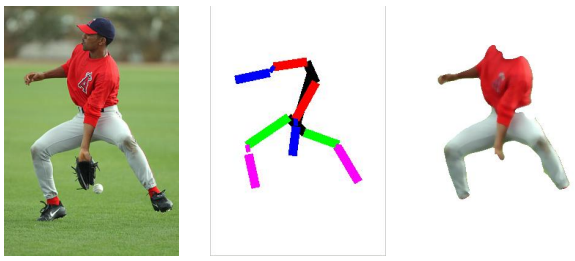
Recovering Human Body Configurations: Combining Segmentation and Recognition

Greg Mori, Xiaofeng Ren, Alexei A. Efros[†] and Jitendra Malik
Computer Science Division
UC Berkeley
Berkeley, CA 94720, USA

[†]Robotics Research Group
University of Oxford
Oxford OX1 3PJ, U.K.



(a) (b) (c)

Figure 1: The problem: (a) Input image. (b) Extracted skeleton of localized joints and limbs. (c) Segmentation mask associated with human figure.

Abstract

The goal of this work is to take an image such as the one in Figure 1(a), detect a human figure, and localize his joints and limbs (b) along with their associated pixel masks (c). In this work we attempt to tackle this problem in a general setting. The dataset we use is a collection of sports news photographs of baseball players, varying dramatically in pose and clothing. The approach that we take is to use segmentation to guide our recognition algorithm to salient bits of the image. We use this segmentation approach to build limb and torso detectors, the outputs of which are assembled into human figures. We present quantitative results on torso localization, in addition to shortlisted full body configurations.

1. Introduction

The goal of this work is to take an image such as the one in Figure 1(a), detect a human figure, and localize his joints and limbs (b) along with their associated pixel masks (c). This problem is arguably the most difficult recognition problem in computer vision. Difficulties arising from appearance variation due to clothing are compounded by articulated deformation. Consider the people in Figure 2. Their poses are not *probable*, yet are definitely *possible*. Moreover, the appearance of their limbs varies rather dramatically between the different people. The ability to accurately find people in images such as these would facilitate many useful applications such as initializing 3D kinematic trackers, understanding human actions, and re-rendering for graphics.

The difficulties described above have led researchers to simplify the problem, often using datasets of unclothed people, or those processed with background subtraction from video sequences. The range of variation in pose is usually limited, and there is little or no background clutter.

In this work we attempt to tackle this problem in a more general setting. The dataset we use is a collection of sports news photographs of baseball players collected from the Internet. The images selected are full body pictures of a single player. These players are in a wide variety of poses, are wearing different clothes that are often textured, and are photographed outdoors under varying lighting conditions, in natural, cluttered scenes.

The structure of this paper is as follows. In Section 2 we discuss related previous work. In section 3 we describe our approach at a high level. Section 4 and 5 provides the details of our algorithm. We discuss the positive aspects and limitations of our method in section 6.

2. Related Work

Some of the earliest research related to this problem is the pedestrian tracking work of Hogg [3]. A vast quantity of work continued in this vein, using high degree-of-freedom 3D models of people, rendering them in the image plane, and comparing them with image data. Gavrila [2] provides a survey of this work. These approaches typically require a hand-initialized first frame, and the large number of parameters in their models lead to difficult tracking problems in high dimensional spaces. More recent developments in pedestrian detection, such as Mohan et al. [7] and Viola et al. [18], are fairly successful in detecting people in common standing poses. However, these template-based window-scanning approaches do not localize joint positions, and it is not clear whether they generalize to finding people in arbitrary poses.

Figure 2: The challenge of unlikely pose

The complexities in 3D model-based tracking have led researchers to pose the problem as one of matching to stored 2D exemplars. Toyama and Blake [16] used exemplars for tracking people as 2D edge maps. Mori and Malik [8], and Sullivan and Carlsson [14] directly address the problem we are interested in. They stored sets of 2D exemplars upon which joint locations have been marked. Joint locations are transferred to novel images using shape matching. The problem with such exemplar based approaches is the combinatorial explosion in the number of exemplars needed to match a wide variety of poses. Shakhnarovich et al. [12] attempt to address variation in pose and appearance through brute force, using a variation of locality sensitive hashing for speed to match upper body configurations of standing, front facing people. It is not clear whether such an approach will scale to handle a variety of poses such as those in Figure 2. One fundamental problem with the exemplar-based methods is that they do not take full advantage of the salient low-level cues.

