In IEEE International Conference on Computer Vision, Nice, France, October 2003

Recognizing Action at a Distance

Alexei A. Efros, Alexander C. Berg, Greg Mori, Jitendra Malik
Computer Science Division, UC Berkeley
Berkeley, CA 94720, USA
http://www.cs.berkeley.edu/~efros/research/action/

Abstract

Our goal is to recognize human actions at a distance, at resolutions where a whole person may be, say, 30 pixels tall. We introduce a novel motion descriptor based on optical flow measurements in a spatio-temporal volume for each stabilized human figure, and an associated similarity measure to be used in a nearest-neighbor framework. Making use of noisy optical flow measurements is the key challenge, which is addressed by treating optical flow not as precise pixel displacements, but rather as a spatial pattern of noisy measurements which are carefully smoothed and aggregated to form our spatio-temporal motion descriptor. To classify the action being performed by a human figure in a query sequence, we retrieve nearest neighbor(s) from a database of stored, annotated video sequences. We can also use these retrieved exemplars to transfer 2D/3D skeletons onto the figures in the query sequence, as well as two forms of data-based action synthesis "Do as I Do" and "Do as I Say". Results are demonstrated on ballet, tennis as well as football datasets.

1. Introduction

Consider video such as the wide angle shot of a football field seen in Figure 1. People can easily track individual players and recognize actions such as running, kicking, jumping etc. This is possible in spite of the fact that the resolution is not high – each player might be, say, just 30 pixels tall. How do we develop computer programs that can replicate this impressive human ability?

It is useful to contrast this medium resolution regime with two others: ones where the figures are an order of magnitude taller ("near" field), or an order of magnitude shorter ("far" field). In near field, we may have 300 pixel tall figures, and there is reasonable hope of being able to segment and label parts such as the limbs, torso, and head, and thus mark out a stick figure. Strategies such as [19, 12, 11] work best when we have data that support figures of this resolution. On the other hand, in far field, we might have only 3 pixel tall figures – in this case the best we can do is to track the figure as a "blob" without the ability to articulate the separate movements of the different locations in it. Blob tracking is good enough for applications such as measuring pedestrian traffic, but given that the only descriptor we can extract is the translation of the blob as a whole, we cannot expect to discriminate among too many action categories.

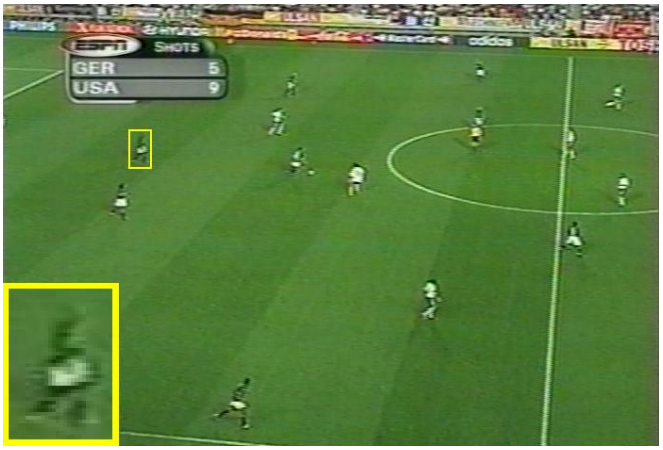


Figure 1. A typical frame from the NTSC World Cup broadcast video that we use as our data. Humans are extremely good at recognizing the actions of the football players, despite the low resolution (each figure is about 30 pixels tall; see the zoomed in player at the lower left corner).

In this paper, we develop a general approach to recognizing actions in "medium" field. Figure 2 shows a flow diagram. We start by tracking and stabilizing each human figure – conceptually this corresponds to perfect smooth pursuit movements in human vision or a skillful panning movement by a camera operator who keeps the moving figure in the center of the field of view. Any residual motion within the spatio-temporal volume is due to the relative motions of different body parts: limbs, head, torso etc. We will characterize this motion by a descriptor based on computing the optical flow, projecting it onto a number of motion channels, and blurring. Recognition is performed in a nearest neighbor framework. We have a stored database of previously seen (and labeled) action fragments, and by computing a spatio-temporal cross correlation we can find the one most similar to the motion descriptor of the query action fragment. The retrieved nearest neighbor(s) can be used for other applications than action recognition – we can transfer attached attributes such as appearance or 2D/3D skeletons

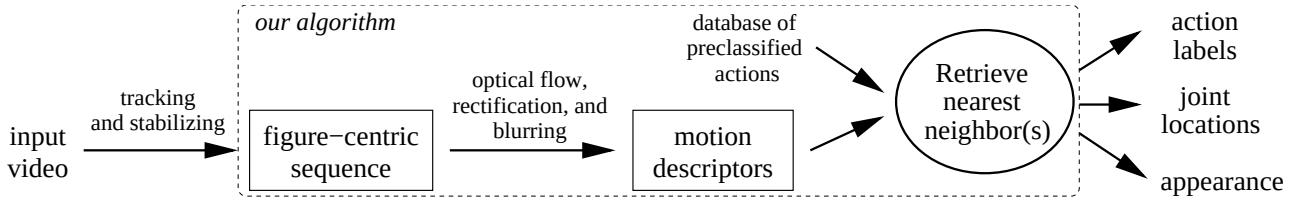


Figure 2. Data flow for our algorithm. Starting with a stabilized figure-centric motion sequence, we compute the spatio-temporal motion descriptor centered at each frame. The descriptors are then matched to a database of preclassified actions using the k-nearest-neighbor framework. The retrieved matches can be used to obtain the correct classification label, as well as other associated information.

from the action fragment in the database to the one in the query video sequence.

