HUMAN MOVEMENT SUMMARIZATION AND DEPICTION FROM VIDEOS

Yijuan Lu[1] *and Hao Jiang*[2]
[1]Texas State University and [2] Boston College

ABSTRACT

Human movement summarization and depiction from videos is to automatically turn an input video into high level action illustrations, in which the movements of the body parts are visualized using arrows and motion particles. Motion depiction compactly illustrates how specific movements are performed. Previous action summarization methods reply on 3D motion capture or manually labeled data, without which depicting actions is a challenging task. In this paper, we propose a novel scheme to automatically summarize and depict human movements from 2D videos without 3D motion capture or manually labeled data. The proposed method first segments videos into sub-actions with an effective streamline matching scheme. Then, to estimate human movement, we propose a novel trajectory following method to track points by using both body part detection and optical flow. With the estimated movement, we depict the human articulated motion with arrows and motion particles. Our experiments on a variety of videos show that the proposed method is effective in summarizing complex human movements and generating compact depictions.

1 Introduction

Summarizing human movement in videos using a small set of static illustrations has many important applications. It is a valuable tool for the educational purpose to demonstrate how a specific movement can be performed. It also helps video browsing and provides compact representations for action recognition and movement analysis. Without 3D motion capture or manual labeling, high level action summarization that depicts the human body part movement is a difficult task. In this paper, we propose novel methods to automatically estimate human articulated motion and generate motion depictions from 2D videos without manually labeled data. A motion depiction example is shown in Fig.1.

In human movement summarization and depiction, we have to solve three basic problems: action segmentation (video segmentation into meaningful sub-actions), human movement estimation, and movement depiction. Action segmentation is to partition a complex action in a video into frame groups and in each group a simple sub-action occurs. We are most interested in segmenting input videos into sub-actions that reflect different movements of human body parts. Most previous research on action segmentation uses 3D motion capture data [1, 2, 5]. Movement segmentation with videos as a direct input is more challenging. Clustering based methods [3] for generic video segmentation can be applied to action segmentation. The downside of a clustering approach is that the number of clusters is hard to determine. Another widely used scheme is to directly detect the action boundaries. Rui et al. [4] propose to use PCA coefficients of dense optical flow to quantify the movement changes; the temporal curves of the features derived from the PCA coefficients are used to detect sub-action boundaries. In this paper, we follow the action boundary detection scheme. Our method uses a cluster of streamlines to capture the salient movement characteristics in action boundary detection.

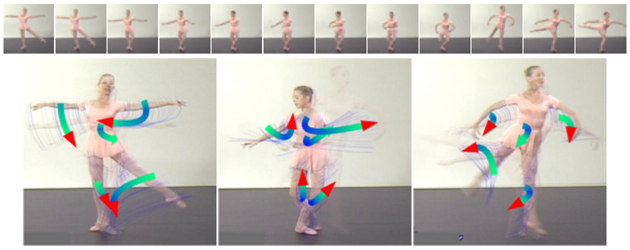

Fig. 1. Our method converts a video sequence to movement depictions, which illustrate body part movements using arrows and subtle local movements using motion particles.

In the second step, we extract the high level movement of a human subject. A high level movement representation has to reflect the body part movement and local subtle motion. To this end, we detect body parts and compute the motion trajectories of feature points. Finding feature point trajectories on human subjects has been studied in a multiple camera setting [9]. For single view videos, finding long trajectories is a hard problem. Simply propagating point location estimate from frame to frame using optical flow would cause the trajectory to drift in a long time span. Occlusions also make direct point tracking a difficult problem. In this paper, we merge body part detection, which can be obtained using methods in [6, 7, 8], and optical flow to achieve reliable results. Compared with previous human tracking methods [10], our scheme can be used to track feature points on human subjects in unconstrained movements. We propose an efficient multiple path optimization method to link body part detections in different video frames. The optimization explicitly models high order dynamics and can be efficiently solved using a linear method. The point cloud trajectory estimation is further formulated as an optimization problem in which we jointly find all the coupled trajectories constrained by the body part detection, optical flow and object foreground estimation.

