MotionMontage: A System to Annotate and Composite Motion Takes for 3D Animations

Ankit Gupta [*], **Maneesh Agrawala** [†], **Brian Curless** [*], **and Michael Cohen** [§]
[*]University of Washington [†]University of California, Berkeley [§]Microsoft Research, Redmond
{ankit,curless}@cs.washington.edu maneesh@cs.berkeley.edu mcohen@microsoft.com

ABSTRACT
We present *MotionMontage*, a system for recording multiple motion takes of a rigid virtual object and compositing them together into a montage. Our system incorporates a Kinect-based performance capture setup that allows animators to create 3D animations by tracking the motion of a rigid physical object and mapping it in realtime onto a virtual object. The animator then temporally annotates the best parts of each take. MotionMontage merges the annotated motions into a single composite montage using a combination of dynamic time warping and optimization of a Semi-Markov Conditional Random Field. Our system also supports the creation of layered animations in which multiple objects are moving at the same time. To aid the animator in coordinating the motions of the objects we provide spatial markers which indicate the positions of previously recorded objects at user-specified points in time. We perform a user study to evaluate the perceived quality of the montages created with our system and find that viewers (including both the original animators and new viewers) generally prefer the animation montage to any individual take.

Author Keywords
Active visual feedback; Depth camera; Animation; Montage;

ACM Classification Keywords
H.5.2. Information Interfaces and Presentation: User Interfaces; I.3.6. Computer Graphics: Methodology and Techniques

INTRODUCTION
It is common practice in the movie industry to capture multiple takes of the same shot. Such re-takes allow the director to capture variations in the dialogue, and try out different positioning of the actors and camera angles. In performance-based 3D animation, a director may ask the animator to perform multiple takes in order to test different styles of motion (e.g. a more energetic performance versus a somber performance) and to find the most appropriate style for the shot.

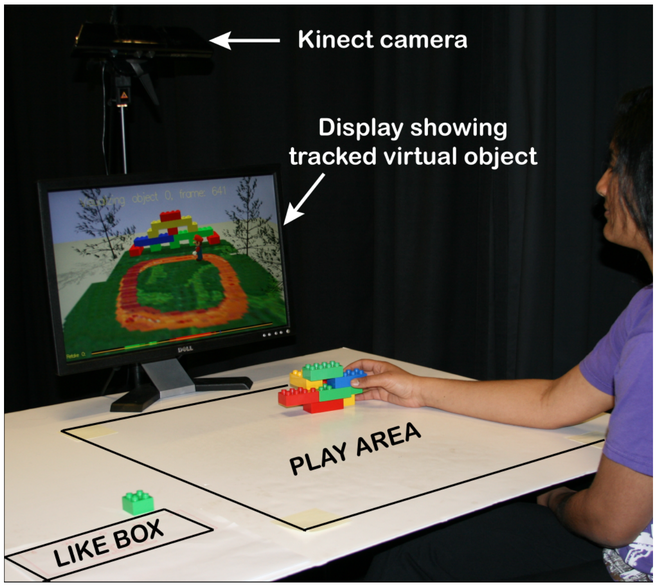

Figure 1. A user working with the MotionMontage system to record multiple takes of a 3D animation.

In film, there are instances of a single shot requiring many takes; for example, one shot in the movie "The Shining" was reported to have needed 148 takes before the director, Stanley Kubrick, was satisfied [13].

Our work explores the concept of multiple takes for performance-based 3D animation. We focus on novice users who do not have any experience with high-end animation editing software such as Maya. We leverage 3D Puppetry [9] and DuploTrack [8], two recent Kinect-based performance capture systems that allow novices to easily create 3D animations by tracking the motions of rigid physical toys and mapping those motions in real-time onto virtual objects. However, one significant drawback of all such performance-based 3D animation systems is that they require the user to discard a take if any aspect of the motion was incorrect or undesired and then repeat the motion from scratch, hoping the next one will be just right. Yet, each take may contain some parts that are better than others even if no complete take is satisfactory. Since most novice users have little prior experience in creating animated stories, they may require a large number of takes to reach a desired animation, leading to fatigue and frustration in generating such re-takes.

We present *MotionMontage*, a system that allows novice animators to combine multiple takes into a desired result, called the *montage*. The animator records the motion of one object at a time, as shown in Figure 1. The takes may vary in style or trajectory of the motion. Our system allows the animator to annotate each take continuously through time based on their like or dislike of various parts of the take. The system then merges the best parts of each annotated take into a single composite montage using a combination of dynamic time warping and optimization of a Semi-Markov Conditional Random Field. The user can repeat the process for the same object to further refine the montage or to animate other objects.

Our system also allows animators to create layered animations in which multiple objects are moving at the same time. Although the animator must perform the motion of one object at a time, our system plays back all of the motions together. To aid the animator in coordinating the motions of the different objects, we provide spatial markers indicating the positions of previously recorded objects at user-specified points in time. The animator can place multiple markers for one or more objects to help plan the motion of the current object.

The key contribution of the paper is our technique for combining multiple takes of an object's motion into a motion montage using a simple annotation-based interface. We report on a formal user study that validates that the animation montage is generally perceived to be better than any individual take, both from the point of view of lay users who created the animations, as well as others who only viewed the animations.

RELATED WORK

Our system for combining multiple takes of performance-based 3D animations builds on several areas of prior work.

Performance-Based Animation Capture

In film, high-end motion capture systems are often used to capture performance-based animations. These systems are designed to accurately track human bodies or objects, but usually require complex hardware setups and/or attaching markers to the tracked objects [21, 6]. More recently, researchers have begun developing low-cost marker-less performance capture systems that work with video cameras or 3D cameras. Video puppetry [4] tracks the motions of paper cutouts using an off-the-shelf video camera and allows users to create 2.5D animations. Researchers have similarly used the Kinect 3D camera to track the human body [20], hands [20], as well as rigid physical objects [9, 8]. However all these methods require the user to repeatedly record motion takes till she is happy with one take. However the user may like different parts of different takes and there is no easy mechanism in these systems to intelligently combine takes based on user feedback. Our MotionMontage system presents a simple interface to annotate the recorded takes and uses a novel algorithm to combine these takes into the montage. Our system leverages the 3D object tracking algorithm of DuploTrack [8] as well as the performance-capture interface of 3D Puppetry [9] to record the takes.

