Yu *et al. IPSJ Transactions on Computer Vision and Applications* (2019) 11:10
https://doi.org/10.1186/s41074-019-0062-2

IPSJ Transactions on Computer Vision and Applications

RESEARCH PAPER Open Access

Pedestrian segmentation based on a spatio-temporally consistent graph-cut with optimal transport

Yang Yu*, Yasushi Makihara and Yasushi Yagi

Abstract

We address a method of pedestrian segmentation in a video in a spatio-temporally consistent way. For this purpose, given a bounding box sequence of each pedestrian obtained by a conventional pedestrian detector and tracker, we construct a spatio-temporal graph on a video and segment each pedestrian on the basis of a well-established graph-cut segmentation framework. More specifically, we consider three terms as an energy function for the graph-cut segmentation: (1) a data term, (2) a spatial pairwise term, and (3) a temporal pairwise term. To maintain better temporal consistency of segmentation even under relatively large motions, we introduce a transportation minimization framework that provides a temporal correspondence. Moreover, we introduce the edge-sticky superpixel to maintain the spatial consistency of object boundaries. In experiments, we demonstrate that the proposed method improves segmentation accuracy indices, such as the average and weighted intersection of union on TUD datasets and the PETS2009 dataset at both the instance level and semantic level.

Keywords: Pedestrian segmentation, Edge sticky superpixel, Optimal transport, Conditional random field

1 Introduction

Silhouette extraction or human body segmentation is widely conducted as the first step in many high-level computer vision tasks of video surveillance systems, such as human tracking [1–4], human action recognition [5–8] and gait-based identification and recognition [9–11]. In human tracking, the extracted human silhouette is used for human full-body localization or human part localization [1–4]. In human action recognition, studies [5, 7, 8] have directly extracted features from a silhouette sequence; Chaaraoui et al. [6] used contour points of the human silhouette for action representation. For gait-based identification and verification, Collins et al. [9] used the silhouette for shape matching; Chen et al. [2] extracted features from the spatio-temporal silhouette for gait recognition while Liu et al. [11] proposed the average silhouette as a feature for recognition.

Pedestrian silhouette extraction has long been studied. This research mainly falls into three categories: supervised methods, unsupervised methods, and semisupervised methods.

Supervised methods [12, 13] have performed well in recent years. A typical approach of supervised pedestrian silhouette extraction requires a manually annotated mask of the target in the first frame and propagates the mask frame by frame. An automatic surveillance system, however, cannot adopt manual annotation.

Unsupervised methods, including methods based on background subtraction (e.g., [14, 15]) and motion segmentation (e.g., [16–19]), are the most popular approaches because they do not require manual annotation. Methods based on background subtraction model the background using statistical models (e.g., a Gaussian mixture model) and extract the silhouettes of moving targets as the foreground. However, methods based on background subtraction only classify the moving target and background and do not realize instance-level silhouette extraction. Multi-label motion segmentation assigns human labels to sparse points or pixels according to motion information (e.g., optical flow), allowing targets with different motion patterns to be discriminated.

*Correspondence: yu@am.sanken.osaka-u.ac.jp
[1]The Institute of Scientific and Industrial Research, Osaka University, 8-1 Mihogaoka, Ibaraki, 567-0047 Osaka, Japan

However, because of the lack of object detection information, motion segmentation still cannot discriminate pedestrians with the same motion pattern (e.g., pedestrians walking in the same direction side by side) and may sometimes assign different labels to human parts with different motion patterns. Motion segmentation therefore suffers from under-segmentation and over-segmentation.

Semisupervised methods that do not require a manually annotated silhouette at the first frame but a bounding box trajectory are more suitable for pedestrian silhouette extraction by an automatic surveillance system, because the trajectory of the bounding box can be automatically extracted using recently advanced approaches of object detection [20–22] and multiple-object tracking [23–25]. To the best of our knowledge, semisupervised methods use optical flow to maintain temporal consistency (e.g., [26]). Because optical flow sometimes fails in handling large displacement, optical-flow-based semisupervised approaches often suffer segmentation errors for human parts having large displacement (e.g., a pedestrian's leg and arm). Moreover, a conditional random field (CRF) framework that uses a color-based Gaussian mixture model (GMM) for the background data term and a simple linear iterative clustering (SLIC) superpixel [27] as nodes in the CRF has been adopted [26]. However, color information is not enough for modeling a nonhuman region (e.g., when a pedestrian and the background have similar colors) and the SLIC superpixel sometimes cannot preserve the object boundary well, which is vital for construction of the spatial pairwise term.

We therefore proposed a semisupervised method that not only handles large displacement but also better preserves the pedestrian's boundary. Given the pedestrian bounding box tracklets, we construct a conditional random field for silhouette extraction that involves a data term, spatial pairwise term, and temporal pairwise term. The contributions of this paper are as follows.

- *Optimal transport (OT)-based temporal consistency.* In contrast to most related work, we adopt OT to maintain temporal consistency. The lack of capacity in terms of handling large displacement is a main drawback of optical flow. Although there are methods that improve the handling of large displacement (e.g., the pyramid strategy [28]), the motion of leg and arm parts still cannot be described correctly. Compared with conventional optical flows, the proposed method successfully handles large displacement between two frames thanks to the global optimal property of the OT framework. As far as we know, the OT framework is usually used to measure the difference between two discrete distributions (e.g., a dissimilarity measure between two color histograms), which is also known as the earth mover's distance. The proposed method does not use the final outcome of the OT framework (i.e., a distance) but the "process" of the OT framework (i.e., flow (or correspondence) between two frames), which is the primal novelty of the proposed method.
- *Combination of the edge-sticky superpixel (ESS) and OT.* The time complexity of the OT increases as the dimension of the discrete distributions (e.g., the number of bins of histograms) increases, and direct application of the OT to pixel-wise image representation is computationally prohibited. We therefore need to appropriately transform the input image into a discrete distribution with a relatively low dimension. Superpixel segmentation is one such effective way to represent an image as a discrete distribution while keeping information, that is, compressing redundancy. More specifically, we regard an input image as a histogram, where the number of superpixels is the number of bins, a gravity center of a superpixel is a representative value of a bin, and a number of pixels (area) of a superpixel is the frequency (or vote) for a bin. Moreover, superpixel segmentation needs to well preserve object boundaries for our final goal, that is, pedestrian silhouette extraction. State-of-the-art superpixel segmentation methods (e.g., the SLIC superpixel [27] and superpixels extracted via energy-driven sampling (SEEDS) superpixel [29]) provide a balance between appearance and shape regularity, and usually perform well in computer vision tasks. However, this balance between appearance and shape regularity does not always guarantee that the object boundary is well preserved. Our ultimate target is to extract pedestrians' silhouettes, and we thus need to adopt a superpixel segmentation method that better preserves object boundaries. We therefore adopt the ESS, which introduces edge detection information explicitly into the process of superpixel generation. As a result, the object boundary can be preserved well while balancing the appearance and shape regularity.
- *Performance improvement on segmentation benchmarks.* We demonstrate that the proposed method improves the performance of pedestrian silhouette extraction at both the instance level and semantic level on public datasets compared with state-of-the-art methods.

2 Related work

The silhouette extraction or human segmentation of multiple pedestrians has been addressed in the literature [12, 13, 16, 26, 30–32]. We categorize typical approaches as follows:

- *Supervised methods.* Supervised methods perform well in video segmentation. The most popular frame-

work [12, 13] is to manually annotate the target's mask in the first frame and propagate the target mask to other frames. In [13], a two-branch approach was proposed whereby the features from ResNet-101 [33] and FlowNet [34] were combined for joint object segmentation and optical flow estimation. In [12], a method of frame-by-frame object segmentation was implemented by learning the appearance of the annotated object. However, because the mask annotation has a manual burden, it is difficult to apply supervised methods to pedestrian silhouette extraction in an automatic surveillance system.

- *Unsupervised methods*. Unsupervised methods require no manual annotation and hence can be applied directly to an automatic surveillance system. Most unsupervised methods are based on motion information. The temporal superpixel [35] involves optical flow into a superpixel segmentation framework to realize a temporally consistent superpixel. Ochs et al. [16] adopted a two-step approach: generate sparse segments by clustering long-term trajectories and then obtain dense segments according to color. However, the temporal superpixel is a superpixel segmentation and thus requires a manual annotator that specifies the pedestrian's superpixel, which is again not possible for an automatic surveillance system. Ochs's approach [16] is also prone to under-segmentation because multiple pedestrians walking in the same direction are likely to be segmented into an identical segment.
- *Semisupervised methods*. Compared with supervised and unsupervised methods, semisupervised methods that only require a bounding box annotation are more suitable for silhouette extraction by a real-world surveillance system. Milan [26] exploited a joint tracking and segmentation method that first applies superpixel segmentation and multiple-pedestrian tracking. A CRF is then constructed and all superpixels are assigned with the labels of pedestrian trajectories. Because optical flow is used in the construction of the CRF, Milan's approach sometimes fails for pedestrian's legs, for which there is large spatial displacement.
- *Pedestrian segmentation methods for a single frame*. In recent years, great strides have been made in cellular neural network (CNN)-based image semantic segmentation and instance segmentation. In [31], a multipath refinement network was presented where CNN features with multiple resolutions are fused so that semantic features can be refined using lower-level features. In [32], an object detection network [20] is concatenated by a fully convectional network [36] so that object detection and instance-level segmentation can be achieved jointly. Single-frame segmentation methods can therefore be easily extended to pedestrian silhouette extraction in video using bounding box trajectories.

3 Proposed method

3.1 Problem setting

The present study presents a method of extracting silhouettes of multiple pedestrians from a video. We assume that the cameras are static and the bounding box trajectories are given by well-established detectors [20] and trackers [23].

3.2 Framework

We adopt a two-step framework that consists of superpixel segmentation and superpixel-wise labeling. The whole framework is shown in Fig. 1.

Superpixel segmentation. Given an input image sequence, superpixel segmentation is first applied frame by frame to reduce the computational cost. We adopt the ESS, which better preserves object boundaries.

Superpixel-wise labeling. Given the superpixel segmentation result and pedestrian trajectories (i.e., a bounding box sequence for a pedestrian), each superpixel is assigned with a trajectory label (i.e., a pedestrian label) in this step, resulting in instance-level segmentation as shown in Fig. 1f.

The label assignment problem has been well studied for decades and recent progress expanded its application area to many computer vision tasks. As an example, Wu [37] proposed an adaptive label assignment method to handle the "one example human re-identification" problem where there is only one example available for each human identity, that is, the labeled data. The adaptive label assignment method can both select a set of candidates from the unlabeled data and assign labels of the candidates using a nearest neighbors (NN) classifier in the feature space extracted by the CNN model.

