Recognizing Posture in Pictures with Successive Convexification and Linear Programming

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

A proposed image-matching method uses successive convexification for recognizing human postures in cluttered images and videos. Using local image features, the scheme accurately locates and matches human objects over large appearance changes.

Recognizing human posture in images and videos is an important task in many multimedia applications, such as multimedia information retrieval, human–computer interaction, and surveillance.

Posture is a snapshot of human body parts' spatial configurations. A sequence of postures can be combined to generate meaningful gestures. Often, a posture in a single image also conveys meaningful information. For example, a human observer could disambiguate actions such as walking, running, standing, and sitting from a single image. In recent years, researchers have grown increasingly interested in recognizing human body postures in images or videos with a good deal of confounding background clutter (see the "Related Work on Posture Recognition" sidebar).

We've developed a posture detection method based on local image features and successive convexification image matching.[1,2] Image matching based on successive convexification operates very differently from previous methods such as relaxation labeling,[3] iterative conditional modes (ICM),[4] belief propagation,[5] graph cut,[6] and other convex-programming-based optimization schemes.[7,8] Our scheme represents target points for each template point with a small basis set. Successive convexification gradually shrinks the trust region for each template site and converts the original hard problem into a sequence of simpler convex programs. This speeds up searching, making the method well suited for large-scale matching and posture-recognition problems.

Posture recognition as a matching problem

Posture recognition is inherently an image-matching problem. After matching a posture template to a target object, we can compare their similarity and perform posture recognition. We can state posture matching as an energy-minimization problem:

$$\min\{E_{\text{Matching}} + \lambda \cdot E_{\text{Smooth}}\} \quad (1)$$

We'd like to find an optimal match from template feature points to target points. The goal is to minimize the matching cost (the first term in Equation 1) while smoothing the matching with the regularization (or smoothness) term (the second term in the equation). The multiplier balances the matching cost and the smoothness term.

In this article, we formulate the energy-minimization problem based on Equation 1:

$$\min_{\mathbf{f}}\left\{\sum_{s\in S} c\left(s, \mathbf{f}(s)\right) + \lambda \sum_{\{\mathbf{p},\mathbf{q}\}\in N} d\left(\mathbf{f}(\mathbf{q}) - \mathbf{f}(\mathbf{p}), \mathbf{q} - \mathbf{p}\right)\right\} \quad (2)$$

where S is the feature point set; N is the neighboring point set; $\mathbf{f}(s)$ maps 2D point s in the template image to a 2D point in the target image; $c(s, \mathbf{f}(s))$ is the cost of matching target point $\mathbf{f}(s)$ to s (for example, our block-based image measure); and $d(.,.)$ is a distance function.

We focus on the problem where $d(.,.)$ is the city block distance. The smoothness term enforces that neighboring template points shouldn't travel too far from each other, once matched. There are different ways to define the neighbor pair set. One natural way is to use a Delaunay triangulation over the feature points in the template and identify any two points connected by a Delaunay graph edge as neighbors.

1070-986X/07/$25.00 © 2007 IEEE

Published by the IEEE Computer Society

Related Work on Posture Recognition

Many experimental and commercial systems exist for recognizing human body configurations in controlled environments. Examples include the Massachusetts Institute of Technology's Media Lab Kidsroom (http://vismod.media.mit.edu/vismod/demos/kidsroom/kidsroom.html), Alive,[1] Emering et al.'s gesture-recognition system,[2] and Vivid Group's gesture-recognition system (http://www.vividgroup.com). These systems rely on segmenting human objects from the background in a specific restricted environment or by position/velocity sensors attached to human subjects. To facilitate the segmentation process, other systems use infrared[3] or multiple cameras.[4] These systems are more expensive to deploy than simple monocular visible-light camera systems.

In uncontrolled environments, recognizing human body postures becomes a challenging problem because of background clutter, articulated structures of human bodies, and a large variability of clothing. To overcome these difficulties, some researchers have proposed methods based on directly matching templates to targets. One method is to detect human body parts[5] and their spatial configuration in images, as Figure A1 illustrates. Body-part methods only involve a few templates to represent each body part. However, body parts are difficult to locate in many uncontrolled cases, mainly because of clothing changes, occlusion, and body-part deformation. Currently, body-part-based schemes are used for recognizing relatively simple human postures such as running.[6] Another method recognizes human postures based on small local image features. As Figure A2 illustrates, this scheme matches postures as whole entities and doesn't distinguish body parts explicitly.

Most previous methods based on matching local image features[7,8] assume a relatively clean background. When background clutter increases, distinguished features are weakened and simple matching schemes can't generate desirable results. Our successive convexification-based scheme robustly and efficiently solves this problem.

References

1. A.P. Pentland et al., "Perceptive Spaces for Performance and Entertainment: Untethered Interaction Using Computer Vision and Audition," *Applied Artificial Intelligence*, vol. 11, no. 4, 1997, pp. 267-284.
2. L. Emering and B. Herbelin, "Body Gesture Recognition and Action Response," *Handbook of Virtual Humans*, John Wiley & Sons, 2004, pp. 287-302.
3. F. Sparacino, N. Oliver, and A. Pentland, "Responsive Portraits," *Proc. 8th Int'l Symp. Electronic Art* (ISEA 97), 1997.
4. K.M.G. Cheung, S. Baker, and T. Kanade, "Shape-from-Silhouette of Articulated Objects and Its Use for Human Body Kinematics Estimation and Motion Capture," *Proc. IEEE Conf. Computer Vision and Pattern Recognition* (CVPR 03), vol. 1, IEEE CS Press, 2003, pp. 77-84.
5. P.F. Felzenszwalb and D.P. Huttenlocher, "Efficient Matching of Pictorial Structures," *Proc. IEEE Conf. Computer Vision and Pattern Recognition* (CVPR 00), vol. 2, IEEE CS Press, 2000, pp. 66-73.
6. D. Ramanan, D.A. Forsyth, and A. Zisserman, "Strike a Pose: Tracking People by Finding Stylized Poses," *Proc. IEEE Conf. Computer Vision and Pattern Recognition* (CVPR 05), IEEE CS Press, 2005, pp. 271-278.
7. S. Carlsson and J. Sullivan, "Action Recognition by Shape Matching to Key Frames," *Proc. IEEE Computer Soc. Workshop Models Versus Exemplars on Computer Vision*, 2001.
8. G. Mori and J. Malik, "Estimating Human Body Configurations Using Shape Context Matching," *Proc. European Conf. Computer Vision* (ECCV 02), LNCS 2352, Springer, 2002, pp. 666-680.

