Motion Cues for Illustration of Skeletal Motion Capture Data

Simon Bouvier-Zappa Victor Ostromoukhov Pierre Poulin

LIGUM, Dép. I.R.O., Université de Montréal

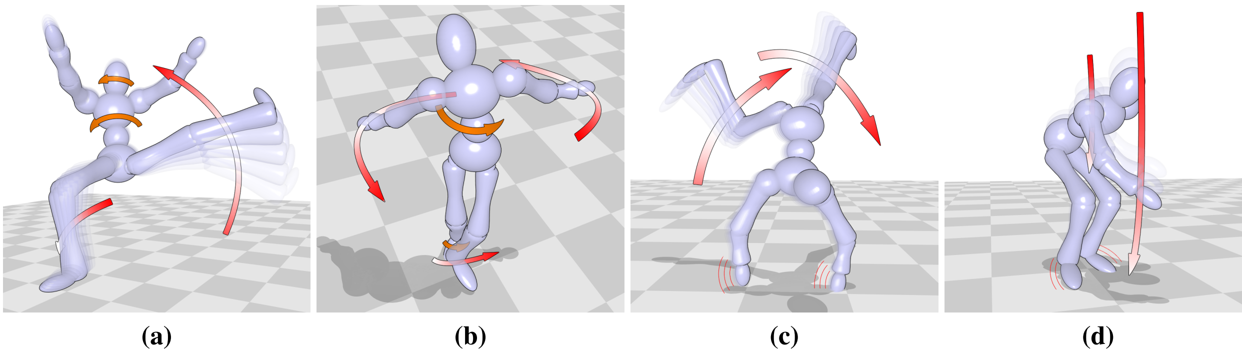


Figure 1: *Non-photorealistic illustration of motion capture sequences:* ***(a)*** *spin kick,* ***(b)*** *dancing pirouette,* ***(c)*** *cart wheel,* ***(d)*** *bending-over. Each motion is emphasized using motion arrows, noise waves, and/or stroboscopic motion.*

Abstract

There are many applications for which it is necessary to illustrate motion in a static image using visual cues which do not represent a physical entity in the scene, yet are widely understood to convey motion. For example, consider the task of illustrating the desired movements for exercising, dancing, or a given sport technique. Traditional artists have developed techniques to specify desired movements precisely (technical illustrators) and suggest motion (cartoonists) in an image.

In this paper, we present an interactive system to synthesize a 2D image of an animated character by generating artist-inspired motion cues derived from 3D skeletal motion capture data. The primary cues include directed arrows, noise waves, and stroboscopic motion. First, the user decomposes the animation into short sequences containing individual motions which can be represented by visual cues. The system then allows the user to determine a suitable viewpoint for illustrating the movement, to select the proper level in the joint hierarchy, as well as to fine-tune various controls for the depiction of the cues themselves. While the system does provide adapted default values for each control, extracted from the motion capture data, it allows fine-tuning for greater expressiveness. Moreover, these cues are drawn in real time, and maintain a coherent display with changing viewpoints.

We demonstrate the benefit of our interactive system on various motion capture sequences.

CR Categories: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation I.3.3 [Computer Graphics]: Picture/Image Generation—Display and Viewing algorithms

1 Introduction

The verbal conveyance of an idea is a powerful medium for sharing precise thoughts. However, a well-illustrated image may also bring an instant understanding of the subject matter. For example, Tufte [1997] says that "*to document and explain a process, to make verbs visible, is at the heart of information design*." This principle has always been a concern in illustrations of all kinds, from strict technical designs to more permissive cartoon comics. Visual cues are powerful tools in graphic arts that compensate the inherent limitations of the visualization medium. Motion cues fall into this category, emphasizing temporal information from a dynamic 3D world into a static 2D representation.

Motion capture is a popular technique to animate articulated characters from tracked markers on real actors. It inherits the actor's natural motions, and therefore, offers great potential to illustrate technical motions. However illustrating motion capture data in a single image can be very difficult. Most often, sequential key poses are simply drawn, which can lead to very cluttered and incomprehensible images. In fact, motion capture data encodes hierarchically and temporally complex articulated models of characters, and as such, it holds tremendous motion information. We propose a method that focuses on illustrating motion suggested by a static pose of such an animation. To do so, we introduce a set of motion cues that allows adaptive illustration of motion capture data. Clarity of information conveyed by a single picture is a subjective notion. Thus, we propose a method that will make use of an intuitive set of parameters to adapt an illustration to a user's perception of motion.

We propose the use of a hierarchy that isolates "illustrable" parts of a skeletal motion capture sequence. With this method, we are able to illustrate motion at different levels of precision. We make use of motion arrows, noise waves, and stroboscopic motions. These three motion cues successfully illustrate most movements when used conjointly. Moreover, we present illustrations by key poses selection and foot step highlighting.

In this paper, we illustrate the use of motion cues with motion capture data, specifically tailored to skeletal animations. Even though it is not the objective of our research, our motion analysis and illustration can be performed in real time, which offers great potential for additional improvements. Such illustrations can be of great use for automatic generation of precise technical maneuvers pictures.

2 Related Work

Illustrating motion in still images has always been a complex task, as the medium itself does not offer 4D support. Tufte [1997] exposes this flaw of 2D media in his work, but also suggests how to make use of the limited media to build temporal representation. Perceptually, an instantaneous image in time does not accurately represent what the human eye sees. Cutting [2002] introduces the *moment*, a temporally unrestricted image that allows the viewer better temporal reconstruction of the action by "*capturing the psychological truth of the event*".