Motion Synthesis and Blending

There have been several efforts to synthesize new motions by applying constraints or leveraging sets of motion capture recordings. Gleicher et al. [7] propose a formulation for synthesizing a motion sequence respecting a set of spacetime constraints. They pose it as a global optimization over a high-dimensional motion space. The optimization is non-convex and grows in complexity as the number of constraints increases. Our problem could also be modeled as this kind of global optimization with the spatial constraints coming from users' annotations and temporal constraints coming from an alignment method like Dynamic Time Warping [14]. Instead, we exploit the structure of our problem to propose a much simpler solution which leads to a one-dimensional optimization that can be solved in realtime. Further, unlike Gleicher et al. [7], our method is guaranteed to closely follow the original takes and the annotations.

Arikan et al. [3] use a database of motion sequences (captured using standard motion capture systems) to synthesize a new motion sequence. Each frame in the motion sequences is first tagged with actions like walk, run, jump etc. The user then specifies the desired action tags on a timeline and the system computes an optimal sequence of motion frames satisfying the tags. Their algorithm is based on representing the motion frames as a complete graph and then finding an optimal path in that graph using dynamic programming [12]. In our case, the motion sequences are semantically bound to a script and also vary in length of time. A complete motion graph would include a lot of transitions that violate the temporal and semantic structure of the sequences. Instead, we use temporal warping to align all the motion takes and then work with a graph that respects the temporal and semantic ordering of the frames. Further, the montage in our graph is non-Markovian in contrast to Arikan et al. due to an additional path constraint that we found necessary for editing the takes. Hence we propose a different solution in this paper based on performing an inference in a semi-Markov Conditional Random Field (CRF). We also visualize and describe these differences in the supplementary video.

Kovar et al. [11] look at the problem of blending multiple, annotated, human motion capture clips. They propose techniques for temporally and spatially aligning the clips and then apply a per-frame weighted average to blend them. We have found that taking weighted averages of existing takes gives undesirable results. For example, if a user happens to give high weight to two distinct motions that overlap in time, then the average at their overlap will not resemble either of the original motions. Instead, we focus on piecing together sections from the original takes, respecting user annotations to the extent possible, while finding good places to transition between the takes. Our work also differs in how we warp the user's annotations and unwarp the montage at the end. Our approach better preserves the original intent of the user and the original speeds of the recordings.

Interactive Compositing of Photos and Videos

Our work on compositing multiple motion takes is inspired by Agarwala et al.'s [2] *Interactive Digital Photomontage* sys-

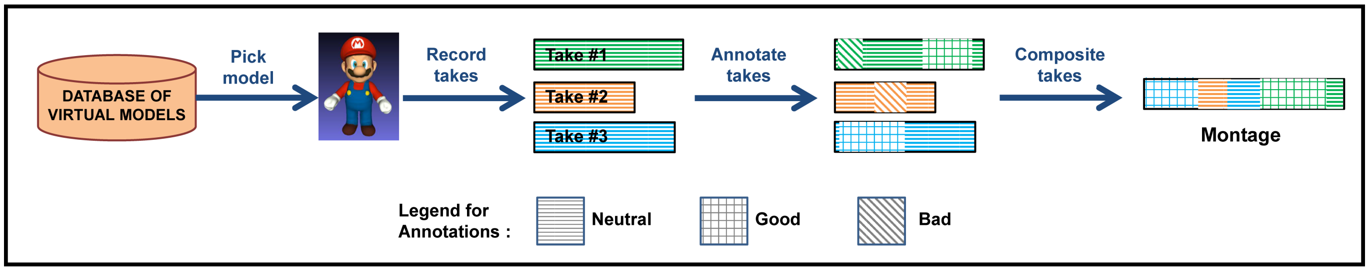

Figure 2. System functionality. The user chooses a virtual object from a pre-loaded database and records one or more motion takes for the object. He then annotates the takes and the system combines them into a montage, which can be longer than any of the original takes. The process can be repeated for subsequent objects.

tem for combining the best parts of a set of photographs. In that work, the user roughly annotates the parts of each photograph that are desired in the composite and the system uses a combination of graph-cut optimization [5] with gradient domain blending [15] to automatically generate the composite image. Ruegg et al.'s [17] *DuctTake* system extends the approach of Agarwala et al. [2] to compositing multiple videos. Our system similarly allows users to annotate the desired parts of each motion take, and automatically combines them into a motion montage. Since we are working with 3D motion data rather than images or video we use very different methods for optimizing the final composite.

SYSTEM SETUP AND INTERACTION

Our MotionMontage system allows the user to record 3D animations involving multiple objects, one at a time. We first discuss animating a single object, and then discuss differences in this procedure for animating multiple objects.

Figure 2 gives an overview of the system. The user first chooses a virtual object from a pre-loaded database. He then records one or more takes for the object, based on a script, by moving a physical object, called a *proxy*, in the 3D space in front of him. The tracked motion of the proxy is mapped in realtime onto a virtual object on a screen. The next step is to annotate each take. The user marks the parts of the take he likes and those he dislikes. The figure shows three discrete levels of annotations although there is a continuous scale. After annotating the takes, the system combines the best parts of each one into a montage animation.

The user can then repeat the process to animate additional virtual objects, and the system combines the montages of all objects to create the final animation. In the remainder of this section, we describe the details of the hardware setup and the various interactions described above. The supplementary video shows the effects of all these interactions in detail.

Hardware Setup

Figure 1 shows the hardware setup of our system. The interaction area is divided into two regions - a *Play area* and a *Like box*. The user acts out the story using the proxy in the Play area. The Like box is used as a slider by moving a green block inside it. The screen in front of the user shows the rendered virtual object in the tracked pose of the proxy. This hardware setup is based on Gupta et al.'s [8] DuploTrack system.

Setting up the Scene, Proxy and Virtual Objects

The user first chooses a virtual background scene. The background scenes are created by placing freely available 3D objects, downloaded from the internet. The play area on the table is mapped to an area in the background scene as in 3D Puppetry [9].