Note that we do *not* use the movement of the figure as a whole – the stabilization step intentionally throws away this information. In far field, this would in fact be the *only* information available for a moving figure blob, and it would certainly make sense for an integrated system for action recognition to capitalize on this cue. Our motivation is scientific – we want to understand the "extra" information available corresponding to relative motions among the different locations of the figure, just as one might ignore color to better understand the role of shape in object recognition. It may also be worth remarking that there are situations such as a person on a treadmill, or when the camera pans to keep an actor in the field of view, when the overall motion of the figure blob is unavailable or misleading.

This paper is organized as follows. Section 1.1 reviews related work. In Section 2, we develop the motion descriptor. This is the core of the paper – it is well known that optical flow measurements are noisy, so to be able to use them in a robust way for action matching is a fundamental contribution. Given the descriptor and matching technique, in Section 3 we show classification results on a variety of datasets – ballet, tennis, football. In Section 4, we show how the process of extracting best matching action fragments from the database has other side benefits. We are able to perform "skeleton transfer" on to the input figure sequence, as well as synthesize novel video sequences in two ways we call "Do as I do" or "Do as I say". We conclude in Section 5.

1.1 Related Work

This work addresses action recognition in "medium field" based on analyzing motion channels. As discussed above, most work in human tracking and activity recognition is only appropriate for "near field" with higher resolution figures. Shah and Jain [16] review the previous work on activity recognition, much of which involves tracking at the level of body parts. Gavrila and Davis' survey paper [7] provides a thorough review of the tracking literature, but it is largely inapplicable for the type of data we are considering in this work.

Another class of methods analyze motion periodicity [10, 15, 5, 4]. Of particular relevance is the work of Cutler and Davis [5], which is one of a few attempts at analyzing poor quality, non-stationary camera footage. Their approach is based on modeling the structure of the appearance self-similarity matrix and can handle very small objects. They report classification results on three categories: "person", "dog", "other". Unfortunately, methods based on periodicity are restricted to periodic motion.

Action classification can be performed in a nearest neighbor framework. Here the main challenge is to find the right representation for comparing novel data with stored examples. Bobick and Davis [3] derive the Temporal Template representation from background subtracted images. They present results on a variety of choreographed actions across different subjects and views, but require two stationary cameras with known angular interval, a stationary background, and a reasonably high-resolution video. Song et al. [17] demonstrate detection of walking and biking people using the spatial arrangement of moving point features. Freeman et al. [6] use image moments and orientation histograms of image gradients for interactive control in video games. Developing this theme, Zelnik-Manor and Irani [20] use marginal histograms of spatio-temporal gradients at several temporal scales to cluster and recognize video events. Despite its simplicity, this representation is surprisingly powerful. The paper reports promising action similarity results on three different datasets with 3-4 classes, assuming a single actor and a static background. However, since only the marginal information is being collected over each frame, the classes of actions that can be discriminated must have substantially different motion speed and orientation profiles.

2. Measuring Motion Similarity

Our algorithm (Figure 2) starts by computing a figure-centric spatio-temporal volume for each person. Such a representation can be obtained by tracking the human figure and then constructing a window in each frame centered at the figure (see Figure 3). Any of a number of trackers are appropriate; in our experiments, we used a simple normalized-correlation based tracker, either on raw video or on regions of interest selected by thresholding the tem-

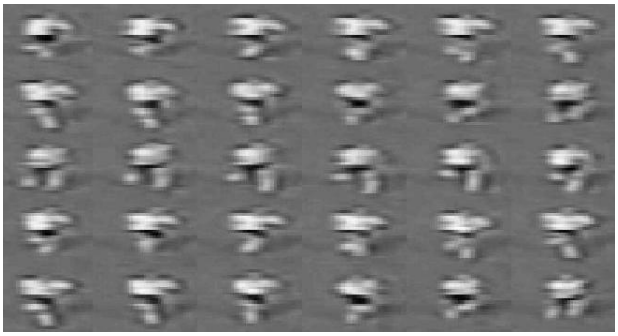


Figure 3. We track each player and recover a stabilized spatio-temporal volume, which is the only data used by our algorithm.

poral difference image. The main requirement is that the tracking be consistent – a person in a particular body configuration should always map to approximately the same stabilized image.

