# Techniques and Approaches in Static Visualization of Motion Capture Data

**William Li**
School of Interactive Arts and Technology
Simon Fraser University
dla135@sfu.ca

**Lyn Bartram**
School of Interactive Arts and Technology
Simon Fraser University
lyn@sfu.ca

**Philippe Pasquier**
School of Interactive Arts and Technology
Simon Fraser University
pasquier@sfu.ca

## ABSTRACT
In this paper we present a state of the art of the current approaches to visualization of motion capture data. We discuss the data representation, pre-processing techniques, and the design of existing tools and systems. Next we outline the advantages and disadvantages of the systems, some of which are explicitly noted by the original authors. Lastly we conclude with an overall summary and future directions.

### Author Keywords
Visualization; motion capture; movement.

### ACM Classification Keywords
A.1 Introductory and survey

## INTRODUCTION
Historically, humans have developed many techniques of representing motions in a single image. The illustration of motion serves to allow the audience to easily understand the nature of the motion. As humans, the visualization of human movement has naturally been an interesting topic. Before the invention of photographs, movements were represented in drawings or paintings. With technological advances, there are new tools to visualize motion. Motion capture is flexible in that data can be recorded for whatever action is considered necessary for the research project. The data is also easily controlled in terms of the sample size and duration. Therefore, many research groups are working with full body motion capture data. However, expanding databases leads to a growing importance to be able to effectively visualize large amounts of data. In this paper, the terms *motion* and *movement* are used interchangeably as they both refer specifically to human motions.

As Alemi et al. [1] notes, works on human movement visualization can be classified as artisitic visualizations, movement information analytics, and movement summarizations. Artistic visualization includes works such as *Bodycloud* [31] where sculptures are created from the spaces of the movement, or EMVIZ [35] where effort qualities are mapped to visual representations. Movement information analytics include visualizations that provide insights to the characteristics of movements and are used to evaluate and understand movements. It illustrates structural information of the movement. Examples include ActionPlot [13], and Synchronous Objects [30]. Movement summarizations are visualizations that are used to provide a synopsis of the movement itself or to compare the contents of movement data, usually by mapping the data to a lower dimension space. This paper aims to present a state of the art on the current techniques and systems in the static movement summarization visualization of human movement. While it can be argued that video renderings will always produce the best visualization of movement, we believe using a rendering defeats the purpose of a summarization as one would have to watch the rendering in its entirety. Therefore, video and rendering techniques are not discussed here.

For the rest of the paper, we start with some common techniques used to visualize motion throughout history before motion capture. Next, we describe the general data representations used by the works presented in this paper as well as introduce commonly used databases. The following sections present the various pre-processing techniques used to handle motion capture data and visualization approaches of the current systems. This paper is not meant to provide implementation details for these systems but rather summarizations of systems that are tackling the problem of motion visualization. Lastly, we conclude with a discussion on the advantages and disadvantages of the presented systems, issues with motion visualization in general, and then a summary of the paper, the presented works, as well as concluding remarks and future directions.

## HISTORY
Without the context of data, James Cutting has outlined the techniques of representing motion that pertains especially to drawings: dynamic balance, multiple stroboscopic images, affine shear, photographic blur, and action lines [14]. Dynamic balance refers to the broken symmetry or instability of the human form. Stroboscopic images are the use of multiple images within one to depict motion. Affine shear is where the subject is shown leaning into the direction of movement. Phtographic blur is where movement is represented by a blur as though from a long-exposure photograph. Action lines are lines or arrows that illustrate movement.

Cutting evaluates these techniques with the following criteria: evocativeness, clarity, direction, and precision. Evocativeness refers to whether the representation succeed in evoking a feeling of motion in the viewer. Clarity means whether the object


*MOCO'16*, July 5–6, 2016, Thessaloniki, Greece.

