Saliency Diagrams

a tool for analyzing animation through the relative importance of keyposes

Nicolas Nghiem
Visual Media Lab, KAIST
Ecole polytechnique

Richard Roberts
CMIC, Victoria University of Wellington

J.P. Lewis
Victoria University of Wellington

Junyong Noh
Visual Media Lab, KAIST

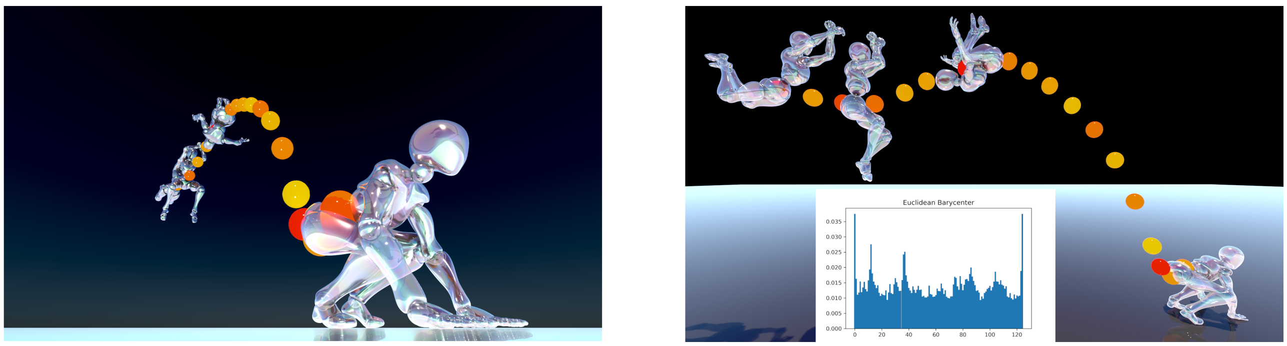

Figure 1: Our method computes a Saliency diagram that encodes a measure of saliency for each frame in the motion. The spikes in the diagram tend to correspond to extremes of the animation and thus can be chosen as the most important frames to consider when analyzing the motion.

ABSTRACT

Keyframes are a core notion used by animators to understand and describe the motion. In this paper, we take inspiration from keyframe animation to compute a feature that we call the “Saliency diagram” of the animation. To create our saliency diagrams, we visualize how often each frame becomes a keyframe when using an existing selection technique. Animators can use the resulting Saliency diagram to analyze the motion.

CCS CONCEPTS

• **Computing methodologies → Motion capture**; • **Theory of computation** → *Computational geometry*.

KEYWORDS

Keyframe animation, motion capture, geometry, saliency

ACM Reference Format:
Nicolas Nghiem, Richard Roberts, J.P. Lewis, and Junyong Noh. 2019. Saliency Diagrams: a tool for analyzing animation through the relative importance of keyposes. In *SIGGRAPH Asia 2019 Technical Briefs (SA ’19 Technical Briefs), November 17–20, 2019, Brisbane, QLD, Australia.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3355088.3365155

SA ’19 Technical Briefs, November 17–20, 2019, Brisbane, QLD, Australia

ACM ISBN 978-1-4503-6945-9/19/11…$15.00
https://doi.org/10.1145/3355088.3365155

1 INTRODUCTION

While planning a shot and animating is the core task for an animator, understanding why and how things are moving is a crucial part of the creative process. “Why” refers to the behaviour of people, so that the animator will craft solid "acting" choices to create believable characters. “How” requires the animator to understand the physics of the motion, leading some animators to study the anatomy of the animals they animate to help understand the principles of their locomotion.

Keyframes are the most important poses of an animation and are enough to understand the story. As introduced by [Williams 2001], *pose-to-pose* is a common workflow among animators. Following this convention involves first crafting those keyposes, then creating breakdowns to have a better control of the overall dynamics, and finally letting the computer – or the inbetweener in traditional animation – interpolate to obtain a fluid motion. An extra step of polishing is then applied to add fine scale details to the motion. In terms of acting, the keyposes will correspond to the main actions and expressions of the character, while in body mechanics, the keyposes tell things about the movement: extremes, weight shifts, contacts, overshoots etc. An important practical application of identifying keyframes is that it allows captured motion to be edited using traditional keyframe animation.

However, previous methods cannot solve the problem of finding the "most important" frames of the animation. For a low number of keyframes, they tend to take points that are not geometrically interesting (Figures 2 and 6). For a high number, they do not provide a criterion to sort those keyframes by order of importance. Since pose-to-pose animation is a common workflow, identifying keyposes of a motion, whether presented as a video or an animation, is a vital skill for an animator.

Providing a visualization of keyposes presents multiple advantages: it can provide a way for beginners to analyze the motion and be used as a tool for simplifying motion editing workflows, furthermore, it is a natural way of encoding the information of the motion.