In step three, motion depiction, we express the object movement in each segmented sub-action using a static illustration. Human movement depiction has been practiced in different artworks for centuries. Graphics elements, such as streamlines, motion blur, and overlapping semi-transparent ghost images have been used to illustrate actions. For computational motion depiction, the challenge is to translate human movement estimation into appropriate graphics representations. Our work is inspired by [2] which uses arrows, noise waves, and stroboscopic motion to depict stick figure movement. [2] uses 3D motion capture data. In contrast, our method does not reply on 3D motion capture or manual labeling; it automatically generates the illustration from a direct 2D video input. We use arrows to illustrate the body part movement, the motion particles to depict the subtle local motion, and ghost images to provide reference transitional and ending poses. In the following sections, we show how a convincing motion depiction can be achieved using the proposed method.

To our best knowledge, the proposed method is the first attempt that automatically converts a 2D video sequence to high level human movements depictions without 3D motion capture or manually labeled data. It is potentially capable of providing compact representations for action recognition and movement analysis. It can be used in many applications especially for education purpose to teach students, patients or people with disabilities how specific movements can be achieved.

2 Motion Summary and Depiction

Our method is composed of three steps: 1) Action segmentation: we segment complex actions into simple ones which can be depicted using directional arrows; 2) Human movement estimation: we detect human body parts and associate them through time. Then, we obtain rough human movement which will be refined for movement depiction. Finally, we augment the movement estimation into body point domain and clean up the error body part movement estimation; 3) Movement depiction: based on the cleaned up point motion estimation, we generate directional arrows to depict the human body part movements. The arrows are overlapped on the images to generate the final rendering results.

2.1 Action Segmentation

Action segmentation is to partition a complex action into simple sub-actions to facilitate movement depiction. We first directly detect the action boundaries and then use motion trajectories to quantify human movements. We randomly select seed points in each video frame and follow the motion field in a fixed time interval. The trajectories are constructed by connecting the points from one frame to the next using the motion vectors in a fixed time interval. In this paper, motion trajectories are computed in 15 frames. In such a simple scheme, there is no guarantee that the motion trajectories will not intersect. However, since we are only interested in the overall motion, the rough representation is sufficient.

After obtaining the motion trajectories starting from each frame and stretching a fixed time interval, we shift the trajectories so that they all start from point $(0, 0, 0)$, where the three coordinates are x, y and time. These clusters of motion trajectories at each frame reflect how the object moves in a small time interval.

To reduce the scale influence, the trajectories are further projected to the xy plane and the 2D coordinates of points on the curve are collapsed to form a normalized vector with unit length. The difference of movements is defined as the distance of these feature vectors. Let $F = \{\mathbf{v}_n, n = 1..N\}$ be the feature vectors for action one and $G = \{\mathbf{u}_m, m = 1..M\}$ be the vectors for action two. The distance d between F and G is defined as

$$d(F,G) = \frac{1}{N}\sum_n \min_m \operatorname{acos}(\mathbf{v}_n^T \mathbf{u}_m) + \frac{1}{M}\sum_m \min_n \operatorname{acos}(\mathbf{u}_m^T \mathbf{v}_n)$$

To detect movement boundaries, we require that the action features be stable when body parts keep their motion direction and the changes of the measurement should be proportional to the motion direction changes. The feature defined above fulfills the requirement.

In movement segmentation, we compute the distances of streamlines between successive time instants and form the results into a 1D curve. Local maxima on the curve indicate potential action changes. To avoid the detection of spurious local peaks, the distance curve is low pass filtered. With the robust streamline feature, the efficient approach achieves sufficient segmentation results for further action depiction.

2.2 Human Movement Estimation

Extracting human movement is a prerequisite for high level movement depiction. Apart from extracting feature point movements on a human subject, we would like to determine which body part each point belongs to. We devise a robust method to extract articulated motion by combining the global body part motion and local optical flow.

2.2.1 The Movement of Body Parts

We detect human body parts in each video frame and track them through time. We use [8] for human body part detection. We detect 10 body parts including head, torso, 4 half arms and 4 half legs as shown in Fig.2 Note that the detector does not distinguish the left and right arms and legs and there are many detection errors. We use the body part detections as a basis for body part tracking, i.e., we associate the corresponding body parts in successive video frames.