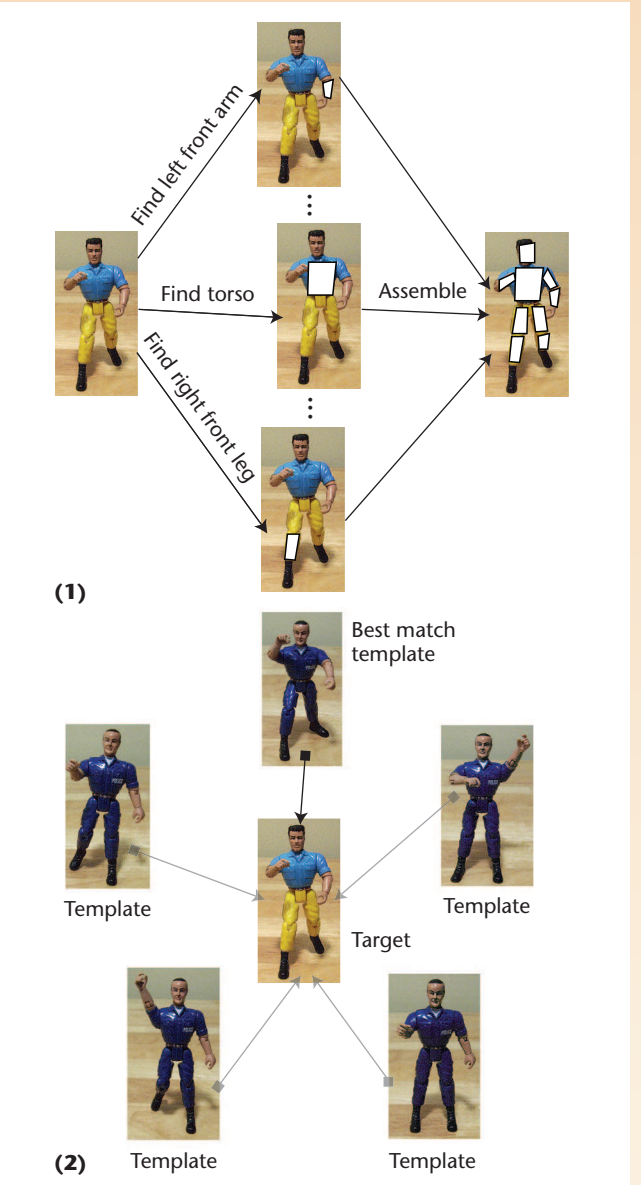


Figure A. Posture recognition by matching: (1) body-part based, and (2) matching a whole entity using local image features.

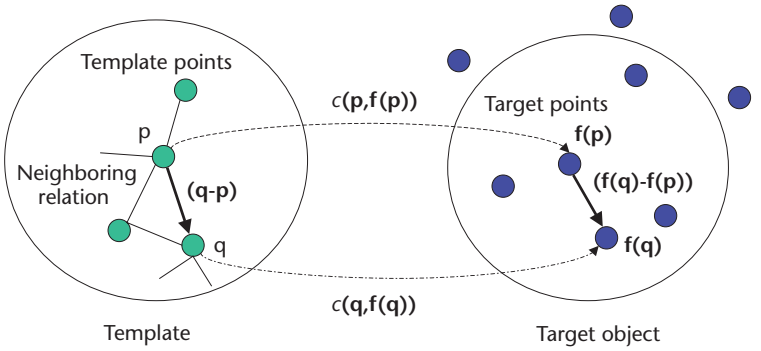


Figure 1. Object matching.

Figure 1 illustrates the matching problem. In the figure, points **p** and **q** are two neighboring template feature points and their targets are **f**(**p**) and **f**(**q**), respectively. Intuitively, we should minimize the matching costs while trying to make the matching consistent by minimizing the difference of vectors **q** – **p** and **f**(**q**) – **f**(**p**).

Features for matching

For posture-recognition problems, the features selected for the matching must be insensitive to human objects' appearance changes. The edge map contains most of an object's shape information, and at the same time isn't very sensitive to color changes.

Researchers have widely applied edge features in Chamfer (edge-based) matching[9] and shape context[10] matching. We've found that small blocks centered on the edge pixels of a distance transform image are expressive local features. Here, a distance transform converts a binary edge map into a corresponding grayscale representation, with the intensity of a pixel proportional to its distance to the nearest edge pixel.

To incorporate more context information, we can further apply a log-polar transform to a distance transform image.[2] We can then represent the matching cost as the normalized mean absolute difference between these local image features. Local image features aren't reliable in image matching, so we need a robust matching scheme.

Linear programming matching

The energy-optimization problem in Equation 2 is usually nonlinear and highly nonconvex—that is, it has many local minima. Such problems are difficult to solve without a good initialization process. Instead of trying to optimize the problem directly, we convert it into an approximated linear programming (LP) problem.[1,2]

Basically, we introduce weights that can be interpreted as a set of (float) soft decisions for matching target points to template feature points. We can then represent a target point as the linear combination of representative target points—the *basis target points*. We approximate the matching cost as the weighted sum of these basis points' costs. Finally, we linearize the smoothness term using auxiliary variables.

In some cases, we can use this linear program to exactly solve the matching problem's continuous extension; in general situations, it's an approximation of the original problem.

The "Properties of LP Formulation" sidebar lists some of LP's properties. Figure 2 illustrates a cost surface, its lower convex hull, and the basis target points.

After convexification, the original nonconvex optimization problem becomes a convex problem and we can use an efficient LP method to yield a global optimal solution for the approximation problem.

Successive convexification

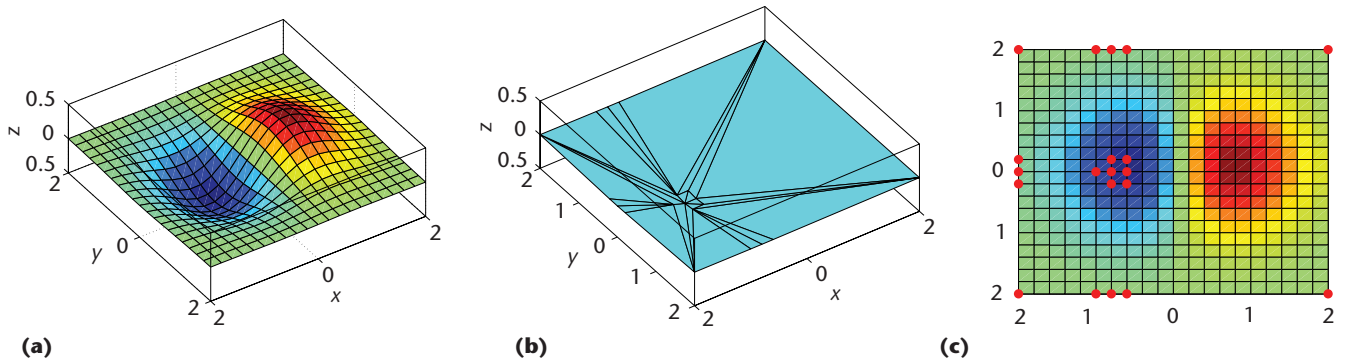
Because of the LP relaxation's convexification effect, the approximation is coarser for larger search regions in the target image. Thus the LP solution will be more precise if we can narrow the searching range. We thus propose a successive relaxation scheme to solve the coarse approxi-

Properties of LP Formulation

The linear programming formulation has several interesting properties:

- For general cost functions, the LP formulation solves the continuous extension of the reformulated matching problem, with each matching cost surface replaced by its lower convex hull.
- The most compact basis set contains the vertex coordinates of the matching cost surface's lower convex hull. By this property, there's no need to include all the matching costs in the optimization: we need only include those corresponding to the basis target points. This is one of the key steps to speeding up the algorithm.
- If the cost function's convex hull is strictly convex, nonzero weighting basis points must be adjacent. Here, adjacent means the convex hull of the nonzero weighting basis target points can't contain other basis target points.
- If we solve the LP problem by the simplex method, each feature point in the template will have at most three nonzero-weight target points.

