Unsupervised Discovery of Action Classes

Yang Wang, Hao Jiang, Mark S. Drew, Ze-Nian Li and Greg Mori
School of Computing Science
Simon Fraser University
Burnaby, BC, Canada
{ywang12,hjiangb,mark,li,mori}@cs.sfu.ca

Abstract

In this paper we consider the problem of describing the action being performed by human figures in still images. We will attack this problem using an unsupervised learning approach, attempting to discover the set of action classes present in a large collection of training images. These action classes will then be used to label test images. Our approach uses the coarse shape of the human figures to match pairs of images. The distance between a pair of images is computed using a linear programming relaxation technique. This is a computationally expensive process, and we employ a fast pruning method to enable its use on a large collection of images. Spectral clustering is then performed using the resulting distances. We present clustering and image labeling results on a variety of datasets.

1. Introduction

Does a single image convey an action? Consider the images in Figure 1. Even though only a single image is available, an action is conveyed. Motion is definitely an important cue for action recognition, but even in still images it is possible to discern actions. Moreover, many actions (such as standing or sitting) are defined by a static body pose, rather than a particular motion pattern. Hence, static pose is an essential component of action recognition. In this paper we will attempt to address this problem of action recognition from still images.

The datasets which we use for our experiments are still frames from figure skating video sequences, and collections of sports news photographs (baseball and basketball) collected from the Internet. We will assume that each image contains human figure(s) performing some action. These are certainly challenging datasets, particularly the sports news photographs. A major challenge is that real-world datasets of this variety contain a substantial amount of "noise", in the form of unrepeated or unusual actions. One of the goals of this work is to try to discover what actions are present in a particular dataset. To this end we will first phrase the problem of action recognition as an unsupervised learning problem, attempting to summarize the actions in a dataset. We will do this by clustering images into groups with people in similar body poses.

Figure 1. Examples of still images from our datasets. Even though only a single image is available, it is still possible to perceive an action being performed.

The main cue which we will try to exploit is the overall shape of the human figure(s) present in an image. We will represent shape as a collection of edges points, obtained via Canny edge detection. We will use shape in order to have a method which is less sensitive to the clothing worn by different people.

At the heart of our method is a technique for deformable matching of the edges of a pair of images, which will be used to measure the distance between images. This method is based upon a linear programming relaxation technique. Even though it is effective, applying it to large collections of images is computationally intractable. As such, we employ a method for fast pruning which will quickly narrow down the search to a shortlist of images which are likely to be similar. The deformable matching is applied only to these shortlisted images. Once we have obtained a sparse matrix of distances between pairs of images, we will apply spectral clustering to obtain our action classes.

Given these clusters of actions which can be discerned using static body pose, we will manually assign them labels. Prototypes from these clusters can then be used to classify new images according to these labels. We perform a quantitative evaluation of our method by measuring accuracy of this labeling relative to a control of random labels. Promising results are shown on challenging datasets.

The structure of this paper is as follows. In Section 2 we review previous work. Section 3 gives an overview of our approach. Sections 4, 5, and 6 describe the pruning, deformable matching, and clustering respectively. We show clustering and image labeling results in Section 7, and conclude in Section 8.

2. Previous Work

The problem of activity recognition has received a large amount of attention from the computer vision community. Shah and Jain [15] provide a review of this body of work. Much of this work is focussed on analyzing patterns of motion. For example, Cutler and Davis [5], and Polana and Nelson [13] detect and classify periodic motions. Little and Boyd [10] perform gait recognition by analyzing the periodic structure of optical flow patterns. Rao et al. [14] describe a view-invariant representation for 2D trajectories of tracked skin blobs. An incremental learning procedure is used to automatically build a vocabulary of actions. Moore et al. [11] use context provided by object detection to aid in activity recognition.

Bobick and Davis [3] develop a representation known as "Temporal Templates" that captures both motion and shape, represented as evolving silhouettes. However, the extraction of these silhouettes is based on background subtraction and would not be applicable for this problem.

Our method considers the shape of the human figure, represented as a collection of edges. Other similar work includes Gavrila and Philomin [7], who consider pedestrian images, and compare them by Chamfer matching on edge maps. They present a method for automatically constructing a hierarchy of pedestrian shapes, for the end goal of efficient detection. Sullivan and Carlsson [18] perform action recognition by matching test images to labelled keyframes, using "order structure" to compare the shape of extracted edges.

