Successive Convex Matching for Action Detection

Hao Jiang, Mark S. Drew and Ze-Nian Li
School of Computing Science, Simon Fraser University, Burnaby BC, Canada V5A 1S6
{hjiangb,mark,li}@cs.sfu.ca

Abstract

We propose human action detection based on a successive convex matching scheme. Human actions are represented as sequences of postures and specific actions are detected in video by matching the time-coupled posture sequences to video frames. The template sequence to video registration is formulated as an optimal matching problem. Instead of directly solving the highly non-convex problem, our method convexifies the matching problem into linear programs and refines the matching result by successively shrinking the trust region. The proposed scheme represents the target point space with small sets of basis points and therefore allows efficient searching. This matching scheme is applied to robustly matching a sequence of coupled binary templates simultaneously in a video sequence with cluttered backgrounds.

1. Introduction

Detecting gestures in controlled environment has been intensively studied and many realtime systems have been implemented [1][2][3]. Finding actions of people in a video recorded in an uncontrolled environment is still a largely unsolved problem, with many important potential applications such as surveillance and content based video retrieval. The main difficulty for action recognition in general video derives from the fact that there is no effective way to segment an object in such videos. Other factors such as the highly articulated character of the human body, large variability of clothing, and strong background clutter further increase the difficulty of action recognition.

In this paper, we study methods to detect a specific human action in such an uncontrolled setting. We represent an action as a sequence of body postures with specific temporal constraints. We can then search for a given action by matching a sequence of coupled body posture templates to the video sequence. We formulate the matching problem as an energy minimization problem. The objective function is minimized such that the matching cost is low and at the same time we try to smooth the intra-frame matching and inter-frame object center's relative position. A center continuity constraint is important to force the matching to stick to one object in cluttered video where multiple objects may appear. As shown in our experiments, a greedy scheme such as ICM [4] is not robust enough if there is strong clutter or large deformation. Robust matching methods such as Graph Cut [5], Belief Propagation (BP) [6] and most recently a Linear Programming (LP) relaxation scheme [7] have been studied for finding correspondence in single image pairs using pairwise constraints. These methods are not easily extended to include the center continuity constraint. In this paper, we consider a more straightforward approach — a successive convex matching scheme to register template image sequences to targets in video. We follow an LP relaxation scheme [8] that has been applied to motion estimation, reshaping the problem so that the inter-frame constraint can be introduced. Instead of directly solving the optimal matching problem, the proposed scheme converts the optimization problem into easier convex problems and linear programming is applied to solve the sub-problems. An iterative process updates the trust region and successively improves the approximation. This convex matching scheme has many useful features: it involves only a small set of basis target points, and it is a strong approximation scheme. It is also found to be robust against strong clutter and large deformations, necessary for success of an action recognition scheme. After template to video registration, we compare the similarity of the matching targets in video with the templates by matching cost and degree of deformation.

Finding people and recognizing human actions is a research area with a long history in vision. Searching for static postures [9][10][11] has been intensively studied. For action recognition, most previous work searches for motion patterns, since motion is resistant to clothing change. Motion based schemes usually need tracking or background motion compensation [12] if the camera is not fixed. One motion matching scheme without explicit motion estimation is also studied [13]. In recent years, appearance based schemes received a lot of interest. In such schemes, an appearance model is explicitly matched to a target video sequence in action detection. Appearance based approaches are more effective if camera motion is involved. One appearance approach is to recognize action by a body parts model [14][15][16]. Detecting the human body configuration based on smaller local features is another appearance matching scheme which has been applied to detecting action in a single image [9][11]. In this paper, we follow the appearance matching direction and study a convex method

0-7695-2597-0/06 $20.00 © 2006 IEEE

for video sequence matching. We explicitly match frames by using local features with intra-frame pairwise constraint and inter-frame position constraint over a longer time interval, thus enabling the scheme to detect complex actions.

2. Matching Templates to Video

We formulate the sequence matching problem as follows. We extract n templates from a video sequence, which represent key postures of an object in some specific action. Template i is represented as a set of feature points S_i and the set of neighboring pairs $\mathcal{N}_i$. $\mathcal{N}_i$ consists of all the pairs of feature points, connected by edges in the Delaunay graph of S_i. Fig. 1 illustrates intra-frame and inter-frame constrained deformable video matching. Matching a template sequence to a video can be formulated as an optimization problem. We search for matching function $\mathbf{f}$ to minimize the following objective function:

$$\min_{\mathbf{f}} \left\{ \sum_{i=1}^{n} \sum_{\mathbf{s}\in S_i} C^i(\mathbf{s}, \mathbf{f}^i_{\mathbf{s}}) + \lambda \sum_{i=1}^{n} \sum_{\{\mathbf{p},\mathbf{q}\}\in\mathcal{N}_i} d(\mathbf{f}^i_{\mathbf{p}} - \mathbf{p}, \right.$$
$$\left. \mathbf{f}^i_{\mathbf{q}} - \mathbf{q}) + \mu \sum_{i=1}^{n-1} d(\bar{\mathbf{s}}^{(i+1)} - \bar{\mathbf{s}}^i, \bar{\mathbf{f}}^{(i+1)} - \bar{\mathbf{f}}^i) \right\}$$

Here, $C^i(\mathbf{s}, \mathbf{f}^i_{\mathbf{s}})$ is the cost of matching feature point $\mathbf{s}$ in template i to point $\mathbf{f}^i_{\mathbf{s}}$ in a target frame; $\bar{\mathbf{f}}^i$ and $\bar{\mathbf{s}}^i$ are centers of the matching target points and template points respectively for the ith template; Distance $d(.,.)$ is a convex function. The first term in the objective function represents the cost of a specific matching configuration. The second and third terms are intra-frame and inter-frame regularity terms respectively. The coefficients λ and μ are used to control the weights of the smoothing terms. In this paper, we focus on problems in which $d(.,.)$ is defined using L_1 norm. As will be shown later, in this case, a linear programming relaxation of the problem can be constructed. To simplify the matching process, we enforce that target points for one template cannot be dispersed into several target frames. The matching frame for template i is specified as $i_0+\ \Delta_i$, in which i_0 is a start frame number and Δ_i is the temporal offset of template frame i.

