present an approach to automatically segmenting non-rigid pedestrians in still images. Inspired by global shape matching as well as interactive figure-ground separation methods, this approach fulfills the task combining shape and appearance cues in a unified framework. The main idea is to initially extract pedestrian silhouette and skeleton via hierarchical shape matching, and then generate an appearance trimap to refine segmentation. The major contributions of this paper include: 1) a novel shape matching scheme, which is proposed to replace the commonly used Chamfer matching in the shape matching stage; 2) a head-torso parsing method, which is developed for localizing pedestrian to reduce the search space; 3) an automatic trimap generation method used to refine segmentation. Experiments on public datasets demonstrate that the approach improves pedestrian segmentation efficiently and effectively.**

***Keywords*-pedestrian segmentation; shape matching; skeleton extraction; head-torso parsing; trimap generation**

I. Introduction

Pedestrians, as the principal actors in daily life, have been widely studied in computer vision. Pedestrian segmentation is a fundamental task for many applications in artificial intelligence field, such as action recognition, scene understanding and human-computer interaction, etc. However, this has proven to be a challenging task due to inherent pedestrian articulation, appearance variances and cluttered backgrounds.

Shape and appearance are two commonly used cues for pedestrian segmentation. Shape is characterized as one-dimensional curve, thus is invariant to lighting conditions and object colors. But a conventional shape matching algorithm [1] is sensitive to cluttered backgrounds. Boundary points on a shape template are often mistakenly matched with edges apart from object contour in the image. Appearance has the advantage of preserving the relative uniform color/texture information for a single object. It is widely used to distinguish the foreground objects from background scene [2][3]. However, without constraint of shape prior, those low-level segmentation methods tend to over or under segment pedestrians. Obviously, neither shape nor appearance alone can automatically extract satisfactory silhouettes.

In this paper, we present an approach for pedestrian segmentation which incorporates shape and appearance cues. We limit our attention to upright pedestrians. It is worth noting that this approach is not limited to pedestrian but can be applied to any non-rigid objects. The input to our framework are pedestrian bounding boxes produced by a person detector, e.g., HOG-LBP [4]. A shape matching scheme is first employed to extracted pedestrian silhouette and skeleton. For speeding up, we organize the set of shape templates in a hierarchical tree, and present a part detector to lock onto the head and torso. Based on the silhouette and skeleton, we then obtain the foreground trimap and refine pedestrian by solving a MRF energy function.

According to the processing stages mentioned above, this paper is organized as follows. In Section II we summarize the related previous work. Section III shows the head-torso parsing process. The details of the hierarchical shape matching are described in Section IV. Section V presents trimap generation and pedestrian refinement. In Section VI we list the experimental results. Some conclusions and discussions are given in the last section.

II. Related work

Numerous approaches have been proposed for pedestrian segmentation. These approaches can be roughly classified into three categories: shape-based, appearance-based, shape-and-appearance-combined approaches.

A. Shape based approaches

The first category uses shape information as the main discriminative cue, including global shape templates and local contour features. Methods based on global shape templates segment pedestrians by matching shape templates with the feature image (e.g., the edge map). Effective shape registration plays a central role. For example, Gavrila [1] match global templates using Distance Transformation(DT) and Chamfer matching. Active contour models [5] try to attach points of the template to object boundaries by iteratively solving a global energy in level-set space. Although global shape matching methods can efficiently localize the object, the pixel-level segmentation is far from satisfactory under cluttered backgrounds and occlusion due to shape variances.

In contrast, local contours are more flexible and tolerant to occlusion, such as the contour features employed in Opelt et al. [6], the edgelet in Wu et al. [7] and the part-template in Lin et al. [8]. Methods based on local contours delineate

1082-3409/11 $26.00 © 2011 IEEE
DOI 10.1109/ICTAI.2011.61

pedestrian boundaries by selection of contour features in supervised manner. The selected features lie on the object boundaries, thus their responses on the query image help to segment the object. Although these methods are more effective for object detection, the segmentation results are unsatisfactory either. For example, in Opelt et al. [6], many similar fragments are selected around the pedestrian boundary, causing ambiguous delineation of the boundary. In Wu et al. [7] and Lin et al. [8], pedestrians are restricted to frontal and rear views, and the arms' segmentation is omitted.

B. Appearance based approaches

The second category uses appearance information to separate class specific objects from background, involving fragment clustering methods [9][10] and interactive figure-ground separation methods [2][3]. Based on "bag of words", Leibe et al. [9] explore the idea of learning a codebook of appearance parts for interleaved segmentation and classification. They arrange the fragments in star-style, detect objects by voting in the Hough space, and backproject foreground fragments to delineate pedestrian silhouette. Another technique is to arrange the learned fragments in CRF and segment the foreground object by solving a global energy function through graph cut [3], such as Larlus et al. [10]. However, these methods fail to extract a clear boundary without the constraint of shape contour.

More recently, interactive figure-ground separation methods draw lots of attention, e.g., GrabCut [2] and matting techniques [11]. Under the indication of some scribbles drawn to distinguish foreground from background, a MRF energy function is built for optimizing the selected object. Optionally, the boundaries can be further optimized using matting methods [11]. However, the interactive property limits its applications and segmenting an object with complex structures is cumbersome.