Once the motion sequences are stabilized it becomes possible to directly compare them in order to find correspondences. Finding similarity between different motions requires both spatial and temporal information. This leads to the notion of the *spatio-temporal motion descriptor*, an aggregate set of features sampled in space and time, that describe the motion over a local time period. Computing such motion descriptors centered at each frame will enable us to compare frames from different sequences based on local motion characteristics.

The important question is what are appropriate features to put into the motion descriptor. Encoding the actual image appearance by storing the pixel values directly is one possibility, which has been successfully used by Schödl el al. [14] to find similarity between parts of *the same* video sequence. However, appearance is not necessarily preserved across different sequences (e.g. people wearing different clothing). The same is true for spatial image gradients which depend linearly on image values. Temporal gradient is another useful feature [6, 20], but it shares the problems of spatial gradients in being a linear function of appearance. For example, temporal gradients exhibit contrast reversal: a light-dark edge moving right is indistinguishable from a dark-light edge moving left (taking the absolute value of the gradient will fix this but it will also remove all information about the direction of motion).

We base our features on pixel-wise optical flow as the most natural technique for capturing motion independent of appearance. In biological vision, neurons sensitive to direction and speed of retinal motion have been found in many different species. On the other hand, computer vision experience suggests that computation of optical flow is not very accurate, particularly on coarse and noisy data, such as typical NTSC video footage. Our insight is to treat these optical flow vectors not as precise pixel displacements at points, but simply as a spatial pattern of noisy measurements which

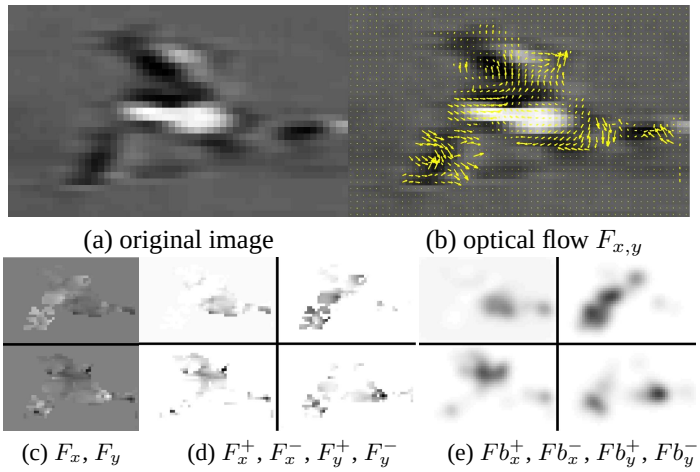


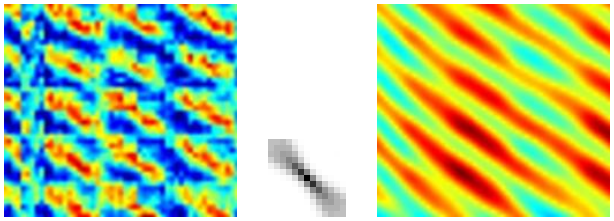
Figure 4. Constructing the motion descriptor. (a) Original image, (b) Optical flow, (c) Separating the x and y components of optical flow vectors, (d) Half-wave rectification of each component to produce 4 separate channels, (e) Final *blurry motion channels*

are aggregated using our motion descriptor. We think of the spatial arrangement of optical flow vectors as a template that is to be matched in a robust way.

The motion descriptor must perform reliably with features that are noisy, and moreover, be able to deal with input data that are not perfectly aligned either temporally or spatially. Matching under noise and positional uncertainty is often done using histograms of features over image regions [20, 13, 1]. Interestingly, a very similar effect can be obtained by simply blurring the input signal in the correct way [2]. This is a very simple yet powerful technique of capturing only the essential positional information while disregarding minor variations. However, one must be careful that important information in the signal is not lost due to blurring together of positive and negative components. In order to deal with this potential loss of discriminative information we use half-wave rectification, separating the signal into sparse, positive-only channels before it is blurred. In the primate visual system, one can think of each of these *blurred motion channels* as corresponding to a family of complex, direction selective cells tuned to roughly the same direction of retinal motion.

2.1 Computing Motion Descriptors

Given a stabilized figure-centric sequence, we first compute optical flow at each frame using the Lucas-Kanade [8] algorithm (see Figure 4(a,b)). The optical flow vector field $\mathbf{F}$ is first split into two scalar fields corresponding to the horizontal and vertical components of the flow, F_x and F_y, each of which is then half-wave rectified into four non-negative channels F_x^+, F_x^-, F_y^+, F_y^-, so that $F_x = F_x^+ - F_x^-$ and $F_y = F_y^+ - F_y^-$ (see Figure 4(c,d)). These are each blurred with a Gaussian and normalized to obtain the final four



(a) (b) (c)

Figure 5. (a) A typical frame-to-frame similarity matrix S_{ff} for running, (b) the "Blurry I" kernel K (not shown to scale) used for aggregating temporal information within the similarity matrix, (c) the resulting motion-to-motion similarity matrix S.