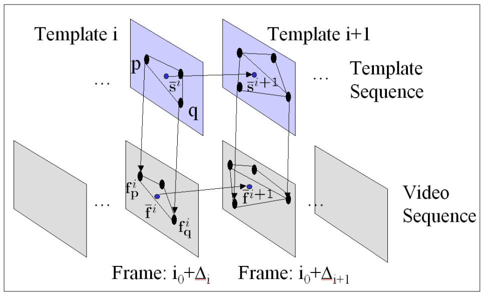


Figure 1. Deformable video matching.

The above optimization problem is non-linear and usually non-convex, because matching cost functions $C^i(\mathbf{s}, \mathbf{t})$ are usually highly non-convex with respect to $\mathbf{t}$ in real applications. In the following, we discuss feature selection and methods to cast the non-convex optimization problem into a sequential convex programming problem, so that a robust and efficient optimization solution can be obtained.

2.1. Features for Matching

To match people across clothing change, we need to choose features that are at the same time not sensitive to colors and robust to deformations. Even though different feature types can be used, here we use edge features to demonstrate the usefulness of the matching scheme. Edge maps have been found to be very useful for object class detection, especially matching human objects. To increase matching robustness, instead of directly matching edge maps, we match a transformed edge map. A distance transform is applied to turn edge maps into a grayscale representation in which values are proportional to the distances to the nearest edge pixels. Small image patches on these distance transform images are found to provide good features in matching. To make the local features incorporate more context, we calculate the log-polar transform of the distance transform image centered on the selected feature points in the target and template images. The log-polar transform simulates the human visual system's foveate property and puts more focus in the center view than the periphery views. This feature is similar to the blurred edge features in [7] for object class detection. The log-polar distance transform feature increases robustness in matching. Nevertheless, without a robust matching scheme the matching is still very likely to fail.

2.2. Linear Programming Relaxation and Simplex Method

To linearize the matching cost term in the non-linear objective function, we select a set of *basis target points* for each feature point in a template. Then, a target point can be represented as a linear combination of these basis points, e.g, $\mathbf{f}^i_{\mathbf{s}} = \sum_{\mathbf{t}\in B^i_{\mathbf{s}}} w^i_{\mathbf{s},\mathbf{t}} \cdot \mathbf{t}$, where $\mathbf{s}$ is a feature point in template i, and $B^i_{\mathbf{s}}$ is the basis target point set for $\mathbf{s}$. We will show that the "cheapest" basis set for a feature point consists of the target points corresponding to the matching cost surface's *lower convex hull* vertices. Therefore $B^i_{\mathbf{s}}$ is usually much smaller than the whole target point set for feature point $\mathbf{s}$. This is a key step to speed up the algorithm. We can now represent the cost term as a linear combination of the costs of basis target points. For template i, the matching cost term can thus be represented as $\sum_{\mathbf{s}\in S_i} \sum_{\mathbf{t}\in B^i_{\mathbf{s}}} w^i_{\mathbf{s},\mathbf{t}} C^i(\mathbf{s}, \mathbf{t})$. A standard linear programming trick of using auxiliary variables can be used to turn L_1 terms in the objective function into linear functions [17]: we represent each term in $|\cdot|$ as the difference of two nonnegative auxiliary variables. Substituting this into the constraint, we replace the term in the objective function with the summation of two auxiliary variables. In our formulation, the summation equals the absolute value of the original term when the linear program is indeed optimized.

The complete linear program is written as:

$$\min \left\{ \sum_{i=1}^{n} \sum_{\mathbf{s} \in S_i} \sum_{\mathbf{t} \in B_{\mathbf{s}}^i} w_{\mathbf{s},\mathbf{t}}^i C^i(\mathbf{s},\mathbf{t}) + \right.$$

$$\lambda \sum_{i=1}^{n} \sum_{\{\mathbf{p},\mathbf{q}\} \in \mathcal{N}_i} (x_{\mathbf{p},\mathbf{q}}^{i+} + x_{\mathbf{p},\mathbf{q}}^{i-} + y_{\mathbf{p},\mathbf{q}}^{i+} + y_{\mathbf{p},\mathbf{q}}^{i-}) +$$

$$\left. \mu \sum_{i=1}^{n-1} (u^{i+} + u^{i-} + v^{i+} + v^{i-}) \right\}$$

$$\begin{aligned}
s.t. \sum_{\mathbf{t} \in B_{\mathbf{s}}^i} w_{\mathbf{s},\mathbf{t}}^i &= 1, \forall \mathbf{s} \in S_i, i = 1..n \\
x_{\mathbf{s}}^i &= \sum_{\mathbf{t} \in B_{\mathbf{s}}^i} w_{\mathbf{s},\mathbf{t}}^i \cdot x(\mathbf{t}),\ y_{\mathbf{s}}^i = \sum_{\mathbf{t} \in B_{\mathbf{s}}^i} w_{\mathbf{s},\mathbf{t}}^i \cdot y(\mathbf{t}) \\
x_{\mathbf{p},\mathbf{q}}^{i+} - x_{\mathbf{p},\mathbf{q}}^{i-} &= x_{\mathbf{p}}^i - x(\mathbf{p}) - x_{\mathbf{q}}^i + x(\mathbf{q}), \\
y_{\mathbf{p},\mathbf{q}}^{i+} - y_{\mathbf{p},\mathbf{q}}^{i-} &= y_{\mathbf{p}}^i - y(\mathbf{p}) - y_{\mathbf{q}}^i + y(\mathbf{q})\ , \\
\forall \{\mathbf{p},\mathbf{q}\} &\in \mathcal{N}_i, i = 1..n \\
u^{i+} - u^{i-} &= \frac{1}{|S_i|} \sum_{\mathbf{s} \in S_i} [x_{\mathbf{s}}^i - x(\mathbf{s})] \\
&\quad - \frac{1}{|S_{i+1}|} \sum_{\mathbf{s} \in S_{i+1}} [x_{\mathbf{s}}^{i+1} - x(\mathbf{s})], \\
v^{i+} - v^{i-} &= \frac{1}{|S_i|} \sum_{\mathbf{s} \in S_i} [y_{\mathbf{s}}^i - y(\mathbf{s})] \\
&\quad - \frac{1}{|S_{i+1}|} \sum_{\mathbf{s} \in S_{i+1}} [y_{\mathbf{s}}^{i+1} - y(\mathbf{s})], \\
&\quad i = 1..n-1 \\
\text{All variables} &\geq 0
\end{aligned}$$

