Computer Vision and Image Understanding xxx (2011) xxx–xxx

Contents lists available at ScienceDirect

Computer Vision and Image Understanding

journal homepage: www.elsevier.com/locate/cviu

Learning spatio-temporal dependency of local patches for complex motion segmentation

Jiang Xu [a,*], Junsong Yuan [b], Ying Wu [a]

[a] Department of EECS, Northwestern University, Evanston, IL 60208, USA
[b] School of EEE, Nanyang Technological University, Singapore 639798, Singapore

ARTICLE INFO

Article history:
Received 5 March 2010
Accepted 16 November 2010
Available online xxxx

Keywords:
Motion segmentation
Learning
Motion profile symmetry correlation
Bipolar segmentation

ABSTRACT

Segmenting complex motion, such as articulated motion and deformable objects, can be difficult if the prior knowledge of the motion pattern is not available. We present a novel method for motion segmentation by learning the motion priors from exemplar motions to guide the segmentation. Instead of modeling the motion field explicitly, we decompose each video frame into a number of local patches and learn the spatio-temporal contextual relations among them, *e.g.*, if their motion relationships are consistent with that from the training data. Based on a novel motion feature to measure the relative motion of two patches, the SVM classifier learns their pairwise relationship. We convert the motion segmentation problem to a binary labeling problem, and propose an iterative solution to group the local patches whose motions are consistent. Compared with other approaches, such as the graph cut and normalized cut methods, this new method is computationally more efficient and is able to better handle the inaccurate inference of pairwise relationships. Results on both synthesized and real videos show that our method can learn to segment different types of complex motion patterns.

1. Introduction

Segmenting a scene based on its motion is an important problem in computer vision. A common approach to motion segmentation is to extract the local motion features (*e.g.*, optical flow [1]) of all the pixels, then perform grouping based on the coherence in these local motion features.

The coherence among these local motion features actually depends on the complexity in the motion of the object, and thus also on the semantic levels of the object. Conventionally, most motion segmentation methods are either based on the coherence in terms of simple motions (such as rigid or affine motion [2–4]), or based on the smoothness in the motion (*e.g.*, smooth motion layers, or Markov random fields [5–9]). Methods based on the coherence in simple motions may result in over-segmentation if the object exhibits more complex motion, *e.g.*, articulation or deformation.

However, these methods are confronted when we want to segment an object presenting more complex motion, *e.g.*, a shearing scissors or a walking person (Fig. 1). Although the motion of the parts can be simple, the entire motion of the object does have more degrees of freedom, and cannot be covered by simple rigid motions or smooth motions. This paper is concerned on a new learning-based solution to this problem.

This is a quite difficult problem as the motion can be quite complex. However, in reality, it is not uncommon that the complex motions may exhibit good structures, implying that the intrinsic dimension of these motions can be actually low. In the example of the shearing scissors, the opening–closing motion pattern is quite predictable so that a specific motion model can be easily specified. However, in the case of a walking person, although the cyclic motion pattern does have a low intrinsic dimension, it is very difficult, if not impossible, to have an explicit model for such a motion.

In this paper we propose to learn the complex motion from training examples for motion segmentation. The complex motion patterns are labeled in the training video. After learning, we can segment objects in video that exhibit the same motion pattern.

To achieve this goal, we decompose each video frame into a few local image patches and focus on learning the spatio-temporal relationship among the patches. Specifically, suppose i and j are two local patches, we formulate the pairwise relationship $f_{i,j}$ as a binary classification problem: $f_{i,j} = 1$ if two patches belong to the target motion, and $f_{i,j} = -1$ otherwise. Given a test video where the pairwise relations among the patches are inferred, a novel segmentation approach is further proposed in this paper to find the most stable group iteratively. As the pairwise relationships are mostly consistent to each other in the stable group, it provides a robust segmentation of the target motion pattern, despite the noisy and cluttered backgrounds.

* Corresponding author.
E-mail address: jxu323@eecs.northwestern.edu (J. Xu).

1077-3142/$ - see front matter © 2010 Elsevier Inc. All rights reserved.
doi:10.1016/j.cviu.2010.11.010

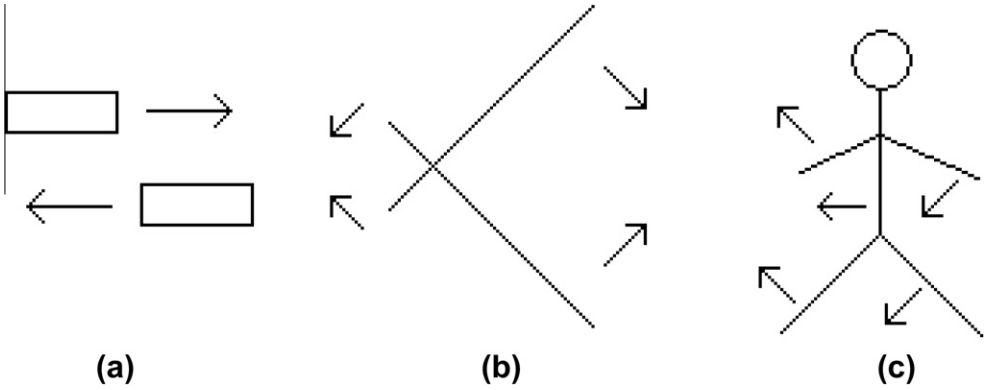

Fig. 1. Examples of motion that cannot be segmented by traditional methods. (a) Center-symmetrical motion, such as two trains passing by. (b) Mirror-symmetrical motion, such as a shearing scissors. (c) Articulated motion, such as human walking. The intrinsic dimensionality of the motion patterns above is not very high. However, we cannot simply use the rigid-body prior or smooth prior to segment those structured motions out.

To provide robust motion segmentation, we address the following two critical questions:

(1) How can we learn the pairwise relationship $f_{i,j}$ between two patches i and j?

Given two local patches, i and j, we need to identify whether their relative motion is consistent with that from the training examples. This is a two-class classification problem, as $f_{i,j}$ is either 1 or -1. To learn the pairwise relation $f_{i,j}$, a novel spatio-temporal contextual feature, called *motion profile symmetry correlation*, is presented to characterize the relative motion of the two patches i and j. This type of feature can be used to describe complex motion, such as opposite motion and mirror-symmetric motion. In the training sequences, we manually label the region of the target motion pattern, in order to obtain the ground truth of the pairwise relations $f_{i,j}$. Then a support vector machine (SVM) is applied to learn the binary relation $f_{i,j}$ using the proposed feature.

(2) How to segment the target motion pattern given the pairwise relation $f_{i,j}$?

To segment a frame in a test sequence, we use all of the local patches x_i to construct a graph, where each x_i corresponds to a

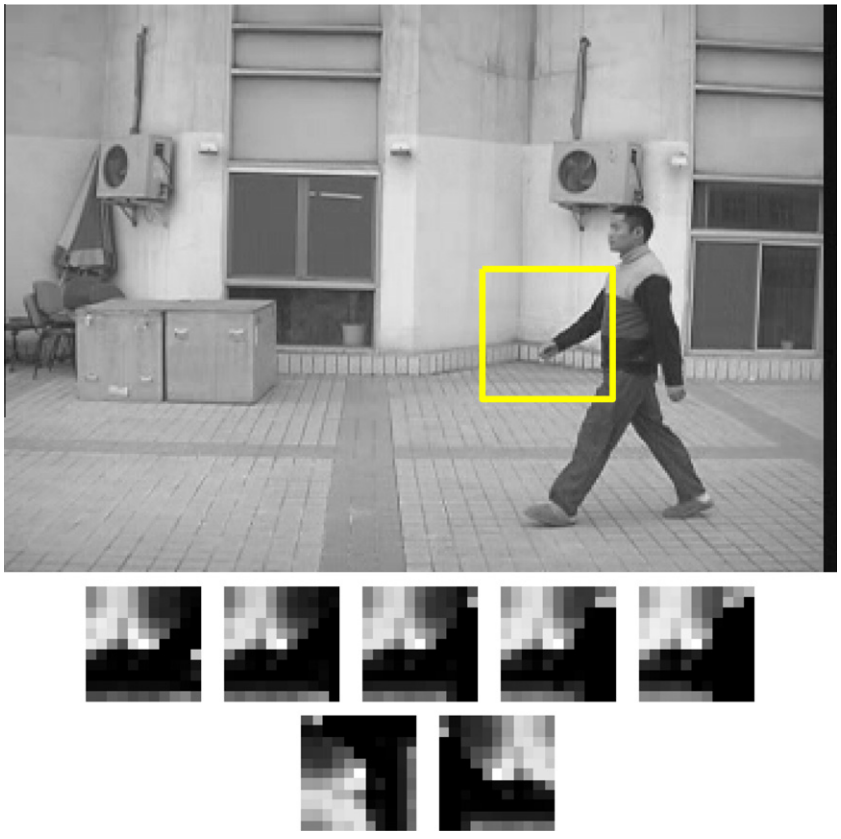

Fig. 2. Top row: the region for motion profile extraction. Middle row: the motion profiles of the patch at the center of the region, from frame $t-2$ to frame $t+2$. From the motion profiles, we can observe the arm region is moving up-left in those frames. Bottom row: the steering and reflection motion profiles. The left one is ${}^{90}P_i^t$, and the right one is $\left({}^{90}P_i^t\right)^T$.

Please cite this article in press as: J. Xu et al., Learning spatio-temporal dependency of local patches for complex motion segmentation, Comput. Vis. Image Understand. (2011), doi:10.1016/j.cviu.2010.11.010

vertex. Instead of labeling the local patches x_i directly, we first label each edge $f_{i,j}$ using the SVM. Then we can infer the binary label of local patches from their pairwise relations $f_{i,j}$. However, this is a non-trivial problem because the estimation of $f_{i,j}$ can be inaccurate. We thus propose a new method, called *bipolar segmentation*, to handle the unreliable $f_{i,j}$ for robust segmentation. It provides an efficient iterative solution to this NP-hard binary labeling problem. Compared with other alternative solutions, such as approximate graph cut and normalized cut, our method shows clear advantages in both numerical experiments and real video sequences. It is robust to the cluttered and noisy backgrounds, in both learning and segmentation processes.

The rest of the paper is organized as follows. We introduce the related work in Section 2 and propose the new feature to learn the pairwise relationship in Section 3. After that, in Section 4, the bipolar segmentation is presented and discussed. Our experiment results are shown in Section 5, followed by the conclusion in Section 6.

2. Related work

Rigid motion segmentation has been studied extensively in the literature. There are mainly two categories of approaches: parametric and non-parametric motion segmentation.

For the parametric method, it assumes that the object motion follows some parameterized model, such as the 2-D affine motion [2], and 3-D rigid body motion [10]. The expectation–maximization (EM) algorithm is popularly utilized for model fitting. Some other works formulate the problem as matrix factorization [11,12,3]. In [13], it discusses how to select the motion model, and GPCA [4] provides a unified view to deal with the problem. Also, there are related works that consider multiple cameras for motion segmentation [14], where the EM algorithm is also applied. However, these approaches cannot well handle complex motions.

The second type of motion segmentation is non-parametric approach. In [15], a hierarchical tree is applied for bottom-up segmentation. In [16], a pyramid was built to organize the local similarities. When assuming that the motion field is smooth [17,18], Markov random fields (MRF) [19], discriminative random fields (DRF) [20], and conditional random fields (CRF) [21] are applied. This category of approaches has a similar objective function, and the segmentation is usually associated with energy minimization, where the solutions include minimal cuts [22], graph cuts [23], or normalized cut [24]. These approaches guarantee the global optimum by the graph theory [22,23] or spectral analysis [25–27,24,28,29]. Despite certain success, these methods are still confronted when segmenting complex motion patterns, *e.g.*, waving water or a traffic flow. The smoothness prior may not be simply applied due to the structure of the motion patterns. Therefore, instead of extracting the motion field explicitly, some existing approaches treat the complex motions as dynamic textures and apply stochastic processes to model them [30,31]. Although the stochastic modeling is effective for motion patterns that follow stationary processes, the method cannot be used when the motion is not a "dynamic texture", for example, a shearing scissor or a walking person (Fig. 1).

