Data-Driven Approach to Synthesizing Facial Animation Using Motion Capture

Kerstin Ruhland, Mukta Prasad, and Rachel McDonnell ▪ *Trinity College Dublin*

Producing cartoon animations is a laborious task, and there is a distinct lack of automatic tools to help animators, particularly with creating facial animation. To speed up and ease this process, the proposed method uses real-time video-based motion tracking to generate facial motion as input and then matches it to existing hand-created animation curves.

In the entertainment industry, cartoon animators have a long tradition of creating expressive characters that are highly appealing to audiences. It is particularly important to get right the movement of the face, eyes, and head because they are the main communicators of the character's internal state and emotions and an indication of its level of engagement with others. However, creating these highly appealing animations is a labor-intensive task that is not supported by automatic tools. Reducing the animator's effort and speeding up this process, while allowing for artistic style and creativity, would be invaluable to production companies that create content for weekly cartoons and feature-length films.

Animation production typically goes through a number of creation stages: acting (where the animator acts out the expressions in front of a mirror), blocking (where key body and face poses are created), refining pass (where overlapping actions are added, movements are exaggerated, and the timing and spacing of motions is refined), and final polish (where animation curves are cleaned and small details are added). We propose an alternate, more automated solution for use when production speed is most important. Our approach aims to produce facial animation that closely resembles the quality of the refining stage, with easily editable curves for final polish by an animator.

The proposed approach replaces the acting and blocking phases by recording the artists' movements in front of a real-time video-based motion tracking and retargeting system (see Figure 1). We use a commercially available and affordable real-time facial-retargeting system for this stage, which creates the initial motion curves of the character animation. These motion curves contain realistic human movements that are typically dense in keyframes, making them difficult for an animator to edit.

Our approach thus focuses on the next stage of the pipeline. The initial motion curves of the character animation provide the input to our algorithm. Our pattern-matching algorithm replaces the refining stage, by matching the motion curves to a database of hand-animated motion curves. This creates synthesized animations that reflect the artist's keyframing and animation style. In the final stage, the animator can focus on polishing the animation for final production. We expect that artists need to be acquainted with polishing synthesized animations, as opposed to animations of their own work, before such a system can be incorporated into a production pipeline. Therefore, our solution will be mostly suitable for productions that require a lot of animations in a short amount of time (such as weekly TV cartoons).

This article demonstrates our approach's ability to dramatically reduce the number of keyframes of the motion-capture data. We also describe a set of

 Published by the IEEE Computer Society

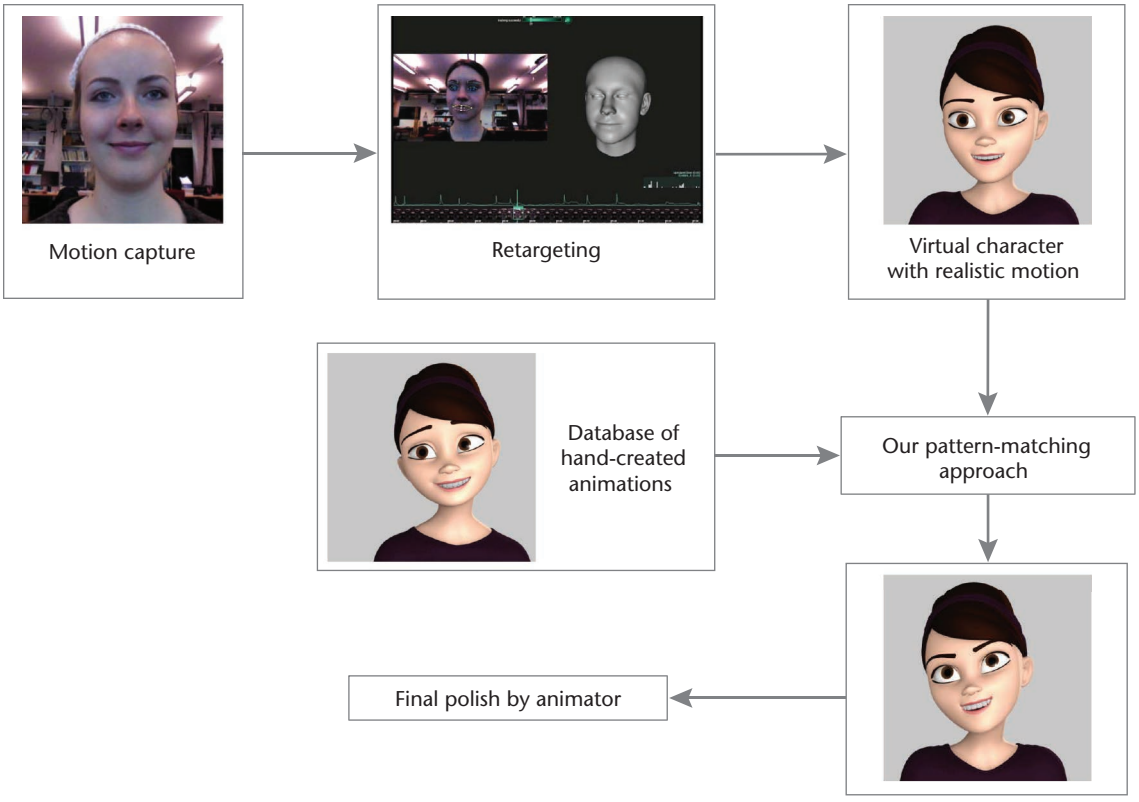

Figure 1. Producing facial animation at the refining stage. A motion-capture system lets us predefine key facial poses and timings. Then, by utilizing a database of hand-created animations, our approach adjusts the motion-captured animations to match the artist's style. The animator can easily edit the animation for final production in the final polish stage.

user studies that show that our synthesized animations can achieve the same expressiveness and cartooniness as hand-created animations and that an animator can improve them with a small amount of polish. (See the web extra video for an illustration of the proposed approach: https://youtu.be/PQP9965jcCk.)