Another family of approaches model people as an assembly of parts. Typically a simple low-level detector is applied to produce a set of candidate parts. Then a top-down procedure makes inferences about the parts and finds the best assembly. Song et al. [13] detect corner features in video sequences and model their joint statistics using tree-structured models. Felzenswalb and Huttenlocher [1] score rectangles using either a fixed clothing model or silhouettes from background subtraction of video sequences and then quickly find an optimal configuration using the distance transform to perform dynamic programming on the canonical tree model. Ioffe and Forsyth [4] use a simple rectangle detector to find candidates and assemble them by sampling based on kinematic constraints.

The difficulty with the tree-based methods is that there are dependencies among the body parts that cannot be captured by a tree. For example, there are anthropometric constraints (we use data from [9]) on the relative scales of the limbs. Symmetry in appearance between left and right limbs, such as the arm in Figure 2(left) is another cue that can't be captured in a tree. Reasoning about self-occlusion cannot be done either. Finally, in a tree model there is no direct mechanism for preventing the reuse of image pixels. An arm with a good low-level score could be labeled as both the right and left arm.

Recently, there has appeared promising work exploring the interplay between low-level cues and high-level knowledge. The line of research by Torralba et al. [15] develops relationships between local object detectors and the global scene context. Others combine segmentation with recognition, making more sophisticated uses of low-level cues. Yu and Shi [19] add object-specific patch-based templates into the Normalized Cuts framework. Tu et al. [17] combine face and letter detectors with segmentation in the DDMCMC framework. However, most of these approaches are tested on simple rigid objects and it is yet to be shown how these frameworks handle complicated objects such as articulated human bodies in arbitrary poses.

3. Motivation and Approach

One classic approach to recognizing objects such as people would be to model them as a collection of generalized cylinders which one could aim to detect in a bottom-up approach ("ribbon finding"). These strategies were common in the 1980s, but have fallen into disfavor in recent years. The basic problem is that trying to reliably detect each part individually is problematic (if not hopeless) in a real-world setting. Zoom in on an image of a person's arm and you will find that it looks no different than a piece of grass, or a tree trunk. But look at that same arm within its context (a hand, a shoulder, a side of a torso), and it instantly becomes recognizable. That is, most low-level features are informative only when considered within their context, which represents the global information present in the scene. But, of course, the context is made up of low-level features as well, leading to a classic chicken-and-egg problem. How do we crack it?

Let us start with the few low-level features that are, in fact, informative independent of the context. These features usually represent parts that are salient, that stand out, having enough information in themselves. This is akin to viewing a modern painting, say a Picasso canvas from his Cubist period: among a crowd of cubes a few salient shapes jump out – an elbow, a hand, a face. We call them *islands of saliency*. Starting from a set of these islands a viewer is usually able to explain the rest of the picture by making educated guesses based on context ("if that's an elbow and that's a torso, then this line must be an arm"), in a sense, bridging the islands of saliency. We operationalize this idea by using a number of low-level pre-processing stages.

The low-level procedure gives us a collection of salient parts, in our case torsos and "half-limbs" (upper and lower legs or arms). Note that we expect this low-level process to be noisy – many correct parts will not appear salient while some salient parts will turn out to be wrong. All we need is *enough* good, salient parts to jump-start our search. "Partial configurations" consisting of a few half-limbs and a torso are assembled from these salient parts. There is a combinatorial problem to determine which parts can be used to-

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

Figure 3: Image analysis at the low-level: (a) Input image. (b) (c) Canny edges at two different scales. Texture on clothing and background clutter pose severe problems for recognition. (d) pb ("probability of boundary") image [6]. It handles high-frequency texture nicely. (e) A Normalized Cuts segmentation with $k = 40$. Salient limbs pop out as single segments; head and torso consist of several segments. (f) A "superpixel" map with 200 superpixels. It captures all the details.

gether to form a partial configuration. We enforce global constraints on human body configuration, namely relative scales, positions and colors, to prune away impossible combinations. The remaining partial configurations are then extended to full human figures by searching for the missing limbs. Full configurations are sorted using a combination of their individual part scores.

A standard way to make use of low-level information is applying the Canny edge detector. Figure 3(b,c) shows the Canny edges of a sample image. In textured regions, simple edge detectors fire frequently, thus limiting their usefulness. In this work we use the boundary finder of Martin et al. [6] (Figure 3(d)), which combines both brightness and texture information to reduce clutter. Furthermore, we use the Normalized Cuts algorithm [5] to group similar pixels into regions. Figure 3(e) shows a segmentation with 40 regions. Many salient parts of the body pop out as single regions, such as the legs and the lower arms. In addition, we use over-segmentation, as shown in Figure 3(f), consisting of a large number of small regions or "superpixels" [10], which has been shown to retain virtually all structures in real images. These segmentations dramatically reduce the complexity of later stages of analysis, e.g., from $400K$ pixels to 200 superpixels. Figure 4 shows the data flow of our complete algorithm.

