Linear Programming for Matching in Human Body Gesture Recognition

Hao Jiang, Ze-Nian Li, and Mark S. Drew

School of Computing Science, Simon Fraser University,
Burnaby, BC, Canada V5A 1S6
{hjiangb, li, mark}@cs.sfu.ca

Abstract. We present a novel human body gesture recognition method using a linear programming based matching scheme. Instead of attempting to segment an object from the background, we develop a novel successive convexification linear programming method to locate the target by searching for the best matching region based on a graph template. The linear programming based matching scheme generates relatively dense matching patterns and thus presents a key feature for robust object matching and human body gesture recognition. By matching distance transformations of edge maps, the proposed scheme is able to match figures with large appearance changes. We further present gesture recognition methods based on the similarity of the exemplar with the matching target. Experiments show promising results for recognizing human body gestures in cluttered environments.

1 Introduction

Human body gesture recognition has attracted a lot of interest in recent years because of its potential important applications in surveillance, human-computer interaction and computer animation. Recognizing body gestures is also a challenging problem because of articulated motion of human limbs and bodies and large appearance variations such as the changes of clothing.

In this paper, we study problems where only a single camera is available. We present a gesture recognition method based on a novel linear programming (LP) matching scheme. The proposed LP scheme can be used to solve large scale L_1 metric labeling problems. Target matching in gesture recognition can be formulated as this subclass of labeling problems. Different from standard matching schemes such as the graph cut and belief propagation, the proposed LP relaxation method represents a label space with a much smaller set of basis labels, and is thus more suited for very large label set matching problems. A successive convexification scheme is proposed to solve the labeling problem. Iteratively, the trust region shrinks based on previous relaxation solution and the approximation becomes more accurate when the trust region becomes small. A new aspect of the algorithm is that the cost function is replaced by the lower convex hull at each stage — we re-convexify the cost, while focusing increasingly closely on the global solution. This is novel. The proposed multi-stage relaxation method is found to be more efficient than schemes such as the graph cut or belief propagation for the object matching problem where a large searching range is involved. It can also solve problems

W. Zhao, S. Gong, and X. Tang (Eds.): AMFG 2005, LNCS 3723, pp. 392–406, 2005.

for which traditional schemes fail. Based on the matching scheme, we propose a gesture recognition method which has the following properties: (1) The method works for cases when reliable background subtraction is unavailable, e.g., for still images; (2) It is quite insensitive to the clothing of the figures in the image. In this paper, local features are used because they have less variation than human parts and are therefore more reliable in matching. Unlike global shape features such as shape context [7], local features also enable the proposed scheme to be applicable to matching problems in cluttered environments. To suppress the influence of appearance changes for humans, we propose to match the distance transformations of the edge maps of the template and target images. This representation makes matching figures in different clothing possible. We further present a method to quantify the similarity of the template and the target object and form a reliable gesture recognition system.

Different schemes have been studied for recognizing human body gestures. Background subtraction has been used in gesture recognition. The difficulty with this scheme is that background subtraction is not robust and not always available, and the method cannot distinguish gestures when body parts are covered by silhouettes. One method to solve the problem is by extracting range data for the character in the scene using multiple cameras [1]. But such an approach is more expensive to deploy than monocular systems. A body-part based matching model [2] is presented for human body gesture recognition. As an extension, an SVM body-part matching method [3] is further presented. Mori [4] presents a segmentation based approach for part-based human body gesture recognition. Another method is to match the target as a whole, e.g. the Chamfer matching based method [5] in which tree structured binary templates are used to detect pedestrians. One shortcoming of this approach is that it usually needs many more templates than part-based schemes. Shape matching methods have also been applied for recognition of human actions [6][7]. Shape matching based methods usually need many fewer templates than the Chamfer matching scheme because the template deforms. These schemes work best in relatively clean background settings.

Object matching can be represented as a consistent labeling problem, and is essential for gesture recognition. Consistent labeling is NP-hard in general. Apart from a few cases in which polynomial algorithms are available, approximation algorithms are preferred for image matching. Much effort has been made to study efficient algorithms for these problems. Relaxation labeling (RL) [14] uses local search, and therefore relies on a good initialization process. ICM – Iterative Conditional Modes [9], another widely applied method for solving labeling problem, is greedy and is found to be easily trapped in a local minimum. In recent years, graph cut (GC) [11] and belief propagation (BP) [10] have become popular methods for solving consistent labeling problems. GC and BP are more robust than traditional labeling schemes and are also found to be faster than the traditional stochastic annealing methods. But GC and BP are still very complex for large scale problems that involve a large number of labels. Spectral graph theory based methods [15] have also been studied for matching. The work most related to our proposed scheme are the mathematical programming matching schemes. The early RL methods belong to this class. One of the big challenges in designing mathematical programming based labeling algorithms is to overcome local minima in the optimization process. Different schemes have been proposed. Deterministic annealing schemes [12]

have been successfully applied to matching point sets. Convex programming is another scheme for labeling problems. Up to now, methods such as quadratic programming and semidefinite programming can only be applied to small scale problems. Because of its efficiency, linear programming has been successfully applied in vision problems, e.g. estimating motion of rigid scenes [17]. A linear programming formulation [16] is presented for uniform labeling problems and approximating general problems by tree metrics. Another general LP scheme, studied in [13], is similar to the linear relaxation labeling formulation [14]. This LP formulation is found to be only applicable to small problems because of the large number of constraints and variables involved.

2 Gesture Estimation with Matching

In this section, we present a scheme for estimating human body gestures based on visual pattern matching using linear programming. First, we present our novel linear programming matching method, which forms the key component for gesture recognition. Then, we study gesture recognition based on similarity measures.

2.1 Matching by Linear Programming

In L_1 metric space, matching can be stated in general as the following consistent labeling problem:

$$\min_{\mathbf{f}} \varepsilon : \sum_{\mathbf{s} \in S} c(\mathbf{s}, \mathbf{f_s}) + \sum_{\{\mathbf{p},\mathbf{q}\} \in \mathcal{N}} \lambda_{\mathbf{p},\mathbf{q}} ||\mathbf{f_p} - \mathbf{f_q}||$$

in which $c(\mathbf{s}, \mathbf{f}_s)$ is the cost of assigning label $\mathbf{f_s}$ to site $\mathbf{s}$; $||.||$ is the L_1 norm and $\mathbf{f}$ are labels defined in L_1 metric space; S is a finite set of sites; $\mathcal{N}$ is the set of non-ordered neighbor site pairs; $\lambda_{\mathbf{p},\mathbf{q}}$ are smoothing coefficients. In the following discussion, we assume that both $\mathcal{S}$ and label sets $\mathcal{L}_\mathbf{s}$ are discrete and $\mathbf{f}$ are 2D vectors. The proposed method can be easily extended to cases where labels have higher dimensionality. We can always convert a discrete labeling problem into a continuous one using the following procedure. First, we interpolate the costs $c(\mathbf{s}, \mathbf{f_s})$ for each site piecewise-linearly such that $c(\mathbf{s}, \mathbf{f_s})$ become surfaces; then we extend the feasible region for $\mathbf{f}$ to the convex hull supported by the discrete labels. The new problem is defined as *continuous extension* of the original discrete problem. To simplify notation, we also use $c(\mathbf{s}, \mathbf{f}_s)$ to represent the continuous extension cost function.