Cartoonists have historically employed various motion indicators to improve information conveyed by illustrations. McCloud [1993] described the use of action lines, stroboscopic effects, and streaking effects as intuitive perceptually imposing ways of suggesting motion. These concepts can also be applied to computer animation. Masuch *et al.* [1999] introduced methods for integrating speed lines into computer animation. Based upon this work, Kawagishi *et al.* [2003] introduced several cartoon-blur techniques for 2D animations. Moreover, Joshi *et al.* [2005] proposed an approach to illustrate speed lines, flow ribbons, and strobe silhouettes for voxelized 3D models. Collomosse *et al.* [2005] introduced a framework for illustrating 2D video sequences with streak lines, ghost lines, and motion blur effects. Nienhaus and Döllner [2003; 2005] presented general tools for illustrating 3D animations exploiting various structures from the scene graph. Their work is similar to ours in spirit. Since they can access any information in the scene graph, they can produce very general illustrations, just like any complete animation system. In contrast, our approach is specifically tailored to motion capture data from articulated figures, and we concentrate our efforts on more automatic motion capture analysis and illustration, given the greater availability and potential enclosed with such data.

While action lines hold great descriptive power, stroboscopic display can be more effective for some animations. Hence, Assa *et al.* [2005] proposed a method for isolating key poses of a motion capture animation. Agarwala *et al.* [2004] also introduced a method for stroboscopic illustration in their digital photomontage framework.

Other motion enhancers provide less information, but still hold artistic value. Liu *et al.* [2005] introduced a technique to amplify deformations produced by subtle motions in order to enhance their perceptual visibility. With his cartoon dioramas, Raskar *et al.* [2002] proposed a way to simulate motion on a static physical model by projecting animation on it. They used affine shearing to simulate acceleration in the model. Also, Brostow and Essa [2001] introduced a technique that produces directed photographic motion blur for 2D animations. Wheeler *et al.* [2006] proposed methods to simulate similar blur for 3D animations.

Technical design holds a more strict temporal illustration. Agrawala *et al.* [2003] proposed a method to generate psychologically acute assembly instructions. Storyboarding illustration uses a similar notation to cartoon. However, since this particular medium must convey technical information to film makers, more precise motion cues will be favored for characters and cameras [Hart 1999]. Dony *et al.* [2005] proposed techniques to simulate cartoon-like trailing line effects on moving characters for storyboard editing. Recently, Goldman *et al.* [2006] extended storyboard illustration with use of motion arrows on both characters and camera movements.

Using 3D animation data, our interactive system will semi-automatically illustrate skeletal motion capture data. We will use technical and cartoon motion cues to convey motion information in a single static 2D image. While motion capture movement segmentations produce interesting results [Barbic et al. 2004; Kovar and Gleicher 2004] for identifying key poses, we offer a compromise between these poses, and manual selection or segmentation by the user, when he judges necessary.

3 Motion Capture Hierarchy

Motion capture data for articulated figures is traditionally organized and stored as a hierarchy of bones (geometry). Each bone has an orientation, expressed with a quaternion, around its coordinate system (bone origin). This orientation is regularly sampled in time into frames (poses).

For the purpose of illustration, we assume we have a selected pose (time) for the articulated figure, and all motion data within the preceding and following key poses. These key poses can be computed analytically, or manually segmented by the user.

At first, consider the illustration of a single bone. Parameter $\gamma \in [0, 1]$ determines a point along the length of this bone. We extract the sequence of points at γ on this bone, for all frames within the two key poses. This sequence defines the point's *motion curve* through time. This motion curve can be represented in two possible spaces. The *local motion curve* with coordinates that origin from the local bone joint referential thus represents the bone motion around its axis. The *global motion curve* represents the bone motion in world space coordinates. The motion curve in both spaces will be used in Section 4.1. The motion curve will not give good results for all situations, therefore, we also use the quaternion data attached to the bone data to evaluate rotations. By applying thresholding on motion curve values, we are able to analyze and illustrate most motions with the motion cues proposed in Section 4.

While the above technique works for one bone, we would prefer to benefit from the hierarchical nature of animated skeletons. Besides, we should not try to illustrate motion on all skeleton bones as this would generate too much information. Therefore, we assemble groups of bones into meaningful anatomical hierarchical structures, given the motion capture skeleton. For each group, we must identify suitable positions on the geometry of the skeleton for which motion illustration would be unambiguous. Therefore we construct an articulated path as a continuous linear sequence of bones, and distance on this path is simply defined over the sum of its bone lengths. The same parameter $\gamma \in [0, 1]$, as illustrated in Figure 2, is intuitively applied on the length along this path, and the same technique on the resulting point defines the motion curve. We use quaternion data at the joint of the root of the articulated path to evaluate rotations.

For certain groups of bones, often higher up in the hierarchical structure, there is no representative position that can be suitably moved on the geometry of the skeleton (*e.g.*, upper body with two arms). For such a group, we indicate one representative point that will not be affected by γ.

Based on the original motion capture skeletal data, we recursively subdivide the skeleton bones into a balanced tree of meaningful groups of bones to form levels of detail for the illustration methods. We define the *skeletal node*, a single node of this tree which refers to an animated sub-skeleton (*i.e.*, a group of bones) of the motion capture skeleton (see Figure 3). Our tree is built in such a way that a parent node must contain the same set of bones as all its children's node's bones. This way, we make sure that each level of the tree includes the whole set of bones contained in the original motion capture data. This particular representation holds another benefit, while illustrating animations. Depending on the current motion, a different level of the motion capture hierarchy may be used to perform illustration. Typically, for a more detailed illustration, the last level of the hierarchy will be favored, while the first level will be used for more general illustrations, such as displacements of an entire body.