Unlike in 3D Puppetry, the user can use any rigid physical object as the proxy, as long as a model of the proxy is available. In our system, the user constructs the proxy with Duplo® blocks using the DuploTrack system [8], which then provides the model of the proxy to be used in tracking. The proxy can then be used to control the motion of a selected virtual object again chosen from a set of rigid virtual objects downloaded from online 3D databases. When recording a take, the system uses the color+depth image feed from a Kinect camera, segments it to remove the irrelevant pixels and performs real-time pose tracking of the proxy. The tracked motion is mapped one-to-one to the virtual object and rendered on the screen in front of the user in realtime. We have adapted the segmentation and tracking algorithms from DuploTrack [8]. There is a slight lag in the tracking currently, as seen in the supplementary video. In the future, this could be addressed by using a faster tracking algorithm, such as the one developed for 3D Puppetry [9], which uses both image features and 3D geometry to perform realtime object tracking.

Recording a Take

The user starts recording a take by pressing a key on the keyboard. He then acts out the motion with the proxy and ends the recording by pressing a key again. The user can record multiple takes with this process. These takes may vary in length. However, the assumption is that all the takes follow the same template story, and vary primarily in motion style. Arrow keys allow the user to scroll through the takes and review them.

Annotating the Takes

The user annotates each take to indicate their level of satisfaction with each part of each take. Annotations are made in realtime – while the take is playing the user moves a green Duplo block left or right in the Like box to indicate the degree of like or dislike for the corresponding part of the take. The Like box acts as a slider, ranging from -5 to $+5$, with the block serving as the thumb of the slider. We track the position of the block using the same tracking algorithm as for the proxy. The user annotates the whole take in one play-through.

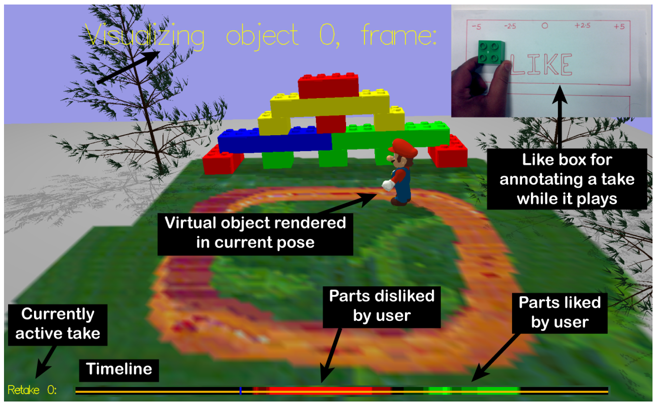
Visualizing object 0, frame:
LIKE
Like box for annotating a take while it plays
Virtual object rendered in current pose
Currently active take
Parts disliked by user
Parts liked by user
Timeline
Retake 0:

Figure 3. Annotating a take. While the take plays on the screen, the user moves a green block in the Like box (top right). The system localizes the position of the block and annotates the take with the corresponding slider value. Disliked and liked parts of the timeline are illustrated with red and green, respectively. The degree of like or dislike, is reflected by the brightness of the color.

Figure 3 shows a screenshot of the user annotating a take with this process.

When annotating a take, the user sees a motion, judges it, and then physically moves the green Duplo block. There is a natural time delay between the actual time of the motion and the placement of the green block in its intended position. To compensate for this lag, we shift the recorded annotations back in time by 0.3 seconds. This value is based on the average human reaction time [1] plus an empirically added delay for block movement. We found that this value worked well in practice for all our users.

The user repeats the annotation process for all the takes that he wants to composite for the montage. He then triggers the montage creation by pressing a key and the system generates the montage in realtime.

COMPOSITING THE TAKES INTO THE MONTAGE

In this section, we discuss the formulation for merging m annotated motion takes into a single motion recording called a *montage*. We denote the l^{th} take as T_l and its annotation as a function a_l where $T_l(t)$ and $a_l(t)$ give the pose and annotation values respectively, at frame t. The like-dislike annotation values lie on a continuous scale of -5 (dislike) to 5 (like). We further decompose the pose $T_l(t)$ into a rotation quaternion and a translation vector, $T_l(t) = (q_l(t), x_l(t))$.

The goal is to create a montage that utilizes the best parts of each take without introducing noticeable artifacts due to frequent switching between takes. The latter constraints mean we should transition between takes where the different takes agree as much as possible, and we should also avoid frequently flipping from one take to another.

To determine the best set of switching points from one take to another, we first align all the takes in time. We temporally warp all the takes independently to match a single reference timeline. After warping, specific segments of each take are selected to create the montage on this timeline. Finally, we unwarp the timeline so that each segment will be played at its original speed, and then blend between adjacent segments over short intervals to create the final montage. We note that unwarping the timeline is important to respect the user's intentions with respect to timing.

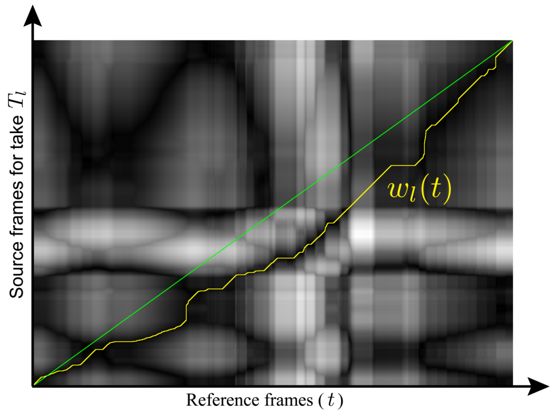

Figure 4. Dynamic Time Warping (DTW) finds an optimal warp function, $w_l(t)$, from the frames of the reference take, T_1 (x-axis), to the frames in the source take, T_l (y-axis). The values at each point in the quadrant represent the match of pose between the corresponding frames in the reference and the source takes. Darker color shows better pose match. The yellow path is optimal warp function computed by the algorithm while the green path shows the naive linear-scaling-based warp.

Temporal Warp of the Takes

We use a Dynamic Time Warping (DTW) [14] algorithm to align all the takes in time. For simplicity, we arbitrarily choose the first take as the reference, and warp all other takes to its timeline.