2.2 Approximation by Linear Programing

The above energy optimization problem is nonlinear and usually non-convex, which makes it difficult to solve in this original form without a good initialization process. We now show how to approximate the problem by a linear programming via linear approximation and variable relaxation as outlined in [8] by Jiang et al. To linearize the first term, the following scheme is applied. A basis $\mathcal{B}_\mathbf{s}$ is selected for the labels of each site $\mathbf{s}$. Then the label $\mathbf{f_s}$ can be represented as a linear combination of the label basis as $\mathbf{f_s} = \sum_{\mathbf{j} \in \mathcal{B}_\mathbf{s}} \xi_{\mathbf{s},\mathbf{j}} \cdot \mathbf{j}$, where $\xi_{\mathbf{s},\mathbf{j}}$ are real valued weighing coefficients. The labeling cost

of $\mathbf{f_s}$ can then be approximated by the linear combination of the basis labeling costs $c(\mathbf{s}, \sum_{\mathbf{j}\in\mathcal{B}_\mathbf{s}} \xi_{\mathbf{s},\mathbf{j}} \cdot \mathbf{j}) \approx \sum_{\mathbf{j}\in\mathcal{B}_\mathbf{s}} \xi_{\mathbf{s},\mathbf{j}} \cdot c(\mathbf{s}, \mathbf{j})$. We also further set constraints $\xi_{\mathbf{s},\mathbf{j}} \geq 0$ and $\sum_{\mathbf{j}\in\mathcal{B}_s} \xi_{\mathbf{s},\mathbf{j}} = 1$ for each site $\mathbf{s}$. Clearly, if $\xi_{\mathbf{s},\mathbf{j}}$ are constrained to be 1 or 0, and the basis contains all the labels, i.e., $\mathcal{B}_\mathbf{s} = \mathcal{L}_\mathbf{s}$, the above representation becomes exact. Note that $\mathbf{f_s}$ are *not* constrained to the basis labels, but can be any convex combination. To linearize the regularity terms in the nonlinear formulation we can represent a free variable by the difference of two nonnegative auxiliary variables and introduce the summation of the auxiliary variables into the objective function. If the problem is properly formulated, when the linear programming problem is optimized the summation will approach the absolute value of the free variable.

Based on this linearization process, a linear programming approximation of the problem can be stated as

$$\min \sum_{\mathbf{s}\in S} \sum_{\mathbf{j}\in\mathcal{B}_\mathbf{s}} c(\mathbf{s}, \mathbf{j}) \cdot \xi_{\mathbf{s},\mathbf{j}} + \sum_{\{\mathbf{p},\mathbf{q}\}\in\mathcal{N}} \lambda_{\mathbf{p},\mathbf{q}} \sum_{m=1}^{2} (f^+_{\mathbf{p},\mathbf{q},m} + f^-_{\mathbf{p},\mathbf{q},m})$$

$$s.t. \qquad \sum_{\mathbf{j}\in\mathcal{B}_\mathbf{s}} \xi_{\mathbf{s},\mathbf{j}} = 1, \forall \mathbf{s} \in S$$

$$\sum_{\mathbf{j}\in\mathcal{B}_\mathbf{s}} \xi_{\mathbf{s},\mathbf{j}} \cdot \phi_m(\mathbf{j}) = f_{\mathbf{s},m}, \ \forall \mathbf{s} \in S, m = 1, 2$$

$$f_{\mathbf{p},m} - f_{\mathbf{q},m} = f^+_{\mathbf{p},\mathbf{q},m} - f^-_{\mathbf{p},\mathbf{q},m}, \forall \{\mathbf{p},\mathbf{q}\} \in \mathcal{N}$$

$$\xi_{\mathbf{s},\mathbf{j}}, \ f^+_{\mathbf{p},\mathbf{q},m}, f^-_{\mathbf{p},\mathbf{q},m} \geq 0$$

where $\mathbf{f_s} = (f_{\mathbf{s},1}, f_{\mathbf{s},2})$. It is not difficult to show that either $f^+_{\mathbf{p},\mathbf{q},m}$ or $f^-_{\mathbf{p},\mathbf{q},m}$ will become zero and thus $f^+_{\mathbf{p},\mathbf{q},m} + f^-_{\mathbf{p},\mathbf{q},m} = |f_{\mathbf{p},m} - f_{\mathbf{q},m}|$ when the linear program is optimized. Therefore, the linear programming formulation is equivalent to the general nonlinear formulation if the linearization assumption $c(\mathbf{s}, \sum_{\mathbf{j}\in\mathcal{B}_\mathbf{s}} \xi_{\mathbf{s},\mathbf{j}} \cdot \mathbf{j}) = \sum_{\mathbf{j}\in\mathcal{B}_\mathbf{s}} \xi_{\mathbf{s},\mathbf{j}} \cdot c(\mathbf{s}, \mathbf{j})$ holds. In general situations, the linear programming formulation is an approximation of the original nonlinear optimization problem.

Property 1: *If* $\mathcal{B}_\mathbf{s} = \mathcal{L}_\mathbf{s}$, and *the cost function of its continuous extension* $c(\mathbf{s}, \mathbf{j})$ *is convex,* $\forall \mathbf{s} \in S$ *, the LP exactly solves the continuous extension of the discrete labeling problem.* $\mathcal{L}_\mathbf{s}$ is the label set of $\mathbf{s}$.

Proof: We just need to show when LP is optimized, the configuration $\{\mathbf{f}^*_\mathbf{s} = \sum_{\mathbf{j}\in\mathcal{B}_\mathbf{s}} \xi^*_{\mathbf{s},\mathbf{j}} \cdot \mathbf{j}\}$ also solves the continuous extension of the nonlinear problem. Since $c(\mathbf{s}, \mathbf{j})$ is convex, $\sum_{\mathbf{j}\in\mathcal{L}_\mathbf{s}} c(\mathbf{s}, \mathbf{j}) \xi^*_{\mathbf{s},\mathbf{j}} \geq c(\mathbf{s}, \mathbf{f}^*_\mathbf{s})$. And, when the LP is minimized we have $\sum_{\{\mathbf{p},\mathbf{q}\}\in\mathcal{N}} \lambda_{\mathbf{p},\mathbf{q}} \sum_{m=1}^{2} (f^+_{\mathbf{p},\mathbf{q},m} + f^-_{\mathbf{p},\mathbf{q},m}) = \sum_{\{\mathbf{p},\mathbf{q}\}\in\mathcal{N}} \lambda_{\mathbf{p},\mathbf{q}} ||\mathbf{f}^*_\mathbf{p} - \mathbf{f}^*_\mathbf{q}||$. Therefore

$$\min \sum_{\mathbf{s}\in S, \mathbf{j}\in\mathcal{L}_s} c(\mathbf{s}, \mathbf{j}) \xi_{\mathbf{s},\mathbf{j}} + \sum_{\{\mathbf{p},\mathbf{q}\}\in\mathcal{N}} \lambda_{\mathbf{p},\mathbf{q}} \sum_{m=1}^{2} (f^+_{\mathbf{p},\mathbf{q},m} + f^-_{\mathbf{p},\mathbf{q},m})$$

$$\geq \sum_{\mathbf{s}\in S} c(\mathbf{s}, f^*_\mathbf{s}) + \sum_{\{\mathbf{p},\mathbf{q}\}\in\mathcal{N}} \lambda_{\mathbf{p},\mathbf{q}} ||\mathbf{f}^*_\mathbf{p} - \mathbf{f}^*_\mathbf{q}||$$