We validate our approach in a number of ways. First, we use hand-labeled images to evaluate the classification performance of our half-limb detector, as well as the individual cues being used. Second, we evaluate the performance of our segmentation-based torso detector, again using hand-labeled joint positions, and compare it to an exemplar-based detector. We then show how the salient half-limbs help constrain the torso position. Finally, we present results on recovering full body configurations from the salient parts. The final scoring of full body configurations is a challenging problem of its own and is not dealt with in this work. Instead, we return a shortlist of candidate body configurations which should be evaluated by a separate procedure. An added advantage of our segmentation-based approach is that we have pixel masks assigned to limbs, which would likely facilitate more accurate final scoring mechanisms.

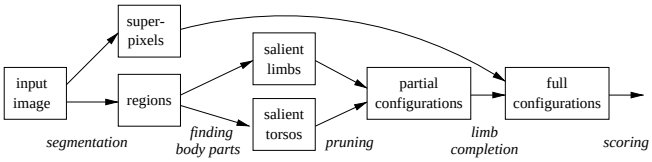


Figure 4: Data flow of the algorithm. First an image is segmented into superpixels and relatively large regions or segments. We detect salient upper and lower limbs from these segments. Simultaneously we detect potential head and torso positions. Both modules return a "shortlist" of top ranked candidates. We then combine these parts into partial body configurations and prune away impossible configurations by enforcing global constraints such as relative scale and symmetry in clothing. In the final stage we complete partial configurations by combinational search in the space of superpixels to recover full body configurations.

4. Finding Body Parts

In this section, we describe our procedure for finding salient half-limbs and torsos. First, image segmentation is used to generate candidate segments. A set of low-level cues is then computed to classify these segments. The procedure is empirically validated using ground truth data.

4.1. Finding Limbs

As we observed in Figure 3(e), salient "half-limbs" often pop out as a single segment. Hence we search among these segments, generated by the Normalized Cuts algorithm with $k = 40$, to find half-limbs. We classify these segments with the set of cues defined below.

4.1.1 Cues for half-limb detection

The set of cues we use for half-limb detection are: contour, shape, shading, and focus.

Contour Cue: The boundary contour cue measures how salient the contour of a segment is and how well-separated it

is from the background. We use the probability of boundary, pb [6], to measure the goodness of a segment at the low-level. The contour cue of a segment is the average pb along its boundary.

Shape Cue: We use a simplified model, namely a rectangle, to capture the basic shape characteristics of half-limbs. For a segment, we estimate its size and its orientation and construct a rectangle template with the estimates. The shape cue is then the area of overlap between the segment and this reconstructed rectangle. We experimented with an exemplar-based approach, using Hausdorff distance to match segments to 5 exemplar half-limbs. The performance is comparable to the simple rectangle model so that is what was used for our final results.

Shading Cue: Because limbs are roughly cylindrical, shading is often a salient cue, providing a strong sense of 3D pop-out (e.g. thighs on Figure 1(a)). As can be seen from Figure 5, the shading effect is present in many of the limbs, though in a weak and noisy form, unsuitable for methods like Shape-from-Shading. Instead, we capture this effect by learning a coarse *shading descriptor* from the hand-labeled limbs, and use it to score the potential limb regions. The shading descriptor is constructed as follows: each segment is first rotated and scaled into a standard 40x100 template image I using the major/minor axis of the best-fitting ellipse. From I, gradient images I_x and I_y are computed and half-wave rectified into four non-negative channels I_x^+, I_x^-, I_y^+, I_y^- so that $I_x = I_x^+ - I_x^-$ and $I_y = I_y^+ - I_y^-$. The four channels are then blurred with a Gaussian and stacked together into a single vector. To describe the shading pattern on a prototypical limb, we compute a mean of all shading descriptors in the training set (Figure 5). This mean shading descriptor is then compared with each new region using normalized correlation to yield a shading score.