However, in the present work, we cannot generate a set of "labeled data" as in [37] owing to the different problem settings. Furthermore, spatio-temporal consistency is strongly required in the present work, and pairwise features that maintain spatio-temporal consistency (e.g., edge-based features) can only be extracted in a pairwise manner instead of using the independently extracted features. As a result, the approach in [37] cannot be applied directly in the superpixel-wise labeling step of the present work.

To better handle the features extracted in a pairwise manner, we adopt the well-established CRF for superpixel-wise labeling. The label assignment problem is then formulated as a CRF problem and solved using the graph-cut with α-expansion algorithm.

Details are discussed in the following subsections.

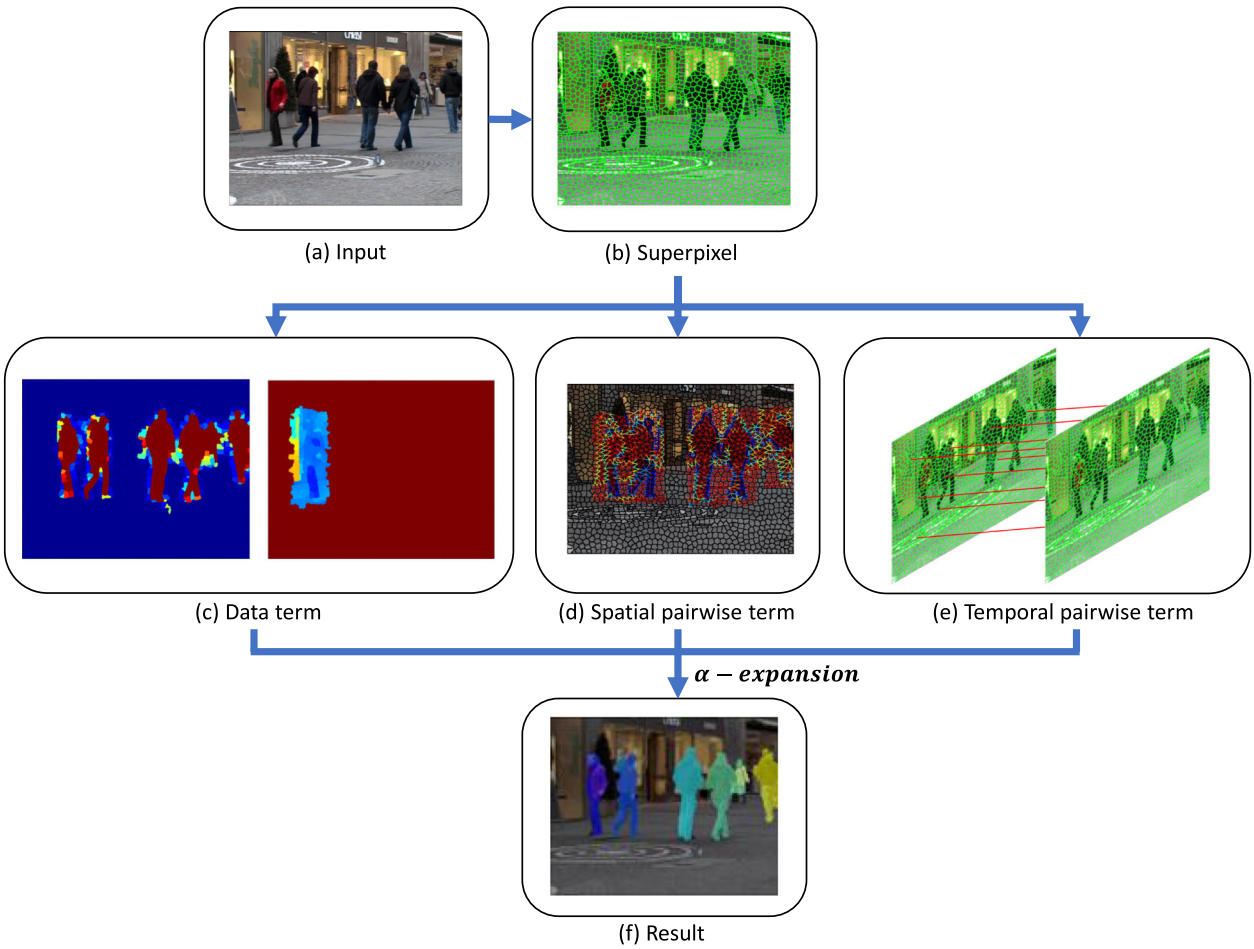

Fig. 1 Framework of the proposed method. **a** Given input images. **b** Superpixel segmentation followed by the construction of a CRF consisting of **c** a data term, **d** spatial pairwise term, and **e** temporal pairwise term. Application of the graph-cut with α-expansion to get **f** the segmentation result

3.3 ESS

The superpixel is a popular technology used to reduce the redundancy of an image and is employed in many computer vision applications. We use the superpixel because not only does it reduce the computational complexity but also it preserves object boundaries.

State-of-the-art approaches (e.g., the SEEDS superpixel [29] and SLIC superpixel [27] approaches) balance the spatial and appearance consistency. However, such balance sometimes affects the capacity to preserve object boundaries. It is therefore necessary to involve edge information when there is a strong need to preserve the object boundary. In this research, we adopt the ESS, which is an extension of Pitor's work [38]. Because there is no corresponding publication[1], we provide a simple illustration of the ESS. We describe the details of the ESS along with Fig. 2 in the following paragraphs.

[1]Code for the ESS is released at https://github.com/pdollar/edges.

We denote a set of pixels in frame t by $\mathcal{P}^t = \{p_i | i \in \mathcal{L}_P^t\}$, where $\mathcal{L}_P^t$ is a set of the indices of pixels in frame t (i.e., the number of elements of $\mathcal{L}_P^t$ is the image size), $t \in \{1, 2, ..., T\}$, where T is the total frame number and p_i is the i-th pixel. Moreover, a set of superpixel indices in frame t is denoted $\mathcal{L}_{\mathrm{SP}}^t$. The superpixel segmentation in frame t can then be formulated as

$$X_{\mathrm{SP}}^t : \mathcal{L}_P^t \rightarrow \mathcal{L}_{\mathrm{SP}}^t, \tag{1}$$

where each pixel is assigned with the label of a super pixel (i.e., the index of a superpixel).

We first initialize each pixel as a superpixel; i.e., $X_{\mathrm{SP}}^t(i) = i; \forall i \in \mathcal{L}_P^t$. Then, for each pixel (e.g., the i-th pixel), we calculate the cost $c(i, l)$ of assigning a neighboring superpixel's label l to the i-th pixel considering the spatial proximity, appearance similarity, edge consistency, and superpixel size as

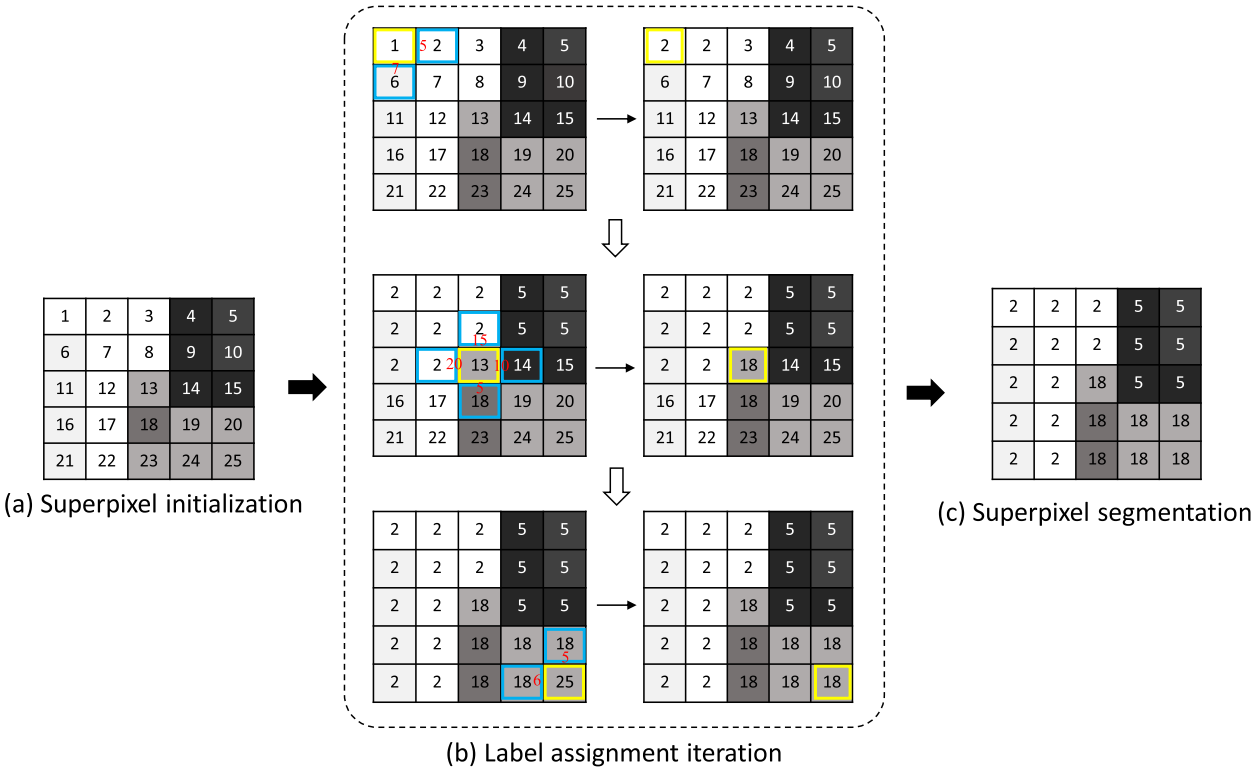

Fig. 2 Framework of the ESS. **a** Each pixel in the input image (e.g., a 5 × 5 grayscale image) initialized as a superpixel, where a black number is the label of a superpixel. **b** Each pixel relabeled under an energy minimization framework. In each iteration, we scan and update the labels of all pixels. For each pixel (yellow), the label assignment costs of its four-connected neighbors (blue) are calculated as shown by red numbers, and each pixel's label is updated with the lowest-cost neighbor's label. The iteration continues until there is no change in each pixel's label. Finally, the superpixel segmentation result is obtained as in **c**

$$\begin{aligned} c(i,l) &= \alpha\|\boldsymbol{\nu}_{\mathrm{loc}}(i) - \boldsymbol{\mu}_{\mathrm{loc}}(l)\|^2 \\ &+ (1-\alpha)(1-\beta)\|\boldsymbol{\nu}_{\mathrm{app}}(i) - \boldsymbol{\mu}_{\mathrm{app}}(l)\|^2 \\ &+ \frac{\gamma\alpha}{A_l} + (1-\alpha)\beta c_{\mathrm{edge}}(i,l), \end{aligned} \tag{2}$$

where α, β, and γ are hyperparameters. The location and appearance vector for the i-th pixel are denoted $\boldsymbol{\nu}_{\mathrm{loc}}(i)$ and $\boldsymbol{\nu}_{\mathrm{app}}(i)$, while the mean location and appearance vector for the l-th superpixel are denoted $\boldsymbol{\mu}_{\mathrm{loc}}(l)$ and $\boldsymbol{\mu}_{\mathrm{app}}(l)$. Moreover, c_{edge} is the edge cost and A_l is the size of the l-th superpixel.

