A Motion Style Toolbox

Klaus Förger and Tapio Takala

Department of Computer Science, Aalto University, Finland
{klaus.forger,tapio.takala}@aalto.fi

Abstract. We present a Matlab toolbox for synthesis and visualization of human motion style. The aim is to support development of expressive virtual characters by providing implementations of several style related motion synthesis methods thus allowing side-by-side comparisons. The implemented methods are based on recorded (captured or synthetic) motions, and include linear motion interpolation and extrapolation, style transfer, rotation swapping per body part and per quaternion channel, frequency band scaling and swapping, and Principal/Independent Component Analysis (PCA/ICA) based synthesis and component swapping.

Keywords: computer animation, human motion, motion style, motion synthesis, toolbox

1 Introduction

Expressiveness of virtual characters can be vital for many practical applications. Bodily movements are a useful modality for creating expressive behaviors. One approach is to utilize symbolic gestures such as waving a fist or scratching ones head. However, interpreting them may depend on the viewer's cultural background. For this reason, modulating motion style, meaning the manner how a gesture or an action is performed, is an attractive way to create expressive variations of motions. We present a toolbox of Matlab functions for motion capture based synthesis and visualization of human motion style (see Fig. 1 for examples), and a collection of code examples. The visualization technique is a novel way to draw attention to motion style.

The toolbox includes implementations of several methods related to creating new style variations from captured motions. Time warping is also included as a necessary synchronization tool when style variations are based on combining or comparing different motions. 14 code examples of animation techniques are provided portraying cases with best and worst performance. For each case, a short video is included that demonstrates the resulting motions. The package also contains acted examples of hand waving and locomotion performed in varying styles.

The toolbox assumes a hierarchical skeleton structure as in the BioVision Header (BVH) format. The joint rotations are represented by quaternions, and translation of the root with xyz coordinates. In our examples, the motions are recorded with the rate of 100 frames per second. We assume that input motions contain the same action and only differ in style. The considered methods enable high level control of style with relatively low computational demands. We exclude methods that require manual key-frame editing or physics simulations, or only concatenate existing motions.

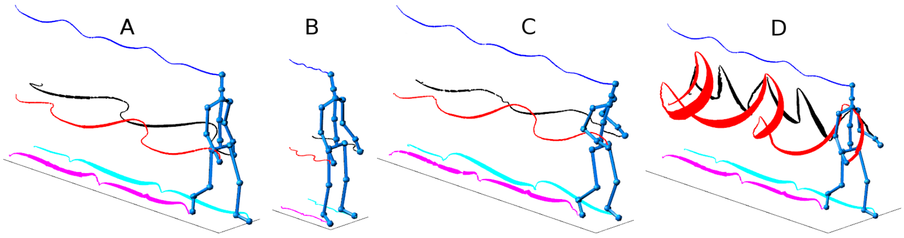

Fig. 1. Walks produced by acting (A), frequency band scaling (B), frequency band swapping (C), and partial joint signal swapping (D). The trajectories indicate velocities.

The toolbox can be used for creating expressive variations of motions, and the included examples may be useful as learning material. When developing new animation methods, the implementations can be used as a baselines to allow comparisons. The toolbox is not readily usable as part of a real-time animation pipeline, but it could be used in finding out what kind of a pipeline allows maximal expressivity. The code is available under an open source BSD license at: `http://iki.fi/klaus.forger/motion_style_toolbox/`

2 Related work

Our goal is to allow comparisons between style related motion synthesis methods, and to map and explore types of styles the methods can produce. This is not an easy task based on existing publications. For example, a presented system may combine many techniques such as swapping body parts between motions, interpolation and signal decompositions [6], or apply post-processing to remove produced artifacts [8]. Therefore, we attempt to outline low-level operations which can be separately implemented and evaluated. Selection of methods into the toolbox has been guided by earlier reviews on motion synthesis [4, 7].

3 Implemented methods

If a method takes multiple input motions, time warping is necessary for making corresponding events happen at the same time [7]. We have implemented two versions (see video/code ex_01), one applying a constant warp and another with segmentation.

Linear interpolation is the standard approach for creating ranges of styles by motion blending [7]. We have implemented two versions, one for pairwise interpolation (ex_02) and another for N-way interpolation (ex_03). Both take as parameters the weights of the input motions which must sum to 100%. Our pairwise implementation is intended for non-repetitive actions and applies the slerp algorithm [9]. For N-way interpolation we have applied normalized linear interpolation (nlerp) instead, as slerp does not generalize to this case. The implementation is optimized for locomotion, and uses segmented time warping.

Extrapolation and style transfer rely on modeling style as numerical differences between motions. The extrapolation extends linear interpolation by allowing the weights of the motions to be under 0% and over 100% while still requiring them to sum to 100%. Style transfer in turn means taking a difference between signals of two motions, scaling it to adjust the amount of style, and adding the result to a third motion. We have implemented pairwise extrapolation (ex_04), style transfer between three motions (ex_05), and N-way extrapolation (ex_06). N-way extrapolation is the most general as it allows combining multiple pairwise interpolations, extrapolations and style transfers.