Focus Cue: The background sometimes lacks texture or is out of focus. The lack of high-frequency information (or energy) is usually an indication that the region is not of interest. For the images of baseball players we work with, this "focusness" is often a useful cue. We measure the focus C_{focus} by high-frequency energy E_{high} normalized by low-frequency energy E_{low}, in the following form: $C_{focus} = E_{high}/((E_{low}{**}a) + b)$. The energy E_{high} and E_{low} are routinely estimated by convolving the image with odd and even filters and combining their responses. The parameters a and b are selected by cross-validation, with roughly $a = 0.3$ and $b = 0$.

Combining the Cues: The cues we have defined above are not directly comparable to one another. We use a sigmoid function to transform each cue into a probability-like quantity and then combine them linearly. The weights are learned through logistic regression, from the hand-marked half-limbs (Figure 5).

4.1.2 Evaluating the Cues

Figure 6 shows the average number of detections for individual cues and for the combined classifier. We find that the contour cue, or the presence of a strong boundary contour, is likely the most useful. The shading cue by itself is also good. The shape cue is relatively poor, showing that the rectangle model is not sufficient to capture the shape variations of half-limbs, especially the upper-limbs. The focus cue, being crude and generic, is the worst. Nevertheless, the four cues together make a good half-limb detector.

Another way to evaluate the performance of our classifier is to look at the percentage of images which have at least k half-limbs being detected. The results are shown in Figure 7. In our dataset 89% of the images have at least $k = 3$ correct half-limbs among the top 8 candidates returned by the trained detector. This motivates us to do a combinatorial search for a triple of half-limbs to be combined with head and torso.

4.1.3 Extending to Full Limbs

Given a half-limb we can instantiate a full-limb detector by attempting to extend it in all possible directions by adding a second rectangle at a joint of the first. Note that this extension is only done to all nearby superpixels, thereby reducing the complexity of the search. The process of scoring the second rectangle is done using a segmentation criterion similar to that of [10].

$$\begin{aligned} S(R) &= \frac{\sum_{i\in R, j\in R^c} W_{ij}^{ext}}{\sum_{i\in R, j\in R} W_{ij}^{int}} \\ W_{ij}^{ext} &= e^{-\chi^2(C_i,C_j)/\sigma_c + B(i,j)/\sigma_b} \\ W_{ij}^{int} &= e^{-\chi^2(C_i,C_j)/\sigma_c} \end{aligned}$$

where $B(i,j)$ is the average pb along the boundary between superpixels i and j. Essentially, this criterion is optimized when the internal similarity of a region is high, while the affinity to neighboring superpixels is low.

This extension search can be made more accurate if we know the body part label (upper or lower leg, arm, etc.), since we can be provided with constraints on the size and appearance of the other half of the limb. Section 5 addresses the constraining of this search.

4.2. Finding Torsos

Dominating in size and relatively rigid, the head/torso combination has a characteristic shape. Connecting the other parts together, the torso is also the most critical in recovering the body configuration. To find the torsos, we again use a segmentation-based approach. We test its performance against an alternative approach based on exemplars. The metric for success that we use in our quantitative experiments is based on the proximity of the recovered torso joints (hips and shoulders) to ground truth data.

Figure 5: Hand-segmented limbs used for training.

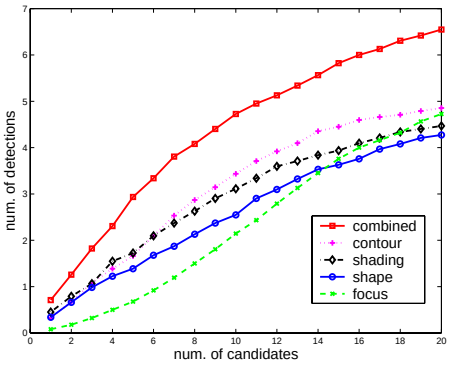


Figure 6: Evaluating half-limb cues: we fix the number of top candidates generated for each image and show the average number of half-limb detections for individual cues and the linear classifier combining all the cues. Among the top 8 candidates, in average there are 4.08 true positive detections.

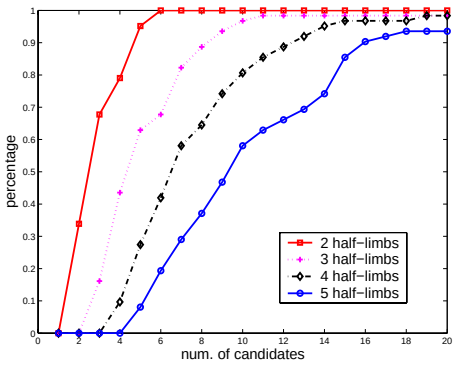


Figure 7: Evaluating half-limb detection: the percentage of images which have at least k half-limbs being detected. 89% of the images have at least 3 half-limbs being detected among the top 8 candidates.