Here we define functions $x(\mathbf{s})$ and $y(\mathbf{s})$ as extracting the x and y components of point $\mathbf{s}$. The matching result $\mathbf{f}_{\mathbf{s}}^i = (x_{\mathbf{s}}^i, y_{\mathbf{s}}^i)$. It is not difficult to verify that either $x_{\mathbf{p},\mathbf{q}}^{i+}$ or $x_{\mathbf{p},\mathbf{q}}^{i-}$ (similarly $y_{\mathbf{p},\mathbf{q}}^{i+}$ or $y_{\mathbf{p},\mathbf{q}}^{i-}$, u^{i+} or u^{i-} and v^{i+} or v^{i-}) will become zero when the linear programming achieves its minimum; therefore we have $x_{\mathbf{p},\mathbf{q}}^{i+} + x_{\mathbf{p},\mathbf{q}}^{i-} = |x_{\mathbf{p}}^i - x(\mathbf{p}) - x_{\mathbf{q}}^i + x(\mathbf{q})|$, $y_{\mathbf{p},\mathbf{q}}^{i+} + y_{\mathbf{p},\mathbf{q}}^{i-} = |y_{\mathbf{p}}^i - y(\mathbf{p}) - y_{\mathbf{q}}^i + y(\mathbf{q})|$, and so on. The second and third regularity terms in the linear program objective function equal the corresponding terms in the original non-linear problem. In fact, if $B_{\mathbf{s}}^i$ contain all the target points and weights $w_{\mathbf{s},\mathbf{t}}^i$ are binary variables (0 or 1), the LP becomes an integer programming problem which exactly equals the original non-convex problem. But, integer programming is as hard as the original non-linear problem, and therefore we are most interested in the relaxed linear programming problem. The linear program has close relation with the continuous extension of the non-linear matching problem: the continuous extension of the non-linear problem is defined by first interpolating the matching cost surfaces $C^i(\mathbf{s},\mathbf{t})$ piecewise-linearly with respect to $\mathbf{t}$ and then relaxing feasible matching points into a continuous region (the convex hull of the basis target points $B_{\mathbf{s}}^i$). In the following, we also use $C^i(\mathbf{s},\mathbf{t})$ to represent the continuous extension cost surfaces.

Property 1: *If $B_{\mathbf{s}}^i = L_{\mathbf{s}}^i$,* where $L_{\mathbf{s}}^i$ is the entire target point set of $\mathbf{s}$ for template i, and *the continuous extension cost function $C^i(\mathbf{s},\mathbf{t})$ is convex with respect to $\mathbf{t}$, $\forall \mathbf{s} \in S_i,\ i = 1..n$, LP exactly solves the continuous extension of the discrete matching problem.*

In practice, the cost function $C^i(\mathbf{s},\mathbf{t})$ is usually highly non-convex with respect to $\mathbf{t}$ for each site $\mathbf{s}$. In this case:

Property 2: *The linear programming formulation solves the continuous extension of the reformulated discrete matching problem, with $C^i(\mathbf{s},\mathbf{t})$ replaced by its lower convex hull for each site* $\mathbf{s}$.

For matching applications, the surface is the matching cost surface. Note that in general the surface may have holes, or consist only of irregular discrete 3D points in the target point vs. cost space, e.g. if we only select edge points in the target images for matching.

Property 3: *We need only consider the basis set $B_{\mathbf{s}}^i$ comprised of the vertex coordinates of the lower convex hull of $C^i(\mathbf{s},\mathbf{t})$, $\forall \mathbf{s} \in S$.*

Thus, we can use only the smallest basis set — there is no need to include all the candidate matching costs in the optimization. This is one of the key steps to speed up the algorithm.

The proposed solution of the relaxation scheme also has the following structure property.

Property 4: *Using the simplex method, there will be at most 3 nonzero-weight basis target points for each site.*

Proof: This property is due to the basic linear programming property: if the optimum of an LP exists, the optimum must be located at one of the "extreme" points of the feasible region. The extreme points of linear programming correspond to the basic feasible solutions of LP. We denote the constraints of our linear program by $A\mathbf{x} = b$, $\mathbf{x} \geq \mathbf{0}$. Each basic feasible solution of LP has the format $[K^{-1}b, 0]^T$ where K is an invertible matrix composed of the columns of matrix A corresponding to the basic variables. For site $\mathbf{s}$, variable $w_{\mathbf{s},\mathbf{t}}^i$ introduces a column $[..., 0, 1, x(\mathbf{t}), y(\mathbf{t}), 0, ...]^T$ in A. It is not difficult to show that the sub-matrix generated by these columns for a single site has a rank at most 3. Therefore, we can have at most three w for each site in the basic variable set. This implies that the optimum solution has at most three nonzero w for each site.

The initial basic variables can be selected in the following way:

- Only one $w_{\mathbf{s},\mathbf{t}}^i$ is selected as basic LP variable for each site $\mathbf{s}$ in template i.
- $x_{\mathbf{s}}^i$, $y_{\mathbf{s}}^i$ are basic LP variables.
- Whether $x_{\mathbf{p},\mathbf{q}}^{i+}$ or $x_{\mathbf{p},\mathbf{q}}^{i-}$, $y_{\mathbf{p},\mathbf{q}}^{i+}$ or $y_{\mathbf{p},\mathbf{q}}^{i-}$, u^{i+} or u^{i-} and v^{i+} or v^{i-} are basic variables depends on the right

hand side of the constraint; if the right hand side of a constraint is greater than 0, the plus term is a basic variable, otherwise the minus term becomes the basic variable.