The first and second terms of Eq. (2) maintain the spatial consistency of the superpixel, while the third term controls the size of the superpixel.

The last term helps to preserve the object boundary by involving the edge probability. The edge probability is calculated using structured edge detection (SED) [38]. SED is briefly introduced together with Fig. 3 below.

SED firstly separates an input image into a set of image patches. A pre-trained random forest is then applied to the set of image patches to achieve a set of binary edge masks as shown in Fig. 3b. Finally, the set of edge masks are aggregated to generate the edge probability (i.e., the edge detection result) as shown in Fig. 3c. We refer the reader to [37] for more details.

The edge probability of the i-th pixel in frame t is denoted $p^t_{\mathrm{edge}}(i)$ and the edge cost function $c_{\mathrm{edge}}(i,l)$ is then defined as

$$c_{\mathrm{edge}}(i,l) = \begin{cases} 0 & \{j | j \in n_4(i), X^t_{\mathrm{SP}}(j) \neq l\} = \emptyset \\ \min\limits_{j \in n_4(i), X^t_{\mathrm{SP}}(j) \neq l} -p^t_{\mathrm{edge}}(j) & \text{otherwise} \end{cases}, \tag{3}$$

where the set of four-connected neighbors of the i-th pixel is denoted $n_4(i)$ and the set of corresponding superpixel labels is $l_4(i) = \{X^t_{\mathrm{SP}}(j) | j \in n_4(i)\}$. Details of the edge cost function will be described along with Fig. 4.

Figure 4 shows that the i-th pixel's four-connected neighbors are j_1 (whose superpixel label is l_1) and j_2, j_3, and j_4 (whose superpixel labels are l_2). The edge probability is represented in pseudo-color, where the edge probability for a red pixel is 0.9 while that for a blue pixel is 0.1, i.e., there is an edge on the left side of the i-th pixel. According to Eq. 3, $c_{\mathrm{edge}}(i,l_1) = -0.1$ and $c_{\mathrm{edge}}(i,l_2) = -0.9$, it is more difficult to assign the label

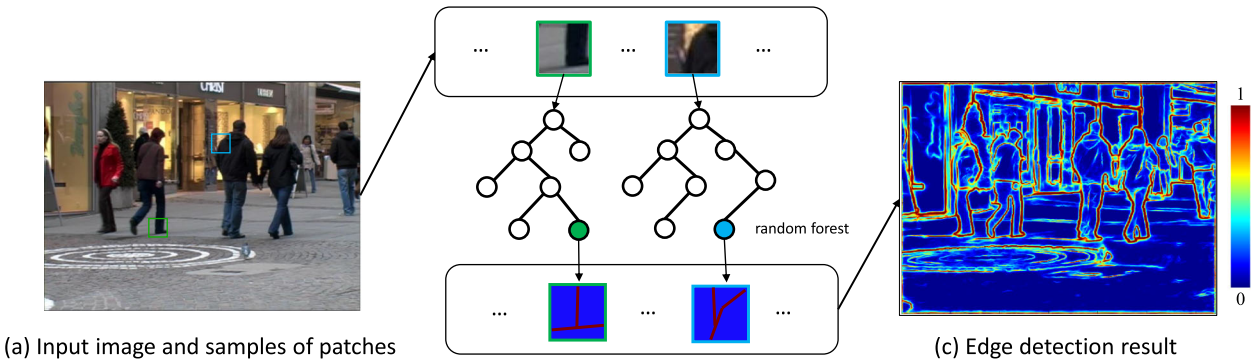

Fig. 3 Framework of the proposed method. **a** Given input images and two samples of image patches (blue and green edged). **b** Binary edge masks obtained using a pre-trained random forest. **c** Edge detection results (i.e., edge probability map) obtained by aggregating all edge masks

l_1 than the label l_2 to the i-th pixel. As a result, the edge cost function helps preserve the object boundary.

We repeat this process until X_{SP}^t stops changing. An example of an ESS result is shown in Fig. 5. We see that the object boundaries (e.g., boundaries between a pedestrian and background) are well preserved.

After obtaining the superpixels for each frame independently, the set of all superpixel labels is defined as $\mathcal{L}_{\mathrm{SP}} = \bigcup_{t=1}^{T} \mathcal{L}_{\mathrm{SP}}^t$. Moreover, we denote the set of all pixels as $\mathcal{L}_{\mathrm{P}} = \bigcup_{t=1}^{T} \mathcal{L}_{\mathrm{P}}^t$. For simplicity, the superpixel segmentation for all frames is defined as

$$X_{\mathrm{SP}} : \mathcal{L}_P \rightarrow \mathcal{L}_{\mathrm{SP}}. \tag{4}$$

3.4 Superpixel-wise labeling

Given superpixel segmentation results and a set of bounding box sequences for n_{TR} pedestrians $\mathrm{TR} = \{tr_i | i \in \mathcal{L}_{\mathrm{TR}}\}$, where tr_i is the bounding box trajectory for the i-th pedestrian, we consider mapping the superpixel labels $\mathcal{L}_{\mathrm{SP}}$ into one of the pedestrian labels $\mathcal{L}_{\mathrm{TR}} = \{l_1^{\mathrm{TR}}, ..., l_{n_{\mathrm{TR}}}^{\mathrm{TR}}\}$, where l_m^{TR} is the m-th pedestrian's label, or a background label $l_{\mathrm{BG}}^{\mathrm{TR}}$. For simplicity, we denote all labels by $\hat{\mathcal{L}}_{\mathrm{TR}} = \mathcal{L}_{\mathrm{TR}} \bigcup \{l_{\mathrm{BG}}^{\mathrm{TR}}\}$. The problem of mapping from superpixels' labels $\mathcal{L}_{\mathrm{SP}}$ to $\hat{\mathcal{L}}_{\mathrm{TR}}$ (i.e., the superpixel-wise labeling problem) can be formulated as

$$X_{\mathrm{CRF}} : \mathcal{L}_{\mathrm{SP}} \rightarrow \hat{\mathcal{L}}_{\mathrm{TR}}. \tag{5}$$

We then formulate the problem of optimizing X_{CRF} as a multi-label CRF problem:

$$X_{\mathrm{CRF}}^* = \arg\min_{X_{\mathrm{CRF}}} E(X_{\mathrm{CRF}}), \tag{6}$$

where the energy function $E(X_{\mathrm{CRF}})$ is defined as

$$\begin{aligned} E(X_{\mathrm{CRF}}) = & \sum_{p \in \mathcal{L}_{\mathrm{SP}}} E_{\mathrm{Data}}\left(p, X_{\mathrm{CRF}}(p)\right) \\ & + \omega_{\mathrm{S}} \sum_{(p,q) \in \mathcal{N}_{\mathrm{S}}} E_{\mathrm{S}}\left(p, q, X_{\mathrm{CRF}}(p), X_{\mathrm{CRF}}(q)\right) \\ & + \omega_{\mathrm{T}} \sum_{(p,q) \in \mathcal{N}_{\mathrm{T}}} E_{\mathrm{T}}\left(p, q, X_{\mathrm{CRF}}(p), X_{\mathrm{CRF}}(q)\right). \end{aligned} \tag{7}$$

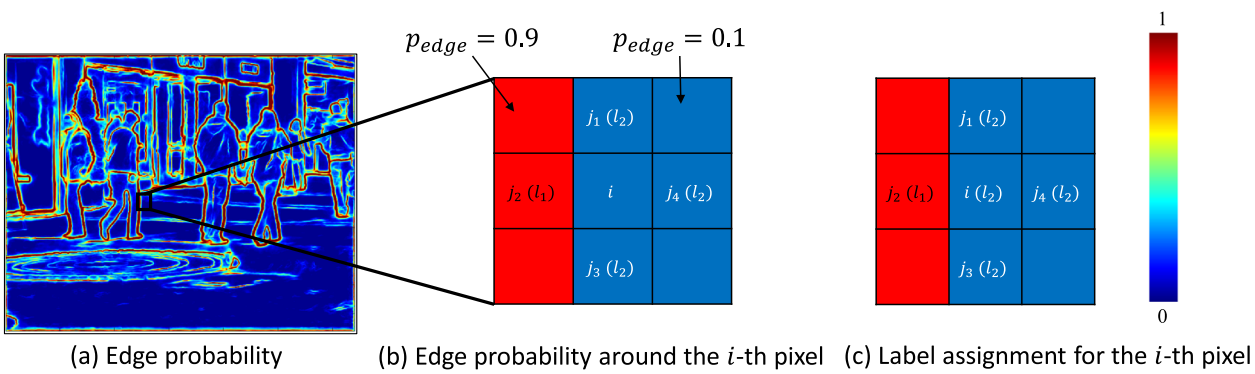

Fig. 4 Example of the edge cost function. **a** Input image of the frame t. **b** Clipping around the i-th pixel. Edge probability $p_{\mathrm{edge}}^t = 0.9$ on the left side (as represented by red) and $p_{\mathrm{edge}}^t = 0.9$ in the middle and on the right side (as represented by blue). **c** Edge cost of assigning the label l_1 to the i-th pixel $c_{\mathrm{edge}}(i, l_1) = -0.1$ while $c_{\mathrm{edge}}(i, l_2) = -0.9$; therefore, l_2 is more likely to be assigned to the i-th pixel

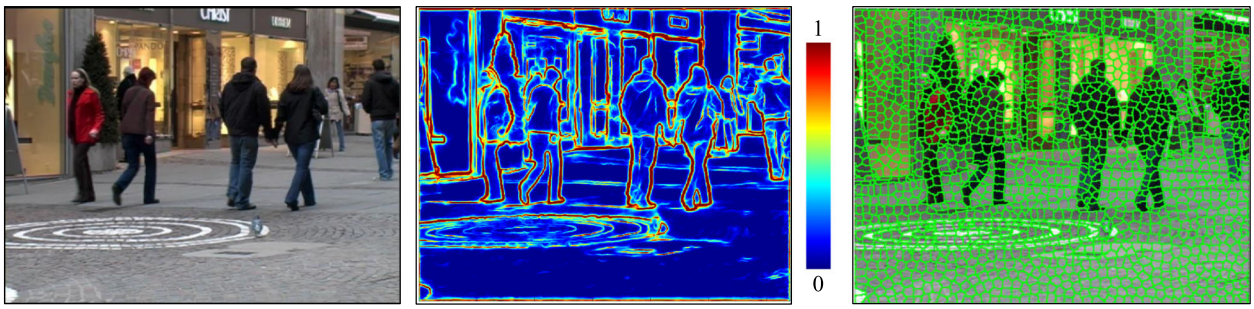

Fig. 5 Example of the ESS. **a** Input image. **b** Edge probability map (represented by pseudo-color). **c** ESS result. The pedestrian's boundary is well preserved by the ESS

Here, the first term is the data term while the second and third terms are respectively spatial and temporal pairwise terms. ω_{S} and ω_{T} are respectively the weights of spatial and temporal pairwise terms. The definitions of $\mathcal{N}_{\mathrm{S}}$, $\mathcal{N}_{\mathrm{T}}$, E_{Data}, E_{S}, and E_{T} are explained in the following sections.