In interpolation and extrapolation, the weight of each input motion is an adjustable parameter. If a set of motions contains redundant versions of styles, dimensionality reduction can be applied to join the redundant parameters. In this context, we have implemented motion synthesis based on Principal Component Analysis (PCA) (ex_07) and Independent Component Analysis (ICA) (ex_08). These methods differ in the way the input motions are decomposed, but are otherwise identical. The PCA synthesis uses Matlab function *princomp* and the ICA applies decomposition using the *FastICA* library[5]. A further comparison between PCA synthesis and N-way extrapolation shows that our implementations can produce exactly the same sets of motions.

Different frequencies have been suggested to be associated with different motion styles [1]. Thus, scaling part of the frequencies of a motion should affect the perceived styles. We have implemented decomposition into frequency bands as originally described by Bruderlin and Williams [1]. Frequency band scaling (ex_09) can be applied to a single motion and it can produce variations on the size of the motions and styles from stiff to energetic (Fig. 1 B), for example. The frequency bands can also be swapped between motions (ex_10) that allows for example transferring the overall posture from one motion to another (Fig. 1 C).

Swapping signals of joints between motions (ex_11) allows creation of many new style variations with minimal artifacts (Fig. 1 D). However, the joints must be carefully selected as otherwise the swap can break the whole motion. It is also possible to swap individual quaternion channels (ex_12), but this can add shakiness and sudden accelerations to the motions as breaking quaternions apart does not guarantee them to remain of unit length.

PCA and ICA can be used to reduce the number of channels that represent rotations. This can for example combine together joints that rotate similarly. New style variations can be created by swapping part of the motion signals from one motion to another in the reduced space. Our implementation is similar to that described by Shapiro et al. [8], with the exception that we use quaternions instead of joint coordinates. After the dimensionality reduction with PCA (ex_13) and ICA (ex_14), the new signal channels can be swapped between motions. This can create new style variations, but may also result in similar artifacts as in the case of swapping individual quaternion channels.

4 Discussion

The toolbox enables exploration and comparison of a multitude of motion editing methods. For example, it reveals that even simple methods such as swapping rotations between motions can create interesting style variations. Also, there are differences between the usability of the methods as for example it may not be easy to describe in practical terms what styles are affected by editing PCA components. More rigorous comparison of the methods would require perceptual studies and quantitative evaluations of the produced motions [3]. Also, the usefulness of the synthesis methods can differ depending on the final application where they would be used in. For example, a virtual character with a cartoonish representation may benefit from exaggerated motions, while naturalness of the motions may be more important for a realistic character. The toolbox could be especially helpful in expanding expressivity of motion collections containing variedly performed versions of actions [2].

Acknowledgements

This work has been supported by the Kone Foundation through the project Social eMotions, and the Hecse graduate school. We also thank people behind the open source Matlab libraries/functions FastICA, GPMat, Quaternion Toolbox, ScreenCapture, Slerp, and SpinCalc.

References

1. Bruderlin, A., Williams, L.: Motion signal processing. In: Proc. of the 22nd annual conference on Computer graphics and interactive techniques. pp. 97–104. ACM (1995)
2. Carreno-Medrano, P., Gibet, S., Larboulette, C., Marteau, P.F.: Corpus creation and perceptual evaluation of expressive theatrical gestures. In: Bickmore, T., Marsella, S., Sidner, C. (eds.) IVA 2014, LNCS (LNAI), vol. 8637, pp. 109–119. Springer International Publishing (2014)
3. Förger, K., Takala, T.: Animating with style: defining expressive semantics of motion. The Visual Computer pp. 1–13, DOI: 10.1007/s00371–015–1064–4 (2015)
4. Geng, W., Yu, G.: Reuse of motion capture data in animation: a review. In: Computational Science and Its Applications, ICCSA 2003, pp. 620–629. Springer (2003)
5. Hyvärinen, A.: Fast and robust fixed-point algorithms for independent component analysis. Neural Networks, IEEE Transactions on 10(3), 626–634 (1999)
6. Kim, Y., Neff, M.: Component-based locomotion composition. In: Proc. of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation. pp. 165–173. Eurographics Association (2012)
7. Pejsa, T., Pandzic, I.S.: State of the art in example-based motion synthesis for virtual characters in interactive applications. In: Computer Graphics Forum. vol. 29, pp. 202–226. Blackwell Publishing (2010)
8. Shapiro, A., Cao, Y., Faloutsos, P.: Style components. In: Proc. of Graphics Interface 2006. pp. 33–39. Canadian Information Processing Society (2006)
9. Shoemake, K.: Animating rotation with quaternion curves. SIGGRAPH Comput. Graph. 19(3), 245–254 (Jul 1985)