Importantly, Property 4 implies that for each site the proposed LP relaxation **searches only the triangles of the lower convex hull vertices**, in an efficient energy descent manner. (And note that the triangles may be degenerate.) Fig. 2 illustrates the solution procedure of the simplex method for an example two-frame video matching problem. In this simple example, three features points are selected on the objects in Figs. 2 (a, b) respectively and form triangular graph templates, shown in Figs. 2 (e, f). Figs. 2 (c, d) show the target objects in clutter. Figs. 2 (g, h) show the matching result. Figs. 2 (i, j, k, l, m, n) show the matching cost surfaces for each of the six points on the template. Figs. 2 (o, p, q, r, s, t) are the lower convex hull surfaces for the respective cost surfaces. The searching process (selected from 32 stages) for each site is illustrated in this example. The blue dots indicate the target points located at the coordinates of the lower convex hull vertices. The target points corresponding to the basic variables are connected by lines. The small rectangle is the weighted linear combination of the target points corresponding to the basic variables at each stage. As expected, the proposed LP only checks triangles (filled-blue) or their degenerates (lines or points) formed by basis target points. When the search terminates, the patch generated by the basic variables for each site must correspond to vertices, edges or facets of the lower convex hull for each site. As shown in this example, a single LP relaxation usually has a matching result near the target but not very accurate. We will discuss how to refine the result by successively "convexifying" the matching cost surfaces.

2.3. Successive Relaxation

As discussed above, a single LP relaxation approximates the original problem's matching cost functions by their lower convex hulls. In real applications, several target points may have equal matching cost and, even worse, some incorrect matches may have lower cost. In this case, because of the convexification process, many local structures are removed which on the one hand facilitates the search process by removing many false local minimums and on the other hand makes the solution not exactly locate on the true global minimum. A successive relaxation method, successive convexification linear programming (SC-LP), can be used to solve the problem. Instead of one step LP relaxation, we can construct linear programs recursively based on the previous searching result and gradually shrink the matching trust region for each site. A trust region for one site is a rectangle area in the target image. Such a scheme can effectively solve the coarse approximation problem in single step LP.

In trust region shrinking, we use control points to anchor trust regions for the next iteration. We keep the control point in the new trust region for each site and we can shrink the boundary inwards. If the control point is on the boundary of the previous trust region, other boundaries are moved inwards. The trust region is the whole target image for the first LP relaxation. Then we can refine the regions based on previous LP's solution. After we shrink the trust region, the lower convex hull may change for each site. Therefore, we have to find the new target basis and solve a new LP. This process is illustrated in Fig. 3.

We select control points using a consistent rounding process. In consistent rounding, we choose a site randomly and check all the possible discrete target points and select the one that minimizes the nonlinear objective function, by fixing other sites' targets as the current stage LP solution. This step is similar to a single iteration of an ICM algorithm by using LP solution as initial value. We also require that new control points have energy not greater than the previous estimation.

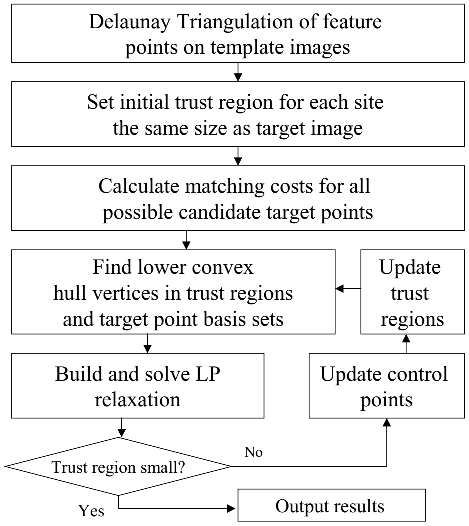
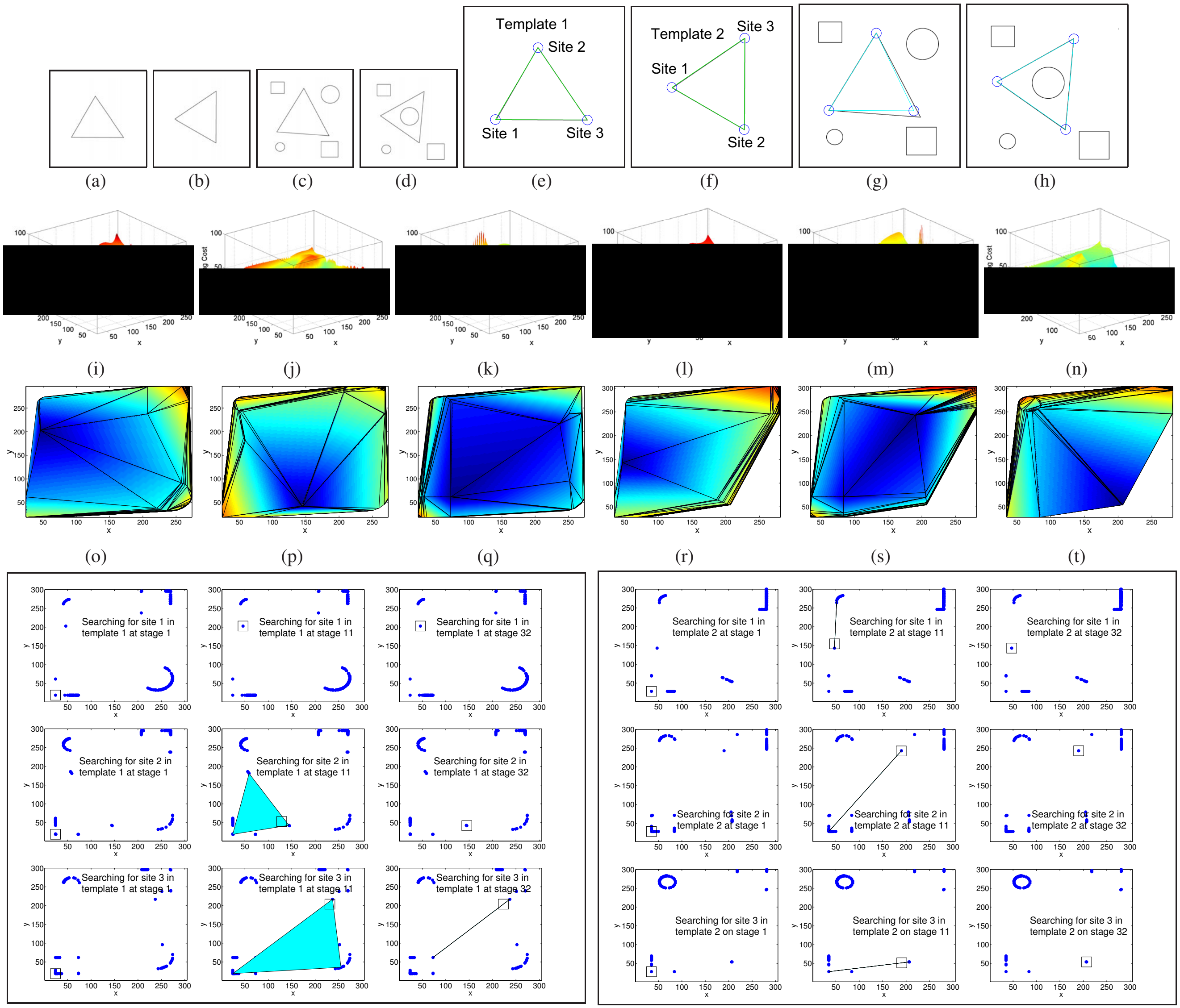