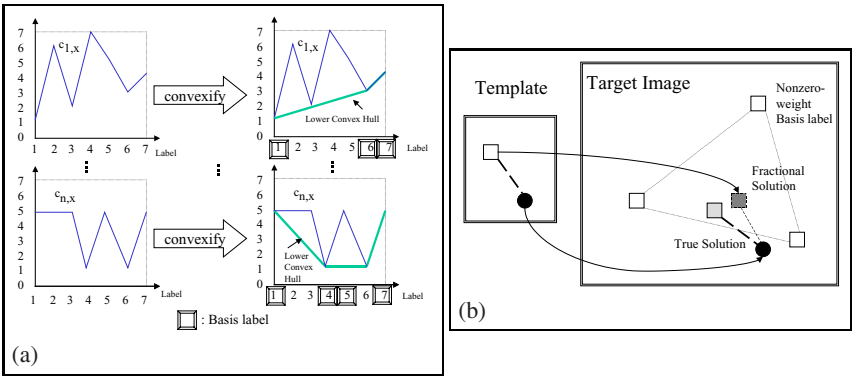


Fig. 1. (a): The convexification process introduced by LP relaxation. (b): An example when the single LP relaxation produces a fractional labeling.

According to the definition of *continuous extension*, $\mathbf{f}_{\mathbf{s}}^{*}$ are feasible solutions of continuous extension of the non-linear problem. Therefore the optimum of the linear programming problem is not less than the optimum of the continuous extension of the nonlinear problem. On the other hand, it is easy to construct a feasible solution of LP that achieves the minimum of the continuous extension of the nonlinear problem. The property follows.

In practice, the cost function $c(\mathbf{s}, \mathbf{j})$ is usually highly non-convex for each site $\mathbf{s}$. In this situation, the proposed linear programming model approximates the original non-convex problem by a convex programming problem.

Property 2: *For general cost function* $c(\mathbf{s}, \mathbf{j})$*, if* $\mathcal{B}_{\mathbf{s}} = \mathcal{L}_{\mathbf{s}}, \forall \mathbf{s} \in S$*, the linear programming formulation solves the continuous extension of the reformulated discrete labeling problem, with* $c(\mathbf{s}, \mathbf{j})$ *replaced by its lower convex hull for each site* $\mathbf{s}$*.*

Its proof is similar to Property 1, by replacing $c(\mathbf{s}, \mathbf{j})$ in the non-linear function with its lower convex hull. Fig. 1(a) illustrates the convexification effect introduced by LP relaxation.

Property 3: *For general cost function* $c(\mathbf{s}, \mathbf{j})$*, the most compact basis set* $\mathcal{B}_{\mathbf{s}}$ *contains the vertex coordinates of the lower convex hull of* $c(\mathbf{s}, \mathbf{j}), \forall \mathbf{s} \in S$*.*

By Property 3, there is no need to include all the labeling assignment costs in the optimization: we only need to include those corresponding to the basis labels. This is one of the key steps to speed up the algorithm.

Property 4: *If the lower convex hull* of *the cost function* $c(\mathbf{s}, \mathbf{j})$ is strictly convex*, nonzero weighting basis labels must be "adjacent".*

Proof: Here "adjacent" means the convex hull of the nonzero weighting basis labels cannot contain other basis labels. Assume this does not hold for a site $\mathbf{s}$, and the nonzero weighting basis labels are $\mathbf{j}_k$, $k = 1..K$. Then, there is a basis label $\mathbf{j}_r$ located inside the convex hull of $\mathbf{j}_k$, $k = 1..K$. Thus, $\exists \alpha_k$ such that $\mathbf{j}_r = \sum_{k=1}^{K} \alpha_k \mathbf{j}_k$ and $\sum_{k=1}^{K} \alpha_k = 1$,

$\alpha_k \geq 0$. According to *Karush-Kuhn-Tucker Condition* (*KKTC*), there exists $\lambda_1, \lambda_2, \lambda_3$ and $\mu_{\mathbf{j}}$ such that

$c(\mathbf{s}, \mathbf{j}) + \lambda_1 + \lambda_2\phi_1(\mathbf{j}) + \lambda_3\phi_2(\mathbf{j}) - \mu_{\mathbf{j}} = 0$ and $\xi_{\mathbf{s},\mathbf{j}}\mu_{\mathbf{j}} = 0, \mu_{\mathbf{j}} \geq 0, \forall \mathbf{j} \in \mathcal{B}_{\mathbf{s}}$

Therefore we have,

$c(\mathbf{s}, \mathbf{j}_k) + \lambda_1 + \lambda_2\phi_1(\mathbf{j}_k) + \lambda_3\phi_2(\mathbf{j}_k) = 0, k = 1..K$

$c(\mathbf{s}, \mathbf{j}_r) + \lambda_1 + \lambda_2\phi_1(\mathbf{j}_r) + \lambda_3\phi_2(\mathbf{j}_r) \geq 0$

On the other hand,

$c(\mathbf{s}, \mathbf{j}_r) + \lambda_1 + \lambda_2\phi_1(\mathbf{j}_r) + \lambda_3\phi_2(\mathbf{j}_r)$

$= c(\mathbf{s}, \sum_{k=1}^{K} \alpha_k \mathbf{j}_k) + \lambda_1 + \lambda_2\phi_1(\sum_{k=1}^{K} \alpha_k \mathbf{j}_k) + \lambda_3\phi_2(\sum_{k=1}^{K} \alpha_k \mathbf{j}_k)$

$< \sum_{k=1}^{K} \alpha_k c(\mathbf{s}, \mathbf{j}_k) + \lambda_1 + \lambda_2 \sum_{k=1}^{K} \alpha_k \phi_1(\mathbf{j}_k) + \lambda_3 \sum_{k=1}^{K} \alpha_k \phi_2(\mathbf{j}_k) = 0$

which contradicts the *KKTC*. The property follows.

After the convexification process, the original non-convex optimization problem turns into a convex problem and an efficient linear programming method can be used to yield a global optimal solution for the approximation problem. Note that, although this is a convex problem, a standard local optimization scheme is found to work poorly because of quantization noise and large flat areas in the convexified objective function.