Based on the body part detection results, each limb that corresponds to an upper or lower body part has two possible locations in a video frame. We need to assign the two part detections to limb one and limb two in each video frame and

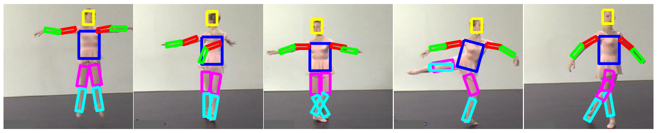

Fig. 2. Body part detection sample results.

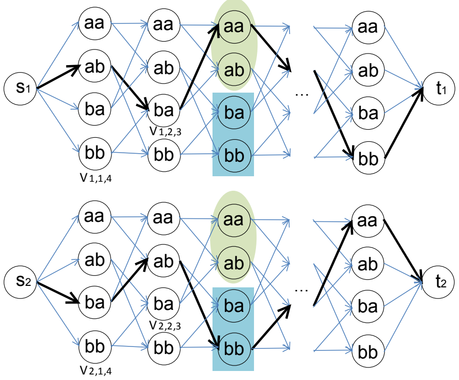

Fig. 3. Trellises for a pair of limbs. The path in each trellis corresponds to body part assignments through time; paths should not conflict.

we have to make sure that each body part moves smoothly in time and space. Unfortunately, naive exhaustive enumeration method has an exponential complexity; for n frames there will be 2^n possible assignments. Such a method cannot be used for body part association in hundreds and thousands of frames. We propose an efficient linear method to solve this problem. In this paper, body part association is formulated as a *multiple shortest path* following problem. The formulation is linear and can be solved efficiently. As follows, we will also illustrate how the second order smoothness constraint can be modeled by properly constructing the transition graph.

To optimize the body part association, we construct two graphs for each pair of limbs. Fig.3 shows two trellises corresponding to a pair of limbs. Each node of the trellises indicates a possible body part assignment. Except for the body part candidate nodes, source nodes and sink nodes are also included. At each layer, we have 4 possible body part assignments and each corresponds to a limb selecting one candidate in the current frame and one in the next frame. Note that each node indicates the assignments of body parts candidate assignment at two instants. Such a setting is necessary since we would like to introduce not only the first order, the position smoothness, but also the second order, the speed smoothness constraint.

We name the type of a node as `aa`, `ab`, `ba` or `bb`. For instance, an `ab` node indicates a limb selecting candidate one in the current frame and candidate two in the next frame; other types of nodes are similarly defined. We link the source nodes, candidate nodes and sink nodes into trellises. Fig.3 shows two trellises corresponding to a pair of arms or legs. Note that the edges between the nodes need follow the pattern of `xy` nodes connecting to `yz` nodes to enforce the consistency of body part assignments. Therefore not every node-node connection is valid. With the constructed graphs, body part association becomes the problem of finding an optimal path in each of the trellis.

As shown in Fig.3, the body part assignments to each limb correspond to a path that starts from the source node and ends in the sink node in each trellis. Each feasible path corresponds to a valid body part association and vice versa. Every path has different cost. The goal is to choose the minimum cost paths on all the trellises. What makes the problem complicated is that the paths are not independent: at each layer, there is at most one node that can be selected in a node conflict group. In Fig.3, the two green ovals in layer three form a conflict group; the other group in the same layer is indicated by two blue rectangles. Within each conflict group, there is at most one path passing. Each conflicting group corresponds to a spatial location that only one limb can be assigned to.

We formulate the problem in details. We introduce a node variable $\eta_{n,m,k}$. It is 1 if the node $v_{n,m,k}$, representing limb n's choice part candidate k in frame m, is on a path, and otherwise $\eta_{n,m,k}$ is 0. We also define the edge variable $\xi_{n,m,p,q}$, which is 1 if edge $(v_{n,m,p}, v_{n,m+1,q})$ is on a path and 0 otherwise. We would like to minimize the cost of paths

$$\sum_{(v_{n,m,p},v_{n,m+1,q})\in E} c_{n,m,p,q} \cdot \xi_{n,m,p,q}$$

where E is the edge set of the trellises; $c_{n,m,p,q}$ is the cost on the edge $(v_{n,m,p}, v_{n,m+1,q})$: for non-source and non-sink edges. We define the cost c on each edge as