The optimization is reduced to just a fast descent through a few triangles in the target point space for each site.



mation problem. We construct linear programs recursively, based on the previous search result, gradually and systematically shrinking each site's trust region. However, we convexify the original cost function again (that is, we "reconvexify") in the smaller region. Figure 3 shows this procedure.

We use anchors to control the trust regions. To locate anchors, we apply a consistent rounding process[2] to the previous stage's LP solution. The new trust region for each site is a smaller rectangular region containing the anchor—for example, a region centered on the corresponding anchor. Example 1 illustrates the successive convexification procedure for a simple 1D matching problem.

Example 1 (1D problem). Assume there are two sites {1, 2}, and for each site {1..7} is the target point set. The objective function is

$$\min_{\{\rho_1,\rho_2\}}\left[c(1,\rho_1)+c(2,\rho_2)+\lambda|\rho_1-\rho_2|\right]$$

In this example, we assume the matching costs are $\{c(1, j)\} = \{ 1.1, 6, 2, 7, 5, 3, 4\}$, $\{c(2, j)\} = \{5, 5, 5, 1, 5, 1, 5\}$, and $\lambda = 0.5$.

Using our scheme, we solve the problem by the sequential LPs: LP_0, LP_1, and LP_2.

In LP_0, the trust regions of sites 1 and 2 are both [1, 7]. Constructing LP_0 based on our scheme corresponds to solving an approximated problem in which we replace $\{c(1, j)\}$ and $\{c(2, j)\}$ by their lower convex hulls (see Figure 4, next page). Step LP_0 uses basis labels {1, 6, 7} for site 1 and basis labels {1, 4, 6, 7} for site 2. Then LP_0 has the 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 rules for anchor selection,[2] we fix site 2 with LP_0 solution 4, and search for the best target point for site 1 in the region [1, 7] using the nonlinear objective function. This gives us the anchor 3 for site 1. Using a similar method for site 2, we get its anchor 4.

Further, we get LP_1's trust region of [1, 5] × [2, 6] by shrinking the previous trust region diameter by a factor of 2. LP_1's solution is $\rho_1 = 3$ and $\rho_2 = 4$. The new anchor is 3 for site 1 and 4 for site 2.

Based on LP_1, LP_2 has a new trust region [2, 4] × [3, 5]. Its solution is $\rho_1 = 3$ and $\rho_2 = 4$. Because 3 and 4 are the anchors for sites 1 and 2, and in the next iteration the diameter shrinks to zero, the iteration terminates. It isn't difficult to verify that the configuration $\rho_1 = 3$, $\rho_2 = 4$ achieves the global minimum.

Interestingly, for the above example, the ICM and graph cut only find a local minimum, if ini-

Figure 2. Lower convex hull: (a) a cost surface, (b) lower convex hull facets, and (c) the label basis B_s, which contains coordinates of the lower convex hull vertices (solid dots are basis points).

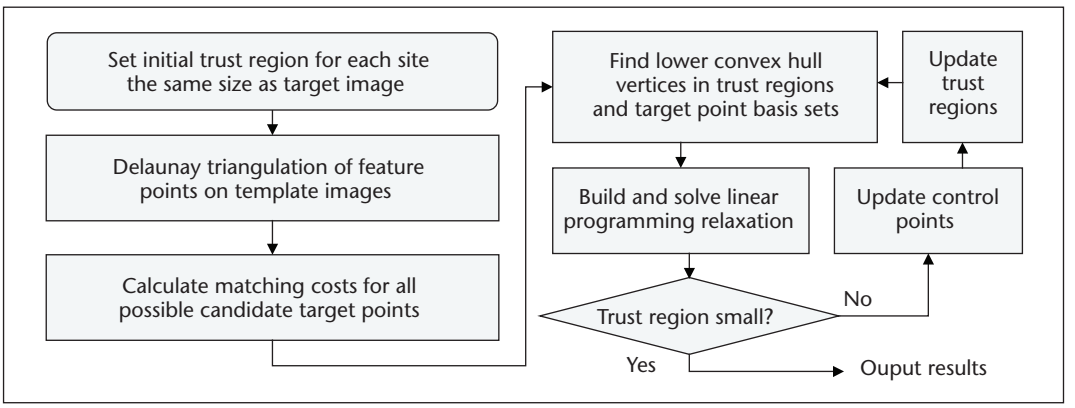


Figure 3. Object matching using successive convexification.

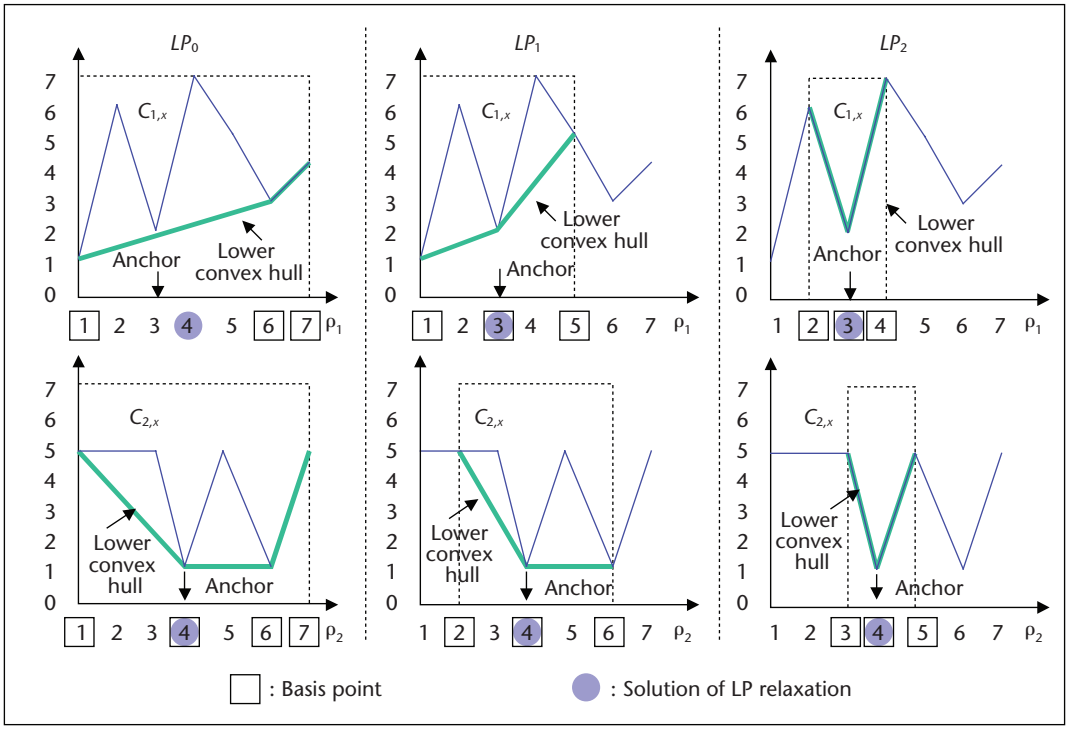


Figure 4. An example of successive convexification matching.

tial values aren't correctly set. For ICM, if ρ_2 is set to 6 and the update is from ρ_1, the iteration will fall into a local minimum corresponding to $\rho_1 = 6$ and $\rho_2 = 6$. The graph cut scheme based on α-expansion will have the same problem if we set the initial values of both ρ_1 and ρ_2 to 6.