The multi-label CRF problem can then be solved using the graph-cut with α-expansion algorithm [39], which is widely used for CRF inference. The algorithm iterates each possible label (i.e., the label α in a given CRF), and in each iteration, the algorithm segments the α and the non-α components with the graph-cut. The energy function of the CRF in this work contains spatial and temporal pairwise terms, and the graph-cut with α-expansion algorithm is thus adopted in a spatio-temporally consistent way.

3.4.1 Data term

The data term defined as

$$\sum_{p\in\mathcal{L}_{\mathrm{SP}}} E_{\mathrm{Data}}\left(p, X_{\mathrm{CRF}}(p)\right) \tag{8}$$

contains two components, namely a pedestrian term $E_{\mathrm{Data}}\left(p, X_{\mathrm{CRF}}(p) \neq l_{\mathrm{BG}}^{\mathrm{TR}}\right)$ and background term $E_{\mathrm{Data}}\left(p, X_{\mathrm{CRF}}(p) = l_{\mathrm{BG}}^{\mathrm{TR}}\right)$ for an arbitrary superpixel p.

We use RefineNet [31], a CNN-based semantic segmentation method, for the background term. Given an input image, RefineNet predicts the pixel-wise probability distribution of a set of object classes. In this work, we adopt a pre-trained model on the Cityscapes dataset [40] using Residual Net (ResNet) [33], which contains 20 object classes. We extract the probability of the label "person" in the input image denoted $p_{Hm}(i)$ for the i-th pixel. The pixel-wise human score of the i-th pixel is then defined as

$$h'_{\mathrm{Hm}}(i) = -\log\left(1 - p_{\mathrm{Hm}}(i)\right). \tag{9}$$

The superpixel-wise human score of the p-th superpixel is defined as the mean pixel-wise human score of the pixels inside the p-th superpixel, which is denoted $h_{\mathrm{Hm}}(p)$. An example of the pixel-wise and superpixel-wise human score map is shown in Fig. 6. It is clear that the superpixel-wise human score map can be directly used as the background data term:

$$E_{\mathrm{Data}}\left(p, X_{\mathrm{CRF}}(p) = l_{\mathrm{BG}}^{\mathrm{TR}}\right) = h_{\mathrm{Hm}}(p). \tag{10}$$

We subsequently sample and train a GMM for multiple pedestrians to define the pedestrian term. We denote a set of pixels belonging to the k-th superpixel as $u_k = \{i | X_{\mathrm{SP}}(i) = k\}$ and pixels inside the bounding box trajectory of the i-th pedestrian t_i as $\mathcal{U}_i$. If the k-th superpixel overlaps with the bounding box sequence of the i-th

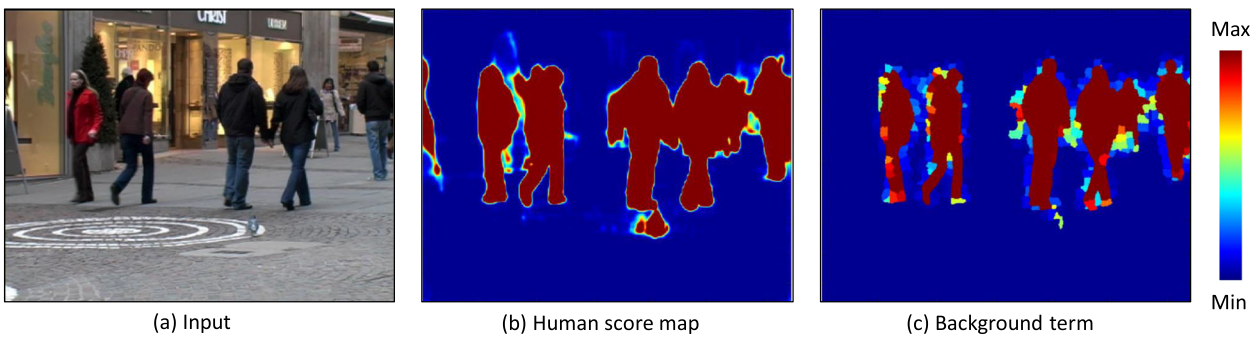

Fig. 6 Example of the background term. **a** Input image. **b** Human score map from RefineNet. **c** Background term

Fig. 7 Example of the pedestrian term. **a** Input image and pedestrian's bounding box. **b** Pedestrian term of the pedestrian inside the bounding box. Outside the bounding box, the pedestrian term is set as a sufficiently large constant

pedestrian t_i (i.e., $u_k \bigcap \mathcal{U}_i \neq \emptyset$), it is sampled for the GMM training of the i-th pedestrian. A superpixel may sometimes overlap with multiple trajectories and we thus adopt a winner-takes-all strategy by which the pedestrian closest to the camera (i.e., the pedestrian with the lowest bound of the bounding box) takes the superpixel.

After the superpixel sampling, we train the GMM for each trajectory according to the mean color of the superpixel. θ_i denotes the GMM parameters of the i-th pedestrian. Moreover, we hypothesize that all superpixels outside the bounding box t_i are hard to be assigned with pedestrian label l_i^{TR}; therefore, the pedestrian term for those superpixels is set with a sufficiently large constant. Finally, the pedestrian term is defined as

$$E_{\mathrm{Data}}\left(p, X_{\mathrm{CRF}}(p) = l_i^{\mathrm{TR}}\right) = \begin{cases} C & u_p \cup \mathcal{U}_i = \emptyset \\ -\log\left(p_{\mathrm{GMM}}\left(\boldsymbol{\mu}_{\mathrm{app}}(p); \theta_i\right)\right) & \text{otherwise} \end{cases}, \tag{11}$$

where C is a sufficiently large constant and $p_{\mathrm{GMM}}\left(\boldsymbol{\mu}_{\mathrm{app}}(p); \theta_i\right)$ is the probability density of the mean appearance $\boldsymbol{\mu}_{app}(p)$ of the p-th superpixel for the i-th pedestrian. An example of the pedestrian term is shown in Fig. 7

3.4.2 *Spatial pairwise term*

The spatial pairwise term

$$\sum_{(p,q)\in\mathcal{N}_{\mathrm{S}}} E_{\mathrm{S}}\left(p, q, X_{\mathrm{CRF}}(p) X_{\mathrm{CRF}}(q)\right) \tag{12}$$

is used to maintain the spatial consistency of X_{CRF}. A set of spatial neighbors $\mathcal{N}_{\mathrm{S}}$ is first defined as

$$\mathcal{N}_{\mathrm{S}} = \left\{(p,q) | p \in \mathcal{L}_{\mathrm{SP}}, q \in \mathcal{L}_{\mathrm{SP}}, \mathrm{conn}_{\mathrm{S}}(p,q) = 1\right\}, \tag{13}$$

where $\mathrm{conn}_{\mathrm{S}}(p,q)$ is the spatial connectivity function and is defined as

$$\mathrm{conn}_{\mathrm{S}}(p,q) = \begin{cases} 1 & \exists i, j, t; X_{\mathrm{SP}}^t(i) = p, X_{\mathrm{SP}}^t(j) = q, \text{i, j are four-connected neighbors} \\ 0 & \text{otherwise} \end{cases}. \tag{14}$$

We then use the color and edge probability to formulate the spatial pairwise energy function E_{S}.

Fig. 8 Example of color-based pairwise energy. **a** Input image. **b** Color-based pairwise energy. If the colors between pedestrians or between a pedestrian and the background are similar, the color-based pairwise energy fails to preserve the object's boundary; e.g., the pedestrian's boundary inside the white bounding box in **b**

(a) Input (b) Edge-based spatial pairwise energy

Fig. 9 Example of edge-based pairwise energy. **a** Input image. **b** Edge-based spatial pairwise energy. The pedestrian's boundary in the bounding box in **b** is better preserved than the same region in Fig. 8

A color-based pairwise energy function is defined as

$$E_{\mathrm{S}}^{\mathrm{Color}}(p,q,X_{\mathrm{CRF}}(p),X_{\mathrm{CRF}}(q)) = \begin{cases} 0 & X_{\mathrm{CRF}}(p) = X_{\mathrm{CRF}}(q) \\ \exp\left(-\lambda\|\boldsymbol{\mu}_{\mathrm{app}}(p) - \boldsymbol{\mu}_{\mathrm{app}}(q)\|^2\right) & \text{otherwise} \end{cases}, \tag{15}$$

and following previous work [41], a parameter λ is subsequently defined as

$$\lambda = \frac{2}{|\mathcal{N}_{\mathrm{S}}|}\sum_{(p,q)\in\mathcal{N}_{\mathrm{S}}}\|\boldsymbol{\mu}_{\mathrm{app}}(p) - \boldsymbol{\mu}_{\mathrm{app}}(q)\|^2 \tag{16}$$

to adapt to high and low color contrast. An example of color-based pairwise energy is shown in Fig. 8.

The color-based pairwise energy function may sometimes fail to maintain spatial consistency when the colors of different pedestrians or a pedestrian and the background are similar as shown in the white bounding box in Fig. 8. We therefore further include the edge probability in the spatial pairwise energy function.