Approximating the matching cost by its lower convex hull is also intuitively attractive since in the ideal case, the true matching will have the lowest matching cost and thus the optimization becomes exact in this case. In real applications, several target points may have equal matching cost and, even worse, some incorrect matching may have lower costs. In this case, because of the convexification process, in a one-step relaxation, the resulting fractional labeling could be not exactly the true solution, as shown in the Fig 1(b). In this simple image matching example, there are 2 sites in the source image and we construct a simple 2-node graph template. There are 5 target points in the target image. In the example, labels are the displacement vectors. We assume that a white rectangle will match a white rectangle with zero cost. And the circles will match with zero cost. Matching between different shape points has large matching cost. The light gray rectangle is in fact the true target for the white one in the source image, but the match cost is a very small positive number because of noisy measurement. By solving the LP relaxation problem, we get a fractional solution as illustrated in Fig 1(b) that has zero cost for LP's objective function but is not the true solution. Adjusting the smoothing parameter will not help because it already achieves the minimal zero cost. A traditional rounding scheme will try to round ξ into 0 and 1. Unfortunately, the rounding will drive the solution even farther from the true solution, in which the rectangle template node will match one of the white points in the target image. Intuitively, we can shrink the searching region for each site based on the current LP solution, and do a further search by solving a new LP problem in the smaller trust region. In the following section, we expand this idea and propose a successive convexification scheme to improve the approximation iteratively.

2.3 Successive Convexification Linear Programming

Here we propose a successive convexification linear programming method to solve the non-linear optimization problem, in which we construct linear programming recursively based on the previous searching result and gradually shrink the matching trust region systematically.

Assume $\mathcal{B}_{\mathbf{s}}^n$ to be the basis label set for site $\mathbf{s}$ at stage n linear programming. The trust region $\mathcal{U}_{\mathbf{s}}^n$ of site $\mathbf{s}$ is determined by the previous relaxation solution $\mathbf{f}_{\mathbf{s}}^{n-1} = (f_{\mathbf{s},1}^{n-1}, f_{\mathbf{s},2}^{n-1})$, and a trust region diameter d_n. We define $\mathcal{Q}_{\mathbf{s}}^n = \mathcal{L}_{\mathbf{s}} \cap \mathcal{U}_{\mathbf{s}}^n$. $\mathcal{B}_{\mathbf{s}}^n$ is specified by $\mathcal{B}_{\mathbf{s}}^n = \{$the vertex coordinates of the lower convex hull of $\{c(\mathbf{s},\mathbf{j}), \forall \mathbf{j} \in \mathcal{Q}_{\mathbf{s}}^n\}\}$, where $c(\mathbf{s},\mathbf{j})$ is the cost of assigning label $\mathbf{j}$ to site $\mathbf{s}$.

Algorithm 1. Successive Convexification Linear Programming

1. Set $n = 0$*; Set initial diameter* $= d_0$;
2. FOREACH$(\mathbf{s} \in S)$
3. { *Calculate the cost function* $\{c(\mathbf{s},\mathbf{j}), \forall \mathbf{j} \in \mathcal{Q}_{\mathbf{s}}^0\}$*;*
4. *Convexify* $\{c(\mathbf{s},\mathbf{j})\}$ *and find basis* $\mathcal{B}_{\mathbf{s}}^0$*;* }
5. *Construct and solve* $\mathcal{LP}_0$;
6. *WHILE* ($n \leq N$ *and* $d_n \geq 1$)
7. { $n \leftarrow n+1$*;*
8. $d_n = d_{n-1} - \delta_n$;
9. *FOREACH*$(\mathbf{s} \in S)$
10. { *IF* ($\mathcal{Q}_{\mathbf{s}}^n$ *is empty*) $\{\mathcal{Q}_{\mathbf{s}}^n = \mathcal{Q}_{\mathbf{s}}^{n-1}; \mathcal{U}_{\mathbf{s}}^n = \mathcal{U}_{\mathbf{s}}^{n-1};\}$
11. *ELSE update* $\mathcal{U}_{\mathbf{s}}^n$, $\mathcal{Q}_{\mathbf{s}}^n$*;*
12. *Reconvexify* $\{c(\mathbf{s},\mathbf{j})\}$ *and relocate basis* $\mathcal{B}_{\mathbf{s}}^n$*;* }
13. Construct and solve $\mathcal{LP}_n$;}
14. Output $f_{\mathbf{s}}^*, \forall \mathbf{s} \in S$*;*

Notice that the relaxed LP gives the lower bound of the original problem; It is easy to verify that the necessary condition for successive LP approaching the global minimum is $\mathcal{LP}_n \leq \mathcal{E}^*, n = 0..N$, where $\mathcal{E}^*$ is the global minimum of the non-linear problem. Since the global minimum of the function is unknown, we estimate an upper bound $\mathcal{E}^+$of $\mathcal{E}^*$ in the iterative process. The configuration of labels that achieves the upper bound $\mathcal{E}^+$ is composed of *anchors* — an anchor is defined as the control point of the trust region for the next iteration. We keep the anchor in the new trust region for each site and shrink the boundary inwards. If the anchor is on the boundary of the previous trust region, other boundaries are moved inwards. A simple scheme is to select anchors as the solution of the previous LP, $\mathbf{r}_{\mathbf{s}} = \mathbf{f}_{\mathbf{s}}^{(n-1)}$. Unfortunately, in the worse case, this simple scheme has solution whose objective function is arbitrarily far from the optimum. In fact, the fractional solution could be far away from the discrete label site. To solve the problem, we present a deterministic rounding process by checking the discrete labels and selecting the anchor that minimizes the non-linear objective function, given the configuration of fractional matching labels defined by the solution of the current stage. This step is similar to a single iteration of an ICM algorithm. In this step, we project a fractional solution into the discrete space. We call the new rounding selection scheme a *consistent rounding* process. Except for $\mathcal{LP}_1$, we further require that new anchors have energy not greater than the previous estimation: the anchors are updated only if new ones have smaller energy. The objective function for $\mathcal{LP}_n$ must be less than or equal to $\mathcal{E}^+$. This iterative procedure guarantees that the objective function of the proposed multi-step scheme is at least as good as a single relaxation scheme. In the following example, we use a simple scalar labeling problem to illustrate the solution procedure.