$$\begin{aligned} c_{n,m,p,q} = \quad & ||\mathbf{u}^a_{n,m,p} - \mathbf{u}^a_{n,m+1,q}|| + \\ & ||2\mathbf{u}^b_{n,m,p} - \mathbf{u}^b_{n,m+1,q} - \mathbf{u}^a_{n,m,p}|| \end{aligned} \tag{1}$$

and c is 0 for source and sink edges. Recall that each node is related to two body part candidates and has a type `xy`. In Eq.1, $\mathbf{u}^a_{n,m,p}$ is the end point vector corresponding to the first body part candidate for node $v_{n,m,p}$; and $\mathbf{u}^b_{n,m,p}$ is the second vector. c is therefore composed of both first order and second order smoothness terms, which enforces position and speed continuity.

ξ follows the flow continuity condition for each trellis:

$$\sum_k \xi_{n,m-1,k,p} = \sum_q \xi_{n,m,p,q} \ .$$

And the flow from each source node should be 1. This condition makes sure the solution is a path on a trellis. To constrain the paths so that they do not conflict, we introduce a node variable η that is related to edge variable ξ by

$$\eta_{n,m,p} = \sum_q \xi_{n,m,p,q} \ .$$

To enforce that paths do not conflict, we introduce constraints:

$$\sum_{v_{n,m,p}\in Q_{m,i}} \eta_{n,m,p} \leq 1, i = 1, 2$$

where $Q_{m,i}$ is the ith conflict node set in frame m. Each conflict set corresponds to a possible body part location in each video frame. This constraint prevents two body parts from being assigned to the same place in one video frame.

Combining everything together, we obtain the following integer linear program:

$$\begin{aligned} & \min\{ \sum_{(v_{n,m,p},v_{n,m+1,q})\in E} c_{n,m,p,q}\cdot\xi_{n,m,p,q}\} \\ s.t. \quad & \sum_k \xi_{n,m-1,k,p} = \sum_q \xi_{n,m,p,q}, \ \sum_l \xi_{s,m_s,n,l} = 1 \\ & \eta_{n,m,p} = \sum_q \xi_{n,m,p,q}, \ n = 1,2 \\ & \sum_{v_{n,m,p}\in Q_{m,i}} \eta_{n,m,p} \leq 1, \ i = 1,2 \end{aligned}$$

where s is the source node and m_s is a single dummy candidate of the source node; V is the node set of the trellises. This integer linear program can be efficiently solved using a relaxation method followed by a rounding procedure to force solutions to be integers. In fact, the relaxed linear program always yields integer solution and therefore achieves global optimum directly. Using the simplex method, we can compute the body part association in thousands of frames in few seconds.

2.2.2 Finding Point Trajectories

The body part association finds the rough locations of body parts in each video frame. However, body part foreshortening and local deformations have not been addressed. Body parts also may have large estimation errors due to the errors in the initial detections. In the following, we study how to correct errors and extract more detailed point trajectories using both body part detection and short term optical flow estimation.

We randomly select points on the object in the first video frame. Each point traverses the spatial and temporal volume and plots a trajectory. We require that the trajectories be controlled by both body part detections and optical flow: each trajectory fits the local motion estimation in the tangent direction; the point following a trajectory moves smoothly in space and with a smoothly changing speed; it complies with body part detection and stays inside the object foreground. The body part tracking result presents a long term movement of body points; however there are often errors. The optical flow presents short term movement of body parts that are usually accurate in a short time span. By merging these two estimations, we can achieve more robust results. Moreover, there is a global constraint that the points on trajectories also act on each other so that their topologies should be consistent at each time instant.

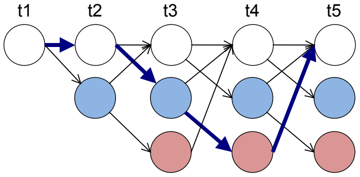

Fig. 4. Error correction trellis. The white nodes indicate point locations determined by part detections. The blue nodes represent predicted candidates from the previous part detections using optical flow. The red nodes represent the predictions from the previous predictions. In this example, two errors at time 3 and 4 are skipped by the "blue" path in the graph.