Formally, we consider take T_1 to be the *reference* take and then compute a monotonically increasing function w_l from the reference take's timeline to the timeline of each *source* take T_l. We can write each warp function w_l as a mapping,

$$w_l : \{1, ..., ||T_1||\} \rightarrow \{1, ..., ||T_l||\} \tag{1}$$

which is monotonically increasing,

$$t_1 > t_2 \Rightarrow w_l(t_1) \geq w_l(t_2) \tag{2}$$

and matches the start and end frames of reference and source,

$$w_l(1) = 1 \;\text{ and }\; w_l(||T_1||) = ||T_l|| \tag{3}$$

We solve for an optimal w_l by minimizing the cost of bringing each reference frame into alignment with a source frame summed over all frames t:

$$w_l^* = \underset{w_l}{\text{argmin}} \sum_t C_{\text{match}} \left(T_1(t), T_l(w_l(t)) \right) \tag{4}$$

The matching cost C_{match} measures the difference between two poses. For two poses $A_1 = (q_1, x_1)$ and $A_2 = (q_2, x_2)$:

$$C_{\text{match}}(A_1, A_2) = ||q_1 - q_2||_2 + \lambda||x_1 - x_2||_2 \tag{5}$$

We set λ to 2.5, based on experimentation.

Figure 4 shows a visualization of the matching cost and the optimal w_l. We use the standard DTW algorithm [14] to solve this problem. Graphically, it finds a monotonically increasing path that tries to stay in the darker regions of the diagram in Figure 4. The resulting path, $w_l(t)$ lets us warp $T_l(t)$ to a new warped take, $T'_l(t) = T_l(w_l(t))$ that follows the reference timeline.

We also need to warp the respective annotations to the reference timeline. This requires special care. Consider the case where $w_l(t)$ is constant for a duration of $N_{w_l(t)}$ frames. This corresponds to a horizontal, flat region of the curve in Figure 4. In this case, the annotation $a_l(w_l(t))$, which applies to only one frame in the source take, will have an influence over $N_{w_l(t)}$ frames in the warped space. As a result, during later optimization over the warped timeline (described in the next section), this annotation will have extra influence because it is artificially sustained by the warp. In this case, we should down-weight the annotation by $1/N_{w_l(t)}$ for each frame in the set that maps to $w_l(t)$.

Conversely, where the slope of $w_l(t)$ is steep, the warped sequence can skip from one source frame to a distant next source frame when moving from frame t to $t+1$ in the reference timeline and thereby skip past all source annotations in between. To ensure that the annotations on such skipped influence on the final result we add all of their influences together and store the sum as the warped annotation.

Formally, we compute the warped annotations as follows:

$$a'_l(t) = \begin{cases} \displaystyle\sum_{t'=w_l(t)}^{w_l(t+1)-1} a_l(t'), & \text{if } w_l(t+1) - w_l(t) > 0 \\ \frac{1}{N_{w_l(t)}} a_l(w_l(t)), & \text{if } w_l(t+1) = w_l(t) \end{cases} \tag{6}$$

Another way of thinking about this is that we are (intelligently) scaling the annotation values so that the summation of the warped annotations $a'_l(t)$ over $||T_1||$ frames of the reference timeline is the same as the summation of the unwarped annotations $a_l(t)$ over $||T_l||$ frames of the (unwarped) source timeline. We have found this scaling to significantly improve the quality of results over naively warping the annotation functions.

Merging the Takes

We now have m warped, annotated takes T'_l's, each comprised of $n = ||T_1||$ frames. We first compute a warped montage, M', such that each frame of the montage comes from a corresponding frame in one of the m takes. M' can be denoted in shorthand as $\{l_1, l_2 ... l_n\}$ where $l_t \in \{1, 2...m\}$ is the take selected for time $t \in \{1, 2...n\}$. (Strictly speaking, the montage is a sequence of poses $\{T'_{l_1}(1), T'_{l_2}(2), ... T'_{l_n}(n)\}$.) We now define the overall cost function for a montage:

$$C_T(M') = \sum_{t=1}^{n} C_d(t, l_t) + \mu \sum_{t=1}^{n-1} C_s(t, l_t, l_{t+1}) \tag{7}$$

The total cost C_T is the sum of two terms: a data cost, C_d, and a smoothness cost C_s. We set $\mu = 100$ based on experimentation.

C_d is the cost of frame t coming from take l_t and is based on the annotation functions, where a higher annotation rating translates to lower cost. Specifically, for a given frame t and take l we define this cost to be:

$$C_d(t, l) = -a'_l(t) \tag{8}$$

The smoothness cost C_s favors temporal coherence by discouraging transitions between takes in regions where their poses are dissimilar. At frame t, it is computed by considering time windows in the takes T_{l_t} and $T_{l_{t+1}}$, centered around the unwarped location of t, and summing up the cost of matching the poses in those windows. Specifically, given frame t and given two takes l and k, we define the cost to be:

$$C_s(t, l, k) = \sum_{h \in \mathcal{W}} C_{\text{match}}\left(T_l(w_l(t) + h), T_k(w_k(t+1) + h - 1) \right) \tag{9}$$

Note that we are comparing poses of the unwarped takes, as the final composite will be comprised of sequences of unwarped takes between which we want smooth transitions. We set the window $\mathcal{W}$ to the range $[-7, ..., 7]$, suitably truncated when the window goes out of bounds in either of the two takes. C_{match} is the same cost function as in Equation 5. Here we choose $\lambda = 0.8$ based on experimentation.

We can solve for the optimal warped montage, M'^*, by minimizing the total cost:

$$M'^* = \operatorname*{argmin}_{M'} C_T(M') \tag{10}$$

This formulation is equivalent to an inference problem over a Markov chain, which can be solved exactly using the dynamic-programming-based Max-Sum algorithm [10].

However, we have found that optimizing this objective, despite the smoothness term, can occasionally lead to sections of the montage that have rapid, frequent flipping between annotated takes. To address this problem, we impose a minimum length on contiguous frames with the same label in the montage.