Delaunay Triangulation of feature points on template images
Set initial trust region for each site the same size as target image
Calculate matching costs for all possible candidate target points
Find lower convex hull vertices in trust regions and target point basis sets
Update trust regions
Build and solve LP relaxation
Update control points
Trust region small?
No
Yes
Output results

Figure 3. Successive convex matching.

2.4. Action Detection

After the template to video registration, we can compare similarity of the matching targets with templates to decide how similar these two constellations of matched points are and whether the matching result corresponds to the same action as in the exemplar. We use the following quantities to measure the difference between the template and the matching object. The first measure is D, defined as the average of pairwise length changes from the template to the target. To compensate for the global deformation, a global affine transform $\mathcal{A}$ is first estimated based on the matching and then applied to the template points before calculating D. D is further normalized with respect to the average edge length of the template. The second measure is the average template matching cost M using the log-polar transform feature. The total matching cost is simply defined as $M + \alpha D$, where α has a typical value of 10 if image pixels are in range of 0-255. Experiments show that only about 100 randomly selected feature points are needed in calculating D and M.



(a) (b) (c) (d) (e) (f) (g) (h)

(i) (j) (k) (l) (m) (n)

(o) (p) (q) (r) (s) (t)

(u) Searching for Template 1

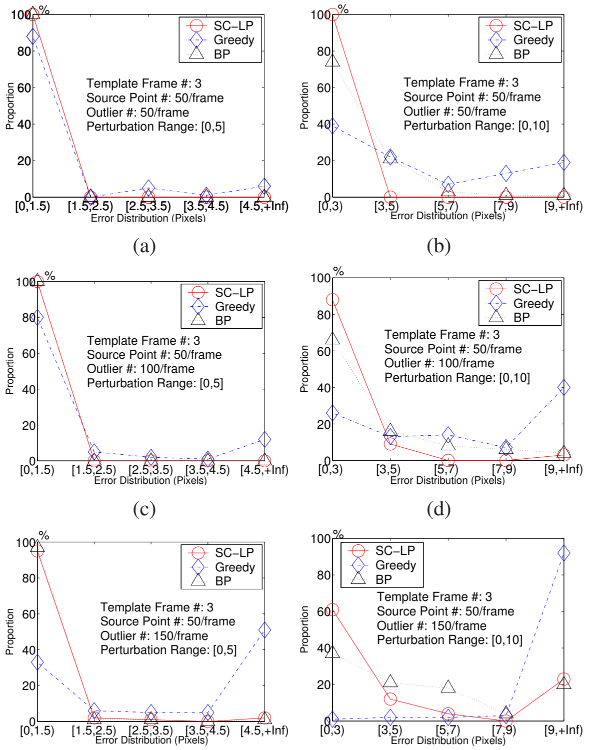
(v) Searching for Template 2

Figure 2. Searching process of the linear program. (a, b) Template images; (c, d) Target images; (e, f) Feature points and template graph; (g, h) Matching result; (i, j, k, l, m, n) Matching cost surfaces for each site on the template; (o, p, q, r, s, t) Convexified matching cost surfaces: lower values on a surface show cooler colors; (u, v) illustrate the searching process of the linear program.

3. Experiment Results

3.1. Matching Random Sequences

In the first experiment, we test the proposed scheme with synthetic images. In each experiment, three coupled random template images are generated. Each template image contains 50 random points in a 128×128 image. The target images contain a randomly shifted and perturbed version of the data points in 256×256 images. The perturbation is uniformly disturbed in two settings: 0-5 and 0-10. The center of the two templates are also randomly perturbed in the range 0-5. We use the log-polar transform feature of the distance transform image in all our experiments. We compare our result with a greedy searching scheme. Other standard schemes, such as BP, cannot be easily extended to solve the minimization problem in this paper. Instead we use BP to match each image pair separately as a benchmark in comparison. The Graph Cut, designed mainly for stereo matching and motion estimation, is not included in the comparison. Each experiment is repeated in a deformation and clutter setting over 100 trials. Fig. 4 shows the average matching error distribution in different assumed-error regions. When both the noise level and distortion level are low, the greedy scheme has comparable performance. Since there is one single target in each image, BP has similar performance as the proposed scheme for low deformation setting experiments. Greedy scheme's performance degrades rapidly when the levels of noise and distortion increase. In



(a) (b)

(c) (d)