DOI: http://dx.doi.org/10.1145/2948910.2948935

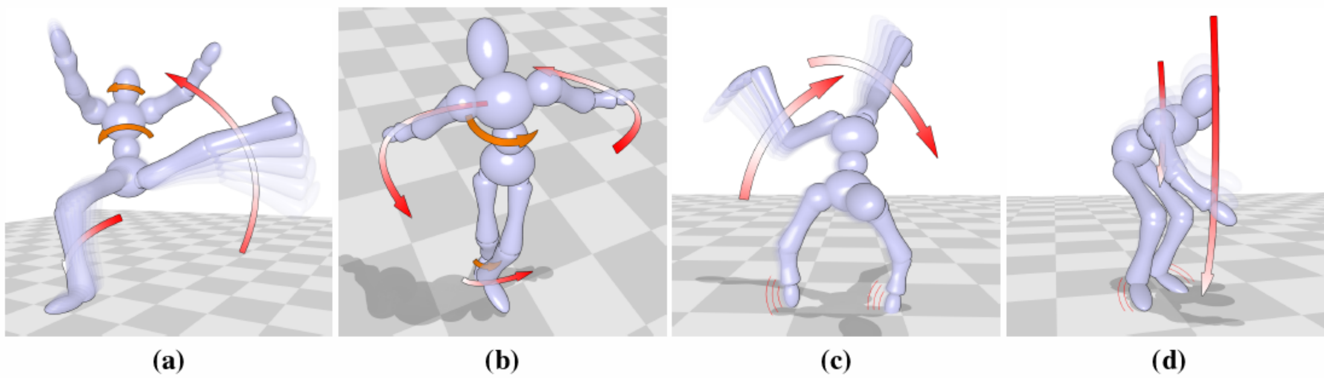


**Figure 1. Motion cues [11] showing movements with motion arrows, noise waves, and stroboscopic motion: (a)spin kick (b)dancing pirouette (c)cart wheel (d)bending over**

that is represented can be accurately identified. Direction is referring to whether the direction of movement is clear to the viewer. And lastly, precision means the amount of motion that has occurred.

Dynamic balance can be traced back as far as the early Greeks [36]. Although it does evoke a sense of motion, Cuttings noted that this technique is not very accurate in representing the direction or precision of motion.

The original use of stroboscopic images is attributed to figures such as Leonardo in the *Vitruvian Man* and Descartes in *Meditations*, and then taking off in the late 19th century with fast films, fast lighting, and photography [14]. Stroboscopic images was shown to be so effective at representing motion that even preschoolers understood them [15]. However, the direction of motion is unclear with this representation.

Affine shear or forward lean is a technique commonly used starting in the 20th century by artists and cartoonists [14]. It shows an effort in overcoming inertia or moving into the wind. The amount of lean is often representative of the speed at which the subject is moving. Although the directionality is clear, it is used at the discretion of the artist, and as such, it is impossible to compare the relative speeds between different images.

With a stationary framework in photographic blur, the sense of motion is evoked in the viewer for the blurred parts. It is only effective in the evocativeness aspect, as the blur renders the movement unclear in its clarity, direction, and precision.

The use of action lines and motion arrows attempts to solve the issues in the previous techniques regarding clarity and directionality. They originate from vectors as defined in mathematics. Action lines can be combined with stroboscopic images with good effect [27]. However, action lines seem more difficult to understand for children [15]. According to Cutting [14], this is the most effective technique in satisfying the four criteria.

In more recent times, other notation techniques have been developed, such as the Labanotation and the Benesh notation. The Labanotation [17] describes movement through motif, effort-shape, and structure. Meanwhile, the Benesh notation is similar to staff music notation and often used to describe dance. [6].

The context of the problem is different today given the development of media from still images to dynamic videos, and then encapsulating the 3D information in the form of motion capture. Although the overall intent is still to visualize motion, motion capture offers a new level of precision for every limb. With time-based media, the problem is one of the quantity of information, hence the need for summarization. To that end, many systems and interfaces have been developed. The exact motive of each system are different. However, they are similar in their goal to reduce the data while still presenting a salient summary, whether it be for viewing or further comparison and analysis.

## DATA AND DATA REPRESENTATION

Full body motion capture data is collected by recording the movements of an actor wearing a body suit with markers and cameras or sensors. Using different marker positions, number of markers, or sensors could result in a different skeleton. The file format of motion capture can also vary, such as BVH, Acclaim, and Collada. However, the content of encoded information is usually similar, such as positions and angles of the joints in a body. Examples of other features that can be computed are outlined by Larboulette and Gibet [23]. All of these encoded or extrapolated information are referred to as the dimensions or features of motion capture data. The process of motion capture is described by Bodenheimer and Rose [10], involving gathering of the data, construction of a virtual skeleton, and the processing required to produce the desired joint angle information.

When viewing the files directly, a sequence of a human stick figure is seen moving through the space. The motion capture information, specifically the joint angles are typically represented using Euler angles or quaternions, with the former being the simpler and more intuitive to implement [10]. However, it has been suggested that quaternions are usually more ideal because they are easier to renomralize and the interpolation or orientation is more easily defined [10, 33]. Luckily, conversion between the two is a fairly trivial process [9]. As the size of databases grow, so does the importance of having efficient visualizations to browse through these databases.

For publicly available data, the CMU motion capture database [16] found at `http://mocap.cs.cmu.edu` is commonly used. They use a Vicon motion capture system with 12 infrared MX-40 cameras. Their skeleton consists of at least

41 markers. The resulting files are in asf and amc formats. The list of motions they have captured include various forms of walking, climbing, playing, dancing, sports performances, and other everyday movement.

Another publicly available source is the HDM05 database [28] found at `http://resources.mpi-inf.mpg.de/HDM05/`. Their recorded motions are tracked by a Vicon MX system with 12 cameras. Their skeleton consists of 40 to 50 markers. They also use asf and amc file formats with documented parsing tools, with videos available as well. The goal of the authors is to provide researchers with motion capture data in addition to the CMU motion capture database. However, the HDM05 database is more limited in the range of different motions that it contains.