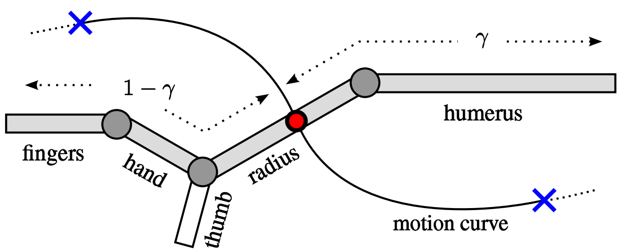


Figure 2: *A group of five bones forms an arm with one fork at the radius bone. The articulated path is the longest branch of the skeleton data in the articulated figure. It contains the humerus, the radius, the hand, and fingers. A motion curve goes through a fixed position on the geometry at a ratio of γ, starting at the root of the articulated path for this group of bones.*

3.1 Human Skeleton Hierarchy

We defined a grouping strategy robust enough to fit any kind of motion capture data represented through animated skeleton hierarchies. Most of our work is applied to human motion capture data, therefore, we define precise groupings of skeleton motion capture data that perceptually segment the human body skeleton in motion representative parts (see Figure 3).

Consider the animated skeleton as a tree that goes from the root of the skeleton (*i.e.*, its pelvis) to its extremities (*i.e.*, feet, hands, head). We can define perceptual groups of bones with each fork in the bones hierarchy. As such, the skeleton is subdivided into upper body and lower body sub-skeletons, and is recursively subdivided to refine separation of the skeleton anatomical parts.

Our groupings for the human motion capture skeleton define four recursive levels of detail. The first level refers to the entire animated skeleton motion data. This level never changes, regardless of the animation which is illustrated. The second level separates the body into two entities, the upper body and lower body. The third level isolates the limbs of the skeleton, as well as its torso and its head. This level defines a precise yet general segmentation of the skeleton body parts and is frequently used in illustrations. The fourth level is our maximal separation allowed for illustration with our human skeleton. Hence, we use this level for more complex motions requiring descriptive illustrations. We did not consider hands and feet bones as most animated skeletons we used did not provide such motion.

4 Motion Cues

Motion cues differ in the degree of information they add to motion. We can classify these indicators according to their evocativeness, their clarity, the motion direction or motion precision they suggest to the viewer [Cutting 2002]. We propose a set of indicators which convey, on various degrees, these properties. In our system, we make use of motion arrows, noise waves, stroboscopic motion, poses illustration, and foot steps illustration.

The motion arrow is the most powerful indicator of motion we synthesize in our work (Section 4.1). Its shape provides an effective representation for direction, velocity, and significance of motion.

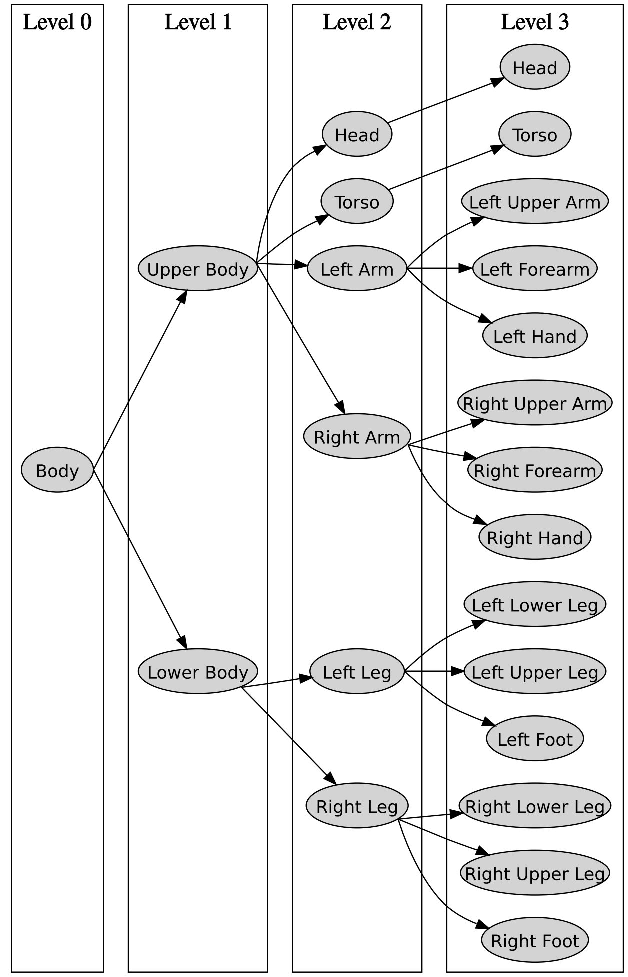


Figure 3: *Perceptual groupings of animated skeleton for human motion capture data illustration. We present anatomical subdivisions that recursively create four possible levels of illustration. Each level contains the whole skeleton data, but segmented into articulated sub-figures referred by skeletal nodes. Some nodes, such as the head and torso, are duplicated into two different levels to allow the creation of a balanced tree.*

On the contrary, noise waves (Section 4.2) produce only hints of motion. They are useful for uncertain motions, or hesitations that have no clear directions. These two cues can easily be used in conjunction with a stroboscopic display (Section 4.3) of previous poses within the illustrated sequence. It can often help in clarifying a static image.

To produce global illustrations of an entire skeleton under displacement, our system uses two traditional techniques. It first handles illustration of the key poses (Section 4.4) of the motion capture data, joined by motion arrows generalized for displacements. Furthermore, we also illustrate the sequence of key foot steps (Section 4.5). Foot steps sequence is frequent for certain motions, such as dancing steps or tactical maneuvers.

4.1 Motion Arrows

Our motion arrows are applied to represent two rigid transformations: translation and rotation. These transformations can successfully represent most motion capture movements. Arrows are usually preferred [Goldman et al. 2006] for their higher precision in motion illustration to cues such as streak lines or ghost lines [Collomosse et al. 2005], motion blur [Brostow and Essa 2001; Kawagishi et al. 2003], or dynamic shearing [Raskar et al. 2002].

To illustrate a motion, the user first selects a pose, for instance at time t_i. He then selects the level in the motion hierarchy at which the illustration will be computed. In our examples, the highest (most detailed) levels were usually preferred, as they convey more information for detailed illustrations of more complex (and interesting) motions. Each skeletal node at this level will result in at most one motion arrow illustration. This motion arrow spans the entire selected sequence that encompasses t_i, that is all poses between the previous key pose and the next key pose.