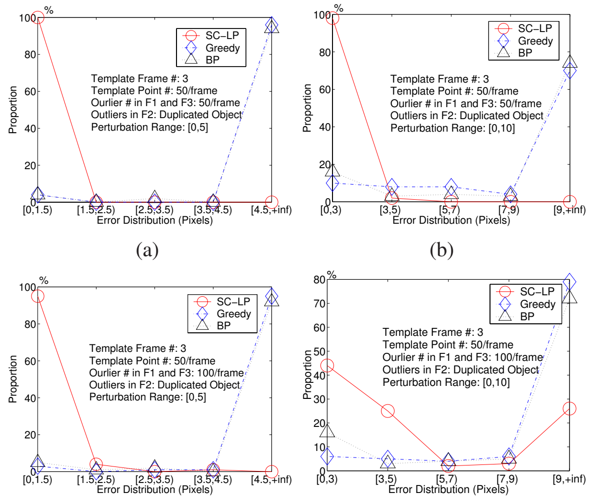
(e) (f)

Figure 4. Random sequence matching.

high noise density and large deformation cases, the proposed scheme greatly outperforms the greedy scheme. It is also better than a baseline BP scheme for large distortion cases. One iteration of linear programming in this experiment takes about 0.3 seconds in a 2.6GHz PC. The typical number of iterations is 3. Fig. 5 shows comparison results of matching random sequences in a different outlier pattern setting which introduces an extra duplicated and perturbed object into the second target frame. For BP and greedy scheme, matching error for the second template frame is the smaller one of matching either of the two objects in the target frame. In this case, the proposed sequence matching scheme yields much better results.

3.2. Matching Human Activities in Video

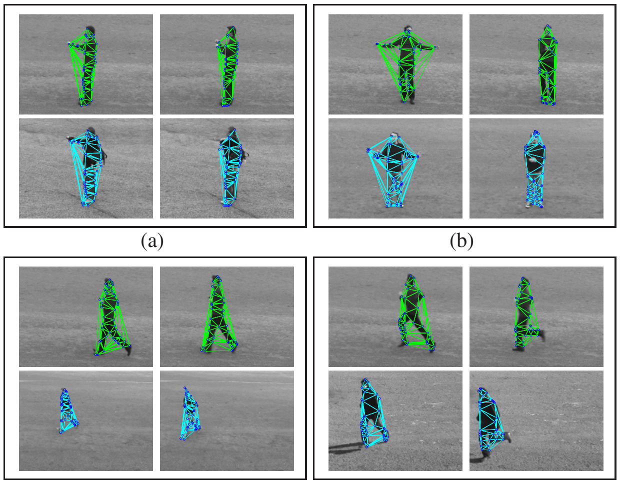
We test the proposed matching scheme with dataset [18] which includes six gestures. We select templates with two key postures from the first video clip in each category and then randomly select 15 other video clips in each class for testing (The video clips with mirror action are not included). Fig. 6 shows examples of the two-key-frame templates in the first row of each sub-figure. We select regions in the two-frame templates for each of the six actions. Graph templates are automatically generated using randomly selected edge points in the region of interest. The templates are then used to compare with each testing video clip at each time instant using the proposed matching scheme and the minimal matching cost is used as the score for a clip. Three time scales 0.75, 1 and 1.25 are used in searching. Spatial



(a) (b)

(c) (d)

Figure 5. Matching sequences with multiple objects.



(a) (b)

(c) (d)

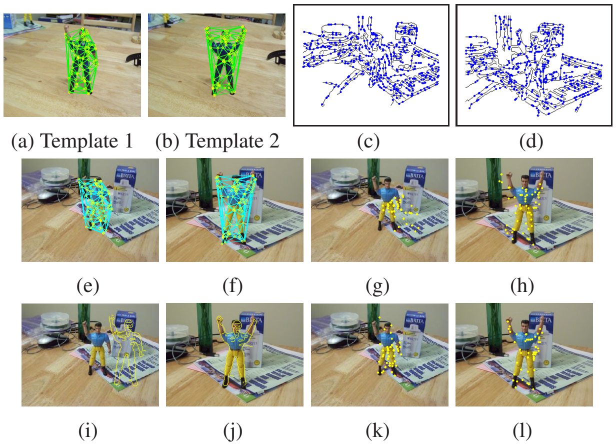
Figure 6. Matching examples. In (a, b, c, d) top rows are templates and bottom rows are matching results.

scale is fixed in the experiment. A video clip is classified as an action if the corresponding key-posture templates generate the lowest match score among six actions. Fig. 6 shows some matching examples and Table 1 shows the detection confusion matrix: the method performs very well.

	box	clap	jog	walk	run	wave
box	14	1	0	0	0	0
clap	2	13	0	0	0	0
jog	0	0	14	0	1	0
walk	2	0	0	12	1	0
run	3	1	0	0	11	0
wave	2	1	0	0	0	12

Table 1. Confusion matrix for 15 randomly selected video clips in each action class. Each row shows classification result for prelabeled testing data in an action category.

Fig. 7 shows an example of matching for cluttered images with the proposed sequence matching scheme, the

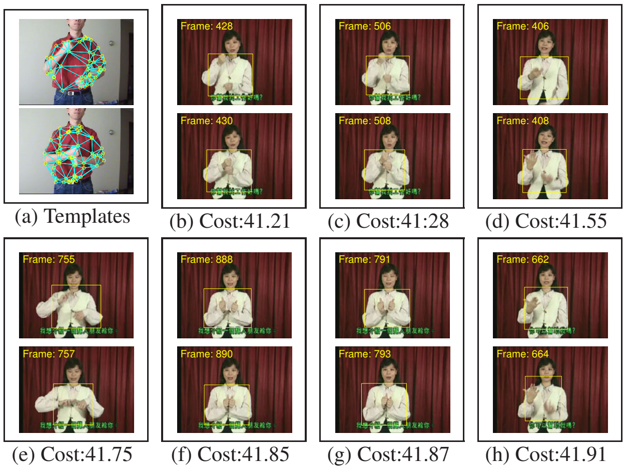

(a) Template 1 (b) Template 2 (c) (d)

(e) (f) (g) (h)

(i) (j) (k) (l)

Figure 7. Matching flexible objects. (a, b) Templates; (c, d) Target image edge maps and feature points; (e, f) Matching with the proposed scheme; (g, h) Matching with greedy scheme; (i, j) Chamfer matching for each image pair; (k, l) Matching with BP for each image pair.