Example 2 (2D problem). Figure 5 shows how we match a triangle in clutter using successive convexification. The figure illustrates the trust region updating and convexification process for two points on the template. The black rectangles in Figures 5d, 5e, and 5f indicate the trust regions for the two selected points in three successive LP stages. Figures 5g through 5l show the convexified matching cost surfaces for each site in these trust regions. A very small number of vertices corresponding to the basis target points support these convex surfaces. The three-stage successive convexification scheme accurately locates the target in clutter.

With the simplex method, an estimate of a successive reconvexification LP's average complexity is $O(|S| \cdot (\log |L| + \log |S|))$, where S is the set of template feature points and L is the target point set. Experiments also confirm that our optimization scheme's average complexity increases more slowly with the target point set's size than previous methods, such as belief propagation, whose average complexity is proportional to $|L|^2$.

Measuring similarity

After posture matching from a template to a target object, we need to decide how similar these two constellations of matched points are and whether the matching result corresponds to the same posture as in the exemplar. We use the following quantities to measure the difference between the template and the matching object.

We first define measure D as the average pairwise length changes from the template to the target.

To compensate for the global deformation, we estimate a global affine transform based on the matching and apply it to the template points before calculating D. We further normalize D with respect to the template's average edge length. The second measure is the average feature-matching cost M. We define the matching score simply as the linear combination of D and M. Experiments show that we need only about 100 randomly selected feature points in calculating D and M.

We can extend this posture-matching method to matching video sequences to detect actions[2] by introducing a center-continuity constraint.

Experimental results

We conducted experiments to compare our matching scheme with a belief propagation and ICM using synthetic ground truth data. Other

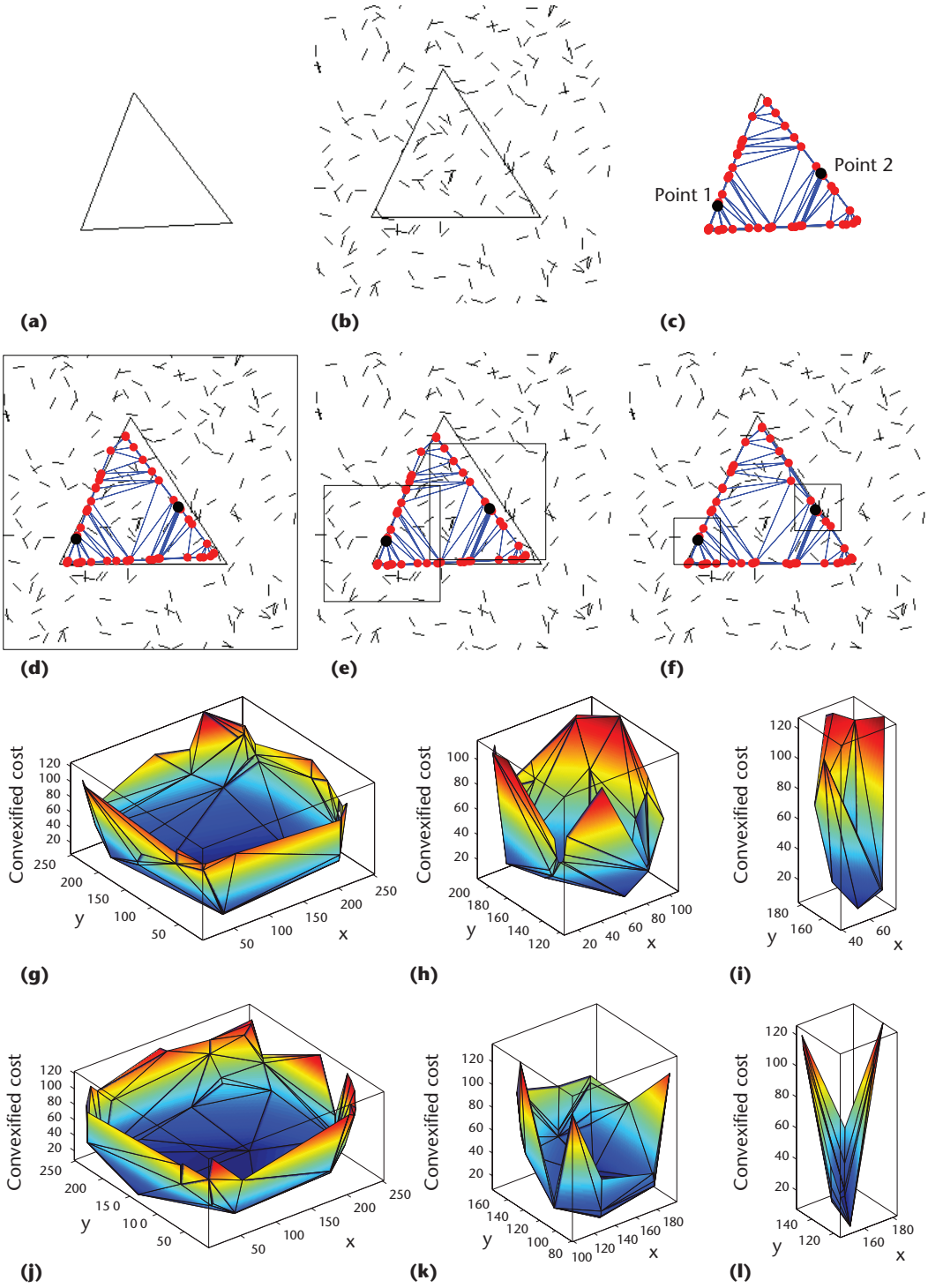


Figure 5. Object matching in cluttered image: (a) template, (b) target in clutter, (c) template mesh, (d) LP_0 matching, (e) LP_1 matching, and (f) LP_2 matching. Corresponding convexified matching cost surfaces: (g) point 1, stage 0; (h) point 1, stage 1; (i) point 1, stage 2; (j) point 2, stage 0; (k) point 2, stage 1; and (l) point 2, stage 2.

experiments tested the proposed human posture detection scheme using real video sequences.

Matching random dots

We compared the performance of successive convexification LP (SC-LP) with belief propagation and ICM for binary object detection in clutter. In our experiments, we generate the templates by randomly placing 50 black dots in a 128×128 white background image. We then synthesize a 256×256 target image by randomly translating and perturbing the block dot positions from those in the template. Next, we add random noise dots to the target image to simulate background clut-