We denote by $p_{\mathrm{edge}}(j)$ the edge probability at the j-th pixel. An edge-based pairwise energy function is subsequently defined as

$$E_{\mathrm{S}}^{\mathrm{Edge}}(p,q,X_{\mathrm{CRF}}(p),X_{\mathrm{CRF}}(q)) = \begin{cases} 0 & X_{\mathrm{CRF}}(p) = X_{\mathrm{CRF}}(q) \\ < 1 - p_{\mathrm{edge}}(j) >_{\mathrm{p,q}} & \text{otherwise} \end{cases}, \tag{17}$$

where $p, q \in \mathcal{L}_{\mathrm{SP}}$ and $< \cdot >_{\mathrm{p,q}}$ denote the expectation over the pixels on the boundary between two spatially neighboring superpixels p and q. An example of the edge-based pairwise energy function is shown in Fig. 9. The boundary of the pedestrian inside the white bounding box in Fig. 9 is better preserved than that in Fig. 8.

Finally, the spatial pairwise energy is defined as the weighted sum of color-based and edge-based pairwise energy:

$$\begin{aligned} E_{\mathrm{S}}(p,q,X_{\mathrm{CRF}}(p),X_{\mathrm{CRF}}(q)) = (1-\omega_e)E_{\mathrm{S}}^{\mathrm{Color}}(p,q,X_{\mathrm{CRF}}(p),X_{\mathrm{CRF}}(q)) \\ + \omega_e E_{\mathrm{S}}^{\mathrm{Edge}}(p,q,X_{\mathrm{CRF}}(p),X_{\mathrm{CRF}}(q)), \end{aligned} \tag{18}$$

where ω_e is a hyperparameter that controls the weight of edge-based spatial pairwise energy.

(a) Manually selected superpixels in frame t (b) Connected superpixel in frame t+1

Fig. 10 Example of OT-based temporal connectivity. **a** Manually selected pedestrian's superpixel in frame t. **b** OT-based temporal connected superpixels in frame $t + 1$

3.4.3 OT-based temporal pairwise term

A temporal pairwise term defined as

$$\sum_{(p,q)\in\mathcal{N}_{\mathrm{T}}} E_{\mathrm{T}}\left(p,q,X_{\mathrm{CRF}}(p),X_{\mathrm{CRF}}(q)\right) \tag{19}$$

is introduced for the purpose of maintaining the temporal consistency of superpixel-wise labeling. $\mathcal{N}_{\mathrm{T}}$ is a set of temporal neighbors defined as

$$\mathcal{N}_{\mathrm{T}} = \left\{(p,q)|p\in\mathcal{L}_{\mathrm{SP}}, q\in\mathcal{L}_{\mathrm{SP}}, \mathrm{conn}_{\mathrm{T}}(p,q)=1\right\}, \tag{20}$$

where $\mathrm{conn}_{\mathrm{T}}$ is the temporal connectivity function.

Different from spatial connectivity, which can be easily defined according to the pixel lattice structure, the temporal connectivity must involve object motion information. To the best of our knowledge, optical flow is the most popular motion information used to define temporal connectivity. However, optical flow usually fails to handle the large displacement that often occurs for the pedestrian leg and arm. We therefore introduce OT-based temporal connectivity for better motion estimation.

The OT distance, also known as the earth mover's distance, is a useful distance with which to compare two probability distributions. The OT problem is described as follows.

Given two probability distributions $\boldsymbol{r}=[r_1,...,r_m]^T$ and $\boldsymbol{c}=[c_1,...,c_n]^T$ and a cost matrix $M\in\mathbb{R}_+^{m\times n}$, the OT problem is to find a transportation matrix P^* such that

$$P^* = \arg\min_{P\in\mathcal{P}(\boldsymbol{r},\boldsymbol{c})} \langle P,M\rangle_{\mathrm{F}}, \tag{21}$$

where $\langle\cdot,\cdot\rangle_{\mathrm{F}}$ denotes the Frobenius dot product. $\mathcal{P}(r,c)=\left\{P\in\mathbb{R}_+^{m\times n}|P\mathbf{1}_n=\boldsymbol{r}, P^T\mathbf{1}_m=\boldsymbol{c}\right\}$, where $\mathbf{1}_m$ and $\mathbf{1}_n$ are m- and n-dimensional vectors of ones.

In this study, we formulate motion estimation as an OT problem. We denote superpixel labels in frame t by $\mathcal{L}_{\mathrm{SP}}^t=\left\{l_1^t,...,l_{|\mathcal{L}_{SP}^t|}^t\right\}$ and then define a superpixel size vector in frame t as $\hat{\boldsymbol{A}}^t=\left[A_1^t,...A_{|\mathcal{L}_{SP}^t|}^t\right]$, where A_i^t is the size of the l_i^t-th superpixel. The normalized size vector is then defined as $\boldsymbol{A}^t=\hat{\boldsymbol{A}}^t/|\mathcal{L}_P^t|$. Because $||\boldsymbol{A}^t||=1$ and $\forall i\in\left\{1,...,|\mathcal{L}_{SP}^t|\right\}, \boldsymbol{A}^t(i)\geq 0$, $\boldsymbol{A}^t$ is a probability distribution. We therefore treat the normalized size vector in two consecutive frames $\boldsymbol{A}^t$ and $\boldsymbol{A}^{t+1}$ as the input of an OT problem.

Moreover, the cost matrix between frames t and $t+1$ is defined as

$$M_{t,t+1}=\left\{m(i,j)|1\leq i\leq|\mathcal{L}_{\mathrm{SP}}^t|, 1\leq j\leq|\mathcal{L}_{\mathrm{SP}}^{t+1}|\right\}, \tag{22}$$

where $m(i,j)$ is defined as

$$\begin{aligned} m(i,j) &= \|\boldsymbol{\mu}_{\mathrm{loc}}(l_i^t)-\boldsymbol{\mu}_{\mathrm{loc}}\left(l_j^{t+1}\right)\|^2 \\ &+ \eta_{\mathrm{app}}\|\boldsymbol{\mu}_{\mathrm{app}}(l_i^t)-\boldsymbol{\mu}_{\mathrm{app}}\left(l_j^{t+1}\right)\|^2 \\ &+ \eta_{\mathrm{Hm}}\left(h_{\mathrm{Hm}}(l_i^t)-h_{\mathrm{Hm}}\left(l_j^{t+1}\right)\right)^2. \end{aligned} \tag{23}$$

The first item of $m(i,j)$ encourages transportation between spatially nearer superpixels while the second term encourages transportation between superpixels that appear similar. Furthermore, we include the third term to encourage transportation between superpixels in the pedestrian region.

The OT between frames t and $t+1$ is defined as

$$P_{t,t+1}^* = \arg\min_{P\in\mathcal{P}(\boldsymbol{A}^t,\boldsymbol{A}^{t+1})} \langle P,M_{t,t+1}\rangle_{\mathrm{F}}. \tag{24}$$

Subsequently, the temporal connectivity is defined as

$$\mathrm{conn}_{\mathrm{T}}(p,q)=\begin{cases} 1 & \exists i,j,t; p=l_i^t, q=l_j^{t+1}, P_{t,t+1}^*(i,j)\geq \mathrm{th}_{\mathrm{temp}} \\ 0 & \text{otherwise}\end{cases}, \tag{25}$$

where $\mathrm{th}_{\mathrm{temp}}$ is the threshold of temporal connectivity, $l_i^t\in\mathcal{L}_{\mathrm{SP}}^t$ and $l_j^{t+1}\in\mathcal{L}_{\mathrm{SP}}^{t+1}$. An example of OT-based temporal connectivity is shown in Fig. 10. In Fig. 10a, we manually select the superpixels belonging to a pedestrian in frame t. All the temporally connected superpixels are shown in Fig. 10b. The temporal consistency is well preserved by the OT-based temporal connectivity.

Finally, the temporal pairwise energy is defined as

$$\begin{aligned} &E_{\mathrm{T}}\left(p,q,X_{\mathrm{CRF}}(p),X_{\mathrm{CRF}}(q)\right) \\ &=\begin{cases} 0 & X_{\mathrm{CRF}}(p)=X_{\mathrm{CRF}}(q) \\ \exp\left(-\lambda\|\boldsymbol{\mu}_{\mathrm{app}}(p)-\boldsymbol{\mu}_{\mathrm{app}}(q)\|^2\right) & \text{otherwise}\end{cases}, \end{aligned} \tag{26}$$

where the definition of λ is the same as in Eq. (16).

4 Experiments

4.1 Experimental setting

4.1.1 Datasets

We test our proposed method on four publicly available image sequences: TUD-Stadtmitte, TUD-Campus, TUD-Crossing and PETS2009 S2L1. Each sequence contains a long-term occlusion that makes segmentation highly challenging. Furthermore, TUD-Stadtmitte and TUD-Campus present the challenges of low contrast and similar clothing.

We use manually annotated pedestrian bounding box trajectories for each dataset when we test the proposed method as well as the other baseline methods. We also annotate ground-truth pedestrian silhouettes (instance segmentation) for the evaluation of pedestrian silhouette extraction.

4.1.2 Evaluation metrics

For the instance-level evaluation, we adopt mean and weighted intersections over union (M.IoU and W.IoU) to evaluate experimental results. M.IoU is a measure of the instance-wise IoU for each ground-truth instance averaged over all frames while W.IoU further weights the sizes of segments.

For simplicity, we denote the mapping from the pixel to the pedestrian's label as the composition of superpixel segmentation and superpixel-wise labeling:

$$X = X_{\mathrm{SP}} \cdot X_{\mathrm{CRF}}. \tag{27}$$

The set of pixels assigned with pedestrian's label i (i.e., the mask-type result of pedestrian i) is then denoted $y_i = \{p|X(p) = i\}$. Correspondingly, the ground truth set of the pixel of pedestrian i is y_i^*. M.IoU and W.IoU are defined as

$$M.\mathrm{IoU} = \frac{1}{n_{\mathrm{TR}}} \sum_{i=1}^{n_{\mathrm{TR}}} IoU(y_i, y_i^*), \tag{28}$$

$$W.\mathrm{IoU} = w_i \sum_{i=1}^{n_{\mathrm{TR}}} IoU(y_i, y_i^*), \tag{29}$$

$$w_i = \frac{|y_i^*|}{\sum_{i=1}^{n_{\mathrm{TR}}} |y_i^*|}, \tag{30}$$

where n_{TR} is the number of pedestrian trajectories.

