Eurographics/ACM SIGGRAPH Symposium on Computer Animation (2004)
R. Boulic, D. K. Pai (Editors)

Enriching a Motion Collection by Transplanting Limbs

Leslie Ikemoto and David A. Forsyth[1]

[1] {lesliei | daf} @cs.berkeley.edu
U.C. Berkeley Computer Science Department

Abstract

This paper describes a method that can significantly increase the size of a collection of motion observations by cutting limbs from one motion sequence and attaching them to another. Not all such transplants are successful, because correlations across the body are a significant feature of human motion. The method uses randomized search based around a set of rules to generate transplants that are (a) likely to be successful and (b) likely to enrich the existing motion collection. The resulting frames are annotated by a classifier to tell whether they look like human motion or not.

We evaluate the method by obtaining motion demands from an application, synthesizing motions to meet those demands, and then scoring the synthesized motions. Motions synthesized using transplants are generally somewhat better than those synthesized without using transplants, because transplanting generates many frames quite close to the original frames, so that it is easier for the motion synthesis process to find a good path in the motion graph. Furthermore, we show classifier errors tend to have relatively little impact in practice.

Finally, we show that transplanted motion data can be used to synthesize motions of a group coordinated in space and time without producing motions that share frames.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Animation

1. Introduction

Human motion is a topic of tremendous current importance: those who build **virtual worlds** would like to populate them. One part of this community is the entertainment industry; another is simulation. Variations in rendering style alter a viewer's perception of motions [HOT98, HOT97], meaning better rendering is creating an increasing demand for realistic motion.

Motions are currently obtained for virtual environments using a body of measurement techniques collectively known as **motion capture**. Reviews of available techniques appear in, for instance [BRRP97, Gle99, Men99, Moe99, SPB*98]. Motion capture data is used in very large quantities by, for example, the movie and computer game industries. For each title that will contain human motion, an appropriate script of motions is produced; typically, this involves a relatively small set of "complete" motions that can be joined up in a variety of different ways (as in [GSKJ03]). This script is captured, and then motions are generated within the game by attaching an appropriate set of these motion building blocks together. Motions captured for a particular title are then usually discarded as re-use presents both economic and legal difficulties.

2. Background

How to build a really effective human motion synthesizer? There are three important threads in this literature. In **motion editing**, one builds a system where an author can interactively modify a motion path — by requiring, say, that a hand pass through a particular spot at a particular time (e.g. [KSG02, LS99, PBM00a]). The motion is then updated to meet such **space-time** constraints, and it is the author's job to manage the constraints and the update process such that the resulting motion looks human. This process can be made to work satisfactorily, by seeing the update process as obtaining the solution to a minimization problem [Gle97, GL98, Gle01] or as obtaining the solution to an inference problem [TSK02]. This work has produced several important insights. The first is that it is quite dangerous to require large changes in a motion signal; typically, the resulting motion path does not look human (e.g. [Gle99]). The second is that motion is the result of extremely complex considerations; requiring that an estimate of the energy ex-

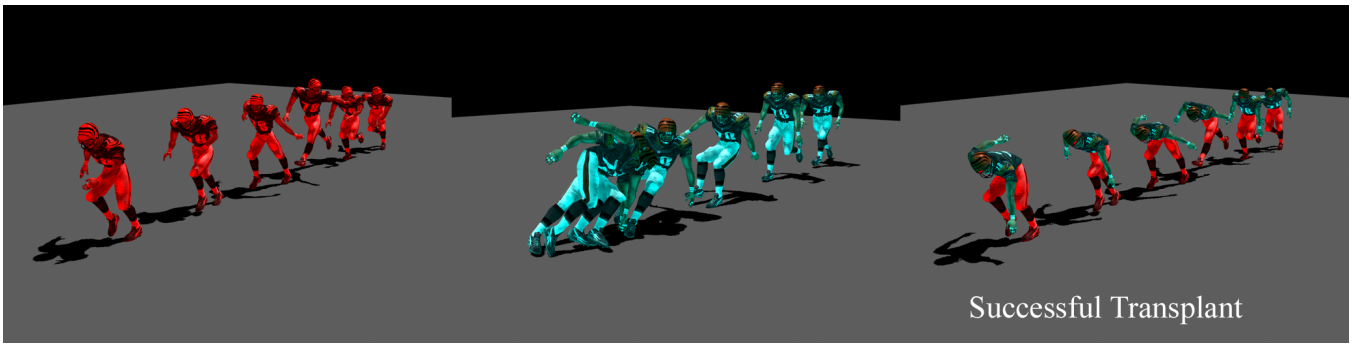

Figure 1: Left and **center** are frames from motion sequences observed using motion capture equipment. **Right** shows a sequence constructed using one of our rules; the upper body from the top sequence has been attached to the lower body from the bottom sequence. This leads to a substantial change in the qualitative structure of the motion — we have no head-butting sequences in our original collection — but retains the human character of the motion. In this figure, the frames and colours correspond in the natural way.