To evaluate our method, we synthesized a series of emotional sentences, using some input motion capture, a database of hand-animated motions, and three different pattern-matching algorithms (distance measurement, symbolic aggregate approximation, and a hidden Markov model). For this work, we were limited to a database consisting of just one minute of high-quality hand-animated content that we commissioned for the purpose of this project. The results show that our approach can effectively synthesize new animations (see Figure 2). The large number of motion-capture keyframes generated from the real-time retargeting, usually one per frame, are replaced by well-placed keyframes and in-betweens.

Motion Capture and 3D Animation

Motion capture is the favored method for applying realistic motion to 3D characters, but the results can appear too realistic and lack expressiveness when applied to highly stylized cartoon characters. Editing motion-capture data is difficult and work intensive because keyframes are created for

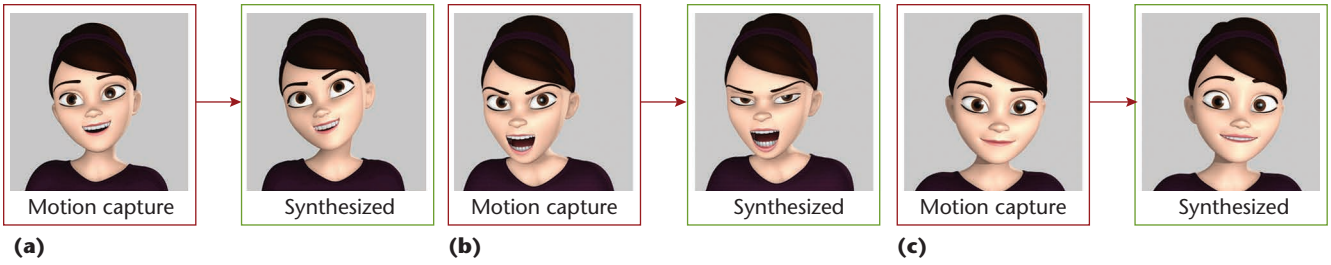

Figure 2. Pattern-matching approach using example curves of professional cartoon animators to match to a motion-capture sequence creating a new synthesized animation. The example emotions are (a) happy, (b) angry, and (c) happy.

each frame, which can limit the animator's creativity. Various keyframe simplification methods have been implemented to make motion capture more accessible.[1] Ideally, any animation tool using motion capture created for an animator should simplify the keyframes. Our method automatically reduces motion-capture keyframes using a pattern-matching approach, so it is well suited as an animator tool.

Professional cartoon animators draw on a wealth of established animation principles to create appealing characters. The literature has comprehensively documented the application of these traditional animation principles to 3D computer animation, explaining their meaning in 2D animation and how they can best be translated into 3D.[2] Squash-and-stretch, for example, makes animations more fluid, while exaggeration and anticipation help to translate the character's emotions and thoughts to the audience.

Our objective is to detect the animator patterns in a given motion-capture curve.

Some previous research focused on synthesizing these principles for 3D objects, for example, by creating squash-and-stretch or exaggerated motions. The Cartoon Animation Filter[3] applies an inverted Laplacian of a Gaussian filter to the motion signal to enrich a variety of motion signals with anticipation, follow-through, exaggeration, and squash-and-stretch. Relatively few papers concentrate on the application of traditional animation principles to virtual characters. Ji-yong Kwon and In-Kwon Lee transformed motion-capture data into rubber-like exaggerated motion by breaking down the original skeleton into shorter segments and applying a Bézier curve interpolation for the bending effect and a mass-spring emulation creating a smooth movement.[4] Jong-Hyuk Kim and his colleagues implemented an anticipation effect added to character animation.[5] With the notion that a main movement is preceded by an opposite movement, they calculated the rotations and translations of the center of mass and then calculated the anticipation pose with a nonlinear optimization technique and incorporated it into the given motion. Eakta Jain and her colleagues utilized an existing motion-capture database to match and compare the 3D poses to hand-drawn 2D character poses, generating a stylized 3D animation.[6] Our proposed method takes the opposite approach by using a database of hand-created stylized motion curves to replace realistic human motion curves.

Although there has been an abundance of work contributing to the automatic generation of realistic facial animation, facial animation stylization has received less attention. Previous attempts have used a 2D approach, typically using video data as input and producing low-resolution facial animation, where details in the face are minimized and the important features are therefore exaggerated. Jung-Ju Choi and his colleagues added the traditional principle of anticipation to motion-captured facial expressions.[7] They used principal component analysis (PCA) to classify facial features into components with similar motions and extract expressions from each component's animation graph. An anticipation effect for a specific expression is found by searching the component's animation graph and finding the best-matching inverted expression, which is blended into the original expression. Unlike this approach, we present a pattern-matching method that matches similar patterns in a hand-created animation to a motion-capture curve. The newly synthesized animation can exhibit an animator's unique style.

Automatically modifying motion-capture curves to add the traditional animation principles is an effective method of changing the style of the realistic motion data and making it appear more cartoon like. However, this approach lacks the input of a trained animator, resulting in somewhat predictable results—for example, the same exaggeration applied to all motions can look repetitive over time. Our aim is to incorporate input from a trained animator in order to create more varied results. We observe that a motion-capture curve retargeted to a virtual character is basically a sequence of data points over a time interval. Therefore, we draw inspiration from the study of time-series data analysis, the goal of which is to detect and estimate patterns and predict future trends. Our objective is similar in that we wish to detect the animator patterns from a given motion-capture curve.

The search for matching patterns in time-series data has been widely researched. The most commonly used strategy to compare two sets of time-series data with each other is the evaluation of Euclidean distance. Dynamic time-warping techniques (DTW) and hidden Markov models (HMMs) have also been successfully applied to mine time-series data. In addition to being robust to noise, offset translation, and amplitude scaling when matching

patterns, Eamonn Keogh's method[8] accounts for longitudinal scaling, which is an important feature for our use. To reduce storage space and speed up the similarity search, Jessica Lin and her colleagues introduced a symbolic aggregate approximation (SAX) that reduces the data's dimensionality and enables indexing with a lower-bound distance measure.[9] The time-series data are approximated by a piecewise aggregate approximation (PAA) model and mapped to SAX symbols generating a word of a predefined alphabet size. The word comparisons are fast and less computationally expensive than the Euclidean distance measure.