4.2.1 Finding Torsos with Segmentations

We use the same Normalized Cuts segmentations to generate candidate torsos. Unlike half-limbs, a torso typically consists of more than one segment. Given a segmentation, we look for all combinations of segments which meet the scale constraint, i.e., contained in a bounding box of suitable scale. We then classify these candidates with a similar set of cues: contour, shape, and focus. The contour cue and the focus cue are exactly the same as in the detection of half-limbs. We again use a simple rectangle model for torso shape. The shading cue is dropped in this case, as we do not expect the torsos to have a characteristic shading pattern.

Putting Head and Torso Together: The segmentation-based torso detector works well in finding the trunk of segments comprising the torso. However, we also need the orientation of a torso to determine the shoulder and hip positions. There are a variety of poses in the baseball player images, including some players lying parallel to the ground plane. The orientation is difficult to estimate by torso itself. For this we need to put head and torso together.

For each candidate torso and each orientation, we find the best matching head. A candidate head consists of one or a pair of segments. The same set of cues, contour, shape and focus, are used to evaluate the score of a candidate head. The shape model of the head is simply a disk, whose scale is determined by the candidate torso. One obvious extension is to add an off-the-shelf frontal face detector as further evidence for the presence of a head. We have not done it here, however, since it won't be too useful on our dataset – most players are either not looking at the camera or their faces are obscured by their caps.

The torso score, the head score, and another simple score for their relative positions (a Gaussian model learned from hand-labeled data) are then multiplied into a score for the head/torso combination, which specifies an orientation of the torso and the joint positions of shoulders and hips.

We conducted experiments using 62 images of baseball players from our set of sports news photographs. Ground truth positions of the torso joints were marked by hand, a recovered torso is deemed to be correct if all 4 torso joints are within 60 pixels of true positions. The results of running the segmentation-based torso detector are shown on Figure 8 (solid green line). For reference, the mean head diameter in the images is about 50 pixels.

Comparing with Exemplar-based Approach: As further evaluation of the segmentation-based torso detector, we compare it with an earlier exemplar-based approach. The exemplar-based torso detector is based on the work in [8]. We hand-label torso joint positions in a set of training images to be used as exemplars. Exemplars are represented as collections of edges, obtained using the texture-suppressing edge detector in [6]. We use the technique of representative shape contexts to match test images to stored exemplars. The shape contexts used in experiments have a rather large spatial extent, with a radius of approximately $\frac{1}{3}$ of the height of a person. Information from the configuration of the entire body is being used in these descriptors.

For the exemplar-based torso detector, a leave-one-out testing strategy was employed. For each test image, we used the other 61 images as the exemplars for matching. Results for this exemplar-based torso detector are shown on Figure 8 (black line). It illustrates that with this set of exemplars we are unable to cope with the variation in appearance among the different players.

5. Assembling Body Parts

Using the candidate limbs and torsos from the procedures described above, we would like to generate a shortlist of

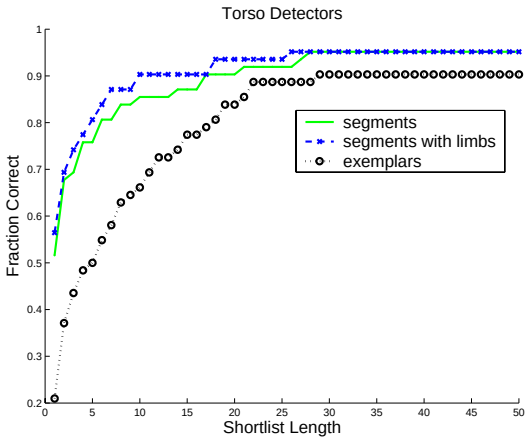


Figure 8: Evaluating the torso finders: the X-axis is the number of top candidate torsos (length of "shortlist"), the Y-axis the percentage of images of which we find a "correct" torso in these candidates. A torso is "correct" if all the four joint positions of shoulders and hips are within a threshold (1 to 1.5 head size) of the ground truth. The segmentation-based detector outperforms the exemplar-based detector, mainly because of the limited amount of training data. The separately detected limbs provide significant constraints for torsos: the combined segmentation-based limb and torso detector performs the best. For 80% performance, the exemplar-based method requires 18 torsos, the segmentation based one 6, the combined limb and segmentation one 5. At 90% the lengths are 29, 17, and 10.

possible complete body configurations. A simple method of assembling configurations is to take each torso and independently select the best limb to connect to the torso joints (hips and shoulders) based on the limb detector outputs. This is akin to the dynamic programming-based approaches used in previous work.