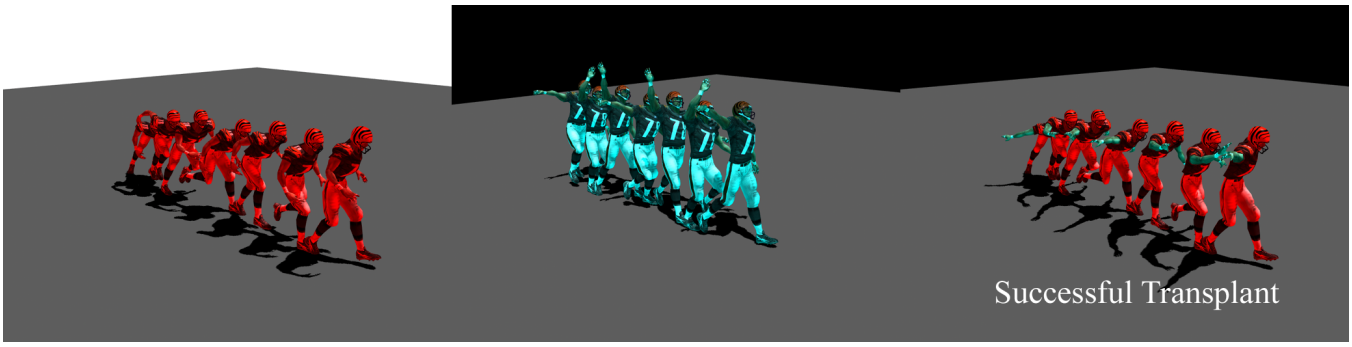

Figure 2: Left and **center** are frames from motion sequences observed using motion capture equipment. **Right** shows a sequence constructed using one of our rules; an arm and shoulder from the top sequence has been attached to the rest of the body from the center sequence. As in this case, this quite commonly leads to relatively small changes in the qualitative structure of the motion, and retains the human character of the motion. In this figure, the frames and colours correspond in the natural way.

pended in a motion be minimized does not usually produce a human-looking motion.

In **motion interpolation**, one attempts to produce motions that interpolate between, or extrapolate from, existing motion-capture measurements. A natural procedure is to produce a controller that can track the measurements and then, when measurements are no longer available, produce motions by controlling some body parameters. Controllers that track motion data provide a useful mechanism for smoothing recorded errors while also adjusting for disturbances not present in the recorded motion [FP03, PBM00b, ZH99, ZH02]. Other approaches make use of hand designed or optimized controllers that operate independently from recorded motion [FvdPT01a, FvdPT01b, GTH98, HWBO95, PW99]. Building controllers that generate human-like motion remains an open research problem.

Data-driven motion synthesis attempts to infer good, new motions from an existing set of measured motions (the origins appear to be with [MTH]). There are a series of **combinatorial** methods, which rearrange measurements to obtain new motions. Different authors represent the object in different ways, but the underlying phenomenon is a graph whose nodes are the frames of motion in the set of measurements, and whose arcs link frames of motion that can be joined. Links are usually directed and weighted with the "goodness" of the link. Legal motions are then paths in this graph. Kovar *et al* search this graph, meaning that given a frame of motion they concentrate on choosing the next frame of motion [KGP02]. Lee *et al.* use a search with a longer horizon, expanding the graph into a tree of some fixed depth rooted at the current node [LCR*02]. Local searches can produce motions on-line, but must suffer from the traditional horizon problem that applies to such searches — if one requires particular start and end frames, and these can be joined only by one particular path, then it may be difficult to find that path locally.

Arikan and Forsyth attack the problem by searching for entire paths [AF02]. The search space has two important qualitative features: first, motion constraints are **ambiguous**, because the family of acceptable paths is usually very large. Second, human motion generally has **low entropy**; if I am walking at time t, I am probably still walking some seconds later.

Arikan *et al.* exploit both properties [AFO03]. We work within this framework, and so describe it in some detail. Their approach allows synthesis of motions that meet qualitative constraints on the timeline. In particular, all frames are annotated with annotations drawn from some fixed vocabulary, and one can author motion by painting desired annotations on the timeline. The author can demand that an annotation be present, be absent, or specify don't care. Start points and end points can be specified, and the motion can be constrained to pass through particular example frames at specified times. The annotation constraints are met by searching for a path that has the right labels at each frame. This is too computationally expensive to solve by dynamic programming. Instead, one can coarsely quantize the graph into blocks of frames that form sequences and then use dynamic programming on a random subset of these blocks. There are then two search activities: refining blocks, and changing the (randomly chosen) working set of blocks. This works, because ambiguity means that one doesn't miss much structure by random sampling and low entropy implies that a quantized path represents the actual solution quite well.

Smoothing methods build new motions from existing measurements. Some approaches are quite successful. Pullen and Bregler use a method similar to texture synthesis to construct motions of the upper body conditioned on motions of the lower body; the approach is applied to measurements of dances, and is quite successful [PB02]. An alternative is to build some form of statistical model from the data set. Kovar and Gleicher time align sequences displaying a particular type of motion, and then build a linear combination of the time aligned sequence [KG03]. Mataric and colleagues build local linear models of arm movements using dimension reduction methods and function approximation methods [FMJ02, JM02, JM03].

While many motion synthesis techniques have been proposed, to our knowledge thorough **evaluation methodologies** have not. Such a methodology should issue a performance guarantee, i.e. a certificate that a percentage of the motion generated from a particular technique is acceptable. Furthermore, it should perform this evaluation on a large-scale in the context of real motion demands derived from relevant applications. In Section 4, we describe such a methodology and use it to evaluate our transplantation technique.

2.1. The Poverty of the Stimulus

Both combinatorial and smoothing methods have serious practical difficulties. Combinatorial methods can produce only motions that have been observed — if one hasn't seen a 180^o turn, it isn't possible to produce one. Smoothing methods are currently successful solely when one ensures that only motions of similar type — all walks, all runs, all reaches — are smoothed. This means that the success of either method is crucially dependent on having sufficient data.

Two phenomena guarantee that sufficient data will never be available. First, **object interaction**: people reach, kick, lift, etc. objects, creating a parametric families of motions. Measuring motion in the exact context of the objects [LCR*02] is practical only for special cases. This is what motivates smoothing methods, but it is generally difficult to preserve fast detail in smoothed motions.