C. Shape and appearance combined approaches

Drawing advantages of the above two categories, some authors suggest combining shape (top-down segmentation) and appearance (bottom-up segmentation) cues for class specific object segmentation. One technique is implemented by grouping the detected over-segmented fragments under shape guide, such as Borenstein et al. [12] and Cour et al. [13]. However, these methods are only suitable for segmenting rigid objects or non-rigid objects with limited shape variation under the limited shape templates. For highly articulated pedestrians, the limited fragment templates cannot fully capture all poses.

Comparably, methods based on part parsing are more suitable for non-rigid object segmentation, such as OBJ-CUT [14]. The OBJ-CUT method [14] is tolerant to pose variances as it considers non-rigid object as a layered pictorial structure with each layer encoding the similar appearance cue. By iteratively parsing and refining the layers in CRF, it obtains good results. Similarly, Eichner et al. [15] present a part appearance model for pedestrian parsing and part-specific soft-segmentation. The part appearance models are built on some generic part detector and further used to improve the part detection. However, the arm and leg parsing in these part-parsing based methods are inaccurate in some cases, resulting in inaccurate segmentation. Guan et al. [16] present a method to segment pedestrian based on pedestrian parsing too. Instead of automatically parsing pedestrian, they manually extract pedestrian skeleton. By incorporating a variety of image cues including silhouette overlap, edge distance, and smooth shading, their method obtains wonderful segmentation results for naked or minimally clothed people. However, the interactive manipulation limits its applications.

Our segmentation approach falls into the last category. Here, motivated by global shape matching method [1] and interactive figure-ground separation approaches [2], we present an automatic pedestrian segmentation approach. Compared to the traditional shape matching approach [1], our approach facilities appearance cue to refine the segmentation results, thus is more tolerant to local appearance variances. Comparing with interactive figure-ground separation [2], our approach can automatically extract pedestrian silhouette and skeleton for segmentation, thus avoids cumbersome manipulation and has more applications. In contrast to previous combined approaches [12][13], this approach is characterized by the utilization of the automatically generated trimap for human segmentation, which encodes the constraint of shape as well as skeleton.

III. HEAD-TORSO PARSING

The input to our approach are pedestrian bounding boxes output by a generic pedestrian detector [4]. Since the pedestrians are only roughly localized in the bounding boxes, we propose a part parsing method to more precisely localize pedestrian head and torso within the bounding boxes for further shape matching. The part parsing consists of two steps: first, a part detector(described in Section III-A and Section III-B) is present to build the part confidence maps for pedestrian head and torso; second, the head and torso confidence maps are combined with their configuration priors to lock onto pedestrian head(described in Section III-C).

A. Part detector — learning phase

Our part detector is built on the Hough voting scheme[6]. In the learning phase, we obtain a set of scale normalized parts with the corresponding edge and mask maps. To construct a star-style constellation, we uniformly extract the sample points along the mask contour. Then we extract the 96-dimensional Shape Context features (SC) [17] for each sample points(as shown in Fig. 1(a)). In this way, a set of features is obtained. We divide these features into several groups with respect to their orientations. An additional

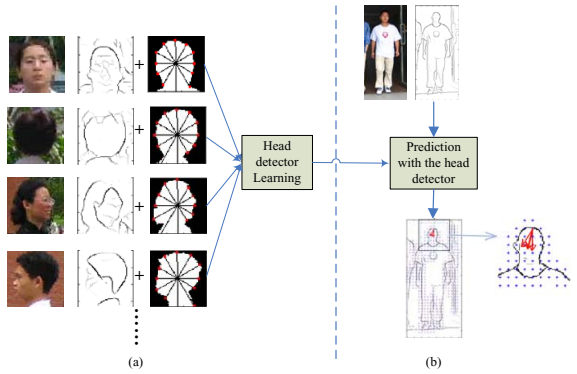


Figure 1. Example of learning and detection with the head detector. (a)Learning a head detector; (b)The head localization with the head detector in Hough voting space, in which the enlarged box shows a sample point voting to three centroids(in red).

group collecting negative samples is generated by randomly taking some feature points in non-part regions. We utilize random forest [18] to train the multi-classifier. Meanwhile, for each group, we statistically obtain the mean and the variance of the relative distances to the centroid, obtaining $\{(\mu_i, \sigma_i)|i = 1, \cdots, M_L + 1\}$. Here, $M_L + 1$ is the group number .

