Moving Crowds

A Linear Animation System for Crowd Simulation

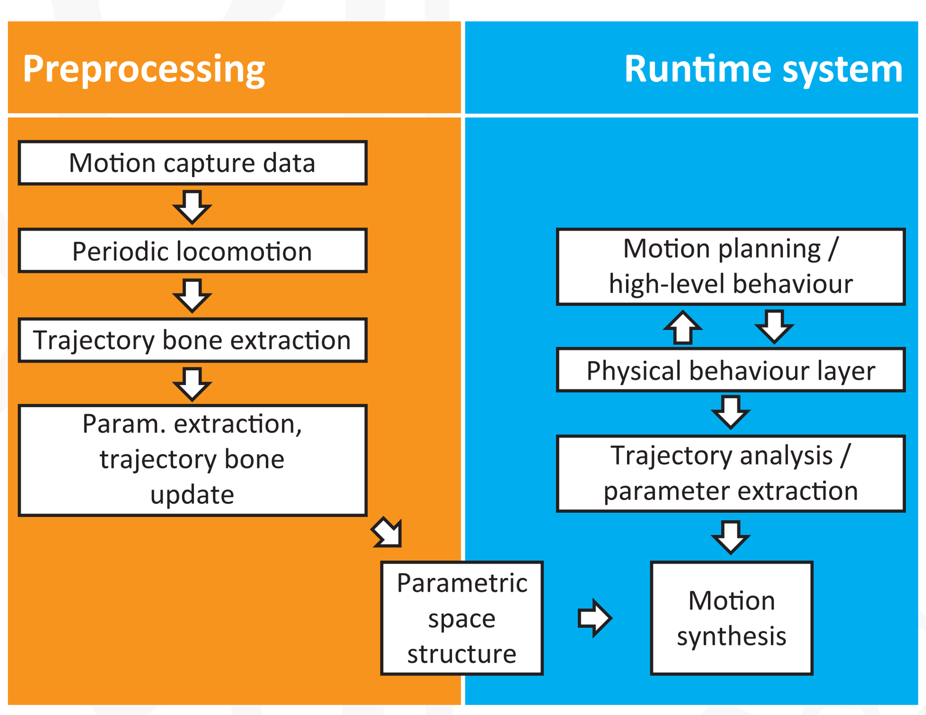
Martin Pražák, Ladislav Kavan, Rachel McDonnell, Simon Dobbyn, Carol O'Sullivan
Graphics, Vision and Visualisation Group, Trinity College Dublin, Ireland

Abstract

The animation of hundreds or even thousands of simultaneously displayed individuals is challenging because of the need for both motion variety and efficient runtime processing. We present a **middle level-of-detail** animation system optimised for handling **large crowds** which takes motion-capture data as input and automatically processes it to create a parametric model of human locomotion. The model is then used in a runtime system, driven by a **linearised motion blending** technique, which synthesises motions based on information from a motion-planning module. Compared to other animation methods, our technique provides significantly better runtime performance without compromising the visual quality of the result.