Second, human motion is **compositional**: movements can be composed across the body, so that, for example, that one can walk and scratch one's head. The range of available compositions is vast, lending a "combinatorial explosion" flavour to the problem. No currently conceivable collection of data will contain examples of each possible composition. As a result, a successful motion synthesis algorithm, whether rooted in combinatorial or smoothing methods, must require some encoding of legal motion compositions (as opposed to simply looking up all motions in a set of examples).

It is natural to attack composition by cutting and pasting across the body. However, many of the resulting motions do not look human, a result of cross-body **correlation**. Motion at some points on the body is often correlated with motion at other parts of the body. There are two phenomena, one active and one passive. The active phenomenon occurs in such movements as relaxed walking, where arms swing out of phase with the legs for energetic reasons. This is a gait that is chosen by the actor, and can be broken at will — for example, to scratch or to reach. However, if a cut-and-paste results in a motion where these correlations are, say, out of phase, the motion will look forced and may not look human at all. Passive correlations occur as a result of generating large torques at some joints; other parts of the body may experience reaction torques. For example, a very fast reaching movement of the arm, launched while walking, may result in a passive twitch of the opposite arm. If the arm that is reaching is replaced with some other arm movement, the resulting motion will not look human because the passive twitch will be unexplained.

2.2. Overview

The main limit on the capacity of combinatorial motion synthesis and its applications is the richness of the pool of available motions. In this paper, we demonstrate that it is possible to enhance a collection of human motions substantially by cutting and pasting across the body — a process we call **transplantation** (Section 3). We also describe an assessment method for our technique that can easily generalize to evaluate other techniques (Section 4). In Section 5, we use our enhanced pool of motions to demonstrate that with an enriched motion graph, we can synthesize multiple motions organized in time and space that do not share frames.

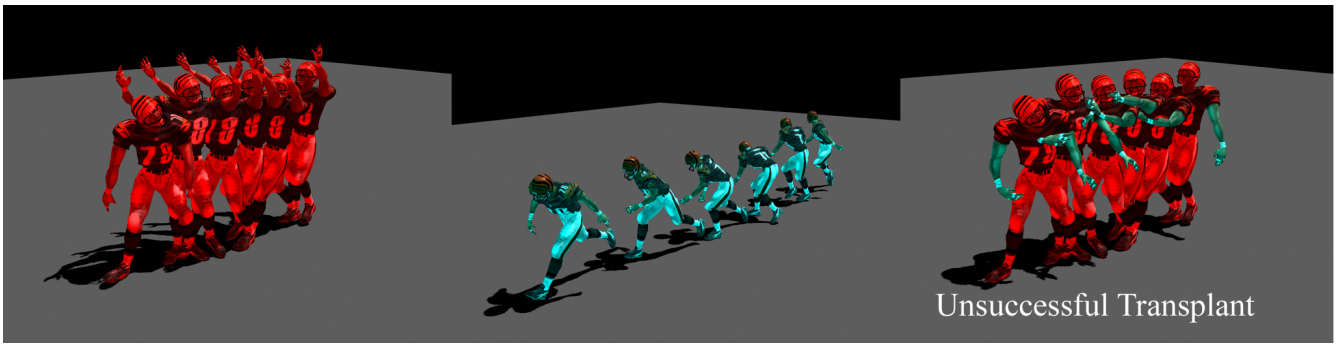

Figure 3: Not every transplant is successful. **Left** and **center** are frames from motion sequences observed using motion capture equipment. **Right** shows a sequence constructed using one of our rules; both arms from the top sequence has been attached to the rest of the body from the center sequence. In this case, we have produced bizarre behaviour in the arms; such frames are usually annotated `not looks human` by our classifier.

3. Enriching a Motion Collection by Transplantation

It is currently difficult to predict precisely which components of motion are used to determine whether a motion looks human or not. However, this decision almost certainly involves observing correlations. In turn, while we cannot currently predict precisely whether a particular cut-and-paste will result in an acceptable motion, we can identify rules that are likely to yield a high percentage of successful results. This suggests using these rules to perform a randomized search through the available cut-and-pastes, with resulting motions accepted or rejected by a classifier. We have found this strategy to be successful.

3.1. Proposing a Transplant

Our approach to transplantation is simple. We invoke one of four rules uniformly and at random, and apply that rule to two sequences (chosen randomly, using parameters determined by the rule). Each rule determines whether the pair of sequences to which it is applied is acceptable. It then prepares a set of new sequences obtained by transplanting various parts of the upperbody from the first sequence to the second, and vice versa. Our current set of rules does not transplant any signals from below the waist, though additional rules could easily be added.

The **transplants** are rule-dependent. For rule 1, we transplant the upper body of the first motion with the lower body of the second motion, and vice-versa. For rules 2 and 3, we transplant:

- The right arm severed just above the shoulder from sequence 1 to the right arm of sequence 2.
- The left arm severed just above the shoulder from sequence 1 to the left arm of sequence 2.
- The shoulders and arms severed at the top torso joint from sequence 1 to sequence 2.
- The left arm, shoulder and torso severed at the top torso joint from sequence 1 to sequence 2 (preserving sequence 2's right arm and shoulder joint).
- The right arm, shoulder and torso severed at the top torso joint from sequence 1 to sequence 2 (preserving sequence 2's left arm and shoulder joint).
- The uppercarriage (torso, shoulders and arms) from sequence 1 to sequence 2.