Algorithm

Our method takes as input an unseen motion-capture curve and, using a pattern-matching approach, finds similar curve segments in a database of hand-animated motion curves. The motion capture is then adjusted to match the detected hand-animated motion curve. The curves for each facial feature are processed independently. This allows more flexibility, as a more exaggerated stylized animation can be achieved by nonsynchronous movement of facial features, and simpler computation. Our work assumes that the distinctiveness of a hand-created animation is derived from the spacing of keyframes, tangents, interpolation methods, and the timing of the curves.

A prerequisite for our approach is that the input motion capture is retargeted to the same character rig as the database of hand-animated motion curves they are matched to. The character rig itself can be blendshape or bone based. Our proposed method works independently of the motion capture and retargeting technique used, so any state-of-the-art method to capture realistic motion and apply it onto a 3D character can be used. For practicality, a markerless video-based method for real-time capture is preferable because it is cheaper and easier to fit into an existing animation pipeline, rather than the more costly and difficult to configure alternative of optical motion capture. Although the transfer of performance capture data to the virtual model are important, we implemented existing approaches because the main focus of our work is on reusing the motion.

In our approach, each retargeted motion-capture curve describing the *xyz* translation and rotation of the corresponding facial feature or body part (such as head or eyes) along the time axis is referred to as the *motion curve*. These motion curves are compared to curves hand-created by an animator using the same character rig and are referred to as *example curves*.

Our method consists of three main stages. First, the example curves from the database of artist-created animations are divided into chunks of progressively smaller sizes, which will serve as patterns to be found in the motion-capture curve. In the second stage, a pattern-matching approach finds subsegments in the example curve that are similar to subsegments in the motion curve. For comparison, we implemented three straightforward and well-established pattern-matching approaches. The first approach finds the similarity of the curves using a distance measurement between subsegments on the data points of both curves. The second approach transfers both curves into a symbolic representation before comparison, and the third approach uses an HMM to find the best match. We assume that the first approach will achieve more accurate results but will have a higher processing overhead, the second approach will speed up the processing but might introduce errors due to the approximation of the original curves, and the third approach will be superior in warping the motion-capture curve to the animator curve as it models the smoothness of the match but, depending on the parameter definition, might require a longer processing time. The third stage of the method is the continuous adjustment of the motion curve, based on the segments found by the chosen pattern-matching approach.

Example Curve Segmentation

In preparation for our approach, the example curves have to be segmented in order to match subsegments to the motion-capture curve. We create a dictionary of all curve segments from the initial animation curves so that each curve segment is at least k keyframes in length and at most as large as the example curve. In our implementation, we choose a top-down segmentation algorithm, where each example curve in the database is recursively partitioned. Each keyframe of a newly defined curve segment stores the time, translation/rotation value, the type of tangent, and the tangent's in and out angles. For later processing, the time codes of the keys are recalculated to start at zero.

Our method's objective is to transfer the artistic keyframing style of the hand-created animation to the motion curve. Experimental validation shows that the best style transfer results are achieved by transferring as much information as possible from larger artist-created segments. Therefore, the dictionary of curve segments is sorted in decreasing order of length when searching for a pattern in the motion curve.

For scale invariance, we adopted Keogh's approach[8] by including scaling in the x axis. To do so, each subsegment of the example sequence is divided into three parts with smaller left q_l and right q_r subsegments. For each q_l and q_r, a matching pattern in the motion curve is found. New segments are created by taking a match found for q_l and q_r and compressing or stretching the middle part along the x axis. The scaling factor with which the x values of the middle part are multiplied is computed as

$$\text{ScalingFactor} = (q_r(x_1) - q_l(x_1))/q_{\text{length}}$$

where $q_l(x_1)$ and $q_r(x_1)$ are the starting points on the x axis of the left and right subsegments of found matches, respectively, and q_{length} is the original length of the example segment q.

Our method transfers the artistic keyframing style of the hand-created animation to the motion-capture curve.

With the available subsequences, each pattern-matching approach must solve the subsequence matching problem. As we explained earlier, we implemented three different pattern-matching approaches for comparison purposes.

First Approach: Distance Measurement. The simplest and most common approach to determine the similarity between two curves is the distance measurement. Let Q represent a dataset of curve segments gleaned from the original example curve. The example curve segments are represented as $q_s(i)$, where $i \in \{1 \ldots n(s)\}$ denotes the time indices for a segment with a start point that is zero centered to account for translational offset. The 1D motion curve is represented by $p(t)$ at every time index t.

For each point in the example curve segments q_s, the corresponding point in the motion curve P must be found. The distance between a segment q_s and a subsequence p of P starting at r is the sum of squared errors (SSE) calculated over the distance measure between each point in q_s and the corresponding point in p. This can be defined as

$$d(p(r), q_s) = \text{SSE}\{q_s(i) - p(r + (i - 1)) \mid i \in \{1 \ldots n(s)\}\}.$$

A distance of zero describes identical segments, as the distances between the points are equal. The distance between each segment q_s and subsequences of P are stored in an $M \times N$ matrix, where M is the number of subsequences of P and N is the length of Q. The goal of subsequence matching is to find r^* such that

$$r^* = \text{argmin}_r\, d(p(r), q_s)$$

The final synthesized animation curve can be further influenced by introducing a threshold. Instead of an exact search for the minimal argmin_r, an approximate search satisfying some threshold can be performed. A low threshold will search for the best match between the motion curve and the example curves, whereas a higher threshold will allow approximate matches. A combination of a low threshold and small subsegments will closely replicate the motion curve. The higher the threshold, the more the final curve will deviate.