Besides the above approaches, there are learning-based approaches for motion segmentation. Some previous works learn the model of the segmentation boundary, such as [32–34]. Other works consider the context of pixels in the image, such as the

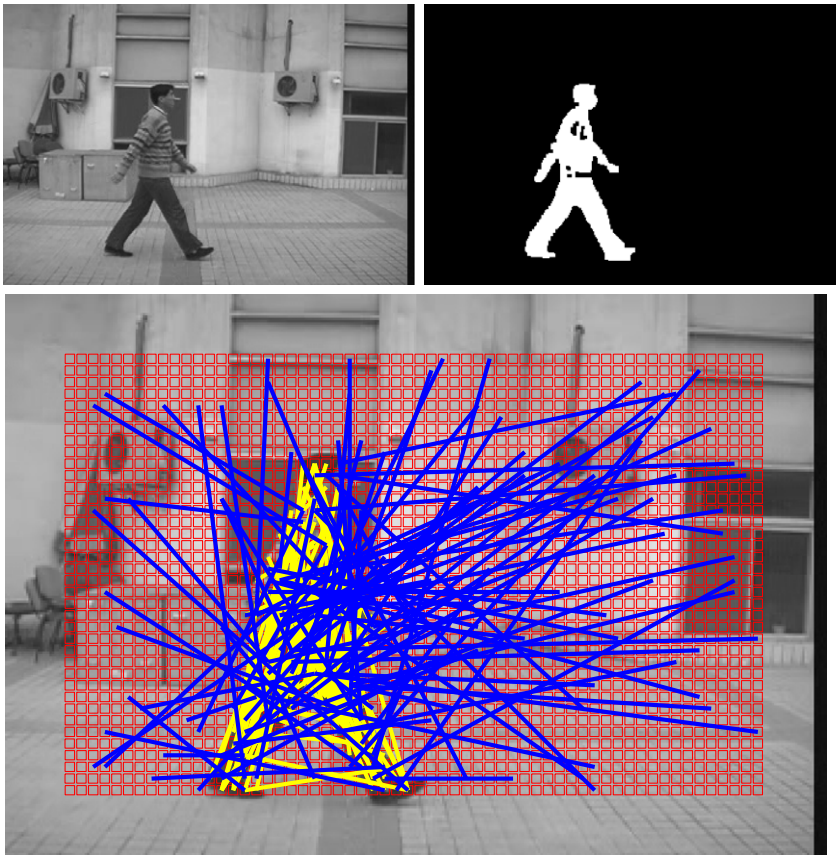

Fig. 3. Top left: one frame for learning. Top right: ground truth. Lower: patches (shown in red boxes). Some positive samples (shown in yellow lines) and some negative samples (shown in blue lines). Because we cannot compute motion profiles at the boundary, the boundary region is neglected in the segmentation algorithm. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Please cite this article in press as: J. Xu et al., Learning spatio-temporal dependency of local patches for complex motion segmentation, Comput. Vis. Image Understand. (2011), doi:10.1016/j.cviu.2010.11.010

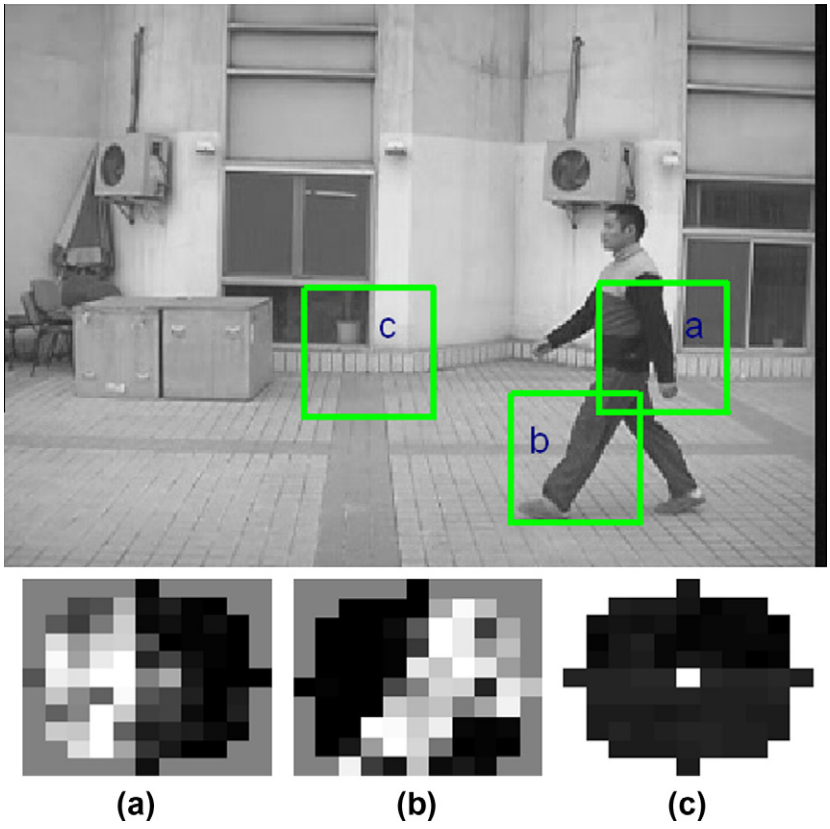

Fig. 4. Top: outlines of three image regions to compute their center patch's force to their nearby patches. Bottom: the corresponding force map. In (a), its center patch is attracted by its left neighbors and repelled by its right neighbors. In (b), its center patch is attracted by its neighbors shaped as the leg. In (c), its center patch is repelled by all of its neighbors.

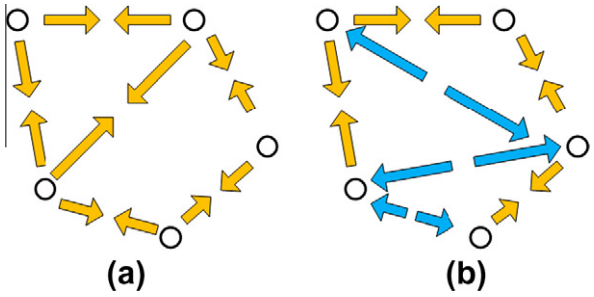

Fig. 5. The difference between our problem and most of others. (a) In unipolar segmentation, a data point can only attract other data. (b) In bipolar segmentation, a data point can attract or repel other data.

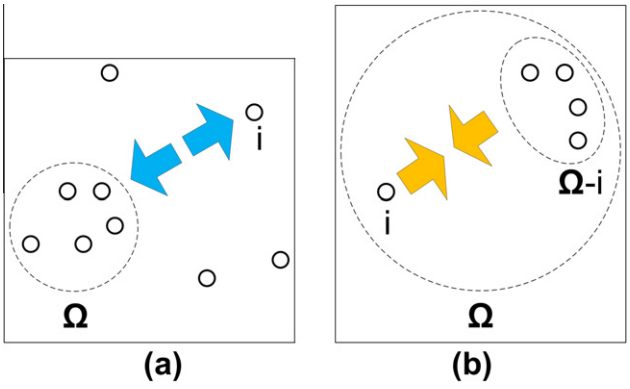

Fig. 6. Illustration of the definition of stable group. (a) The patch i is out of the group $\boldsymbol{\Omega}$, and it is repelled by the group. (b) The patch i is inside the group $\boldsymbol{\Omega}$, and it is attracted by the group's other patches.

auto-context [35]. Moreover, if the moving object can be learned from the training examples, segmentation can be interlaced with object detection [36], or object recognition [37]. For example, to segment the vehicle from static backgrounds, motion segmentation becomes the detection and tracking problem [38–40]. For segmenting articulated bodies, bottom-up segmentation based on learning can be utilized [41].

3. Learning the pairwise relation $f_{i,j}$

Our goal of learning is to identify the condition under which the two patches should be grouped together, *i.e.*, they both belong to the same target object. To describe the pairwise relation between i and j, we extract the motion profiles for each individual patch, and then extract the motion profile symmetry correlation from the two patches. Only motion feature is considered here, although other types of features, *e.g.*, appearance, can also be applied to further improve the segmentation result.

3.1. Steering/reflecting motion profile

Let's first review the motion profile [8]. Let $\mathbf{I}^t(\mathbf{X}_i)$ denote the center pixel of patch i at time t, the similarity measure w.r.t. spatial difference $\mathbf{dx}$ is based on the sum of squared differences (SSD):

$$S_i^t(\mathbf{dx}) = \exp\left(-\sum_{\mathbf{w}}(\mathbf{I}^t(\mathbf{X}_i+\mathbf{w}) - \mathbf{I}^{t+1}(\mathbf{X}_i+\mathbf{dx}+\mathbf{w}))^2/\sigma_{\text{ssd}}^2\right),$$

Please cite this article in press as: J. Xu et al., Learning spatio-temporal dependency of local patches for complex motion segmentation, Comput. Vis. Image Understand. (2011), doi:10.1016/j.cviu.2010.11.010

where $\mathbf{w}$ denotes a local neighborhood, and σ_{ssd} is the stand variance of SSD. We obtain motion profile by normalizing the similarity measure

$$P_i^t = \left[\frac{S_i^t(\mathbf{dx})}{\sum_{\mathbf{dx}} S_i^t(\mathbf{dx})}\right]_{\mathbf{dx}}.$$

We keep P_i^t in a matrix form, and extend the motion profile idea by rotating and transposing it. We propose the steering motion profile ${}^{\alpha}P_i^t$, where α represents the rotation angle in degrees. For example, ${}^{90}P_i^t$ is the original motion profile rotated by 90°. We also propose the reflecting motion profile $\left({}^{\alpha}P_i^t\right)^T$, where T represents transposing. For example, $\left({}^{90}P_i^t\right)^T$ is the original motion profile rotated by 90°, then transposed (Fig. 2). For notional convenience, we also denote ${}^{\alpha}P_i^t$ by $\left({}^{\alpha}P_i^t\right)^I$.

3.2. Motion profile symmetry correlation

If we use the original motion profiles of patch i and patch j to describe the motion coherence of the two patches, one straightforward consideration is to take the inner product of the motion profiles

$$\left\langle P_i^{t+dt_i}, P_j^{t+dt_j} \right\rangle,$$

where $\langle\cdot,\cdot\rangle$ is the inner product, and this is also the approach taken in [8]. The inner product is a scalar. The larger the value, the more similar the two motion profiles, *i.e.*, they move toward the same direction.

However, the inner product of the motion profile can only describe whether the two patches perform the same-direction motion. To describe more complex pairwise relations, such as the opposite motion, the mirror-symmetric motion, and motion the coherence with a time delay, we need a more advanced measure of the two motion profiles. In view of this, we construct the *motion profile symmetry correlation* feature using the steering/reflecting motion profile. This feature is a vector, by concatenating the inner products of steering/reflecting motion profiles:

$$\mathbf{C}_{i,j}^t = \left[\left\langle \left({}^{\alpha}P_i^{t+dt_i}\right)^{\beta}, P_j^{t+dt_j} \right\rangle\right]_{\alpha,\beta,dt_i,dt_j},$$

where $[\cdot]$ represents concatenation, dt_i, dt_j are the time differences, α is the steering angle and takes value of 0°, 90°, 180°, 270°, and β takes value of I (identical) and T (reflecting). In our experiments, the time difference ranges from -2 to +2 (frames). Therefore, five frames are used, and thus $\mathbf{C}_{i,j}^t$ is a $5 \times 5 \times 8 = 200$ dimensional vector (four steering motion profile and four reflecting motion profile). This feature has much higher dimensionality. As each