Before we optimize trajectories, we estimate initial body point trajectories to correct gross body part detection errors. We use a dynamic programming approach. At each instant, apart from the point locations determined by the body part detection, we include the point candidates predicted from point locations in previous frames. The basic idea is if there is a wrong large jump of point from one frame to the next, the prediction of the current point using optical flow should be used as the location estimate in the next frame. As illustrated in Fig.4, we use nodes of a graph to indicate point locations and the edges to indicate possible transitions. Apart from the point locations estimated by body part detection, the candidate locations also include the predicted locations using optical flow. The graph therefore provides alternative routes to bypass the wrong point estimations. The weight on each edge equals the distance of the points associated with the incidence nodes. The optimal point locations through time correspond to the shortest path from the first layer to the last layer of the graph. It can be solved using dynamic programming. By introducing more prediction steps, this method can be used to correct multiple errors.

After estimating the initial locations for all the points, we optimize all the point locations over all the image frames by minimizing the following energy:

$$\sum_{n=2}^{N-1}\{||\mathbf{x}_{n,k}+\mathbf{f}(\mathbf{x}_{n,k})-\mathbf{x}_{n+1,k}||^2+\lambda_1||\mathbf{x}_{n-1,k}+\mathbf{x}_{n+1,k}-$$
$$2\mathbf{x}_{n,k}||^2+\lambda_2\sum_{m\in\mathcal{N}(k)}||\mathbf{x}_{n,k}-\mathbf{x}_{n,m}-\mathbf{x}_{n+1,k}+\mathbf{x}_{n+1,m}||^2+$$
$$\lambda_3||\mathbf{x}_{n,k}-\mathbf{x}^0_{n,k}||^2+\lambda_4 g(\mathbf{x}_{n,k})\}$$

where N is the number of frames; $||.||$ is the L_2 norm; $\mathbf{x}_{n,k}$ is the intersection point of trajectory k with video frame n; $\mathbf{f}(\mathbf{x}_{n,k})$ is the motion vector at point k in frame n; $\mathbf{x}^0_{n,k}$ is the initial estimate of the trajectory k, obtained by dynamic programming; $\mathcal{N}(k)$ is the set of points that are the neighbors of point k. A point is defined as a neighbor of point k if the Delaunay triangulation of the point set in the first video frame has an edge connecting the point to point k. g is a function that penalizes trajectories deviating from the object foreground. In this paper, g is the Gaussian filtered distance transform of the object foreground. g is an optional term; it is set to zero when the foreground map is unavailable. $\lambda_1, \lambda_2, \lambda_3, \lambda_4$

are constant coefficients.

We use a gradient descent method to solve the optimization. $\mathbf{x}_{n,k}$ is updated with the following rule:

$$\begin{aligned}\mathbf{x}^{t+1}_{n,k} = \mathbf{x}^{t}_{n,k} - \delta((\mathbf{x}^{t}_{n,k} + \mathbf{f}(\mathbf{x}^{t}_{n,k}) - \mathbf{x}^{t}_{n+1,k})+ \\ \lambda_1(6\mathbf{x}^{t}_{n,k} - 4\mathbf{x}^{t}_{n-1,k} - 4\mathbf{x}^{t}_{n+1,k} + \mathbf{x}^{t}_{n-2,k} + \mathbf{x}^{t}_{n+2,k})+ \\ \lambda_2 \sum_{m \in \mathcal{N}(k)} (2\mathbf{x}^{t}_{n,k} - 2\mathbf{x}^{t}_{n,m} - \mathbf{x}^{t}_{n+1,k} + \mathbf{x}^{t}_{n+1,m} - \\ \mathbf{x}^{t}_{n-1,k} + \mathbf{x}^{t}_{n-1,m}) + \lambda_3(\mathbf{x}^{t}_{n,k} - \mathbf{x}^{0}_{n,k}) + \lambda_4 \nabla g(\mathbf{x}^{t}_{n,k}))\end{aligned}$$

where δ is a small constant. We use about 1000 iterations to obtain the trajectory clusters for hundreds of frames.

2.3 Movement Depiction

With the extracted articulated motion, we are ready for movement depiction. We construct a single image for each sub-action and use graphics components such as arrows and motion particles to illustrate the body part movements.