Second Approach: SAX. The second approach applies symbolic aggregate approximation (SAX)[9] to the motion data. SAX approximates a time series by converting the curve into symbols, thus reducing the curve's dimensionality to allow faster comparison. We described only the main steps here. For more detailed information, see earlier research.[9]

Before converting the example curves Q and motion curves P into symbols, they are normalized to have a mean of zero and a standard deviation of one. This is necessary because previous tests have shown that the distance measurement between nonnormalized time-series data is of no relevance due to possible differences in offset and amplitude. Therefore, both curves are converted using a piecewise aggregate approximation (PAA) method. PAA reduces a curve of length n into a representation of w dimensions. The user-defined w specifies the number of PAA segments representing the curve. The curve is divided into w segments of equal size for each segment, and the mean value of the data points is calculated. The quality of this approximation of the original curve can be controlled by the number of segments.

The next step discretizes the data representation further, based on the assumption that normalized time-series data has a Gaussian distribution. Given this knowledge, breakpoints are determined from a statistical table, which divides a Gaussian distribution into a user-defined number (3 to 10) of equiprobable areas. One SAX symbol is mapped to each of these areas. Finally, the PAA coefficients are converted into a symbolic representation by

applying the corresponding SAX symbol depending on the area under the Gaussian distribution curve in which the PAA coefficient falls.

To compare two symbols, SAX provides a lookup table with the distances between two symbols. The minimum distance between two equally sized curves Q' and P' in a SAX representation can then be calculated as

$$\min \operatorname{dist}(Q', P') = \sqrt{\frac{n}{w}} \sqrt{\sum_{i=1}^{w} \left(\operatorname{dist}\left(q_i', p_i'\right)\right)^2} .$$

The comparison of curves of varying length is implemented by using a sliding-window approach. As in the previous approach, we can introduce a threshold to influence the result of the synthesized animation curve.

Third Approach: HMM-Viterbi Algorithm. In the last approach, the first curve is compared with the second, based on both the quality of individual feature matches and the smoothness of the overall mapping across all points. This problem is formulated using an HMM, where the example curve Q forms the chain graph nodes and the keyframes of the motion curve P are the states that each node on the first curve can take (denoting matches). The model parameters consist of the initial state distribution I, the unitary emission matrix E, and the pairwise transition matrix T.

The emission matrix $E(P \times Q)$ usually defines how likely it is that qth example nodes in the graph can take the pth motion curve label based on similarity between the features. For each point $p(x)$ in the motion and $q(x)$ in the example curve, a feature vector is defined based on curve characteristics. For example, for P

$$\phi\left(p(x)\right) = \begin{bmatrix} p(x) \\ p'(x) \\ p''(x) \end{bmatrix}.$$

The aim is to match each point in the example curve to a point in the motion curve that is similar. The distance between the mathematical representation of both points in terms of features defines their similarity:

$$d(p(i), q(j)) = ||\phi(p(i)) - \phi(q(j))||$$

where $i \in \{1 \ldots p_m\}$ and $j \in \{1 \ldots q_n\}$.

The pairwise transition matrix $T(P \times P)$ defines how likely it is for two neighboring nodes to take one particular combination of two state labels. Without temporal smoothness, correspondences can be determined, but using one helps us define smoothly varying, monotonic correspondences. The initial state distribution I will be constrained to always start with the first segment $I = [1, 0, 0, \ldots, 0]$. To account for the fact that the two curves have different overall scales (similar to Keogh's approach[8]), we devised a normalization factor λ based on the ratio of overall lengths, as constant:

$$\lambda = \frac{\sqrt{\sum_{i=1}^{P} \left(p(i) - p(i+1)\right)^2}}{\sqrt{\sum_{j=1}^{Q} \left(q(j) - q(j+1)\right)^2}} .$$

If $T(i, j, k)$ is the cost of going to state k in node $i + 1$ when the process is in state j in node i, the transition matrix takes the following form:

$$T(i,j,k) = \begin{cases} \left| \sqrt{\left(p(j) - p(k)\right)^2} - \lambda * \sqrt{\left(q(i) - q(i+1)\right)^2} \right| & ,k > j \\ \infty & ,k \le j \end{cases} .$$

In this implementation, we are interested in finding the most probable correspondence assignment (between the motion and example curves) so that the correspondences are smooth and monotonic in time. Thus, T is right triangular and the rest of the elements have infinite cost. Such an HMM solved with the Viterbi algorithm is similar to a dynamic time warping (DTW) formulation. Here the parameters are empirically set, but in the presence of more data, these can be learned in a maximum likelihood framework.

Curve Warping

Once the best match is found, the motion curve is warped. Each subsequence r^* of the motion curve is manipulated such that the motion curve segment r^* matches the corresponding example curve segment q_s as closely as possible. This can be written as a warping function with the given parameter θ:

$$\min_{\theta} d\left(w\left(p\left(r^*, \theta\right), q_s\right)\right)$$

To ensure a smooth transition between two successive segments, endpoint constraints are defined as

$$p(r^*) = w(p(r^*), \theta),$$
$$p(r^* + n(s) - 1) = w(p(r^* + n(s) - 1), \theta).$$

and endpoint tangent constraints are defined as

$$p'(r^*) = w'(p(r^*), \theta),$$
$$p'(r^* + n(s) - 1) = w'(p(r^* + n(s) - 1), \theta).$$

These functions denote C^1 continuity at joining points.

A simplified approach is to compute the difference of the values on the y axis of each point in matching subsegments q_s and r^*. The deviation of the values in r^* from the values in q_s defines how much the motion curve has to be warped (implemented similar to Poisson interpolation). All points in the motion curve that are not part of a matching pattern are deleted.

Evaluation

For our evaluation, we chose a stylized female character that resembles characters from top animation studios and features blendshapes and squash-and-stretch functionalities needed to create convincing cartoon animation. For all our experiments, the lip-syncing was directly derived from the motion-capture animation and not altered by our approach. We chose this approach because lip motion is complex and is controlled by multiple blendshapes and controllers. To synthesize animation of this level of complexity, we would need a much larger database. Furthermore, the use of automated lip-sync software (from audio) is already a commonplace approach to speeding up animation production of lip movement, so we do not attempt to tackle this particular issue here.