B. Part detector — detection phase

In the detection phase (as shown in Fig. 1(b)), the response at a candidate point is calculated using probabilistic voting, in which votes are accumulated in a circular search window around the candidate point. More specially, we first uniformly take sample points in the bounding box and extract their SC features, and then obtain their responses for each group through the multi-classifier. Now, a sample point set $Q_R = \{q_k, v_k, s_{k,i}|k = 1, \cdots, K_R, i = 1, \cdots, M_L + 1\}$ is generated, where K_R is the number of sampled points, $q_k = (q_{k,x}, q_{k,y})$ is the position of the k-th sample point, v_k is the 96-dimensional feature vector, $s_{k,i}$ is the response score for the group i. The confidence map C is formulated as the accumulation of responses from all sample points:

$$C = \sum_{k=1}^{K_R} \sum_{i=1}^{M_L} C_{k,i} \tag{1}$$

where $C_{k,i}$ is the individual response map with the same size to the image, and calculated by the voting of the k-th sample point in the i-th orientation. For the position z in the confidence map $C_{k,i}$, its response is defined as:

$$C_{k,i}(z) = \begin{cases} w(r_i) \min(s_{k,i}, \tau_s), & \|r_i\|_2 < \sigma_i \& s_{k,i} > T_s \\ 0, & otherwise \end{cases} \tag{2}$$

Here, τ_s is the truncating value, T_s is a threshold, r_i is the offset distance relative to the candidate centroid, i.e., $r_i = z - (q_k + c_i)$, in which $c_i = (c_{i,x}, c_{i,y})$ denotes the relative candidate centroid and is formulated as:

$$\begin{cases} c_{i,x} = \mu_i * cos(\theta_i) \\ c_{i,y} = \mu_i * sin(\theta_i) \end{cases} \tag{3}$$

$w(r_i)$ is the weighting value, $w(r_i) = 1 - \sqrt{r_{i,x}^2 + r_{i,y}^2}/\sigma_i$, $(\mu_i, \sigma_i, \theta_i)$ are the learned distance mean, the distance variance and the orientation for the i-th group respectively, as described in Section III-A.

This simple star-style constellation is flexible enough to cope with large variation in shape and appearance. However, as illustrated in Fig. 2(c), the confidence maps tend to generate several candidates in some cases. For example, the feet often give high responses in the head's confidence map. Thus, a further global parsing is necessary to sharply localize pedestrian parts, as stated in the next subsection.

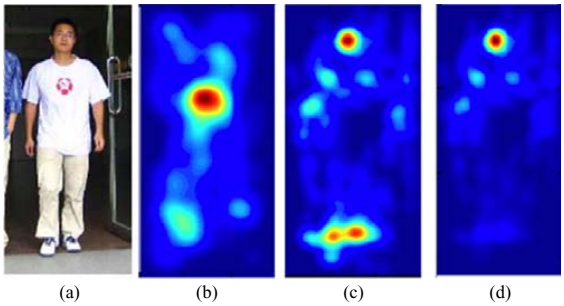


Figure 2. Example of part detection for the head and torso. (a) Input image; (b) The confidence map for the torso after part detection; (c) The confidence map for the head after part detection; (d) The confidence map for the head after part parsing.

C. Part parsing

The head-torso parsing is performed by combining the head and torso confidence maps with their configuration priors. Based on the Bayesian perspective, given the image evidence I, the posterior of the part configuration L is modeled as $p(L|I) \propto p(I|L)p(L)$, where $p(I|L)$ is the likelihood of the image evidence given a particular body part configuration, $p(L)$ is the configuration prior.

Let the location of the torso and head be parameterize as $L = \{l_0, l_1\}$. Assuming that the different part evidence maps are conditionally independent given the configuration L, and that the part map I_i for part i only depends on its own configuration l_i, the likelihood map can be simplified as: $p(I|L) = \prod_{i\in\{0,1\}} p(I_i|L) = \prod_{i\in\{0,1\}} p(I_i|l_i)$. In our framework, the likelihood map is represented by the confidence map, i.e., $p(I_i|l_i) \propto C_i(l_i), i \in \{0, 1\}$.

The configuration prior is factorized as: $p(L) \propto p(l_0 - l_1) \prod_{i\in\{0,1\}} p(l_i)$, in which the priors $p(l_i)$ for the torso and head are modeled using independent Gaussian distribution in the image coordination, $p(l_0 - l_1)$ is the relative spatial position prior, also modeled as a Gaussian distribution. We statistically learn the mean and variance values for

these three priors. Combining the likelihood and priors, the posterior of the configuration can be rewritten as:

$$p(L|I) \propto \prod_{i\in\{0,1\}} C_i(l_i) \prod_{i\in\{0,1\}} p(l_i)p(l_0 - l_1) \quad (4)$$

To quickly maximize the posterior, we utilize message propagation [19] for optimization in several bounds. In our framework, there are only two message maps, i.e, $m(0,1)$ and $m(1,0)$, representing the torso-to-head and head-to-torso message transferring. The initial message maps are derived as: $m_0(i,j) = \log(C_i(l_i)) + \log(p(l_i)), i,j \in \{0,1\}$. Iteratively, the message maps are updated as follows: $m_t(i,j) = m_{t-1}(j,i) + \log(p(l_i|l_j))$. Note that the confidence map $C_i(l_i)$ and the message maps $m_t(i,j)$ are all normalized before being applied to transfer messages. As shown in Fig. 2(d), this global part parsing scheme can significantly prune out the false locations.