A third and newly developed database is the MoDa database [29], found at `http://moda.movingstories.ca`. As a more recent database, the quantity of data and documentation is more limited than the other two databases. However, MoDa contains data captured with Vicon, video camera, as well as Kinect. MoDa also contains generated motion capture data. It uses Mova [1] as its front end. There are also addons such as MOTATE [29] and MoComp [26] under development.

## PRE-PROCESSING

With skeletons containing over 40 markers, the joint positions and angles that are generated from these markers can result in vectors with hundreds of dimensions for every frame of the motion capture. This leads to a huge amount of data to be visualized. Therefore, dimensionality reduction for the purpose of data abstraction is a significant process for all motion visualization tools. The goal is to retain the important characteristics that are indicative for that motion while simplifying the dimensions of the data so that it can be effectively visualized in some way.

To abstract the data, many systems use some variation of clustering algorithms or dimensionality reduction techniques, keyframe extraction, or a combination of all of the above. There have been extensive work in the development of cluster data mining techniques to handle multivariate data [7], including hierarchical, spectral, density-based, and partition-based clustering. Keyframes are considered a good visualization of motion capture data because they are the frames containing the representative poses of that motion.

### Clustering and Dimensionality Reduction

Out of the many clustering techniques, the self-organizing map (SOM) [22] have been noted by Hu et al. [18] as effective for visualization as it projects the data onto a grid-like topology. SOM converts high-dimensional input to a low-dimensional map. Nodes that are in closer proximity within the map are more similar in the original data. In the context of motion capture data, this would mean that nodes in a SOM that are close together had similar joint positions and angles. A straightforward visualization would then be to simply place all the frames within the SOM with weighted positions as shown by Wu et al. [38]. However, the grid resolution can easily become unmanageable with this method. Therefore, other techniques are often needed in addition to SOM.

Hierarchical clustering algorithms have been used in systems such as GestureAnalyzer [19]. Similar gestures are grouped together and then given a tree structure. Nodes that are closer within the tree mean those gestures are more similar.

Principal component analysis (PCA) [21] is another well-known technique for dimensionality reduction that has been applied to motion capture data. For example, Barbič et al. [5] used PCA to segment motion data into distinct motions. However, it has been noted that PCA is more suitable for datasets that contain more varied poses [18], which may not be the case for motion data.

### Keyframe Extraction

The goal of keyframe extraction is to select the frames that best represent the motion. The methods usually fall under one of three categories: curve simplification, clustering, and matrix factorization [12]. These methods differ mainly in their representation of the data.

In curve simplification, the data is represented as a curve in high-dimensional space. Curve simplification algorithms are then applied to this trajectory. The intersections of the simplified curve segments are defined as the keyframes. An example is the work of Xiao et al. [39] where they showed this method is able to compress and summarize the features of the data efficiently as well as maintain consistency between similar motion sequences.

Clustering methods refer to those described in the previous section. In the context of keyframe extraction, a frame in each cluster is selected to be the keyframe. This is the method used by Liu et al. [25] where they extracted keyframes for motion data retrieval.

Lastly, in matrix factorization, frames are represented using matrices containing the feature vectors. The motion is then summarized using techniques such as singular value decompositions. An example of this method is seen in the action synopsis system of Assa et al. [2].

## VISUALIZATION APPROACHES

There are many visualization approaches to render clusters or keyframes produced from pre-processing. Most approaches result in the form of image summaries, interactive platforms or graphical user interfaces (GUI), or less commonly, videos and animation.

### Image Summaries

Using an image or picture to represent motion is arguably the most intuitive approach. Combined with the nature of keyframes, it is not a surprise that many groups choose to visualize motions using keyframes in some way.

#### *Motion Map*

Initially developed for efficient motion data retrieval using image-based keys, the motion map system by Sakamoto et al. [32] results in a visualization of motion that is close to simply displaying the SOM. However, they do realize that displaying every frame in the SOM will result in clutter. By using aforementioned clustering and keyframe extraction techniques, a resulting motion map is produced. Because their system is run on multiple motion files, the resulting map can be considered as an overview for all the motions present in those files. Although it is a rather simplified visual representation, a viewer is able to at least extrapolate the motions within the batch of files as well as see similar motions clustered to some extent.