The idea of attempting to discover the set of action classes from a large collection of images is motivated by the work of Sivic et al. [17] and Fergus et al. [6] who attempt to discover object classes from collections of images.

For action recognition, in addition to the aforementioned work of Rao et al. [14], Hoey [8] presents a method for unsupervised learning of HMMs of facial expressions. Zhong et al. [20] cluster segments of long video sequences by looking at co-occurrences of patterns of motion and appearance. Xiang and Gong [19] automatically discover activity classes from video sequences, modelled using a variant of HMMs. Boiman and Irani [4] explain a video sequence using patches from a database, declare non-matching components to be unusual.

This work is also motivated by that of Berg et al. [2], who detect faces in a large collection of news photographs, and learn a mapping of names to faces, based on the names contained in the associated captions. Some of our datasets are also collections of news photographs. One might hypothesize that the action verb contained in the caption corresponds to the action of the person in the image. Unfortunately, we found this not to be the case. The caption frequently refers to events at a coarser level of detail. For example, a caption of "The New York Yankees *beat* the Los Angeles Dodgers", does not indicate the presence of any beating occurring in the actual image.

3. Approach

We will attempt to take a large collection of still images and form clusters corresponding to people in similar body poses. Spectral clustering [16] will be used to compute these clusters. In spectral clustering, with n images an n-by-n affinity matrix W is constructed, where W_{ij} stores the affinity, or similarity between images i and j.

We desire a measure of affinity which will place a high value on images of people in similar poses, and a low value on those in dissimilar poses. As such, we will define a distance measure based on a deformable template matching cost between a pair of images. We will then turn this distance into an affinity in the usual fashion, passing it through an exponential.

The deformable template matching algorithm we use extracts a set of sample points from the edges in the images, and tries to find an optimal assignment between these sample points in the two images. The distance for matching one image to another is defined as a sum of two terms. The first measures similarity between matched landmarks, and the second measures relative spatial deformation between pairs of landmarks. This matching problem is an instance of an Integer Linear Programming (ILP) problem, which we solve via relaxation to Linear Programming (LP) and iteratively making convex the original nonconvex first term of the cost function within the current trust region, as done by Jiang et al. [9].

This LP relaxation technique is very efficient, and our implementation solves for the matching between a pair of images in approximately 2-3 seconds. However, if we need to build a 4000-by-4000 affinity matrix, it is intractable to run this technique to compare every pair of images. Instead, for each of our n images, we will run a fast pruning method to reduce the set of candidate matches down to a shortlist of manageable size.

The fast pruning method we use is the representative

shape contexts algorithm [12], which uses the shape context descriptor of Belongie et al. [1]. The shape context is a large scale shape descriptor, and is useful for capturing the coarse shape of objects. This coarse shape cue is adept at capturing the rough overall body pose of the people in an image. When constructing a shortlist of candidate matches, this is the type of information we require.

There are definitely issues with using shape contexts in this fashion. Since they are large scale descriptors, shape contexts will be affected by background clutter. Also, for reasons of efficiency, the representative shape contexts algorithm disregards spatial structure, and simply attempts to match individual shape contexts. We will be willing to make these two sacrifices in order to have an efficient pruning method. The deformable template matching algorithm will attempt to remedy mistakes made by the shape context-based pruning algorithm. It will use a larger set of different, local scale descriptors, and will measure deformation of spatial structure.

In the following sections we describe the details of the pruning, deformable matching, and clustering algorithms.

4. Fast Pruning using Shape Contexts

The deformable matching process mentioned above is computationally expensive. With a large set of images (our datasets contain 1000s of images), performing an exhaustive comparison between every pair of images is not feasible. Instead, for each image we use an efficient pruning algorithm to compute a shortlist of promising candidates. Each image will only be compared to its small set of candidates using the expensive deformable matching process.

In particular, we use the *representative shape contexts* pruning algorithm [12] to construct this shortlist of candidates. This method relies on the descriptive power of just a few shape contexts. Given a pair of images of very different human figures, such as a baseball pitcher throwing a ball, and a batter swinging, none of the shape contexts from the pitcher will have good matches on the batter – it is immediately obvious that they are different shapes. The representative shape contexts pruning algorithm uses this intuition to efficiently construct a shortlist of candidate matches.