IV. Hierarchical shape matching

Shape matching is used to search the best matched template and to extract pedestrian silhouette. In general case, it is the edge map instead of the original image to be aligned with the templates. In our approach, the edge map is given by the Pb edge detector [20] which encodes the real-valued magnitude and orientation information. To measure the similarity between a template and the edge map, various matching schemes are presented, of which Chamfer matching is most commonly used [1][7][8]. However, Chamfer matching is not tolerant to local deformation in cluttered backgrounds. Here, we present a smoother shape measurer to extend Chamfer matching.

A. Template alignment

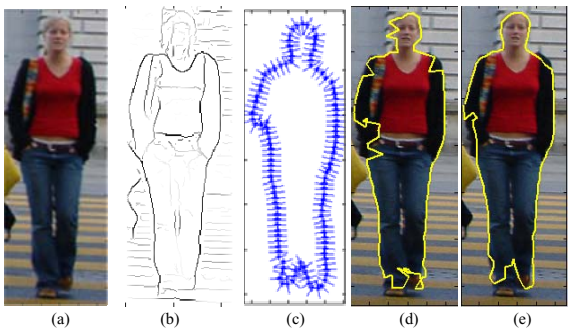
Chamfer matching: In its complete form, Chamfer matching takes two point sets: $E = \{e\}$ for the edgels of the edge map, $T = \{t\}$ for the sample points of the shape template, and evaluates the Chamfer distance as a function of relative position p :

$$D(T,E,p) = \frac{1}{|T|}\sum_{t\in T} \min_{e\in E}(D_1(t,e,p) + \alpha D_2(t,e,p)) \quad (5)$$

where $D_1(t,e,p) = min(\|(t+p) - e\|, \tau_1)$, $D_2(t,e,p) = min(|o(t+p) - o(e)|, \tau_2)$, $o(\cdot)$ denotes the orientation value. Both τ_1 and τ_2 are truncating values, and α is a weighting value.

From the Chamfer matching definition and experiments, we notice that: 1) the neighboring sample points along the template T tend to be inconsistently aligned with points of the edge map in cluttered backgrounds, resulting in the zigzags (see Fig. 3(d)); 2) it is time-consuming to search all points in the fixed search window. The time complexity is $O(n^2)$, n is the radius of the search window.

Based on the above two observations, we present a novel shape matching scheme. To reduce search space, we only search along the one-dimensional normal line segments of the template sample points(see Fig. 3(c)). For sample point t, the normal line segment is defined as a point set: $S(t) = \{s(i,t)|i = -M_S, \cdots, M_S\}$, where $s(i,t) = \langle i * sin(o(t)), i * cos(o(t))\rangle$, $2M_S + 1$ is the total length of the line segment (in pixels), and $o(\cdot)$ indicates the orientation value. Thus the shape matching can be considered as a labeling problem. It is to assign a unique label $l(t) \in [-M_S, M_S]$ to sample point $t \in T$. Under the label $l(t)$, the matched point to t in the edge map is $q(l(t)) = p + t + s(l(t))$.



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

Figure 3. Example of silhouette extraction by aligning the matched shape template with the edge map. (a)Input image; (b)The edge map obtained by the Pb edge detector; (c) The matched template with the overlaid normal line segments on the sample points; (d) The silhouette extracted by the Chamfer matching; (e) The silhouette extracted by the proposed shape matching.

To provide smoother alignment, we add a smooth term to the Chamfer distance. Hence, with the constraint of neighboring labeling, a MRF function is formulated as:

$$D(T,\hat{L},p) = \frac{1}{|T|}\sum_{t\in T}(D_d(l(t)) + \alpha_1 D_s(l(t), l(t+1))) \quad (6)$$

where $\hat{L}$ is the label set, i.e., $\hat{L} = \{l(t)|t \in T\}$, $D_d(\cdot)$ is the data term and $D_s(\cdot,\cdot)$ is the smooth term. $D_d(\cdot)$ encodes the cost when the sample point t is labeled as $l(t)$, and formulated as:

$$D_d(l(t)) = \min(\bar{g}(q(l(t))), \tau_1) + \alpha_2 \min(\bar{o}(t) - \bar{o}(q(l(t))), \tau_2) \quad (7)$$

$D_s(\cdot,\cdot)$ represents the cost when the adjacent sample points t and $t+1$ are labeled as $l(t)$ and $l(t+1)$. It is defined by:

$$D_s(l(t), l(t+1)) = \min(|l(t) - l(t+1)|, \tau_3) \quad (8)$$

In the above formulations, τ_1, τ_2, τ_3 are truncating values, α_1, α_2 are weighting values, $\bar{g}(\cdot)$ and $\bar{o}(\cdot)$ denote the normalized magnitude and orientation information respectively.

Obviously, the data term encourages assigning a point in the edge map with strong magnitude and similar orientation to the sample point t, and the smooth term penalize assigning different labels for neighboring sample points. They can be seen as an external energy and internal energy individually. The graph cut algorithm [3] is invoked to minimize

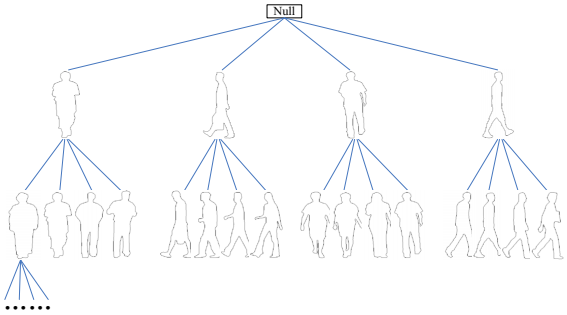


Figure 4. The hierarchical shape tree.