Animation Database

For testing, we commissioned a small database (just 1 minute) of high-quality artist-created animation. The cost of commissioning this database was approximately €6,000, due to the amount of time required for the animators to create it. The expense of this small animation database highlights the need for more cost-effective production of cartoon facial animation and data reuse. Ideally, our approach would be used in a production setting, where high-quality pregenerated animation would be readily available.

The database consisted of two emotional 30-second monologues (anger and happiness), for which we used a professional female actor and recorded the audio. The professional 3D character animator we commissioned created high-quality cartoon animations in his own style following the provided audio clips.

Motion-Captured Sequences

To test our approach, we generated a set of motion-capture sequences that were used as input. To provide expressive content, the dataset consisted of animations that convey anger and happiness, two of the six basic emotions in Paul Ekman's classification of emotional speech.[10] We chose a sentence for each emotion from previously validated lists of affective sentences for emotion identification through content.[11,12]

A state-of-the-art video-based head-mounted camera system was used to capture the professional female actor's facial movements while she acted out the two sentences. The motion capture was then applied to the character rig using the real-time, example-based retargeting software Faceware (facewaretech.com). Faceware analyzes the video input from the head-mounted camera and automatically tracks 37 feature points (five for each brow, four for the outer eye, two for the pupil, 14 for the outer lip, and 12 for the inner lip). The motion is then retargeted to the user-defined controls (including 14 facial blendshape controllers and additional controllers for the eyes, head, neck, and chest). The final curves are the *xyz* translation and rotations of these controllers over time. From the recordings, the best interpretations were hand-picked and were each approximately 4 seconds long.

Synthesized Sequences

For the synthesized motions, the motion-captured sequences were used as the "unseen input," and the algorithms searched the animation database for matches. The three proposed approaches (distance measure, SAX, and HMM) were implemented as a plug-in for Autodesk Maya using the Maya Python API 1.0 interface.

Keyframe Reduction

The aim of our approach is to synthesize animation curves, which reflect the keyframing and animation style of an animator. Keyframe reduction is achieved by adjusting the motion-capture curve to the pattern found in the hand-created animations. Both the motion-captured and synthesized animations were approximately 4-seconds long with 30 frames per second.

Figure 3 shows the average number of keyframes for the motion-captured curve and the three different pattern-matching methods. Because we used retargeting software, the number of keyframes differs slightly among the different motion-capture curves. All the tested methods achieved a keyframe reduction of approximately 90 percent of the motion-capture curves. The number of keyframes depends on the matching patterns found and therefore varies among the different methods.

By reducing the number of keyframes, we are able to transform the formerly difficult-to-edit motion-capture curves into those that are much

easier for an animator refine in the final polishing stage.

Perceptual Tests

To evaluate how the synthesized results (produced by applying the three different methods) compared with the animations created by the artists and the original motion capture, we conducted a set of user studies. The stimuli set consisted of 20 animations: two examples each of the five animation types (artist-created, motion capture, distance measure, SAX, and HMM) portraying both the anger and happiness emotions.

Experiment 1: Unrefined

Our first experiment consisted of three blocks displaying animations in random order with three repetitions. The first two blocks showed each emotion at a time without sound. The sound was muted for this part of the experiment to avoid influencing the participant's judgment of the character's motion. The third block showed all animations, with audio, in random order. To avoid ordering effects, the first group viewed the angry animations first, followed by the happy animations, and the second group viewed them in reverse order.

In total, the participants were presented with 10 animations (5 animation types × 2 examples × 1 emotion) in blocks one and two and 20 animations (5 animation types × 2 examples × 2 emotions) in block three, with three repetitions in each block. University ethical guidelines were followed as we administered this experiment. Eighteen participants (seven female and 11 male), aged 19 to 30, took part in the experiment, which lasted approximately 20 minutes. The participants were not trained animators and had different disciplinary backgrounds, were recruited mainly from the university student and staff mailing lists, and were all unaware of the intention of the experiment.

After the participants read and signed the consent form, they viewed each animation in the first two blocks and were asked to answer three questions after each clip:

- How expressive was the virtual character in portraying the emotion? The participants were asked to base their decision on the strength of their impression of the character's indicated emotion using a seven-point Likert scale, where 1 was not expressive at all and 7 was extremely expressive.
- How appealing did you find the virtual character's motion? Participants were advised to base their decision on the appeal of the character's overall motion using a seven-point Likert scale, where 1 was not appealing at all and 7 was extremely appealing.
- How cartoony did you find the virtual character's motion? Cartoony was described to the participants as a highly stylized cartoon animation, similar to what would be seen in animated movies with highly exaggerated and animated character facial expressions. In their answers, the participants used a seven-point Likert scale, where 1 was not cartoony at all and 7 was extremely cartoony.

The participants used the number keys on the keyboard to input their ratings.

After completing the first two blocks, the participants completed the third block and then were asked a final question:

- How good was the facial expression at conveying the message? The participants were asked to concentrate on their overall impression of the character's facial and head movement and how well it conveyed what the character was trying to say, Again, the participants used a seven-point Likert scale, where 1 was not good at all and 7 was extremely good.

Because we used the motion-captured lip movements for all the clips, we asked the participants not to focus on how well the lip movements matched the audio.