The decision of representing a movement or not by an arrow, and whether to use a translation or a rotation arrow, is based on an analysis of the motion curve for the point at γ in the local space of its skeletal node. Because translation appears to be predominant, and can even replace some rotations, we first try to fit a translation arrow to the motion curve. Our fitting criteria are based upon the total length of the curve, its average velocity, and its average acceleration. We therefore favor longer and faster motion curves. For rotation arrows, our method uses the quaternion data at the skeletal node. When a bone, or set of bones rotates around its orientation axis (*i.e.*, it rolls), the motion curve for a point at γ might not bring satisfactory results. In such a case, we extract the roll angle from the quaternions. To do so, we compute the roll angle between the two extrema quaternions recorded within the two key poses.

Illustration of motion arrows uses scene information displayed through the motion curve. A translation arrow geometry approximates the complete motion curve over the sequence by a quadratic curve, through least-square interpolation. We found that a quadratic curve often conveys sufficient motion information, while retaining aesthetic characteristics and flexibility for efficient display. Our tests with curves of higher degrees were much less conclusive. The rotation arrow uses the quaternion at the current pose t_i to retrieve the referential axis around which the arrow is displayed. Our method creates a circular arc geometry which illustrates the roll rotation. Furthermore, the arc radius is automatically adapted to the character shape. Using the skeletal node, we consider the maximal distance between the bone axis and the skinned geometry as the arc radius for our illustrations. Motion arrows have several parameters to control the arrow shape (see Figure 4).

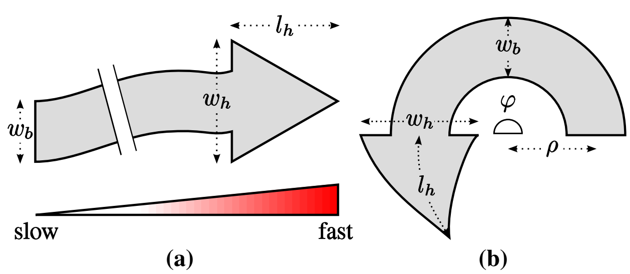


Figure 4: *Motion Arrows. **(a)** The translation arrow is defined by a quadratic curve. The local color saturation indicates the motion velocities along the illustrated sequence. The arrow length depends on the length of illustrative curve. **(b)** The rotation arrow is defined by a portion of circle around the featured node referential. Its arc angle φ is fixed to a default value but can be fine-tuned by user. Its radius ρ is dependent on the illustrated character geometry. Body width w_c, head width w_h, and head length l_h are parameters of both types of arrows. In our implementation, we fixed the values such that $4w_c = 2w_h = l_h$.*

Our method of illustration ensures viewpoint robustness on the motion arrow. Our method modifies the orientation of the arrow's body and head so that its geometry is always perpendicular to the view direction, hence offering maximal display coverage toward the viewer. To do so, we divide the arrow body into a number of short segments. Each segment is oriented along its tangent. We evaluate the cross product between this direction and the camera look-at vector, which defines the orientation for the width dimension of the arrow segment in 3D space. Near cross product degeneracies, we fix the orientation of the arrow to the last calculated vector for all subsequent segments in the arrow. We favor planar display of the arrow head as it is the most representative part of the geometry. Hence, arrow construction starts at the head and finishes at the tail. Viewpoint robustness constraints in our system make it impossible to provide twisting freedom to arrow.

The arrow must be visible at all time in the illustration and thus, must not be occluded by the character's geometry. We proceed with a two-pass rendering of the skeleton character to enable geometry transparency, only when it occludes an arrow. The arrows are first rendered in the stencil buffer. The character is then rendered as transparent when the stencil is 1 and opaque when the stencil is 0 (no arrow). Stroboscopic characters (Section 4.3) are always rendered as transparent.

4.2 Noise Waves

Noise waves are used to illustrate subtle movements in the motion capture data. A noise wave is a type of action line mostly used in cartoons to emphasize jiggly motions [Dony et al. 2005; Collomosse et al. 2005; McCloud 1993]. This motion cue is therefore not about precision, but seeks to highlight movements that cannot be well illustrated through motion arrows [Liu et al. 2005].

Again, assume illustration of a sequence for a pose at time t_i, at a selected level of the motion hierarchy, and at positions defined by γ. This creates a motion curve for each skeletal node at this level.

We analyze each motion curve separately, in global 3D space. Noise waves will appear when the motion has a larger amplitude than noisy data th_{min} in typical motion capture data, but a lower amplitude than those subjected to more advanced illustrations th_{max}, such as motion arrows (see Figure 5a). The length of the diagonal d of the axis-aligned 3D bounding box for the motion curve determines the amplitude of the motion.

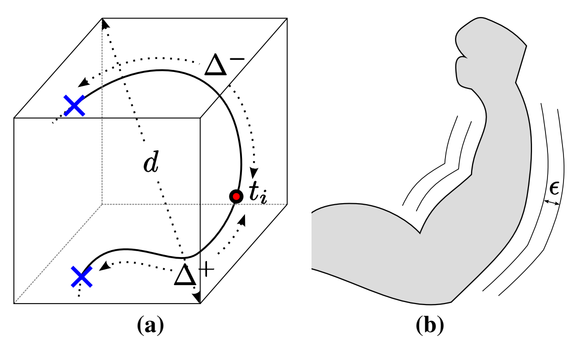


Figure 5: *Noise Waves. **(a)** Analysis for noise waves refers to the bounding box of the motion curve at time t_i. Noise waves are displayed when $th_{min} < d < th_{max}$. **(b)** A noise wave is illustrated through n_w repetitions of the character geometry outline. A distance of ϵ separates each outline. In our implementation, we fixed the wave line width to 1 pixel to reduce its visual importance.*

We simulate noise waves by rendering the outline of the character geometry a number of times with an increasing scaling factor (see Figure 5b). To evaluate whether the wave is unidirectional or bidirectional, we compute approximate directions for the motion curve before and after t_i using least-square fitting. If the two portions of the curve follow the same direction, the motion is considered unidirectional and only its associated side of the noise waves is displayed. Otherwise, the motion is considered bidirectional. Similarly to motion arrows, the appearance of the noise waves adapts to the viewpoint. We gradually erase these noise waves when the wave direction becomes parallel to the look-at vector to avoid misleading information.

