Human Action Recognition with Extremities as Semantic Posture Representation

Elden Yu and J.K. Aggarwal
Computer and Vision Research Center
The University of Texas at Austin, Austin TX 78712
elden.yu@gmail.com, aggarwaljk@mail.utexas.edu

Abstract

In this paper, we present an approach for human action recognition with extremities as a compact semantic posture representation. First, we develop a variable star skeleton representation (VSS) in order to accurately find human extremities from contours. Earlier, Fujiyoshi and Lipton [7] proposed an image skeletonization technique with the center of mass as a single star for rapid motion analysis. Yu and Aggarwal [18] used the highest contour point as the second star in their application for fence climbing detection. We implement VSS and earlier algorithms from [7, 18], and compare their performance over a set of 1000 frames from 50 sequences of persons climbing fences to analyze the characteristic of each representation. Our results show that VSS performs the best. Second, we build feature vectors out of detected extremities for Hidden Markov Model (HMM) based human action recognition. On the data set of human climbing fences, we achieved excellent classification accuracy. On the publicly available Blank et al. *[3] data set, our approach showed that using only extremities is sufficient to obtain comparable classification accuracy against other state-of-the-art performance. The advantage of our approach lies in the less time complexity with comparable classification accuracy.*

1. Introduction

Semantic knowledge plays an important role in human visual perception of motion and actions. For example, people walk on a flat surface, while the same action is called climbing if it's on a mountain or stairwell; slow running is called jogging in the context of exercising; something moving faster than a speeding car is unlikely a human being; a person on top of a fence cannot walk, etc. Computer vision researchers work on different levels of semantic knowledge applications explicitly or implicitly. Context information may be used to limit possible choices on models or parameters [18], while a semantic representation can be useful for complex activity recognitions [13]. In this paper, we regard human extremities as a compact semantic representation of human posture, and show that it is a powerful tool to achieve surprisingly accurate action recognition rates with less time complexity than many state-of-the-art algorithms.

The task of finding human extremities is challenging due to the articulated human body, self occlusion, ambiguity from the absence of depth information, appearance variations caused by camera viewpoints, illumination and loose clothing. In this paper, we concentrate on detecting human extremities including head, hands and feet, from a body contour with star skeleton representations. We apply the detected extremities to human action recognition, and show that simple extremities are sufficient to recognize most common actions. We do not focus on tracking and obtaining human contours in this paper. For videos taken with fixed cameras, one can usually get human silhouettes with background subtraction followed by morphological operations, and then contours by additional border following algorithms. For videos of mobile cameras, there are contour tracking algorithms as surveyed by Yilmaz *et al*. [17].

Fujiyoshi and Lipton [7] proposed a star skeleton representation (SS) to analyze human motion. The center of mass of a human silhouette is extracted as a single star. Distances from all contour points to the star are computed as a function of indices of clockwise sorted contour points. Local peaks in the function correspond to extreme contour points, which are approximations of human extremities. Their initial goal is to use such a representation for feature extraction to recognize cyclic human motions such as walking and running. Although SS is simple and fast for computation, its accuracy for detecting human head and limbs needs further improvement.

In order to accurately detect extremities for classifying fence climbing actions, Yu and Aggarwal [18] proposed a two-star skeleton representation (2SS) by adding the highest contour point as the second star. Two sets of local peaks are estimated, and their indices are simply paired up and averaged to find more precise extreme points. However, Yu and Aggarwal [18] did not explain why two stars should perform better than the single star.

978-1-4244-3993-5/09/$25.00 ©2009 IEEE

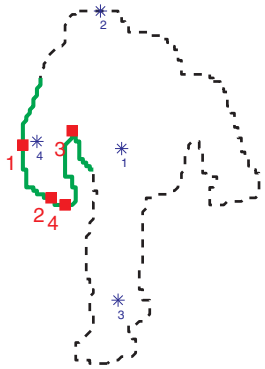

Figure 1. The four stars are shown with blue asterisks and their respective detections of left hand shown with solid red squares with corresponding numbers.