In concrete terms, the pruning process proceeds in the following manner. For each image i, we precompute a large number s (about 500) of shape contexts $\{SC_i^j : j = 1, 2, \ldots, s\}$. When constructing a shortlist for image i, we only use a small subset of these shape contexts, of size r ($r \approx 5 - 10$ in experiments), called representative shape contexts (RSCs). To decide on the location of these r RSCs we randomly select r sample points from the edges of the image via a rejection sampling method that spreads the points over the entire image. To compute the distance between image i and another image j, we find the best matches for each of the r RSCs from image i, using the entire set of shape contexts for image j.

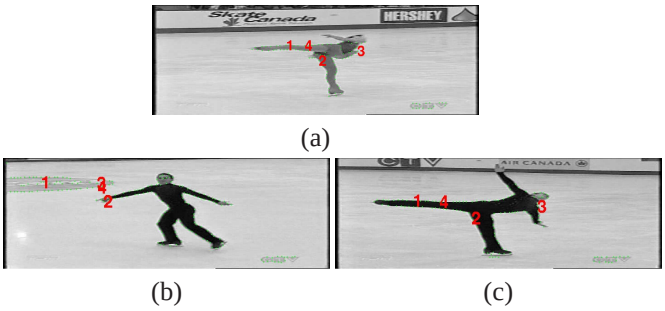

Figure 2. Example of representative shape context method. (a) Shows image from which 5 representative shape context (RSC) locations (numbered in red) have been chosen. (b) and (c) show best matches for each of the RSCs from (a). Images (b) and (c) have similar d_S distances from (a). Note that spatial structure is not preserved in image (b).

We take the best k of these r matches, denoted by the set G, and the distance between images i and j is then:

$$d_S(i,j) = \frac{1}{k} \sum_{u \in G} \min_v d_{SC}(SC_i^u, SC_j^v) \tag{1}$$

where $d_{SC}(\cdot,\cdot)$ denotes the distance between a pair of shape contexts.

We determine the shortlist by sorting these distances $d_S(\cdot,\cdot)$. Figure 3 shows some examples of shortlists retrieved via RSC pruning. Some of the images on the shortlist are of human figures in a similar pose to that of the query image. Note that many of the errors are caused by the lack of spatial structure when using RSC pruning, along with background clutter corrupting the shape contexts. The subsequent deformable matching stage will attempt to refine this shortlist, using spatial structure and more local features. Figure 2 illustrates an example of RSC matching.

5. Deformable Matching

The deformable template matching algorithm we use extracts a set of sample points from the edges in two images, and tries to find an optimal matching of sample points in one image to those in the other. The distance from one image to another is the minimum, over all possible matchings, of the energy function E below. The energy function is defined as a sum of two terms. The first measures similarity between matched sample points, and the second measures relative spatial deformation between pairs of sample points. Note that this distance is not symmetric.

$$\min_{\mathbf{f}} E : \sum_{\mathbf{s} \in S} c(\mathbf{s}, \mathbf{f_s}) + \sum_{\{\mathbf{p},\mathbf{q}\} \in \mathcal{N}} \lambda_{\mathbf{p},\mathbf{q}} ||(\mathbf{f_p} - \mathbf{p}) - (\mathbf{f_q} - \mathbf{p})|| \tag{2}$$