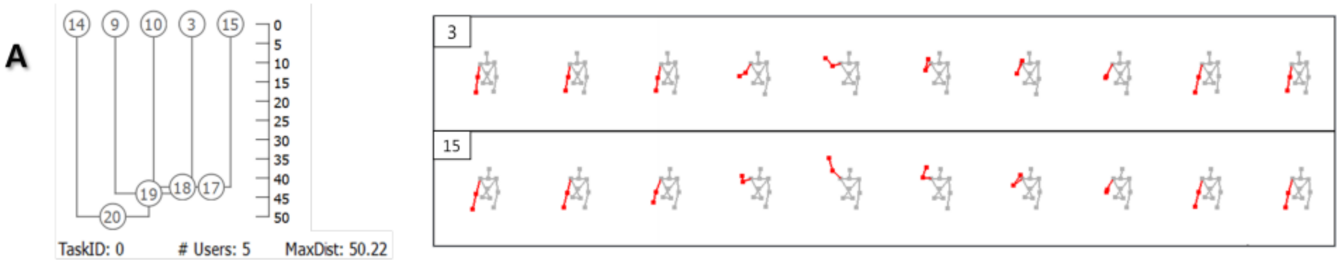


**Figure 2. Motion visualization of GestureAnalyzer [19]**

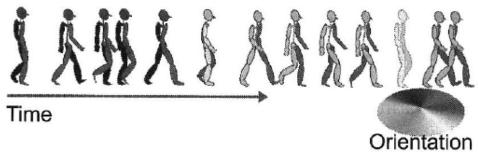


**Figure 3. An example of a motion belt [40] output for walking with a 90 degree turn. The color indicates the hip orientation.**

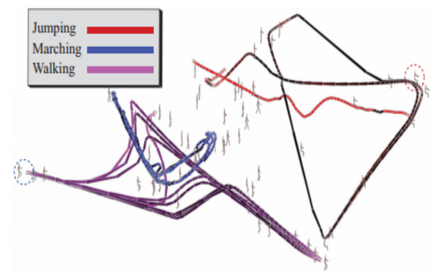


**Figure 4. Generated motion tracks [18] for jumping, marching, and walking.**

### *Motion Cues*

Similar to the techniques presented by Cutting [14], Bouvier-Zappa et al. [11] adds motion cues onto the extracted keyframes. An example is shown in Figure 1. They also note that they can easily highlight the motions of a particular limb. However, the authors note that it was sometimes difficult to link several key poses to reconstruct the original motion. Therefore, their technique was more effective for movements such as walking and running. They also attempted a more specific application of their system in foot sequence illustration for use in dance notation.

As the motion cues are similar to the works presented by Cutting [14], they can also be evaluated in terms of evocativeness, clarity, direction, and precision. Bouvier-Zappa et al. [11] also notes that they had these criteria in mind when they designed their system. However, they did not explicitly test the effectiveness of their visualization via those or other criteria. Another drawback is that the temporal element of the original motion capture data is lost in their final output, as it is not possible to determine the duration of the original movement.

### *Motion Belt*

Yasuda et al. [40] wanted to present the temporal element of the data in their motion visualization as well as illustrate the motion itself. To that end they developed the motion belt. Similar to many other approaches, the motion belt can be seen as a presentation of a few selected keyframes that are able to demonstrate to the viewer the nature of the motion. The keyframes are placed with their correct relative distances apart on a horizontal timeline, showing the location from which the keyframes were extracted in the original data. Unlike the motion cue system, the motion belt is unable to highlight particular limbs. An example of the motion belt representation is shown in Figure 3. Using a greyscale of colors, they are able to show the changes in orientation that occured.

As their goal for the motion belt was efficient data browsing, Yasuda et al. [40] tested their system by measuring the time to find a given motion. The result was that participants took a shorter time using a motion belt compared to just using content description and a link to the target clip. However, although the authors claim the motion belt can also be used for motion comparison, Hu et al. [18] notes that the motion belt is insufficient for comparison. Furthermore, comparing multiple motion belts at once can becomes an increasingly difficult problem.

### *Motion Track*

In order to visualize the variations between motion data, Hu et al. [18] developed the motion track. Keyframes in the low dimensional space are connected using colored lines to form a track. As shown in Figure 4, the jumping track is clearly separated from the others because jumping is a very different motion. However, walking and marching is expected to have some similar poses, and thus they show some overlap. Hu et al. also demonstrates the ability of motion tracks to differentiate motions. They use colors to indicate the motion speed: the brighter the color, the slower the motion. Through the use of visually clustering tracks and colors for highlighting, a viewer is able to distinguish the relative differences in features between motions.

The authors note that a drawback of their system is that some features of the motion track are only available for motions with high stability such as walking, running, and marching. Furthermore, although the keyframes are shown on the sides of the tracks, they are not definitive enough to understand the nature of the motion without the accompanying label for every track. It is also implied that a longer original sequence would result in a longer track, but even if this is the case, that is a difficult distinction to make from simply viewing the tracks. It is also impossible to see where those keyframes lie within the original data.