To utilize structure information available in the star skeletons, Chen *et al*. [5] defined a distance function between two SS. Each SS is converted to a vector of five extremities. If less than five are detected, they pad zero in the vector. If more are detected, they increase noise smoothing to remove extra. The distance function is defined as the sum of Euclidean distances between five matched pairs of skeletons. They utilize such a distance function in their HMM-based action recognition system.

To the best of our knowledge, no work has been devoted to analyzing how many stars one needs and where the stars should be placed to produce extreme contour points as the best approximation to human extremities. Our contribution in this paper is to introduce the notion of visibility and star polygon (more details later in Section 2.2) from computational geometry to analyze shortcomings of SS and 2SS for detecting human extremities. In addition, motivated by the above concepts, we develop a variable star skeleton representation (VSS) to improve the accuracy of finding extremities. We construct a medial axis of a binary human body contour, and treat all junction points of the medial axis as stars. For each star, we produce a set of candidate extreme points, from local peaks of distances between the star and sorted contour points. Each candidate is either kept, discarded, or merged with nearby candidates, depending on cues including robustness to the smoothing parameter, visibility to the associated star, and proximity to other candidates. We implement SS, 2SS, and VSS, and compare their performance over a set of 1000 frames from 50 sequences of persons climbing fences. Results show that VSS performs the best. Moreover, we analyze and present typical situations when VSS performs better than the other two. In order to show that the detected extremities are sufficient for action recognition, we propose a simple histogram to build feature vectors from extremities and employ HMM for classification. The results presented in this paper on both the fence climbing data set [18] and the popular Blank *et al*. [3] data set are excellent. Our approach is very simple thus with less time complexity and still achieves comparable accuracy with other state-of-the-art methodologies.

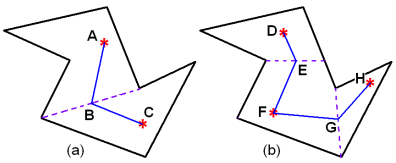

Figure 2. Two different decomposition on the same simple polygon.

2. On the number and position of stars

In this section, we first illustrate our motivation for analyzing the number and position of stars with an example in Section 2.1. Next we connect our observation with the visibility and star polygon concepts in Section 2.2. Then we analyze both the single and two star skeleton representations with the concepts in Section 2.3.

2.1. Observation

In a SS representation, the position of the star greatly effects, if not determines, whether a contour point could be a local peak in the contour neighborhood hence be a possible human extremity.

An example is given in Figure 1 showing a climbing person. We focus on detecting the left hand here. The part of the contour around the left hand is highlighted with a green solid line, while the other parts are shown with a black dash line. For illustration purposes, we choose four stars as shown with blue asterisks and numbered. The detected hand from each star is shown with a solid red square and numbered accordingly in Figure 1. From this example, we can see that the fourth star provided the best approximation, the second star made a close one, and the other two produced incorrect extremities.

2.2. Visibility and the star polygon

Before we proceed, we briefly review some concepts from the computer graphics community to make the paper self-contained.

Given a human contour represented by a set of clockwise sorted contour points, we treat it as a simple polygon P. In geometry [14], a *simple polygon* is a polygon whose sides do not intersect unless they share a vertex. A point in the polygon (including interior and boundary) is *visible* with respect to another point in the polygon if their line segment falls completely within the polygon. For example, in Figure 2, point D is visible to point E, F, G while not visible to H.

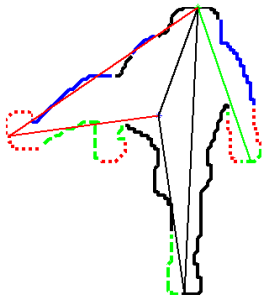

Figure 3. The two stars are in blue and green respectively. Contour points visible only to the center of mass are shown with a blue solid line, visible only to the highest contour point shown with a green dash line, visible to both shown with a black solid line, visible to neither shown with a red dotted line. Best viewed in color.