For the semantic-level evaluation, we use the pedestrian IoU (P.IoU) to illustrate that the proposed method improves the semantic-level segmentation performance as

$$P.\mathrm{IoU} = \mathrm{IoU}\left(\bigcup_{i=1}^{n_{\mathrm{TR}}} y_i, \bigcup_{i=1}^{n_{\mathrm{TR}}} y_i^*\right). \tag{31}$$

Furthermore, we compute IoUs along the boundary regions to verity that the object boundaries are well preserved as suggested. For this purpose, we define a boundary region of a pedestrian silhouette as a subtracted region between a dilated region and an eroded region (see Fig. 12 for examples) and then define the IoU for the boundary region. More specifically, for instance-level evaluation, given the i-th pedestrian region y_i, we compute the dilated region y_i^{Di} and also the eroded region y_i^{Er} and then compute the boundary region y_i^B as $y_i^B = y_i^{Di} \backslash y_i^{Er}$. We similarly define the boundary region y_i^{B*} of the ground-truth region y_i^{B*} for the i-th pedestrian. We then define the mean IoU along the boundary regions (denoted $M.\mathrm{IoU}_B$) as

$$M.\mathrm{IoU}_B = \frac{1}{n_{\mathrm{TR}}} \sum_{i=1}^{n_{\mathrm{TR}}} \mathrm{IoU}\left(y_i^B, y_i^{B*}\right), \tag{32}$$

where n_{TR} is the number of pedestrian trajectories. For semantic-level evaluation, we similarly define the pedestrian IoU for the boundary region (denoted $P.\mathrm{IoU}_B$) as

Table 1 Component comparisonon TUD-Campus

	M. IoU [%]	W. IoU [%]	M. IoU$_B$ [%]	Time [min]
SLIC + OT	46.22	77.55	15.26	*8.5*
ESS + optical flow	44.41	77.09	14.79	11.4
ESS + OT (Proposed)	*48.08*	*77.89*	*16.85*	9.6

The best performance data were italicized.

$$P.\mathrm{IoU}_B = \mathrm{IoU}\left(\bigcup_{i=1}^{n_{\mathrm{TR}}} y_i^{B*}, \bigcup_{i=1}^{n_{\mathrm{TR}}} y_i^{B*}\right). \tag{33}$$

Finally, we adopt the computational time as an evaluation metric with which to quantitatively analyze the efficiency of the proposed method.

4.1.3 Baseline methods

For instance-level segmentation, we adopt the methods of Milan et al. [26], He et al. [32] and Ochs et al. [16] as baseline methods. For fair comparison, we modify the baseline methods as follows.

Milan's method generates an overcomplete set of trajectory hypotheses and then assigns superpixels to trajectories. We substitute the trajectory hypothesis with the

Table 2 Instance-level results

	TUD-Stadtmitte			
	M. IoU [%]	W. IoU [%]	M. IoU$_B$ [%]	Time [min]
Ochs's	13.19	25.96	3.39	1255
Milan's	50.97	44.80	20.70	209
He's	*70.36*	79.21	20.23	*0.5*
Proposed	57.48	*81.12*	*20.44*	17.7
	TUD-Campus			
	M. IoU [%]	W. IoU [%]	M. IoU$_B$ [%]	Time [min]
Ochs's	15.08	34.87	2.85	500
Milan's	51.38	49.29	14.27	83
He's	*63.92*	71.54	13.04	*0.2*
Proposed	48.08	*77.89*	*16.85*	9.6
	TUD-Crossing			
	M. IoU [%]	W. IoU [%]	M. IoU$_B$ [%]	Time [min]
Ochs's	6.5	26.65	1.50	1512
Milan's	14.30	22.87	5.89	240
He's	*38.00*	56.17	10.08	*0.65*
Proposed	30.83	*64.18*	*13.54*	20.5
	PETS2009			
	M. IoU [%]	W. IoU [%]	M. IoU$_B$ [%]	Time [min]
Ochs's	14.41	29.54	5.40	4355
Milan's	33.17	34.39	2.24	870
He's	*79.25*	*85.61*	*42.11*	*2.1*
Proposed	68.20	80.61	37.75	118

The best performance data were italicized.

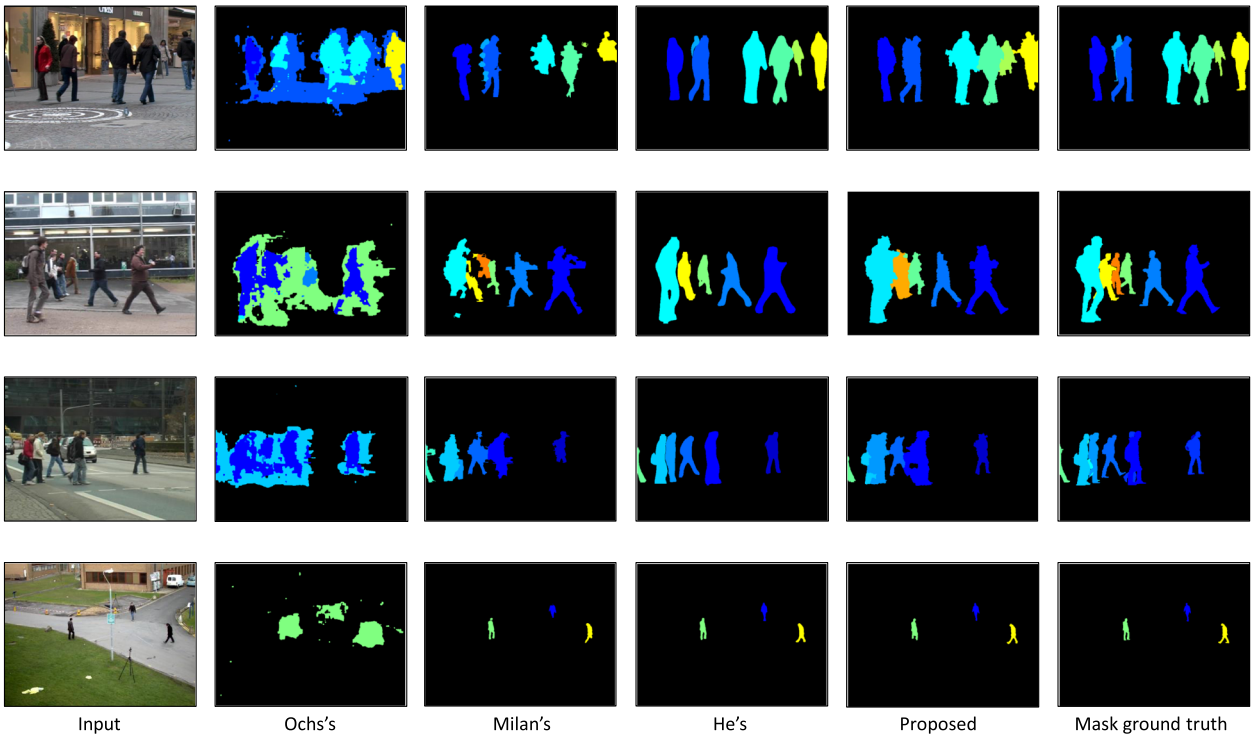

Fig. 11 Instance-level mask-type result

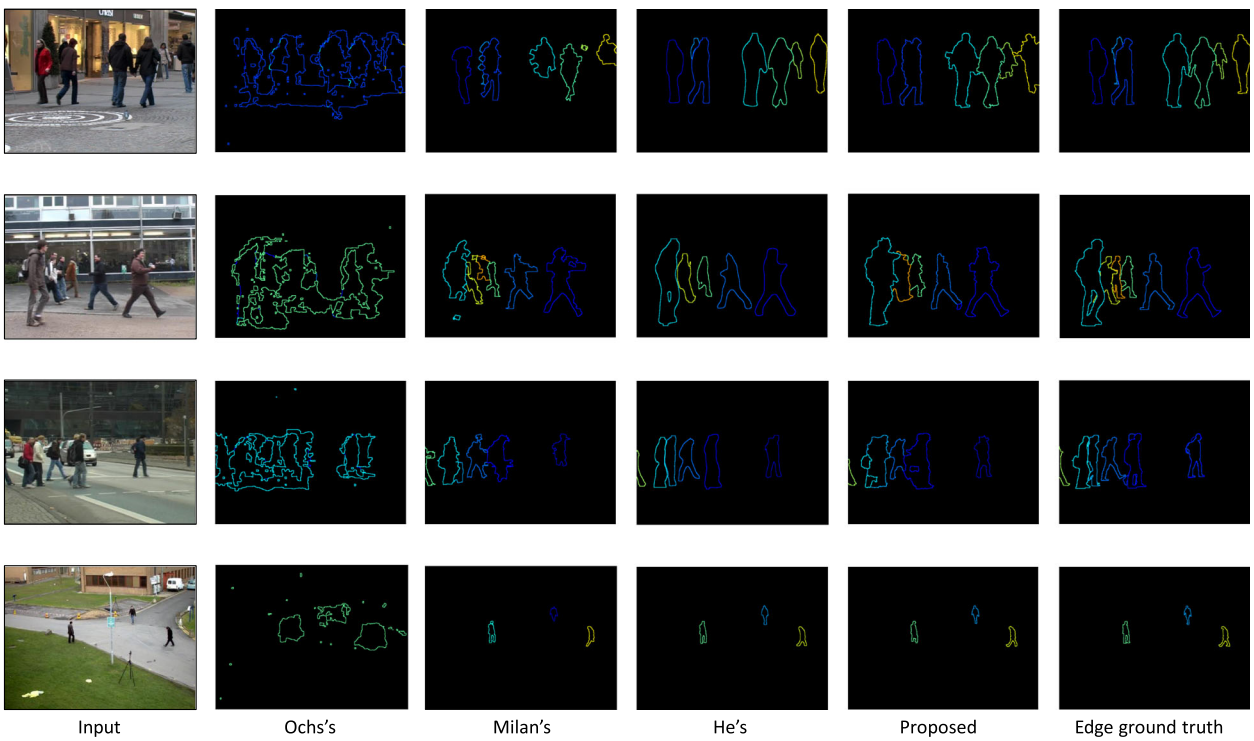

Fig. 12 Instance-level edge-type result

trajectory ground truth and eliminate the update of the trajectory.

He's method and Ochs's method have different pedestrian labeling schemes and thus need to be relabeled. We use a greedy assignment method by which, from the largest ground-truth segments to the smallest, we assign label i of trajectory tr_i to the segment with the highest IoU with y_i^*. Moreover, because He's method generates multi-category instance-level segmentation, we apply the greedy assignment to both human segments and bag segments for the reason that the ground truth of the pedestrian contains both human and bag regions.

We adopt Lin's method (i.e., RefineNet [31]) as a baseline method for the semantic-level segmentation. We use a pre-trained model on the Cityscapes dataset [40] whose output contains 20 labels. We focus only on the quality of the pedestrian silhouette and thus convert the original RefineNet output into a binary mask that only contains the "human" label and "non-human" label. An example of the binary mask is shown in the second column of Fig. 15.