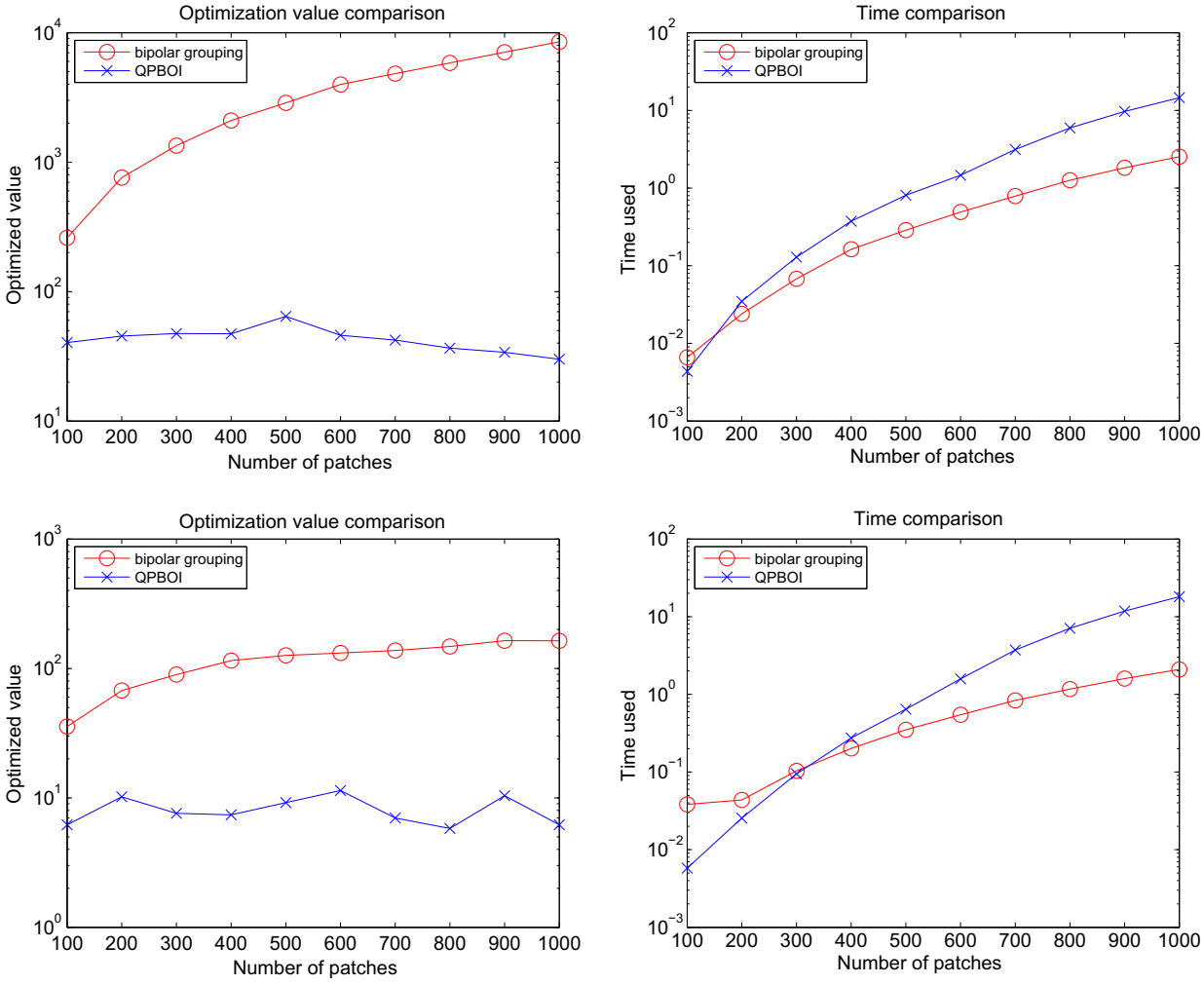

Fig. 7. The comparison of bipolar segmentation and QPBOI. The upper row is the value of the objective function and time usage when positive/negative forces are equally 10%, and the lower row is the objective function and time usage when positive/negative forces are 5% and 20%. From both cases, we observe that bipolar segmentation always obtains much larger optimization value than QPBOI, and uses much less time when the number of patches is not so small. (*Note*: All the figures use logarithmical y-axis.)

Please cite this article in press as: J. Xu et al., Learning spatio-temporal dependency of local patches for complex motion segmentation, Comput. Vis. Image Understand. (2011), doi:10.1016/j.cviu.2010.11.010

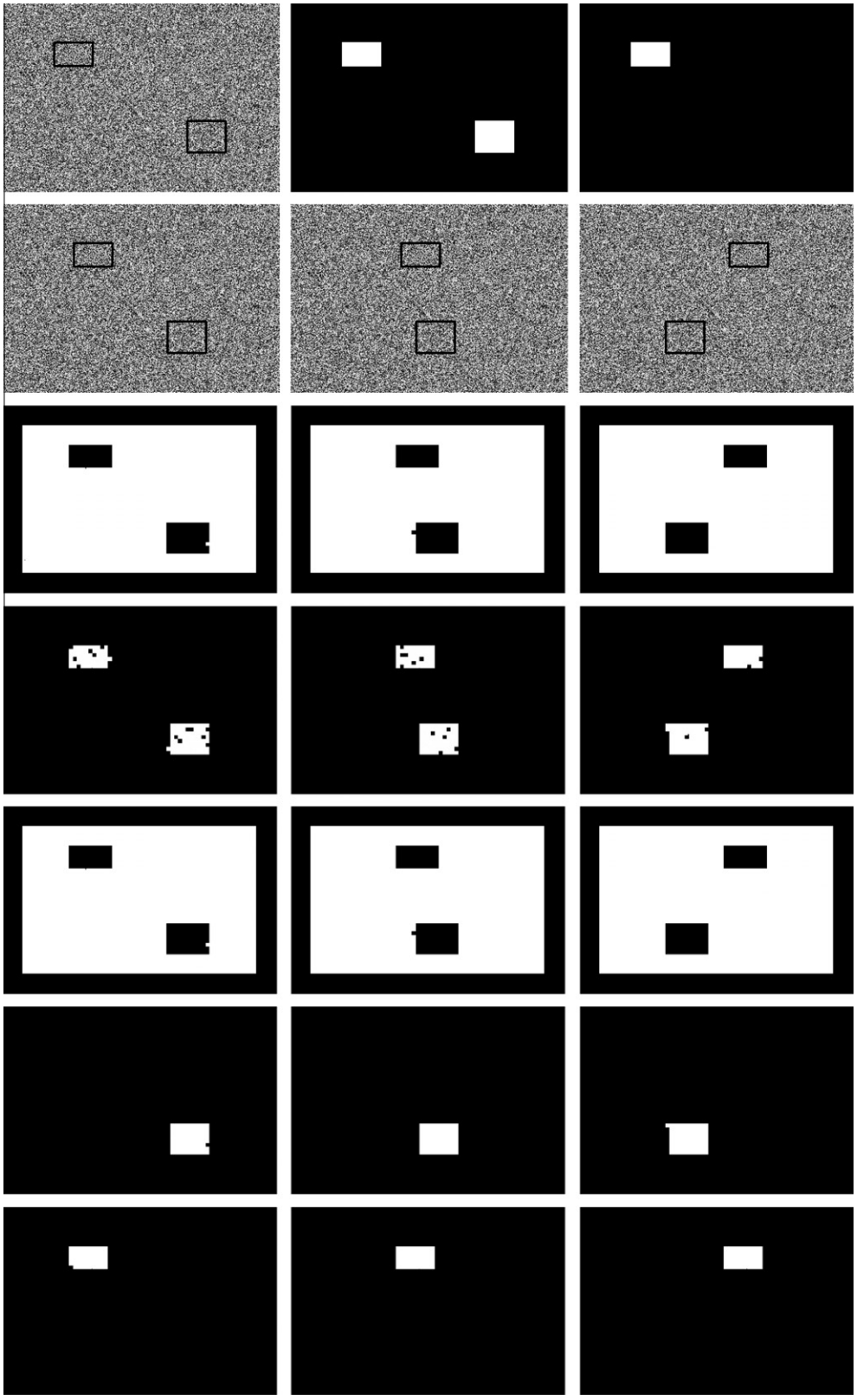

Fig. 8. Row 1: the frame for training, and two different labeling for this frame. Row 2: some frames for segmentation. The black lines show the outlines of the boxes and in the real video there are no black lines. Row 3–4: corresponding segmentation results for the first labeling. Row 5–7: corresponding segmentation results for the second labeling. Because we cannot compute motion profiles at the boundary, the boundary region is neglected in the segmentation algorithm.

dimension of the feature favors one specific coherence pattern, such as opposite motion, mirror-symmetric motion or identical motion with time delay, this feature vector is able to describe more complex motions.

3.3. Classification using support vector machine

To provide the training data, we manually label the frames in a video sequence. In the learning process, if patch i and patch j

Please cite this article in press as: J. Xu et al., Learning spatio-temporal dependency of local patches for complex motion segmentation, Comput. Vis. Image Understand. (2011), doi:10.1016/j.cviu.2010.11.010

belongs to the same object, we treat the patch pair $\{i, j\}$ as a positive example, and *vice versa*. Therefore, for each $\{i, j\}$ pair, we have positive–negative class label and feature vector $\mathbf{C}_{i,j}^{t}$, and we need to learn a classifier to classify the feature vectors based on the class labels.

As shown in Fig. 3, we randomly choose patch pairs to generate positive and negative samples, and then use SVM to learn a classifier. As we want to learn a classifier that can produce "soft" classification result, *i.e.*, output values in the range of $(-1,1)$ instead of the discrete values $\{-1,1\}$, we choose the SVM classifier in LIBSVM

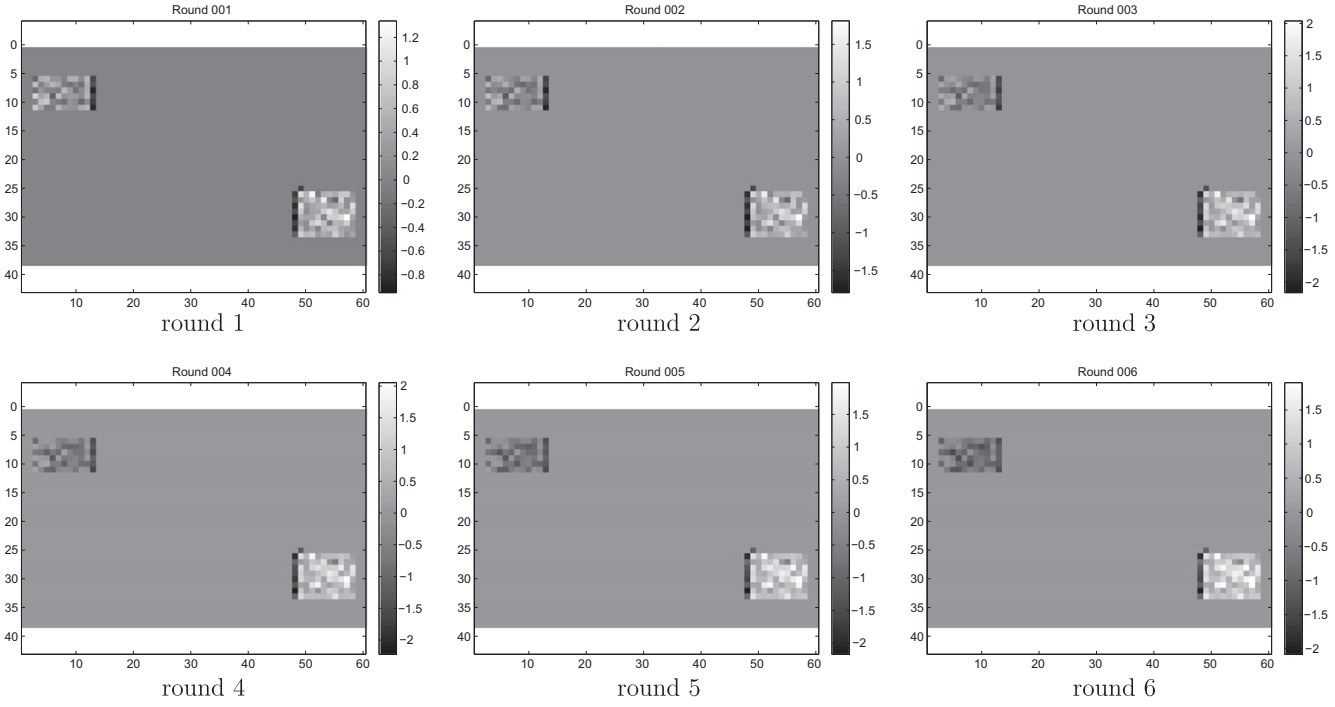

Fig. 9. Example of one iteration process of bipolar segmentation. At first, the two boxes are assigned equal value. During the iterations, the value of the larger box becomes larger and larger. After six iterations, the algorithm terminates and the larger box is segmented out.

Fig. 10. Segmentation of articulated motion. Row 1: one labeled frame. Row 2: testing frames. Row 3: the segmented patches.

Please cite this article in press as: J. Xu et al., Learning spatio-temporal dependency of local patches for complex motion segmentation, Comput. Vis. Image Understand. (2011), doi:10.1016/j.cviu.2010.11.010

[42], which can produce "positive probability" estimation ranges in $(0,1)$. We map the probability to $(-1,1)$ linearly as our final classification result.

3.4. Pairwise relation as force

Given a testing sequence, at time t, for each patch pair $\{i,j\}$, we are able to compute their corresponding pairwise feature $\mathbf{C}_{i,j}^t$, then use the SVM classifier learnt to determine the pairwise coherence of the patches, which ranges in $(-1,1)$. We denote by $f_{i,j}^t$ and call it *force* between patch i and patch j. Coherent patches have positive force between them and attract each other, and *vice versa* (Fig. 4).

4. Bipolar segmentation

Given a test video, after learning the pairwise relation $f_{i,j}$ between local patches, we can perform motion segmentation. Before describing our method, it worths comparing with some existing segmentation methods that also rely on affinity matrix, such as minimum cut and normalized cut. In those cases, the affinity value $f_{i,j}$ describes the relations between two pixels or patches (similarity or dissimilarity) and $f_{i,j}$ is always positive, *e.g.*, ranges in $(0,1)$. As an analogy, there are forces among data points, and for any data point, it "attracts" all the rest at different strengths. We call this type of segmentation as *unipolar segmentation*, which refers to the situation when only one kind of force is considered.

