Pacific Graphics 2012
C. Bregler, P. Sander, and M. Wimmer
(Guest Editors)

Volume 31 (2012), Number 7

Retrieval and Visualization of Human Motion Data via Stick Figures

M. G. Choi[1,2] K. Yang[3] T. Igarashi[2,4] J. Mitani[2,5] and J. Lee[3]

[1]The University of Edinburgh, UK [2]JST ERATO, Japan [3]Seoul National University, South Korea
[4]The University of Tokyo, Japan [5]University of Tsukuba, Japan

Abstract

We propose 2D stick figures as a unified medium for visualizing and searching for human motion data. The stick figures can express a wide range or human motion, and they are easy to be drawn by people without any professional training. In our interface, the user can browse overall motion by viewing the stick figure images generated from the database and retrieve them directly by using sketched stick figures as an input query. We started with a preliminary survey to observe how people draw stick figures. Based on the rules observed from the user study, we developed an algorithm converting motion data to a sequence of stick figures. The feature-based comparison method between the stick figures provides an interactive and progressive search for the users. They assist the user's sketching by showing the current retrieval result at each stroke. We demonstrate the utility of the system with a user study, in which the participants retrieved example motion segments from the database with 102 motion files by using our interface.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation H.5.1 [Information Interfaces and Presentation]: Multimedia Information Systems—Animations H.3.3 [Information Storage And Retrieval]: Information Search and Retrieval—Retrieval models

1. Introduction

Since human motion data is spatio-temporal, looking into this data to understand the contents or finding a specific part of them requires some effort from the users watching the animation from first to last frame while changing the viewing angles. As the size of the database increases, the problem becomes more serious. Therefore, most public motion libraries use keywords as the medium for browsing and searching through motion data [CGL03]. The user can look through the database by reading the keywords and directly retrieve the motion data by typing one of a keyword as an input query.

However, the keywords have obvious limitations as the medium for motion data. The keywords may be subject to the person who decided on them. In addition, a few words are not sufficient to explain and identify many parts of human motion. For example, when there are two different free-styled dances in the database, it would be hard to decide on a proper keyword for each, which will uniquely describe the dances well and will be easy for the user to think of when he or she tries to directly retrieve it with a keyword. If we value the success rate of the retrieval, a choice of compromise would be one general word such as "Dance" or "Free-Style Dance" for both.

In terms of motion browsing, displaying static images of well-selected key postures would be a good alternative. The key postures can be selected in procedurally [ACCO05], so we do not need to rely on a subjective decision. Anonymous motion and even meaningless movements can be represented in this way, and the user can understand the contents of the motion data quickly and intuitively by looking through the key postures in time sequence. However, with this method it is difficult to support a direct retrieval interface with any input query. When the database size is large, searching for a large volume of the images would again be a burden for the users. In this paper, we suggest stick figures as a unified medium for both browsing and searching for human motion data (Figure 1). The stick figure is a simple drawing style, which is used to depict a brief human motion. It is very easy to draw, even as children can do it without any training, and can expressive a wide range of human motion well. In our interface, each motion data file in the database is visualized as a consecutive stick figure image, each of which describes

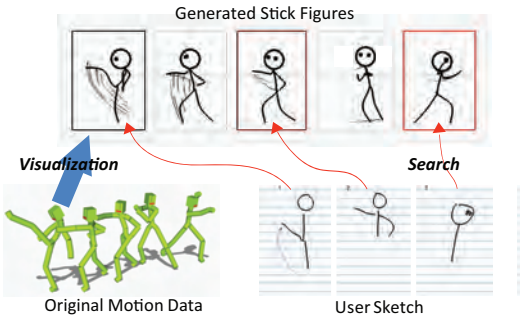


Figure 1: *The 3D motion data is visualized as 2D stick figure images. The generated stick figures can be retrieved by using sketched stick figures as an input query.*

a selected key posture. The user can browse the database by viewing the *generated* stick figures and retrieve a consecutive sequence of them by using *sketched* stick figures as an input query.

The challenge in the motion retrieval is to reduce the difference between the users' sketches and generated images of the same motion as much as possible. Visualizing the motion as the same style as the user's sketches is the key idea to achieve this goal. Not only the drawing style, but also the procedures generating the image imitates people's drawing. We conducted a preliminary survey, in which the participants were asked to explain given motion data through their hand-sketched stick figures. The observed rules were applied to our stick figure generating algorithm.

For the robust comparison between a generated stick figure from data and a noisy sketch, we also introduce a feature-based method, which enable to use an incomplete stick figure image as an input query and conduct thousands of comparisons in interactive response time. In our interface, the retrieval result is updated immediately, every time when the user adds or removes a stroke, so that the user can edit the sketch quickly when the result does not produce the intended motion.

To demonstrate the utility of our approach, we constructed a large motion database with 102 motion data files and generated 2308 stick figure images. With this database, we conducted a user study in which users retrieved given motion segments by using our interface to show the usefulness of our method.