### *Action Synopsis*

The last example of visualization using keyframes is the action synopsis system developed by Assa et al. [2]. Their goal with this system is to be able to summarize and describe a

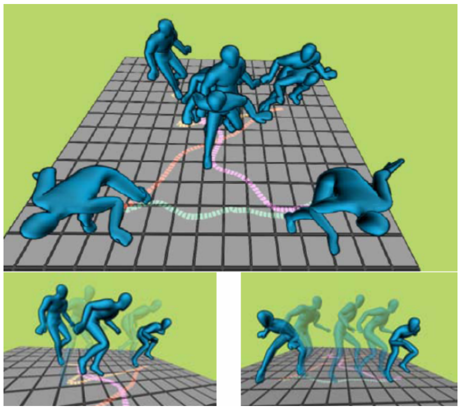

**Figure 5. An example output of the action synopsis system [2] for a sneaking sequence.**

**Figure 6. Hierarchical tree visualization in MotionExplorer [8]**

complex motion. The overall process is similar to previously described, from raw data to pre-processing to rendering the keyframes as the final representation. A difference in their system is that they are able to use animation sequences and video clips as input in addition to the motion capture data used by the other systems. An example of their final output is shown in Figure 5. Through the use of stroboscopic images and colored lines to indicate the path, a summary of the motion is depicted.

Assa et al. [2] verify their results by conducting a survey to determine whether their system selected the keyframes chosen by users, achieving good results. For future work they plan to improve the key frame selection rules and analyze more complex motions. However, there is no way to determine how long a sequence had taken other than the implication that a longer sequence probably results in more keyframes shown. Furthermore, a relatively complex motion such as the sneaking sequence shown in Figure 5 resulted in needing more than one image to summarize the motion.

### Interactive Platforms

Other groups have opted to use a form of interactive platform or GUIs to visualize the features of a motion capture sequence. The most notable advantage of this approach is that the system is no longer restricted to summarizing the entire motion into a single or a few images. Furthermore, an interface means the visualization can be customized to some extent. Due to the interactive nature, these platforms can be easily be used in fields such as education, healthcare, movement annotation, dance, and other creative applications. On the other hand, the intricacies of the platform itself could render the visualization more difficult for a user to understand or at the very least require more learning time.

#### *Mova*

Alemi et al. [1] developed an interactive web-based platform for movement visualization called Mova. The aim of the platform is movement feature extraction, visualization, and analysis of movement data as well as to provide a research tool for the evaluation of movement feature extraction techniques. Mova consists of feature extraction components, a visualization engine, and the graphical user interface. In addition to the regular motion capture features, Mova is also able to extract Laban effort parameters [34], body shape measures, gestures, emotions, health conditions, and gait patterns. The visualization engine can illustrate a figure sketch of the movement as well as specific joints, using multi-hued color scales and highlighting for emphasis between figures. Mova also allows for animating of the visualizations for both full body and trajectories of individual joints. The complete source code is available at `https://github.com/omimo/Mova`.

The authors tested Mova using 3 theoretical scenarios. In the first case, the researcher implements feature extraction methods. By viewing the extracted features in parallel with imported annotated values, the researcher is able to compare the extracted and annotated values and validate them. In the second case, Mova can be used by experts in dance or healthcare to examine the movements of a dancer or patient and draw relevant conclusions. In the third case, Mova is used as the front-end for a movement database where the user can easily browse for data.

#### *MotionExplorer*

Bernard et al. [8] developed the MotionExplorer system for searching and exploring motion databases. Poses in the data are clustered and displayed as a hierchical tree structure as shown in Figure 6. A viewer can determine which motions are similar in this way. Furthermore, a viewer can make conclusions on how dissimilar two motions are based on the number of branches they are apart. An individual sequence can also be viewed in a graph format where nodes show key poses or frames and edges indicate the transition between poses. The graph representation, allows for visualization of cycles in the motion but is not able to clearly visualize the starts and ends of a motion. The nodes of the tree and graph structures are organized by color in terms of the similarity in features.

Based on the works of Balzer and Deussen [4] and Von Landesberger et al. [37], Bernard et al. allows for customization in the level of detail or aggregation that a user wants to view in the graph structure. In their design process, Bernard et al. included both experts and non-experts to refine the interface using questionaires and informal interviews. To test their interface and system, they asked five domain experts to test MotionExplorer and then give feedback in various aspects such as usability, effectiveness, and efficiency. The system was well-received.

#### *GestureAnalyzer*

With more specific applications in examination of gestures, Jang et al. [19] developed the GestureAnalyzer. Due to using a Kinect sensor to capture the data, their motion range is limited to gestures. Just like the MotionExplorer, the GestureAnalyzer uses a hierarchical clustering method to group their data into a tree structure. However, instead of showing the poses or frames as nodes in the tree, GestureAnalyzer simply uses text labels and IDs. They also make use of color in their tree organization.

An actual visualization of a gesture is shown in Figure 2 with the tree structures on the left. The poses are shown with the active joint highlighted in red, where the active joint is the limb used to perform the gesture. The highlighting allows a viewer to visually compare differences in joint movement between different motions. Gestures can also be viewed as an animation. The authors conducted user studies to test the effectiveness of their system and claims that GestureAnalyzer allowed users to easily identify gesture patterns. Jang et al. note that limitations of their system include the input data being restricted to gestures as well as being unable to quickly display a large amount of data. Furthermore, unlike MotionExplorer, GestureAnalyzer uses a predetermined clustering structure and does not allow users to modify.