Previous work has recognized the importance of such visualization. For example, [Assa et al. 2008] introduce a low-dimensional embedding technique to find a few poses that best summarize an action when rendered and then composed together into an image. As another example, [Yasuda et al. 2008] apply the same low-dimensional embedding technique (paired with a different representation of the motion) to assemble a linear timeline of rendered poses that can be used to browse or compare animations. While these visualization techniques are already useful, they only communicate the presence (or absence) of keyposes. In particular, they miss the opportunity to provide additional information about the keyposes: all keyposes are treated as equal, where no one keypose can be said to be more important than any another.

In this paper, we present a new way to visualize a motion. To build the visualization, we apply "Salient Poses" [Roberts et al. 2019]: an optimal keyframe selection algorithm that can be used to query different numbers of keyframes in an animation. In particular, we apply the Salient Poses algorithm for all possible numbers of keyframes and collate the information into a histogram that encodes an overall level of importance for each frame. From the histogram, we can read keyposes as those frames with high importance.

As the primary contribution of this work, we present our method to build Saliency diagrams. Importantly, our histogram not only encodes which frames correspond to keyposes but also the relative importance of those keyposes, which is distinct from previous work. The analysis of these keyposes can be particularly interesting for various purposes such as training, stylization or database storing. As a secondary contribution, we present an efficient mechanism to capture rotation information when expressing the animation as a high-dimensional curve (the input for the keyframe selection algorithm).

2 KEYFRAME SELECTION WITH SALIENT POSES

Proposed by Roberts et al. [Roberts et al. 2019], Salient Poses is a keyframe selection algorithm designed to find sets of keyframes ideal for editing. Animators can use it to perform keyframe reduction to easily recover an editable animation from motion capture.

As a summary of the algorithm, it takes as input an animation and a value function. Precisely, the animation is represented as a high-dimensional curve[1] and the value function computes a score describing how distant an interpolation of the given set of keyframes is from the original animation.[2] Given these inputs, the algorithm successively applies dynamic programming to calculate all optimal sets of keyframes (one set of three keyframes, one of four, and so on). Each set is optimal in the sense that its keyframes best summarize the motion as defined by the value function.

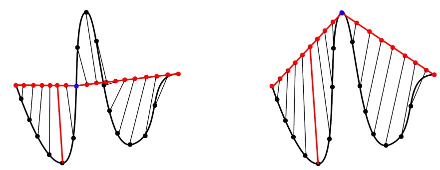

Figure 2: Possible selections of three keyframes. Salient Poses will choose the left configuration over the right one since the maximum distance of its reconstruction (in red) to the original animation is smaller. The dots represent the discretization of the motion as frames: black dots correspond to the original animation, red to the linear interpolation using the keyframe selection, and blue to the selected keyframe.

While keyframe reduction is useful, the goal of our work is to help animators examine the sets of keyframes to better understand the motion. Unfortunately, the sets of keyframes provided by Salient Poses suffer from two key flaws from when applied for this purpose:

2.1 Poor performance for small values of k

From an animator's perspective, we expect keyframes to give priority to the extreme poses of the motion. However, when k is small, Salient Poses often fails to select poses related to distinctive extrema. As illustrated by Figure 2, in cases where extremes are distant in value but not time, choosing one extreme will typically produce an interpolation very distant from another extreme. Consequently the value function discourages the choice and, instead, the algorithm will choose poses unrelated to extrema. Furthermore, the choice of keyframes will often change dramatically between neighbouring sets (i.e. the set of $k-1$ often features keyframes that occur at significantly different times than those in k, and so on). Importantly, in these cases, an animator cannot gain insight when examining the poses provided by the algorithm.

2.2 No rotation information

The value function defined in [Roberts et al. 2019] uses only the position of the joints and does not take into account their rotation information. Imagine an animation of a character standing still and nodding their head: as no positional information has changed, Salient Poses will not be able to find the keyposes related to the nodding (the head occurs at the end-effector of the joint chain). Furthermore, being able to distinguish keyframes with respect to translation and to rotation may provide important insights that can help animator's further analyze the motion.

[1] The animation is comprised of a set of joints that each have three translation-based coordinates. A configuration of these joints can be described as a single point in $\mathbb{R}^{3n_j+3}$ where n_j is the number of joints. The animation can then be expressed by plotting the configuration of each frame into this space, forming a discrete high-dimensional curve.

[2] In [Roberts et al. 2019], the value function calculates this score by measuring the maximum Euclidean distance between the original animation and a linear interpolation of the keyframes.

3 METHOD

To address these problems we will:

- introduce a proxy object that expresses rotation information as translation
- use the output of the Salient Poses algorithm to compute the "Saliency diagram" of the motion.