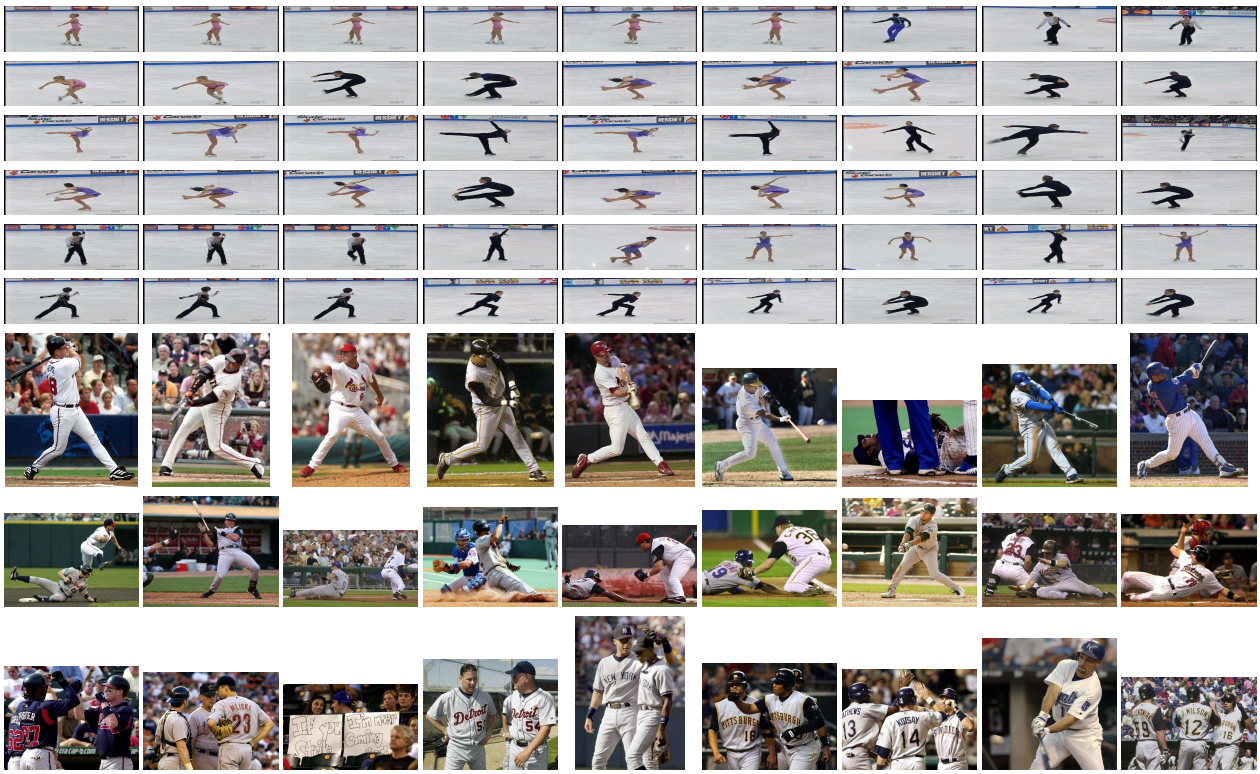

Figure 3. Example shortlists. Column (a) shows query image, columns (b-i) columns show shortlist of candidate matches from representative shape context pruning. For baseball images, these are the first 8 entries of the shortlist. In the figure skating images, every second shortlist image is shown since there is much redundancy due to temporal structure. In our experiments a shortlist of length 50 is kept for each image. Human figures in poses similar to that in the query image are retrieved, along with some incorrect images. Examples such as row 4 are common, and illustrate the difficulties in our datasets.

In this optimization problem, $\mathbf{f_s}$ is the matched point in the target image for feature point $\mathbf{s}$ on a template; The cost of matching feature point $\mathbf{s}$ with target feature point $\mathbf{f_s}$ is denoted as $c(\mathbf{s}, \mathbf{f_s})$. The second term in the objective function is a regularity term, used to smooth the matching vectors for neighbor feature points. $\lambda_{p,q}$ are coefficients to control the weight of the smoothness term. The norm $||\cdot||$ is the L_1 norm.

In our experiments, we first convert the binary edge maps into gray-scale images by distance transformation. $c(\mathbf{s}, \mathbf{f_s})$ is measured by taking a small 9×9 patch of pixels around each feature point, and calculating the normalized sum of absolute value distance between two patches.

The connectivity pattern among template sample points, $\mathcal{N}$, is the set of edges in the Delaunay triangulation of these sample points. The weighting parameter $\lambda_{\mathbf{p},\mathbf{q}}$ is fixed at 0.8 for all pairs.