We then transpose sequences 1 and 2, and repeat the process, resulting in twelve new sequences. For rule 4, we do not transplant. Instead, we pick one of the two motions at random, and select joint angles on the arms to perturb. We generate a sequence where we perturb the selected angles only on the right side of the body, then a sequence where we perturb only the angles on the left, and finally a sequence where we perturb both.

Our rules are designed to balance the goal of obtaining successful transplants with that of searching a wide range of possible transplants. One can reasonably believe that transplantation will work best if the source sequences are similar. However, confining transplants to similar sequences means that one is unlikely to obtain any significantly new compositions. This suggests that one should try a broad range of transplants. An attractive feature of randomized search is that one can incorporate new rules as the process is better understood, perhaps using reinforcement learning to choose the rule applied.

Rule 1 searches for general compositions by choosing two sequences at random; we start at the first frame of each sequence and clip the longer sequence to the length of the shorter. Our observed motions tend to be in phase, meaning there is no particular advantage to be obtained by searching for a better alignment.

Rule 2 again tries to find pairs of sequences that are likely to work well, but now uses the criterion that sequences with similar footplants are likely to allow successful composition. This criterion is appropriate for finding motions where the correlation across the body is unlikely to be disturbed

by transplantation. For some motions, one can regard correlation within the motion as being governed by an internal clock (e.g. arms and legs swinging in counter phase in a walk). The footplants are a rough guide to the internal clock; this suggests that pairs of motions whose footplants are sufficiently similar share a similar clock, and so may admit successful transplantation. The attraction of this criterion is that, while the footplants may be similar, the upper body may engage in quite different activities, meaning that transplantation could lead to quite significantly different motions. We implement this criterion by taking two sequences uniformly and at random, and forming for each a string representing whether and which foot is planted at each frame. The start of the aligned subsequences is found by running along time until the two sequences have a simultaneous footplant; we then continue advancing through time until the footplants are poorly aligned.

Rule 3 searches for transplants that are likely to work well, by using the distance matrix of Kovar *et al.* [KGP02] to obtain a joint probability matrix. The rule then uses importance sampling to choose according to these probabilities. We write the distance matrix from that paper as $\mathbf{D}$; now zero the diagonal to obtain $\hat{\mathbf{D}}$. We then form the matrix $\mathbf{J}$ whose i, j'th element is $\mathbf{e}^{-\mathbf{d_{ij}}/\langle 2\sigma_{\mathbf{d}}^2\rangle}$, where d_{ij} is the i, j'th element of $\hat{\mathbf{D}}$. The joint probability matrix is obtained by normalizing $\mathbf{J}$ to sum to one. This rule selects sequences preferentially if the sequences are "similar". We now use Dijkstra's algorithm to determine a possible time alignment between the sequences (as in [KG03]). If this time alignment is sufficiently similar to the identity (i.e. its slope is everywhere close to 1), we accept the pair of sequences and transplant. We choose a random start point within each sequence to begin the alignment, and the endpoint is provided by the time alignment procedure.

Rule 4 first picks one of the two motions at random, then randomly selects whether to perturb the shoulder angles, the elbow angles, or both. The rule then chooses an angle offset between 20 and 90 degrees, favoring smaller angles by sampling according to a skewed distribution. Note that one can easily expand Rule 4 to include other angles (such as those on the torso), and that it could be applied to a transplant created by Rules 1, 2, or 3.

3.2. Determining which Motions Look Human

None of the transplants proposed by any rule is guaranteed to look human. We must check each frame to determine whether it does. It is infeasible to check large quantities of motion by hand. In addition, no algorithm currently exists to check automatically. Our solution is to regard `looks human` as yet another annotation within the Arikan *et al.* framework. As in that framework, we use a support vector machine (SVM) to determine whether a motion appears human or not. Using a classifier to distinguish human-looking motion is an appealing option because a classifier generalizes its training labeling without concerning itself with the underlying semantics. It is also a feasible solution because in practice motions that do not look human typically fail to do so as a result of either fast twitches uncorrelated with the rest of the motion (Figure 3), or self-interpenetration. These are both phenomena relatively easily picked up on by a classifier.

We used a radial basis function kernel SVM, from the public domain library libsvm [CL00]. (Note that the SVM is one of a small number of classifiers that tend to perform well on a wide-range of problems, but one could choose other classifiers such as logistic regression, neural nets, or a naive Bayes classifier and expect comparable results.) Every frame of our observed motions is annotated with `looks human`. We classify every frame of the transplanted motion sequences; if the frame is accepted by the classifier it is annotated with `looks human`, otherwise it is annotated with `not looks human`. The classifier uses joint positions in the frame to be classified, the 30 frames before, and the 30 frames after as a feature vector.

The classifier is trained using two pools of examples. First, every frame of each observed human motion we possess is used as a positive example. Second, we classified a pool of 16,127 frames of transplanted motion by hand. One can create an accurate training set for the classifier because humans are very skilled at discriminating human-looking motion from other candidates. Also, as we said above, in practice motions that appear non-human often feature uncorrelated fast twitches or self-interpenetration. Both phenomena are easy for a person training the classifier to spot. One can also create a large training set quickly because the annotation flag tends to change relatively seldom for a sequence, so that it is possible to label many frames efficiently.

3.3. Annotating Transplanted Motions

In much of what follows, we use annotations attached to motions. This means that transplanted motions must be annotated. We use the annotation scheme of [AFO03], which we sketch briefly here for the reader's convenience. In particular, we follow Arikan *et al* by choosing to focus on annotations that describe the qualitative properties of motion, and are using a database of 7 minutes of American football motions. The vocabulary used to annotate this database consisted of: `Run`, `Walk`, `Wave`, `Jump`, `Turn Left`, `Turn Right`, `Catch`, `Reach`, `Carry`, `Backwards`, `Crouch`, `Stand`, and `Pick up`. Any combination of annotations is allowed, though some combinations may not be used in practice. Motions are then annotated using an active learning method based around an SVM to set each flag (i.e. `Run` vs. `Not Run`) independently. The classifier uses the joint positions for one second of motion centered at the frame being classified as a feature vector. The out of margin cost for the SVM is kept high to force a good fit within the capabilities of the basis function approximation.

