

Immersive Interfaces for Building Parameterized Motion Databases

Yazhou Huang, Carlo Camporesi, and Marcelo Kallmann

University of California, Merced

Abstract. We present new interface tools for the interactive motion modeling and construction of parameterized motion databases for virtual agents. The tools provide different forms of visual exploration of database coverage, providing an intuitive way to model and refine motions by direct demonstration.

In interactive virtual training and assistance applications, virtual agents are driven by realistic motion synthesis techniques with parameterized variations in respect to a given scenario. Building such database can be complex, time-consuming and in many cases must be done by experts. Common solutions for the motion modeling process rely on hand-crafted motions [2, 12, 13], gestures synthesized with algorithmic procedures [5, 6], motion blending based on captured data [7, 8, 10, 11], etc. We improve the modeling phase of our existing interactive motion modeling framework by direct demonstration [1]. The framework is based on a full-scale 3D display facility, and has been extended to include intuitive interfaces to build, visualize, evaluate and refine a motion database in respect to the spatial coverage inside a simulated workspace, guiding the on-line programming of scenario-specific examples.

Interface Description Our system targets situations where the user is able to model clusters of action or gesture motions by direct demonstration. This is done via either a wearable *gesture vest* [3] or Vicon tracking system for capturing upper-body motions together with data gloves for capturing hand motions. A WiiMote serves as an interface to create new cluster types, start/stop capture, playback, trim, annotate stroke points, save or delete motions. Clusters defined by examples is an important concept for specifying parameterized action or gesture types, and motions within a same cluster are blended to consistently represent variations of the same type. The user can also examine the database spatial coverage inside the virtual workspace with two visualization methods (described below) for guidance on improving the database coverage as needed. The coverage refers to how well each motion cluster is parametrized using *inverse blending* [4] from the discrete examples in order to satisfy specified spatial constraints. Fig 1 top left outlines framework pipeline, top right shows one motion being reviewed for editing.

Database Spatial Coverage Visualization The ability to enforce constraints for new motions greatly depends on the existing variations among the example motions. In general, a small number of carefully selected example motions can provide good coverages for the regions of interest (ROIs) in the workspace. We propose two specific visualization methods rendering a palette of colors [9, 14, 15] inside the workspace to intuitively guide the user during the process of adding new motions to refine the database for improved coverage: Workspace Volume Visualization (WV) and Local

Coverage Visualization (LV). See Fig 1. WV conducts a coarse uniform sampling of the workspace and presents the overall spatial coverage with colored cubes for the entire workspace without the need to define an overly fine subdivision of the constraint space. Each cube represents a reaching target (spatial constraint), and a motion synthesized towards each cube is measured by reaching precision (error e^*) using a constraint evaluation function, and the value $e^*/e_{max} (\in [0, 1])$ is mapped onto a hue color space then assigned to each cube. For a reasonably sized database WV takes a few seconds to generate, then the user can immediately spot areas with low coverage by the color of the cubes (red or orange), and add additional motion towards these areas. LV renders a transparent colored mesh geometry covering a small ROI, delimiting the coverage evaluation within its volume. It focuses on the local coverage visualization taking only milliseconds to be computed, and it is suitable for fine tuning coverage of smaller volumes when only small local regions are of interest. LV uses the same color mapping but applied to mesh vertices. LV follows the movement of the user's hand, its size and shape can be iteratively changed for either fast sweeping over large ROIs (a table surface) or for carefully checking small ROIs (buttons, etc). LV is also able to utilize motions dynamically added to the database without any pre-computation lag. Please refer [4] for details on motion synthesis and error evaluation with spatial constraints.