The energy optimization problem in Eq. 2 is nonlinear and usually non-convex, which makes it difficult to solve in this original form without a good initialization process. We followed the convexification scheme of Jiang et al. [9] and relax the problem into linear programming. To linearize the first term, the following scheme is applied. A basis $\mathcal{B}_\mathbf{s}$ is selected for the targets for each site $\mathbf{s}$. Then the point $\mathbf{f_s}$ can be represented as a linear combination of the target point basis as $\mathbf{f_s} = \sum_{\mathbf{t}\in\mathcal{B}_\mathbf{s}} \xi_{\mathbf{s},\mathbf{t}} \cdot \mathbf{t}$, where $\xi_{\mathbf{s},\mathbf{t}}$ are real-valued weighting coefficients. The matching cost of $\mathbf{f_s}$ can then be approximated by the linear combination of the cost of the basis labeling costs $c(\mathbf{s}, \sum_{\mathbf{t}\in\mathcal{B}_\mathbf{s}} \xi_{\mathbf{s},\mathbf{t}} \cdot \mathbf{t}) \simeq \sum_{\mathbf{t}\in\mathcal{B}_\mathbf{s}} \xi_{\mathbf{s},\mathbf{t}} \cdot c(\mathbf{s}, \mathbf{t})$. We also further set constraints $\xi_{\mathbf{s},\mathbf{j}} \geq 0$ and $\sum_{\mathbf{j}\in\mathcal{B}_s} \xi_{\mathbf{s},\mathbf{j}} = 1$ for each site $\mathbf{s}$. Apparently, if $\xi_{\mathbf{s},\mathbf{j}}$ are constrained to be 1 or 0, and the basis contains all the target points, the above representation becomes exact. To linearize the regularity terms in the nonlinear formulation we can represent a variable in the absolute value term by the difference of two nonnegative auxiliary variables and introduce the summation of the auxiliary variables into the objective function. If the problem is properly formulated, when the linear programming problem is optimized the summation will approach the absolute value of the free variable.

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

$$\min LP: \quad \sum_{\mathbf{s}\in S}\sum_{\mathbf{j}\in\mathcal{B}_\mathbf{s}} c(\mathbf{s},\mathbf{j})\cdot\xi_{\mathbf{s},\mathbf{j}} + \sum_{\{\mathbf{p},\mathbf{q}\}\in\mathcal{N}} \lambda_{\mathbf{p},\mathbf{q}} \sum_{m=1}^{2} (f^+_{\mathbf{p},\mathbf{q},m} + f^-_{\mathbf{p},\mathbf{q},m}) \quad (3)$$

$$s.t. \quad \sum_{\mathbf{j}\in\mathcal{B}_\mathbf{s}} \xi_{\mathbf{s},\mathbf{j}} = 1, \forall \mathbf{s}\in S \quad (4)$$

$$\sum_{\mathbf{j}\in\mathcal{B}_\mathbf{s}} \xi_{\mathbf{s},\mathbf{j}}\cdot\phi_m(\mathbf{j}) = f_{\mathbf{s},m},\ \forall \mathbf{s}\in S,\ m=1,2 \quad (5)$$

$$f_{\mathbf{p},m} - \phi_m(\mathbf{p}) - f_{\mathbf{q},m} + \phi_m(\mathbf{q}) = f^+_{\mathbf{p},\mathbf{q},m} - f^-_{\mathbf{p},\mathbf{q},m}, \quad (6)$$

$$\forall\ \{\mathbf{p},\mathbf{q}\}\in\mathcal{N},\ m=1,2 \quad (7)$$

$$\xi_{\mathbf{s},\mathbf{j}},\ f^+_{\mathbf{p},\mathbf{q},m}, f^-_{\mathbf{p},\mathbf{q},m} \geq 0 \quad (8)$$

where $\mathbf{f_s} = (f_{\mathbf{s},1}, f_{\mathbf{s},2})$ and function ϕ_m returns the mth component of its argument. When the linear program is optimized, we have either $f^+_{\mathbf{p},\mathbf{q},m}$ or $f^-_{\mathbf{p},\mathbf{q},m}$ will become zero, and thus $|f_{\mathbf{p},m}-\phi_m(\mathbf{p})-f_{\mathbf{q},m}+\phi_m(\mathbf{q})| = f^+_{\mathbf{p},\mathbf{q},m}+f^-_{\mathbf{p},\mathbf{q},m}$.