2. Related Works

Visualizing human motion data has been interesting subject because of its spatio-temporal domain and high-dimensionality. Bouvier-Zappa et al. [BZOP07] added cartoon-like signs such as speed lines and noise waves to the character body to show a short past motion of the current moment of the motion. Yasuda et al. [YKSN07] displayed densely sampling key postures on the horizontal time-bar. The user can see the change of the postures by looking through it from left to right. Assa and his colleagues [ACCO05] introduced a systematic method selecting representative key postures from motion data. We also visualize motion data by displaying the selected key posture with speed lines in a raw, but the main difference is that our images will be compared to hand-sketched images for the retrieval. We try to make the images as similar to people's hand drawing of human motion as possible.

The complexity of motion data also makes difficult its retrieval using a simple input query. The motion clip itself would be the finest query to search for similar motion clips [MRC05, MBS09, DGL09, KTWZ10]. For example, Müller et al. [MRC05] defined and extracted the simple features from motion data and then used it for an efficient comparison between different motion data. Deng et al. [DGL09] suggested the partial matching of the motion, in which the query can be a single part of body motion (e.g. left leg) or a mixture of body parts in different motion clips. In these methods, however, if the user did not have the query motion which is similar to the desired one, the only way to get it would be to capture it directly. To reduce the effort of the motion capture, a human-like puppets can be used [cFGDJ08, NNSH11]. Because the queries may not require high quality motion data, the capture systems for the puppets can be much simpler than real-human motion capture system. However, the user still should take the cost for new devices. Our interface uses hand-sketched stick figures as the input queries, and they can be sketched by using a general mouse or tablet.

Sketch input, in the character animation, are mostly used to define a trajectory constraint, which the character or its specific joint point should move along [LCR*02, MCC09, LZ10]. Thorne et al. [TBvdP07] used predefined gestures each of which corresponds to a specific style of locomotion. By sketching multiple gestures continuously on the 3D environment, the user can define both constraints, the travelling path and walking styles. The kind of locomotion is limited by the number of defined gestures.

Since stick figures are as simple as people can draw without training, they have been used as the user input for posture constraints. Pan et al. [PZ11] developed a system in which the user can set the key-postures of a 2D character animation by sketching its skeletal structure. There also have been studied of reconstructing a 3D key-posture from a 2D sketched stick figure without the knowledge of the captured database [DAC*03, MQW08, LIMS10]. On the other hand, we use the input stick figures as queries to retrieve similar motion segments from the large database. Wei end Chai [WC11] developed a likelihood model of 3D character poses given stick figure-like 2D strokes. Because the model was learned from a large motion data, this system also could be used an interactive static posture retrieval system. Our definition of stick figures includes the moving trajectories of

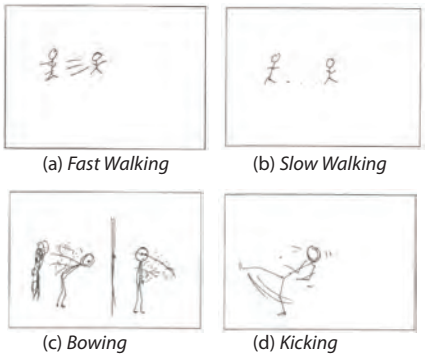


Figure 2: *The four results of preliminary survey. The participants drew stick figures depicting given motion data. All viewing angles are perpendicular to the moving direction. To show the fast movement, speed lines are used. For the bowing and kicking, the postures when the moving direction is changing are selected as key postures.*

joint points, so that the user can depict and retrieval a dynamic motion as well as a static posture directly.

3. How People Draw Stick Figures

To observe how people draw stick figures when they explain human motion, we conducted a preliminary survey. The participants were 15 Master's and PhD students studying computer science. We provided the example motion clips to them and asked to draw stick figures that explain the given motion clips. The motion clips were bowing, slow/fast walking, slow/fast kicking, and tunning kick. We did not limit the numbers of stick figures for a motion clip, and they were allowed to add any auxiliary signs as necessary. Figure 2 shows four images of them.

Although the images were quite different in appearance depending on the participant, we could find common rules of the basic components such as selected key postures, added path lines, and viewing angle.

Key Postures The key postures are mostly selected at the moment when the motion changes the direction or stops. For example, the key postures of the kicking motion were at the moment when the kicking foot is at the peak and the kicking is finished.

Moving Path In some motion data, the points of changing the moving direction are not clear. For example, while the hand is moving along a circle, the direction changes continuously. In these cases, participants usually added a curved line to depict the moving trajectory. In addition, when the motion is relatively fast, multiple lines were drawn along the trajectory like speed lines in comics.

View Plane The view planes selected by the participants were always an arbitrary side view. In terms of the camera, its tilt and elevation angles are fixed to zero, and only the azimuth angle changes. The normal of the plane is decided by considering both the shape of the key posture and the direction of the motion. For example, when the motion data is a motionless standing motion with outspread arms, the front view, which faces the actor, was used mostly since it gave the broadest image of the posture. On the other hand, for the walking motion, the side view, which is perpendicular to the moving direction, was used mostly since it emphasizes the movement well.

We could also see the inconsistency between stick figures of the same motion data. The body proportions of the stick figures varied especially, depending on the person. For example, one drew much shorter arms than legs, but another drew a torso longer than each limb. Sometimes a limb that was regarded as less important in the motion was drawn in an arbitrary way, regardless of the truth (for example, arms in kicking).

4. Generating Stick Figures

The generated stick figures should express the corresponding motion well, and it should be close to what people may draw to retrieve the motion. To achieve both goals, our algorithm follows the rules that we have learned from the preliminary survey in the previous section.