However, there are dependencies between the limbs of the body that would not be captured in such an approach. Instead, we reason about which limb candidates can and can not be used together, using constraints on relative widths and lengths of limbs, and symmetry in clothing between certain limbs.

5.1 Enforcing Global Constraints

From L candidate half-limbs and T candidate torsos, we generate a set of partial body configurations. A partial configuration consists of 3 of the candidate half-limbs (L choose 3)[1], each fixed as one of the 8 body segments in our model ($8 \cdot 7 \cdot 6$), along with one of the T torso candidates. Once we have fixed candidate half-limbs to specific body parts, we can instantiate the limb detectors described above. Since each candidate half-limb also needs an associated "polarity" (e.g. for lower arm, wrist-to-elbow or elbow-to-wrist, so 2^3), there are:

$$\begin{pmatrix} L \\ 3 \end{pmatrix} \cdot 8 \cdot 7 \cdot 6 \cdot 2^3 \cdot T$$

possible partial configurations. In our experiments, L is between 5 and 7, and T is about 50, leading to 2-3 million partial configurations.

Of the millions of possible partial configurations enumerated above, many are physically impossible due to their violation of global anthropometric and kinematic constraints. We can efficiently prune the set of partial configurations using these constraints while assuming scaled orthographic projection. The specific constraints we use are:

Relative widths: Projection leads to foreshortening of the lengths of body segments. However, we are able to estimate the widths of the proposed candidate segments. We use these estimated widths, along with anthropometric data to prune combinations that have relative widths more than 4 standard deviations away from their estimated means.

Length given torso: We also assume that the torso is not horribly foreshortened. In our experiments we assume that the torso forms an angle with the image plane between -40^o and 40^o. We estimate the length of the torso in the image, then use this assumption and our data on relative lengths of segments in turn to yield conservative upper bounds on the lengths of the limbs. Again we prune configurations containing a limb 4 standard deviations greater than its estimated mean length.

Adjacency: If a partial configuration contains an upper limb, it must be adjacent to its corresponding joint on the torso. Similarly, if both the upper and lower portions of a limb are set, they must be adjacent at the elbow/knee. Furthermore, given upper bounds on limb lengths as above, constraints exist for solitary lower limbs as well. These simple kinematic constraints are used to prune disconnected partial configurations.

Symmetry in clothing: The final constraint we use is that the color histograms (measured in Lab space) of corresponding segments (e.g. left and right lower arms) must not be wildly dissimilar (χ^2 distance < 0.3). This often prevents an arm to be matched to a piece of grass, for example.

Fixing 3 half-limbs to specific body parts activates some subset of these constraints. For our experiments, we selected a threshold for each of these pruning mechanisms such that we reduce the original 2-3 million configurations down to approximately 1000.

5.2 Sorting and Completing Configurations

After the pruning phase we have a set of partial body configurations, each consisting of a torso along with 2 or 3 limbs (assembled from 3 half-limbs). With each of these configurations, there are the outputs of the associated limb detectors

[1] The number 3 is chosen based on Figure 7.

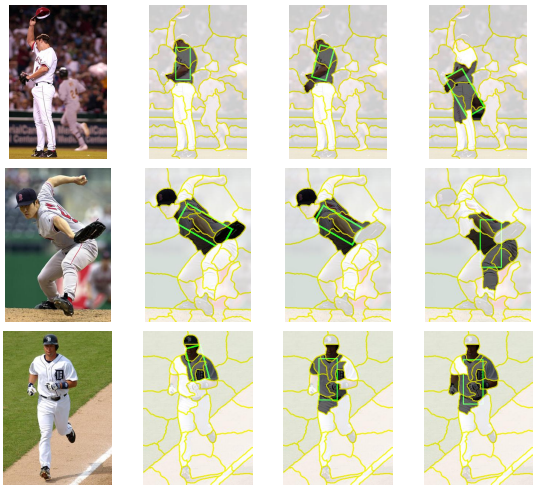


Figure 9: Examples of top-ranked torsos.

and torso detectors. We assign the final score for a configuration to be a linear combination of these limb and torso scores. The results of using this combined score as a new torso detector are also shown in Figure 8 and Figure 9.

To go from partial configurations (with missing limbs) to the full human figures, we search for entire limbs off of empty torso joints. The resulting full configurations are shown in Figure 1 and Figure 10. The configurations are ordered by their combined limb and torso score. Figure 10 shows one selected entry (and its corresponding rank) on the final shortlist of body configurations for each image. The problem of automatically evaluating these shortlists of configurations to find the correct one is a challenging problem, and is not addressed here.