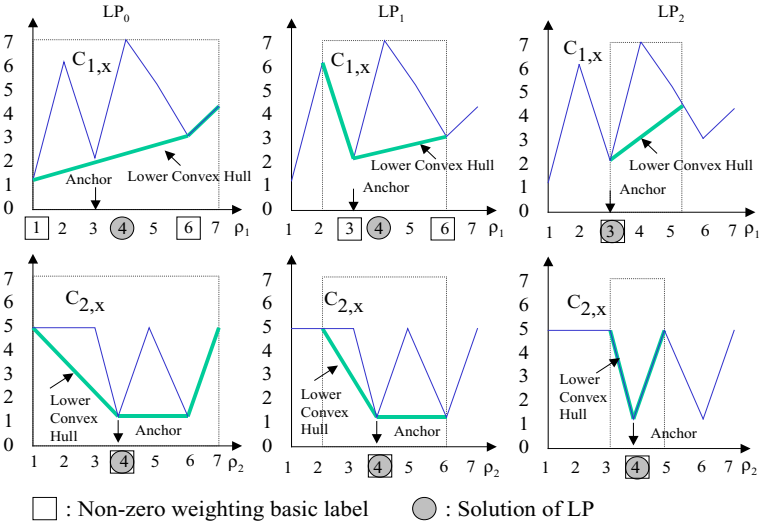


Fig. 2. Successive convexification LP

Example 1. (A scalar labeling problem): Assume there are two sites $\{1, 2\}$ and for each site the label set is $\{1..7\}$. The objective function is $\min_{\{\rho_1,\rho_2\}} c(1, \rho_1) + c(2, \rho_2) + \lambda|\rho_1 - \rho_2|$. In this example we assume that $\{c(1, j)\} = \{\,1.1\ 6\ 2\ 7\ 5\ 3\ 4\,\}$, $\{c(2, j)\} = \{\,5\ 5\ 5\ 1\ 5\ 1\ 5\,\}$ and $\lambda = 0.5$.

Based on the proposed scheme, the problem is solved by the sequential LPs: $\mathcal{LP}_0$, $\mathcal{LP}_1$ and $\mathcal{LP}_2$. In $\mathcal{LP}_0$ the trust regions of sites 1 and 2 are both $[1, 7]$. Constructing $\mathcal{LP}_0$ based on the proposed scheme corresponds to solving an approximated problem in which $\{c(1, j)\}$ and $\{c(2, j)\}$ are replaced by their lower convex hulls respectively (see Fig. 2). Step $\mathcal{LP}_0$ uses basis labels {1,6,7} for site 1 and basis labels {1, 4, 6, 7} for site 2. $\mathcal{LP}_0$ finds solution $\xi_{1,1} = 0.4$, $\xi_{1,6} = 0.6$, $\xi_{1,7} = 0$, $\rho_1 = (0.4 * 1 + 0.6 * 6) = 4$; and $\xi_{2,4} = 1$, $\xi_{2,1} = \xi_{2,6} = \xi_{2,7} = 0$, $\rho_2 = 4$. Based on the proposed rules for anchor selection, we fix site 2 with fractional label 4 obtained by solving $\mathcal{LP}_0$, and search the best label for site 1 in the region [1,7] using the non-linear objective function; we get the *anchor* 3 for site 1. Using similar method, we fix site 1 with its fractional label 4 and search the best label for site 2, and we get its anchor 4. At this stage, using anchor labels we get $\mathcal{E}^+ = c(1, 3) + c(2, 4) + 0.5 * |3 - 4| = 3.5$. Further, the trust region of $\mathcal{LP}_1$ is $[2, 6]$ for site 1 and $[2, 6]$ for site 2 by shrinking the previous trust region diameter by 2. The solution of $\mathcal{LP}_1$ is $\rho_1 = 4$ and $\rho_2 = 4$. The anchor is 3 for site 1 and 4 for site 2 with $\mathcal{E}^+ = 3.5$. Based on $\mathcal{LP}_1$, $\mathcal{LP}_2$ has new trust region $[3, 5] \times [3, 5]$ and its solution is $\rho_1 = 3$ and $\rho_2 = 4$. Since LP achieves the upper bound 3.5 in the trust region, there is no need to further shrink the trust region and the iteration terminates. It is not difficult to verify that the configuration $\rho_1 = 3$, $\rho_2 = 4$ achieves the global minimum. Fig. 2 illustrates the proposed successive convexification process method for this example.

Interestingly, for the above example ICM or even the graph cut scheme only finds a local minimum if initial values are not correctly set. For ICM, if ρ_2 is set to 6 and the updating is from ρ_1, the iteration will fall into a local minimum corresponding to

$\rho_1 = 6$ and $\rho_2 = 6$. The GC scheme based on α-expansion will have the same problem if the initial values of both ρ_1 and ρ_2 are set to 6.

A revised simplex method is used to solve the LP problem. Therefore, an estimate of the average complexity of successive convexification linear programming is $O(|S| \cdot |Q|^{1/2} \cdot (\log |Q| + \log |S|))$, where Q is the label set. Experiments also confirm that the average complexity of the proposed optimization scheme increases more slowly with the searching window size than previous methods such as the graph cut scheme, whose average complexity is linear with respect to $|Q|$.

2.4 Model Generation

The basic idea of body gesture recognition is to match a human body gesture image with different templates; The best matching template indicates the gesture and location of the human object in the image. The problem is challenging because we do not have a segmentation mask in the target image, and therefore we have to deal with strong background clutters. Another difficult problem is to make the algorithm resistant to different clothing and other large appearance changes. For gesture recognition problems, the features selected for the matching process must be insensitive to appearance changes of human objects. The edge map contains all the shape information of an object, and at the same time is not sensitive to color changes. Edge features have been widely applied in Chamfer matching schemes [5]. We propose the use of small blocks, centered on the edge pixels, of the *distance transform* of an image's edge map as the matching feature. A distance transform converts a binary edge map into its corresponding grayscale representation, where the intensity of a pixel is proportional to its distance to the nearest edge pixel. Denoting the square block of the distance transform of I's edge map centered at the edge pixel $\mathbf{x}$ as $\mathbf{d_x}(I)$, the cost of matching is defined as

$$C_{\mathbf{x},\mathbf{y}} = \frac{1}{\Delta^2 \sqrt{\sigma_\mathbf{x} \sigma_\mathbf{y}}} ||\mathbf{d_x}(I_\mathbf{s}) - \mathbf{d_y}(I_t)||$$

where I_s and I_t are the template and target images respectively; $||.||$ is the cityblock norm in this paper; $\sigma_\mathbf{x}$ and $\sigma_\mathbf{y}$ are the standard deviations of $\mathbf{d_x}(I_s)$ and $\mathbf{d_y}(I_t)$ respectively; Δ is the size of the square block. The orientation information is now integrated in the proposed feature. For instance, there is now a big difference for two features on orthogonal edges. In this paper, the features are randomly selected on the edges of the template. The neighboring relation $\mathcal{N}$ is defined by the edges of the graph generated by Delaunay triangulation of the feature points on the template. In this problem, source set S contains the feature points on the template, and labels are the displacement vectors of target points to each feature point on the template. Therefore, $c(\mathbf{s}, \mathbf{f_s})$ in the optimization problem equal $C_{\mathbf{s},\mathbf{f_s}+\mathbf{s}}$.