We obtained this annotated collection from Arikan *et al*. The question now is to annotate transplanted motions with

minimal extra effort. In practice, we have found it sufficient use nearest neighbours. We take a 17 frame sequence centered around the transplanted frame to be annotated, and find the nearest such sequence in the original annotated motions by blank search. The search cost is relatively insignificant, and is an off-line cost. The annotation sequence is then smoothed by one pass of opening and one pass of closing (as in [FF99]). As figure 1 suggests, our transplantation procedure may result in motions that are best controlled with an expanded annotation vocabulary (the figure might well be a `head-butt`). We have not explored this point in detail.

3.4. Transplantation Results

We use a test set of transplants to establish the classifier's total error rate. The annotation bit supplied by the classifier tends to change seldom on test sequences, too. This means that it is difficult to check large numbers of individual frames, and our estimates of total error rate are rough. We arrive at these estimates by looking at short runs of motion, and count errors in terms of these runs. If the whole motion looks human to the observer, but the classifier marks only a quarter of the frames as human, we count a quarter motion as true positive and three-quarters as false negative. This gives a rough estimate of the error rates. On 102 test motions of a total of 14,108 frames, the total error rate is approximately 13%. The false positive rate is approximately 12%.

We apply our transplant strategy in batch mode, meaning that transplanted sequences are not available for re-transplantation. We have not attempted every possible application of the rules, but can estimate how many successful transplants are available because the rules are applied uniformly at random. From 118 observed motions, we applied the rules a total of 106 times, obtaining 234 motions and a total of 27491 frames. It is not possible to classify the first and last 30 frames of each motion (because features are incomplete), and so we have 13451 annotatable frames, of which the classifier marked 10226 frames as human.

Rule 1 is applied 29 times. Of a total of 6903 possibilities, all applications find pairs that could be transplanted, yielding 3904 new frames. Approximately 35% of the resulting frames are (a) annotatable and (b) annotated as `looks human`. This suggests that rule 1 could yield approximately a quarter million good frames annotated as `looks human` if applied exhaustively. Rule 2 is applied 41 times. Of a total of 6903 possibilities, 19 applications found pairs of sequences from which motions could be transplanted, yielding 12660 new frames. Approximately 44% of the resulting frames are (a) annotatable and (b) annotated as `looks human`. This suggests that rule 2 could yield approximately a million new frames annotated as `looks human`. Rule 3 is applied 36 times. Of a total of 6903 possibilities, 18 applications found pairs of sequences from which motions could be transplanted, yielding a total 15432 new frames. Approximately 28% of the resulting frames are (a) annotatable and (b) annotated as `looks human`. This suggests that rule 3 could yield approximately three-quarters of a million new frames annotated as `looks human`. Rule 4 is applied 30 times. Of a total of 118 possibilities, all applications find angle perturbations, yielding 2,841 new frames. Approximately 35% of the resulting frames are (a) annotatable and (b) annotated as `looks human`. This suggests that rule 4 could yield approximately 4000 new frames annotated as `looks human`. There is no particular reason to believe that the rules lead to sets of frames that are distinct, but it is reasonable to believe that from half-a-million to a million new, *good* frames are available via this route. We have no statistics on multiple applications of the transplant process (i.e. transplanting a transplant), but believe much new motion is available this way. Comprehensive application of the rules in this manner is currently computationally impractical, however.

These results suggest that transplants lead to many frames that look acceptable, at least to the classifier. Figure 2 shows a successful transplant (one that looked good to the classifier). In this case, the motion has changed, but not greatly. Figure 3 shows a transplant that has produced a poor motion that was identified as such by the classifier. Transplants can potentially produce motions quite different in character from those in the original collection of observations, as figure 1 indicates.

To date, we have used our technique to generate 340,596 frames of motion data accepted by the classifier as human (from a pool of 477,362 annotatable candidate frames). At 60 frames per second, this means we have generated slightly over an hour and a half of new data from approximately 7 minutes of original motion capture.

4. Assessing Motion Modification Strategies

One can evaluate a motion modification strategy only in the context of motion demands. The demands should come from a relevant application. A strategy can then be evaluated by generating many demands, synthesizing a large quantity of motion sequences, and scoring them. We have implemented a system that evaluates our motion transplantation technique in this manner, and believe it is easily extensible to evaluate other synthesis techniques.

Creating a large, canonical set of motion demands is not trivial. The types of demands typically needed for useful applications are not random, so we cannot simply place annotations randomly on a timeline. Additionally, creating demands by hand is not feasible at a large scale. Instead, we use the popular video game Unreal Tournament 2004 to generate motion demands. At every one-fifth of a second, we query the game state for position, velocity, rotation, and an annotation supplied by the game of either running or falling for every player. We enrich this set of annotations by examining the velocity vector at every tick the game describes as running. We annotate the tick with `standing`, `walking`, or `running` by setting thresholds on the magnitude of the velocity vector. In this manner, we generated over 1200 anno-