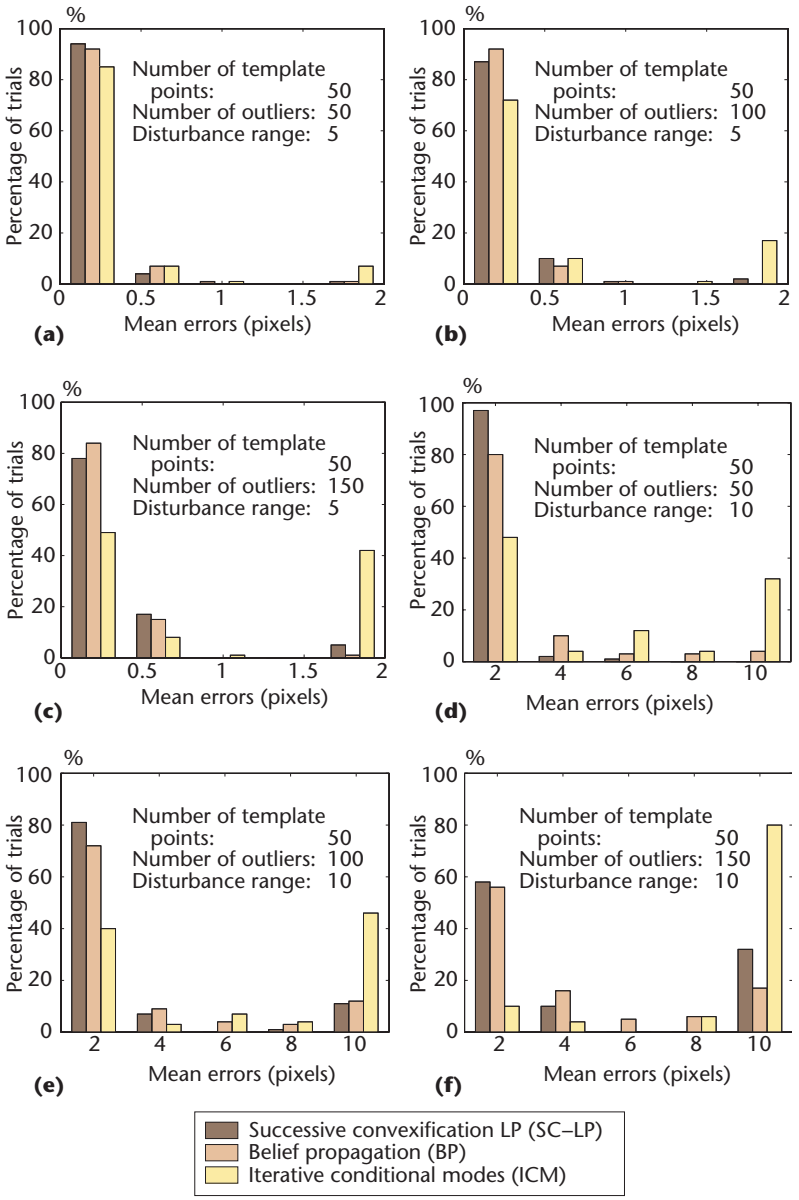


Figure 6. Histogram of matching errors using successive convexification LP (SC-LP), belief propagation, and ICM: (a) 50 outliers and disturbance in [0,5], (b) 100 outliers and disturbance in [0,5], (c) 150 outliers and disturbance in [0,5], (d) 50 outliers and disturbance in [0,10], (e) 100 outliers and disturbance in [0,10], and (f) 150 outliers and disturbance in [0,10].

ter. For each testing situation, we generate 100 template and target images. Then we match the gray-level distance transformation of the template and target images.

Figure 6 compares results using our matching scheme with results using belief propagation and ICM. The histograms show the methods' error distributions. In this experiment, all of the methods use the same energy function. SC-LP performs similarly to belief propagation and much better than the greedy ICM scheme in cases of large distortion and cluttered environments. SC-LP is much more efficient than belief propagation when the number of target points exceeds 100. With a 2.8-GHz PC, for a matching problem with 80 template points and 1,000 target candidate points, SC-LP has an average matching time of 10 seconds with four iterations, whereas belief propagation takes about 100 seconds for just one iteration.

Finding postures in video

We first test our method for finding postures in video sequences with an (approximately) 30-minute-long yoga sequence. We chose three different posture exemplars from another section of the video. By specifying the region of interest, we automatically generate graph templates from the exemplars. We then compare each template with video frames in the test video. Figures 7b, 7c, and 7d show the short lists based on their matching scores. The templates are shown as the first image in each short list. Figure 8a shows the recall-precision curves.

Figure 9 (page 34) illustrates our scheme's performance in matching objects with large appearance differences. We use a flexible toy as the template object and search video sequences for similar postures of actual human bodies. We test two sequences: the first, shown in Figure 9, has 500 frames, and the other has 1,000 frames. The video sequence contains fewer than 10 percent true targets. The vertical and horizontal edges in the background are similar to the edge features on human bodies, presenting a major challenge for object location and matching. Figures 9b and 9d show the short lists of matching results, and Figure 8b shows the recall-precision curves.

In another experiment, we search an (approximately) 30-minute-long figure-skating sequence to locate similar postures as exemplars. The figure-skating program contains five skaters with quite different clothing. The audience in the scene presents strong background clutter, which can cause problems for most matching algorithms. The video's sampling rate is 1 frame per second. Figure 10 shows short lists of posture searching based on the matching scores for three different postures. The first image in each short list shows the templates. Figure 8c shows the recall-precision curves.

In the previous three experiments, we search for postures in videos containing a single object in each video frame. Next, we consider posture recognition for videos that might contain multi-