6. Discussion

In this work we have demonstrated how to use low-level segmentation to drive recognition. Segments and superpixels generated by the Normalized Cuts algorithm are used to propose candidates for limbs and torsos. These candidates are then verified using a variety of cues. After that, finding consistent body configurations becomes a *Constraint Satisfaction Problem* [11]. While not pursued in this paper, it would be straightforward to apply techniques from the AI heuristic search literature (e.g., best-first search and A^*) to reduce the computational complexity.

The problem of recovering human body configurations in a general setting is arguably the most difficult recognition problem in computer vision. By no means do we claim to have solved it here; much work still remains to be done. An approach such as ours, based on assembling configurations from detected salient parts, has the advantage of "compositionality" – there is no need to store templates for all possible configurations as in an exemplar-based approach. However, this modularity comes at a definite cost. Detecting and localizing parts is more reliably done when provided with the context that an entire exemplar gives. We believe that combining these two approaches in a sensible manner is an important topic for future work.

Acknowledgments: We thank Jaety Edwards for help in collecting the dataset. This work was supported by ONR grant N00014-01-1-0890 (MURI), NSF ITR grant IIS-00-85864, and a Digital Library Grant 1R1-9411334.

References

[1] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient matching of pictorial structures. In *Proc. IEEE Comput. Soc. Conf. Comput. Vision and Pattern Recogn.*, 2000.

[2] D. M. Gavrila. The visual analysis of human movement: A survey. *Computer Vision and Image Understanding: CVIU*, 73(1):82–98, 1999.

[3] D. Hogg. Model-based vision: A program to see a walking person. *Image and Vision Computing*, 1(1):5–20, 1983.

[4] S. Ioffe and D. Forsyth. Probabilistic methods for finding people. *Int. Journal of Computer Vision*, 43(1):45–68, 2001.

[5] J. Malik, S. Belongie, T. Leung, and J. Shi. Contour and texture analysis for image segmentation. *Int. Journal of Computer Vision*, 43:7–27, 2001.

[6] D. Martin, C. Fowlkes, and J. Malik. Learning to find brightness and texture boundaries in natural images. *NIPS*, 2002.

[7] A. Mohan, C. Papageorgiou, and T. Poggio. Example-based object detection in images by components. *IEEE Trans. PAMI*, 23(4):349–361, 2001.

[8] G. Mori and J. Malik. Estimating human body configurations using shape context matching. In *European Conference on Computer Vision LNCS 2352*, volume 3, pages 666–680, 2002.

[9] NIST. Anthrokids - anthropometric data of children, http://ovrt.nist.gov/projects/anthrokids/, 1977.

[10] X. Ren and J. Malik. Learning a classification model for segmentation. In *Proc. 9th Int. Conf. Computer Vision*, volume 1, pages 10–17, 2003.

[11] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice-Hall, Englewood Cliffs, NJ, second edition, 2003.

[12] G. Shakhnarovich, P. Viola, and T. Darrell. Fast pose estimation with parameter sensitive hashing. In *Proc. 9th Int. Conf. Computer Vision*, volume 2, pages 750–757, 2003.

[13] Y. Song, L. Goncalves, and P. Perona. Unsupervised learning of human motion. *IEEE Trans. PAMI*, 25(7):814–827, 2003.

[14] J. Sullivan and S. Carlsson. Recognizing and tracking human action. In *European Conference on Computer Vision LNCS 2352*, volume 1, pages 629–644, 2002.