In contrast, in our case, in order the describe whether two patches belong to the same target motion pattern or not, the pairwise relation $f_{i,j}$ learned through an SVM can be either positive or negative, *e.g.*, ranges in $(-1,1)$. Therefore we use *bipolar segmentation* to refer to our new segmentation solution, because both attraction and repulsion forces are considered (Fig. 5). Interestingly, most existing approaches cannot be applied directly for bipolar segmentation, because of the negative values of $f_{i,j}$. For example, the min-cut max-flow solution requires the capacity of each edge, $f_{i,j}$, to be a positive value, thus cannot be applied here. Some other works [43,44] consider both positive and negative affinities, but their formulation and objective function are totally different and cannot be applied in our case. As a result, we need a new segmentation algorithm for our task. In the experiment section, we will justify that choosing $f_{i,j}$ within $(-1,1)$ instead of $(0,1)$ provides better result.

4.1. Stable group

We consider our segmentation problem based on the physical meaning of our force modeling. In the real world, an object is stable in its own form, because it attracts its own atoms, and repels other atoms. Therefore, following this analogy, we propose the definition of *stable group* as: given all patches $\mathbf{P}$ and the forces between each pair of patches, a stable group $\mathbf{\Omega} \subset \mathbf{P}$ is a collection of patches that satisfy the following constraints (Fig. 6):

(1) The patches outside the group is repelled by the group, or

$$\forall i \notin \mathbf{\Omega}, \sum_{j\in\mathbf{\Omega}} f_{i,j}^t \leqslant 0.$$

(2) The patches inside the group is attracted by the group, or

$$\forall i \in \mathbf{\Omega}, \sum_{j\in\mathbf{\Omega}\setminus\{i\}} f_{i,j}^t \geqslant 0.$$

In our problem, we expect each stable group corresponds to a target complex motion and we can segment them one by one.

4.2. Solution of one stable group

To find one stable group, we define the inverse-potential function of the group $\mathbf{\Omega}$ as

$$S(\mathbf{\Omega}) = \sum_{i\in\mathbf{\Omega}, j\in\mathbf{\Omega}} f_{i,j}^t,$$

and the solution to the following optimization problem gives a stable group:

$$\mathbf{\Omega}^* = \arg_{\mathbf{\Omega}} \max_{\mathbf{\Omega}\subseteq\mathbf{P}} S(\mathbf{\Omega}), \tag{1}$$

which means, the group that gives the smallest potential is the most stable one, and the proof is given in A.

Assume we want to segment frame t, and denote the segmentation label of patch i at frame t by $x_i^t = \pm 1$. If $x_i^t = 1$, patch i should be segmented out, and *vice versa*.

We convert the subset optimization in Eq. (1) to an indicator optimization problem. For the patches of frame t, given a group $\mathbf{\Omega}$, let $\mathbf{x} = \{x_i^t\}_{i=1}^N$. As the value of $\mathbf{x}$ and group $\mathbf{\Omega}$ are one-to-one correspondence, Eq. (1) can be equivalently rewritten as

$$\max_{\mathbf{x}} \sum_{\left(x_i^t=1, x_j^t=1\right)} f_{i,j}^t, \tag{2}$$
$$\text{s.t.} \quad x_k^t = \pm 1, \quad k = 1,\ldots,N.$$

The objective function only has pairwise terms, so it can be optimized by graph cuts [23]. Converting Eq. (2) to an energy minimization problem for graph cuts is straightforward: we define the pairwise energy term as

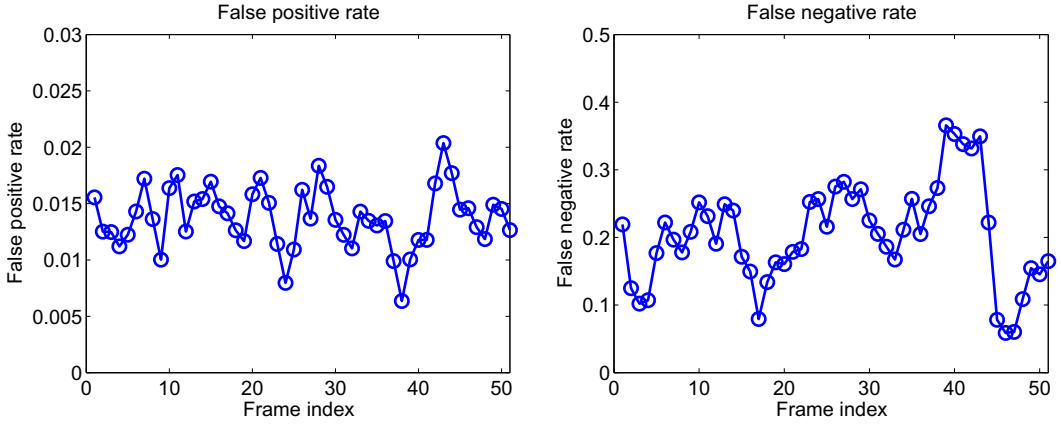

Fig. 11. False positive rate and false negative rate versus frame index (time) when the background is fixed. The false positive rate does not fluctuate much (±0.01 at most), but the false negative rate raises up a lot at about frame 35, which corresponds to the time when the person walks into the homogeneous background (the white wall).

Please cite this article in press as: J. Xu et al., Learning spatio-temporal dependency of local patches for complex motion segmentation, Comput. Vis. Image Understand. (2011), doi:10.1016/j.cviu.2010.11.010

$$E^{i,j}\left(x_i^t, x_j^t\right) = \begin{cases} -f_{i,j}^t, & x_i^t = 1, x_j^t = 1, \\ 0, & \text{otherwise}, \end{cases}$$

and define the energy function to minimize as

$$E(x_1^t, x_2^t, \ldots) = \sum_{i<j} E^{i,j}\left(x_i^t, x_j^t\right), \tag{3}$$

then minimizing Eq. (3) is equivalent to optimizing Eq. (2). According to [23], if all the energy terms satisfies the inequality

$$E^{i,j}(0,0) + E^{i,j}(1,1) \leqslant E^{i,j}(0,1) + E^{i,j}(1,0),$$

the graph cuts algorithm can obtain the global optimum. Otherwise, the optimization is NP-hard. We notice that when $f_{i,j}^t = -1$, the inequality above is not satisfied. Therefore, our optimization is an NP-hard problem.

We relax Eq. (2) to make the optimization feasible. We relax each dimension of the indicator vector x_i^t from $\{-1, +1\}$ to be real-valued, but keep the L_2 norm of $\mathbf{x}$, *i.e.*, $\|\mathbf{x}\| = \sqrt{N}$. Using the fact that $\frac{\mathbf{1}+\mathbf{x}}{2}$ is the indicator vector for $x_i = 1$, where $\mathbf{1}$ is the all-one vector with proper dimension. We rewrite Eq. (2) by relaxing its constraints as

$$\begin{aligned} &\max_{\mathbf{x}} \quad (\mathbf{1}+\mathbf{x})^T \mathbf{F} (\mathbf{1}+\mathbf{x}), \\ &\text{s.t.} \quad \mathbf{x}^T\mathbf{x} = N \end{aligned} \tag{4}$$

where $\mathbf{F}$ is the force matrix of the system, and its element at position (i, j) is $f_{i,j}^t$. The diagonal elements $f_{i,i}^t = 0$, as one patch cannot attract itself.

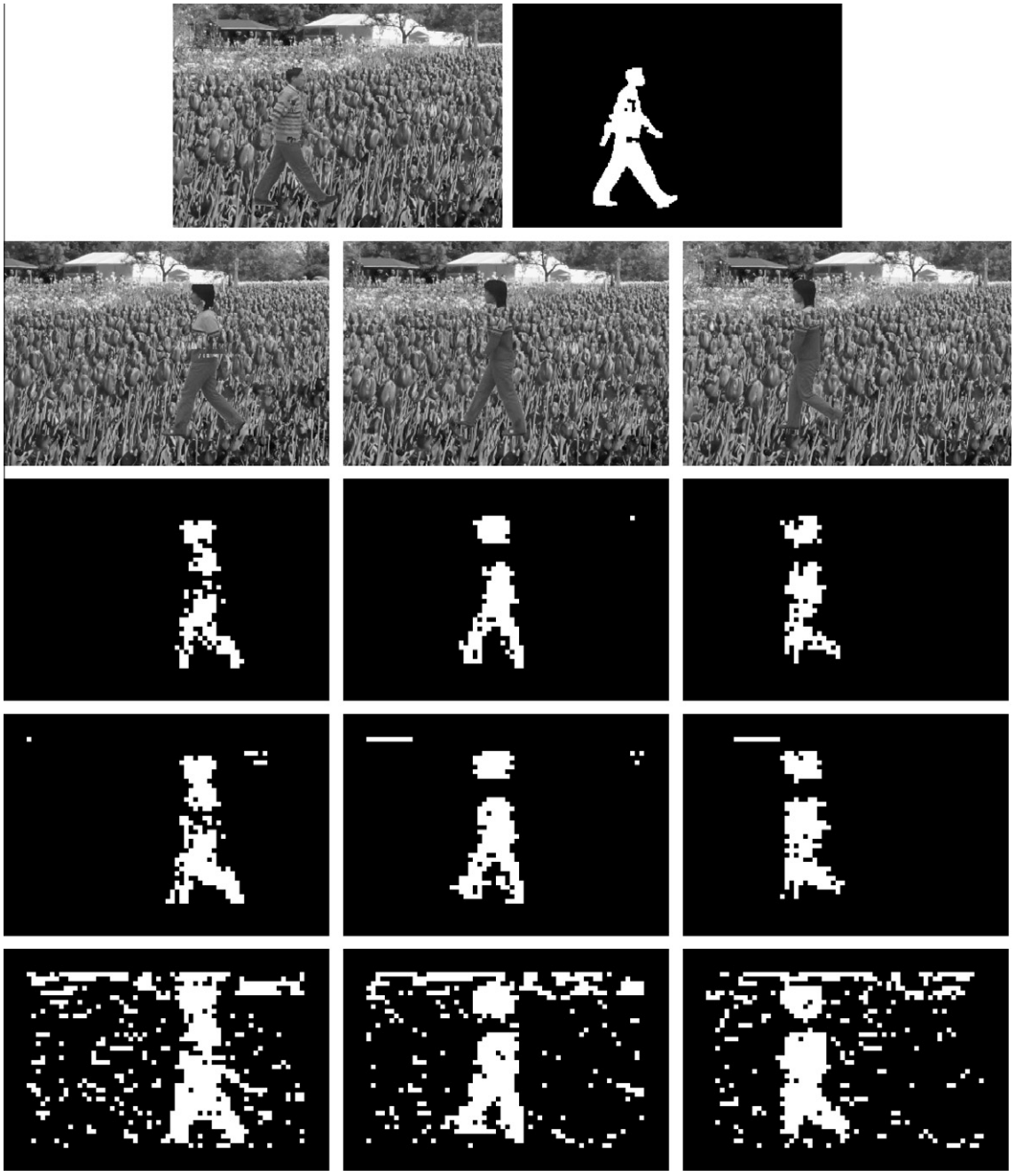

Fig. 12. Segmentation of human walking with moving background. The background is a moving picture. Row 1: some input frames. Row 2: result of bipolar segmentation. Row 3: result of QPBOI. Row 4: the first segment of normalized cut (totally cut into four parts).

Please cite this article in press as: J. Xu et al., Learning spatio-temporal dependency of local patches for complex motion segmentation, Comput. Vis. Image Understand. (2011), doi:10.1016/j.cviu.2010.11.010

Solving the optimization in Eq. (4) is still difficult, as it cannot be converted to some eigenvector-related optimization. Therefore, we look at the Lagrangian

$$\mathscr{L}(\mathbf{x},\lambda) = (\mathbf{1}+\mathbf{x})^T\mathbf{F}(\mathbf{1}+\mathbf{x}) - \lambda(\mathbf{x}^T\mathbf{x} - N).$$

Taking the derivative and let $\frac{\partial \mathscr{L}(\mathbf{x},\lambda)}{\partial \mathbf{x}} = 0$, $\frac{\partial \mathscr{L}(\mathbf{x},\lambda)}{\partial \lambda} = 0$, we obtain

$$\begin{aligned} &\mathbf{F}(\mathbf{1}+\mathbf{x}^*) = \lambda\mathbf{x}^*, \\ &\mathbf{x}^{*T}\mathbf{x}^* = N. \end{aligned} \tag{5}$$

This is a necessary condition.