*MotionFlow*

Building on the works of MotionExplorer and GestureAnalyzer, Jang et al. [20] later developed the MotionFlow system, also to be used for gesture analysis. Similar to the previous two systems, they cluster motions but using K-means, a partition-based structure. They claim this allows for easier cluster customization from the user.

Although there are similarities in the approach of MotionFlow compared to the two previous examples, they employ several differences as well. They maintained the tree, subtree, and graph structure. As in GestureAnalyzer, the gesture is visualized as an animation. However, they have also combined the trees with a flow diagram to indicate the transition of motion as well as a treemap, seen in (c). The thickness of lines in the tree indicate the frequency of transitions between those frames. The lines are also color-coded in accordance with the treemap to give an alternate representation to the tree.

The authors tested MotionFlow with user studies which introduced users to the system and then conducting interviews and questionaires afterwards. Results showed that MotionFlow was able to generate representative trees for the input gestures and that it was good for comparing multiple gestures. The authors note that a major limitation in MotionFlow is that they have been working with gestures that start with a common pose. Using different starting poses would result in messy trees due to the different root node.

### Video and Animation

Each of the interactive platforms can all show motion data as animation. Tools used to animate motion capture include Markerless Motion Capture, Autodesk, Xsens, and OptiTrack. There are also systems that are more focused in reproducing the raw data in the form of a video or animation. Examples include the work of Assa et al. [3] where they create a motion overview video and Lee et al. [24] where they visualize the motion as animated avatars. There are differences in the pre-processing of these approaches such as creating a camera path that are presented by aforementioned studies. It is undeniable that renderings can produce effective visualizations of movement. However, the issues of note for this form of visualization is that a viewer can obviously see the motion in its entirety. But as the entire sequence obviously has to be seen, they do not serve as good summaries or overviews of the motion, nor can viewers easily examine aspects such as particular limbs as in an interactive platform.

| System | Pre-processing | Visualization |
|---|---|---|
| Motion map | Clustering via SOM and keyframe extraction | Single image, simplified SOM |
| Motion cues | Keyframe extraction | Single image, keyframes with motion cues |
| Motion belt | Keyframe extraction | Single image, keyframes placed on horizontal timeline |
| Motion track | Clustering via SOM, LLE, and keyframe extraction | Single image, multiple colored tracks |
| Action synopsis | Keyframe extraction | Multiple images, keyframes with lines to indicate path and stroboscopic images |

**Table 1. Summary of the image-based systems presented in this paper.**

## DISCUSSION

As with any summary, the loss of data is inevitable. Combined with the high dimensionality of motion capture data, the visualization of said data is not a trivial problem. A common problem during the pre-processing process is computational load and scalability of the overall system.

Summarizing motions with images offers the advantage of being easy to understand. However, transforming the data from many frames to a single or several frames results in inevitable loss of data or features. The temporal aspect of the original motion is most often lost in the final visual representation. In almost all cases, the viewer can no longer determine the length of the original motion either in timespan or relative to other motions in a comparison. Even so, these approaches have shown good results within the contexts of their original goal, even if the goal usually involves other applications than simply producing an effective visualization.

The other common approach to visualizing motion is to use an interactive platform. Compared to producing images, this method has the benefit of being able to present different levels of detail or aspects of the raw data, such as visualizing individual joints. An interactive interface also allows for easier customization and input from the user regarding how the data should be handled. The user can manually tag the data and have more control during pre-processing. It is also common to make use of other visualization methods such as trees and graphs to depict the motion. However, the interactive interface is potentially more difficult for users to learn.

An important issue is that most systems do not test for the effectiveness of their visualization. This is especially evident in the systems producing image summaries or focused in videos and animations. Most of them test for the efficiency of their system such as search times in motion belt [40] and retrieval times in motion map [32]. Therefore, although we can ana-

| System | Pre-processing | Visualization |
|---|---|---|
| Mova | Feature extraction and computation | Horizontal timeline, colored highlighting, can be joint-specific, animations |
| MotionExplorer | Hierarchical clustering | Tree and graph views, color-coded clustering |
| GestureAnalyzer | Hierarchical clustering | Tree view, gesture-specific, colored highlighting, animations |
| MotionFlow | K-means partition-based clustering | Treemap, tree with flow-integration and graph views, gesture-specific, colored highlighting, animations |

**Table 2. Summary of the interactive platforms presented in this paper.**

lyze the advantages and disadvantages of their techniques, it is impossible to make an empirical statement about the effectiveness of the visualization itself. In fact, given the different goals and intents of these systems, there is not even a set of standard criteria with which to evaluate these visualizations.

Evaluation is more well-established with interactive systems, as all examples except Mova had documented use cases. Perhaps this is due to the recognition that an interactive platform would require more input from the user in creating the visualization. Even then, there is no standard set of questionaire or interview questions which is deemed required to fully evaluate the effectiveness of the visualization. Furthermore, only in the case of MotionExplorer was there input at the design stage from domain experts. Lastly, other than Mova, most systems do not explicitly make their code publicly available.