2.5 Similarity Measures

After finding the matches of the feature points in the template with corresponding points in the target image based on the proposed method, we need to further decide how similar these two constellations of matched points are and whether the matching result corresponds to the same event 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 warped template matching cost M, which is defined as the average absolute difference of the target image distance transform and the warped reference image distance transform in the region of interest. The warping is based on cubic spline. The total matching cost is simply defined as $M + \alpha D$, where α has a typical value from 0.1 to 0.5. Experiments show that only about 100 randomly selected feature points are needed in calculating D and M.

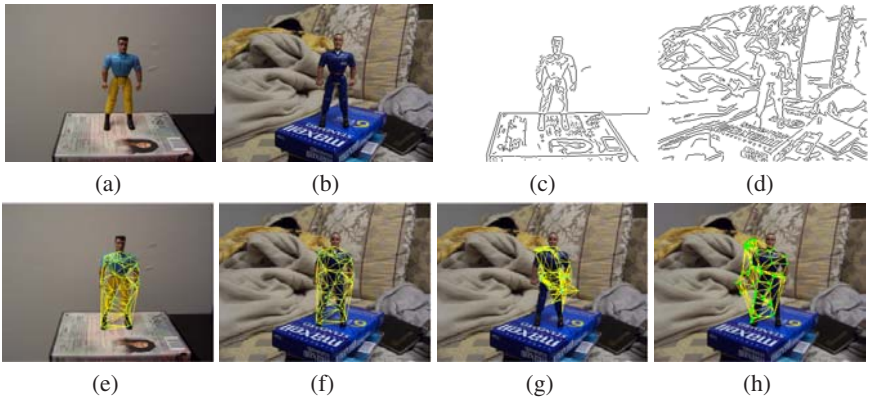


Fig. 3. An example where traditional methods fail. (a): Template image; (b): Target image; (c): Edge map of template image; (d): Edge map of target image; (e): Template mesh; (f): Matching result of the proposed scheme; (g): ICM matching result; (h): Graph cut matching result.

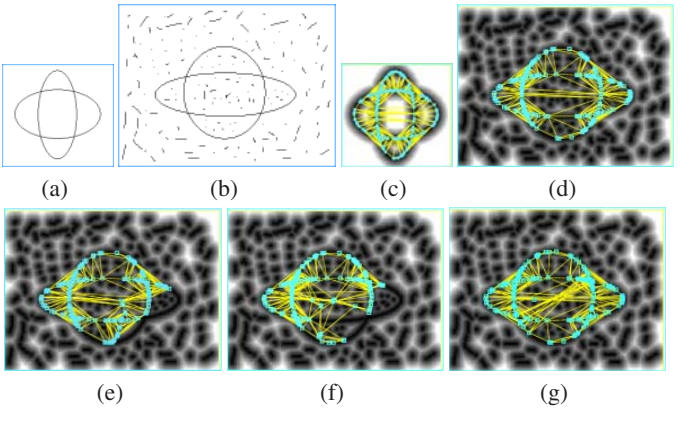


Fig. 4. Binary to grayscale. (a, b): Template image and target image. (c): Template model showing distance transform; (d): Matching result of proposed scheme; (e): Matching result by GC; (f): Matching result by ICM. (g): Matching result by BP.

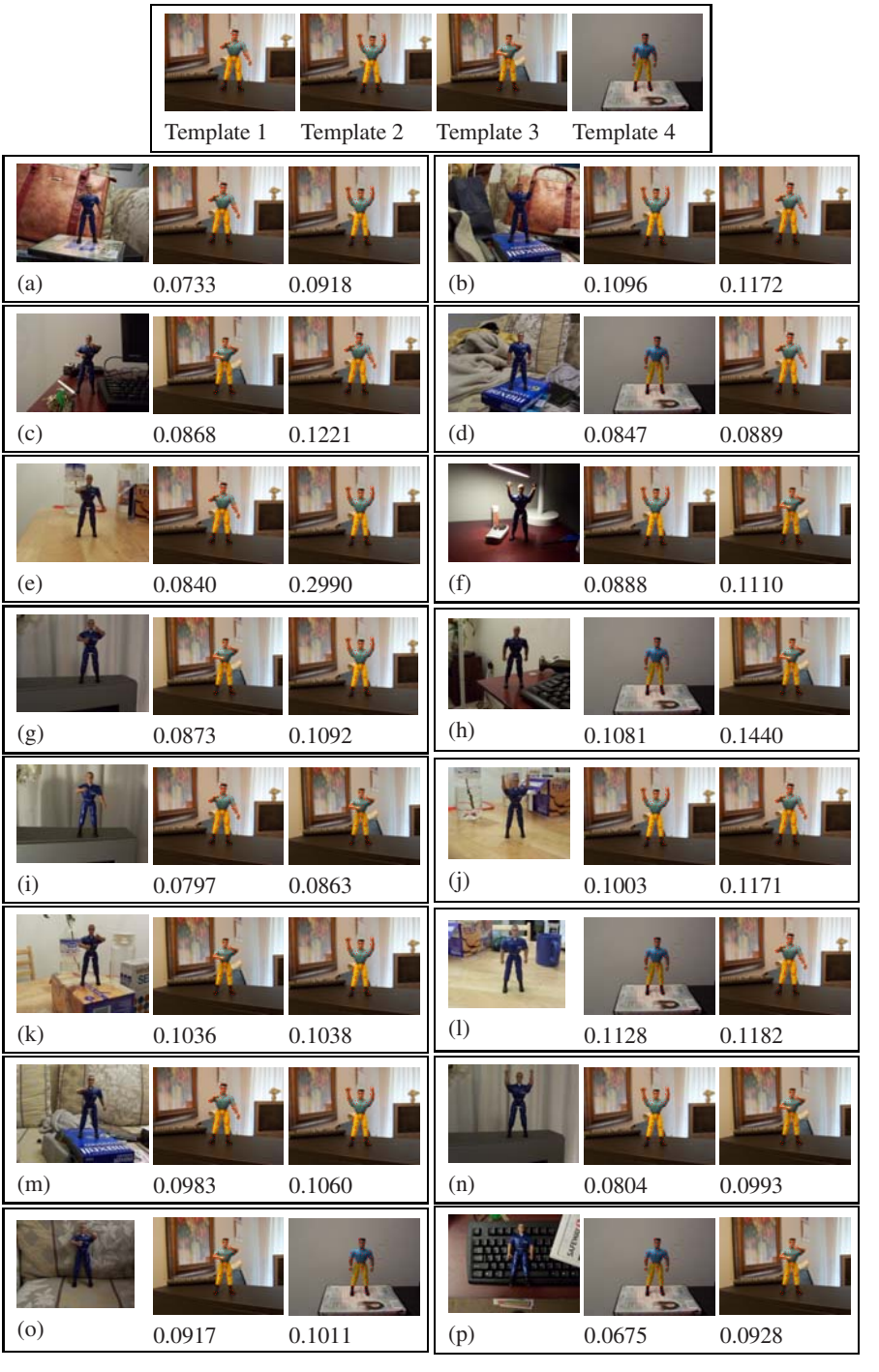
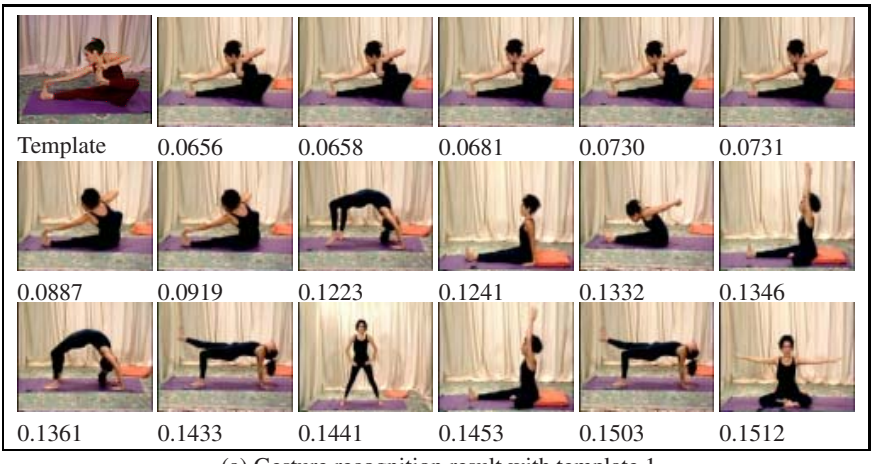
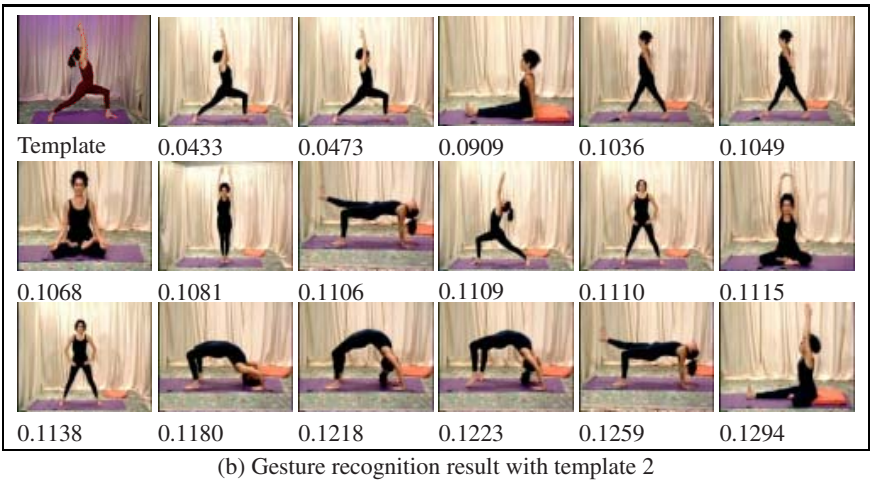