According to Shapira and Rappoport [14], if there exists a point $v \in P$ that is visible from any other point inside P, then P is a *star polygon* and v is a star point. Since not every polygon is a star polygon, they further defined the star skeleton to decompose a simple polygon as a star set and the associated skeleton. Simply speaking, the star set is a set of star polygons such that each shares at least one edge with another star polygon; the skeleton is a tree that connects star points and mid points of the shared edges.

Shown in Figure 2 is the same simple polygon decomposed into two star polygons in (a) and into three in (b). In Figure 2(a), points A, C are star points and the connection ABC is the skeleton. In Figure 2(b), points D, F, H are star points and the connection $DEFGH$ is the skeleton. Obviously the star-polygon decomposition is not unique.

2.3. Characteristics of star skeleton representations

Why do different stars produce different approximations of the left hand in Figure 1? There are many possible explanations such as distance, scale, and visibility. Among all the factors, we regard visibility as the most important one. The reason that the second and fourth star perform better is because the left hand is visible to them, while not visible to the other two.

Fujiyoshi and Lipton [7] considered the human centroid as a single star. As human contours are usually not star polygons, a single star cannot be visible to all contour points. Hence SS will easily miss true human limbs or produce false alarms. In extreme conditions, the centroid may not even be inside the human silhouette.

Yu and Aggarwal [18] added the highest contour point as the second star. It can be interpreted as an intention to make all those points not visible to the center of mass visible to the second star. This way, they hope most contour points will be visible to at least one of the two stars. This strategy is intuitive and reasonable; however, its practical effect is weakened in two aspects. First, it is a problem whether or not to treat the highest contour point as an extremity. In most human postures, the highest contour point is the head, hence it is desirable to include the second star as one of the detected extremities. When the assumption is violated, the inclusion might produce false alarms, as shown in Figure 9(f). Second, they make two detected limbs (each from a different star) a pair and average them, which means a good detected extremity is compromised by a bad one. We would rather have the algorithm select the good ones and discard the bad ones.

Using a frame of a person climbing fence, we show in Figure 3 the visibility of each contour point with respect to the center of the mass and the highest contour point. Details of computing such visibility is described later in Section 3.3. It is obvious that with only the center of mass as the single star, a considerable portion of the contour is not visible. With the addition of the second star, more contour pieces are covered, while there is still a significant portion not visible.

Note that in both works [7, 18], the so-called stars are just approximations of star points as defined in Section 2.2. Considering human contours as simple polygons, our ultimate desire is to choose an appropriate number of "stars" and their positions so that many contour points are visible to at least one "star", e.g. making the approximation as good as possible.

3. A variable star skeleton representation

In this section, we develop a variable star skeleton (VSS) representation, motivated by observing that more and well positioned stars make contour points more visible. Although built upon previous works [7, 18], our new representation is considerably different in two aspects, including finding stars and producing extremities out of multiple sets of candidates. We take as stars, junction points in the medial axis of the human silhouette, which may be regarded as a rough approximation of human body joints. Each star will produce a set of extreme points, as previously done in SS and 2SS. As a candidate, each extreme point will be processed according to its robustness to noise smoothing, visibility to the generating star, and proximity to its neighbors.

3.1. Detecting junctions of a medial axis

For contours, a medial axis is the union of all centers of inset circles that are tangent to at least two contour points. In order to compute the medial axis, we choose the augmented Fast Marching Method by Telea and Wijk [15] among many existing algorithms such as [4, 10]. There is a threshold t controlling how short each branch of the medial axis may be. Shown in Figure 4 is the computed medial axis in magenta dotted line with $t = 10$ and $t = 30$ respectively.

In order to find the junction points, we employ a lookup

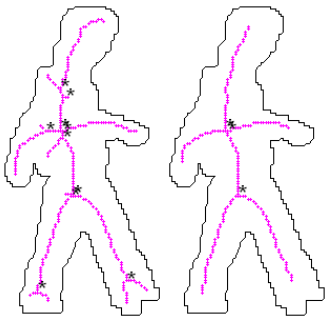

Figure 4. The left image shows in magenta line the medial axis obtained with $t = 10$, while the right one with $t = 30$. Each detected junction point is annotated with a black asterisk.