Specifically, we can rewrite the warped montage M' as a sequence of subsequences, $\{M'_1, M'_2, ...\}$, such that each subsequence of frames or *segment* $M'_j = \{l, l, ..., l\}$ comes from a single take l. We then impose the constraint that the length of each segment M'_j must be greater than a threshold d, set to 60 frames in our experiments. We hereon refer to this constraint as the *Segment Length Constraint* and discuss its quantitative and qualitative effect in the results of the user study.

The formulation for the optimal montage M'^* now becomes,

$$M'^* = \operatorname*{argmin}_{M'} C_T(M') \tag{11}$$

$$s.t. \qquad ||M'_j|| > d, \qquad \forall j \tag{12}$$

This formulation is equivalent to a Semi-Markov Conditional Random field that can be solved exactly with another dynamic-programming-based algorithm [18]. The computational complexity of the algorithm is $O(n^2 m^2)$ where again n is the number of frames and m is the number of takes.

Unwarping the Montage

We now have an optimal warped montage M' of length n frames where the frames come from the warped takes. We next unwarp the montage to preserve the original speed at which the takes were recorded. We consider the span of frames for each segment M'_j in the warped montage and replace it with the corresponding interval of frames, M_j from the original take. Each corresponding interval is determined by the corresponding mapping function w_l. Thus, the unwarped montage, M, is a sequence of intervals of poses from the original takes T_l's.

Motion Blending

As a last step, we blend the poses around the transitions between takes in M to ensure temporal coherence. We consider a window of length 15 frames centered at each transition point and blend the poses using a linear weighting scheme. For blending, we use linear interpolation of the translation vectors and spherical linear interpolation of the rotation quaternions [19].

CREATING MONTAGES FOR MORE THAN ONE OBJECT

When animating more than one object, the user can record the takes of the first object at his preferred speed. However, the motion of any new object typically have to be synchronized with those of a previous object. Hence after recording the first object, the motions of all previous objects are played in realtime on-screen as the new object's motion is recorded. This approach is similar to the layered animation recording approach of 3D Puppetry [9]. In the end, all the takes for new objects are of the same length. They can then be annotated and composited into the montage by the same process as described above, skipping the time warping stage.

Recording motion for a current object while the previously recorded motions of other objects play on the screen can be difficult since the user has to plan the current object's motion based on where other objects will be in future. For example, if the two objects are supposed to touch each other, they may end up passing by each other without touching or perhaps intersecting each other.

We provide *spatial markers* to alleviate this problem. The user adds a spatial marker for an object by navigating to a frame in the timeline of the object's montage and pressing a key. The spatial marker appears as a grey version of the object in that pose of the montage. It is also marked on the timeline as a pink bar. Figure 5 shows the usage of spatial markers in a sample script. Please see the supplementary video to see the use of the markers in action. To avoid visual clutter, we do not allow two markers for the same object to be placed within 60 frames of each other. We chose this empirically. The user can remove the markers by clicking on them and pressing a key. Spatial markers can serve as a sparsely sampled motion trail for an object where the user chooses the sampling.

SYSTEM PERFORMANCE

The system runs in realtime on a desktop PC with two 6-core 3.33GHz Xeon processors and uses at most 400MB of RAM. To achieve this performance, the implementation is

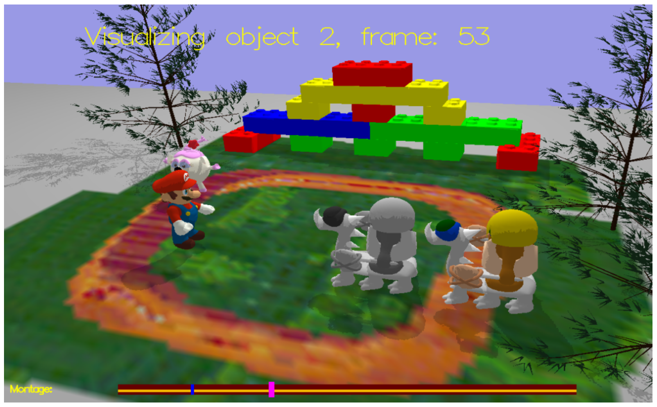

Figure 5. Mario (left, with red hat) is supposed to hit the Monster's head (right) in the script. The user adds a spatial marker for the Monster's position at the supposed time of the hit (3D replica in grey) to help plan Marios motion. The markers position in time is shown as a pink bar on the timeline.

multithreaded with separate threads for processing the camera feed, tracking and rendering. The algorithm for merging the takes usually operates in realtime. In our experiments, the length of a take is on the order of a few thousand frames (few minutes at 30fps) and the typical number of takes on is about 10. Computing a montage at this scale takes under a second.

USER STUDY: SINGLE OBJECT MONTAGE

MotionMontage combines takes in a way that tries to preserve the parts most liked by the animator and discard the disliked parts, while keeping the whole animation temporally coherent. We conducted a user study to understand the performance of the system in two ways. First, we observed users creating and annotating the original takes. Second, we evaluated whether the montage is indeed perceived to be better than the original takes. We also analyzed the effect of the Segment Length Constraint in the animators' montages.

Phase 1: Creating animations

The goal in the first phase was to familiarize the participants with using the system, then have them record takes and create a montage for a story script. We call the participants of this phase the *animators*.

Introduction to System

The animators were first given an introduction to the hardware setup and the capabilities of the system. We then demonstrated the process of recording takes, annotating them, and creating a montage for a simple script –

Mario

"Mario is happily walking around the park. He first takes a round in a clockwise direction and then turns back and takes a round in an anti-clockwise direction."

Figure 6 shows a screenshot of this story. After the demo, the animators were asked to practice using the system by acting out this script, thus familiarizing themselves with the system.

Task Design

After the demo, we asked the animators to record three takes for the following script –

Figure 6. A screenshot of the demo script - *Mario*.

Figure 7. A screenshot of the script - *Soldier takes a Break*.

Soldier takes a Break

"A soldier is guarding a castle by marching in front of it, back and forth. "This is so boring", he thinks. The weather is nice and he decides to take a walk around the castle's pathway. He looks around to see if anybody is watching him. Nobody else is there, so he starts his casual stroll. Slowly he becomes more and more carefree. Jumping around with joy, he does not notice a banana peel that is lying on his path. He steps on the banana peel, slips and falls down. He gets up slowly with effort and limps towards back to the castle. The monotonous routine starts again, guarding the castle by marching in front of it, back and forth, now with a limp. He finds it very hard to walk now and stops marching after reaching the other end."