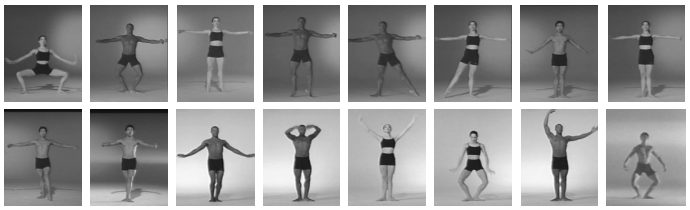


Figure 6. Representative frames from the sixteen ballet actions used for our experiments. The actions are (left to right): 1) 2^{nd} pos. plies, 2) 1^{st} pos. plies, 3) releve, 4) down from releve, 5) point toe and step right, 6) point toe and step left, 7) arms 1^{st} pos. to 2^{nd} pos., 8) rotate arms in 2^{nd} pos., 9) degage, 10) arms 1^{st} pos. forward and out to 2^{nd} pos., 11) arms circle, 12) arms 2^{nd} to high fifth, 13) arms high fifth to 1^{st}, 14) port de dras, 15) right arm from high fifth to right, 16) port de bra flowy arms

channels, $\hat{Fb}_x^+$, $\hat{Fb}_x^-$, $\hat{Fb}_y^+$, $\hat{Fb}_y^-$, of the motion descriptor for each frame (see Figure 4(e)). Alternative implementations of the basic idea could use more than 4 motion channels – the key aspect is that each channel be sparse and non-negative.

The spatio-temporal motion descriptors are compared using a version of normalized correlation. If the four motion channels for frame i of sequence A are a_1^i,a_2^i,a_3^i, and a_4^i, and similarly for frame j of sequence B then the similarity between motion descriptors centered at frames i and j is:

$$S(i,j) = \sum_{t \in T} \sum_{c=1}^{4} \sum_{x,y \in I} a_c^{i+t}(x,y) b_c^{j+t}(x,y) \qquad (1)$$

where T and I are the temporal and spatial extents of the motion descriptor respectively. To compare two sequences A and B, the similarity computation will need to be done for every frame of A and B so Eq. 1 can be optimized in the following way. First, a frame-to-frame similarity matrix of the blurry motion channels (the inner sums of the equation) is computed between each frame of A and B. Let us define matrix A_1 as the concatenation of a_1's for each frame stringed as column vectors, and similarly for the other 3 channels. Then the frame-to-frame similarity matrix $S_{ff} = A_1^T B_1 + A_2^T B_2 + A_3^T B_3 + A_4^T B_4$. To obtain the final motion-to-motion similarity matrix S, we sum up the frame-to-frame similarities over a T temporal window by convolution with a $T \times T$ identity matrix, thus $S = S_{ff} \star I_T$.

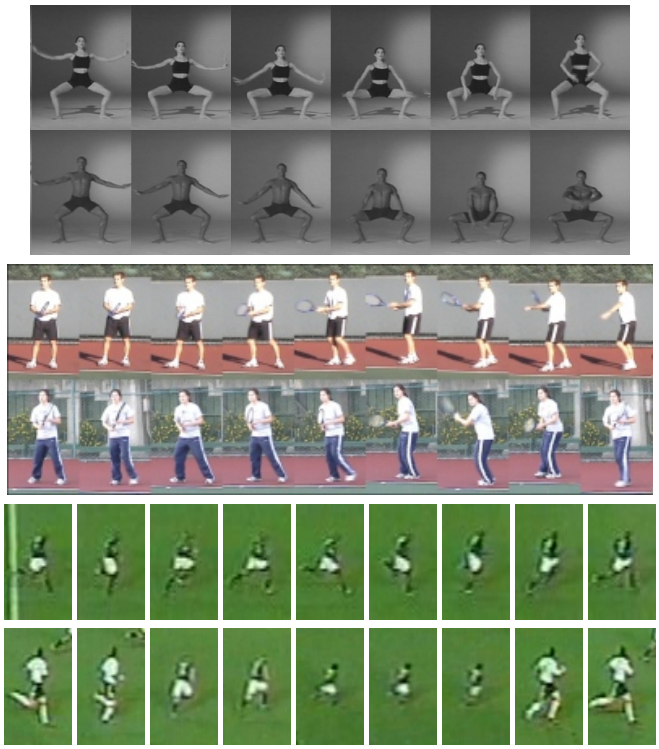
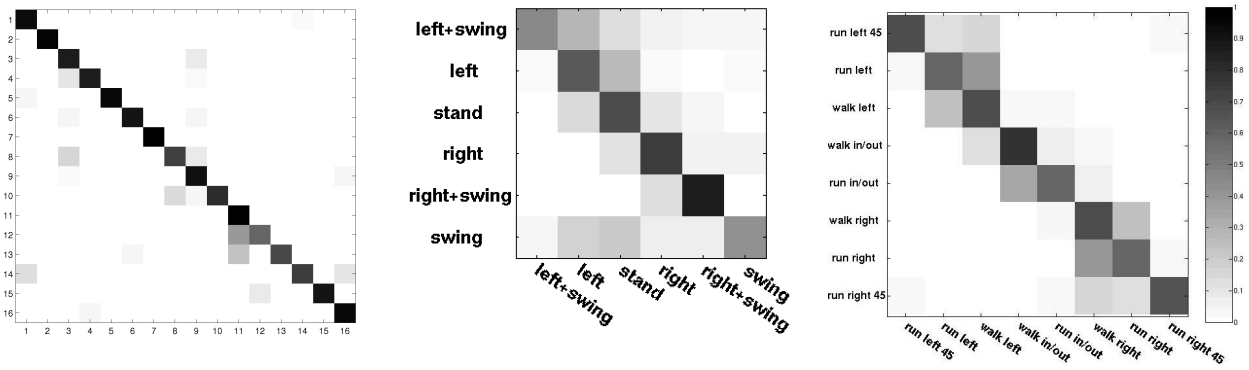