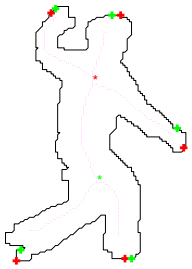

Figure 5. The medial axis is shown with a magenta line, junctions as asterisks, and candidate extremities as crosses in the same color as the corresponding star.

table (LUT) in the 3 by 3 neighborhood of every pixel on the medial axis. As each cell in the neighborhood take binary values, we have 256 total possible combinations of the 8 connected neighbors. For each combination we determine if the center pixel is a junction point, as denoted by a black asterisk in Figure 4. One may notice in the figure that sometimes two junctions are too close; in such cases, we merge those junctions that are closer than a threshold (w) and use their mean as the estimated junction. In rare cases, the parameter t is too strict to produce any junction from the medial axis; we opt to use the center of the mass as the single star, although we can also choose to reduce t until we get at least one junction point.

3.2. Generating candidate extreme points

Suppose there are N stars denoted as $star_j$ ($j = 1, 2...N$). Starting with the highest contour point, each point in the contour of length NC is sorted clockwise, and denoted as P_i ($i = 1, 2...NC$). As in previous works [7, 18], we compute the Euclidean distance from $star_j$ to P_i as a function $dist_j(i)$. The function is then smoothed by a one-dimensional Gaussian kernel with standard deviation δ. Contour points with a local peak are chosen as candidate extreme points.

3.3. Filtering

In this section, we determine if a candidate extreme point is kept, discarded or merged with a nearby candidate. We first associate each candidate with two properties, including robustness to the smoothing parameter and visibility to the generating star.

The robustness R may be viewed as a measurement of how much a possible human limb protrudes out of the torso. As we have located all the local peaks from the distance function $dist_j$ described above, we can easily modify it to locate all the local valleys as well. Given a local peak with value $dist_j(Ind_K)$ at position Ind_K, it must have an adjacent valley both on the left and on the right. Suppose the higher adjacent valley has value $dist_j(Ind_{K'})$ at position $Ind_{K'}$, we define robustness R associated with the candidate extreme point P_{Ind_K} in the following equation.

$$R = \frac{dist_j(Ind_K) - dist_j(Ind_{K'})}{|Ind_K - Ind_{K'}|} \quad (1)$$

We connect from the candidate to the star generating it, to form a line segment. The visibility V is computed as a proportion of the line segment that lies inside a silhouette. Given two points, we use the basic raster algorithm [11] on line drawing to produce the set of points between them. Then the intersection of the set with the binary human silhouette produces line points inside the silhouette.

With these properties, we proceed with the following procedure where the input is all those candidates as generated in Section 3.2, and the output is the detected extremities.

1. **Select candidates chosen by more than one star**. We group all those candidates by hierarchical agglomerative clustering with single linkage, so that any two candidates whose indices are closer than w are put into one group. The means of all those clusters with more than one members form set A, and all the single member clusters form set B.

2. **Select candidates with better visibility and robustness.** Select from B all those candidates with R bigger than threshold $MaxR$ and V bigger than $MaxV$ into set A.

3. **Discard bad candidates** from B with R smaller than threshold $MinR$ or V smaller than $MinV$.

4. **Make at most 5 extremities.** We denote the number of elements of A as $|A|$. If $|A| > 5$, sort A by product of R and V, stop and output the top 5 only. If $|A| <= 5$, sort B by product of R and V. Select the top $min|B|, 5 - |A|$ candidates from B into set A, stop and output A.

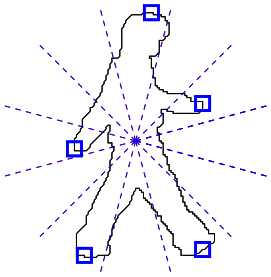

Figure 6. A simple histogram to extract feature vectors from frames.