4.1.4 Implementation details

The pedestrian bounding box trajectories used in the experiment are manual annotations. For the ESS, we set $\alpha = 0.7$ and $\beta = 0.7$, and to keep the average size of superpixels the same, we set $\gamma = 545$ for TUD-Stadtmitte, $\gamma = 560$ for TUD-Campus, $\gamma = 475$ for TUD-Crossing and $\gamma = 300$ for PETS2009; i.e., there are approximately 2000 superpixels per frame for TUD datasets and 2850 per frame for PETS2009.

Thresholds th_{Hm} and th_{temp} are set as 0.5. In the spatial pairwise term, ω_e is set as 300 while for CRF, ω_S is set as 8 and ω_T is set as 12. Finally, to handle an arbitrary length of frames, we use a batch process that sets the batch length as 20 frames.

Both instance-level and semantic-level evaluations are conducted on a personal computer with an Intel I7 CPU, 64 GB memory and a NVIDIA GTX 1080Ti GPU. We further address the use of the GPU for each method as follows.

For Ochs's method and Milan's method, GPUs are not used in the computation because no GPU version of codes was provided. For He's method, the experiments are conducted using GPUs. For the proposed method, we only use a GPU for the RefineNet-based background term and not other parts.

4.2 Component comparison

4.2.1 Superpixel

To demonstrate the merits of the ESS, we run a component comparison experiment in which the SLIC superpixel [27] is used in a baseline method. We tune the number of SLIC superpixels to be the same as the number of ESSs. The experimental results presented in Table 1 show that the ESS outperforms the SLIC superpixel.

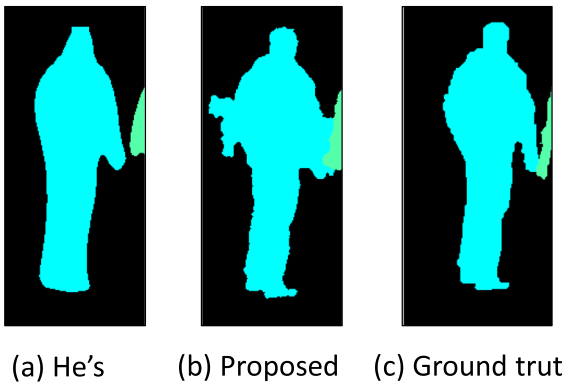

Fig. 13 Instance-level mask-type result for large pedestrians

4.2.2 Temporal pairwise term

We run another component comparison experiment to demonstrate the merits of the OT-based temporal pairwise term compared with the optical-flow-based temporal pairwise term. We follow Liu's work [42] for the optical-flow calculation. We then define an optical-flow-based connectivity function $\mathrm{conn}_{\mathrm{flow}}(p,q)$ with which to substitute $\mathrm{conn}_{\mathrm{T}}(p,q)$.

We denote the average motion vector of superpixel p in frame t as $\boldsymbol{\mu}_{\mathrm{flow}}(p)$, where $p \in \mathcal{L}_{\mathrm{SP}}^t$ and the integral rounding of the vector is $\left[\boldsymbol{\mu}_{\mathrm{flow}}(p)\right]$ with $[\,\cdot\,]$ being the integral rounding function. The set of pixel location vectors of the p-th superpixel is $\mathcal{V}_p = \left\{\boldsymbol{\nu}_{\mathrm{loc}}(j) | X_{\mathrm{SP}}(j) = p\right\}$, and the corresponding locations in frame $t+1$ obtained via $\left[\boldsymbol{\mu}_{\mathrm{flow}}(p)\right]$ are denoted $\hat{\mathcal{V}}_p = \left\{\boldsymbol{\nu}_{\mathrm{loc}}(j) + \left[\boldsymbol{\mu}_{\mathrm{flow}}(p)\right] | X_{\mathrm{SP}}(j) = p\right\}$. Moreover, denoting by $q \in \mathcal{L}_{\mathrm{SP}}^{t+1}$ a superpixel whose pixel location vector set is $\mathcal{V}_q$, the optical-flow-based temporal connectivity function is then defined as

$$\mathrm{conn}_{\mathrm{flow}}(p,q) = \begin{cases} 1 & |\mathcal{V}_p \bigcup \mathcal{V}_q| / |\mathcal{L}_{SP}^t| \geq th_{\mathrm{temp}} \\ 0 & \text{otherwise} \end{cases}. \quad (34)$$

Fig. 14 Instance-level mask-type result for small pedestrians

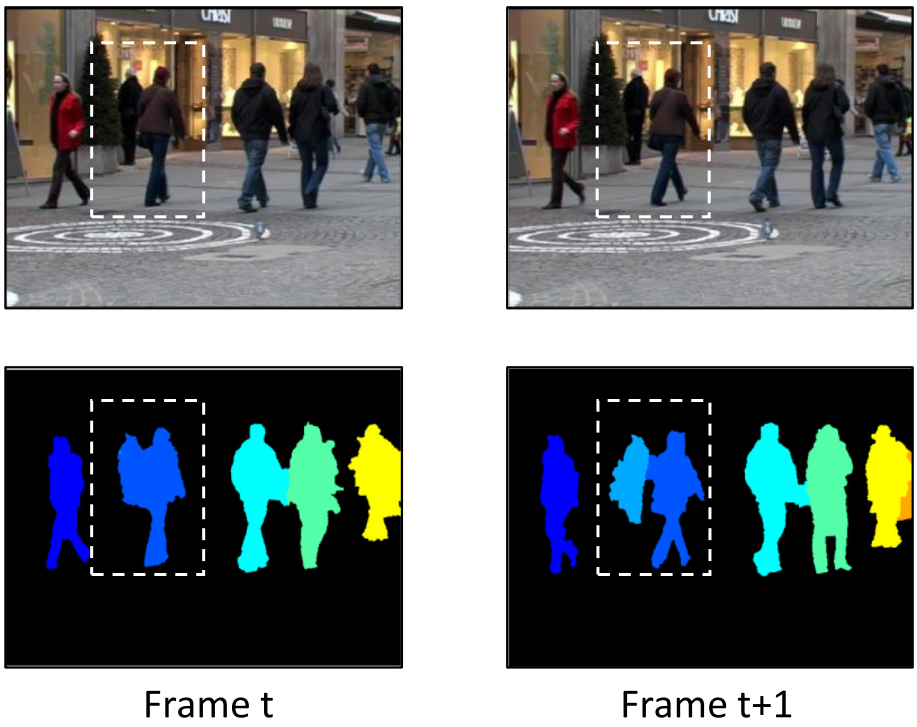

Fig. 15 Examples of failure cases

A set of optical-flow-based temporal neighbors is then defined as

$$\mathcal{N}_{\text{flow}} = \left\{(p,q) | p \in \mathcal{L}_{\text{SP}}, q \in \mathcal{L}_{\text{SP}}, \text{conn}_{\text{flow}}(p,q) = 1\right\}. \tag{35}$$

Subsequently, the optical-flow-based temporal pairwise term is defined similarly to Eq. 19:

$$\sum_{(p,q)\in\mathcal{N}_{\text{flow}}} E_{\text{T}}(p,q,X_{\text{CRF}}(p),X_{\text{CRF}}(q)). \tag{36}$$

We then substitute the OT-based temporal pairwise term with the optical-flow-based term and run the component comparison experiment without changing other settings on the TUD-Campus dataset.

The experimental results are also given in Table 1. The OT-based temporal term performs better than the optical-flow-based temporal term.

4.3 Experimental results

4.3.1 Instance-level evaluation

The instance-level experimental result is presented in Table 2 while examples of visualization mask-type and edge-type results are respectively shown in Fig. 11 and Fig. 12.

The proposed method outperforms Ochs's and Milan's methods for all metrics. On the TUD datasets, the proposed method outperforms He's method in terms of W.IoU and M.IoU$_B$ while underperforming He's method in terms of M.IoU. Furthermore, on the PETS2009

Table 3 Semantic-level results

	TUD-Stadtmitte			TUD-Campus		
	P. IoU [%]	P. IoU$_B$ [%]	Time [min]	P. IoU [%]	P. IoU$_B$ [%]	Time [min]
Lin's (RefineNet)	72.74	11.10	*1.1*	71.74	10.15	*0.5*
Proposed	*79.12*	*30.00*	17.7	*80.45*	*24.05*	9.6
	TUD-Crossing			PETS2009		
	P. IoU [%]	P. IoU$_B$ [%]	Time [min]	P. IoU [%]	P. IoU$_B$ [%]	Time [min]
Lin's (ReFineNet)	73.75	12.26	*1.3*	57.68	11.62	*6.7*
Proposed	*78.82*	*28.27*	20.5	*75.42*	*79.12*	118

The best performance data were italicized.

Fig. 16 Semantic-level mask-type result

dataset, the proposed method fails to outperform He's method.

The performance of the proposed method compared with He's method is explained below.

The sizes of ESSs are almost the same because the third term in Eq. 2 controls the size of the superpixels. Therefore, more superpixels are used to represent a larger pedestrian; i.e., a larger pedestrian is more robust against superpixel labeling error. As a result, our proposed method works better on large pedestrians than small pedestrians as shown in Fig. 13 and Fig. 14. Because the TUD datasets have a higher proportion of large pedestrians, compared with He's method, the proposed method has a higher W.IoU, which gives a higher weight

Fig. 17 Semantic-level edge-type result

to the large pedestrian and results in a lower M.IoU, which normalizes the size of the pedestrian. However, on the PETS2009 dataset, because most pedestrians are small, the proposed method fails to outperform He's method in terms of both M.IoU and W.IoU. The equally sized ESSs are considered the main drawback of the proposed method.

Another drawback of our proposed method is a lack of ability to handle occlusion. Figure 15 shows that the proposed method fails to segment the two pedestrians in frame t because of heavy occlusion. This relates to our adoption of a color-based GMM for pedestrian modeling, which may fail when the appearances of two pedestrians are similar.

4.3.2 Semantic-level evaluation

We also run a semantic-level experiment to illustrate that the use of the proposed method improves the performance of semantic segmentation. Results are presented in Table 3. On all datasets, the proposed method has a much better P.IoU and P.IoU$_B$. This is because not only does the ESS preserve the pedestrian boundary better but also the OT-based temporal pairwise term eliminates the temporally inconsistent segments. Examples of semantic-level mask-type and edge-type results are shown in Fig. 16 and Fig. 17.