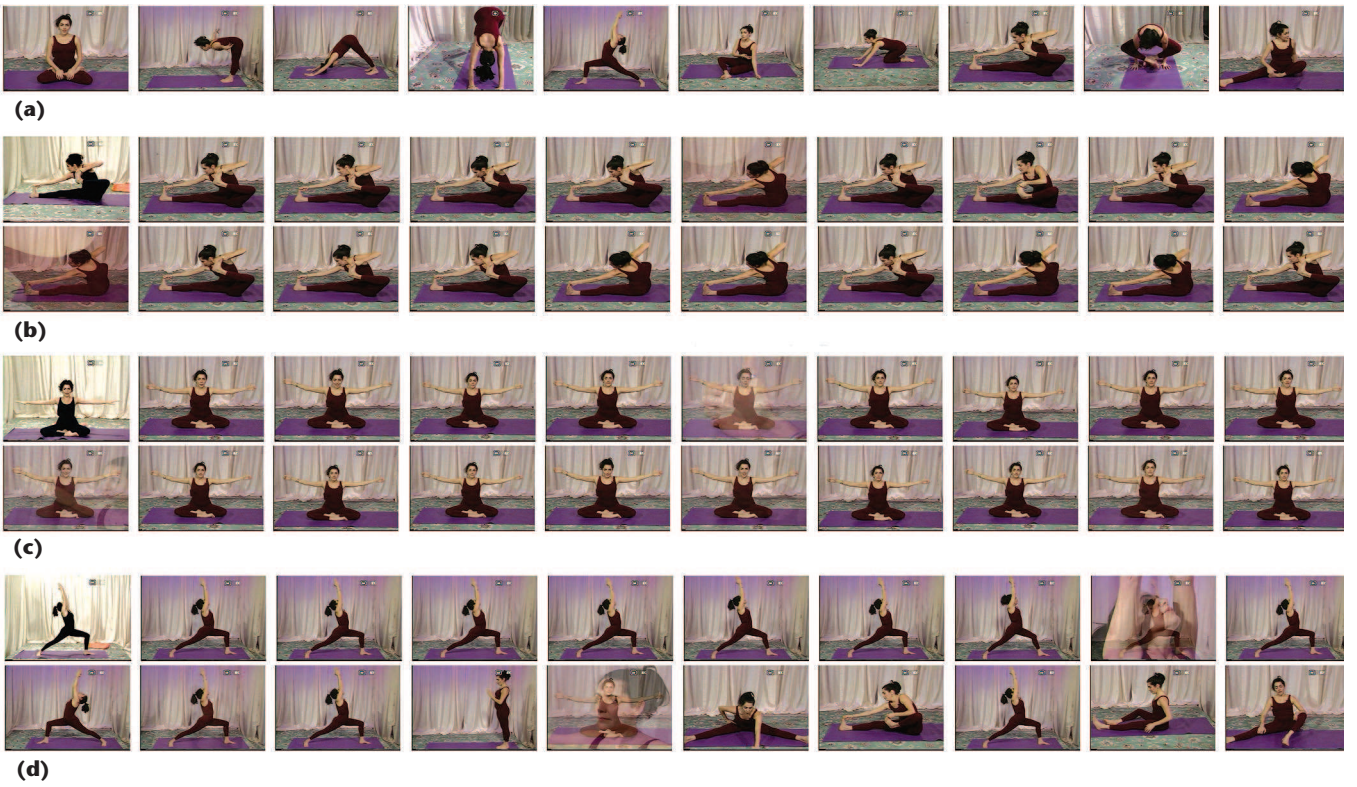


Figure 7. Matching human postures in a yoga sequence: (a) sample frames from video, (b) short list of matching for yoga posture 1, (c) short list of matching for yoga posture 2, and (d) short list of matching for yoga posture 3.

ple objects in each frame. We want to locate objects with specific postures in hockey games. Hockey is a fast-paced game, with quick player movements and camera motion. Detecting hockey players' activities is an interesting and challenging application. The background audience and patterns on the ice also make posture recognition difficult.

To deal with multiple targets in images, we first apply composite filtering. We construct the composite template as the average of 200 randomly selected hockey players. To reduce clothing's influence, we convert these images to distance-transformed images for composite template construction and composite filtering. For each input video frame, the positions of local valleys of the composite filter residue image are potential object centers. We cut rectangular image patches centered on these object centers from each video frame and forward them to LP detail matching to compare their similarity with the posture template.

Figure 11a (page 35) shows the short list of searching for a shooting action in a 1,000-frame video sequence. Our scheme successfully detects two instances of the shooting action at the top of

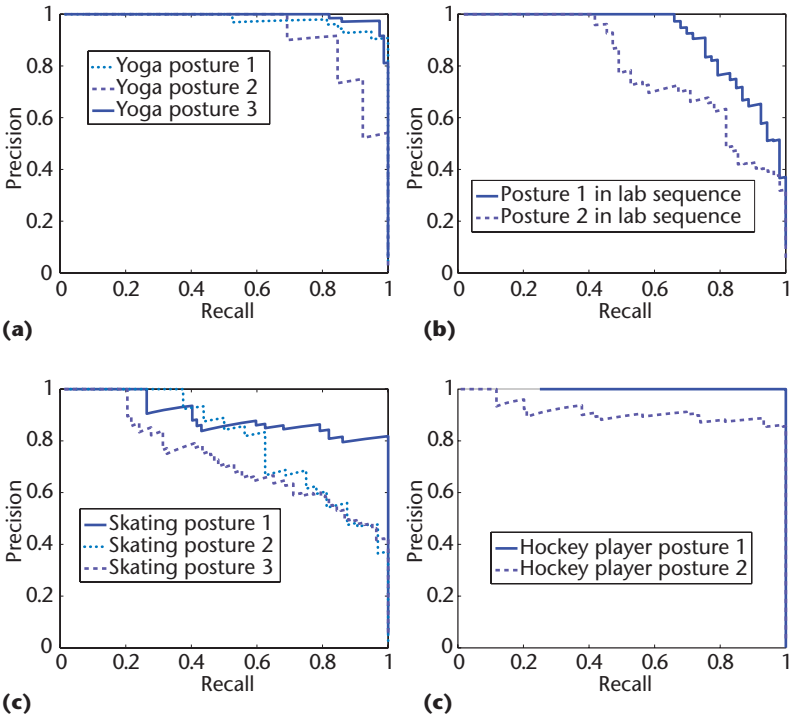


Figure 8. Recall-precision curves: (a) yoga sequences, (b) lab sequences, (c) figure-skating sequences, and (d) hockey sequences.

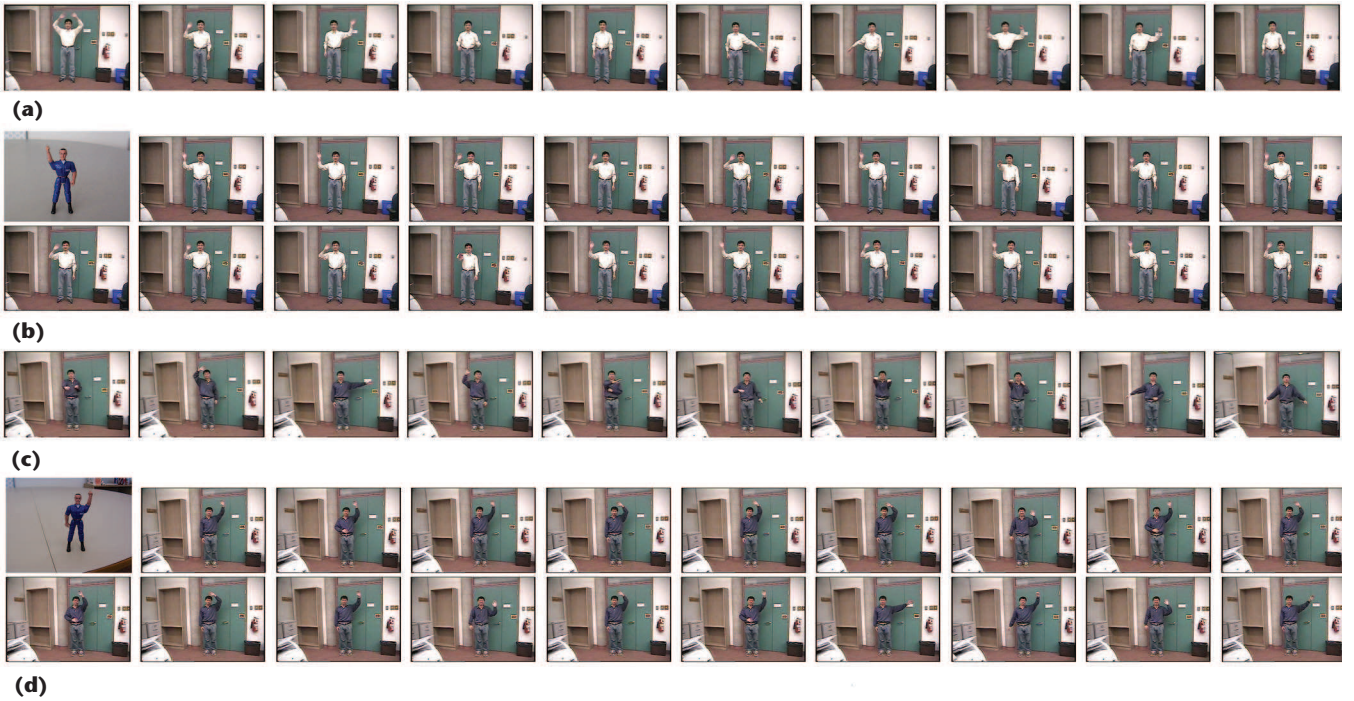


Figure 9. Matching human postures using a flexible toy object template: (a) sample frames from video 1, (b) top 19 matches for video 1, (c) sample frames from video 2, and (d) top 19 matches for video 2.