We use arrows to illustrate the movements of torso, arms and legs. From the cluster of trajectories of a body part, we compute the mean trajectory and use it as the center line of the arrow. However, the mean trajectory may still have errors. To solve this problem, we fit each trajectory in a sub-action to a second-order polynomial. These low order polynomials are sufficient to quantify the shapes of the trajectories in sub-actions and to further remove the gross motion errors. The width of an arrow is pre-defined while the brightness at each point on the arrow is proportional to the speed of the corresponding point on the body part. The color on the arrows is important to illustrate the coordination of different body parts. To reduce clutter, only arrows with enough length are kept. Apart from the arrows, we scatter particles on the trajectories of limbs to depict the detailed movements. Semi-transparent ending frame and intermediate frame are also overlapped on the depiction to show pose transition.

3 Experimental Results

We test the proposed motion depiction method on two ballet sequences and two recorded videos. These videos contain complex movements and some have strong clutter. It is a great challenge to summarize and depict the human movements in these videos.

Fig.5 (row 1-3) shows our results on the ballet-I sequence. The motion segmentation curve and the action boundary detection results are shown in the second row. The proposed method extracts the long trajectories of feature points on each body part as shown in the second row of Fig.5. The motion depiction results are shown in row 2-3. The illustrations clearly show the movements of the subject. The spin actions are also well illustrated. The brightness of the arrows represents the speed of the corresponding body part: the brighter the color, the faster it is at a specific instant. The blue motion particles illustrate the subtle local motion.

The results of motion depictions for another longer ballet sequence are shown in Fig.5 (row 5-8). This sequence contains complex body part movement and self-occlusion. The proposed method illustrates these movements using a compact set of static images. Fig.5 (row 10) shows another result for the girl fitness sequence which contains fast motion and the video is shot with a shaky hand held camera. The proposed video segmentation, motion extraction and depiction method still work robustly. The proposed method can also deal with cluttered videos as demonstrated in Fig.5 (row 12).

4 Conclusion

In this paper, we propose an automatic method to generate human movement depictions using 2D videos as direct input without 3D motion capture and manually labeled data. The proposed method segments human movements into sub-actions by streamline matching. We propose a novel trajectory following method to track points on human body based on both body part detection and optical flow. An efficient linear method is used to optimize the part association; a dynamic programming approach is proposed for error correction; and a gradient descent method is used to optimize all the coupled trajectories simultaneously. Based on the extracted articulated motion, we depict the high level body part movement using color coded arrows and detailed movement using motion particles. Our experiments on a variety of videos show that the proposed action depiction method is efficient, effective and robust against complex movement, fast action, camera motion and cluttered background.

Acknowledgment This research is supported by the United States NSF funding 1018641, 1058724 and Army Research Office grant W911NF-12-1-0057.

5 References

[1] J. Assa, Y. Caspi and D. Cohen-or, “Action synopsis: pose selection and illustration”, SIGGRAPH 2005.

[2] S. Bouvier-Zappa, V. Ostromoukhov and P. Poulin, “Motion cues for illustration of skeletal motion capture data”, Symposium on Non-Photorealistic Animation and Rendering 2007.

[3] Y. Gong and X. Liu, “Video summarization using singular value decomposition”, CVPR 2000.

[4] Y. Rui and P. Anandan, “Segmenting visual actions based on spatio-temporal motion patterns”, CVPR 2000.

[5] J. Barbic, A. Safonova J. Pan, C. Faloutsos, J.K. Hodgins and N.S. Pollard, “Segmenting motion capture data into distinct behaviors”, ACM Graphics Interface 2004.

[6] D. Ramanan, “Learning to parse images of articulated objects”, NIPS 2006.

[7] P. Felzenszwalb and D. Huttenlocher, “Pictorial structures for object recognition”, IJCV, v.61, n.1, 2005.

[8] H. Jiang, “Human pose estimation using consistent max-covering”, ICCV 2009.

[9] K. Varanasi, A. Zaharescu, E. Boyer and R.P. Horaud, “Temporal surface tracking using mesh evolution”, ECCV 2008.

[10] R. Urtasun, D. Fleet and P. Fua, “Temporal motion models for monocular and multiview 3D human body tracking”, CVIU, vol.104, no.2, pp. 157-177, 2006.