The linear program in fact convexifies the matching cost surface $c(\mathbf{s},\mathbf{t})$ for each site $\mathbf{s}$. By fixing $\mathbf{s}$, $c(\mathbf{s},\mathbf{t})$ is a surface with respect to $\mathbf{t}$, and the surface is replaced by its lower convex hull for each site implicitly by the linear program. Because of this property, we can choose the most compact basis set $B_\mathbf{s}$ by using only the target points corresponding to the vertices of the lower convex hull surface for each $\mathbf{s}$. This number is usually much smaller than the whole target set and enable more efficient searching. It is not difficult to show that any basic feasible solution of the linear program has at most 3 basic variables from ξ of each site. Therefore, when using the simplex method, there will be at most 3 nonzero-weight basis targets for each site. This makes the searching process efficient. To further refine the matching, instead of using one step relaxation, we iteratively shrink the the matching trust region and build new LP relaxations in the smaller regions. In most cases, 3 to 4 iterations are involved.

A standard simplex method is used to solve the LP problem. The estimate of the average complexity of successive linear programming is $O(|S|\cdot|Q|^{1/2}\cdot(\log|Q|+\log|S|))$, where S is the template point set and Q is the target point set. Experiments also confirm that the average complexity of the proposed optimization scheme increases more slowly with the size of target point set than other methods whose average complexity is usually linear with respect to $|Q|$.

6. Clustering

We use spectral clustering [16] to cluster the images into groups of similar body poses. Spectral clustering requires an n-by-n affinity matrix, where n is the number of images. Each entry W_{ij} is calculated as $e^{-d^2_{ij}/h}$, where d_{ij} is the distance from image i to j obtained from the deformable matching in section 5. We notice that d_{ij} might not be available, and d_{ij} and d_{ji} may not be equal, so we set W_{ij} as $e^{-(\frac{d_{ij}+d_{ji}}{2})^2/h}$ if both d_{ij} and d_{ji} are available, as $e^{-d^2_{ij}/h}$(or $e^{-d^2_{ji}/h}$) if one of them is available, as 0 if neither is available. Since we only compute the deformable matching for an image to a shortlist of candidate images, most of the entries in W are zeros. So we can use eigensolvers for sparse matrices to find its top eigenvectors.

7. Experimental Results

We applied our clustering algorithm to three datasets. The first is a collection of images from six videos of different figure skaters. These videos were automatically filtered, frames with complicated backgrounds (consisting of a large number of edges) were removed, resulting in a simplified set of 1400 images. These images were further processed to automatically remove background clutter. A Hough transform was used to remove extended straight lines that typically correspond to the boards of the rink.

The second two datasets, of baseball and basketball sports news images, consist of 4500 and 8500 images respectively. These images were collected by querying the captions of sports news photos for professional sports team names. These datasets are significantly more challenging than the figure skating set, containing substantial background clutter, and a wide range of content.

7.1. Clustering Results

We choose the number of clusters to be between 100 and 200, then sort the clusters based on the average distances to their respective cluster centers and manually look through the first 50 clusters. Sample images in some clusters being found are shown in Figure 4, Figure 5, and Figure 6, where each row corresponds to a cluster.

Qualitatively, the clustering results for the figure skating dataset are quite good. The clusters in Figure 4 are typical, and most clusters consist of people performing the same action, with a few outliers.

The baseball and especially basketball datasets prove substantially more challenging. Approximately 10-15 reasonable clusters appear among the first 50 for each of these datasets. There is enough repetition in the baseball images so that clusters containing a single person performing actions such as throwing, swinging, and sliding are found. The basketball images almost always contain multiple people, and the actions are not as stereotyped as baseball, and hence are difficult to cluster.

We also tried Chamfer matching instead of our deformable matching for the matching cost. Since Chamfer

matching cannot handle large deformation or background clutter, usually there are almost no clear clusters, unless all the images in a cluster are almost identical.

In the next section, we attempt to use good clusters, which are manually selected, to label new images. This will give us a quantitative evaluation of how well our method is working.

7.2. Image labeling

We also use the manually selected clusters for the task of automatic image labeling. Firstly, we assign each cluster a short text description. The figure skating clusters were given the following 10 labels: face close-up picture, skates with arms down, skates with one arm out, skater leans to his right, skates with both arms out, skates on one leg, sit spin leg to left of image, sit spin leg to right of image, camel spin leg to left of image, camel spin leg to right of image.

The baseball clusters were given 7 labels: face close-up picture, right-handed pitcher throws, right-handed pitcher cocks his arm to throw, runner slides into base, team celebrates, batter swings, batter finished swinging.