4.3.3 Sensitivity analysis

We conduct an analysis of the sensitivity of the segmentation accuracy on the number of superpixels. We test the performance for an approximately exponentially increasing superpixel number on the TUD-Campus dataset and present the results in Table 4. Although the performance increases with the number of superpixels, the computational time is unacceptable if the number is too great; for example, 5000 superpixels per frame. In conclusion, 2000 superpixels per frame is considered a good tradeoff between the segmentation quality and processing time.

5 Conclusion

We proposed a method of extracting multiple pedestrian silhouettes. The proposed method is formulated as a CRF inference problem that incorporates the ESS, semantic segmentation-based human score, and OT-based temporal pairwise term. In addition, we tested the proposed method on public datasets and achieved competitive performance.

A detector of human parts [43] and multiple-detector fusion for the tracking of multiple objects [44] have recently been developed, and a future avenue of research will apply the human-part detector to occlusion reasoning.

Table 4 Sensitivity analysis on TUD-Campus

SP amount	M. IoU [%]	W. IoU [%]	M. IoU$_B$[%]	Time [min]
500	38.27	64.16	11.28	*2.8*
1000	43.36	73.25	14.48	5.3
2000	48.08	77.89	16.85	9.6
5000	*52.08*	*79.67*	*17.21*	30.1

Abbreviations

CRF: Conditional random field; GMM: Gaussian mixture model; OT: Optimal transport; ESS: Edge-sticky superpixel; SLIC: Simple linear iterative clustering; SEEDS: Superpixels extracted via energy-driven sampling; M. IoU: Mean intersections over union; W. IoU: Weighted intersections over union; P. IoU: Pedestrian intersections over union

Acknowledgments

We thank Glenn Pennycook, MSc, from Edanz Group (www.edanzediting.com/ac) for editing a draft of this manuscript.

Authors' contributions

YY executed the experiments, analyzed results, and wrote the initial draft of the manuscript. MY managed the advisor position for the collection of data, designed the experiment, and reviewed the manuscript. YY supervised the design of the work and provided technical support and conceptual advice. All authors read and approved the final manuscript.

Funding

This work was supported by a JSPS Grant-in-Aid for Scientific Research (A) JP18H04115.

Availability of data and materials

The datasets used and/or analyzed during the current study are available from the corresponding author upon reasonable request.

Competing interests

The authors declare that they have no competing interests.

Received: 4 December 2018 Accepted: 6 November 2019
Published online: 29 November 2019

References

1. Plaenkers R, Fua P (2002) Model-based silhouette extraction for accurate people tracking. In: European Conference on Computer Vision. Springer, Berlin. pp 325–339
2. Chen X, He Z, Anderson D, Keller J, Skubic M (2006) Adaptive silhouette extraction and human tracking in complex and dynamic environments. In: Image Processing, 2006 IEEE International Conference On. IEEE, New York. pp 561–564
3. Ahn J-H, Choi C, Kwak S, Kim K, Byun H (2009) Human tracking and silhouette extraction for human–robot interaction systems. Patt Anal Appl 12(2):167–177
4. Howe NR (2004) Silhouette lookup for automatic pose tracking. In: Computer Vision and Pattern Recognition Workshop, 2004. CVPRW'04. Conference On. IEEE, New York. pp 15–22
5. Wang L, Suter D (2007) Recognizing human activities from silhouettes: Motion subspace and factorial discriminative graphical model. In: Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference On. IEEE, New York. pp 1–8
6. Chaaraoui AA, Climent-Pérez P, Flórez-Revuelta F (2013) Silhouette-based human action recognition using sequences of key poses. Patt Recogn Lett 34(15):1799–1807
7. Wang L, Suter D (2007) Learning and matching of dynamic shape manifolds for human action recognition. IEEE Trans Image Process 16(6):1646–1661
8. Ikizler N, Duygulu P (2009) Histogram of oriented rectangles: a new pose descriptor for human action recognition. Image Vision Comput 27(10):1515–1526

9. Collins RT, Gross R, Shi J (2002) Silhouette-based human identification from body shape and gait. In: Automatic Face and Gesture Recognition, 2002. Proceedings. Fifth IEEE International Conference On. IEEE, New York. pp 366–371
10. Wang L, Tan T, Ning H, Hu W (2003) Silhouette analysis-based gait recognition for human identification. IEEE Trans Patt Anal Mach Intell 25(12):1505–1518
11. Liu Z, Sarkar S (2004) Simplest representation yet for gait recognition: averaged silhouette. In: Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference On. IEEE, New York Vol. 4. pp 211–214
12. Caelles S, Maninis K-K, Pont-Tuset J, Leal-Taixé L, Cremers D, Van Gool L (2017) One-shot video object segmentation. In: CVPR 2017. IEEE, New York
13. Cheng J, Tsai Y-H, Wang S, Yang M-H (2017) Segflow: joint learning for video object segmentation and optical flow. In: 2017 IEEE International Conference on Computer Vision (ICCV). IEEE, New York. pp 686–695
14. Migdal J, Grimson WEL (2005) Background subtraction using markov thresholds. In: Application of Computer Vision, 2005. WACV/MOTIONS'05 Volume 1. Seventh IEEE Workshops On. IEEE, New York Vol. 2. pp 58–65
15. Zivkovic Z (2004) Improved adaptive gaussian mixture model for background subtraction. In: Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference On. IEEE, New York Vol. 2. pp 28–31
16. Ochs P, Malik J, Brox T (2014) Segmentation of moving objects by long term video analysis. IEEE Trans Patt Anal Mach Intell 36(6):1187–1200
17. Narayana M, Hanson A, Learned-Miller E (2013) Coherent motion segmentation in moving camera videos using optical flow orientations. In: Computer Vision (ICCV), 2013 IEEE International Conference On. IEEE, New York. pp 1577–1584
18. Unger M, Werlberger M, Pock T, Bischof H (2012) Joint motion estimation and segmentation of complex scenes with label costs and occlusion modeling. In: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference On. IEEE, New York. pp 1878–1885
19. Chen Y-M, Bajic IV (2011) A joint approach to global motion estimation and motion segmentation from a coarsely sampled motion vector field. IEEE Trans Circ Syst Vid Technol 21(9):1316–1328
20. Ren S, He K, Girshick RB, Sun J (2015) Faster r-cnn: Towards real-time object detection with region proposal networks. In: IEEE Transactions on pattern analysis and machine intelligence, 39. pp 1137–1149
21. Girshick R, Donahue J, Darrell T, Malik J (2014) Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, New York
22. Girshick R (2015) Fast R-CNN. In: Proceedings of the International Conference on Computer Vision (ICCV). IEEE, New York
23. Kim C, Li F, Ciptadi A, Rehg JM (2015) Multiple hypothesis tracking revisited. In: Proceedings of the IEEE International Conference on Computer Vision. IEEE, New York. pp 4696–4704
24. Choi W (2015) Near-online multi-target tracking with aggregated local flow descriptor. In: Proceedings of the IEEE International Conference on Computer Vision. IEEE, New York. pp 3029–3037
25. Keuper M, Tang S, Zhongjie Y, Andres B, Brox T, Schiele B (2016) A multi-cut formulation for joint segmentation and tracking of multiple objects. Computing Research Repository (CoRR):1–14
26. Milan A, Leal-Taixé L, Schindler K, Reid I (2015) Joint tracking and segmentation of multiple targets. In: Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference On. IEEE, New York. pp 5397–5406
27. Achanta R, Shaji A, Smith K, Lucchi A, Fua P, Süsstrunk S (2012) Slic superpixels compared to state-of-the-art superpixel methods. IEEE Trans Pattern Anal Mach Intell 34(11):2274–2282
28. Brox T, Bruhn A, Papenberg N, Weickert J (2004) High accuracy optical flow estimation based on a theory for warping. In: European Conference on Computer Vision. Springer, Berlin. pp 25–36
29. Van den Bergh M, Boix X, Roig G, Van Gool L (2015) Seeds: Superpixels extracted via energy-driven sampling. Int J Comput Vis 111(3):298–314
30. Makihara Y, Tanoue T, Muramatsu D, Yagi Y, Mori S, Utsumi Y, Iwamura M, Kise K (2015) Individuality-preserving silhouette extraction for gait recognition. IPSJ Trans Comput Vis Appl 7:74–78
31. Lin G, Milan A, Shen C, Reid I (2017) Refinenet: multi-path refinement networks for high-resolution semantic segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, New York
32. He K, Gkioxari G, Dollár P, Girshick R (2017) Mask r-cnn. In: Computer Vision (ICCV), 2017 IEEE International Conference On. IEEE, New York. pp 2980–2988
33. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. IEEE, New York. pp 770–778
34. Dosovitskiy A, Fischer P, Ilg E, Hausser P, Hazirbas C, Golkov V, van der Smagt P, Cremers D, Brox T (2015) Flownet: Learning optical flow with convolutional networks. In: Proceedings of the IEEE International Conference on Computer Vision. IEEE, New York. pp 2758–2766
35. Chang J, Wei D, Fisher III JW (2013) A video representation using temporal superpixels. In: Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference On. IEEE, New York. pp 2051–2058
36. Long J, Shelhamer E, Darrell T (2015) Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. IEEE, New York. pp 3431–3440
37. Wu Y, Lin Y, Dong X, Yan Y, Bian W, Yang Y (2019) Progressive learning for person re-identification with one example. IEEE Trans Image Process 28(6):2872–2881
38. Dollár P, Zitnick CL (2013) Structured forests for fast edge detection. In: Computer Vision (ICCV), 2013 IEEE International Conference On. IEEE, New York. pp 1841–1848
39. Á2 À. (2001) Fast approximate energy minimization via graph cuts. IEEE Trans Patt Anal Mach Intell 23(11):1
40. Cordts M, Omran M, Ramos S, Rehfeld T, Enzweiler M, Benenson R, Franke U, Roth S, Schiele B (2016) The cityscapes dataset for semantic urban scene understanding. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. IEEE, New York. pp 3213–3223
41. Rother C, Kolmogorov V, Blake A (2004) Grabcut: interactive foreground extraction using iterated graph cuts. In: ACM Transactions on Graphics (TOG). ACM, New York Vol. 23. pp 309–314
42. Liu C, et al (2009) Beyond pixels: exploring new representations and applications for motion analysis. PhD Thesis:48–50
43. Cao Z, Simon T, Wei S-E, Sheikh Y (2017) Realtime multi-person 2d pose estimation using part affinity fields. In: CVPR. IEEE, New York
44. Henschel R, Leal-Taixé L, Cremers D, Rosenhahn B (2017) Fusion of head and full-body detectors for multi-object tracking. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). pp 1509-150909

Publisher's Note