We selected the key postures at the moment of stopping or changing the moving direction, as our participants did. However, each part of the human body can possibly move independently at the same time, and the moving direction, sometimes, changes continuously without an explicit turning point. Therefore, it would be difficult to pick a specific frame by analyzing the moving path itself.

We use an approximate method to decide on the frame where the moving direction is changed. For the rigid motion, a zero-crossing of the acceleration is a good approximation of the point where the direction is changed. The same approach is used for motion data. We calculate the frame-speed for each discrete frame that is represented as a real number. It is calculated by summing all instantaneous speeds of joint points in the frame. This summed speed could be meaningful only if based on the assumption that the movement of human body parts is highly correlated in the spatial-temporal domain [Ari06]. The consecutive frame-speed values from the first to the last frame comprise one-dimensional signal data. To diminish noise and skip subtle movements that may not be meaningful, the Gaussian low pass filter is applied to the signal. We choose all postures on the zero-crossing of the acceleration as the key postures. In addition, the postures where the frame-speed is near to zero are also selected. If the zero appears continuously during two or more consecutive frames, only the last one is selected as the key posture.

Our method may produce more number of images for the same motion than previous methods such as [ACCO05], since we do not consider the relative salience of each be-

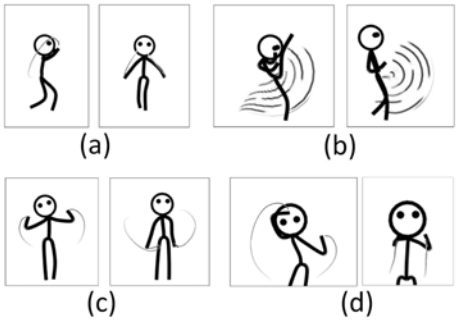


Figure 3: *(a) Trajectory lines for moving joints. (b) Speed lines for fast joints. (c) Trajectory lines for circular moving joints. (d) Cropped view-port to focus on the upper body.*

havior in the motion data. It could be a drawback, in terms of motion browsing, because it requires more space on the screen, and the user may need to spend a more time browsing the motion data. However, our stick figures are not just the representative images for the motion data, but the key values that enable the user to retrieve the corresponding motion. We need to keep the stick figure images, even though they represent less significant parts of the motion data.

4.1. Moving Joints

If a joint of a key posture has moved significantly from the last key posture, we add its trajectory line to the stick figure of the key posture. In terms of visualizing, these lines help the viewers to understand the unpredictable movements between key postures such as circular movement (see Figure 3-(c)). In addition, they can represent relative speed of the motion with different line styles (see Figure 3-(b)). In terms of retrieval, the trajectory lines enable the users to specify a dynamic motion of the body in the input query as well as a static posture (see Figure 5).

In order to determine the moving joints of a key posture, we measure the total travel distance of each joint from the last key posture to the current. If the distance is greater than a certain threshold, ε, the corresponding joint is classified as a *moving joint*. If the average speed of a moving joint during the period is greater than the other threshold, ν, it is classified as a *fast joint*.

4.2. View Plane

As a result of the preliminary survey, we use an arbitrary side view, which is always vertical to the ground and not allowed to be rotated along its normal axis. Since we use orthographic projection, the translation of the plane can be ignored. Finally, the view plane has one degree of freedom, which is the rotational angle θ along the vertical axis. This restriction of the view plane makes it easier for the users to draw a similar stick figure with the generated one.

According to our preliminary survey, the view angle should consider both the motion and the posture. We select an angle that makes the widest image of the stick figure and the trajectory lines on that plane. To do so, we sample a set of 3D points from the stick figures and trajectories, and then find the plane that maximizes the standard deviation of the projected 2D points. To get the optimized angle, we simply sample the angle θ by the step size of 5 degrees from 0 to 180 degrees, test every case, and then select the optimal one. With this method, as the trajectories get longer, the normal of the view plane becomes more perpendicular to the moving direction. When the trajectories are very short or no, the view plane on which only the posture of the stick figure becomes the broadest is selected.

The view planes for key postures are calculated independently, so it can change greatly between two consecutive stick figure images. In this case, the viewer can misunderstand the motion between two key postures. To maintain the consistency of the viewpoint in the local range, we sequentially check two consecutive viewing angles, θ_t and θ_{t+1}, from the first to the last. If the shortest distance between θ_t and θ_{t+1} is greater than 90 degrees, we add 180 degrees to θ_{t+1}. Note that it is same as flipping the image; therefore, the standard deviation value of the image does not change.

4.3. Rendering

We first render the key posture and its trajectories in a stick figure style in the 3D space and project them on the selected plane. The joint points of the skeleton are used as control points. The Catmul-Rom spines [CR74] are used to create a smooth path, passing all the control points. The trajectory lines of the moving joints are drawn by connecting the joint points sequentially from the first to the end frame of the range. The trajectories of the fast joints are rendered by using a texture image of a speed line . We construct a sweep surface along the trajectory line with the width of the limb or torso including the joint (see Figure 3-(b)). The head and two eyes are drawn as spheres. It shows the orientation of the head in the 2D image. Finally, we project them on the selected view plane. If there are not any moving joints or fast joints on both legs, we crop the image to focus only on the upper body.