The basketball clusters were given the following 8 labels: a player goes for a lay-up above the defenders, a player goes for a lay-up against a defender, a player goes for a jumpshot while another one tries to block, a player goes for a lay-up leaning to his right, a player drives past another, a player has his shot blocked, a player leaps by his defender for a shot, a player posts up.

For each new test image, we match it to one of the clusters using k nearest neighbor method. Then the text description of that cluster is used to describe this new image. The quality of the matching is measured by the ratio of the images in the cluster that match this test image. This gives us a measure of confidence in each matching.

We sort the test images according to this quality of matching to their respective clusters, and present these automatically selected top 100 test images to a set of naive human subjects for evaluation. Examples of these selected images and their labels are shown in Figure 7. The human subjects are shown in advance the set of available captions for the dataset, in addition to an "other" caption. The subjects are then asked, for each image, whether the given caption is the best possible caption for the image. For a control, we did the same thing by randomly assigning the text description to test images, and asking the human subjects for their opinions on these randomly assigned labels.

Quantitative results of this labeling are presented in Figure 8. The x-axis shows the number of test images matched, and the y-axis shows cumulative accuracy. For example, a point (20,0.65) means that of the 20 images which our algorithm automatically decides are the best matching 20 of our test set, 65% of them are correctly labelled according to our human observers.

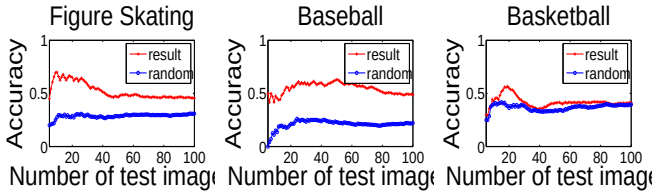

Figure 8. Quantitative results on image labeling.

Results are averaged across 4 trials of naive observers. The labeling results of our algorithm are significantly above chance performance, and achieve reasonable accuracy rates for the high quality matching images. Performance definitely degrades as we move further into the lower quality matching images.

This experiment suggests that our method has potential as a method of labeling and summarizing the repetitive actions in a large collection of images, and could be effective for image retrieval applications.

8. Conclusion

In this paper we have presented a method for discovering classes of actions in collections of still images by clustering images of people in similar body poses. We use spectral clustering, which requires making pairwise comparisons between images. The technique for making these pairwise comparisons is a deformable template matching scheme which is computationally expensive. As such, we employ a fast pruning method based on shape contexts to speed up the search for similar images.

In experiments on three challenging datasets, we have demonstrated that it is possible to extract some clusters of repetitive actions. By no means do we claim to have a method that can accurately label all images. However, we believe we have made progress on the interesting and difficult problem of action recognition from a single image.

Acknowledgement

This work is supported by a grant from NSERC (RGPIN-312230).

References

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

[2] T. L. Berg, A. C. Berg, J. Edwards, M. Maire, R. White, Y. W. Teh, E. Learned-Miller, and D. A. Forsyth. Names and faces in the news. In *Proc. IEEE Comput. Soc. Conf. Comput. Vision and Pattern Recogn.*, 2004.

[3] A. Bobick and J. Davis. The recognition of human movement using temporal templates. *IEEE Trans. PAMI*, 23(3):257–267, 2001.

Figure 4. Examples of clusters in figure skating images. Each row corresponds to a cluster.

Figure 5. Examples of clusters in baseball images. Each row corresponds to a cluster.

[4] O. Boiman and M. Irani. Detecting irregularities in images and in video. In *Proc. 10th Int. Conf. Computer Vision*, 2005.

[5] R. Cutler and L. Davis. Robust real-time periodic motion detection, analysis, and applications. *IEEE Trans. PAMI*, 22(8), 2000.

[6] R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman. Learning object categories from google's image search. In *Proc. 10th Int. Conf. Computer Vision*, 2005.

[7] D. Gavrila and V. Philomin. Real-time object detection for smart vehicles. In *Proc. 7th ICCV*, pages 87–93, 1999.

[8] J. Hoey. Hierarchical unsupervised learning of facial expression categories. In *IEEE ICCV Workshop on detection and recognition of events in video (EVENT 01)*, 2001.

[9] H. Jiang, M. Drew, and Z.-N. Li. Linear programming matching and appearance-adaptive object tracking. In *Energy Minimization Methods in Computer Vision and Pattern Recognition (EMMCVPR'05) LNCS vol. 3757*, pages 203–219, 2005.