Figure 7. Best matches for classification (ballet, tennis, football). The top row of each set shows a sequence of input frames, the bottom row shows the best match for each of the frames. Our method is able to match between frames of people performing the same action yet with substantial difference in appearance.

If motions are similar, but occur at slightly different rates then the strong frame to frame similarities will occur along directions close to diagonal but somewhat slanted (note the angle of bands in Fig. 5a). In order to take advantage of this fact, we look for strong responses along directions close to diagonal in the frame to frame similarity matrix between A and B. In practice this is achieved by convolving the frame to frame similarity matrix S_{ff} with the kernel shown in Figure 5(b) instead of the identity matrix to obtain the final similarity matrix. The kernel is a weighted sum of near diagonal lines, with more weight put closer to the diagonal.

$$K(i,j) = \sum_{r \in R} w(r) \chi(i, rj) \qquad (2)$$

where $w(r)$ weights values of r near one relatively more, and R is the range of rates. (Note that we set $\chi(i, rj)$ to one if i and rj round to the same value and zero otherwise). The similarity between two sequences centered at two particular frames can be read from the corresponding entry in the final similarity matrix.



(a) Ballet (b) Tennis (c) Football

Figure 8. Confusion matrices for classification results. Each row represents the probabilities of that class being confused with all the other classes. **(a) Ballet dataset** (24800 frames). The 16 classes are defined in Figure 6. Video of the male dancers was used to classify the video of the female dancers and vice versa. Classification used 5-nearest-neighbors. The main diagonal shows the fraction of frames correctly classified for each class and is as follows: [.94 .97 .88 .88 .97 .91 1 .74 .92 .82 .99 .62 .71 .76 .92 .96]. The algorithm performs quite well on the ballet actions, matching or exceeding previous work in this area. However, the highly controlled, choreographed nature of the actions make this a relatively easy test. **(b) Tennis dataset.** The video was subsampled by a factor of four, rendering the figures approximately 50 pixels tall. Actions were hand-labeled with six labels: "swing", "move left", "move right", "move left and swing", "move right and swing", "stand". Video of the female tennis player (4610 frames) was used to classify the video of the male player (1805 frames). Classification used 5-nearest-neighbors. The main diagonal is: [.46 .64 .7 .76 .88 .42]. While the classification is not as good as in the previous experiment, the confusions make sense. For example, the "go left and swing" class gets confused with "go left". In addition some of the swing sequences are misclassified because optical flow occasionally misses the low contrast, motion blurred tennis racket. **(c) Football dataset** (4500 frames, taken from 72 tracked sequences, supplemented by mirror flipping some of the sequences). We hand-labeled subsequences with one of 8 actions: "run left $45°$", "run left", "walk left", "walk in/out", "run in/out", "walk right", "run right", and "run right $45°$". The classification used a 1-nearest-neighbor classifier on the entire data set with a leave-one-sequence-out testing scheme. The main diagonal is: [.67 .58 .68 .79 .59 .68 .58 .66]. The classes are sorted according to the direction of motion – confusion occurs mainly between very similar classes where inconsistent ground truth labeling occurs. There is virtually no confusion between very different classes, such as moving left, moving straight, and moving right. Here as with the tennis example the player's direction of motion is successfully recovered even though the algorithm uses *no translational information at all*. This means that the method correctly interprets the movement of human limbs without explicitly tracking them. The results are particularly impressive considering the very poor quality of the input data. Figure 7 shows nine consecutive frames from a "run right" sequence (top row) together with the best matching frames from the rest of the database (bottom row). Note that while the best matches come from different players with different appearance and scale, the motion is matched very well.

3. Classifying Actions

Given a novel sequence to be classified and a database of labeled example actions, we first construct a motion similarity matrix as outlined above. For each frame of the novel sequence, the maximum score in the corresponding row of this matrix will indicate the best match to the motion descriptor centered at this frame (see Figure 7). Now, classifying this frame using a k-nearest-neighbor classifier is simple: find the k best matches from labeled data and take the majority label.

We show results on three different domains:

Ballet: choreographed actions, stationary camera. Clips of motions were digitized from an instructional video for ballet showing professional dancers, two men and two women, performing mostly standard ballet moves. The motion descriptors were computed with 51 frames of temporal extent.