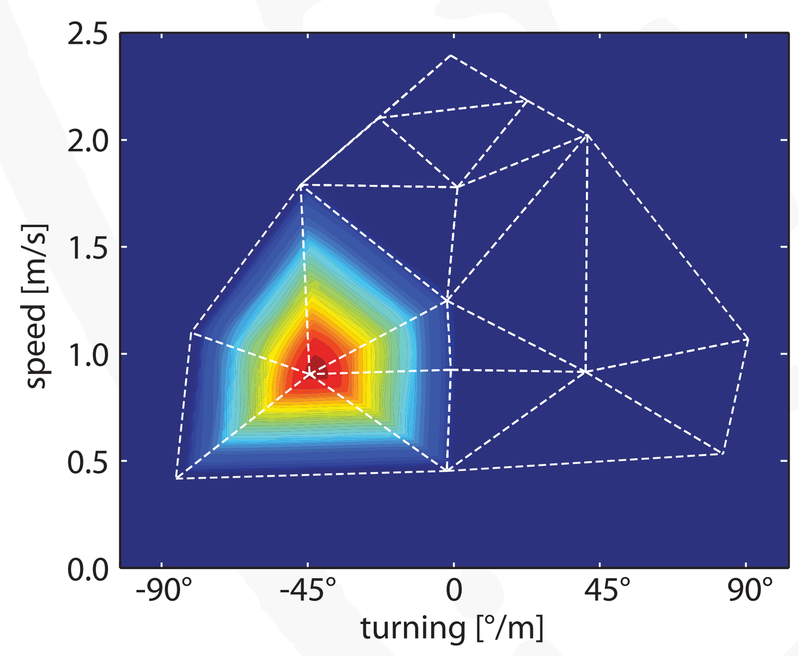
1 A Linear Animation System for Crowds

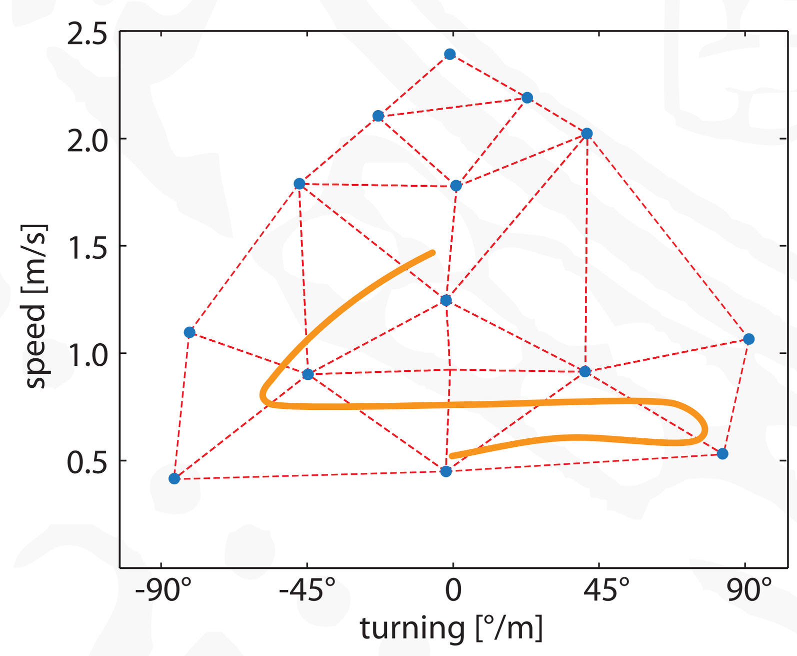
Our system consists of two main parts – the preprocessing phase, where the underlying data structure is created, and the runtime system, which uses the preprocessed data to generate character motion for the crowd system. Both parts are built on and linked by our parametric space structure.



1.1 Parametric Space

The core concept of our model is the **parametric space** – a space partitioning structure allowing locomotion to be synthesised from input motion clips by motion blending and time-warping. The structure itself is built from a point cloud, with every point representing the parameters of a motion clip. The space is then partitioned using Delaunay triangulation. This allows for the generation of a **clip with arbitrary parameters** by **linear blending (and timewarping)** three clips from our original database.





1.2 Preprocessing

The preprocessing of input motion capture data consists of the following steps:

- Converting the original motion clip into a periodic, loopable clip
- Extracting the trajectory bone, which represents the projection of motion trajectory onto the ground plane
- Parameter extraction and update of the trajectory bone
- Synchronisation of all database clips

1.3 Runtime System

The runtime system consists of a preprocessed database of animation clips, a parametric space structure and a module to alter trajectories (provided by a higher-level behaviour system) for use in our parametric space.

The process uses a **damped PD controller** which closely follows the non-smooth input path while projecting the trajectory into parametric space and enforcing constraints (minimal and maximal speed, maximal turning, and maximal values of their respective derivatives).

2 Linearised Animation

The standard animation generation and blending model uses a local-space representation of transformations (rotation matrices, Euler angles, quaterions) to describe localised orientations and to blend between multiple animation clips. This representation has to be then converted into skinning matrices which describe transformations of model segments in world space.

In order to lower computational complexity, we use **skinning matrices for animation blending** which in the case of our parametric space and preprocessing do not introduce any noticeable artefacts.