For the analysis, we averaged the data over the two examples and the three repetitions. A repeated measures analysis of variance (ANOVA) was conducted for each question with the animation type (artist-created, motion capture, distance measure, SAX, and HMM), emotion (anger and happiness)

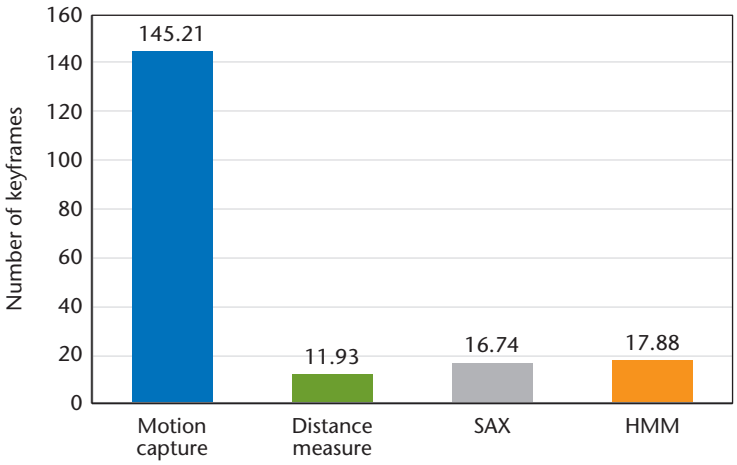

Figure 3. Average number of keyframes for a 4-second animation clip for the motion captured curve and the three different pattern-matching methods. All the tested methods achieved a keyframe reduction of approximately 90 percent.

Table 1. Main results of experiment 1, unrefined.

Variable	Effect	*F*-test	Post hoc analysis
Expressiveness	Animation type	$F(4, 64) = 36.633$, $p \approx 0$	Artist-created animations were rated most expressive; synthesized and motion-capture animations were similar.
	Emotion	$F(1, 16) = 8.434$, $p = 0.0103$	Angry emotions were more expressive than happy.
	Animation type and emotion	$F(4, 64) = 12.992$, $p \approx 0$	Artist-created and synthesized HMM angry animations had similar ratings.
Appeal	Animation type	$F(4, 64) = 21.588$, $p \approx 0$	Synthesized animations were relatively appealing, but less than artist-created animations.
Cartooniness	Animation type	$F(4, 64) = 13.640$, $p \approx 0$	Artist-created animations were rated most cartoony.
	Animation type and participant sex	$F(4, 64) = 4.109$, $p = 0.005$	Female participants rated the HMM and artist-created animations similarly.
	Animation type and emotion	$F(4, 64) = 6.166$, $p = 0.0003$	HMM was rated best at synthesizing the animation style from the artist-created animations.
Expression of the message	Animation type	$F(4, 64) = 46.031$, $p \approx 0$	Artist-created and motion-capture animations were rated better at conveying the message than synthesized animations.
	Emotion	$F(1, 16) = 18.335$, $p = 0.001$	Anger was better expressed than happiness.
	Animation type and emotion	$F(4, 64) = 2.916$, $p = 0.028$	Motion-capture, distance measure, and HMM were rated similarly for angry animations; synthesized happy animations were rated the lowest.

within-subject factors as well as the participant sex between-subject factor. For all post hoc analysis, Newman-Keuls tests were conducted for a comparison of means. Table 1 summarizes all the significant main effects.

The post-hoc analysis showed that the artist-created animations were rated by far the most expressive ($p < 0.0002$ for all). Between the other animation types, the results show only a significant difference between the distance measure and the HMM method ($p = 0.0148$), which indicates that the synthesized animations using HMM were perceived as more expressive than the synthesized animations using the distance measure. However, in general motion capture and all three synthesized animations were found to be similarly expressive.

We also found angry emotions to be more expressive than happy ($p = 0.0086$). An interaction between animation type and emotion shows the differences between the expressiveness ratings in more detail. Post hoc tests showed that there was no difference in the ratings of the artist-created and synthesized HMM animations for the angry emotion and that both differ significantly from the other animations ($p < 0.012$), indicating that the unrefined synthesized angry animations using the HMM method were as expressive as the artist-created animations. On the other hand, for the happy emotions, the artist-created and motion-capture animations were significantly more expressive than the synthesized animations ($p < 0.036$ for all).

The artist-created animations received the highest ratings (mean 5.25) in terms of appeal ($p < 0.0004$ for all). The motion-capture (mean 4.49) and distance measure (mean 4.2) animations received similar ratings, whereas the SAX (mean: 3.55) and HMM (mean 3.67) animations received the lowest ratings. No other main effects or interactions were found. This implies that participants found the unrefined synthesized animations relatively appealing, but less so than the artist-created animations. This is unsurprising because the animations were generated directly from the algorithm and had not yet undergone a final polishing by an animator.

The artist-created animations were also perceived as the most cartoony ($p < 0.0002$ for all). However, this was more true for male participants, whereas the female participants rated the unrefined HMM animations equally highly as the artist-created on cartooniness. An interaction was found between animation type and emotion (Figure 4).

As seen with expressiveness, the post-hoc analysis shows no difference between the high ratings of the artist-created and HMM synthesized angry animations for cartooniness, indicating that the HMM method was the best at synthesizing the animation style from the artist-created animations. However, for the happy emotion, the ratings for the artist-created animations were significantly different than all other animations ($p < 0.0001$ for all).

Lastly, our results show that anger was better expressed than happiness ($p = 0.001$). The artist-created and motion-capture animations were found to be significantly better at conveying the message than the synthesized animations ($p < 0.003$ for all). An interaction was found between animation type and emotion, indicating that

Table 2. Main results of experiment 2, refined.

Variable	Effect	*F*-test	Post hoc analysis
Expressiveness	Animation type	$F(6, 72) = 10.123$, $p \approx 0$	No significant improvement
	Animation type and emotion	$F(6, 72) = 5.7226$, p = 0.00007	Angry HMM animations were rated similarly to artist-created animations; refined happy animations using distance measure were rated higher than unrefined animations
Appeal	Animation type	$F(6, 72) = 11.364$, $p \approx 0$	No improvement
Cartooniness	Animation type	$F(6, 72) = 6.881$, $p = 0.00001$	Artist-created animations were rated significantly more cartoony
	Animation type and emotion	$F(6, 72) = 5.521$, $p = 0.00009$	Refined angry HMM animations were rated as cartoony as artist-created animations
Expression of the message	Animation type	$F(6, 72) = 14.201$, $p \approx 0$	Refined animations conveyed message as well as motion-capture animations
	Animation type and emotion	$F(6, 72) = 5.202$, $p = 0.0002$	Refined happy animation using distance measure were rated similarly to artist-created animations

artist-created angry and happy animations were the best at conveying the message ($p < 0.0002$ for all). No difference was found among the motion-capture, distance measure, and HMM angry animations. However, the analysis of the happy animations does show a significant difference between the high ratings of the motion-capture animations and the low ratings of the synthesized animations ($p < 0.005$).