Figure 7 shows a screenshot of this script. The animators were shown a sample path of the soldier and the goal was to record takes which roughly followed this path, while acting out the story script. We encouraged participants to try different motion styles on each take to better express the script. After recording the takes, the participants first viewed each take and then annotated them in a second viewing. The system used the annotated takes to create a montage. To conclude, the animators filled out a short questionnaire about their experience with the system.

Participants

Twenty participants (ten female, ten male, ages 20 to 30) volunteered to create animations with our system. None of them had prior 3D animation experience. The study took about one hour for each participant. At the end of this phase, we had 3 takes and 1 montage for each of the 20 animators.

Phase 2: Comparing animations.

A week after the first phase we ran the second phase of the experiment. The goal in phase 2 was to compare the quality of the animations recorded in the first phase. We refer to the participants in this phase as *scorers*. Note that some of the scorers were also the animators in the first phase.

Task Design

We asked the scorers to perform binary comparisons between a montage and one of its takes recorded by a randomly chosen set of 10 animators. The scorers first read the script which the animators had acted out. In each comparison, the scorer watched one of the three takes and the montage in a random order. The scorer then selected the animation they would have preferred if they were the script's director. Note that scorers only rated one of the animator's takes against the montage instead of ranking all 3 takes and the montage. We designed the task this way to reduce visual fatigue for the scorer. If a scorer was also an animator in Phase 1, we made sure that he did not do a binary comparison for his own animations.

In addition, if the scorer was also one of the animators, we added a second task. We asked them to rank all four animations (3 takes and 1 montage) that they created, ranking them from 1 to 4 (1 being the best) after watching them in a random order. This approach gave us a complete ranking of the animations from the animator's own perspective.

Participants

42 participants (21 female, 21 male, ages 20 to 30) volunteered for this phase. Of these, 20 were the animators from Phase 1. Each scorer's study lasted for about 50 minutes. At the end of the phase, we had 420 comparison samples, or 7 samples per comparison between a take and a montage for each animator. Additionally, we had each animator's ranking of their own takes and resulting montage.

Results

We analyze the results of the experiment in four ways. First, we present a quantitative analysis of the perceived quality of the montages from the animators' perspectives. Second, we present a similar analysis for the perceived quality of the montages from scorers' perspectives. Third, we present some qualitative feedback from the animators about their experience with the system. Finally, we present a brief discussion about the effect of the Segment Length Constraint on the montages.

Animators' perspective

We analyze animators' rankings of their own animations (Take 1, Take 2, Take 3, Montage). Figure 8 shows the that the average rank of the montage, averaged over all the animators, is much better than any of the three takes. We run a Friedman test with *rank* as an ordinal variable and *animation* as a nominal variable. *animation* is found to have a statistically significant impact on *rank*, $\chi^2(df = 3, N = 20) = 26.88, p < 0.0001$. Thus, we can conclude that the animators

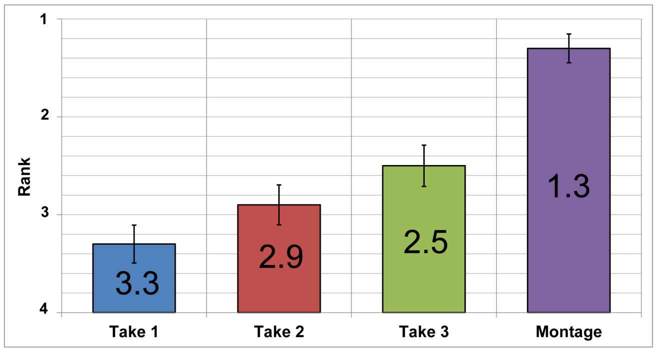

Figure 8. Average ranks for the takes (shown in the order originally recorded) and the resulting montage, as reported by animators evaluating their own work, averaged over the animators.

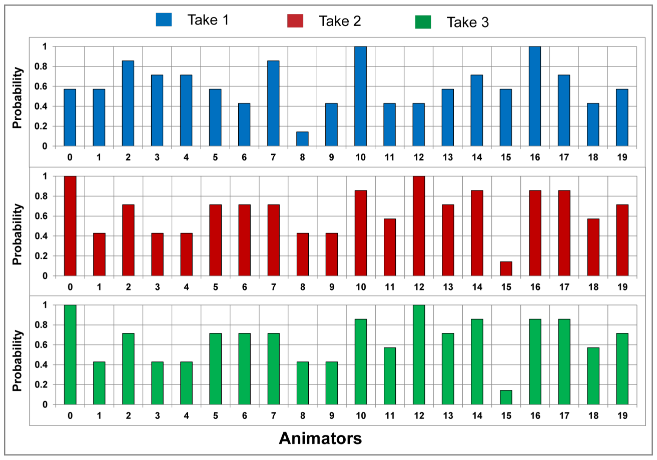

Figure 9. Probability that montage is better than a take for animators.

significantly prefer their montages to any of their individual takes.

Scorers' perspective

Next, we analyze how scorers rate other people's animations, to see if they generally prefer montages to original takes. We first compute how frequently the montage is perceived to be better than an individual take. The proportion of the scorers who voted for the montage vs. each of the 60 takes is shown in Figure 9. For further analysis, we denote the fraction of the ratings that chose the montage for animator i better than take j as m_{ij}, i.e., if $m_{ij} = 1.0$ then all 7 scorers rated the montage better, and a score of 0 has the opposite meaning.

We test three hypotheses.

H1. The average probability of an animator's montage being better than a take is greater than random chance, i.e., $\mu(m_{ij}) > 0.5$. A one-tailed single sample t-test indicates that the probability of the montage being better ($\mu = 0.6457$, $\sigma = 0.133$) is significantly greater than 0.5 ($t(19) = 5.18$, $p < 0.0001$).

We note that this t-test may be biased by the fact that we had only 7 binary samples to compute each m_{ij}. Hence we also test if the outcome of a binary comparison between the montage and a take is random. A chi-squared test over the set of all 420 binary comparisons indicates that the chance of montage being better than a take is not random ($p < 0.0001$).