## CONCLUSION

This paper presents the current approaches to visualizing motion capture data. The process of acquiring motion capture data is explained by Bodenheimer et al. [10]. Commonly used and publicly available motion capture data can be found at the CMU [16], HDM05 [28], and MoDa [29] databases. Due to the high dimensionality of motion capture, pre-processing involving dimensionality reduction, clustering algorithms, or keyframe extraction is often required. After pre-processing, most groups produce a visualization of the motion using either images, interactive platforms, or less commonly, videos and animations. A summary of the image-based systems and interactive platforms presented in this paper can be found in Table 1 and Table 2 respectively. All of the works presented in this paper uses full body motion capture data as input. An exception is the action synopsis system by Assa et al. [2] also being able to use animation sequences or video clips and the GestureAnalyzer and MotionFlow using Kinect data.

As shown in the presented examples, attempts in visualizing motion often results in loss of data, especially in image-based systems. Using multiple interfaces, interactive platforms can show all available features in one way or another but has the drawback of potentially being a more complex system to learn for the user.

The presented systems also focus mostly on the representation of the whole body. Some exceptions such as Mova [1], and motion cues [11] has some support for individual limb visualizatio. GestureAnalyzer and MotionFlow also deal exclusively with gestures. However, this is also an area for potential future exploration, as it is feasible that many applications in fields such as healthcare or dance would be interested in more systems that tailor visualizations to certain limbs. Furthermore, with the development of more advanced sensors, different motion tracking data such as facial expression recognition, and eye tracking can be included in the visualization.

In the end, it is difficult to evaluate the effectiveness of one technique or system in the context of another. The image-based systems are especially lacking in explicit tests for the effectiveness of the visualizations. The interactive systems are more diligent in testing their systems with user studies, questionaires, and interviews. Even then there is not a standardized set of tests by which these platforms can be evaluated. As such, there is the potential in future demands for standardized criteria by which to evaluate all of these systems as well as tools to be developed for the purpose of evaluation. A good start could simply be to gather feedback specifically on the visualization aspects of these tools from end-users or visualization experts.

## REFERENCES


1. Alemi, O., Pasquier, P., and Shaw, C. Mova: Interactive movement analytics platform. In *Proceedings of the 2014 International Workshop on Movement and Computing*, MOCO '14 (2014), 37–42.

2. Assa, J., Caspi, Y., and Cohen-Or, D. Action synopsis: Pose selection and illustration. *ACM Trans. Graph. 24*, 3 (2005), 667–676.

3. Assa, J., Cohen-Or, D., Yeh, I.-C., and Lee, T.-Y. Motion overview of human actions. *ACM Trans. Graph. 27*, 5 (2008), 1–10.

4. Balzer, M., and Deussen, O. Level-of-detail visualization of clustered graph layouts. In *Visualization, 2007. APVIS'07. 2007 6th International Asia-Pacific Symposium on*, IEEE (2007), 133–140.

5. Barbič, J., Safonova, A., Pan, J.-Y., Faloutsos, C., Hodgins, J. K., and Pollard, N. S. Segmenting motion capture data into distinct behaviors. In *Proceedings of Graphics Interface 2004* (2004), 185–194.

6. Benesh, R., and Benesh, J. *Reading dance: The birth of choreology*. International Specialized Book Services, 1977.

7. Berkhin, P. A survey of clustering data mining techniques. In *Grouping multidimensional data*, Springer (2006), 25–71.

8. Bernard, J., Wilhelm, N., Kruger, B., May, T., Schreck, T., and Kohlhammer, J. Motionexplorer: Exploratory search in human motion capture data based on hierarchical aggregation. *Visualization and Computer Graphics, IEEE Transactions on 19*, 12 (2013), 2257–2266.

9. Bobick, N. Rotating objects using quaternions. *Game Developer 2*, 26 (1998), 21–31.

10. Bodenheimer, B., Rose, C., Rosenthal, S., and Pella, J. *The process of motion capture: Dealing with the data*. Springer, 1997.

11. Bouvier-Zappa, S., Ostromoukhov, V., and Poulin, P. Motion cues for illustration of skeletal motion capture data. In *Proceedings of the 5th International Symposium on Non-photorealistic Animation and Rendering*, NPAR '07, ACM (2007), 133–140.

12. Bulut, E., and Capin, T. Key frame extraction from motion capture data by curve saliency. In *Computer Animation and Social Agents* (2007), 63–67.

13. Carlson, K., Schiphorst, T., and Shaw, C. Actionplot: A visualization tool for contemporary dance analysis. In *Proceedings of the International Symposium on Computational Aesthetics in Graphics, Visualization, and Imaging* (2011), 113–120.

14. Cutting, J. E. Representing motion in a static image: constraints and parallels in art, science, and popular culture. *Perception 31*, 10 (2002), 1165–1194.