Figure 10. Figure-skating posture detection: (a) top 19 matches for figure-skating posture 1 (the first image is the exemplar), (b) top 19 matches for figure-skating posture 2 (the first image is the exemplar), and (c) top 19 matches for figure-skating posture 3 (the first image is the exemplar).

the short list. Figure 11b shows another posture-detection result, for a 1,000-frame video with another posture template. The figure shows a short list of video frames and hockey players based on the matching scores. We define the matching score for a video frame as the smallest object-matching score in the frame. Figure 8d shows the recall-precision curves.

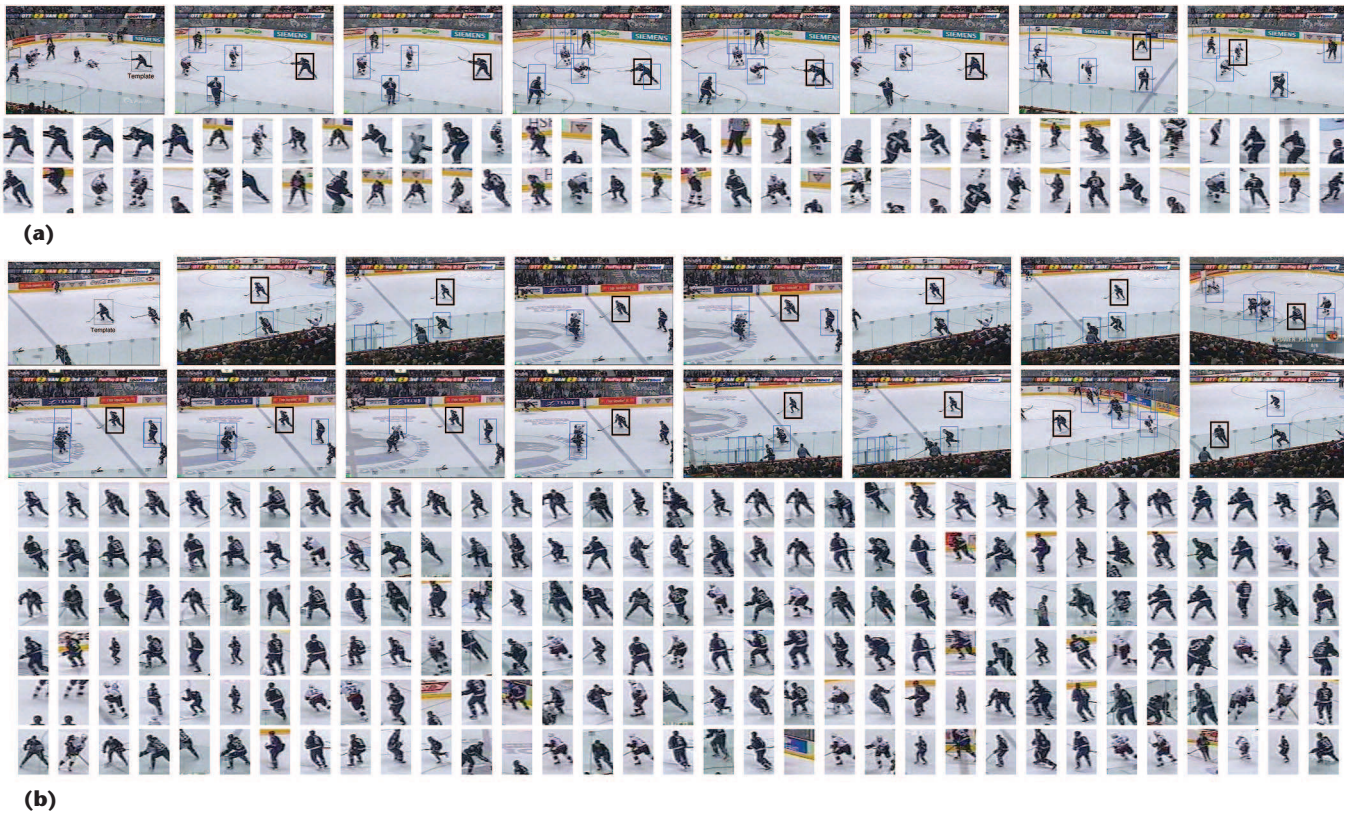


Figure 11. Finding postures in hockey: (a) locating shooting posture in video with an exemplar 1 template, and (b) locating postures in video with exemplar 2.

Figure 12. Figure-skating posture detection using Chamfer matching. The first image is the template image.

We also compare Chamfer matching with the proposed scheme for posture detection. Figure 12 shows the figure-skating posture-detection result using Chamfer matching. The template posture is the same as that of Figure 10c. As this result shows, Chamfer matching doesn't work well when there's strong clutter or large posture deformation.

Finding activities in videos

We conducted further experiments to search for a specific action in video using time–space matching.[2] An action is defined by a sequence of body postures. In these test videos, a specific action appears only a few times. The template sequence is swept along the time axis with a step of one frame, and for each instance we match video frames with the templates.

Figures 13 and 14 (next page) 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 that are similar to human limbs. Our scheme finds both kneeling actions in the short lists' top two, and all 11 waving hand actions in the top 13.

Figure 15 shows the search results for a "throwing" action in a 1,500-frame baseball sequence. Our method merges closely connected matching frames and finds all three of the action's appearances at the top of the list. False detection in our experiments was mainly because of similar structures in the background near the subject. Very strong clutter can also cause the matching scheme

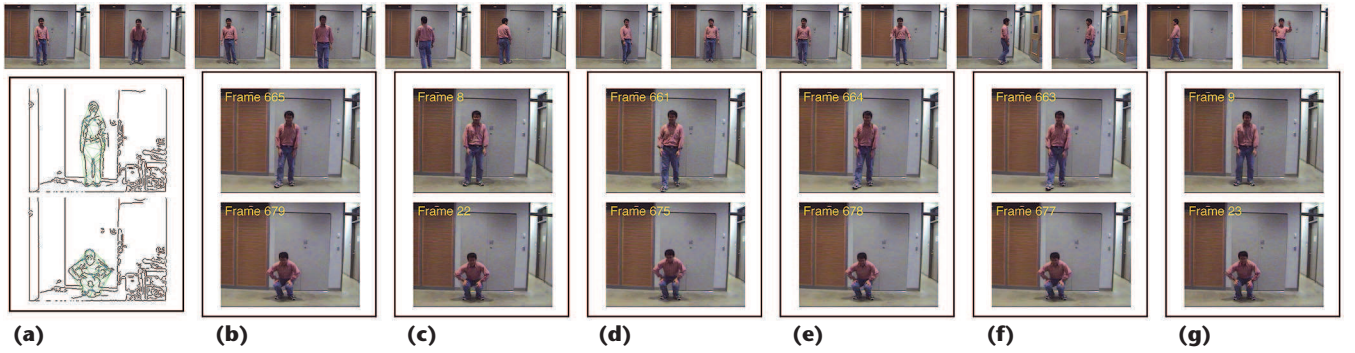


Figure 13. Searching "kneeling" in an 800-frame indoor sequence: (a) templates and (b through g) the top six matches.