3.1 Adding rotation information

To take into account the rotation of the joint J, we introduce an additional object that we call L (aka locator) that we constrain to the joint J, i.e L is fixed in the local space of J. Applying Salient Poses to L would take into account the rotation but there will still remain the influence of the translation. To avoid that, we can consider the position of L_J in $(J, \vec{X}, \vec{Y}, \vec{Z})$ rather than $(O, \vec{X}, \vec{Y}, \vec{Z})$. Indeed, we have:

$$L_{(J,\vec{X},\vec{Y},\vec{Z})} = \vec{JL} = \vec{OL} - \vec{OJ}$$

where $\vec{JL}(t)$ depends only on the joint rotation:

$$\vec{JL}(t) = R_J(t)\,\vec{JL}_0$$

In practice, such a mechanism is straightforward to implement. For example, we create an additional point, constrain it to the desired joint, calculate its animation when using that constraint, then subtract the position of the joint at each frame. As expected, the locator moves on a sphere of center O and radius $||\vec{JL}_0||$. We can now apply Salient Poses to L if we want to study the rotation only, or to (J, L) if we want to take both translation and rotation into account. Note that when applying Salient Poses to the locator, it is more accurate to take into account the fact that it moves on a sphere, by using spherical linear interpolation rather than linear interpolation and calculating the distance along geodesics rather than the basic Euclidean distance.

3.2 Saliency diagram

As in previous work [Lim and Thalmann 2001; Roberts et al. 2019], we define a keyframe selection as an element of $\mathcal{S}$, the set of all possible keyframe choices which is the set of all subsets of $\{1, ..., N_f\}$ with N_f the number of frames of the animation. We call $\mathcal{S}_k$ the subset of the elements of cardinality exactly k of $\mathcal{S}$ i.e the set of keyframe selection of exactly k keyframes.

We can apply Salient Poses to our desired animation $\mathcal{A}$ and obtain all the $S^*(k)$ for $k \in \{1, ..., N_f\}$ – we only need to run Salient Poses once since it computes all the $S^*(k)$ in one pass. We can compute the Saliency diagram of $\mathcal{A}$ by averaging all those keyframe selections.

To do so, rather than seeing $S^*(k)$ as an element of $\mathbb{N}^k$ we consider it as a weight distribution. Let us say that the animation $\mathcal{A}$ has a total weight of 1, choosing a k-keyframe selection is equivalent to distribute this weight on those k frames. (cf. Figure 3)

Once represented as a weight distribution, we average $S^*(k)$ using a summation (see Figure 4). Finally, to improve the clearness of the spikes, we apply gaussian convolution to the diagram (recall that selecting a frame f does not exactly mean putting the weight on f but rather putting it somewhere on $[f - \delta f, f + \delta f]$).

Notably, the averaging we computed (c.f. Figure 5) is not a keyframe selection but rather a distribution. The histograms feature both major and minor spikes: a spike indicates that a keyframes was picked more than usual and is more truly a "salient" poses than other less-often picked keyframes. Using the diagram, selecting the most important keyframes becomes the task of selecting the major spikes in the histogram.

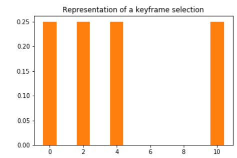

Figure 3: Keyframe selection {0,2,4,10} as a distribution.

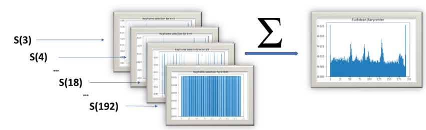

Figure 4: Computing the Saliency diagram.

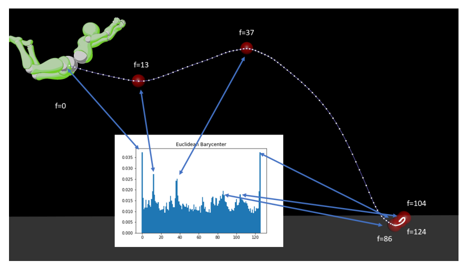

Figure 5: An animation and its corresponding Saliency diagram. The spikes in the histogram corresponds to the extremes of the motion trail.

4 RESULTS

First, we apply Salient Poses with a very low number of keyframes on multiple animations. An example is shown in Figure 6. In this example, the keyposes chosen by Salient Poses are not well suited for visualizing the motion: a blocking using only those poses cannot convey the motion, and we would not understand that it is a jump. The main spikes obtained by our approach does not correspond to the frames selected in $S^*(2)$, in particular, we choose more meaningful extreme poses. This figure illustrates how we outperform Salient poses for low values of k.