5. Retrieving Stick Figures

Now, we can build a stick figure database from a motion database. To search for them by using a sketched stick figure, we need an efficient comparison method between two stick figures. Since the sketched stick figures would be noisy and vary depending on the users, comparing them in the image resolution would not be a reliable method. Instead, we first recognize human body from the sketched image and then compare the selected feature values that are less sensitive to the input noises and personal drawing styles.

Recognizing human body posture from a 2D image is

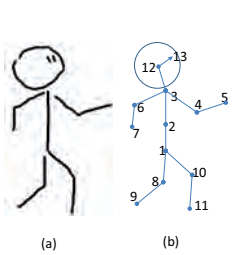
a well-known problem in the computer vision community [MHK06]. The target inputs of those methods are mostly photo images taken by a camera or a video camera, so the algorithm has to include sophisticated image process techniques such as decomposing human parts from the background. The sketched stick figure is also an 2D image of human poses, but it is much simpler as the elaborate image processes are unnecessary. Zamora and Sherwood [ZS10] introduced a human body recognition method specialized to the stick figure images. The method assumes that the input sketch contains whole body parts and matches each segment to a body part in a way to minimize the value regarded as the ambiguity. The recognition performance of those methods would be enough to be used for our purpose. However, since we aim to search in a large database, we introduce a relatively simple method that is more focused on the efficiency of the comparison between stick figures and the progressive search with incomplete stick figure inputs.

In our search, the system returns the retrieval results from the first input stroke and update them progressively as the user adds the other stroke one by one. It saves the trouble of completing stick figures every time, and it also make the search results abundant by considering only the critical constraints. For example, when searching for an arbitrary kicking motion, specifying a certain arm position will limit the range of the search.

We consider the stick figure body and the trajectory lines independently. The user can select one of two brushes (body or trajectory) on the user interface. For the body, the skeleton tree is constructed from the sketched and generated stick figure, and then we compare the directions of the links. The trajectories are first classified by its shapes and compared only within the same class.

5.1. Skeleton Construction and Labeling

We construct a skeleton tree for the stick figure body, which is similar to the 3D skeleton of the general motion data (see Figure 4-(b)). It is a tree structure with a maximum of 13 labeled nodes, and each node corresponds to a human body joint. Extracting the 2D skeleton from the generated stick figures is straightforward: It is the 2D projection of the corresponding joints and bones of the original 3D skeleton. One unusual node is the Eye node, while is linked to the Head node. In the original motion data, the orientation of the head is represented as a three-dimensional rotation, but in the 2D stick figure, we lose one dimension. We compensate for it by representing the eye position explicitly on the stick figure image. Though there are two eyes, we define only one node at the center of both because the distance between them is invariant. For the sketched stick figures, the simplest method is to have the user label all segments and joints manually, but it would be a tedious task for the user. Instead, we automate this process by using a heuristic skeleton-recognizing



Pelvis	1			
Spine	2			
Neck	3			
Left Elbow	4	4	6	6
Left Hand	5	5	7	7
Right Elbow	6	6	4	4
Right Hand	7	7	5	5
Right Knee	8	10	8	10
Right Foot	9	11	9	11
Left Knee	10	8	10	8
Left Foot	11	9	11	9
Head	12			
Center of Eyes	13			

(c)

Figure 4: *(a) Sketched stick figure. (b) Skeleton tree. (c) All possible labeling ways. When the skeleton tree is complete, four different labelings are possible.*

method. It constructs the skeleton procedurally as the input stroke is added.

For the purpose of simplicity, we set two rules in the stick figure sketching. The first rule is that one stroke must be one of four limbs, torso, two eyes, or head. The user cannot draw the upper arm and the lower arm separately nor two arms at once by one stroke. The second rule is that the skeleton is always one connected tree during sketching. For example, if the user sketches two strokes at this moment, they cannot be a pair of arms and legs; the torso connecting between them must exist. These two restrictions are not too strict for the users but makes it significantly easier to extract the skeleton from it.

Each time when an input stroke is sketched, we first check whether it is a circle or not. We use the Hough transform method [Bal81] for circle recognition. If it is a circle and the Head node does not exist yet, the stroke is regarded as a head circle, and its center point is added to the skeleton tree and labeled as Head node. The strokes that are drawn inside of the head circle are regarded as an eye. Up to two eyes can be drawn, and the center point of the eye strokes is added to the tree and labeled as the Eye node, and then linked to the Head node.

In all other cases, one input stroke is regarded as one of the four limbs or the torso. Three nodes $\mathbf{n}_{e1}$, $\mathbf{n}_{e2}$, $\mathbf{n}_m$ are generated at both ends and middle points of the stroke. The middle node $\mathbf{n}_m$ links the two other nodes: $\mathbf{n}_{e1}$ and $\mathbf{n}_{e2}$. This subtree is combined into the existing skeleton tree, if it exists. If the existing tree has a Head node but not a Neck node, the nearest one between $\mathbf{n}_{e1}$ and $\mathbf{n}_{e2}$ to the Head node links to the Head node, and it is labeled as the Neck node. If there are other nodes, which are not Head or Eye nodes in the existing tree, we check the distance between all pairs of existing nodes and both end nodes, $\mathbf{n}_{e1}$ and $\mathbf{n}_{e2}$. The minimum distance pair is combined as one node. During this process, all wrong topology, which is not a subset of the full skeleton tree, is ruled out.