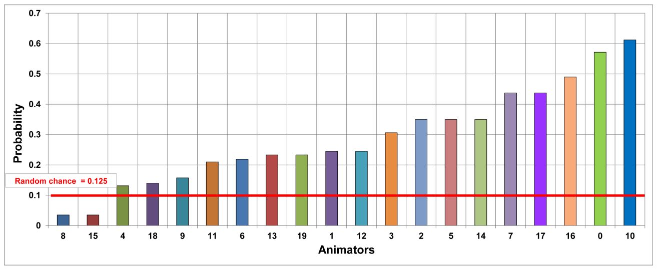

Figure 10. Probability that montage is better than all the takes for animators. The red line is the probability for random chance, 0.125.

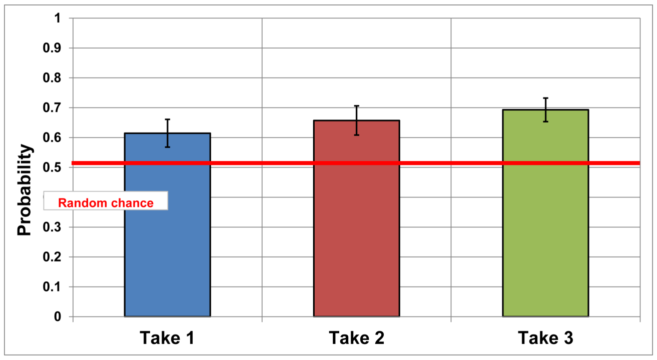

Figure 11. Probability that montage is better than a take (in the order they were recorded).

H2. The probability of the montage being better than all the three takes for an animator is greater than random chance. This probability for an animator i is given by $(m_{i1} * m_{i2} * m_{i3})$, and we denote it as b_i. Thus the hypothesis can be restated as $\mu(b_i) > 0.125$. Figure 10 shows the probabilities b_i's for individual users. For all but 2 of the animators, the probability is higher than random chance. We first use a *goodness of fit* test and confirm that the b_i's follow a normal distribution. Then we perform a one-tailed single sample t-test; the results of this test indicate that the probability of the montage being better than all the takes ($\mu = 0.299$, $\sigma = 0.126$) is significantly greater than 0.125 ($t(19) = 4.555$, $p < 0.0002$).

H3. The probability of the montage being rated better than an individual take significantly depends on the order of the takes, i.e. m_{ij} is related to j. Figure 11 shows the average probability and standard errors of the montage being better than individual takes averaged over animators. We conduct a mixed-model analysis with take j as the fixed effect, user i as random effect and probability m_{ij} as the observed variable. The analysis indicates that there is no significant effect of the order of take on the probability of montage being better, $(F(2, 38) = 0.8277, p = 0.4448)$. We also validate the correctness of this analysis by running a *goodness of fit* test on the residuals and verify that they are indeed normally distributed. Hence we reject this hypothesis.

Qualitative feedback

All the users said that they liked the capability of recording multiple takes for the story, since they did not have any prior animation experience and wanted to try out different styles. They were excited about the ability to annotate different parts of the takes and that the system could combine the takes for them. A few users suggested that the method of annotations required patience as they had to watch whole animations while annotating them. Further, they had to mentally remember the different parts from takes that they wanted to appear in the montage and annotate the takes accordingly. This leads to an interesting future direction of research of developing ways to summarize and visualize the takes together.

Effect of the Segment Length Constraint

The Segment Length Constraint in computing the montage ensures that the minimum length of a contiguous set of frames taken from a take must exceed a minimum threshold. This threshold was empirically chosen to be 60 frames, i.e., a time duration of 2 seconds. The goal of this constraint was to prevent quick take transitions in the montage.

For half the animators, the optimized montage contained no quick transitions even without the constraint. For the other half, adding the constraint avoids such segments but results in a montage with a slightly higher cost. However, the increase in the cost was found to be less than 1% in all the 10 cases. Hence the proposed algorithm is able to enforce the Segment Length Constraint without significant penalty.

Results summary and discussion

The results suggest that both the animators and the scorers significantly preferred montages. It is interesting to note that the average rank assigned by the animator is affected by the order in which takes were recorded. The rank is better for a take that was recorded later. We speculate that the animators, who were creating animations for the first time, grew better at acting out stylized motions with subsequent takes. However, the non-animator scorers' preference for the montage over the takes does not depend on their order to a statistically significant degree. This result does not support the conclusion that overall animation quality increases as animators gain more experience. We do not have a hypothesis for this difference and leave the understanding of visual or psychological perception in such animations to future work.

CONCLUSION AND FUTURE WORK

We have presented a system, MotionMontage, which allows users to record 3D animations involving multiple objects, one object at a time. The user can experiment with different motion styles and trajectories and record multiple takes for each object. The system allows the user to temporally annotate each take to indicate which parts of which takes are considered better or worse. The system then uses a novel formulation to combine these annotated takes into a montage. The same process can be performed sequentially for all the objects in the animation. The recording of takes for more than one object is handled via the traditional layered animation approach. We provide spatial markers to help the user see the recorded motions of other objects while recording the current object's motion.

We also reported on a user study to qualitatively and quantitatively measure the efficacy of MotionMontage from the animators' and the scorers' perspectives. The results indicate that the montage is significantly preferred over the original takes by both. In the supplementary video, we also show some multi-object animations created by a few users. The users found the system intuitive and indicated that the spatial markers helped them to record object interactions better in the animation.

We have identified a few directions for future work.

Better animation recording setup.

The users of our system suggested demarcating the physical volume of play area. This would prevent a user from moving the controller outside the volume which leads to loss of tracking. In addition, the current system uses a standard (and cumbersome) mouse-driven interface to adjust the camera's viewpoint on the screen; however, we could instead use a tracked object to adjust the camera.

Currently, a big challenge while recording the animations is to correctly handle the interactions between the animated object with the virtual scene props and with previously animated objects. The spatial markers can help to some extent, but the lack of physical proxies for other objects makes it hard. Allowing users to use physical proxies or automatically inferring the user's intent from the recording can alleviate this challenge and suggests an interesting avenue for future work.