4. Histogram of extremities as posture representation

With detected body extremities, we recognize a variety of common human actions by the discrete HMM technique [12]. As each action is represented by an image sequence or video, the key procedure is to convert each frame to an observation symbol so that each action may be represented by an observation sequence. Note that we are using only a set of human extremities for each frame. Motivated by the shape context descriptor proposed by Belongie *et al*. [2], we use a simple histogram to build a feature vector for each frame. As shown in Figure 6, we find the relative coordinates of each extremity with respect to the center of mass of the human silhouette. The entire plane is evenly divided into N ($N = 12$) sectors, and the histogram is a N-element vector with each element indicating if there is an extremity in the sector.

In order to reduce the number of observation symbols, vector quantization is commonly employed to cluster the feature vectors. The cluster label of each feature vector acts as the observation symbol for HMM usage. However, it is not always necessary if there are a limited number of unique features. In our cases, we simply use the index of each feature vector in the unique feature vector set as the observation symbol.

5. Experiments

We used three data sets from two sources in our experiments. The first data set is built from 50 sequences of persons climbing fences provided by Yu and Aggarwal [18]. Shown in Figure 7 are sample frames of a sequence. We collect 20 frames evenly distributed from each sequence to form a data set of 1000 frames. It is to test if our proposed VSS performs better than previous methods including SS and 2SS. As each sequence consists of one *climbing* action and at least one *walking* action (before or after the *climbing* or both), we manually divide it so that each resulting sequence is either *climbing* or *walking*. We got 50 *climbing* and 90 *walking* to form the second data set for action recognition. The third data set is the publicly available human action recognition data set contributed by Blank *et al*. [3].

Figure 7. Sample climbing frames from a sequence.

5.1. A comparison of VSS against SS and 2SS

For each frame in the first data set, we have all the three star skeleton representations performing detection of extreme points as approximation of head and human limbs. We manually check the results and determine the number of ground truth extreme points, true positives and false alarms. To empirically validate the relative importance of visibility and robustness criteria for human extremities, we also did experiments on the first data set without the visibility or robustness criteria. Comparisons are shown in Table 1.

5.2. Parameter selection

There are several parameters involved in all the three star skeleton representations. The common parameter among the three is the Gaussian smoothing factor δ. There is a trade off between detecting more global or more local extreme points when selecting different scales of smoothing parameter. We used $\delta = 10$. The t threshold is set as 30, which yields reasonable medial axis for most binary blobs. We usually get one, two or three junctions from a medial axis. We set $w = 10$ for both merging junctions and clustering candidates. The two thresholds for R in filtering process are set as 0.6, 0.1, and the two thresholds for V are set as 0.9, 0.5. All the parameter values are chosen empirically and used throughout the experiments.

5.3. When is VSS better?

When there is no junction point detected, VSS is reduced to the single star skeleton, except that we have the filtering process. Fortunately this does not occur frequently due to proper selection of t (the minimum length of a branch). When there is only one junction point, VSS is different from SS in that the single star has a different position. An example is shown in Figure 9(a), where VSS can successfully detect the two hands while both SS and 2SS fail.

The 2SS improves from the single SS by fixing the highest contour point as the second star. As displayed in the left image of Figure 9(c), the head is detected as the highest contour point. This implicit assumption of the highest

	SS	2SS	VSS w/o robustness	VSS w/o visibility	VSS
Ground truth	3691	3691	3691	3691	3691
True positive	3107/84.2%	3381/91.6%	3617/98.0%	3580/97.0%	3440/93.2%
False alarm	779/21.1%	146/4.0%	705/19.1%	384/10.4%	98/2.7%

Table 1. Results from the three representations on the first data set.

contour point being the head does not always hold. When the assumption does not hold, VSS easily wins over 2SS. Figure 9(b,e) shows the hand is higher than the head, and Figure 9(f) shows the back is higher than the head.