4.3 Stroboscopic Motion

Stroboscopic illustration of an animation replicates several poses of the animation to reconstitute the motion in a single static image [Agarwala et al. 2004; Cutting 2002; McCloud 1993]. This particular representation of motion is widely used in photography, in drawings, and more recently in computer graphics. It holds an easy-to-understand form without requiring technical symbols of any sort. While stroboscopic motion has similarities with motion blur, the two motion cues do not convey the same level of information. Motion blur usually filters the image by blending continuous motion to produce a basic emphasis of motion. It cannot reproduce as well a motion trajectory composed by discrete stroboscopic replicates.

Our implementation of stroboscopic motion simply uses the direct information from the motion capture data to draw with an increasing transparency term α, n_s previous frames of the current motion sequence. These frames can be consecutive in time, or separated by a given number of frames. Transparency is proportional to a user-defined value α_0, which enables to regulate the fading effect of the stroboscopic motion (see Figure 6). This approach displays a discretized sample of poses through time, that roughly approximates the general motion in a single illustration.

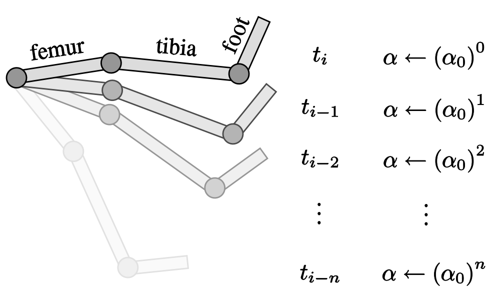


Figure 6: *Stroboscopic motion illustrates previous motion through n_s poses of increasing transparency α. Using transparency value α_0, transparency for previous frame t_{i-j} is defined as $(\alpha_0)^j$.*

4.4 Key Poses Illustration

Illustration of key poses can help understand an animation as a whole by juxtaposing multiple key poses of the animation in a single image [Assa et al. 2005]. However, it may be difficult to mentally link multiple key poses to reconstruct the original action. Similarly, displaying a fraction of all intermediate poses may still overload the resulting image. We build on the work of Assa *et al.* [2005] to detect key poses, but apply our translation motion arrows (see Section 4.1) to illustrate movements between these poses (see Figure 10).

Starting from an automatically computed set of key poses, our system illustrates the intermediate movements with translation arrows by analyzing the information given by the skeletal animation in time. Typically, because we mainly search for general movements between two key poses, we concentrate the analysis on the first or second levels of motion hierarchy (see Figure 3), by applying the method described in Section 4.1. This time, we illustrate a virtual pose at time t_i. This pose is comprised between two key poses of the sequence which define the motion curve for analysis and illustration.

The key poses illustration is naturally more effective with spatially well spread-out animations such as walking and running.

4.5 Foot Sequence Illustration

A foot step sequence illustration highlights on the ground traces (prints) left by the feet through the animation. Such a representation can help understand complex feet sequences that could be difficult to illustrate with other motion cues. While more precise visual cues exist in dance notation, a foot step sequence holds an intuitive representation easy to relate to [Hutchinson 1984]. We propose this representation as a possible extension to our illustration technique.

For this particular motion cue, we make use of the research made into foot plant retrieval in motion capture and enforcement to prevent the well-known foot skating artifact [Glardon et al. 2006; Kovar et al. 2002]. We use the motion hierarchy to retrieve the feet of the character, and identify foot plant poses. A manual foot plant selection is also available to fine-tune the desired illustration.

For a particular foot plant, we illustrate the footprint geometry in 2D on the ground by aligning it with the skeletal foot axis at time t_i. The footprints associated with one foot share a unique color, different from the other foot. We use again translation motion arrows (see Section 4.1) to illustrate on the ground the in-between movement connecting for each foot, one footprint to the next one. Finally, to improve the understanding of the sequence, numbers are drawn on each footprint geometry, oriented with the footprint (see Figure 11).

While our classic in-between arrow illustration proves sufficient for short foot steps animations, it may not respond as well to long complex animations. More work is still needed to reduce the number of overlapping footprints and arrows.

5 Results

We have presented in the previous sections a set of methods that enable robust analysis and illustration of movements from motion capture data. We apply these methods to various examples of skeletal motion capture data. Our method successfully illustrates complex motion for a single pose of animation (see Figures 1, 7, 8, and 9), or for multiple poses combined in a single image (see Figures 10 and 11).

To illustrate human motion capture data, we use the perceptually based motion hierarchy proposed in Section 3.1. Our examples illustrate subtle motions, which require to select levels 2 or 3 in the motion capture hierarchy. We have not applied any pre-processing to the motion capture data used for our results. Our method can handle the raw data, while the user may modify parameters of the method to his liking. Analysis and illustration of motion cues are fast enough to enable manipulation in real time.

Figure 7 illustrates in four images various stages of an exercising motion capture sequence. The character performs jumping jacks, side twists, arm rotations, and reaches for his feet. No single image could illustrate all these motions, but multiple images give a good temporal reconstitution. We used the second level of the motion hierarchy for this animation. Our default illustrations produced good results, but we fine-tuned some parameters to improve visual cues. In the first image, we increased the value of γ to obtain more circular arrows at the forearms. In the last image, we increased the number of key poses in motion analysis to get longer motion curves and therefore, longer arrows. Such fine-tuning only took about 10 seconds and gave fast, intuitive update of the motion cues.