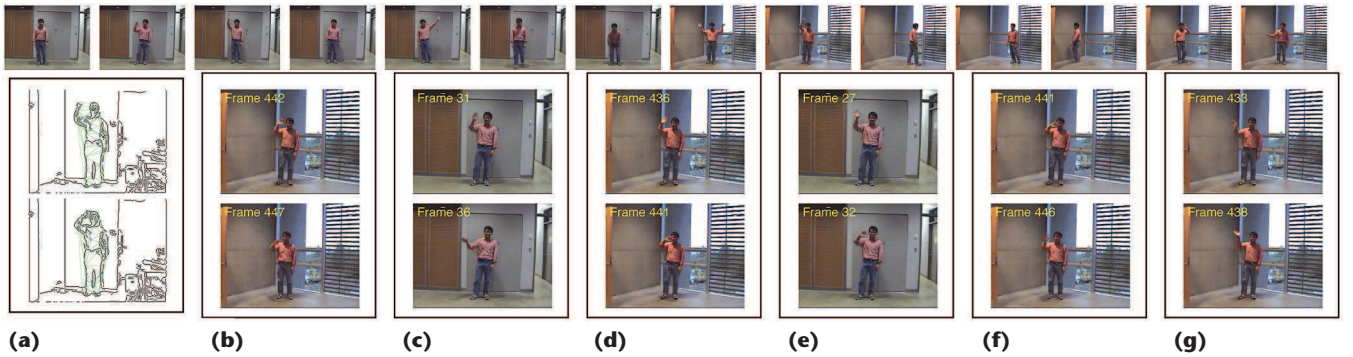


Figure 14. Searching "right-hand waving" in a 500-frame indoor sequence: (a) templates and (b through g) top six matches.

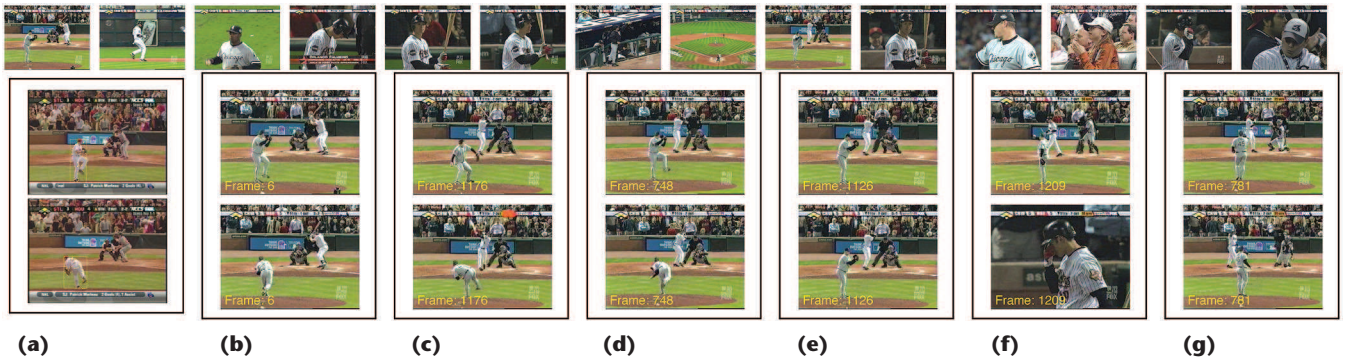


Figure 15. Searching "throwing ball" in a 1,500-frame baseball sequence: (a) templates and (b through g) top six matches.

to fail. Prefiltering or segmentation operations to partially remove the background clutter can further increase detection's robustness.

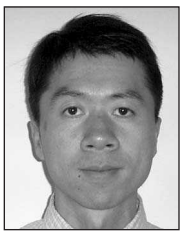
Conclusion

Our posture-recognition method is more efficient and effective than previous methods involving a large target point set. By prefiltering video, we can partially eliminate confounding features from the target image and further improve efficiency, making it possible to conduct real-time matching. We could directly apply our scheme to general object-recognition problems. We're working on extensions to the proposed scheme and building indexing tools for automatic digital sports video retrieval. **MM**

References

1. H. Jiang, Z.N. Li, and M.S. Drew, "Optimizing Motion Estimation with Linear Programming and Detail-Preserving Variational Method," *Proc. IEEE Conf. Computer Vision and Pattern Recognition* (CVPR 04), IEEE CS Press, 2004, pp. 738-745.
2. H. Jiang, M.S. Drew, and Z.N. Li, "Successive Convex Matching for Action Detection," *Proc. IEEE Conf. Computer Vision and Pattern Recognition* (CVPR 06), IEEE CS Press, 2006, pp. 1646-1653.
3. A. Rosenfeld, R.A. Hummel, and S.W. Zucker,

"Scene Labeling by Relaxation Operations," *IEEE Trans. Systems, Man, and Cybernetics*, vol. 6, no. 6, 1976, pp. 420-433.
4. J. Besag, "On the Statistical Analysis of Dirty Pictures," *J. Royal Statistical Soc.*, series B, vol. 48, no. 3, 1986, pp. 259-302.
5. Y. Weiss and W.T. Freeman, "On the Optimality of Solutions of the Max-Product Belief Propagation Algorithm in Arbitrary Graphs," *IEEE Trans. Information Theory*, vol. 47, no. 2, 2001, pp. 723-735.
6. Y. Boykov, O. Veksler, and R. Zabih, "Fast Approximate Energy Minimization via Graph Cuts," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 23, no. 11, 2001, pp. 1222-1239.
7. J. Kleinberg and E. Tardos, "Approximation Algorithms for Classification Problems with Pairwise Relationships: Metric Labeling and Markov Random Fields," *Proc. 40th Ann. IEEE Symp. Foundations of Computer Science* (FOCS 99), IEEE Press, 1999, pp. 14-23.
8. A.C. Berg, T.L. Berg, and J. Malik, "Shape Matching and Object Recognition Using Low Distortion Correspondence," *Proc. IEEE Conf. Computer Vision and Pattern Recognition* (CVPR 05), IEEE CS Press, 2005, pp. 26-33.
9. D.M. Gavrila and V. Philomin, "Real-Time Object Detection for Smart Vehicles," *Proc. Int'l Conf. Computer Vision* (ICCV 99), IEEE CS Press, 1999, pp. 87-93.
10. S. Belongie, J. Malik, and J. Puzicha, "Shape Matching and Object Recognition Using Shape Contexts," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, no. 4, 2002, pp. 509-522.



Hao Jiang is a postdoctoral research fellow in the Department of Electrical and Computer Engineering at the University of British Columbia. His research interests include computer vision, image and video processing, artificial intelligence, and computer graphics. Jiang has a PhD in computer science from Simon Fraser University.

Ze-Nian Li is a professor and director of the Vision and Media Lab in Simon Fraser University's School of Computing Science. His current research interests include computer vision, pattern recognition, multimedia, image processing, and artificial intelligence. Li has MS and PhD degrees in computer science from the University of Wisconsin-Madison.

Mark S. Drew is an associate professor in the School of Computing Science at Simon Fraser University. His research interests include multimedia, computer vision, image processing, color, photorealistic computer graphics, and visualization. Drew has a PhD in physics from the University of British Columbia. He holds a US patent in digital color processing and a US patent application in color computer vision.

For further information on this or any other computing topic, please visit our Digital Library at http://computer.org/publications/dlib.