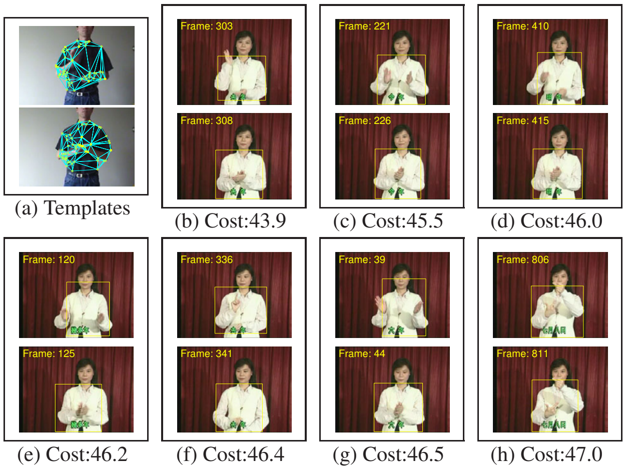
greedy scheme, Chamfer matching and the BP matching. Chamfer matching and BP match each single image pair separately. The proposed scheme still works well in cluttered images, while the greedy scheme, Chamfer matching and BP fail to locate the target. BP is also about 100 times slower. We further conducted experiments to search for a specific gesture in video. In these test videos, a specific action only appears a few times. Target objects also have large deformation with respect to the templates. The templates we use have roughly the same scale as the testing sequence. The template sequence is swept along the time axis with a step of one frame, and for each instant we match video frames with the templates. We first applied the matching scheme to detect specific sign language gestures. Sign language is challenging because of the very subtle differences. Fig. 8 shows a searching result for the gesture "work" in a 1000-frame video. The template sequence is generated from a different subject. The two gestures in the video are successfully located in the top two rank positions of the shortlist. Fig. 9 shows a searching result for the gesture "year" in a 1000-frame video. The starting and ending frames of actions in video are ranked based on their matching score. Five appearances of the gesture are located in top 6 of the shortlist. One false detection is inserted at rank 5. Fig. 10 and Fig. 11 show experiments to locate two actions, kneeling and hand-waving, in indoor video sequences of 800 and 500 frames respectively. The two-frame templates are from videos of another subject in different environments. The videos are taken indoors and contain many bar structures which are very similar to human limbs. The proposed scheme finds all the 2 kneeling actions in the test video in the top two of the shortlist; and all the 11 waving hand actions in the top 13 ranks. Fig. 12 shows the result of search for a "throwing" action in a 2500-frame baseball sequence. The object occupies very small part of the video. There is large deformation and strong background clutter. Closely interlaced matching results are merged and our method finds all the 4 appearances of the action at the top of the shortlist. We found that false detection in our experiments is mainly due to similar structures in the background near the subject. Prefiltering or segmentation operations to partially remove the background clutter can further increase the robustness of detection.



(a) Templates (b) Cost:41.21 (c) Cost:41:28 (d) Cost:41.55

(e) Cost:41.75 (f) Cost:41.85 (g) Cost:41.87 (h) Cost:41.91

Figure 8. Searching gesture "work" in a 1000-frame sign language sequence. (a) Templates; (b..h) Top 7 matches of the shortlist.



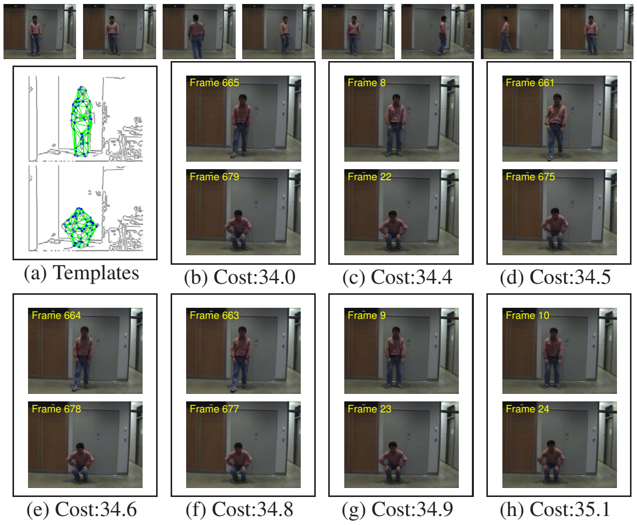
(a) Templates (b) Cost:43.9 (c) Cost:45.5 (d) Cost:46.0

(e) Cost:46.2 (f) Cost:46.4 (g) Cost:46.5 (h) Cost:47.0

Figure 9. Searching gesture "year" in a 1000-frame sign language sequence. (a) Templates; (b..h) Top 7 matches of the shortlist.

4. Conclusion

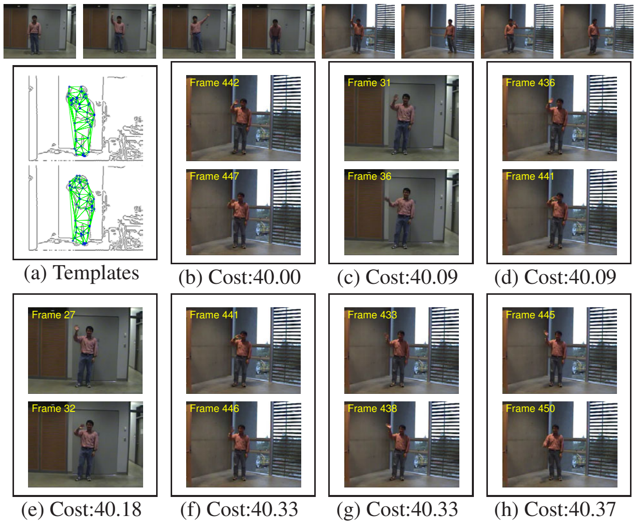
In this paper, we present a successive convex programming scheme to match video sequences using intra-frame and inter-frame constrained local features. By convexifying the optimization problem sequentially with an efficient linear programming scheme which can be globally optimized in each step, and gradually shrinking the trust region, the proposed method is more robust than previous matching schemes. The matching scheme has unique features in searching: it involves a very small number of basis points and thus can be applied to problems that involve large number of target points. The proposed scheme has been successfully applied to locating specific actions in video sequences. Because the template deforms, this scheme can deal with