Tennis: real actions, stationary camera. For this experiment, we shot footage of two amateur tennis players outdoors. Each player was video-taped on different days in different locations with slightly different camera positions. Motion descriptors were computed with 7 frames of temporal extent.

Football: real actions, moving camera. We digitized several minutes of a World Cup football game (called *soccer* in the U.S.) from an NTSC video tape. We used wide-angle shots of the playing field, which have substantial camera motion and zoom (Figure 1). We take only the odd field of the interlaced video in grayscale, yielding, on average, about 30-by-30 noisy pixels per human figure. All motion descriptors were computed with 13 frames of temporal extent.

Figure 7 shows the best matches for each frame of some example sequences while Figure 8 shows the quantitative classification results in the form of confusion matrices.

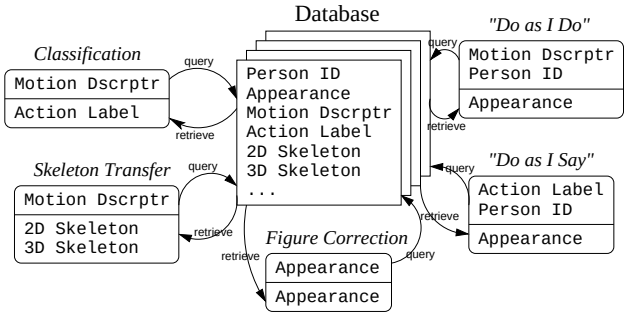


Figure 9. Action Database. Our classification algorithm can be interpreted as a database query: given a motion descriptor, retrieve the best matching action label. Other similar queries are possible, resulting in a number of useful applications, such as skeleton transfer, action synthesis, and figure correction. (This diagram does not show the temporal aspect of our action synthesis method.)

4. Querying the Action Database

The classification procedure described above can be thought of as a particular type of database query. Given a database of example sequences annotated with action labels, the classifier uses the motion descriptor as a key to query this database. The result of such a query is to retrieve the action label of the database entry with the most similar motion descriptor. We can also store additional fields in each of these database entries, facilitating several other applications (see Figure 9).

4.1 Skeleton Transfer

Recovering joint positions (i.e. the skeleton) of a human figure from video is an important and difficult problem. Most approaches rely on a person's appearance in each frame to identify limbs or other salient features (e.g. [9]). This will not work for our data – the figures are usually much too small to have identifiable parts at any given frame. Here, again, our solution is to rely on motion instead of appearance. The idea is simple: we annotate each frame in our database with hand-marked joint locations. This means that a novel sequence can now be automatically labeled with joint position markers, essentially transferring a 2D skeleton from the stored example onto the novel sequence (see Figure 10, second row). Note that since the motion descriptor is designed to be robust to misalignment, the skeleton transfered in this way may not be placed precisely on the figure. Hence we use a simple refinement step to better align the two sequences by searching for the scale and shift that maximizes the motion descriptor matching score.

An alternative to hand-marking the joint locations is to use available 3D motion capture data (produced in a lab using special markers) to *generate* a suitable database. We can render the MoCap data (using a stick figure) from several viewing directions to create a database of synthetic 2D motion sequences, fully annotated with the original 3D joint locations. Figure 10 shows how, given a video sequence (first row), we are able to recover a 3D skeleton (third row). Alternatively we could go to the 3D skeleton from the 2D skeleton, as in [18]. While lifting a 2D figure into 3D is clearly ambiguous (e.g. in side view, the left and right legs often get confused), nonetheless we believe that the information obtained this way is quite valuable.

4.2 Action Synthesis

The visual quality of our motion descriptor matching (see Figure 7) suggests that the method could be used in graphics for *action synthesis*, creating a novel video sequence of an actor by assembling frames of existing footage. The idea is in the spirit of Video Textures [14], except that we would like to have control over the actions that are being synthesized. The ultimate goal would be to collect a large database of, say, Charlie Chaplin footage and then be able to "direct" him in a new movie.

"Do as I Do" Synthesis. Given a "target" actor database T and a "driver" actor sequence D, the goal is to create a synthetic sequence S that contains the actor from T performing actions described by D. This problem can be posed as simple query: retrieve the frames from T associated with motion descriptors best matching those from D. However, this process alone will produce a video that is too jerky, since no smoothness constraint is present. In practice, the synthesized motion sequence S must satisfy two criteria: the actions in S must match the actions in the "driver" sequence D, and the "target" actor must appear natural when performing the sequence S. We pose this as an optimization problem.

Let $W_{act}(u, v)$ contain the motion descriptor similarity between frame u of D and frame v of T. A second matrix W_s is used to enforce the smoothness of the synthesized sequence. Let $W_s(u, v)$ hold the similarity in appearance (frame-to-frame normalized correlation) and in motion (motion descriptors) of frames u and v, both from the target database T. Since we are comparing frames from the *same* actor, we are able to use actual pixel values in the computation of the appearance term. We define the following cost function on S, a sequence of frames $\{\pi_1, \pi_2, ..., \pi_n\}$ picked from T:

$$C(S) = \sum_{i=1}^{n} \alpha_{act} W_{act}(i, \pi_i) + \sum_{i=1}^{n-1} \alpha_s W_s(\pi_{i+1}, \mathrm{succ}(\pi_i)),$$

where $\mathrm{succ}(\pi_i)$ is the frame that follows π_i in T. The cost function has only local terms, and therefore lends itself to being optimized using dynamic programming. A sequence of length n can be chosen from m frames in T in $O(nm^2)$ time. Figure 11 shows a few frames from our "Do as I Do" results. See the web page for our video results.