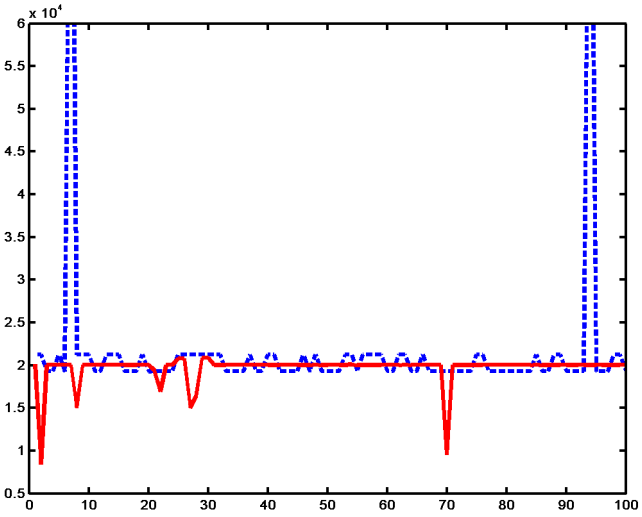

Figure 4: This plot shows the scores for 100 sequences generated from for the original motion graph (in **dashed blue**) and the transplantation-enriched motion graph (in **solid red**). This figure indicates that transplantation can create significantly better motions for certain demands, and avoid catastrophes for others.

tation streams of slightly less than 300 frames apiece from less than an hour of logging 3-person games. The streams generated in this manner do not test all possible annotations for our particular motion database of American football. We created an additional 40 streams by hand that include annotations from the full annotation set.

To evaluate a motion graph, we need a system that takes a set of demands and a motion graph, and tries to generate motion sequences that meet the demands. It should also score how well the generated motion sequence meets its demand. We use the system described by Arikan *et al.* to perform these tasks.

For each demand, the system generates two motion sequences. It generates the first using our original motion capture database as the motion graph, and the second using our transplantation-enriched database. The system is permitted to perform 300 iterations of search to solve for each demand. Arikan *et al.*'s procedure assigns a score to each motion sequence. Though this score is not canonical, we believe it is a reasonable proxy because it reflects how well a motion sequence meets its demand; and, in general, better scores tend to indicate better motions.

We expect to see the enriched motion graph produce sequences whose scores are better than the original motion graph for some demands, and it does (Figure 4). We also expect that the motion sequences generated seldom use frames which are labelled `not human`. They rarely do (of 28,800 frames synthesized to meet our 100 demands, only 980 — or 3.4% — are annotated `not human`). Lastly, we verify that the motion sequences use frames that were newly generated, and do not solely contain frames from the original motion capture collection.

4.1. Examples and Results

To assess the impact of mislabelled frames, we need to check all the frames by hand. We subsampled 14,108 frames from our transplantation-enriched database, annotated them with the classifier, and then had a human check the annotations. Of 7988 annotatable frames, the classifier labelled 5044 of them as `human`. For this experiment, we test our original motion capture graph against one that contains all of the original motion capture data plus the subsampled version of our enriched graph. This gives us a total of 36,753 frames. The original graph contains 22,645 frames.

Our set of motion demands consists of 40 annotation streams created by hand, and 60 gathered from Unreal Tournament 2004. For each graph, we generate 100 motion sequences from these demands. Then we compare the scores for the motion sequences. A lower score indicates that the sequence met its demand better than a sequence with a higher score.

Is the transplantation-enriched motion graph better than the original? As Figure 4 shows, the enriched graph generates significantly improved motion sequences for some demands. This is because adding even a small number of frames from transplantation makes the graph easier to search. Transplantation tends to locally explore the space around existing examples, creating frames that are similar to original frames. Thus, the search used in Arikan *et al.*'s system can choose between several similar frames, allowing small refinements that can increase continuity between motion blocks. The figure also illustrates that the enriched graph avoids catastrophic synthesis results for motion demands that are difficult to meet with the original graph.

Now we assess the impact of frames that do not look human. We first examine the effect of mislabelled frames — frames that the classifier labels as human looking, but a human thinks are not. Such frames, which form about 12% of the graph, are potentially dangerous because their inclusion in a sequence can make the motion look implausible. For each sequence generated from the enriched database, we count the number of falsely labelled frames the motion contains. Only 7 out of the 28,800 frames used were false-positives. Therefore, one can safely ignore their presence. Recall that frames that do not look human usually contain fast twitches that are uncorrelated with the rest of the motion, or interbody penetration. We conjecture that our success at avoiding false-positive frames may be because such frames are not well-connected to the rest of the motion graph.

Secondly, we determine how often we visit frames which

the classifier labelled `not human`. 3.4% of the frames used (or 980 of the 28,800 total frames in the sequences) were classified not human. However, this percentage is controllable. We left the frames labeled `not human` in the database because such frames can potentially improve continuity. We demand motions in which every frame has a `looks human` annotation, and we will get such motions unless inserting frames that are not so annotated produces a significantly smoother motion. Our experience is that the tendency to produce smoother motions significantly outweighs the presense of the occasional frame — which is usually kinematically acceptable — from a problem sequence. However, the frames labeled `not human` can be stripped before synthesis if the user wishes.

Lastly, we gauge the number of times we visit frames that were not in our original motion collection. 20.1% of the frames (or 5,984 of 28,800) were generated from transplants. This figure indicates that the frames generated by transplantation were beneficial in meeting the motion demands.

5. Generating Multiple Unique Sequences from a Single Motion Demand

One application of our technique is the generation of multiple sequences from a single motion demand that do not share frames in detail. This is an interesting problem in crowd synthesis, for example. We concentrate on getting a small group of actors to move in a fashion coordinated across space and time. This means that each should engage in the same type of motion at the same time. This creates difficulties for current motion synthesis algorithms, because, while the actors should engage in the same type of motion, they should *not* use exactly the same motions. Doing so produces extremely distracting effects.