The possible labels of an unlabeled node can be limited by considering its topological position and the label of adjacent nodes. For example, an unlabeled node linked to the Neck

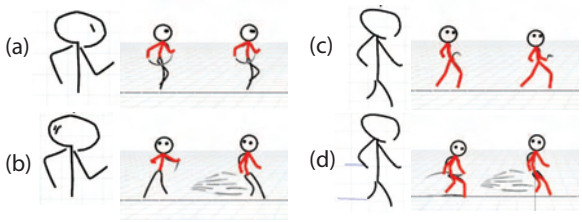


Figure 5: *The sketched stick figures and two retrieved results for each. The sketches (a) and (b) are exactly same except the position of eyes, but this makes a broad distinction between the retrieved results from both. The only difference between sketches (c) and (d) is the existence of the trajectory lines. The sketch (d) retrieves more dynamic motions.*

node must be one of the Left-Elbow node, Right-Elbow node, and Spine node. In the case of a full skeleton tree, four different labelings are possible as listed in Figure 4-(c). When two strokes are drawn without the head circle, its skeleton has the maximum number of cases; 12 labelings are possible. Before being used as an input query, the label of each node should be determined as one. We simply consider every possible case as an independent input query, conduct retrieval multiple times, and then take the best result. From now on, we assume that the skeleton tree of any sketched stick figure is fully labeled.

5.2. Trajectory Classification

A trajectory line always belongs to a node (or a joint in motion) of the skeleton tree, so it is labeled with the same name with of the owner node. The owner node of a sketched trajectory is decided when the stroke starts being drawn, and it is the nearest node to the starting point of the stroke.

A trajectory of human motion is basically a free form line, but it is a difficult to measure the similarity between two free form lines. The trajectory line generated by our algorithm does not reach over the previous key posture, which is regarded as the last point where the moving direction is changed. It may be a reasonable assumption that most trajectories are simple curves such as lines or arcs with some noise. We classify the trajectories into four groups: Line, Arc, Ellipse, and Complex. The Line and Arc include straight lines and arcs without any cross points, respectively. Ellipse, in our definition, includes circles, arc-with-one-cross, as well as the ellipses. All other curves are classified as Complex. We use the stroke classification algorithm based on analyzing linearity and convexity introduced by Qin [Qin05].

5.3. Feature Vector Extraction

We first extract the feature vector from the sketched stick figure, and then extract a corresponding feature vector from each generated stick figure in the database for comparison.

5.3.1. From Sketched Stick Figure

The body proportions of the sketched stick figure can be different from the data and be greatly varied depending on the user. The absolute link sizes or node positions of the skeleton tree will not be reliable information for the comparison. We take the directions of links as the features of the skeleton. All links are represented as 2D vectors, and they are normalized to the length one except for the link between the Head node and Eye node. The length of this link (or distance from the Eye node to Head node) changes as the head rotates (see Figure 5). We normalize the length of the Eye link vector in the range from zero to one. It becomes zero when the Eye node is at the head center and one when it is at the boundary of the head circle. The skeleton feature vector $\mathbf{f}^s$ is represented as:

$$\mathbf{f}^s = \{l_1^x, l_1^y, l_2^x, l_2^y, ..., l_n^x, l_n^y\} \quad (1)$$

where l_i^x and l_i^y are x and y elements of the normalized 2D vector with respect to ith link, and n is the number of links in the current skeleton tree. Note that the size of the tree changes with the input.

From the trajectory line, we sample 10 points with the uniform distribution along the path from the front to the rear. The feature vector for kth trajectory line is:

$$\mathbf{f}_k^t = \left\{x, y | x = p_j^x - p_1^x, y = p_j^y - p_1^y, j = 1, 2, ..., 10\right\} \quad (2)$$

where p_j^x and p_j^y are x and y elements of the jth sampling point, and m is the number of sketched trajectories. The first point (p_1^x, p_1^y) is subtracted from every sampling point to represent it in a coordinate invariant form.

5.3.2. From Generated Stick Figure

Before extracting the feature vector frame a generated stick figure R, we first check the *trajectory-correspondence condition* with the sketched stick figure I. For each trajectory in a I, corresponding trajectory must exist in R, and each pair must be in the same shape class. If not, R is skipped in the search. The missing trajectories in the sketched stick figure are not considered in the comparison.

Extracting the skeleton feature vector $\overline{\mathbf{f}^s}$ from a generated stick figure is trivial. The normalized direction vectors are taken from only the links corresponding to the links existing in the sketched one. The kth trajectory feature vector of the generated stick figure $\overline{\mathbf{f}_k^t}$ is also similar with the one of the sketched stick figure, but the scale factor s is multiplied for all elements. s is a real number that is calculated by dividing all link lengths of the sketched stick figure by all corresponding link lengths of the generated stick figure.

$$\overline{\mathbf{f}_k^t} = \left\{x, y | x = s(\overline{p_j^x} - \overline{p_1^x}), y = s(\overline{p_j^y} - \overline{p_1^y}), j = 1, 2, ..., 10\right\} \quad (3)$$