Figure 6. Examples of clusters in basketball images. Each row corresponds to a cluster.

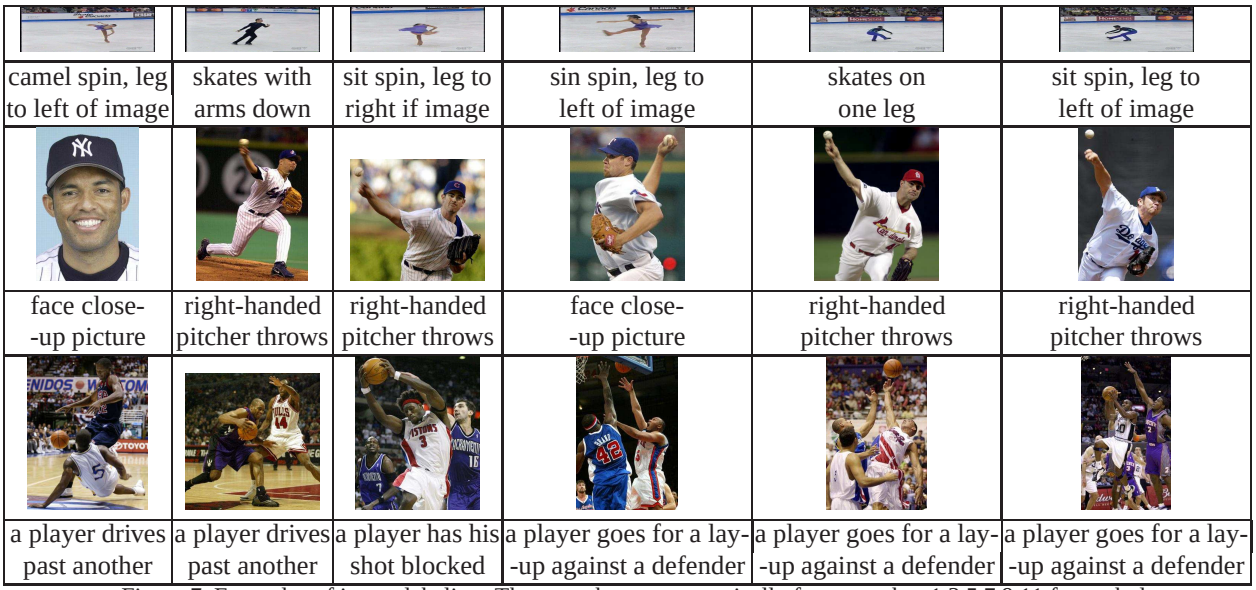

Figure 7. Examples of image labeling. They are chosen automatically from matches 1,3,5,7,9,11 for each dataset.

[10] J. J. Little and J. E. Boyd. Recognizing people by their gait: The shape of motion. *Videre*, 1(2), 1998.

[11] D. Moore, I. Essa, and M. Hayes. Exploiting human actions and object context for recognition tasks. In *Proc. 7th Int. Conf. Computer Vision*, 1999.

[12] G. Mori, S. Belongie, and J. Malik. Efficient shape matching using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(11):1832–1837, 2005.

[13] R. Polana and R. C. Nelson. Detection and recognition of periodic, non-rigid motion. *Int. Journal of Computer Vision*, 23(3):261–282, 1997.

[14] C. Rao, A. Yilmaz, and M. Shah. View-invariant representation and recognition of actions. *Int. Journal of Computer Vision*, 50(2), 2002.

[15] M. Shah and R. Jain. *Motion-Based Recognition*. Computational Imaging and Vision Series. Kluwer Academic Publishers, 1997.

[16] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Trans. PAMI*, 22(8):888–905, 2000.

[17] J. Sivic, B. C. Russell, A. A. Efros, A. Zisserman, and W. T. Freeman. Discovering objects and their location in images. In *Proc. 10th Int. Conf. Computer Vision*, 2005.

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

[19] T. Xiang and S. Gong. Video behaviour profiling and abnormality detection without manual labelling. In *Proc. 10th Int. Conf. Computer Vision*, 2005.

[20] H. Zhong, J. Shi, and M. Visontai. Detecting unusual activity in video. In *Proc. IEEE Comput. Soc. Conf. Comput. Vision and Pattern Recogn.*, 2004.