4.1 Auxiliary rotation signal

To verify that our locator-design is useful, we applied our algorithm to the locator of the root joint of animations where there are clear offsets between the rotation and the translation. Jump

Figure 6: Keyposes selected by Salient Poses (top) and our method (bottom) for $k = 4$. The position of the hips is highlighted: our algorithm chooses the extreme positions of the root while Salient Poses chooses non-extreme positions.

movements are one such example as the rotation generally occurs after a translation in order to balance the center of gravity. In such a case, we expect to see that our Saliency diagram finds different keyposes for the locator animation and the translation animation. Figure 7 shows the result for one of those jump motions which is in accordance with our expectations.

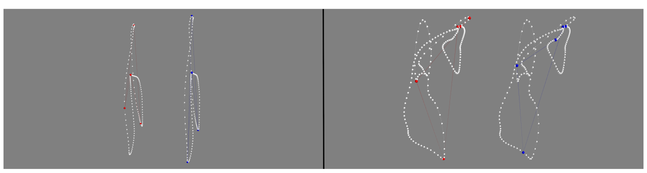

Figure 7: The motion trail of the rotation locator (right) and the motion trail of the position of the hips (left) with the selected frames colored. Red dots are the keyframes selected by our algorithm applied to the rotation locator, blue dots are the keyframes applied to the position of the hips.

4.2 Discussion

Our algorithm is complementary to Salient Poses: it is designed to work particularly well for low values of k where Salient Poses fails to find geometrically distinctive poses. Importantly, our algorithm can also be used to reduce computation time. In the original paper, [Roberts et al. 2019] suggest that to limit computation time for larger animations, the animation should be segmented into pieces. However, they do not provide a method to choose such segments. Using our Saliency diagram, we can choose the most important keyposes as candidates for segmentation: we can then redesign a version of Salient Poses that continuously updates the current Saliency diagram – which is not costly to compute – and subdivides the animation when an important keyframe is detected via a threshold. If we make the assumption that the expected distance between two important keyframes is bounded, our algorithm is linear whereas Salient Poses is polynomial.

Ultimately, our Saliency diagrams expose the statistics of Salient Poses and, consequently, provide a new way to analyze the motion. For example, one can create multiple Saliency diagrams, using different error functions, and examine the spikes to identify which set of keyframes are most prevalent under different criteria.

5 DISCUSSION

5.1 Limitations

The most obvious limitation of our work is the lack of metric for evaluation. We do not have a way, other than using our own eyes, to state quantitatively whether a pose is suitable as keypose or not. This problem remains ill-posed.

In the particular case where our locator is positioned on an axis of rotation, our approach will not be able to capture the rotation information. This degenerate case would occur rarely in practice as we pick our locator randomly on a sphere centered in the corresponding joint.

When we express our keyframe selection as a weight distribution, we choose to distribute our weight uniformly over the chosen keyframes. However, all frames are not equivalent and using information from Salient Poses (especially how taking this keyframe benefits the value function $\mathcal{V}$) to compute those weights would give better results.

Finally, our method effectively computes an Euclidean Barycenter from the weight distributions. However, Euclidean distance is not well suited to compare distributions and is particularly bad when considering distributions with non-intersecting support (as is this case when working simultaneously with different animations).

5.2 Future work

Our immediate future work will seek a metric to quantify whether a given frame is suitable as a keyframe under different contexts (such as visualization, editing, and more). Other future work will examine how our Saliency diagram can be used as a way to encode additional information about the motion, much like color histograms or saliency maps do in Computer Vision. We also aim to employ our Saliency diagrams to study well-known animation concepts such as *overlaps* or the *leading part*, and to compare different animations to find correspondences between them, such as those related to morphing.

ACKNOWLEDGMENTS

This research is supported by Ministry of Culture, Sports and Tourism (MCST) and Korea Creative Content Agency (KOCCA) in the Culture Technology (CT) Research & Development Program.

REFERENCES

Jackie Assa, Daniel Cohen-Or, I-Cheng Yeh, and Tong-Yee Lee. 2008. Motion Overview of Human Actions. *Transactions on Graphics* 27, 5 (Dec. 2008), 115:1–115:10. https://doi.org/10.1145/1409060.1409068

Ik Soo Lim and D. Thalmann. 2001. Key-posture Extraction Out of Human Motion Data. In *Proceedings of the 23rd Annual International Conference of the IEEE*, Vol. 2. 1167–1169. https://doi.org/10.1109/IEMBS.2001.1020399

Richard Roberts, J. P. Lewis, Ken Anjyo, Jaewoo Seo, and Yeongho Seol. 2019. Optimal and interactive keyframe selection for motion capture. *Computational Visual Media* 5, 2 (01 Jun 2019), 171–191.

Richard Williams. 2001. *The Animator's Survival Kit.* Faber.

Hiroshi Yasuda, Ryota Kaihara, Suguru Saito, and Masayuki Nakajima. 2008. Motion Belts: Visualization of Human Motion Data on a Timeline. *IEICE Transactions on Information and Systems* E91.D, 4, 1159–1167. https://doi.org/10.1093/ietisy/e91-d.4.1159