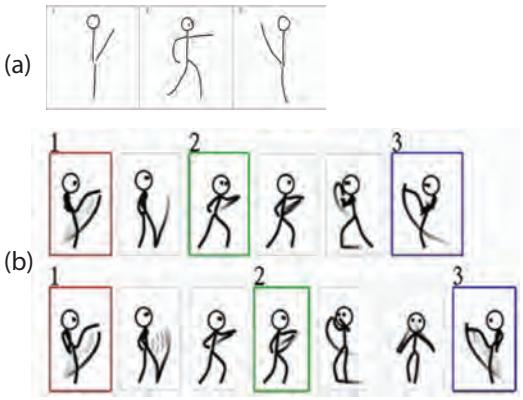


Figure 6: *By sketching a sequence of stick figures, the user can retrieve a longer motion data. (a) is a sequence of three sketched stick figures representing kicking, punching, and kicking. (b) shows the two retrieved results.*

5.4. Comparison and Retrieval

Given a sketched stick figure I and a generated stick figure R, we first check the trajectory-correspondence condition between them. If is passes, the difference between them is calculated as follows.

$$D(I,R) = \sqrt{w^s|\mathbf{f}^s - \overline{\mathbf{f}^s}|^2 + \sum_1^m w^t|\mathbf{f}_k^t - \overline{\mathbf{f}_k^t}|^2} \qquad (4)$$

where w^s and w^t are the weight values that control the priority between the posture and the trajectory in the comparison. In our implementation, w^s is set to the inverse of the size of vector $\mathbf{f}^s$, and w^t is set to the inverse of the size of the vector $\mathbf{f}_k^t$ ($\frac{1}{10}$ in our implementation). To search for the best matched stick figure, we compare with every stick figure one by one. The time complexity is proportional to the number of stick figure images in the database.

To search for a longer motion, we take a sequence of sketched stick figures as the input query. The expected retrieval result is a consecutive sequence of the generated stick figures, in which similar stick figures appear in the same order and as closely from each other as possible. When n number of the input stick figures $\{I_1, I_2, ..., I_n\}$ is given, we first search n times for each input independently, and then retrieve the best m results for each of them. In our implementation, m is set to ten percent of the total number of generated stick figures. Let $\mathbf{S}_i$ be the set of m retrieved stick figures from the input sketch I_i. We check the motion database file by file and select every file that contains any set of n stick figures $R_1 \in \mathbf{S}_1, R_2 \in \mathbf{S}_2, ..., R_n \in \mathbf{S}_n$ in the same order. In the stick figure sequence of a selected file, there might be more stick figures of the transition motion between $R_i and R_{i+1}$. We evaluate each of selected files by the function E:

$$E = \exp(-\frac{T}{\sigma})\sum_1^n D(I_i,R_i), \quad T = \sqrt{\sum_1^{n-1} t(R_i,R_{i+1})^2} \qquad (5)$$

where the function $t(R_i,R_{i+1})$ counts the number of frames between the last frame of R_i and the first frame of R_{i+1} in the motion file. As the frame distances between R_i and R_{i+1} increase, the output of the function T also increases. The *sigma*, in our implementation, is set to 500. Finally, the selected files are displayed on the screen in increasing order of the values evaluated by function E. The time complexity is $\mathcal{O}(n\mathcal{N}+\mathcal{F})$ where n, $\mathcal{N}$, and $\mathcal{F}$ are the number of input stick figure, the number of stick figure in the database, and the number of motion files in the database, respectively. Figure 6 shows an retrieved results when the three stick figures are used as an input query.

6. Experimental Results

All experiments were done on a PC equipped with Intel Core2 Duo CPU 2.8GHz and 4GB main memory. We built a motion database with 102 motion data files which were downloaded from MRL public motion database [SMRL09]. The frame rate of all motion data is 30 frames/second and the length is various, from 5 second to 30 seconds. The motion data includes locomotion, interaction with an environment, sports, dances, and general motions. Totally, 2308 stick figure images with 2623 trajectory lines were generated from the database. The percentages of Line, Arc, Ellipse, and Complex in total trajectory lines are 55%, 19%, 5%, and 21%, respectively. The average time to process 1000 frames of motion data was 1.5 seconds. When search for two input strokes without the head circle (it has the maximum labels cases), the average response time of 100 retrievals was 0.170 seconds. When we add one more straight trajectory line to the two stroke, the average time reduced to 0.116 seconds. This happened because the database was filtered by the trajectory-correspondence condition.

6.1. User Study

With the constructed motion database, we conducted a user study. We separated the participants into two groups. Both groups were asked to search for the given motion data by using different types of user interface, respectively. Our sketch-based interface was provided to the first group. They could retrieve motion files directly by sketching a stick figure as a query, and multiple tries were allowed for each task. To the second group, we allowed only to browse the generated stick figures. They could look over the sequences of stick figures file-by-file on the provided program. The displaying order in the list was randomly shuffled at the beginning of each task. Each group consisted of eight Master or Ph.D students studying computer science. There was no overlap with the participants in the preliminary survey. They had had about five minutes training time before the user study begun.