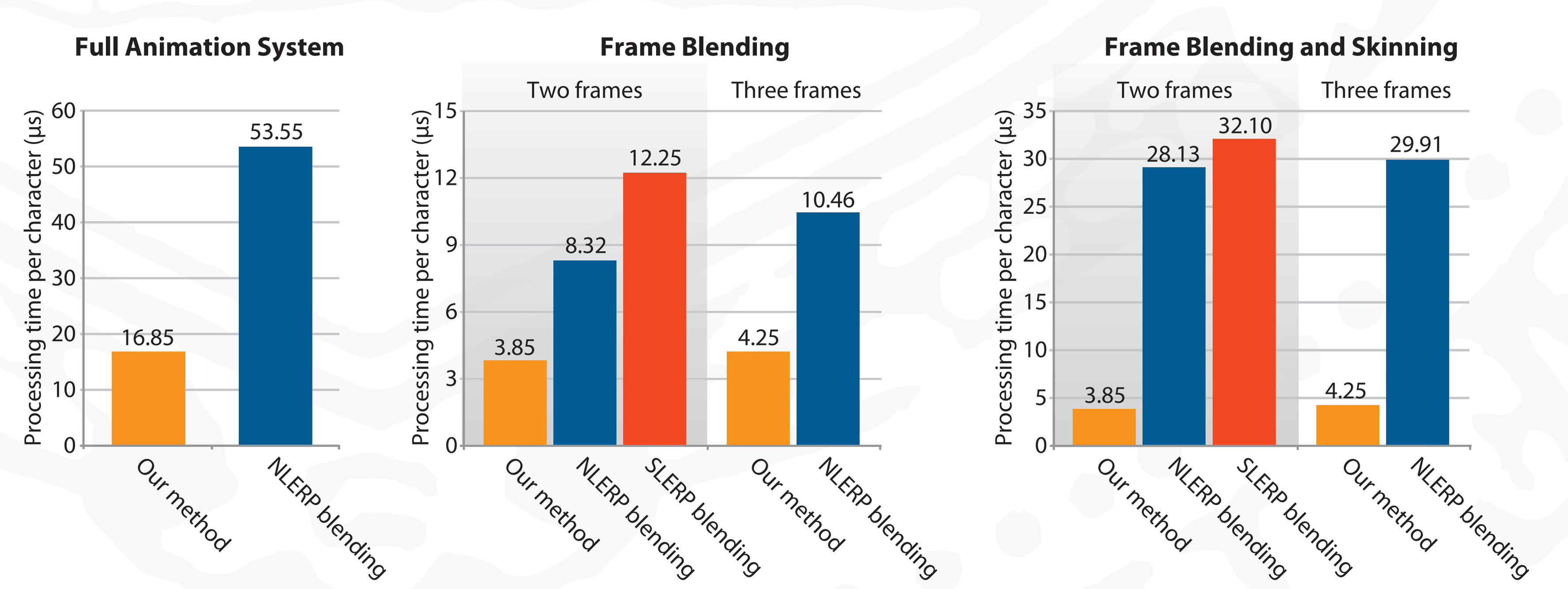
This description is mathematically equivalent to the linear blending of mesh animations, allowing us to **use pre-simulated cloth animation** on our characters without introducing any artefacts or computational overhead.

3 Results

A **theoretical evaluation** of two aspects was performed – the computational cost of playing an animation clip (e.g., converting the internal representation into skinning matrices) and the cost of interpolation between n animation clips. Based on this analysis, we conclude that our system performs significantly better than standard algorithms for typical numbers of simultaneously interpolated frames.

Algebra / operation	+, -	*, /	fn
Conversion to C_i matrices (i.e., playing the animation)			
Quaternions	129	102	0
Euler angles	123	106	0
Matrix interpolation	0	0	
Interpolation between n transformations + conversion			
Quaternion NLERP	4n + 128	4n + 107	1
Euler angles	3n + 120	3n + 106	6
Matrix interpolation	12n - 12	24n - 24	0

Practical tests were performed in an isolated environment, performing only the computations required for blending source animations. Our system, again, proved to provide significantly better performance than standard techniques.



References

Lewis, J. P., Cordner, M., and Fong, N. 2000. Pose space deformation: a unified approach to shape interpolation and skeleton-driven deformation. *International Conference on Computer Graphics and Interactive Techniques.*

Park, S., Shin, H., and Shin, S. 2002. On-line locomotion generation based on motion blending. *Proceedings of the 2002 ACM SIGGRAPH/Eurographics Symposium on Computer Animation.*

science foundation ireland
fondúireacht eolaíochta éireann