$D(T, \hat{L}, p)$ to obtain the global minimum energy(similarity score) as well as the silhouette points in the image. As shown in Fig. 3(e), this shape matching scheme gives a smoother alignment to the edge map, thus is more tolerant to local deformation, slight scale, and position than Chamfer matching.

B. Construction of hierarchical shape tree

As pedestrian shapes are highly variant, a set of shape templates is used to search the best matched template. For efficiency, those templates are organized in a hierarchical tree, in which similar templates are grouped together and represented with a prototype, as shown in Fig. 4. Shape matching is implemented as a process of traversing the tree to find the best matched prototype. Once the similarity score with a prototype is above a threshold T_t, its following subtrees will not be visited, thus a significant speed-up can be achieved.

Taking each shape template as a node of an Undirected Complete Graph (UCG) $G =< V, E, W >$, the construction of the tree can be considered as a problem of hierarchical graph clustering. This is a well-studied NP-hard problem in graph theory, involving some bottom-up clustering methods and top-down partition methods. Here, following the theory of spectral clustering [21], we construct the hierarchical tree in top-down manner. For the UCG $G =< V, E, W >$, we first calculate the edge weights matrix $W = \{w(i,j)|i,j \in V, (i,j) \in E\}$. The entity of the matrix W is defined as : $w(i,j) = D(i,j) + D(j,i)$. $D(i,j)$ is the similarity score between the template shape i and the template mask j, as described in (6). $D(j,i)$ is obtained similarly. Note that all the templates and masks have been aligned and scale normalized.

Spectral clustering partitions a graph into K subsets based on the normalize cut criterion:

$$Ncut_K = \sum_{i=1}^{K} \frac{cut(A_i, V - A_i)}{assoc(A_i, V)} \tag{9}$$

where $assoc(A,B) = cut(A,B) = \sum_{u \in A, v \in B} w(u,v)$.

We utilize this approximate solution to iteratively divide the graph to construct a hierarchical tree. At first, nodes in the Graph G are divided into K_1 subsets. Then, for each subset, the spectral clustering is employed again to partition it into K_2 sub-subsets. The process is recursively implemented until the number of clustering nodes is lower than a constant value K_t. The prototype of a subtree is taken as the template with the smallest mean similarity score to the other templates in the subset. Taking each subset with its prototype as a subtree, the hierarchical tree is constructed.

Shape matching is applied as a coarse-to-fine traversal along the tree. In the traversal procedure, all visited templates with the similarity score as well as the labeling results(as described in Section IV-A) are added to a visiting list. At the leaf level, all template exemplars are to be matched, whereas, at the non-leaf level, it is the prototypes derived to be aligned with the edge map. If the similarity score of a prototype is above a threshold, all of its subtrees would not be visited, otherwise, the prototype is added to the list and the subtrees are traversed recursively. At last, we choose the template with the minimum similarity score in the visiting list as the best matched shape, meanwhile obtain the pedestrian silhouette it represents.

V. Constraint Pedestrian Refinement

Due to pose variances, pedestrian silhouettes produced in the shape matching stage are inaccurate in some cases. In this section, we refine pedestrian segmentation with appearance cue. Comparing with the interactive figure-ground separation methods [16][2], this refinement is automatically performed through the generated trimap which encodes pedestrian shape and skeleton information. Pedestrian shape is derived directly from shape matching, and the skeleton is transferred from the template skeletons. In Section V-A, we describe how to estimate the skeleton and the trimap. And in Section V-B, we state the pedestrian refinement procedure.

A. Skeleton and trimap generation

Pedestrian skeleton is composed of a set of line segments each being connected by two joints, indicating the head, torso, upper or lower arm, upper or lower leg parts, as shown in Fig. 5(b). In the learning phase, we manually click joints in shape masks to yield the skeleton. For each point in the skeleton, we calculate its normal line and obtain the left and right crossing point between the normal line and the mask contour, resulting in a set $\{sp_i, lp_i, rp_i | i = 1, \cdots, K_S\}$. Here, sp_i is the skeleton point, lp_i and rp_i are the left and right crossing point, K_S is the skeleton length(in pixels).

In the testing phase, we have obtained pedestrian silhouette in which each point is matched to a sample point of the template contour(see Section IV-A), thus the skeleton can be easily transferred from the template to still image under the guide of silhouette. The skeleton in the image is denoted by: $\{sp'_i, lp'_i, rp'_i | i = 1, \cdots, K_S\}$, where $sp'_i = p + lp'_i + r_1 \| rp'_i - lp'_i \|$, p is the relative position, lp'_i is the matched point of

Figure 5. The generation of trimap based on initial silhouette and skeleton. (a) Input image; (b) The matched template overlaid with joints and skeleton; (c) The aligned contours as well as the skeleton on the original image(the green contour corresponds to the template shape, the yellow one indicates the aligned silhouette, and the blue lines show the aligned skeleton); (d) The generated trimap, in which the "Foreground" region is denoted in white, the "Unknown" region is in gray and the "Background" region is in black.

lp_i, rp'_i is the matched point of rp_i, r_1 is a pre-computed value and defined as: $r_1 = \|rp_i - sp_i\|/\|rp_i - lp_i\|$.