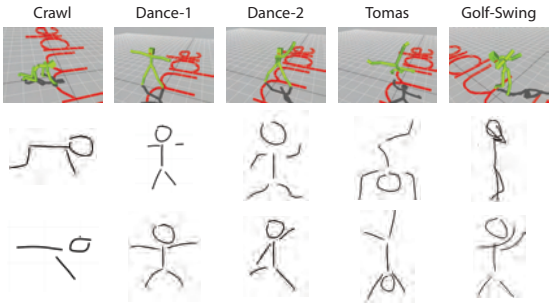


Figure 7: *Each column shows the example motion and two sketched stick figures drawn by users to search for them.*

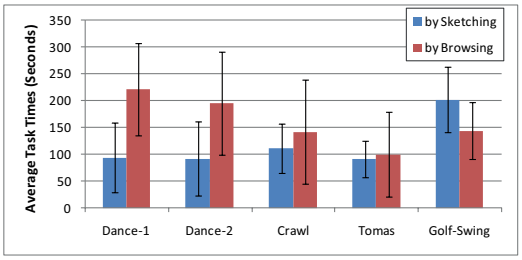


Figure 8: *Average search times and standard deviations in the user study. First group retrieved motion by sketching (Blue). Second group searched by only browsing sequences of the generated stick figures (Red).*

After taking five minutes training time, one participant conducted five tasks with five example motions: Dance-1, Dance-2, Crawl, Tomas, and Golf-Swing. The animation of the example motion was being displayed on one monitor while the user was searching it on the other monitor with the provided user interface according to the belonged group. We measured the time taken to complete each task. Figure 8 shows the average times and their standard deviations.

In all cases except the Golf-Swing, the average search time of the second group are higher than of the first group. The most representative posture of Golf-Swing is at the last moment of the swing in which both arms are twisted around the neck. Most participants in the first group tried to describe this posture rather than other postures (see fifth column in Figure 8). However, since the posture is relatively complex to draw, they had been failed several times at the beginning. On the other hand, the uniqueness of this posture could be an advantage to be found by browsing for the second group.

All participants in the first group sketched the body stick figures repeatedly without trajectory lines at the beginning of each task. They used the trajectory lines after several failures of the retrieval. No users sketched the Eye stroke during the user study. The second group participants usually checked each motion from the first to the end one by one. If the correct answer had located at the front of the list after shuffling,

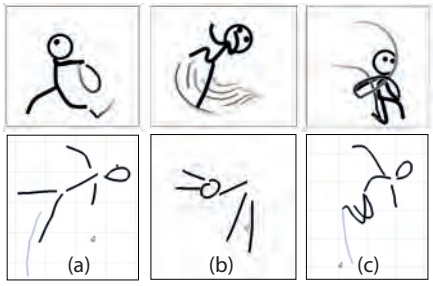


Figure 9: *(Top row) Three consecutive stick figures generated from a Back-Tumbling motion. (Bottom row) Three stick figures skethced by the user to retrieve the Back-Tumbling. These three trials failed to retrieve the correct motion because of the poor descriptions (a,b) or the selection of missing frame in the generated stick figures (c).*

the search time would be short, and vise versa. Hence, their standard deviations were much higher than of the first group.

6.2. Limitation

Although stick figures is a very simple drawing style, some human motion are still difficult to be drawn by general people. For example, when we asked to search a Back-Tumbling motion, the user had difficulty to describe it. Finally, the poor descriptions resulted in failure to retrieve a correct motion (see Figure 9-(a) and (b)).

The generated stick figures includes only the selected key-frame, but it may not same the user's selection. For example, in the Back-Tumbling example, a user sketched a posture when the body is on the air (see Figure 9-(c)). However, our key-frame selection algorithm skipped this frame, and the retrieval was failed. We may improve the key-frame selection algorithm, but a fundamental solution would be to develop an algorithm that search on continuous time space and fast enough for the interactive retrieval.

The simple 2D view of the stick figures also sometimes limits the expressiveness of human motion. We limited the range of view planes in one of side-views, but a bird's-eye view would be more appropriate (e.g. Kicking with outstretched arms). This problem can be resolved by rendering the key postures with a 3D character in a 3D space. We may evaluate the ambiguity of each generated stick figure, and then, as for the highly ambiguous stick figures, we can display both the stick figure and the well-rendered image together.

7. Conclusion and Discussion

In this paper, we used stick figures as the unified medium for visualizing and retrieving for human motion data. The user can browse the motion database through the generated stick figure images, and retrieve them by sketching a similar stick

figures. To increase the success rates of the retrieval, we tried to make the generated stick figures similar to people's drawing. The user interface was also designed to assist the user's sketching by taking an incomplete stick figure and updating the retrieval results interactively. We showed the usefulness of the our method though the user study.

Our visualization of motion data is simplified as much as the general user can draw it without difficulty. However, if we ignore the search, we have more place to improve the expressiveness of the stick figure images. In some motion, the emotion could be a critical element to be fully understood by the user. (e.g. delight jumping or a furious punch). To visualize the emotion, we could draw cartoon-like facial expressions on the character face. The head of our stick figure has only two eyes. When the motion is looking in the same direction with the user, the user cannot anticipate the detail-orientation of the head. We can add the hair or ears on the head to show the direction of the head. In addition, our stick figures do not give any information about the global movement. One idea is to place the stick figures at the related positions on a 2D map which is corresponding to the real world where the motion had been captured.