Thus, although in general the hand-created artist animations received significantly better results, our unrefined HMM approach was found to be as expressive and cartoony for the angry animation. The synthesized animations were not intended to be final polished animations for production, whereas the artist-created and motion-captured animations were. We therefore tested if a quick polishing of the animations by a professional animator would improve the ratings of the synthesized animations.

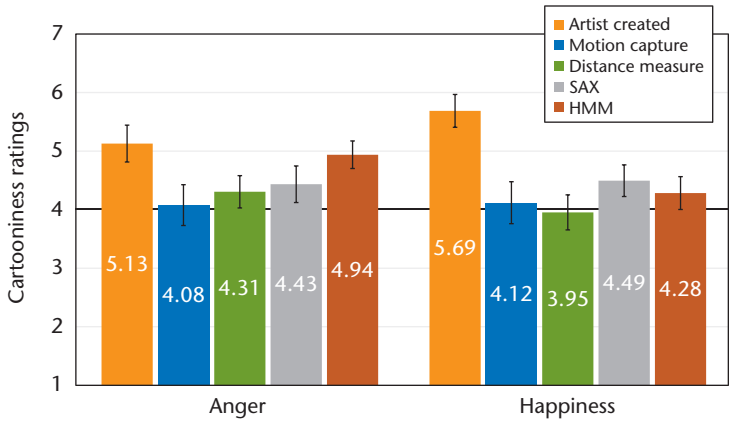

Figure 4. Cartooniness ratings for the unrefined experiment. The bars show the expressiveness ratings averaged over all the participants for each emotion.

Experiment 2: Refined

To investigate if the results for the synthesized animations could be easily improved, we asked a professional animator to refine a subset of the synthesized animations. For refinement, we chose the animations that were rated the best in the unrefined experiment. In total, we asked the animator to refine five animation clips (two happy and three angry synthesized animations) and fix artifacts, improving the overall animation. The animator worked for approximately 30 minutes on each animation clip, making small refinements. The mouth animation was excluded from the refinement because we used the motion-capture lip movements for all the clips.

Next, we conducted a user study with 14 of the participants who took part in the previous experiment (six female and eight male). The participants were asked to rate the refined animations using the same set of questions. The stimuli for this experiment consisted of the five refined animations: one angry animation synthesized using distance measure, two angry animations synthesized using HMM, one happy animation synthesized using the distance measure, and one happy animation synthesized using HMM. As before, the animations were shown in three blocks in random order with three repetitions. The experiment lasted approximately 15 minutes.

For the analysis, we averaged the data over the three repetitions and the two examples. To compare the new ratings with the ratings of the previous experiment, a two-way repeated measures ANOVA was conducted on the data with animation type (artist-created, motion capture, distance measure, SAX, HMM, distance measure refined, and HMM refined) and emotion (anger and happiness) within-subject factors. Table 2 summarizes all significant main effects.

In terms of expressiveness, we found a small, but not significant improvement in the ratings for the

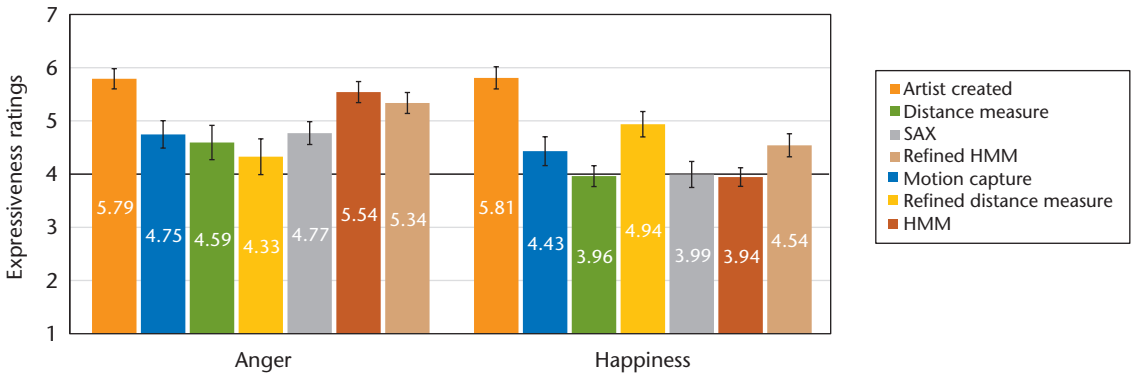

Figure 5. Expressiveness ratings for the refined experiment. The bars show the expressiveness ratings averaged over the participants for each emotion.

refined animations. As before, the artist-created animations were perceived as the most expressive ($p < 0.0003$ for all). An interaction between animation type and emotion provides a more detailed analysis. For the angry animations, no difference was found between the artist-created and the unrefined and refined synthesized HMM animations. For the happy animations, the refined synthesized animations using distance measure were rated significantly higher than the unrefined animations ($p < 0.028$), as Figure 5 shows. Therefore, there was an improvement in expressiveness for the refined animations.

Our results show the refinement did not improve the appeal of the animations. The two refined animations were rated equally high on appeal as the motion-capture animations, but therefore were still also equal to the unrefined animations.

The artist-created animations were still found to be significantly more cartoony than any other animation ($p < 0.0004$ for all). An interaction was found between animation type and emotion, where post hoc analysis shows that the angry refined animations using HMM were found to be as cartoony as the artist-created animations ($p > 0.065$). Both refined happy animations were perceived as less cartoony than the artist-created animations ($p < 0.0004$ for both). No difference was found between the unrefined and refined angry and happy animations, implying that the refinement did not improve cartooniness.