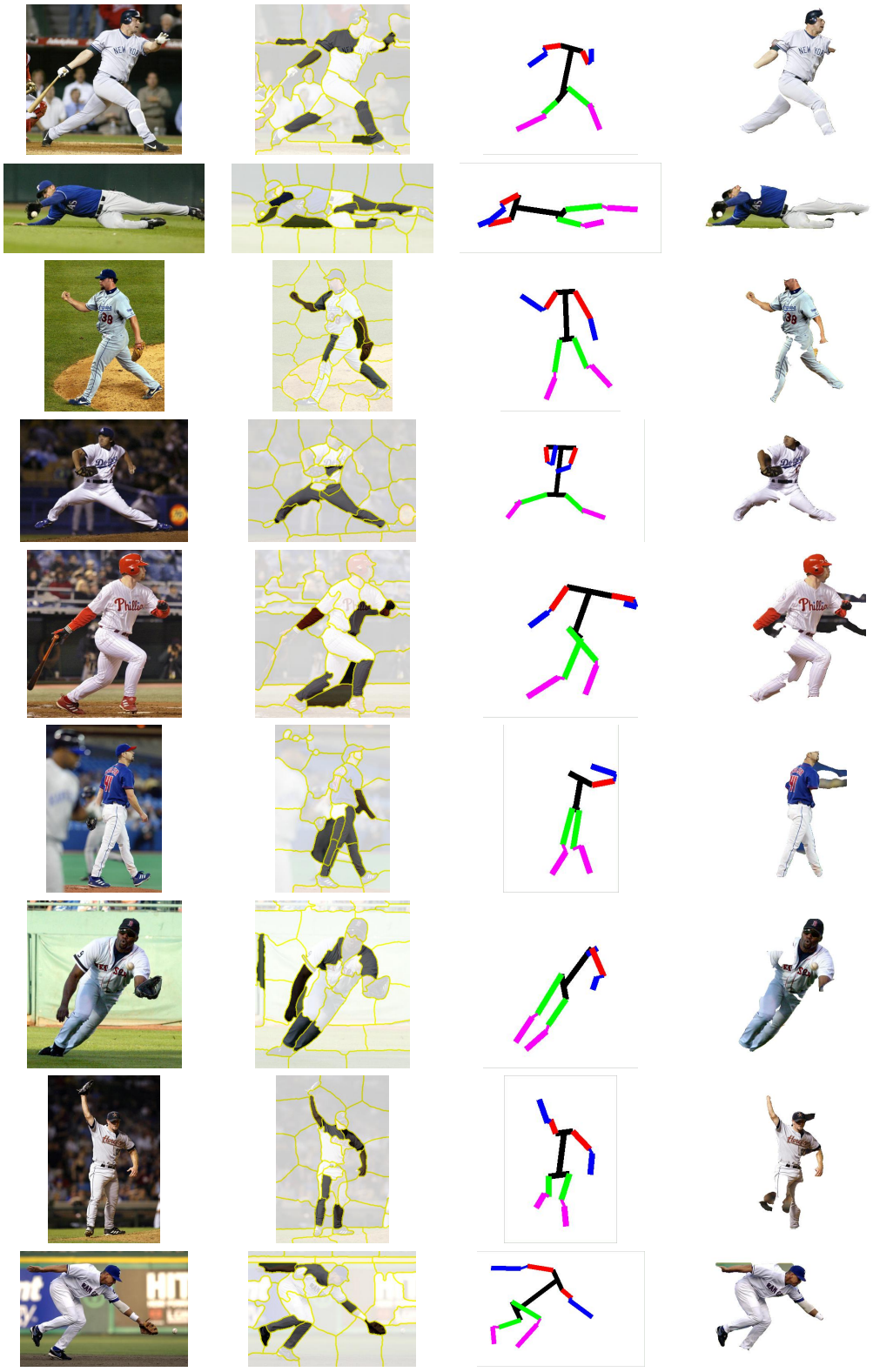
[15] A. Torralba, K. P. Murphy, W. T. Freeman, and M. A. Rubin. Context-based vision system for place and object recognition. In *ICCV*, pages 273–280, 2003.

[16] K. Toyama and A. Blake. Probabilistic exemplar-based tracking in a metric space. In *Proc. 8th Int. Conf. Computer Vision*, volume 2, pages 50–57, 2001.

[17] Z. Tu, X. Chen, A. Yuille, and S. Zhu. Image parsing: segmentation, detection, and recognition. In *Proc. 9th Int. Conf. Computer Vision*, pages 18–25, 2003.

[18] P. Viola, M. Jones, and D. Snow. Detecting pedestrians using patterns of motion and appearance. In *Proc. 9th Int. Conf. Computer Vision*, pages 734–741, 2003.

[19] S. X. Yu and J. Shi. Object-specific figure-ground segregation. In *Proc. IEEE Comput. Soc. Conf. Comput. Vision and Pattern Recogn.*, volume 2, pages 39–45, 2003.



(a) (b) (c) (d)

Figure 10: Selected results from shortlist of final configurations. (a) Input image, (b) Candidate half-limbs, (c) Extracted body configurations, (d) Associated segmentation. One body configuration from the shortlist for each image is shown. Shortlist rankings for each configuration, rows top to bottom: (1^{st},1^{st},11^{th},1^{st},18^{th},1^{st},3^{rd},3^{rd},1^{st})