6. Action recognition

We test action recognition on two different data sets with the same strategy. For each data set, we build a feature vector from each frame with our simple histogram as described in Section 4. The procedure itself can be viewed as vector quantization as well, since we have much less unique feature vectors. We adopted the leaving-one-out cross-validation strategy in our HMM classification framework. In each iteration, we just pick one test sequence in turn, and use all the rest as a training set to train for each class a HMM with 2 hidden states. Finally each sequence is used exactly once as a test sequence, and the confusion matrix is produced.

6.1. On the fence climbing data set

There are a total of 12652 frames in the 50 *climbing* and 90 *walking* sequences. After feature extraction there are 685 unique feature vectors. After 140 iterations, the confusion matrix produced only 3 misclassifications, e.g. the overall accuracy is 97.9%. We found all three misclassifications are due to their very short durations, including 16, 12, and 19 frames. Since the frame rate is 30 (fps), these short sequences do not even show a full step, as validated by manual inspection.

As a baseline comparison, Yu and Aggarwal [18] reported 3 misclassifications on 18 walking and 10 climbing test sequences, which is approximately 5 times our error rate. They used 2SS to find extremities and built features such as how many extremities are above or under the fence, in addition to motion features including the direction of the centroid velocity. In comparison, our approach involves no explicit motion features.

6.2. Comparison on the Blank *et al*. [3] data set

In this data set, there are 93 sequences of 9 persons performing 10 different actions. Using the provided human silhouette, we extracted the human extremities for all 5687 frames. To get a flavor of the accuracy of the proposed VSS on this particular data set, we manually checked all 701 frames from 10 sequences performed by one person (*Daria*). Our VSS detected 1889 (96.1%) out of 1966 ground truth extremities, while making only 18 false alarms. There are only 179 unique feature vectors. After 93 iterations, the confusion matrix is produced as in Figure 8. There are 2 misclassifications between *jump* and *pjump*, as they are essentially the same action taken in different views.

	bend	jack	jump	pjump	run	side	skip	walk	wave1	wave2
bend	9									
jack		9								
jump			7	1			1			
pjump			1	8						
run					10					
side						9				
skip					1		9			
walk						1		9		
wave1									8	1
wave2										9

Figure 8. The confusion matrix of action recognition on the Blank *et al*. [3] data set.

The overall accuracy is 93.6%, as compared in Table 2. Although we didn't achieve the perfect recognition rate, our methodology is faster in the sense that our VSS to detect extremities is linear in the number of contour points; and our feature extraction procedure is very simple.

Note that the first four papers [3, 1, 9, 8] worked on the old version of the data set without the *skip* action. In Blank *et al*. [3], each sequence is further split into cubes and classification is done per cube. Their algorithm has linear time complexity in the number of space-time points, *e.g*. the total number of pixels inside all silhouettes. Ali *et al*. [1] assume the six body joints including head, belly, hands and feet are available for further action recognition. In their experiment, they used the end points of medial axis as approximation of body joints, which is very close to our idea of using junctions as stars. Both Niebles and Fei-fei [9] and Jhuang *et al*. [8] have the advantage of avoiding the difficult segmentation step, but their time complexity is at least linear in the number of all pixels in a video. In Fathi and Mori [6], a computation of the optical flow is necessary for each frame as the first step. Considering they have tracked the human figure as a rectangle, the time complexity is linear in the number of all pixels in the tracked region. In Wang and Suter [16], the module of dimension reduction by Locality Preserving Projection (LLP) has square time com-

Method	Accuracy	Data	Comments
Blank *et al*. [3]	99%	81 seq. no *skip*, chopped as cubes	linear in the number of all space-time points
Ali *et al*. [1]	92.6%	81 seq. no *skip*	assuming six body joints available
Niebles and Fei-fei [9]	72.8%	83 seq. no *skip*	linear in the number of all pixels
Jhuang *et al*. [8]	98.8%	81 seq. no *skip*	linear in the number of all pixels
Fathi and Mori [6]	100%	93 seq. 10 actions	linear in the number of pixels in tracked region
Wang and Suter [16]	100%	93 seq. 10 actions	LLP module square in the number of frames
Ours	93.6%	93 seq. 10 actions	linear in the number of all contour points

Table 2. Comparison of different methods on the Blank *et al*. [3] data set.

plexity in the number of frames, as the construction of the adjacency matrix need to find K nearest neighbors for each frame.