Solving Eq. (5) explicitly is impractical, because the order can be very high. Therefore, we design a fixed-point iteration solution to solving all the $N+1$ unknowns. We denote by

$$\mathbf{y}^* = \mathbf{F}(\mathbf{1}+\mathbf{x}^*),$$

then from Eq. (5), we obtain

$$\begin{aligned} &\mathbf{y}^* = \lambda\mathbf{x}^*, \\ &\mathbf{y}^{*T}\mathbf{y}^* = (\lambda\mathbf{x}^*)^T\lambda\mathbf{x}^* = \lambda^2 N. \end{aligned}$$

Based on the three equations above, we obtain our iterative method to solve Eq. (5) by going through the following iterations:

$$\begin{aligned} &\mathbf{y}(\tau+1) = \mathbf{F}\mathbf{1} + \mathbf{F}\mathbf{x}(\tau), \\ &\lambda(\tau+1) = \sqrt{\mathbf{y}(\tau+1)^T\mathbf{y}(\tau+1)/N}, \\ &\mathbf{x}(\tau+1) = \mathbf{y}(\tau+1)/\lambda(\tau+1). \end{aligned}$$

When the number of patches is large, the matrix $\mathbf{F}$ becomes huge. Fortunately, the matrix $\mathbf{F}$ is usually very sparse, because we only consider a small fraction of all forces. Another good thing is that we only care about the relative value of the indicator vector $\mathbf{x}$ to a threshold, so the precision requirement of $\mathbf{x}$ is low. In our implementation, we terminate the iteration when

$$\frac{\|\mathbf{x}(\tau) - \mathbf{x}(\tau-1)\|}{\|\mathbf{x}(\tau)\|} \leqslant \varepsilon,$$

and the threshold ε can be set to a robust not-so-small value because of the low precision requirement of $\mathbf{x}$. In our experiments, we set $\varepsilon = 0.1$.

We set $\mathbf{x}(0)$ to be an all-one vector and start the iteration. When the iteration terminates, we segment out all patches i that satisfies $x_i^t > \eta$. We determine η by trying some possible η values in $[-1, 1]$, and select the optimal one that can maximize Eq. (1). There exists a trivial solution for Eq. (5) when $\mathbf{x} = -1$ and $\lambda = 0$. However, in our real and synthesized cases, we have not encountered this case.

After some patches are segmented out, we continue our algorithm on the remaining patches until the number of remaining patches is less than a threshold. In most of our experiments, we choose the threshold to be 2% of the total number of patches.

5. Experiments

We design three types of experiments: (1) Numerical experiments to compare our bipolar segmentation to QPBOI optimization [45]. (2) Motion segmentation comparison on different training data sets. (3) Complex motion segmentation. All experiments are run on a desktop PC, with Intel Core 2 (TM) CPU 2.40 GHz and 3.0 GB of RAM. We implement our algorithm using Matlab.

5.1. Comparison between bipolar segmentation and QPBOI

To show the advantages of our method, we compare our method with the quadratic pseudo-Boolean optimization improved (QPBOI) [45] algorithm, which is the state-of-the-art algorithm to find approximate solution in the graph cut problem where the energy function does not satisfy the sub-modular condition, as in our Eq. (2). We synthesize a force matrix $\mathbf{F}$ to test (1) how much better our solution can achieve and (2) how much faster it can be. Because QPBOI requires integer parameters to obtain good results, we let $f_{i,j} = \{-1, 0, 1\}$. The QPBOI implementation is obtained from http://www.robots.ox.ac.uk/ojw/software.htm.

In Fig. 7, we compare the performance and speed between our algorithm and QPBOI. From the figure, it is clear that bipolar segmentation can find a much better solution, *i.e.*, a much larger maximization of the objective function, and converges much faster than QPBOI algorithm.

We test the two algorithms in different synthesized data sets. The top row of Fig. 7 shows the results of the first set of synthesized data. In these experiments, we synthesize different number of patches from 100 to 1000. Then we randomly choose 10% patch pairs and assign their $f_{i,j} = +1$, and another 10% of the patch pairs and assign their force $f_{i,j} = -1$. For performance evaluation, we run 10 times and report the average of the optimized value for bipolar segmentation and QPBOI, and the average running time. The top-left figure in Fig. 7 compares the maximization value found by the two algorithms. When the number of patches is 100, the solution found by bipolar segmentation is about 10 times larger than that found by QPBOI. If we increase the number of patches, bipolar segmentation performs much better than QPBOI: when the number of patches is 1000, the solution found by bipolar segmentation is more than 100 times larger than that of QPBOI.

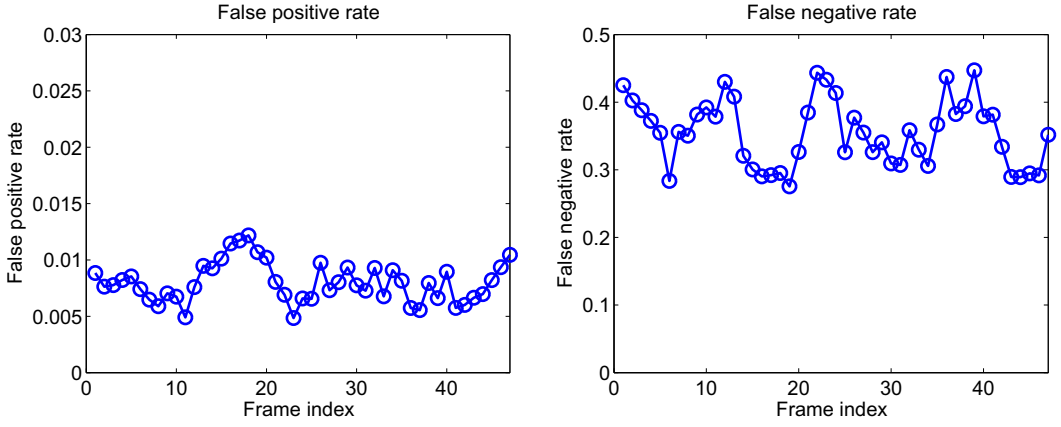

Fig. 13. False positive rate and false negative rate versus frame index (time) when the background is moving. Comparing to the experiment with static background, the false positive rate and false negative rate does not fluctuate much, because the background does not have large homogeneous regions as the previous experiment.

Please cite this article in press as: J. Xu et al., Learning spatio-temporal dependency of local patches for complex motion segmentation, Comput. Vis. Image Understand. (2011), doi:10.1016/j.cviu.2010.11.010

Moreover, the top-right figure in Fig. 7 compares the running time of both algorithms. We observe that both algorithms are efficient when the number of patches is small. For example, both algorithms converge within about 0.2 s to optimize a graph with 200 patches. However, when the number of patches increases, bipolar segmentation becomes much faster than QPBOI. For example, in the case of 1000 patches, bipolar segmentation finishes within about 1 s, while QPBOI needs about 10 s.

The bottom row in Fig. 7 shows the results of the second synthesized experiments. In this case, we randomly choose 5% of the patch pairs and assign their $f_{i,j} = +1$, and 20% of the patch pairs with $f_{i,j} = -1$. We keep the rest of the pairs with $f_{i,j} = 0$. Similar to the previous experiment, the solution found by bipolar segmentation is more than 10 times larger than that by QPBOI. When the number of patches is small, *e.g.*, 100 or 200 patches, QPBOI runs faster than bipolar segmentation. But when the number of patches increases, QPBOI becomes more than 10 times slower than bipolar segmentation. From the experiments above, we conclude that for optimizing our objective function in Eq. (2), bipolar segmentation is much better than QPBOI, and is scalable to a large number of patches.

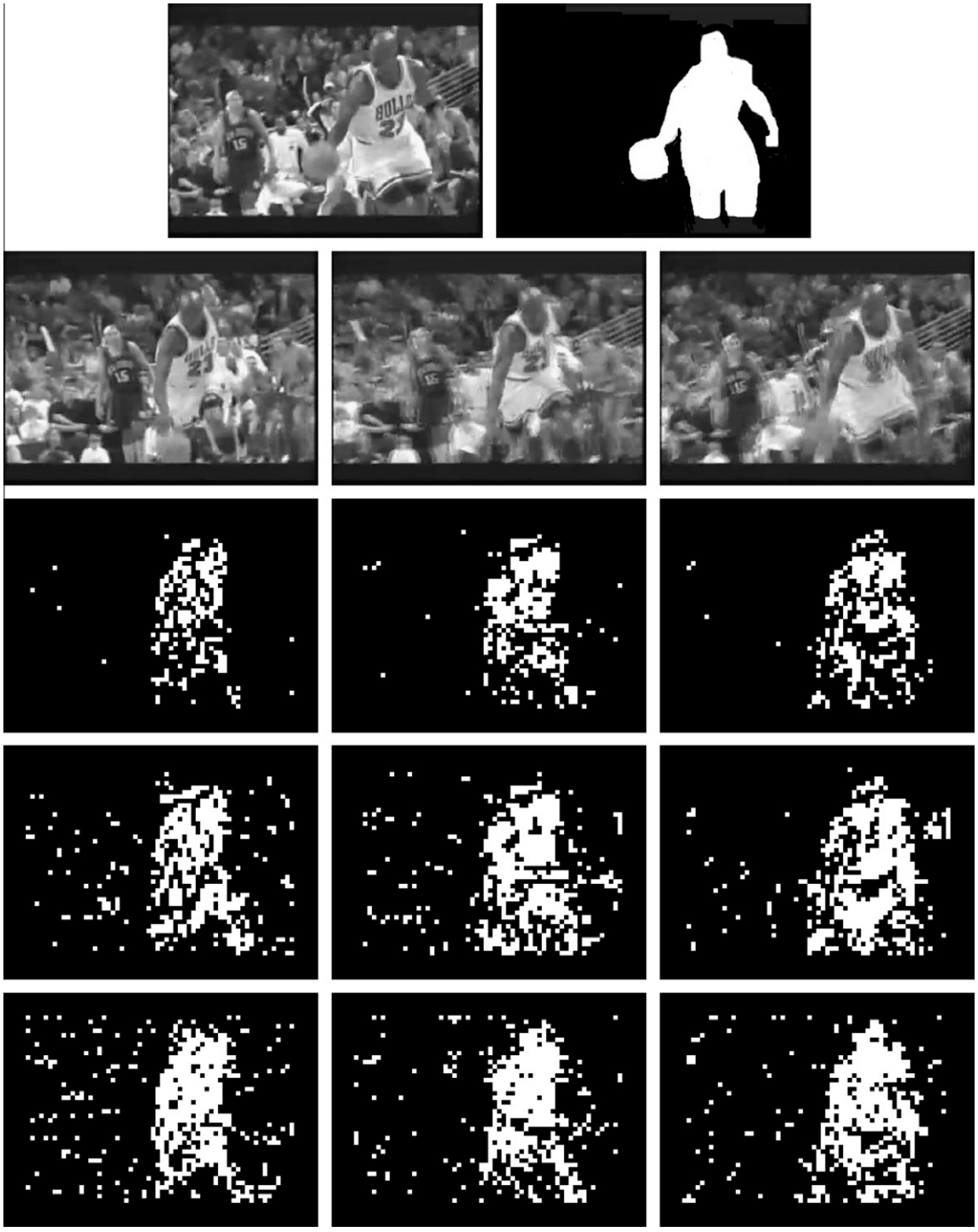

Fig. 14. Segmentation of basketball playing. The background is full of moving audiences. Row 1: some input frames. Row 2: result of bipolar segmentation. Row 3: result of QPBOI. Row 4: the third segment of normalized cut (totally cut into four parts).

Please cite this article in press as: J. Xu et al., Learning spatio-temporal dependency of local patches for complex motion segmentation, Comput. Vis. Image Understand. (2011), doi:10.1016/j.cviu.2010.11.010

5.2. Motion segmentation comparison on different training data

5.2.1. Experimental settings

The image resolution of both training and testing video is 352×240. For each frame, we decompose it into non-overlapping patches of size 5×5. There are in total 2280 valid patches for each frame, excluding those boundary patches that cannot extract the motion profiles. It takes 40 s to extract all of the 2280 motion profiles with our Matlab implementation. Among the 2280 patches, we randomly select 1000 patch pairs (500 positive, 500 negative) for training. It costs 3 s to obtain the motion profile symmetry correlations. We then use PCA to reduce the dimension by keeping 95% energy, and use an SVM to train an RBF-kernel SVM classifier. We perform grid search [42] to select the best parameters for the SVM.