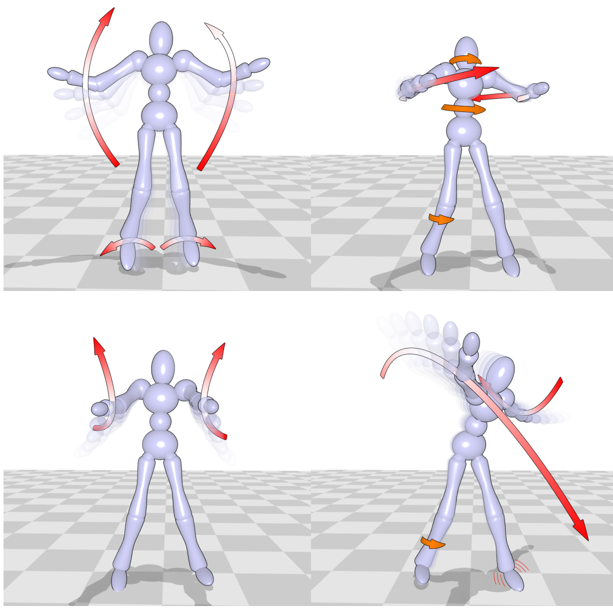


Figure 7: *An exercising sequence illustrated in four images. These four images combine motion arrows, stroboscopic motion, and noise waves.*

A weight-lifting sequence is illustrated in two images in Figure 8. As lifting involves movements of the forearm, we illustrate this motion with the third level of the hierarchy to obtain more precise results. Notice the noise waves appended to character's upper arms. While there is no visually present motion, the character struggles to lift the weights, thus initiating subtle directed noise waves.

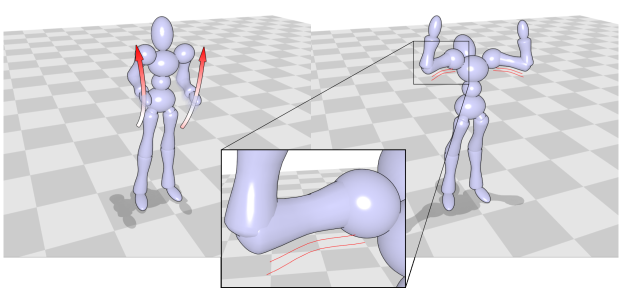


Figure 8: *A weight-lifting sequence illustrated in two images. These two images combine motion arrows and noise waves.*

Figure 9 illustrates a character jumping forward and kicking a ball in two separate images. Notice colors in the motion arrows appended to the right leg of the character in the two images. The gradient emphasizes the decline of speed when the character jumps in the first image and the boost of speed at the end of the final kick in the second image.

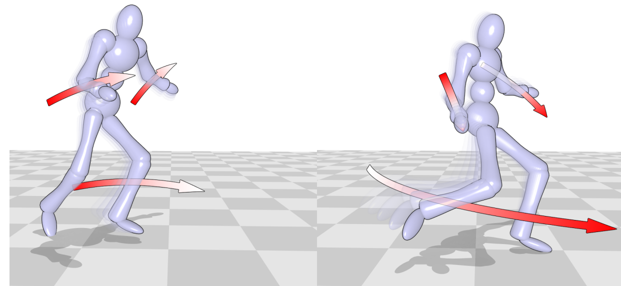


Figure 9: *A soccer kick sequence illustrated in two images. Motion arrows enhanced with stroboscopic motion were sufficient for this animation.*

Our key poses illustration in Figure 10 shows an action sequence from an obstacle-avoidance walk with eight different poses. In this image, the length of the arrows is set to half the distance between key poses. We increased the default width to enhance arrow's visibility. Motion arrows are appended only where temporal coherence could be ambiguous. Our system automatically discarded arrows between the first and second poses, and between the last two poses, according to their length and mean velocity.

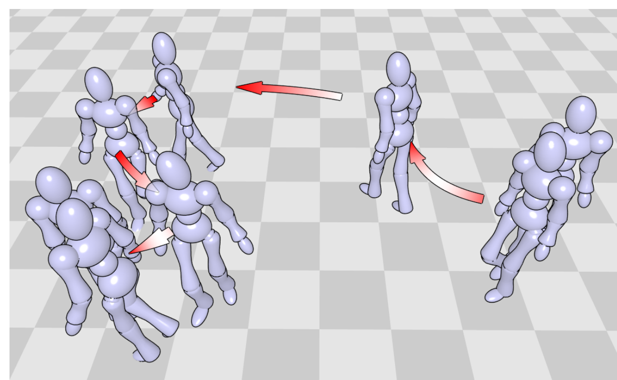
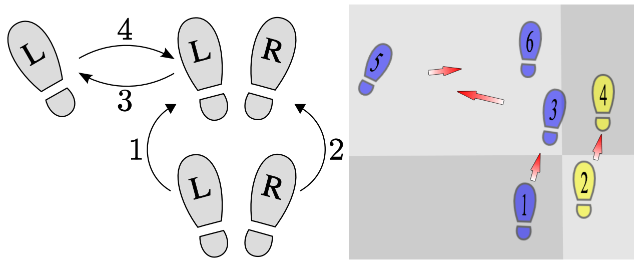


Figure 10: *Eight key poses illustrate the movements in an obstacle-avoidance walk sequence.*

Figure 11 shows foot steps illustration for a lambada dance. The character performs a sequence of six steps, which forms a repetitive pattern throughout the animation. We reduced the width for the default arrows and their lengths to fit a smaller representation and prevent overlapping the arrows and foot prints.