7. Conclusions

In this paper, we analyze how the number and positions of stars may effect the detected extremities as approximations of human head and limbs. We discuss the shortcomings of SS and 2SS, and develop VSS to overcome disadvantages of SS and 2SS. Our experiments include performance of the three representations over a data set of 1000 frames, and comparison and analysis show VSS is better than the other two. Moreover, we build features out of a spatial histogram of detected extremities and presented HMM classification results on two human action data sets, to prove that detected human extremities are simple but strong enough for action recognition. Our classification accuracy is better than the previous approach on the fence climbing data set, and comparable to other state-of-the-art algorithms on the Blank *et al*. [3] data set. The less time complexity of our algorithm is needed for systems with real time performance requirements.

8. Acknowledgement

This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. HR0011-08-C-0135. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of DARPA.

References

[1] S. Ali, A. Basharat, and M. Shah. Chaotic invariants for human action recognition. In *ICCV*, Rio de Janeiro, 2007.

[2] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Trans. PAMI.*, 24(4):509–522, 2002.

[3] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. In *ICCV*, pages 1395–1402, Beiging, 2005.

[4] H. Blum and R. N. Nagel. Shape description using weighted symmetric axis features. *Pattern Recognition*, 10(3):167–180, 1978.

[5] H.-S. Chen, H.-T. Chen, Y.-W. Chen, and S.-Y. Lee. Human action recognition using star skeleton. In *International Workshop on Visual Surveillance & Sensor Networks*, pages 171–178, Santa Barbara, 2006.

[6] A. Fathi and G. Mori. Action recognition by learning mid-level motion features. In *CVPR*, Anchorage, 2008.

[7] H. Fujiyoshi and A. Lipton. Real-time human motion analysis by image skeletonization. In *IEEE Workshop on Applications of Computer Vision*, pages 15–21, Princeton, 1998.

[8] H. Jhuang, T. Serre, L. Wolf, and T. Poggio. A biologically inspired system for action recognition. In *ICCV*, Rio de Janeiro, 2007.

[9] J. C. Niebles and L. Fei-Fei. A hierarchical model of shape and appearance for human action classification. In *CVPR*, Minneapolis, 2007.

[10] R. L. Ogniewicz and O. Kubler. Hierarchic voronoi skeletons. *Pattern Recognition*, (28):343–359, 1995.

[11] J. O'Rourke. *Computational Geometry in C*. Cambridge University Press, 1994.

[12] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2), 1989.

[13] M. S. Ryoo and J. K. Aggarwal. Hierarchical recognition of human activities interacting with objects. In *The 2nd International Workshop on Semantic Learning Applications in Multimedia*, Minneapolis, 2007.

[14] M. Shapira and A. Rappoport. Shape blending using the star-skeleton representation. *IEEE Computer Graphics and Applications*, 15(2):44–50, 1995.

[15] A. Telea and J. J. van Wijk. An augmented fast marching method for computing skeletons and centerlines. In *ACM Symposium on Data Visualisation*, pages 251–ff, Aire-la-Ville, Switzerland, 2002. Eurographics Association.

[16] L. Wang and D. Suter. Analyzing human movements from silhouettes using manifold learning. In *IEEE International Conference on Advanced Video and Signal Based Surveillance*, Sydney, 2006.

[17] A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey. *ACM Comput. Surv.*, 38(4):13, 2006.

[18] E. Yu and J. K. Aggarwal. Detection of fence climbing from monocular video. In *International Conference on Pattern Recognition*, pages 375–378, Hong Kong, 2006.