In principle, Arikan *et al.*'s framework allows this type of synthesis, but does so poorly in practice. This is because the search tends to be quite efficient at finding motions that meet the particular annotation and geometric demand: different actors tend to share the same motion example. In turn, this suggests we can finesse the difficulty with shared motions by synthesizing each actor with a distinct component of the original collection of motions. Of course, this applies only if we have sufficient motion examples to split up the collection safely.

We are able to significantly increase the size of our motion collection with transplantation. Furthermore, a natural property of our process is that it tends to generate many slightly different examples of similar motions. All this suggests that the result of transplantation should allow this type of synthesis in Arikan *et al.*'s framework. This turns out to be successful.

As a base motion collection, we use all the results of transplantation whether annotated with `looks human` or not. As in the experiment in Section 4, we do this because such frames can potentially improve continuity.

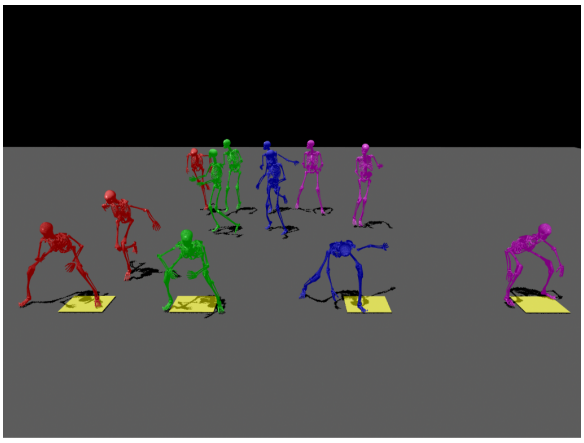

Figure 5: A motion synchronised in time and space. The motion demand specifies actors run from the start point, reach their specified end points and crouch. If one simply synthesizes four motions with these demands, the resulting motions are likely to share frames. By expanding, then splitting, the motion collection using our transplantation strategy, we are able to produce coordinated motions that do not share frames.

We have animated sequences involving four actors in this way. We used only the result of transplantation — 204 sequences in total. For each possible annotation string, we ensure that each actor has at least 30 example frames with that string, and then choose motions uniformly at random so that each actor has approximately a quarter of the total pool of frames. Motions are then synthesized with the techniques of Arikan *et al.*, resulting in sequences where actors (a) obey start and end constraints, (b) obey frame constraints, and (c) can be synchronised in time without sharing frames. Figure 5 shows an example of a synchronised motion.

6. Conclusion

The compositional nature of human motion means that a spanning set of motion examples will probably never be available. Instead, we must search for acceptable methods for deriving good new motions from observed motions. We have shown that good motion sequences can result from removing part of one body and attaching it to another. We used a simple randomized search to increase the size of our pool of motion examples nine times, obtaining both more examples of existing types of motion as well as novel motions in the process.

A motion synthesis technique, however, can only be evaluated in the context of relevant motion demands. We describe a methodology for evaluating our technique using demands generated by hand and from a popular video game. This methodology can also be used to evaluate other synthesis strategies.

There is a substantial advantage to possessing enough

motion examples. We have demonstrated that a sufficiently large pool of examples makes it possible to synthesize motions of groups that are qualitatively synchronized in both space and time without obtaining motions that share frames; if one has too small a pool of motion examples, this strategy fails.

There remains a great deal of research to be done on motion composition. Attractive topics include: obtaining a (near) complete set of rules; understanding what aspects of correlations are passive and what active; and determining when motions should be seen as new motions and when as composites of existing motions.

7. Acknowledgements

We thank Andrew Gardner for gathering and processing the Unreal Tournament 2004 data, and the Berkeley graphics group for their helpful comments. This research was supported by the Office of Naval Research grant N00014-01-1-0890 as part of the MURI program, and by a Berkeley Edge graduate student fellowship. We also would like to thank Electronic Arts for furnishing us with the motion data.

References

[AF02] ARIKAN O., FORSYTH D. A.: Interactive motion generation from examples. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques* (2002), ACM Press, pp. 483–490. 2

[AFO03] ARIKAN O., FORSYTH D., O'BRIEN J.: Motion synthesis from annotations. *SIGGRAPH* (2003). 3, 5

[BRRP97] BODENHEIMER B., ROSE C., ROSENTHAL S., PELLA J.: The process of motion capture: Dealing with the data. In *Computer Animation and Simulation '97. Proceedings of the Eurographics Workshop* (1997). 1

[CL00] CHANG C. C., LIN C. J.: *Libsvm: Introduction and Benchmarks*. Tech. rep., Department of Computer Science and Information Engineering, National Taiwan University, 2000. 5

[FF99] FORSYTH D., FLECK M.: Automatic detection of human nudes. *Int. J. Computer Vision 32* (1999), 63–77. 6

[FMJ02] FOD A., MATARIĆ M. J., JENKINS O. C.: Automated derivation of primitives for movement classification. *Autonomous Robots 12*, 1 (2002), 39–54. 3

[FP03] FANG A. C., POLLARD N. S.: Efficient synthesis of physically valid human motion. *ACM Transactions on Graphics 22*, 3 (July 2003), 417–426. 2

[FvdPT01a] FALOUTSOS P., VAN DE PANNE M., TERZOPOULOS D.: Composable controllers for physics-based character animation. In *Proceedings of ACM SIGGRAPH 2001* (Aug. 2001), Computer Graphics Proceedings, Annual Conference Series, pp. 251–260. 2