Based on the silhouette and skeleton point set $\{sp'_i, lp'_i, rp'_i | i = 1, \cdots, K_S\}$, the trimap in the bounding box is automatically generated in the following way: For each pixel q, we first find a sample point in the skeleton with the minimum distance to it, i.e., $(D_{min}, k_{min}) = \min_i(|q - sp'_i|)$, where D_{min} is the minimum distance and k_{min} is the index. Then we compute the ratio $r_2 = \min(|q - lp'_{k_{min}}|, |q - rp'_{k_{min}}|)/D_{min}$. Given two thresholds T_1 and T_2 satisfying $0 < T_1 < T_2$, if $r_2 < T_1$, the pixel is assigned as "Foreground", else if $r_2 > T_2$, the pixel is assigned as "Background", otherwise, the pixel is assigned as "Unknown". So far a trimap is generated(see Fig. 5(d)).

B. Human refinement via graphcut

To refine the "Unknown" region in the trimap, we follow the standard MAP-MRF approaches [2], and formulate a global energy function as follows:

$$D(\hat{L}) = \sum_{i \in V} D_d(l(i)) + \lambda \sum_{(i,j) \in E} D_s(l(i), l(j)) \qquad (10)$$

Here, $\hat{L}$ is the label set, i.e., $\hat{L} = \{l(i) | i \in V\}$, V is the set of pixels in the "Unknown" region, E is the set of neighboring pixel, $l(i) \in \{0, 1\}$ is the labeling assignment for pixel i ($l(i) = 0$ means it is assigned to the "Background", and $l(i) = 1$ means to the "Foreground"), λ is the weighting value, D_d and D_s are the data and smooth term respectively.

Several color models have been suggested for the definition of the data term, including K-Means, Histogram and Gaussian Mixture Model (GMM). We use the GMM in our implementation. Based on the foreground and background region of the trimap, two GMM models are estimated. Each GMM model, one for the background and one for the foreground, is taken to be a full-covariance Gaussian mixture

Figure 6. Two example results after refinement and matting. (a) Input images; (b)The pedestrian masks obtained after refinement; (c) The alpha images obtained after matting; (d) The final extracted pedestrians.

with K_G components. The parameters of GMM models are defined as: $\{(\mu_k^J, \Sigma_k^J) | k = 1, \cdots, K_G, J \in \{B, F\}\}$, where (μ_k^F, Σ_k^F) is the mean and covariance for the foreground, and (μ_k^B, Σ_k^B) for the background. For a pixel with the foreground labeling in the trimap, the data term is defined as: $D_d(l(i) = 0) = 0$ and $D_d(l(i) = 1) = \infty$. For a pixel with the background labeling, the data term is defined as: $D_d(l(i) = 0) = \infty$ and $D_d(l(i) = 1) = 0$. For the pixels on the "Unknown" regions, the data term is:

$$\begin{cases} D_d(l(i) = 0) = d_i^F/(d_i^F + d_i^B) \\ D_d(l(i) = 1) = d_i^B/(d_i^F + d_i^B) \end{cases} \qquad (11)$$

where $d_i^J = \min_k \|(I(i) - \mu_k^J)' \, \Sigma_k^J (I(i) - \mu_k^J)\|$ is the similarity value between its color and the GMM components.

D_s is the smoothness term, which is defined as:

$$D_s(l(i), l(j)) = \|I(i) - I(j)\|_2 |l(i) - l(j)| \qquad (12)$$

This term encourage coherence in regions with similar appearance.

An energy minimization solver - graph cut[3] is applied to optimize $D(\hat{L})$ to obtain the refined pedestrian segmentation(as shown in Fig. 6(b)). As an initialization of graph cut, the pixels in the foreground region of the trimap are labeled as $l(i) = 1$; the background pixels are labeled as $l(i) = 0$; the undefined pixels as $l(i) = 0$ if $D_d(l(i) = 0) > D_d(l(i) = 1)$, or $l(i) = 1$ if $D_d(l(i) = 0) \leq D_d(l(i) = 1)$.

To further refine the foreground boundary, we invoke the Bayesian matting [22] for soft-segmenting an eroded narrow region along the boundary(as shown in Fig. 6(c)(d)).