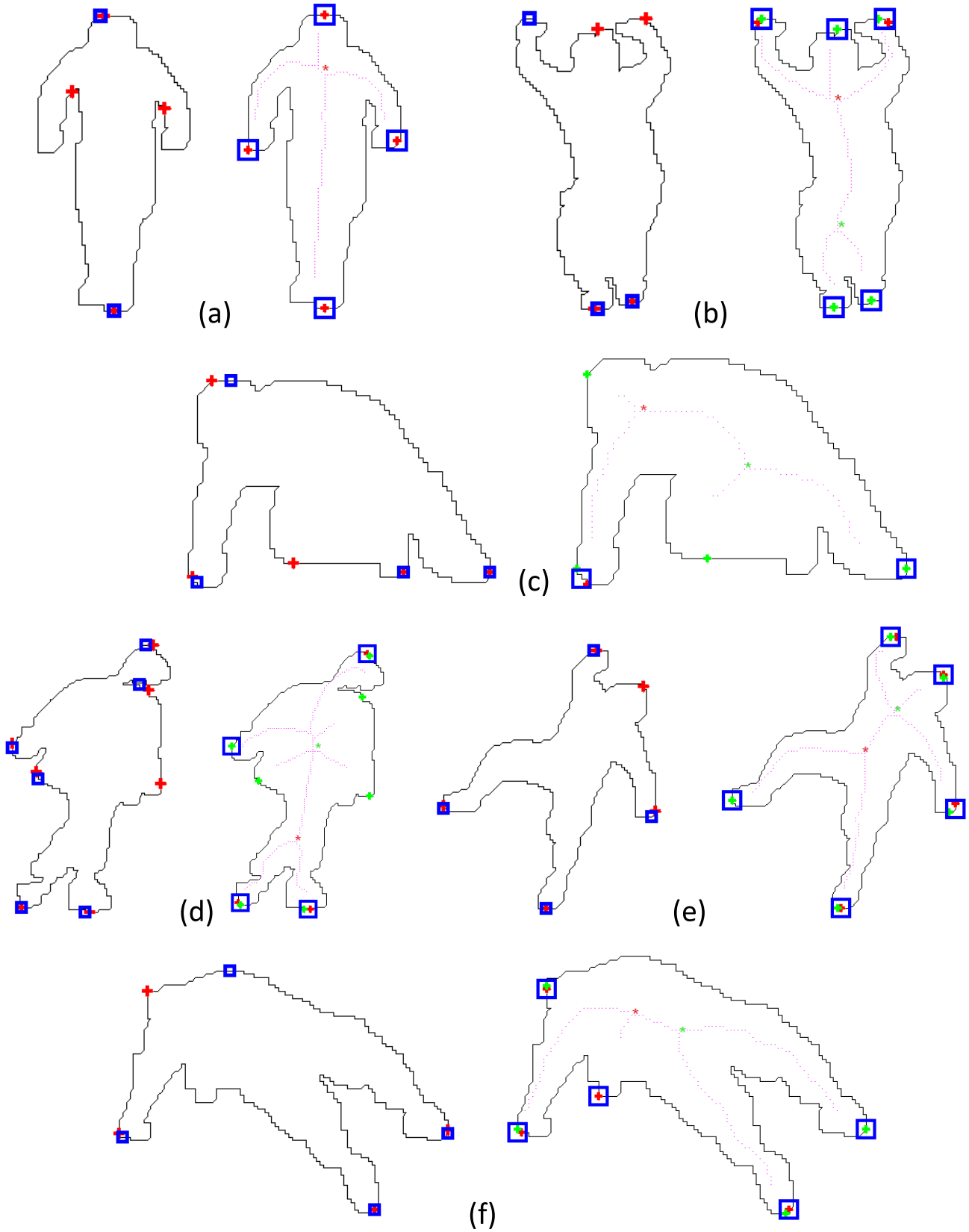

Figure 9. For each pair of images, the left image shows the result of SS in red crosses, the result of 2SS in blue squares; the right image shows the result of VSS in blue squares. In the right images, stars are shown in colors and their associated extremity candidates are shown in the same color crosses.