[FvdPT01b] FALOUTSOS P., VAN DE PANNE M., TERZOPOULOS D.: The virtual stuntman: dynamic characters with a repertoire of autonomous motor skills. *Computers & Graphics 25*, 6 (Dec. 2001), 933–953. 2

[GL98] GLEICHER M., LITWINOWICZ P.: Constraint-based motion adaptation. *The Journal of Visualization and Computer Animation 9* (1998), 65–94. 1

[Gle97] GLEICHER M.: Motion editing with spacetime constraints. In *Proceedings of the 1997 Symposium on Interactive 3D Graphics* (1997). 1

[Gle99] GLEICHER M.: Animation from observation: Motion capture and motion editing. *Computer Graphics* (1999). 1

[Gle01] GLEICHER M.: Comparing constraint-based motion editing methods. *Graphical Models* (2001). 1

[GSKJ03] GLEICHER M., SHIN H. J., KOVAR L., JEPSEN A.: Snap together motion: Assembling run-time animation. In *2003 Symposium on Interactive 3D Graphics* (2003). 1

[GTH98] GRZESZCZUK R., TERZOPOULOS D., HINTON G.: Neuroanimator: Fast neural network emulation and control of physics-based models. In *Proceedings of SIGGRAPH 98* (July 1998), Computer Graphics Proceedings, Annual Conference Series, pp. 9–20. 2

[HOT97] HODGINS J. K., O'BRIEN J. F., TUMBLIN J.: Do geometric models affect judgments of human motion? In *Graphics Interface '97* (May 1997), pp. 17–25. 1

[HOT98] HODGINS J. K., O'BRIEN J. F., TUMBLIN J.: Perception of human motion with different geometric models. *IEEE Transactions on Visualization and Computer Graphics 4*, 4 (Oct. 1998), 307–316. 1

[HWBO95] HODGINS J. K., WOOTEN W. L., BROGAN D. C., O'BRIEN J. F.: Animating human athletics. In *Proceedings of SIGGRAPH 95* (Aug. 1995), Computer Graphics Proceedings, Annual Conference Series, pp. 71–78. 2

[JM02] JENKINS O. C., MATARIC M. J.: Deriving action and behavior primitives from human motion data. In *Proceedings, IEEE/RSJ International Conference on Robotics and Intelligent Systems (IROS)* (2002), pp. 2551–2556. 3

[JM03] JENKINS O. C., MATARIC M. J.: Automated derivation of behavior vocabularies for autonomous humanoid motion. In *Proceedings, Second International Joint Conference on Autonomous Agents and Multiagent Systems* (2003). 3

[KG03] KOVAR L., GLEICHER M.: Flexible automatic motion blending with registration curves. In *Proceedings of 2003 Symposium on Computer Animation* (2003). 3, 5

[KGP02] KOVAR L., GLEICHER M., PIGHIN F.: Motion graphs. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques* (2002), ACM Press, pp. 473–482. 2, 5

[KSG02] KOVAR L., SCHREINER J., GLEICHER M.: Footskate cleanup for motion capture editing. In *Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation* (2002), ACM Press, pp. 97–104. 1

[LCR*02] LEE J., CHAI J., REITSMA P., HODGINS J., POLLARD N.: Interactive control of avatars animated with human motion data. *Proc. SIGGRAPH 2002* (2002). 2, 3

[LS99] LEE J., SHIN S. Y.: A hierarchical approach to interactive motion editing for human-like figures. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques* (1999), ACM Press/Addison-Wesley Publishing Co., pp. 39–48. 1

[Men99] MENACHE A.: *Understanding Motion Capture for Computer Animation and Video Games*. Morgan-Kaufmann, 1999. 1

[Moe99] MOESLUND T.: *Summaries of 107 Computer Vision-based Human Motion Capture Papers*. Tech. Rep. LLA 99-01, University of Aalborg, 1999. 1

[MTH] MOLINA-TANCO L., HILTON A.: Realistic synthesis of novel human movements from a database of motion capture examples. 2

[PB02] PULLEN K., BREGLER C.: Motion capture assisted animation: Texturing and synthesis. *SIGGRAPH 02* (2002). 3

[PBM00a] POLLARD N. S., BEHMARAM-MOSAVAT F.: Force-based motion editing for locomotion tasks. In *In Proceedings of the IEEE International Conference on Robotics and Automation* (2000). 1

[PBM00b] POLLARD N. S., BEHMARAM-MOSAVAT F.: Force-based motion editing for locomotion tasks. In *Proceedings of the IEEE International Conference on Robotics and Automation* (2000). 2

[PW99] POPOVIĆ Z., WITKIN A. P.: Physically based motion transformation. In *Proceedings of SIGGRAPH 99* (Aug. 1999), Computer Graphics Proceedings, Annual Conference Series, pp. 11–20. 2

[SPB*98] SILAGHI M.-C., PLÄNKERS R., BOULIC R., FUA P., THALMANN D.: Local and global skeleton fitting techniques for optical motion capture. In *Modelling and Motion Capture Techniques for Virtual Environments* (Nov. 1998), pp. 26–40. Proceedings of CAPTECH '98. 1

[TSK02] TAK S., SONG O.-Y., KO H.-S.: Spacetime sweeping: An interactive dynamic constraints solver. In *Computer Animation 2002* (2002), p. 261. 1

[ZH99] ZORDAN V. B., HODGINS J. K.: Tracking and modifying upper-body human motion data with dynamic simulation. In *Computer Animation and Simulation '99* (Sept. 1999). 2

[ZH02] ZORDAN V. B., HODGINS J. K.: Motion capture-driven simulations that hit and react. In *ACM SIGGRAPH Symposium on Computer Animation* (July 2002), pp. 89–96. 2