In the segmentation process, we first extract the motion profiles of the patches from the testing video, then randomly select patch pairs and estimate $f_{i,j}$ between them. To speed up, we control the total number of patch pairs and each patch is selected 50 times on average. If a patch pair is not selected, we set its $f_{i,j} = 0$. Such a treatment will have the least effect on our formulation in Eq. (2), in which the forces are additive.

In bipolar segmentation, there are 57,000 non-zero $f_{i,j}$ for the 2280×2280 matrix. Our iteration terminates after less than 10 rounds and only costs about 0.2 s for each frame. After one object is segmented out, we keep segmenting the remaining patches until the number of the remaining patches is less than 2% of the total number of the patches.

To evaluate the performance of our algorithm, we compare the segmented object to the labeled ground truth. We use two

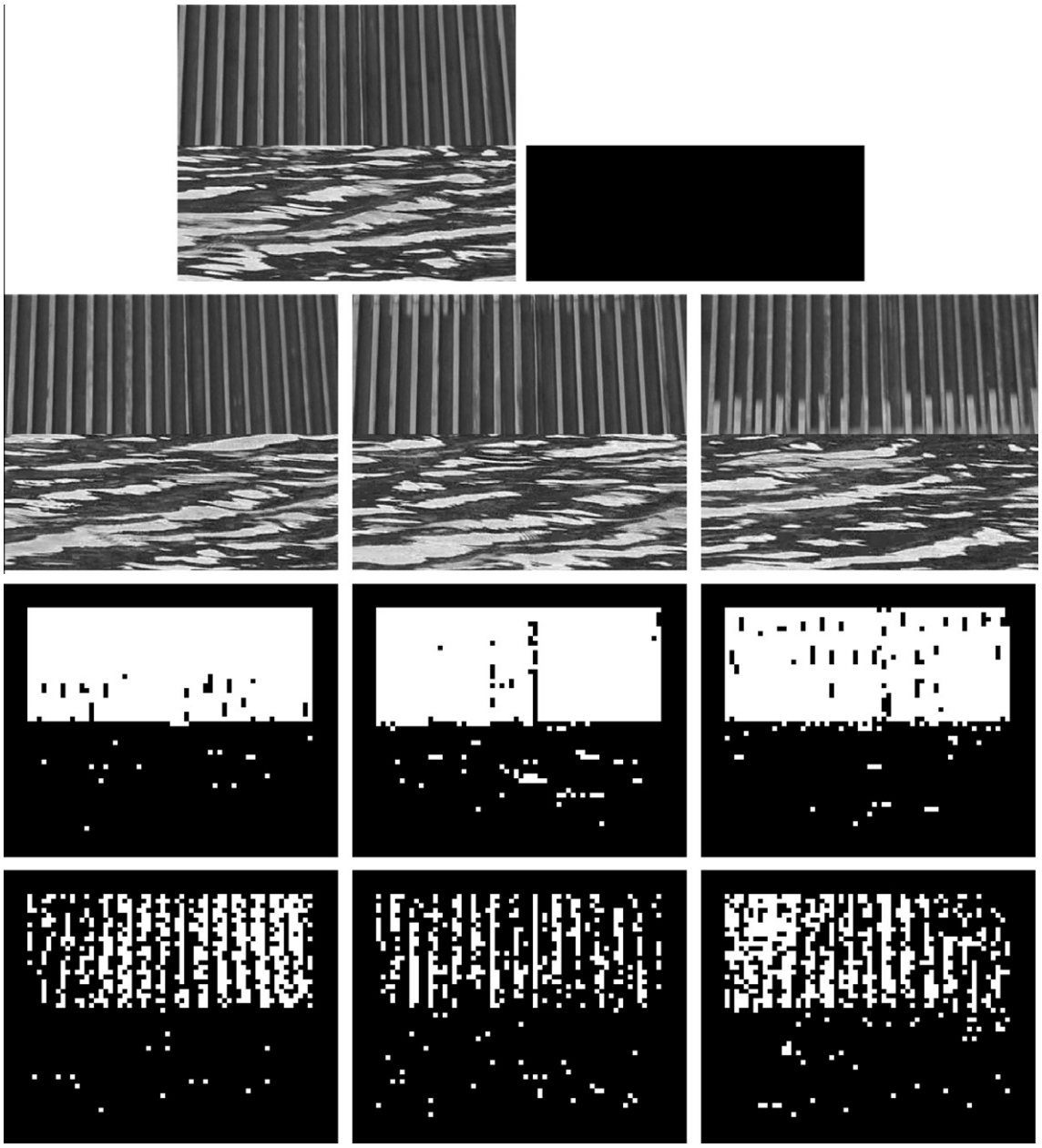

Fig. 15. Experimental result of segmenting two dynamic textures: moving escalator and waving water. Our method performs well, but normalized cut fails in this case. Row 1: the left one is the original frame for training, and the right one is the labeled frame, where the white region (upper half) is the foreground object. Row 2: testing frames. Row 3: segmentation result of bipolar segmentation. Row 4: segmentation result of normalized cut (totally cut into four parts, and the part which is correspondent to the coherent segment is posted).

Please cite this article in press as: J. Xu et al., Learning spatio-temporal dependency of local patches for complex motion segmentation, Comput. Vis. Image Understand. (2011), doi:10.1016/j.cviu.2010.11.010

quantities for evaluation: (1) false positive rate, which is the number of patches falsely segmented as the object, divided by the number of true patches that belong to the object. (2) False negative rate, which is the number of missed patches, divided by the number of patches that belong to the object.

5.2.2. Segmentation of opposite motion

To test if our algorithm can learn the complex motion, we use the opposite motion as an example, and want to see if different learning (ground truth labeling) may produce different segmentation results. We synthesized a 48-frame video for our experiment. In the synthesized video, there are two moving boxes. One is on the upper side, moving from left to right. The other is on the lower side, moving from the right to the left. The background is fixed.

We test our bipolar segmentation in two different scenarios: (1) we label the two boxes as one object and (2) we label the two boxes as two different objects. In the first setting, the algorithm segments the background first, and then segments the two boxes as a whole. We obtain the false positive rate 4.51%, and the false negative rate 14.22%. In the second setting, the algorithm always segments the background first, and then segments the larger boxes, followed by the smaller one. The false positive rate is 2.42%, and the false negative rate is 1.34%. The comparison is shown in Fig. 8.

From the experiment result in the first scenario, we observe that the opposite motion coherence can be learnt and adopted in segmentation. By comparing the results in the first and the second scenarios, we observe that different learnings have different effects. There are two major reasons to explain the good performance: (1) the background is static and the motion is relatively simple and (2) our method can learn the opposite motion coherence. On the contrary, if we use original motion profile feature in [8], it cannot segment two opposite-moving boxes together.

Fig. 9 shows the iteration process of the second round of segmentation in the second scenario. Initially, both boxes are assigned by the same value. During the iterations, the value corresponding to the larger box becomes larger and larger. After six iterations, the algorithm terminates and the larger box is segmented out. This example also shows that our algorithm converges fast.

5.3. Segmentation results of complex motion

5.3.1. Motion segmentation of articulated motion

We test our segmentation algorithm on the human articulation, to see if our approach can segment walking people from videos. We use the Dataset A of CASIA Gait Database [46]. In this dataset, there are people walking in a static background. The length of each video is about 50 frames, and the ground truth is provided in the database. We convert the original videos to grayscale videos. To learn the human walking coherence, we use eight frames in one sequence for training. After that, we use the rest sequences to evaluate the segmentation. In the training video, the person walks from left to right. In the testing video, the person walks from the right to the left. We test our algorithm on the original dataset first. On average, the false positive rate is 1.392%, and the false negative rate is 21.59%. Some segmentation examples are shown in Fig. 10.

We observe that the performance is not as good as the result of segmenting opposite motion. The reasons are: (1) articulated motion is much more complex than the opposite motion and (2) the background in the human walking videos has large pieces of homogeneous regions, and our motion feature cannot work well in those regions. For example, in the third frame of Fig. 10, the shoulder part of the person is not segmented out. This is because the shoulder is in the same color as its local background: one is with white clothes and the other is a white wall. The white wall is a large homogeneous region, so our motion profile symmetry correlation feature cannot distinguish the motion in that region well. The false positive rate and false negative rate of every frame are shown in Fig. 11. We observe that the false negative rate raises up a lot at about frame 35, which corresponds to the time when the person walks into the homogeneous background (the white wall).

To make the articulated motion segmentation a more challenging task, we substitute the background by a moving picture to make a "moving background" effect, as if the camera is following the person. We perform the same learning and testing procedure, and some segmentation results are shown in Fig. 12.

To further validate our method, we also compare our method with other methods. The first method to compare is the normalized cut. As our feature incorporates the learnt information, it is unfair to compare to normalized cut that uses normal motion profile. Therefore, we map our forces $f_{i,j}$ back to $(0,1)$ as similarity, then apply normalized cut using the similarity. The second method to compare is the quadratic pseudo-Boolean optimization improved (QPBOI) [45] algorithm.

From the figures shown in Fig. 12, we observe that our bipolar segmentation performs the best, much better than QPBOI's result. The normalized cut simply fails if we convert the $f_{i,j}$ from $(-1,1)$ to $(0,1)$. Using our algorithm, the false positive rate is 0.8%, and the false negative rate is 35.4% on average. The false positive rate and false negative rate of every frame are shown in Fig. 13. We observe that comparing to the previous experiment, the average false

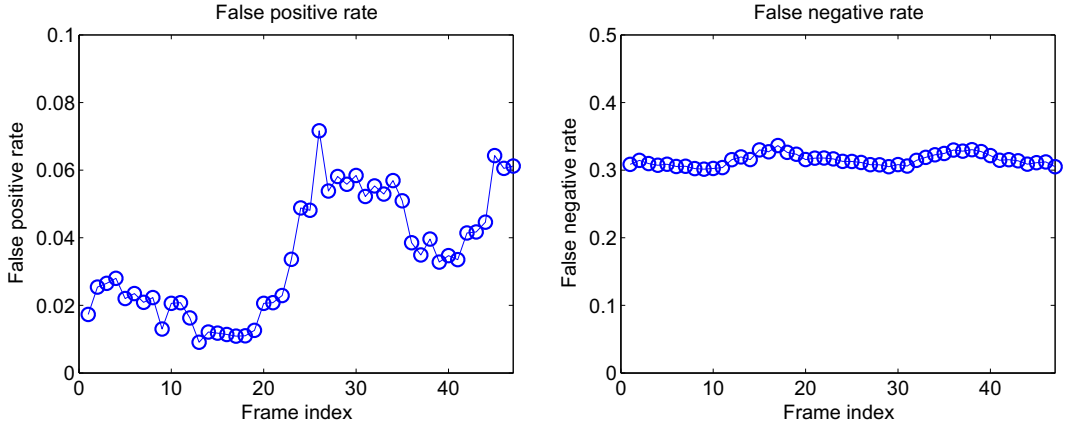

Fig. 16. False positive rate and false negative rate versus frame index (time) of our algorithm in segmenting moving escalator and waving water dynamic textures. The false positive rate is always less than 0.1, and the false negative rate does not fluctuate much.