"Do as I Say" Synthesis. We can also synthesize a novel "target" actor sequence by simply issuing commands, or action labels, instead of using the "driver" actor. For example,

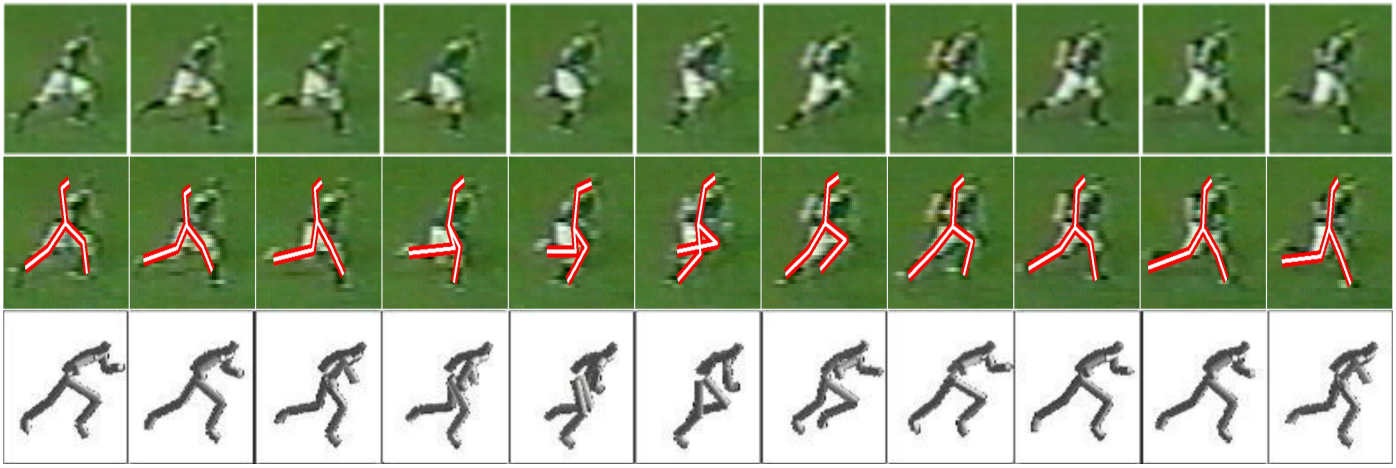


Figure 10. Skeleton Transfer. Given an input sequence (top row) we are able to recover rough joint locations by querying the action database and retrieving the best-matching motion with the associated 2D/3D skeleton. Second row shows a 2D skeleton transferred from a hand-marked database of joint locations. Third row demonstrates 3D skeleton transfer, which utilizes Motion Capture data rendered from different viewing directions using a stick figure.

Figure 11. "Do as I Do" Action Synthesis. The top row is a sequence of a "driver" actor, the bottom row is the synthesized sequence of the "target" actor (one of the authors) performing the action of the "driver".