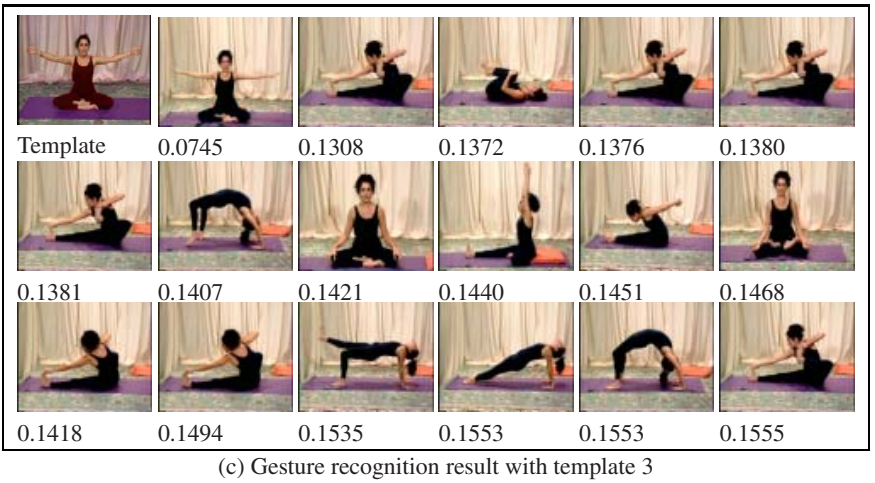
Fig. 5. Testing images and their top two matches from four body gestures



(a) Gesture recognition result with template 1

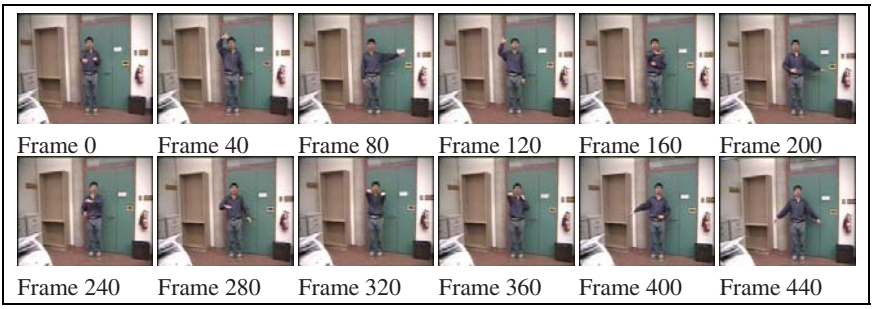


(b) Gesture recognition result with template 2

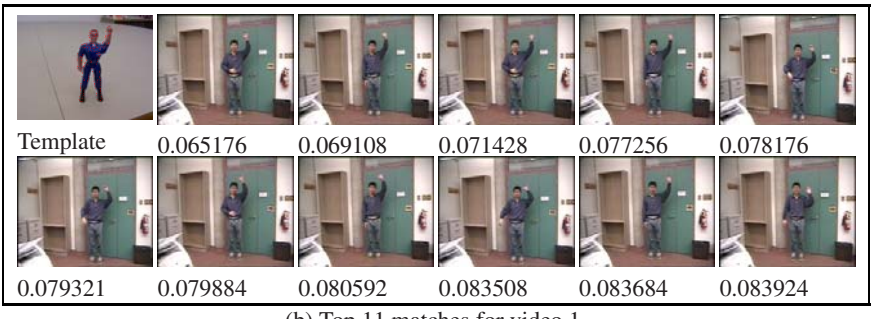


(c) Gesture recognition result with template 3

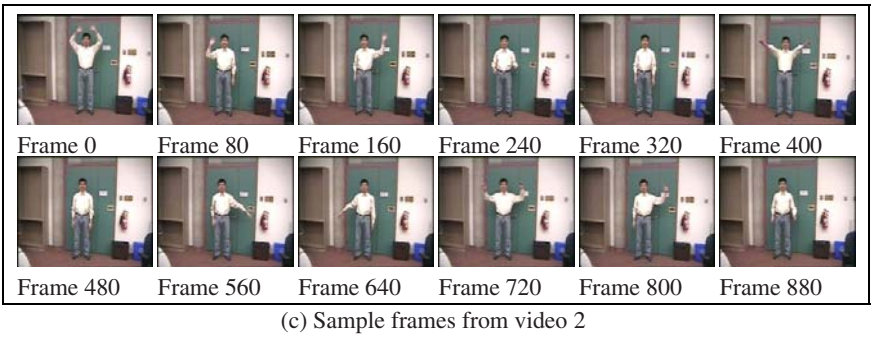
Fig. 6. Matching result for Yoga images. The first image in each subfigure is the template and the rest are the top 17 candidate matching images. Numbers show the matching cost.