Please cite this article in press as: J. Xu et al., Learning spatio-temporal dependency of local patches for complex motion segmentation, Comput. Vis. Image Understand. (2011), doi:10.1016/j.cviu.2010.11.010

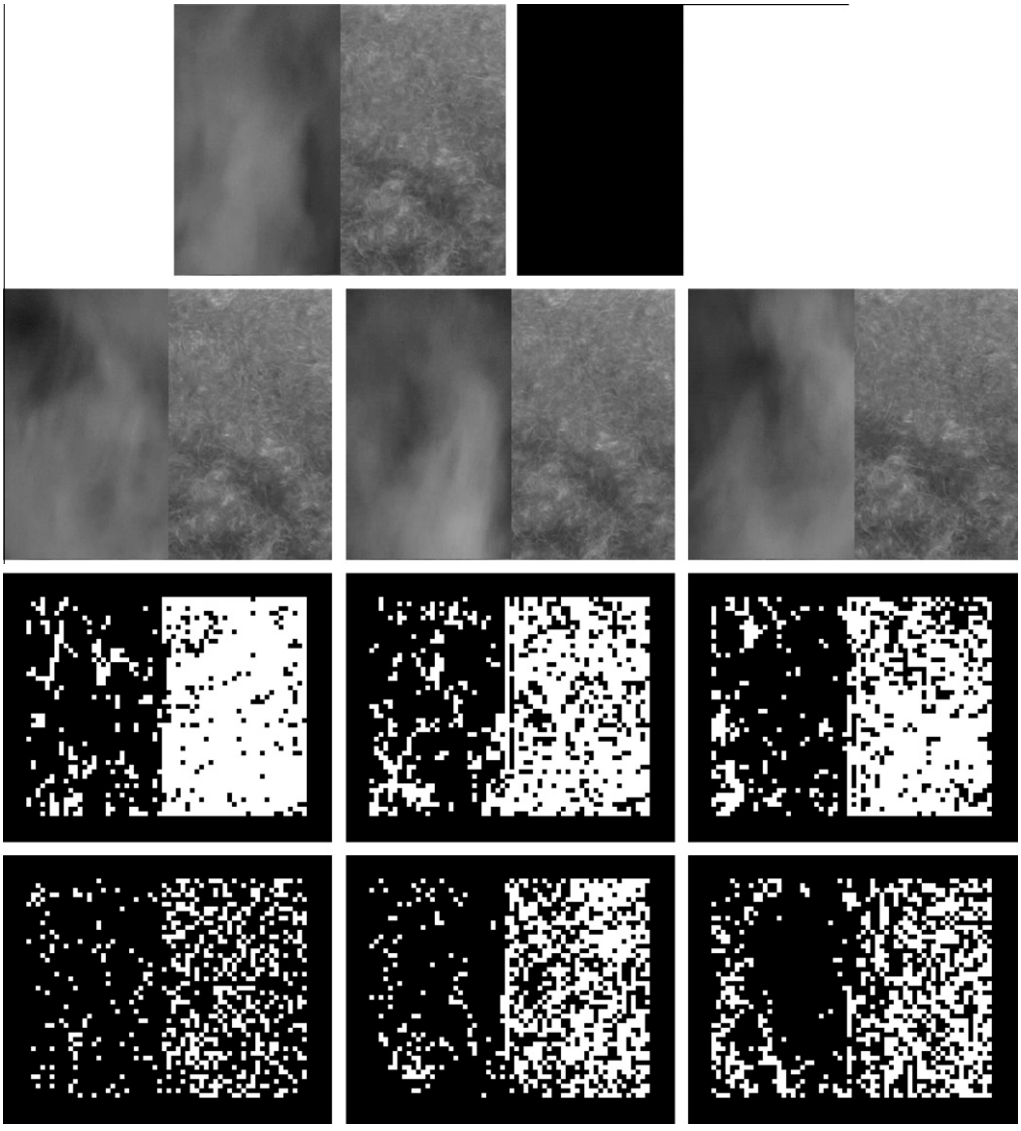

Fig. 17. Experimental result of segmenting two dynamic textures: flying smoke and oscillating material. Our method performs better than normalized cut. Row 1: the left one is the original frame for training, and the right one is the labeled frame, where the white region (right half) is the foreground object. Row 2: testing frames. Row 3: segmentation result of bipolar segmentation. Row 4: segmentation result of normalized cut (totally cut into four parts, and the part which is correspondent to the coherent segment is posted).

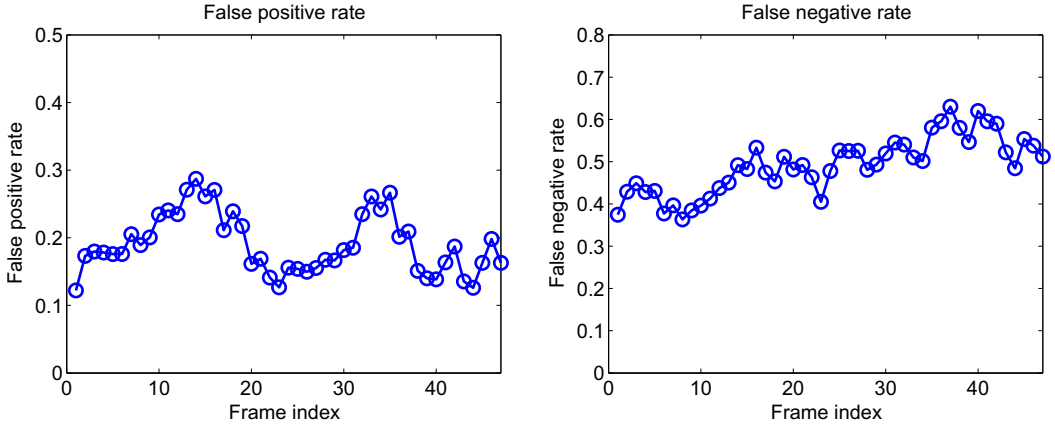

Fig. 18. False positive rate and false negative rate versus frame index (time) of our algorithm in segmenting flying smoke and oscillating material dynamic textures. As the smoke pattern is not so stable, the result is not as good as the one in the previous experiment.

Please cite this article in press as: J. Xu et al., Learning spatio-temporal dependency of local patches for complex motion segmentation, Comput. Vis. Image Understand. (2011), doi:10.1016/j.cviu.2010.11.010

positive rate decreases, while the false negative rate increases. The variations of false positive rate and false negative rate are both decreased, because there is no large homogeneous region in the background.

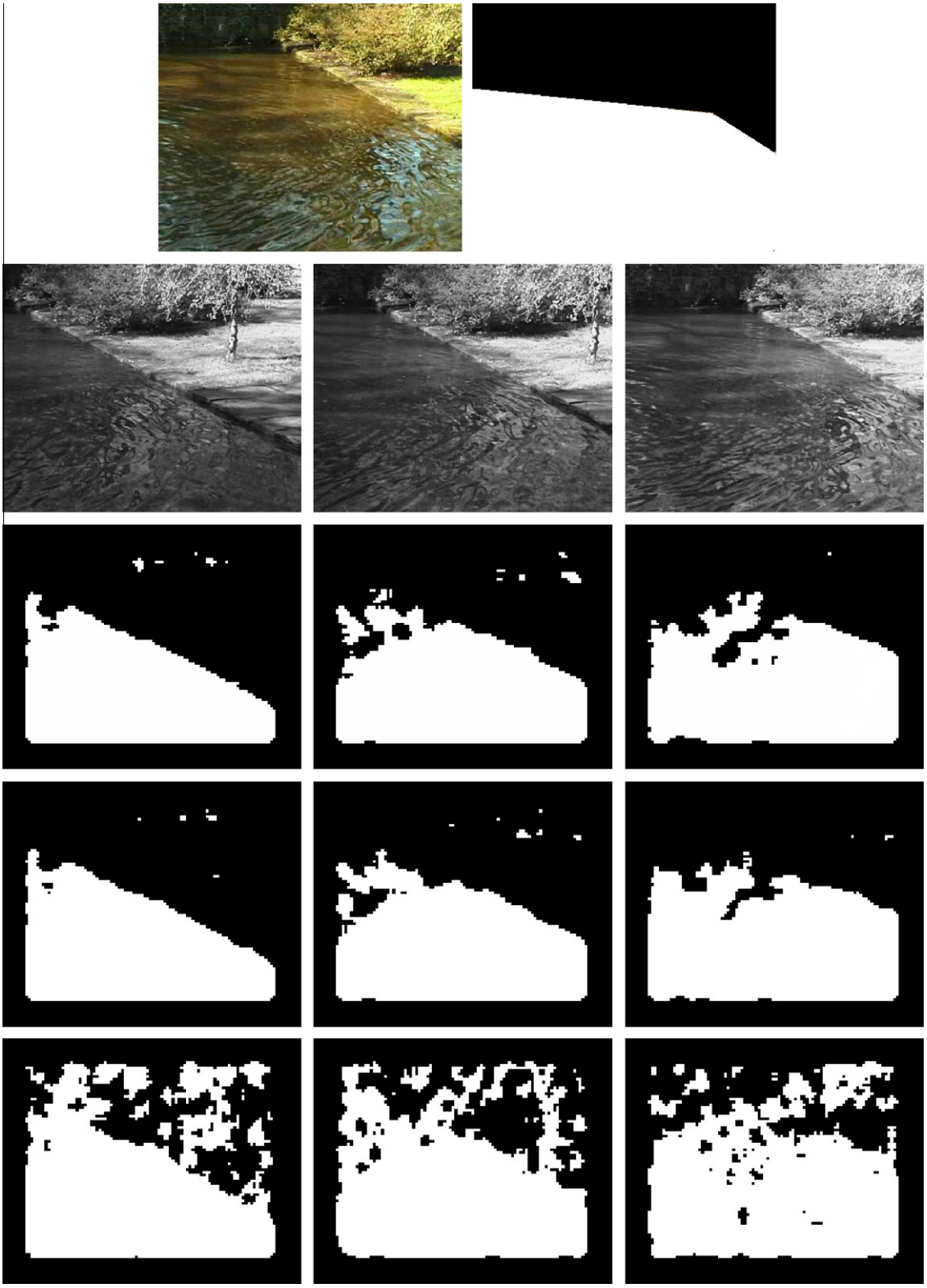

Fig. 19. Experimental result of segmenting waving water. The camera is moving, and the water nearby is waving, while the water far away is static. Our method and QPBOI performs well: they can distinguish the waving water, but mis-segment some waving foliage as well. Normalized cut fails in this case. Row 1: the left one is the original frame for training, and the right one is the labeled frame, where the white region is the foreground object. Row 2: testing frames. Row 3: segmentation result of bipolar segmentation. Row 4: segmentation result of QPBOI. Row 5: segmentation result of normalized cut (totally cut into four parts, and the part which is correspondent to the coherent segment is posted).

Please cite this article in press as: J. Xu et al., Learning spatio-temporal dependency of local patches for complex motion segmentation, Comput. Vis. Image Understand. (2011), doi:10.1016/j.cviu.2010.11.010

Some articulated motions do not have consistent patterns, for example, basketball players always change their motion pattern. The performance of our algorithm is degraded in that situation (Fig. 14).

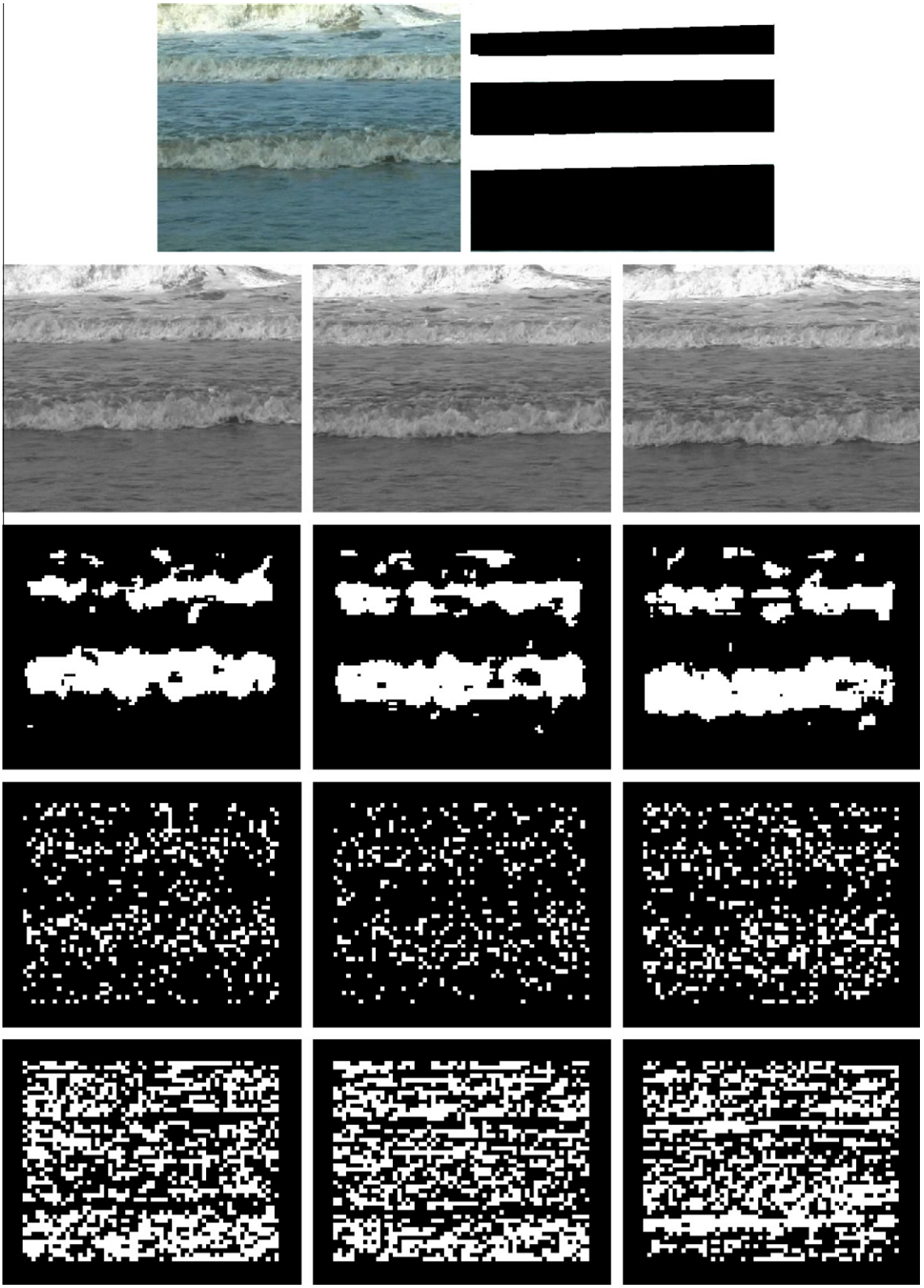

Fig. 20. Segmentation of wave fronts. Row 1: the left one is the original frame, and the right one is the labeled frame. Row 2: testing frames. Row 3: segmentation result of bipolar segmentation. Row 4: segmentation result of QPBOI. Row 5: segmentation result of normalized cut (totally cut into four parts, and no part looks like the wave fronts). Because we cannot compute motion profiles at the boundary, the top boundary region is neglected in the segmentation algorithm. Thus the top foreground regions are largely missed in all of three scenes.