Acknowledgements

This work was supported by JST ERATO and the National Research Foundation of Korea(NRF) grant funded by the Korea government(MEST) (No. 2012-0001242).

References

[ACCO05] ASSA J., CASPI Y., COHEN-OR D.: Action synopsis: pose selection and illustration. *ACM T Graphic (SIGGRAPH 2005)* (2005), 667–676. 1, 2, 3

[Ari06] ARIKAN O.: Compression of motion capture databases. *ACM T Graphic (SIGGRAPH 2006) 25*, 3 (2006), 890–897. 3

[Bal81] BALLARD D.: Generalizing the Hough transform to detect arbitrary shapes. *Pattern Recognition 13*, 2 (1981), 111–122. 5

[BZOP07] BOUVIER-ZAPPA S., OSTROMOUKHOV V., POULIN P.: Motion cues for illustration of skeletal motion capture data. In *Proceedings of the 5th international symposium on Non-photorealistic animation and rendering* (2007), p. 140. 2

[cFGDJ08] CHIENG FENG T., GUNAWARDANE P., DAVIS J., JIANG B.: Motion capture data retrieval using an artist's doll. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference* (2008), pp. 1–4. 2

[CGL03] CMU-GRAPHICS-LAB: CMU motion capture database, http://mocap.cs.cmu.edu/, 2003. 1

[CR74] CATMULL E., ROM R.: A class of local interpolating splines. In *Proceedings of Computer aided geometric design* (1974), p. 317. 4

[DAC*03] DAVIS J., AGRAWALA M., CHUANG E., POPOVIĆ Z., SALESIN D.: A sketching interface for articulated figure animation. In *Proceedings of the ACM SIGGRAPH/Eurographics symposium on Computer animation* (2003), Eurographics Association, pp. 320–328. 2

[DGL09] DENG Z., GU Q., LI Q.: Perceptually consistent example-based human motion retrieval. In *Proceedings of the 2009 symposium on Interactive 3D graphics and games* (2009), I3D '09, pp. 191–198. 2

[KTWZ10] KRÜGER B., TAUTGES J., WEBER A., ZINKE A.: Fast local and global similarity searches in large motion capture databases. *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2010), 1–10. 2

[LCR*02] LEE J., CHAI J., REITSMA P. S. A., HODGINS J. K., POLLARD N. S.: Interactive control of avatars animated with human motion data. *ACM T Graphic (SIGGRAPH 2002) 21*, 3 (2002), 491–500. 2

[LIMS10] LIN J., IGARASHI T., MITANI J., SAUL G.: A sketching interface for sitting-pose design. *EUROGRAPHICS Symposium on Sketch-Based Interfaces and Modeling* (2010). 2

[LZ10] LO W.-Y., ZWICKER M.: Bidirectional search for interactive motion synthesis. *Computer Graphics Forum 29*, 2 (2010), 563–573. 2

[MBS09] MÜLLER M., BAAK A., SEIDEL H.-P.: Efficient and robust annotation of motion capture data. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2009), pp. 17–26. 2

[MCC09] MIN J., CHEN Y.-L., CHAI J.: Interactive generation of human animation with deformable motion models. *ACM T Graphic 29* (2009), 9:1–9:12. 2

[MHK06] MOESLUND T. B., HILTON A., KRÜGER V.: A survey of advances in vision-based human motion capture and analysis. *Comput. Vis. Image Underst. 104*, 2 (2006), 90–126. 5

[MQW08] MAO C., QIN S. F., WRIGHT D. K.: A sketch-based gesture interface for rough 3d stick figure animation. *Perspective* (2008). 2

[MRC05] MÜLLER M., RODER T., CLAUSEN M.: Efficient content-based retrieval of motion capture data. *ACM T Graphic (SIGGRAPH 2005) 24* (2005), 677–685. 2

[NNSH11] NUMAGUCHI N., NAKAZAWA A., SHIRATORI T., HODGINS J. K.: A puppet interface for retrieval of motion capture data. In *The proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2011). 2

[PZ11] PAN J., ZHANG J. J.: Transactions on edutainment vi. 2011, ch. Sketch-based skeleton-driven 2D animation and motion capture, pp. 164–181. 2

[Qin05] QIN S. F.: Intelligent classification of sketch strokes. In *EUROCON* (2005), vol. 2, pp. 1374–1377. 6

[SMRL09] SNU-MOVEMENT-RESEARCH-LAB: SNU motion database, http://mrl.snu.ac.kr/~mdb/, 2009. 7

[TBvdP07] THORNE M., BURKE D., VAN DE PANNE M.: Motion doodles: an interface for sketching character motion. *ACM T Graphic (SIGGRAPH 2007)* (2007), 24. 2

[WC11] WEI X., CHAI J.: Intuitive interactive human-character posing with millions of example poses. *IEEE Comput. Graph. Appl. 31*, 4 (2011), 78–88. 2

[YKSN07] YASUDA H., KAIHARA R., SAITO S., NAKAJIMA M.: Motion belts. In *ACM SIGGRAPH 2007 sketches* (2007). 2

[ZS10] ZAMORA S., SHERWOOD T.: Sketch-based recognition system for general articulated skeletal figures. In *Proceedings of the Seventh Sketch-Based Interfaces and Modeling Symposium* (2010), SBIM '10, Eurographics Association, pp. 119–126. 5