Eurographics/ ACM SIGGRAPH Symposium on Computer Animation (2007), pp. 1–3
D. Metaxas and J. Popovic (Editors)

Responsive Video-Based Motion Synthesis

C. Johnson and G. Mori

Vision and Media Lab, School of Computing Science, Simon Fraser University, Canada

1. Introduction

In this research we present a method for synthesizing videos of human motion. The synthetic videos are produced by splicing together clips of input video. The main challenge we address in this work is allowing for responsive, high-level control of the figure by specifying actions to be performed.

Synthesizing videos of human motion directly from existing clips is an alluring goal. Ignoring the possible pitfalls, the prospect of controlling realistic looking characters performing accurately portrayed actions with properly deforming clothing is very appealing.

2. Motion Graphs

We use a motion graph to define which sequences of motions are valid for building new videos. This graph will also include bunches of clips for each motion type so that the same sequence of motions can produce two different looking videos.

We build two similar motion graphs for each video domain. The first is a higher level "motion type graph" that explicitly states how the motions are structured. The second is a lower level "video clip graph" that states specifically which video frames belong to which higher level nodes, and how these clips can be rendered together without producing noticeable visual artifacts. The motion type graph represents the types of motions available and how they may be composed together. Each type of motion is represented by a node in the graph. A directed edge is placed from node i to node j if motion j can follow motion i. This graph is manually defined. We choose what types of motions we want to include in the data set and then film a video with an actor performing those motions. The video clip graph represents the actual clips of video in the input video. Each video clip is represented by a node. Each directed edge represents a transition and indicates which specific clips can be rendered sequentially. There will be no edge between a node i and another node j in the video clip graph that if edge (i, j) does not exist between the corresponding nodes in the motion type graph.

3. Building the Graph

We need to populate the video clip graph with examples from the input video. We automatically find the video clips using activity recognition. We label an "exemplar" clip for each type of motion and then automatically search the video for other similar clips based off those exemplars. Once we have labeled an exemplar ε we compare each frame in ε to every other frame in the video. The comparison we use is based off of optical flow channels [EBMM03].

We run the optical flow algorithm to get values of where the pixels have moved from one frame to the next. The Euclidean distance is calculated between these sets of values to get a distance cost. We record all of these values into a similarity matrix and then search for diagonals of low values as these will indicate a range of frames that have a high similarity to our exemplar clip. We label clips at the indices from these diagonals that have a value below a specified threshold. The threshold we choose has to be low enough so that we are retrieving clips which actually do appear similar to the exemplar, and high enough so that we are retrieving as many of these clips as possible. We run this process to get a collection of clips for each exemplar. We start building the video clip graph by creating a node for each clip. We then fill in the edges in the graph based off of a cost measure for a given transition between two clips a and b. The cost is computed based on the Euclidean distance between the last frames in clip a and the first frames in clip b. We attempt to measure the quality of a transition by this cost. The lower the cost, the more likely the transition between a and b will appear realistic. Once we have calculated the cost for each possible transition we create an edge in the graph for any transition below a set threshold cost. Any transition with a cost higher than this value likely will not look realistic when rendered. The threshold must be chosen carefully to allow enough transitions to populate the graph while still achieving a level of realism in the rendered videos.

To ensure that we can render any possible ordering of motions we must ensure that each node in the video clip graph has an outgoing edge to at least one clip for each type of

Figure 1: *Set of frames from one of our data sets in an output video. The same figure is scaled and rendered in four different sequences, overlapping each other*

motion that the current node's type has an outgoing edge to in the motion type graph. We must remove any node that does not have this property and any incoming and outgoing edges connected to that node. Removing a node from the video clip graph may remove the last outgoing edge from another node. We have to recheck the graph after we remove a node to make sure that every clip is still linked to the motion types it requires. After this pruning is completed we are assured that we will be able to render a new video with any sequence of motions.

4. Image Matting

We need to separate the figure from the background in every frame so that we can freely translate the figure on any given background. We take each frame in the video and perform alpha-matting to separate the figure from the background. We use a closed form solution [LLW06] to automate this process. The input trimap for the method of Levin et al. is manually constructed, but we need to construct thousands of trimaps. To accomplish this we begin by generating binary images of the figures' silhouettes using background subtraction. We perform morphological operations on the silhouette to construct a trimap for the frame. Any foreground pixel that is within a chosen distance to a background pixel is labeled as unknown. Similarly with background pixels within the distance to a foreground pixel. Giving these pixels the label of unknown in the trimap allows the closed form solution to determine for us their classification.

The output of the image matting is an image providing the colour of the foreground object at each pixel and a corresponding alpha matte. The silhouette and trimap are both automatically generated. This reduction in manual labour allows for long videos to be processed for image matting.

5. Results

The two motion graphs allow us to generate a new realistic video based on clips from the input video. For controlled videos we specify a sequence to be rendered and then build a video based off of that sequence. Rendering one video clip after another clip will produce some jittering when viewed as a video. To combat these irregularities we perform a cross-fade and alignment between two clips. To ensure that the figure moves on a new background realistically, we record the vertical and horizontal translations between the found center of the figure between two frames and translate the figure in the new video by these relative amounts.

We applied our technique to the domain of a person walking back and forth with some turns included. We have rendered a video based off the walking motion graphs that shows that it is possible to recreate realistic looking motions from existing video. Figure 1 shows some frames from this video. We filmed another subject playing tennis on a tennis court performing five separate motions. We rendered a video of the figure performing a sequence of motions. The main visual artifact in this video is the racket sometimes moving suddenly between transitions. The alignment we perform does not correct this. The racket would need to be held in the same place at the beginning and ending of each clip. We chose one additional domain. An actor was filmed in front of a green screen strumming a guitar. The optical flow descriptor works on this video for recognizing similar strum actions, but because the hand on the bridge of the guitar is not moving, all strum actions are categorized the same. Once we have automatically found a set of clips for a strumming type of motion, we can look at them to see if the hand of the bridge is in the desired position. If not, we can manually exclude that clip from the set.

6. Conclusion

In this paper we have demonstrated a method for creating responsive, controllable animations directly from input video. Our method builds a motion graph representation in order to allow the generation of this responsive animation. The construction of this graph involves minimal user interaction, only a set of "exemplar" video clips need to be specified. After this, computer vision techniques for action recognition are used to automatically find similar motions, which are then automatically matted and aligned for rendering.

References

[EBMM03] EFROS A., BERG A., MORI G., MALIK J.: Recognizing action at a distance. In *Proc. 9th Int. Conf. Computer Vision* (2003), vol. 2, pp. 726–733.

[LLW06] LEVIN A., LISCHINSKI D., WEISS Y.: A closed form solution to natural image matting. *Proc. IEEE Comput. Soc. Conf. Comput. Vision and Pattern Recogn.* (2006).