Figure 1 shows four different motion capture sequences for which the motion is illustrated within a single pose. The spin kick pose (Figure 1a) represents a character spinning in mid-air while giving a rotating kick with his left leg. These movements are illustrated respectively with two rotation arrows around the torso and head, and a longer translation arrow near the tip of the leg. The pirouette pose (Figure 1b) illustrates a character spinning on himself. We use motion arrows to emphasize the spinning motion on the character's torso and translation arrows on both arms. Moreover, the character slides his left foot and spins standing on his right foot. We respectively emphasize the prior motion with a translation arrow on the right ankle and with a rotation arrow around the left ankle. The cart wheel pose (Figure 1c) illustrates a character tipping his body over in a wheel motion. We emphasize the translation motion of the character's legs by appending long curved arrows. Moreover, we il-



(a) (b)

Figure 11: *Foot sequence.* ***(a)*** *Technical illustration of a lambada dance steps. This image was hand drawn and illustrates common concepts found in dance steps representation [Hutchinson 1984].* ***(b)*** *Our illustration for the same steps using motion capture data. The left steps are displayed in blue; the right in yellow. Numbers and arrows are appended to the illustration to increase temporal sequence.*

lustrate the hands subtle motion with noise waves. While the hands do not move, they slightly shift in the main movement direction, thus illustrated with unidirectional noise waves. The bending-over pose (Figure 1d) depicts a character crouching to reach a virtual object on the ground. We illustrate the crouching motion with a translation arrow on the character's back. A second motion arrow is appended to the hand to indicate a picking-up motion.

Table 1 details the specifications for the illustration parameters used for the teaser images.

Sequence	Parameters						
		Arrows		Waves		Strob.	
	γ	w_c	φ	n_w	ϵ	n_s	α_0
Spin kick	0.65	0.60	160^o	–	–	6	0.45
Pirouette	0.75	0.50	160^o	–	–	4	0.30
Cart wheel	0.60	1.00	–	3	0.55	10	0.40
Bend-over	0.75	0.60	–	2	0.60	7	0.30

Table 1: *Parameters used for the teaser images (see Figure 1). While automatic illustration provided satisfying results, the real-time display allowed to fine-tune them within seconds with these final parameter values.*

An accompanying video sequence and additional illustrations are available at the website associated with this paper from *www.iro.umontreal.ca/labs/infographie/papers*. The video shows actual motion capture sequences that were used in our illustrations. It also demonstrates the intuitive tuning of parameters, and real-time display of our system. This can be very useful to get the best camera angle, while the various illustrations are automatically adapted to the camera position. While we focused in this paper on a still image illustrating motion, some sequences of the video show how the system can be used in conjunction with the animated sequence. This can be very useful to bring special attention to particular movements.

6 Conclusion

6.1 Summary

We have presented an interactive system for illustrating various movements extracted from skeletal motion capture data, and represent them in a single image. Our approach analyzes the movements at different levels of detail and applies a number of non-photorealistic motion cues to illustrate the most important movements. We have demonstrated the illustrativeness of motion arrows, noise waves, and stroboscopic motion. Moreover, we have applied our motion cues to improve illustration of movements conveyed through automatically detected multiple poses, as well as automatically detected multiple foot steps. Our perceptually suggestive motion cues can bring immediate understanding of motions in a limited static 2D image.

Our hierarchical organization of the articulated figure applied to the movements allow to isolate "illustrable" parts of the motion capture data at different levels of detail. This method enables a fast and effective motion analysis, as well as motion illustration, and requires no pre-processing. Finally, we introduced a limited set of parameters for our motion. While the default values extracted from the motion analysis are often satisfying, the real-time display in our system allows for interactive fine-tuning of the appearance of the motion cues, as well as the camera position. All updates are displayed in real time, appropriately oriented with respect to the camera.

6.2 Future Work

Our short term goal includes the extension to a number of other motion cues, such as speed lines and motion blur, thus increasing the realm of artistic expressions in our system. We also plan on improving our motion analysis tools for robustness and better recognition of behaviors and patterns in motions. They would improve automatic selection of the pose(s), motion cues and their default values, as well as preferred camera positions. A thorough user evaluation of our system would also help define precise values for motion cues parameters.

Our long term goal is to automatically recognize specific movements from a database, and apply the most appropriate high or low level motion cues for the desired type of illustrations. This could include illustrating different motions in ballet, martial arts, and other high-precision sports or arts. A possible extension would then be to find significant differences between motions from a professional and a learner, thus illustrating the most important differences in order to improve the learner's skills.

Acknowledgements

This work was supported by grants from NSERC. The first author also benefited from a Ubisoft scholarship. Most human skeletal motion capture data used in this project was obtained from mocap.cs.cmu.edu. The database was created with funding from NSF EIA-0196217. The authors thank Eric Bourque for his helpful comments.

References

AGARWALA, A., DONTCHEVA, M., AGRAWALA, M., DRUCKER, S., COLBURN, A., CURLESS, B., SALESIN, D., AND COHEN, M. 2004. Interactive digital photomontage. *ACM Trans. Graph. 23*, 3, 294–302.

AGRAWALA, M., PHAN, D., HEISER, J., HAYMAKER, J., KLINGNER, J., HANRAHAN, P., AND TVERSKY, B. 2003. Designing effective step-by-step assembly instructions. *ACM Trans. Graph. 22*, 3, 828–837.

ASSA, J., CASPI, Y., AND COHEN-OR, D. 2005. Action synopsis: pose selection and illustration. *ACM Trans. Graph. 24*, 3, 667–676.

BARBIC, J., SAFONOVA, A., PAN, J.-Y., FALOUTSOS, C., HODGINS, J. K., AND POLLARD, N. S. 2004. Segmenting motion capture data into distinct behaviors. In *Proc. Graphics Interface*, 185–194.

BRAND, M., AND HERTZMANN, A. 2000. Style machines. In *Proc. SIGGRAPH '00*, 183–192.

BREGLER, C., LOEB, L., CHUANG, E., AND DESHPANDE, H. 2002. Turning to the masters: motion capturing cartoons. In *Proc. SIGGRAPH '02*, 399–407.

BROSTOW, G. J., AND ESSA, I. 2001. Image-based motion blur for stop motion animation. In *Proc. SIGGRAPH '01*, 561–566.