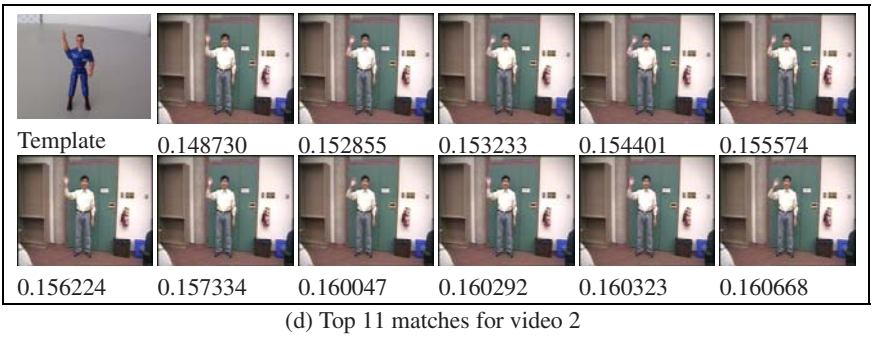
(a) Sample frames from video 1



(b) Top 11 matches for video 1



(c) Sample frames from video 2



(d) Top 11 matches for video 2

Fig. 7. Matching human gestures using flexible toy object template

3 Experimental Results

Fig. 3 shows the advantage of using our deformable matching scheme when we only have one template available. We try to match the distance transform of the template and target images. As shown in this example, greedy schemes such as ICM meet with great difficulty since there are a lot of ambiguities in matching distance transformation images. Comparing with the graph cut scheme, the proposed LP based method can solve the problem more robustly. Fig. 4 shows a comparison result using synthetic binary images. All the methods in the comparison use the same set of energy functions and parameter settings. With a 2.66GHz Pentium IV Linux machine, each LP iteration takes about 1 second for a problem with 100 nodes and 10000 target points. The typical number of iterations is 3 to 4 for most problems.

Fig. 5 shows body gesture recognition results using two articulated objects. Four body gestures are involved. A single template is generated for each gesture using the first object. The region of the object is set for the template object and about 100 features are randomly selected from the edge pixels automatically. Another object with different appearance is used for testing in different background settings. Distance transform images are used in matching to compensate for the appearance changes. A linear combination of the deformation measure D and the matching error M are used to form a matching score. We set the coefficient to be 0.1 for deformation D and unity for matching error M. Top two match candidates and their matching cost are shown in each of Figs. 5 (a) to (p). These experiments show that the proposed scheme can reliably match the target in complex background settings.

In another experiment, we study the following retrieval problem: we use a template image and retrieve the best match in an image data set. The data set is extracted from a video sequence. In this experiment, there is only one human object in the image. The search range is the whole target image. The template images and target images are extracted from different sections of the video, which have a large number of different body gestures and small number of similar body gestures. The character in the image has different clothing and somewhat different size in the template and target image. The test set contains 40 images with about 20 different gestures. Fig. 6 shows retrieval results for 3 different gestures. The matcher is reliable and all the correct matches are located in the first several best matches.

In Fig.7 we conducted experiments to test the performance of the proposed scheme in matching objects with large appearance differences. We use a toy as the template object and search for similar human body gestures in video sequences. Two sequences are used in testing. The first one shown in Fig.7 has 500 frames and the other has 1000 frames. There are fewer than 10% of true targets in the video sequence. The first sequence has a precision of 90% when the recall is 55%; Precision drops to 82% when recall goes up to 100%. The second sequence has a precision of 92% when the recall is 50%; Precision drops to 81% when recall reaches 100%.

4 Conclusion

We propose a novel linear programming method using successive convexification which is more efficient and effective than schemes such as the graph cut or belief propagation

methods for the object matching problem where a large searching range is involved. It can also solve problems for which other schemes fail. As well, we propose using distance transformations of the edge maps to match the template and target images. This representation facilitates matching some types of objects with large appearance variations. Experiments show very promising results for human gesture detection in cluttered environments. In future work, we will extend this method to dynamic gesture and human activity recognition problems. The proposed scheme has the potential to be directly applied to general object recognition problems.

References

[1] 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, 2003.
[2] P.F. Felzenszwalb, D.P. Huttenlocher, "Efficient matching of pictorial structures", CVPR, pp. II:66-73, 2000.
[3] R. Ronfard, C. Schmid, and B. Triggs, "Learning to parse pictures of people", ECCV, LNCS 2353, pp. 700-714, 2002.
[4] G. Mori, X. Ren, A. Efros, and J. Malik, "Recovering human body configurations: combining segmentation and recognition", CVPR, pp. II:326-333, 2004
[5] D. M. Gavrila and V. Philomin, "Real-time object detection for smart vehicles", ICCV, pp. 87-93, 1999.
[6] 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.
[7] G. Mori and J. Malik, "Estimating human body configurations using shape context matching", ECCV, LNCS 2352, pp. 666-680, 2002.
[8] H. Jiang, Z.N. Li, and M.S. Drew, "Optimizing motion estimation with linear programming and detail-preserving variational method", CVPR, pp. I:738-745, 2004.
[9] J. Besag, "On the statistical analysis of dirty pictures", J. R. Statis. Soc. Lond. B, 1986, Vol. 48, pp. 259-302.
[10] 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.
[11] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts", PAMI, Vol.23, pp. 1222-1239, 2001.
[12] H. Chui and A. Rangarajan, "A new algorithm for non-rigid point matching", CVPR, vol. 2, pp. 44-51, 2000.
[13] C. Chekuri, S. Khanna, J. Naor, and L. Zosin, "Approximation algorithms for the metric labeling problem via a new linear programming formulation", Symp. on Discrete Algs. pp. 109-118, 2001.
[14] A. Rosenfeld, R.A. Hummel, and S.W. Zucker, "Scene labeling by relaxation operations," IEEE Trans. Systems, Man, and Cybernetics, vol. 6, no. 6, pp. 420-433, 1976.
[15] B. Luo and E.R. Hancock, "Structural matching using the em algorithm and singular value decomposition," PAMI, vol. 23, pp. 1120-1136, 2001.
[16] J. Kleinberg and E. Tardos. "Approximation algorithms for classification problems with pairwise relationships: Metric labeling and Markov random fields". IEEE FOCS, pages 14–23, 1999.
[17] M. Ben-Ezra, S. Peleg, and M. Werman, "Real-time motion analysis with linear programming", ICCV, pp. 703-709, 1999.