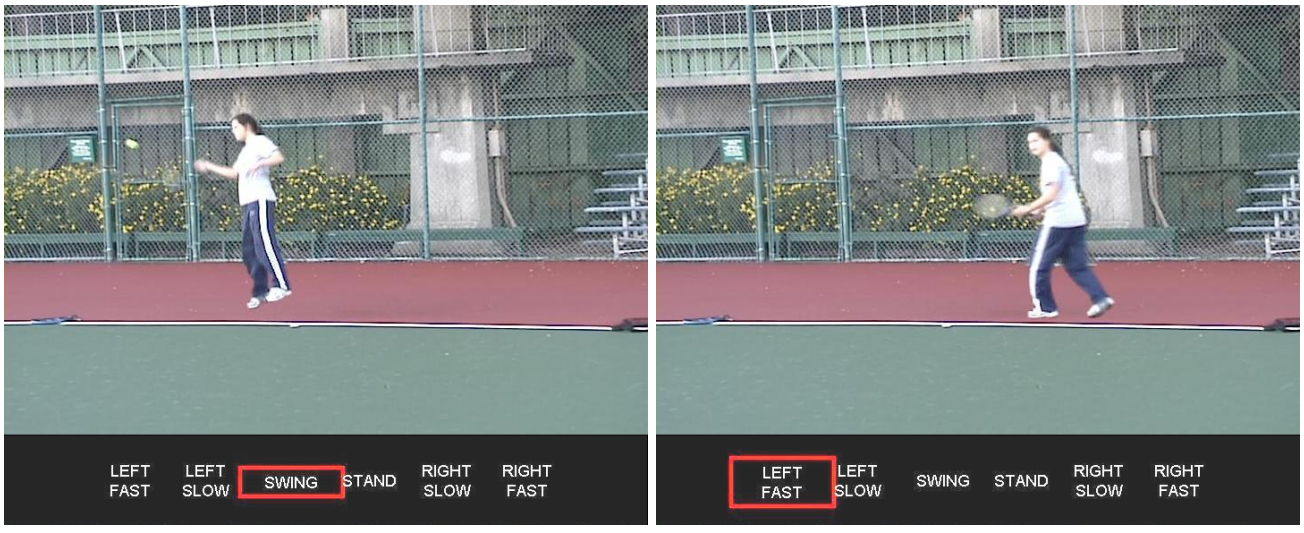


Figure 12. "Do As I Say" Action Synthesis. Shown are two frames from a synthesized video of a tennis player performing actions as specified by the commands (at the bottom). For the full video, visit our website.

Figure 13. Figure Correction. We use the power of our data to correct imperfections in each individual sample. The input frames (top row) are automatically corrected to produce cleaned up figures (bottom row).

one can imagine a video game where pressing the control buttons will make the real-life actor on the screen move in the appropriate way. The first step is to classify the "target" data T using our classification algorithm. Now the same approach can be used as in the previous section, except W_{act} now stores the similarity between the desired commands and the frames of T. Figure 12 shows two frames from a sequence where the tennis player is being controlled by user commands (shown at the bottom). Note that since dynamic programming is an off-line algorithm, this approach would not directly work for interactive applications, although there are several ways to remedy this.

Figure Correction. Another interesting use of the action database is to "clean up" human action sequences of artifacts such as occlusion and background clutter (see top row of Figure 13). The main idea, inspired by the dictionary-less spelling correction in search engines like Google, is to use the power of the data as a whole to correct imperfections in each particular sample. In our case, for each frame, we retrieve the k closest frames from the rest of the database (excluding the few neighboring frames that will always be similar). These k frames are used to compute a median image which becomes the new estimate for the current frame. The idea is that, given enough data, the common part among the nearest neighbors will be the figure, while the variations will be mostly due to noise and occlusions. The median filter averages out the variations thus removing most occluders and background clutter as shown on the bottom row of Figure 13.

5. Conclusion

Our primary contribution is a new *motion descriptor* based on smoothed and aggregated optical flow measurements over a spatio-temporal volume centered on a moving figure. We demonstrate the use of this descriptor, in a nearest-neighbor querying framework, to classify actions, transfer 2D/3D skeletons, as well as synthesize novel action sequences.

Acknowledgments. We thank Pooja Nath for implementing motion-based alignment, and Jane Yen and Chris Harrelson for playing tennis for us. This work was supported by ONR grant N00014-01-1-0890 (MURI) and NSF ITR IIS-00-85864 grant.

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] A. C. Berg and J. Malik. Geometric blur for template matching. In *Computer Vision and Pattern Recognition*, pages 607–614, 2001.

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

[4] R. Collins, R. Gross, and J. Shi. Silhouette-based human identification from body shape and gait. In *5th Intl. Conf. on Automatic Face and Gesture Recognition*, 2002.

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

[6] W. T. Freeman, K. Tanaka, J. Olita, and K. Kyuma. Computer vision for computer games. In *IEEE 2nd Intl. Conf. on Automatic Face and Gesture Recognition*, 1996.

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

[8] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *DARPA Image Understanding Workshop*, April 1981.

[9] G. Mori and J. Malik. Estimating human body configurations using shape context matching. In *European Conference on Computer Vision LNCS 2352*, 2002.

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

[11] D. Ramanan and D. A. Forsyth. Automatic annotation of everyday movements. Technical report, Tech. Rep. No. UCB//CSD-03-1262, Berkeley, CA, July 2003.

[12] C. Rao and M. Shah. View-invariance in action recognition. *CVPR*, 2:316–321, 2001.

[13] Henry Schneiderman and Takeo Kanade. A statistical method for 3d object detection applied to faces and cars. In *Proc. IEEE Conf. on Comp. Vision and Patt. Recog.*, volume 1, pages 746–751, 2000.

[14] Arno Schodl, Richard Szeliski, David H. Salesin, and Irfan Essa. Video textures. In *Proceedings of SIGGRAPH '00*, pages 489–498, 2000.

[15] S. M. Seitz and C. R. Dyer. View-invariant analysis of cyclic motion. *Int. Journal of Computer Vision*, 25(3), 1997.

[16] Mubarak Shah and Ramesh Jain. *Motion-Based Recognition*. Computational Imaging and Vision Series. Kluwer Academic Publishers, 1997.

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

[18] C. J. Taylor. Reconstruction of articulated objects from point correspondences in a single uncalibrated image. *CVIU*, 80:349–363, 2000.

[19] Y. Yacoob and M. J. Black. Parameterized modeling and recognition of activities. *Computer Vision and Image Understanding*, 73(2):232–247, 1999.

[20] Lihi Zelnik-Manor and Michal Irani. Event-based video analysis. In *CVPR*, 2001.