VI. Experiments

Experimental detail

In the learning phase, we collect 423 pedestrian shape templates from Fudan-Penn pedestrian set [23] for training. All the templates are resized to 320 pixels in height and manually labeled on the joints. We utilize these templates for two purposes: constructing a hierarchical shape tree and learning part detectors. To construct the hierarchical shape tree, we set the parameters as: $K_1 = K_2 = 4$ and $K_3 = 10$ (as described in Section IV-B), resulting in a 5-level tree. To learn the part (head or torso) detector, we first extract the part masks automatically based on clicked joints, then independently train the head and torso detector(as described in Section III-B). In all our experiments, the other parameters are set as: $M_L = 24$, $M_S = 10$, $T_s = 1.0$, T_1=0.8, T_2=2.0, $\tau_s = 8.0$, $\tau_1 = \tau_1 = 0.8$, $\tau_3 = 16$, $\alpha_1 = 0.1$, $\alpha_2 = 1.0$, $\lambda = 0.5$, $K_G = 5$.

Experimental results

Our approach extends shape matching [1] by exploiting the ability of the smoother Chamber matching and appearance consistency. To evaluate our approach's improvement, we quantitatively compare it with the original shape matching methods [1] in form of $F\_measure$ [24]. $F\_measure = 2 * precision * recall/(precision + recall)$, where $precision$ is defined as the ratio of the true positive pixels (i.e., the pixels labeled as foreground actually belong to foreground) to the all labeled foreground pixels, and $recall$ is defined as the ratio of the true positive pixels to the ground truth pixels. The test samples are the 212 pedestrian images of Fudan-Penn pedestrian set. The original shape matching method achieves the average accuracies of 82.1%. Using our approach, the accuracies are improved to 86.7%.

Two state-of-the-art pedestrian segmentation algorithms, Lin et al. [8] and Wu et al. [7], are also compared here. They both extract pedestrian silhouette using the local shapelets. Although the local shapelets are more flexible than global shape template, their methods still have three limitations: 1) constraint to frontal/rear view pedestrians; 2) ignoring the segmentation of arms; 3) the ambiguous delineation of local contour. The second rows of Fig. 7 and Fig. 8 show some examples of the segmentation results derived from their algorithms. In the third rows of Fig. 7 and Fig. 8, we demonstrate the inferred segmentation of our approach. As we can see, our pedestrian extraction gives more accurate delineation of pedestrian silhouette. The pedestrian arms are also segmented. In addition, our algorithm can be applied to segmenting side profile view pedestrians, as shown in Fig. 9.

Computational Cost

Our experiments are implemented on a 2.2GHz 32-bit Pentium PC with some employed functions, including graph cut [3], random forest [18], and Bayesian matting [22]. In the training procedure, the construction of the hierarchical tree needs about 6 hours, and the learning for two part detectors (each with 150 masks) takes about 1 hour.

Figure 7. Example of segmentation results. The first row shows the input images from CAVIAR and Zurich dataset. The second row displays the segmentation results of Wu et al. [7], in which the extract silhouettes are displayed in green. The third row shows our results.

Figure 8. Example of segmentation results. The first row shows the input images from the INRIA dataset. The second row shows the cropped segmentation results of Lin et al. [8], in which the extract silhouettes are displayed in green. The third row shows our results.

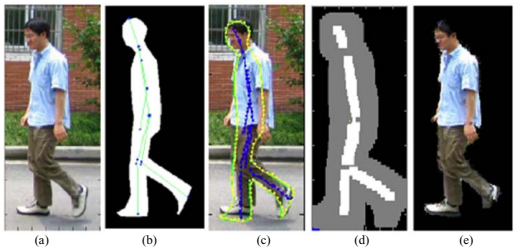


Figure 9. Example of side view pedestrian segmentation. (a) Input image; (b) The matched template with the joints and skeleton; (c) The aligned contour as well as the skeleton on the original image; (d) The generated trimap; (e) The final segmentation result.

To lock onto a pedestrian head in a 260*320 image, the head-torso parsing takes about 7s. Shape matching with a template is performed in 0.15s. As the hierarchical tree reduces the number of matching templates from 423 to 25, the initial silhouette extraction in a fixed position is correspondingly reduced from 63.4s to 3.7s. That is, in the first two stages of pedestrian segmentation, the computational time is 10.7s. The third stage for pedestrian refinement takes about 10s. Thus, the total time is about 20.7s.

VII. Conclusions

In this paper we propose an approach combining shape and appearance cues for automatic pedestrian segmentation in single image. The major contribution of this approach is the utilization of automatically generated trimap to encode the skeleton and shape information for pedestrian segmentation. The skeleton is extracted based on shape matching with a set of shape templates. For speed-up, we organize the template set in a hierarchical tree to reduce matching exemplars and develop a quick head-torso parsing method to lock onto pedestrian to reduce search space. To extract smoother silhouettes, a novel shape matching method is presented, which is more tolerant to local deformation than Chamber matching under cluttered backgrounds.

Although our approach can handle the majority of standing pedestrian segmentation, some misaligned pixels still exist due to the faint figure-ground differences. Future work will consider improving it more robust to cluttered scene. To further speed up implementation, more efficient optimization solutions for pedestrian refinement should be adopted and some processes could be re-designed for implementation in parallel graphics hardware, including pedestrian localization and template matching. Another direction is to extend it for video pedestrian segmentation by incorporating the motion cue into this framework.

Acknowledgment

This research was supported by the National 973 Program of China (No.2009CB320805), the Natural Science Foundation of China (No.61073070), and Fundamental Research Funds for the Central Universities of China.