CHENNEY, S., PINGEL, M., IVERSON, R., AND SZYMANSKI, M. 2002. Simulating cartoon style animation. In *Proc. Symposium on Non-photorealistic Animation and Rendering*, 133–138.

COLLOMOSSE, J. P., AND HALL, P. M. 2005. Motion analysis in video: Dolls, dynamic cues and modern art. In *Proc. International Conference on Vision, Video and Graphics*, 109–116.

COLLOMOSSE, J., ROWNTREE, D., AND HALL, P. M. 2005. Rendering cartoon-style motion cues in post-production video. *Graphical Models 67*, 6, 549–564.

COLLOMOSSE, J. P. 2004. *Higher Level Techniques for the Artistic Rendering of Images and Video*. PhD thesis, University of Bath, United Kingdom.

CUTTING, J. E. 2002. Representing motion in a static image: constraints and parallels in art, science, and popular culture. *Perception 31*, 10, 1165–1193.

DONY, R., MATEER, J., AND ROBINSON, J. 2005. Techniques for automated reverse storyboarding. *IEE Journal of Vision, Image & Signal Processing 152*, 4, 425–436.

EISNER, W. 1985. *Comics And Sequential Art*. Poorhouse Press.

GLARDON, P., BOULIC, R., AND THALMANN, D. 2006. Robust on-line adaptive footplant detection and enforcement for locomotion. *Visual Computer 22*, 3, 194–209.

GOLDMAN, D. B., CURLESS, B., SALESIN, D., AND SEITZ, S. M. 2006. Schematic storyboarding for video visualization and editing. *ACM Trans. Graph. 25*, 3, 862–871.

HALLER, M., HANL, C., AND DIEPHUIS, J. 2004. Non-photorealistic rendering techniques for motion in computer games. *ACM Computers in Entertainment 2*, 4, 11–24.

HART, J. 1999. *The Art of the Storyboard: storyboarding for file, TV and animation*. Focal Press.

HEISER, J., PHAN, D., AGRAWALA, M., TVERSKY, B., AND HANRAHAN, P. 2004. Identification and validation of cognitive design principles for automated generation of assembly instructions. In *Proc. Working Conference on Advanced Visual Interfaces*, 311–319.

HSU, S. C., AND LEE, I. H. H. 1994. Drawing and animation using skeletal strokes. In *Proc. SIGGRAPH '94*, 109–118.

HUTCHINSON, A. 1984. *Dance Notation: the process of recording movement on paper*. Dance Horizons.

JOSHI, A., AND RHEINGANS, P. 2005. Illustration-inspired techniques for visualizing time-varying data. In *Proc. IEEE Visualization '05*, 679–686.

KAWAGISHI, Y., HATSUYAMA, K., AND KONDO, K. 2003. Cartoon blur: Non-photorealistic motion blur. In *Proc. Computer Graphics International '03*, 276–281.

KOVAR, L., AND GLEICHER, M. 2004. Automated extraction and parameterization of motions in large data sets. *ACM Trans. Graph. 23*, 3, 559–568.

KOVAR, L., SCHREINER, J., AND GLEICHER, M. 2002. Footskate cleanup for motion capture editing. In *Proc. Symposium on Computer Animation*, 97–104.

LAKE, A., MARSHALL, C., HARRIS, M., AND BLACKSTEIN, M. 2000. Stylized rendering techniques for scalable real-time 3D animation. In *Proc. Symposium on Non-photorealistic Animation and Rendering*, 13–20.

LIU, C., TORRALBA, A., FREEMAN, W. T., DURAND, F., AND ADELSON, E. H. 2005. Motion magnification. *ACM Trans. Graph. 24*, 3, 519–526.

MASUCH, M., SCHLECHTWEG, S., AND SCHULZ, R. 1999. Speedlines: depicting motion in motionless pictures. In *ACM SIGGRAPH '99 Conference abstracts and applications*, ACM Press, New York, NY, USA, 277.

MCCLOUD, S. 1993. *Understanding Comics-The Invisible Art*. Kitchen Sink Press.

MORIYA, T., TAKAHASHI, T., AND TANAKA, A. 2006. Non-photorealistic rendering technique for depicting 3D objects motion. In *ACM SIGGRAPH '06 Research Posters*, 114.

NIENHAUS, M., AND DÖLLNER, J. 2003. Dynamic glyphs – Depicting dynamics in images of 3D scenes. In *Proc. Symposium on Smart Graphics '03*, 102–111.

NIENHAUS, M., AND DÖLLNER, J. 2005. Depicting dynamics using principles of visual art and narrations. *IEEE Computer Graphics and Applications 25*, 3, 40–51.

OBAYASHI, S., KONDO, K., KONMA, T., AND WAMOTO, K.-I. 2005. Non-photorealistic motion blur for 3D animation. In *ACM SIGGRAPH '05 Conference Abstracts*.

RASKAR, R., ZIEGLER, R., AND WILLWACHER, T. 2002. Cartoon dioramas in motion. In *Proc. Symposium on Non-photorealistic Animation and Rendering*, 7–12.

THORNE, M., BURKE, D., AND VAN DE PANNE, M. 2004. Motion doodles: an interface for sketching character motion. *ACM Trans. Graph. 23*, 3, 424–431.

TUFTE, E. R. 1997. *Visual Explanations: Images and Quantities, Evidence and Narrative*. Graphics Press.

WANG, J., DRUCKER, S. M., AGRAWALA, M., AND COHEN, M. F. 2006. The cartoon animation filter. *ACM Trans. Graph. 25*, 3, 1169–1173.

WHEELER, A., THIEL, J., AND HART, G. 2006. Cartoon motion blur for 3D animation. In *ACM SIGGRAPH '06 Sketches*, 95.

WILLIAMS, R. 2001. *The Animator's Survival Kit*. Faber and Faber Limited.