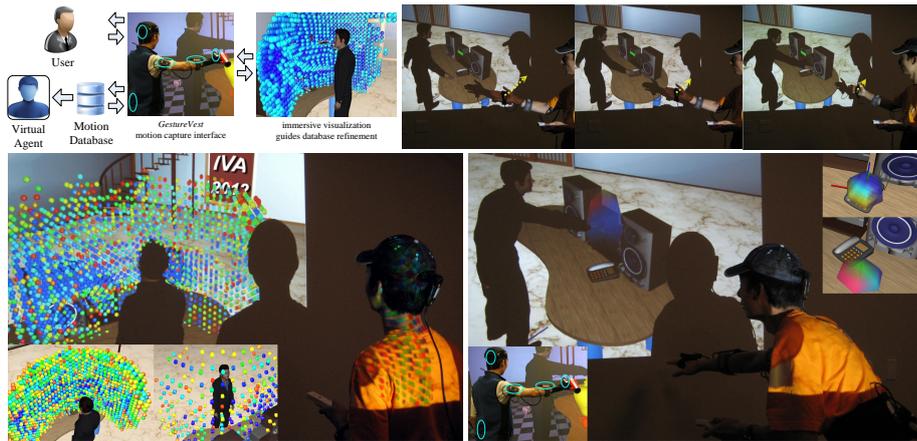


Fig. 1. Top-left: framework pipeline. Top-right: user moves his hand to scroll through a motion being edited. Bottom-left: Workspace Volume Visualization mode gives an overview of database coverage, density and error threshold can be adjusted for clear viewing. Bottom-right: Local Coverage Visualization mode, ideal for checking small ROIs like dials and buttons.

Conclusions and Acknowledgments Our proposed tools greatly improve the process of interactive motion modeling and the overall approach constitutes a powerful approach for programming virtual agents. This work was partially funded by NSF award IIS-0915665.

References

1. C. Camporesi, Y. Huang, and M. Kallmann. Interactive motion modeling and parameterization by direct demonstration. In *Proceedings of the 10th International Conference on Intelligent Virtual Agents (IVA)*, 2010.
2. P. Gebhard, M. Kipp, M. Klesen, and T. Rist. What are they going to talk about? towards life-like characters that reflect on interactions with users. In *Proc. of the 1st International Conference on Technologies for Interactive Digital Storytelling and Entertainment (TIDSE'03)*, 2003.
3. Y. Huang and M. Kallmann. Interactive demonstration of pointing gestures for virtual trainers. In *Proceedings of 13th International Conference on Human-Computer Interaction*, San Diego, CA, 2009.
4. Y. Huang and M. Kallmann. Interactive motion modeling and parameterization by direct demonstration. In *Proceedings of the 3rd International Conference on Motion in Games (MIG)*, 2010.
5. M. Kallmann. Analytical inverse kinematics with body posture control. *Computer Animation and Virtual Worlds*, 19(2):79–91, 2008.
6. S. Kopp and I. Wachsmuth. Synthesizing multimodal utterances for conversational agents: Research articles. *Computer Animation and Virtual Worlds*, 15(1):39–52, 2004.
7. L. Kovar and M. Gleicher. Automated extraction and parameterization of motions in large data sets. *ACM Transaction on Graphics (Proceedings of SIGGRAPH)*, 23(3):559–568, 2004.
8. T. Mukai and S. Kuriyama. Geostatistical motion interpolation. In *ACM SIGGRAPH*, pages 1062–1070, New York, NY, USA, 2005. ACM.
9. I. Rodriguez, M. Peinado, R. Boulic, and D. Meziat. Bringing the human arm reachable space to a virtual environment for its analysis. In *IEEE International Conference on Multimedia and Expo*, 2003.
10. C. Rose, B. Bodenheimer, and M. F. Cohen. Verbs and adverbs: Multidimensional motion interpolation. *IEEE Computer Graphics and Applications*, 18:32–40, 1998.
11. C. F. RoseIII, P.-P. J. Sloan, and M. F. Cohen. Artist-directed inverse-kinematics using radial basis function interpolation. *Computer Graphics Forum (Proceedings of Eurographics)*, 20(3):239–250, September 2001.
12. M. Stone, D. DeCarlo, I. Oh, C. Rodriguez, A. Stere, A. Lees, and C. Bregler. Speaking with hands: creating animated conversational characters from recordings of human performance. *ACM Transactions on Graphics*, 23(3):506–513, 2004.
13. M. Thiebaut, A. Marshall, S. Marsella, and M. Kallmann. Smartbody: Behavior realization for embodied conversational agents. In *Seventh International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, 2008.
14. J. Yang, T. Sinokrot, K. Abdel-Malek, S. Beck, and K. Nebel. Workspace zone differentiation and visualization for virtual humans. *Ergonomics*, 51(3):395–413, 2008.
15. F. Zacharias, I. S. Howard, T. Hulin, and G. Hirzinger. Workspace comparisons of setup configurations for human-robot interaction. In *IROS*, pages 3117–3122, 2010.