15. Friedman, S. L., and Stevenson, M. B. Developmental changes in the understanding of implied motion in two-dimensional pictures. *Child Development* (1975), 773–778.

16. Gross, R., and Shi, J. The cmu motion of body (mobo) database.

17. Guest, A. H., and Anderson, D. *Labanotation; Or, Kinetography Laban: The System of Analyzing and Recording Movement. Illus. by Doug Anderson*. Oxford University Press, 1970.

18. Hu, Y., Wu, S., Xia, S., Fu, J., and Chen, W. Motion track: Visualizing variations of human motion data. In *Visualization Symposium (PacificVis)* (2010), 153–160.

19. Jang, S., Elmqvist, N., and Ramani, K. Gestureanalyzer: visual analytics for pattern analysis of mid-air hand gestures. In *Proceedings of the 2nd ACM symposium on Spatial user interaction*, ACM (2014), 30–39.

20. Jang, S., Elmqvist, N., and Ramani, K. Motionflow: Visual abstraction and aggregation of sequential patterns in human motion tracking data. *Visualization and Computer Graphics, IEEE Transactions on 22*, 1 (2016), 21–30.

21. Jolliffe, I. *Principal component analysis*. Wiley Online Library, 2002.

22. Kohonen, T. The self-organizing map. *Neurocomputing 21*, 1 (1998), 1–6.

23. Larboulette, C., and Gibet, S. A review of computable expressive descriptors of human motion. In *Proceedings of the 2nd International Workshop on Movement and Computing*, ACM (2015), 21–28.

24. Lee, J., Chai, J., Reitsma, P. S. A., Hodgins, J. K., and Pollard, N. S. Interactive control of avatars animated with human motion data. *ACM Trans. Graph. 21*, 3 (2002), 491–500.

25. Liu, F., Zhuang, Y., Wu, F., and Pan, Y. 3d motion retrieval with motion index tree. *Computer Vision and Image Understanding 92*, 2 (2003), 265–284.

26. Malmstrom, C., and Zhang, Y. Mocomp: A tool for comparative visualization between takes of motion capture data. *Under review* (2016).

27. McCloud, S. Understanding comics: The invisible art. *Northampton, Mass* (1993).

28. Müller, M., Röder, T., Clausen, M., Eberhardt, B., Krüger, B., and Weber, A. Documentation mocap database hdm05.

29. Nixon, M., Bernardet, U., Alaoui, S., Alemi, O., Gupta, A., Schiphorst, T., DiPaola, S., and Pasquier, P. Moda: an open source movement database. In *Proceedings of the 2nd International Workshop on Movement and Computing*, ACM (2015).

30. Palazzi, M., Shaw, N. Z., Forsythe, W., Lewis, M., Albright, B., Andereck, M., Bhatawadekar, S., Ban, H., Calhoun, A., Drozd, J., et al. Synchronous objects for one flat thing, reproduced. In *ACM SIGGRAPH 2009 Art Gallery* (2009), 37.

31. Perret, R. Bodycloud. *Master's thesis* (2009).

32. Sakamoto, Y., Kuriyama, S., and Kaneko, T. Motion map: image-based retrieval and segmentation of motion data. In *Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation* (2004), 259–266.

33. Shoemake, K. Animating rotation with quaternion curves. In *ACM SIGGRAPH computer graphics*, vol. 19, ACM (1985), 245–254.

34. Silang Maranan, D., Fdili Alaoui, S., Schiphorst, T., Pasquier, P., Subyen, P., and Bartram, L. Designing for movement: Evaluating computational models using lma effort qualities. In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems* (2014), 991–1000.

35. Subyen, P., Schiphorst, T., and Pasquier, P. Emviz (flow): an artistic tool for visualizing movement quality. In *Proceedings of Electronic Visualisation and the Arts(EVA 2013)* (2013).

36. Summers, D. Maniera and movement: the figura serpentinata. *Art Quarterly 35*, 3 (1972).

37. Von Landesberger, T., Kuijper, A., Schreck, T., Kohlhammer, J., van Wijk, J. J., Fekete, J.-D., and Fellner, D. W. Visual analysis of large graphs: state-of-the-art and future research challenges. In *Computer graphics forum*, vol. 30, Wiley Online Library (2011), 1719–1749.

38. Wu, S., Xia, S., Wang, Z., and Li, C. Efficient motion data indexing and retrieval with local similarity measure of motion strings. *The Visual Computer 25*, 5-7 (2009), 499–508.

39. Xiao, J., Zhuang, Y., Yang, T., and Wu, F. An efficient keyframe extraction from motion capture data. In *Advances in Computer Graphics*. Springer, 2006, 494–501.

40. Yasuda, H., Kaihara, R., Saito, S., and Nakajima, M. Motion belts: Visualization of human motion data on a timeline. *IEICE Trans. Inf. & Syst. 91*, 4 (2008), 1159–1167.