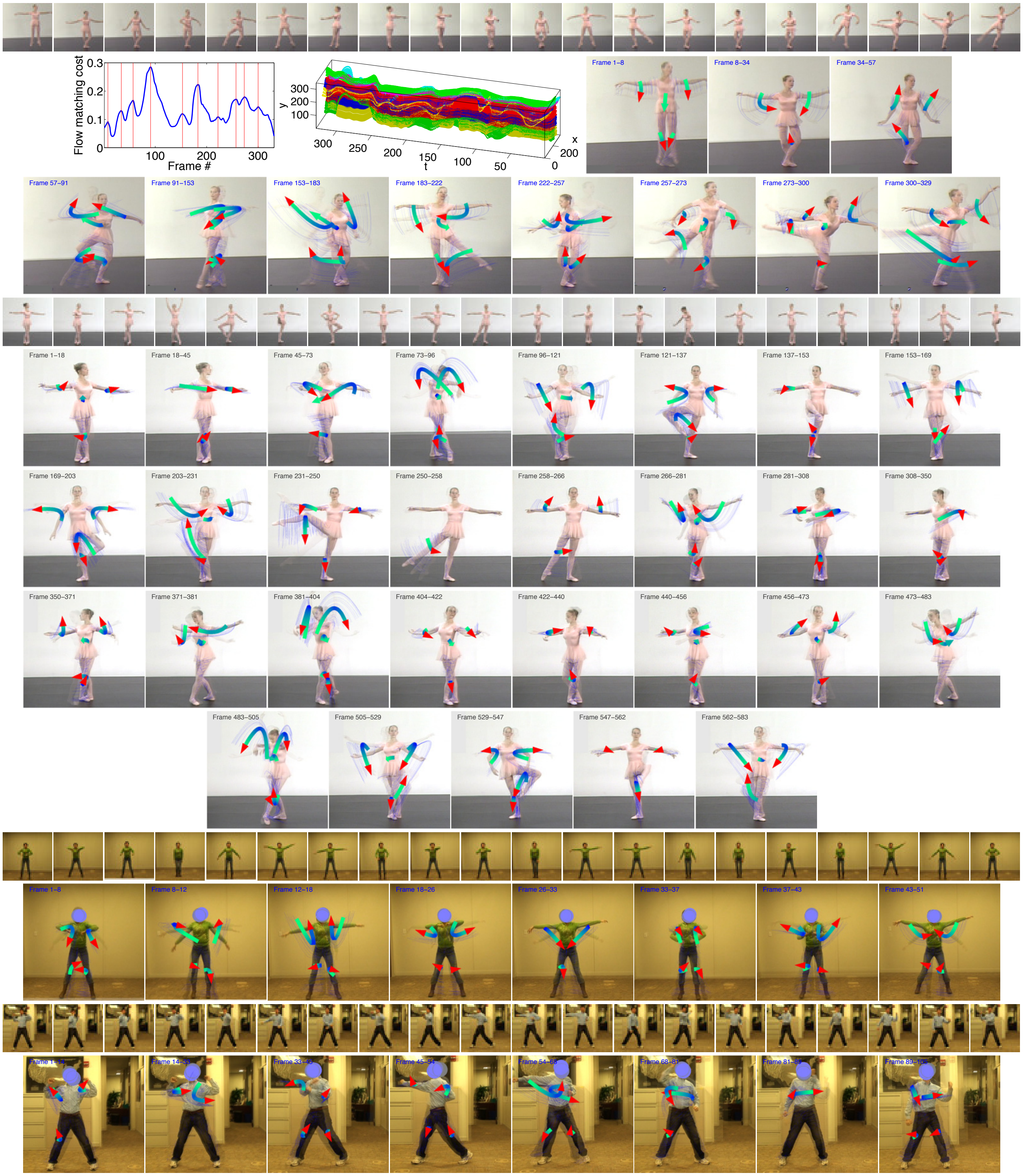

Fig. 5. Motion depiction results on the ballet-I (row 1-3), ballet-2 (row 4-8), girl (row 9-10) and man (row 11-12) sequences. The video segmentation curve and the body point trajectories for ballet-I are shown in the 2nd row. With the proposed method, the 329-frame ballet-I, 583-frame ballet-II, 51-frame girl and 105-frame man sequences have been compactly depicted as small number of images.