To further improve the richness of the 3D animation, we are interested in enabling articulated 3D characters in the animation. Difficulties in mapping the articulation of physical objects to a character's motion will require new, intuitive puppetry interfaces.

The current interaction design of MotionMontage can be extended to a multi-modal interface involving voice, gestures, multiple physical controller objects, etc. Using multiple modalities may provide more natural interfaces but may make the system more complex to use. In the future we would like to explore this multi-modal interaction space and try to find a good trade-off between user empowerment and intuitive, easy-to-use interfaces for creating 3D animations.

Exploration of the annotation setup.

We enabled the users to annotate a recording easily by manipulating a physical slider while the recording played in realtime. The users may find it hard to keep up with the realtime pace of the recording and hence we added a lag in the annotations based on the human reaction time. However, this is just a heuristic. There can still be parts of the recording where the annotations are slightly delayed or ahead in time, specially in case of fast paced recorded motion. Our algorithm creates a montage which transitions smoothly between the takes while respecting these rough annotations. We also tried a keyboard and mouse-based interface which allowed users to select parts of different takes, play the selected parts, and then annotate them. This interface gives more control to the user and hence more accurate annotations. However, this process can become tedious for lay users. Hence we chose a simpler interface that results in rough annotations and focused more attention on developing an algorithm which combines these roughly annotated takes into the montage. In the future, it will be interesting to study the design of different annotation interfaces and their effect on the montage.

Enable audio recordings.
We would like to add the capability of recording audio voiceovers in the system. Currently, audio can be added afterward using standard audio-video processing software. However, some users indicated that they were actually humming and speaking in their mind while acting out the story, and would have liked to record audio while animating. Rubin et al. [16] have recently developed tools for editing and merging multiple audio takes of a story; it would be interesting to combine their tools with our motion editing interface.

In sum, we have presented a novel system that allows novice users to explore their creativity through the medium of 3D animated stories. We learned from our user studies that such animation creation systems can empower the user while being intuitive, with easy-to-use interfaces. We believe that MotionMontage is a small step in this direction and hope that our work motivates more researchers to explore this domain.

REFERENCES

1. Human benchmark - reaction time statistics. www.humanbenchmark.com/tests/reactiontime.
2. Agarwala, A., Dontcheva, M., Agrawala, M., Drucker, S., Colburn, A., Curless, B., Salesin, D., and Cohen, M. Interactive digital photomontage. In *ACM SIGGRAPH 2004 Papers*, SIGGRAPH '04, ACM (New York, NY, USA, 2004), 294–302.
3. Arikan, O., Forsyth, D. A., and O'Brien, J. F. Motion synthesis from annotations. *ACM Trans. Graph. 22*, 3 (July 2003), 402–408.
4. Barnes, C., Jacobs, D. E., Sanders, J., Goldman, D. B., Rusinkiewicz, S., Finkelstein, A., and Agrawala, M. Video puppetry: a performative interface for cutout animation. In *ACM SIGGRAPH Asia 2008 papers*, SIGGRAPH Asia '08, ACM (New York, NY, USA, 2008), 124:1–124:9.
5. Boykov, Y., Veksler, O., and Zabih, R. Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Anal. Mach. Intell. 23*, 11 (Nov. 2001), 1222–1239.
6. Dontcheva, M., Yngve, G., and Popović, Z. Layered acting for character animation. In *ACM SIGGRAPH 2003 Papers*, SIGGRAPH '03, ACM (New York, NY, USA, 2003), 409–416.
7. Gleicher, M. Motion editing with spacetime constraints. In *Proceedings of the 1997 symposium on Interactive 3D graphics*, I3D '97, ACM (New York, NY, USA, 1997), 139–ff.
8. Gupta, A., Fox, D., Curless, B., and Cohen, M. Duplotrack: a real-time system for authoring and guiding duplo block assembly. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*, UIST '12, ACM (New York, NY, USA, 2012), 389–402.
9. Held, R., Gupta, A., Curless, B., and Agrawala, M. 3d-puppetry: a kinect-based interface for 3d animation. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*, UIST '12, ACM (New York, NY, USA, 2012), 423–434.
10. Koller, D., and Friedman, N. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.
11. Kovar, L., and Gleicher, M. Flexible automatic motion blending with registration curves. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*, SCA '03, Eurographics Association (Aire-la-Ville, Switzerland, Switzerland, 2003), 214–224.
12. Kovar, L., Gleicher, M., and Pighin, F. Motion graphs. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '02, ACM (New York, NY, USA, 2002), 473–482.
13. LoBrutto, V. Stanley kubrick: a biography. Da Capo Press, Incorporated (1999), page 430.
14. Myers, C. S., and Rabiner, L. R. Comparative Study of Several Dynamic Time-Warping Algorithms for Connected-Word Recognition. *The Bell System Technical Journal 60*, 7 (1981).
15. Pérez, P., Gangnet, M., and Blake, A. Poisson image editing. In *ACM SIGGRAPH 2003 Papers*, SIGGRAPH '03, ACM (New York, NY, USA, 2003), 313–318.
16. Rubin, S., Berthouzoz, F., Mysore, G. J., Li, W., and Agrawala, M. Content-based tools for editing audio stories. In *Proceedings of the 26th annual ACM symposium on User interface software and technology*, ACM (2013), 113–122.
17. Ruegg, J., Wang, O., Smolic, A., Gross, M., Auzinger, T., Guthe, M., and Jeschke, S. Ducttake: Spatiotemporal video compositing. *Computer Graphics Forum (Proceedings of EUROGRAPHICS 2013) 32*, 2 (May 2013).
18. Sarawagi, S., and Cohen, W. W. Semi-markov conditional random fields for information extraction. In *In Advances in Neural Information Processing Systems 17* (2004), 1185–1192.
19. Shoemake, K. Animating rotation with quaternion curves. In *Proceedings of the 12th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '85, ACM (New York, NY, USA, 1985), 245–254.
20. Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A., and Blake, A. Real-time human pose recognition in parts from single depth images. In *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '11, IEEE Computer Society (Washington, DC, USA, 2011), 1297–1304.
21. Sturman, D. J. Computer puppetry. *IEEE Comput. Graph. Appl. 18*, 1 (Jan. 1998), 38–45.