(a) Templates (b) Cost:34.0 (c) Cost:34.4 (d) Cost:34.5

(e) Cost:34.6 (f) Cost:34.8 (g) Cost:34.9 (h) Cost:35.1

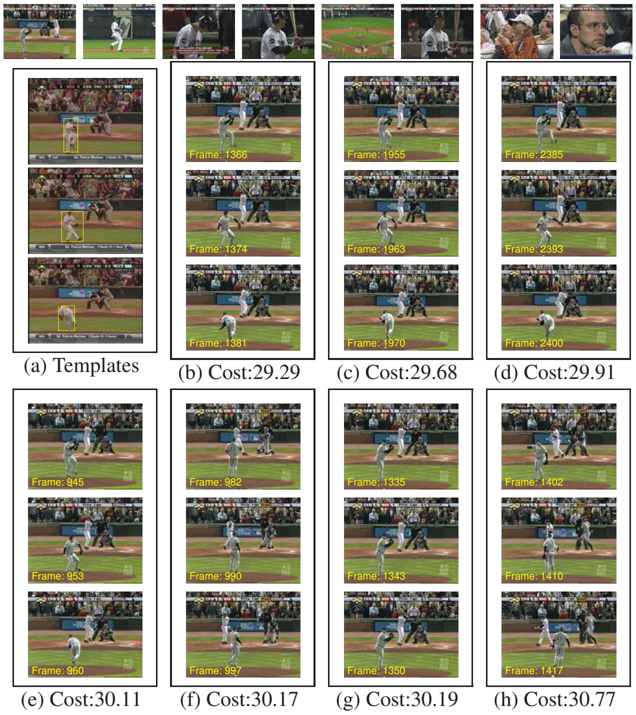
Figure 10. Searching “kneeling” in a 800-frame indoor sequence. (a) Templates; (b..h) Top 7 matches of the shortlist.



(a) Templates (b) Cost:40.00 (c) Cost:40.09 (d) Cost:40.09

(e) Cost:40.18 (f) Cost:40.33 (g) Cost:40.33 (h) Cost:40.37

Figure 11. Searching “right hand waving” in a 500-frame indoor sequence. (a) Templates; (b..h) Top 7 matches of the shortlist.



(a) Templates (b) Cost:29.29 (c) Cost:29.68 (d) Cost:29.91

(e) Cost:30.11 (f) Cost:30.17 (g) Cost:30.19 (h) Cost:30.77

Figure 12. Searching “throwing ball” in a 2500-frame baseball sequence. (a) Templates; (b..h) Top 7 matches of the shortlist.

large distortions between the template and the target object.

References

[1] Kidsroom – An Interactive Narrative Playspace. http://vismod.media.mit.edu/vismod/demos/kidsroom/kidsroom.html 1

[2] K.M.G. Cheung, S. Baker, T. Kanade, “Shape-from-silhouette of articulated objects and its use for human body kinematics estimation and motion capture”, CVPR, pp. I:77-84 vol.1, 2003. 1

[3] L. Emering and B. Herbelin, “Body gesture recognition and action response”, Handbook of Virtual Humans, Wiley 2004, pp.287-302. 1

[4] J. Besag, “On the statistical analysis of dirty pictures”, J. R. Statis. Soc. Lond. B, 1986, Vol.48, pp. 259-302. 1

[5] Y. Boykov, O. Veksler, and R. Zabih, “Fast approximate energy minimization via graph cuts”, PAMI, Vol.23, pp. 1222-1239, 2001. 1

[6] Y. Weiss and W.T. Freeman. “On the optimality of solutions of the max-product belief propagation algorithm in arbitrary graphs”, IEEE Trans. on Information Theory, 47(2):723-735, 2001. 1

[7] A.C. Berg, T.L. Berg, J. Malik, “Shape matching and object recognition using low distortion correspondence”, CVPR 2005. 1, 2

[8] H. Jiang, Z.N. Li, M.S. Drew, “Optimizing motion estimation with linear programming and detail preserving PDE”, CVPR 2004. 1

[9] S. Carlsson and J. Sullivan, “Action recognition by shape matching to key frames”, IEEE Computer Society Workshop on Models versus Exemplars in Computer Vision, 2001. 1

[10] G. Mori, X. Ren, A. Efros, and J. Malik, “Recovering human body configurations: combining segmentation and recognition”, CVPR, pp.II:326-333, 2004. 1

[11] G. Mori and J. Malik, “Estimating human body configurations using shape context matching”, ECCV, LNCS 2352, pp. 666–680, 2002. 1

[12] A. A. Efros, A. C. Berg, G. Mori, and J. Malik, “Recognizing action at a distance”, ICCV 2003. 1

[13] E. Shechtman and M. Irani, “Space-time behavior based correlation”, CVPR 2005. 1

[14] P.F. Felzenszwalb, D.P. Huttenlocher, “Efficient matching of pictorial structures”, CVPR, pp. II:66-73 vol.2, 2000. 1

[15] R. Ronfard, C. Schmid, and B. Triggs, “Learning to parse pictures of people”, ECCV 2002, pp. 700–714, 2002. 1

[16] D. Ramanan, D. A. Forsyth, and A. Zisserman. “Strike a pose: tracking people by finding stylized poses”, CVPR 2005. 1

[17] V. Chvátal, Linear Programming, W.H. Freeman and Co. New York 1983. 2

[18] C. Schuldt, I. Laptev, and B. Caputo, “Recognizing human actions: A local SVM approach”, ICPR 2004. 6