Please cite this article in press as: J. Xu et al., Learning spatio-temporal dependency of local patches for complex motion segmentation, Comput. Vis. Image Understand. (2011), doi:10.1016/j.cviu.2010.11.010

5.3.2. Motion segmentation of dynamic textures

As another example of segmenting complex motion patterns, we test our algorithm on dynamic textures, and the dataset is from DynTex [47].

In this set of experiments, we only manually label one frame for learning, and the learning/testing procedure is the same as previous experiments.

In the first set of experiments, we synthesize dynamic texture videos by concatenating two different dynamic textures. Fig. 15 shows some synthesized frames, which are composed of escalator steps (upper part) and water waves (lower part). We can easily obtain the false positive rate and false negative rate of our algorithm for each frame using the synthetic data. Fig. 15 shows the experiment results of three frames using our method and normalized cut. From the results, we observe that our method works well, but the normalized cut cannot perform well. The false positive rate and false negative rate of our algorithm are given in Fig. 16.

In the second experiment, we segment the smoke and oscillating material. Fig. 17 shows some synthesized frame and the corresponding experiment results using our method and normalized cut. The result of our method is better than normalized cut, and the false positive rate and false negative rate of our algorithm are given in Fig. 18. This set of results is not as good as the previous one, because the smoke pattern is not so stable as the escalator, or the waving water.

In the next set of experiments, we try to segment one type of dynamic texture from its real background. Fig. 19 shows a challenging example to segment waving water, with a moving camera. We label one frame of waving water for learning, and use other frames for testing. From the results, we observe that by using the pairwise relations, both our method and QPBOI can segment the waving water out, while normalized cut fails again. Fig. 20 shows another challenging example: segmenting out wave fronts. We label only one frame for learning, and use other frames for testing. Again, in this example, our method performs better than QPBOI and normalized cut.

6. Conclusions and future work

Learning motion priors can help segment complex motions. By using our motion profile feature to characterize the relative motion, our method is able to learn the pairwise relationship between image patches, and model the complex motion. The experiments show that our method can correctly learn the motion coherence and segment different types of complex motions from extremely noisy backgrounds, includes articulated motion and non-rigid object motion. Compared with conventional graph cut algorithm, such as QPBOI, the proposed bipolar segmentation algorithm provides better solution with faster convergence. Such a bipolar segmentation method is generally applicable to other applications.

The performance of our learning-based algorithm largely depends on the reliability of inferring the pairwise relationship. Besides motion feature, we will explore other types of features to learn the pairwise relationship of local patches, including appearances, texture, to improve the learning of pairwise relation and the segmentation results.

Acknowledgments

We thank the anonymous reviewers for their helpful comments. This work was supported in part by National Science Foundation IIS-0347877, IIS-0916607, and US Army Research Laboratory and the US Army Research Office under Grant ARO W911NF-08-1-0504. This project is partly supported by the Nanyang Assistant Professorship (SUG M58040015) to Junsong Yuan.

Appendix A. Proof by contradiction

If $\exists k \notin \mathbf{\Omega}^*, \sum_{j\in\mathbf{\Omega}} f_{kj} > 0$, then we construct $\mathbf{\Omega}' = \mathbf{\Omega}^* \cup \{k\}$, and obtain

$$S(\mathbf{\Omega}') = \sum_{i\in\mathbf{\Omega}', j\in\mathbf{\Omega}'} f_{ij} = \sum_{i\in\mathbf{\Omega}^*, j\in\mathbf{\Omega}^*} f_{ij} + \sum_{j\in\mathbf{\Omega}^*} f_{kj} > \sum_{i\in\mathbf{\Omega}^*, j\in\mathbf{\Omega}^*} f_{ij} = S(\mathbf{\Omega}^*).$$

This contradicts $\mathbf{\Omega}^*$ is the optimal solution of Eq. (1).

If $\exists k \in \mathbf{\Omega}^*, \sum_{j\in\mathbf{\Omega}^*\setminus\{k\}} f_{kj} < 0$, then we construct $\mathbf{\Omega}' = \mathbf{\Omega}^*\setminus\{k\}$, and obtain

$$S(\mathbf{\Omega}') = \sum_{i\in\mathbf{\Omega}', j\in\mathbf{\Omega}'} f_{ij} = \sum_{i\in\mathbf{\Omega}^*, j\in\mathbf{\Omega}^*} f_{ij} - \sum_{j\in\mathbf{\Omega}^*\setminus\{k\}} f_{kj} > \sum_{i\in\mathbf{\Omega}^*, j\in\mathbf{\Omega}^*} f_{ij} = S(\mathbf{\Omega}^*).$$

This contradicts $\mathbf{\Omega}^*$ is the optimal solution of Eq. (1).

References

[1] B. Lucas, T. Kanade, An iterative image registration technique with an application to stereo vision, in: International Joint Conference on Artificial Intelligence, vol. 81, 1981, pp. 674–679.
[2] J. Wang, E. Adelson, Layered representation for motion analysis, in: CVPR, 1993.
[3] Y. Wu, Z. Zhang, T. Huang, J. Lin, Multibody grouping via orthogonal subspace decomposition, in: CVPR, 2001.
[4] R. Vidal, Y. Ma, A unified algebraic approach to 2-D and 3-D motion segmentation and estimation, Journal of Mathematical Imaging and Vision 25 (3) (2006) 403–421.
[5] S. Ayer, H. Sawhney, Layered representation of motion video using robust maximum-likelihood estimation of mixture models and MDL encoding, in: ICCV, 1995.
[6] Y. Weiss, E. Adelson, A unified mixture framework for motion segmentation: incorporating spatial coherence and estimating the number of models, in: CVPR, 1996.
[7] Y. Weiss, Smoothness in layers: motion segmentation using nonparametric mixture estimation, in: CVPR, 1997.
[8] J. Shi, J. Malik, Motion segmentation and tracking using normalized cuts, in: ICCV, 1998.
[9] P. Torr, R. Szeliski, P. Anandan, An integrated Bayesian approach to layer extraction from image sequences, PAMI 23 (3) (2001) 297–303.
[10] X. Feng, P. Perona, Scene segmentation from 3D motion, in: CVPR, 1998, pp. 225–231.
[11] P. Sturm, B. Triggs, A factorization based algorithm for multi-image projective structure and motion, Lecture Notes in Computer Science 1065 (1996) 709–720.
[12] P. Beardsley, A. Zisserman, D. Murray, Sequential updating of projective and affine structure from motion, International Journal of Computer Vision 23 (3) (1997) 235–259.
[13] P. Torr, O. Faugeras, T. Kanade, N. Hollinghurst, J. Lasenby, M. Sabin, A. Fitzgibbon, Geometric motion segmentation and model selection [and discussion], Philosophical Transactions: Mathematical, Physical and Engineering Sciences 356 (1740) (1998) 1321–1340.
[14] R. Hartley, R. Vidal, The multibody trifocal tensor: motion segmentation from 3 perspective views, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, 2004.
[15] G. Lance, W. Williams, A general theory of classificatory sorting strategies: 1. Hierarchical systems, The computer journal 9 (4) (1967) 373.
[16] K. Cho, P. Meer, Image segmentation from consensus information, Computer Vision and Image Understanding 68 (1) (1997) 72–89.
[17] B. Horn, B. Schunck, Determining optical flow (1980).
[18] A. Spoerri, S. Ullman, The early detection of motion boundaries (1991).
[19] S. Geman, D. Geman, Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images, PAMI 6 (1984) 721–741.
[20] S. Kumar, M. Hebert, Discriminative random fields: a discriminative framework for contextual interaction in classification, in: ICCV, 2003.
[21] X. He, R. Zemel, M. Carreira-Perpinan, Multiscale conditional random fields for image labeling, in: CVPR, 2004.
[22] D. Karger, C. Stein, A new approach to the minimum cut problem, Journal of the ACM (JACM) 43 (4) (1996) 601–640.
[23] V. Kolmogorov, R. Zabih, What energy functions can be minimized via graph cuts?, IEEE Transactions on Pattern Analysis and Machine Intelligence (2004) 147–159
[24] J. Shi, J. Malik, Normalized cuts and image segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence (2000) 888–905.
[25] S. Sarkar, K. Boyer, Quantitative measures of change based on feature organization: eigenvalues and eigenvectors, in: CVPR, 1996, pp. 478–483.
[26] P. Perona, W. Freeman, A factorization approach to grouping, in: ECCV, 1998, pp. 655–670.
[27] Y. Weiss, Segmentation using eigenvectors: a unifying view, in: ICCV, vol. 2, 1999.
[28] L. Zelnik-Manor, P. Perona, Self-tuning spectral clustering, Advances in Neural Information Processing Systems 17 (1601-1608) (2004) 16.

[29] M. Leordeanu, M. Hebert, A spectral technique for correspondence problems using pairwise constraints, in: ICCV, 2005.
[30] G. Doretto, D. Cremers, P. Favaro, S. Soatto, Dynamic texture segmentation, in: ICCV, vol. 2, 2003, pp. 1236–1242.
[31] A. Chan, N. Vasconcelos, Layered dynamic textures, PAMI 31 (10).
[32] X. Ren, J. Malik, Learning a classification model for segmentation, in: ICCV, vol. 1, 2003, pp. 10–17.
[33] D. Martin, C. Fowlkes, J. Malik, Learning to detect natural image boundaries using local brightness, color, and texture cues, PAMI (2004) 530–549.
[34] E. Borenstein, S. Ullman, Learning to segment, in: ECCV, 2004, pp. 315–328.
[35] Z. Tu, Auto-context and its application to high-level vision tasks, in: CVPR, 2008.
[36] M. Everingham, L. Van Gool, C.K.I. Williams, J. Winn, A. Zisserman, The PASCAL Visual Object Classes Challenge 2009 (VOC2009) Results, http://www.pascal-network.org/challenges/VOC/voc2009/workshop/index.html.
[37] B. Leibe, A. Leonardis, B. Schiele, Robust object detection with interleaved categorization and segmentation, International Journal of Computer Vision 77 (1) (2008) 259–289.
[38] N. Friedman, S. Russell, Image segmentation in video sequences: a probabilistic approach, in: Uncertainty in Artificial Intelligence: Proceedings of the Thirteenth Conference, 1997.
[39] C. Stauffer, W. Grimson, Learning patterns of activity using real-time tracking, IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (8) (2000) 747–757.
[40] S. Kamijo, K. Ikeuchi, M. Sakauchi, Segmentations of spatio-temporal images by spatio-temporal markov random field model, in: Proceedings of the Third International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition, 2001, p. 313.
[41] M. Pawan Kumar, P. Torr, A. Zisserman, Learning layered motion segmentations of video, IJCV 76 (3) (2008) 301–319.
[42] C.-C. Chang, C.-J. Lin, LIBSVM: a library for support vector machines, 2001.
[43] S. Yu, J. Shi, Segmentation with pairwise attraction and repulsion, in: ICCV, 2001.
[44] S. Yu, J. Shi, Grouping with directed relationships, Lecture Notes in Computer Science (2001) 283–297.
[45] C. Rother, V. Kolmogorov, V. Lempitsky, M. Szummer, Optimizing binary MRFs via extended roof duality, in: CVPR, 2007.
[46] Casia gait database, http://www.sinobiometrics.com.
[47] R. Péteri, M. Huskies, S. Fazekas, DynTex: A Comprehensive Database of Dynamic Textures, Online Dynamic Texture Database. http://www.cwi.nl/projects/dyntex.

Please cite this article in press as: J. Xu et al., Learning spatio-temporal dependency of local patches for complex motion segmentation, Comput. Vis. Image Understand. (2011), doi:10.1016/j.cviu.2010.11.010