Lastly, the artist-created animations ($p < 0.0005$ for all) still received the best ratings in terms of the expression of the message. However, the refined animations were found to convey the message as well as the motion-capture animations ($p > 0.212$) and significantly better than the unrefined animations ($p < 0.005$ for all). Furthermore, the interaction between animation type and emotion shows that the refined happy animation using distance measure received similar ratings as the artist-created animations ($p = 0.469$) and significantly better than the unrefined animations ($p = 0.0002$). The happy animations using HMM were only found to be similar to the unrefined distance measure and motion-capture animations ($p > 0.084$). For the angry animations, we found no differences between the unrefined and refined motion-capture animations.

Our comparison of the three pattern-matching methods revealed no significant differences. They achieved good results in keyframe reduction and were rated to be similar in terms of quality in our user studies. The functional difference between the methods is their similarity search. Distance measure is the most accurate due to the point-to-point distance calculation. When using SAX, it is important to find the correct combination of the number of segments representing the curve and the number of SAX symbols. A high abstraction of the curves can result in less suitable matching patterns. HMM has been found to be the best in warping the motion-capture curve to the artist-created curve by finding the best matching pairs of keyframes.

The results of our perceptual tests showed that the unrefined animations created by our approach received good ratings in terms of expressiveness, cartooniness, and appeal for the animations with an angry emotion. With some minor refinement by a professional animator, we achieved significantly better results for the expressiveness and expression of the message for the happy animations.

The general trends in research show that the method shown here (and those in other, more complex, machine-learning applications) do generalize and perform better, but the choice of model and the amount of data is crucial for learning. In our tests, our database consisted of just one minute of artist-created animation, and our results were still good. With more publicly available data, the methods and results should only improve.

One drawback of our approach is that the resulting synthesized animations sometimes contain artifacts. Processing curves individually means that the overall animation can exhibit uncoordinated movements or unnatural timing. However, these artifacts are generally easy for an animator to edit, remove, and refine in the final polishing stage. In future work, we plan to extend our method to handle curves jointly.

In the future, we could also create a more sophisticated curve segment dictionary to improve the accuracy and matching speed, without increasing the memory constraint. This could be achieved by segmenting the curves based on salient points or using a graph simplification method[1] to optimize the set of keyframes before segmentation. Going forward, we would like to extend this work to use machine learning on more data and a larger range of emotions to test generalizability. Recent deep-learning techniques can exploit large amounts of data, even with annotations on a small portion. While an initial unsupervised learning algorithm learns features, a subsequent supervised phase could learn to perform a task such as curve mapping.

Acknowledgements

This research was funded by Science Foundation Ireland under the ADAPT Center for Digital Content Technology (grant 13/RC/2106) and the Cartoon Motion Project (12/IP/1565). We thank the Mery Project for the character model and rig.

References

1. Y. Seol et al., "Artist Friendly Facial Animation Retargeting," *ACM Trans. Graphics*, vol. 30, no. 6, 2011, article no. 162.
2. J. Lasseter, "Principles of Traditional Animation Applied to 3D Computer Animation," *Proc. 14th Ann. ACM Conf. Computer Graphics and Interactive Techniques* (Siggraph), 1987, pp. 35–44.
3. J. Wang et al., "The Cartoon Animation Filter," *Proc. ACM SIGGRAPH 2006 Papers*, 2006, pp. 1169–1173.
4. J.-Y. Kwon and I.-K. Lee, "Exaggerating Character Motions Using Sub-Joint Hierarchy," *Computer Graphics Forum*, vol. 27, no. 6, 2008, pp. 1677–1686.
5. J.-H. Kim et al., "Anticipation Effect Generation for Character Animation," *Advances in Computer Graphics*, 2006, pp. 639–646.
6. E. Jain, Y. Sheikh, and J. Hodgins, "Leveraging the Talent of Hand Animators to Create Three-Dimensional Animation," *Proc. 2009 ACM SIGGRAPH/Eurographics Symp. Computer Animation*, 2009, pp. 93–102.
7. J.-J. Choi, D.-S. Kim, and I.-K. Lee, "Anticipation for Facial Animation," *Proc. 17th Int'l Conf. Computer Animation and Social Agents*, 2004, pp. 1–8.
8. E. Keogh, "Fast Similarity Search in the Presence of Longitudinal Scaling in Time Series Databases," *Proc. 9th Int'l Conf. Tools with Artificial Intelligence* (ICTAI), 1997, pp. 578–584.
9. J. Lin et al., "A Symbolic Representation of Time Series, with Implications for Streaming Algorithms," *Proc. 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, 2003, pp. 2–11.
10. P. Ekman, "An Argument for Basic Emotions," *Cognition & Emotion*, vol. 6, nos. 3–4, 1992, pp. 169–200.
11. B.M. Ben-David, P.H. van Lieshout, and T. Leszcz, "A Resource of Validated Affective and Neutral Sentences to Assess Identification of Emotion in Spoken Language after a Brain Injury," *Brain Injury*, vol. 25, no. 2, 2011, pp. 206–220.
12. O. Martin et al., "The eNTERFACE'05 Audio-Visual Emotion Database," *Proc. 22nd IEEE Int'l Conf. Data Eng.*, 2006; doi:10.1109/ICDEW.2006.145.

Kerstin Ruhland *is a PhD student at Trinity College Dublin. Her research interests include realistic facial and eye animation. Ruhland has a diploma in computer science from the Ludwig Maximilian University of Munich. Contact her at ruhlandk@tcd.ie.*

Mukta Prasad *is an adjunct assistant professor of creative technologies at Trinity College Dublin and a research lead at Daedalean AG (an autonomous flight startup). Her research interests include computer vision, machine learning, and graphics. Prasad has a PhD in computer vision from the University of Oxford. Contact her at mp@daedalean.ai.*

Rachel McDonnell *is an assistant professor of creative technologies in the School of Computer Science and Statistics at Trinity College Dublin. Her research interests include computer graphics, character animation, and perceptually adaptive graphics, with a focus on emotionally expressive facial animation and perception. McDonnell has a PhD in computer science from Trinity College Dublin. Contact her at ramcdonn@tcd.ie.*