References

[1] D. M. Gavrila, "A Bayesian exemplar-based approach to hierarchical shape matching," *IEEE Trans Pattern Anal Mach Intell*, vol. 29, 2007, pp. 1408-1421, doi:10.1109/TPAMI.2007.1062.

[2] C. Rother, V. Kolmogorov, and A. Blake, "'Grabcut': interactive foreground extraction using iterated graph cuts", *ACM Trans Graph*, vol. 23, 2004, pp. 309-314, doi:10.1145/1015706.1015720.

[3] Y. Y. Boykov and M. Jolly, "Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images", *IEEE International Conference on Computer Vision*, 2001, doi:10.1109/ICCV.2001.937505.

[4] X. Wang, T. X. Han, and S. Yan, "An HOG-LBP human detector with partial occlusion handling", *IEEE International Conference on Computer Vision*, 2009, doi:10.1109/ICCV.2009.5459207.

[5] L. Alvarez, L. Baumela, N. P. Marquez, and P. Henriquez, "Morphological snakes", *IEEE Proc. Conf. on Computer Vision and Pattern Recognition*, 2010, doi:10.1109/CVPR.2010.5539900.

[6] A. Opelt, A. Pinz and A. Zisserman, "A boundary-fragment-model for object detection", *European Conference on Computer Vision*, 2006.

[7] B. Wu and R. Nevatia, "Detection and segmentation of multiple, partially occluded objects by grouping, merging, assigning part detection responses", *Int J Comput Vis*, vol. 82, 2009, pp. 185-204, doi:10.1007/s11263-008-0194-9.

[8] Z. Lin and L. S. Davis, "A pose-invariant descriptor for human detection and segmentation", *European Conference on Computer Vision*, 2008, doi:10.1007/978-3-540-88693-8_31.

[9] B. Leibe, E. Seemann, and B. Schiele, "Pedestrian detection in crowded scenes", *IEEE Proc. Conf. on Computer Vision and Pattern Recognition*, 2005, doi: 10.1109/CVPR.2005.272.

[10] D. Larlus and F. E. Jurie, "Combining appearance models and markov random fields for category level object segmentation", *IEEE Proc. Conf. on Computer Vision and Pattern Recognition*, 2008, doi:10.1109/CVPR.2008.4587453.

[11] J. Wang and M. Cohen, "Image and video matting: a survey", *Foundations and Trends in Computer Graphics and Vision*, vol. 3, 2007, pp. 1-78, doi:10.1561/0600000019.

[12] E. Borenstein and J. Malik, "Shape guided object segmentation", *IEEE Proc. Conf. on Computer Vision and Pattern Recognition*, 2006, doi:10.1109/CVPR.2006.276.

[13] T. Cour, J. Shi, "Recognizing objects by piecing together the segmentation puzzle", *IEEE Proc. Conf. on Computer Vision and Pattern Recognition*, 2007, doi:10.1109/CVPR.2007.383051.

[14] M. P. Kumar, P. Torr, and A. Zisserman, "OBJ CUT", *IEEE Proc. Conf. on Computer Vision and Pattern Recognition*, 2005, doi:10.1109/CVPR.2005.249.

[15] M. Eichner and V. Ferrari, "Better appearance models for pictorial structures", *British Machine Vision Conference*, 2009.

[16] P. Guan, A. Weiss, A. Alan, and M. J. Black, "Estimating human shape and pose from a single image", *IEEE International Conference on Computer Vision*, 2009, doi:10.1109/ICCV.2009.5459300.

[17] S. Belongie, J. Malik, and J. Puzicha, "Shape context: a new descriptor for shape matching and object recognition", *Advances in Neural Information Processing Systems*, 2000.

[18] B. Leo, "Random Forests", *Machine Learning*, vol. 45, 2001, pp. 5-32, doi:10.1023/A:1010933404324.

[19] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient belief propagation for early vision", *Int J Comput Vis*, vol. 70, 2004, pp. 41-54, doi:10.1007/s11263-006-7899-4.

[20] D. Martin, C. Fowlkes, and J. Malik, "Learning to detect natural image boundaries using local brightness, color and texture cues", *IEEE Trans Pattern Anal Mach Intell*, vol. 26, 2004, pp. 530-549, doi:10.1109/TPAMI.2004.1273918.

[21] J. Shi and J. Malik, "Normalized cuts and image segmentation", *IEEE Trans Pattern Anal Mach Intell*, vol. 22, 2000, pp. 888-905, doi:10.1109/34.868688.

[22] Y. Chuang, B. Curless, D. Salesin, and R. Szeliski, "A Bayesian approach to digital matting", *IEEE Proc. Conf. on Computer Vision and Pattern Recognition*, 2001, doi:10.1109/CVPR.2001.990970.

[23] L. Wang, J. Shi, G. Song and I. F. Shen, "Object Detection Combining Recognition and Segmentation", *Asian Conference on Computer Vision*, 2007.

[24] C. Rijsbergen, "Information Retreival", 1979, available from http://www.dcs.gla.ac.uk/Keith/Preface.html