Data-driven On-line Character Control: Philosophy and Promise

Sung Yong Shin

Computer Science Division
Korea Advanced Institute of Science and Technology
373-1 Guseong-dong, Yuseong-gu, Daejeon 305-701, Republic of Korea

e-mail: syshin@jupiter.kaist.ac.kr

ABSTRACT

Real-time character control is an important issue to increase interactivity in computer graphics. Due to ever-increasing demand from applications such as computer games and interactive movies, real-time character control has drawn attention from both industry and academia for the last decade. Under nation support of Korea, we conducted a research project on real-time computer animation for five years from 1999 to 2004. The objective of this project was to develop on-line character control technology for digital contents production. The project covers three areas of character animation: motion capture and reuse, facial expression cloning, and image-based navigation. In this talk, we discuss the underlying philosophy of this project and introduce some of representative results.

BACKGROUND

Recently, rich results in both theory and practice have been presented in computer animation. From synthesis point of view, those results can be classified into two categories : procedure-driven animation and data-driven animation. In the former category, an animation is internally driven by a procedure. Typical examples are key-frame animation, physically-based animation, and behavioral animation. In the latter, an animation is externally driven by data captured from the real world. There are two kinds of approaches in this category, that is, off-line animation and on-line animation. The main applications of off-line animation include movie films, advertisements, and broadcasting. Here, what is important is how to synthesize a realistic animation. In on-line animation, however, what is important is how to achieve a real-time performance to support applications such as computer games and virtual worlds. In this talk, our concern is data-driven on-line animation.

Under a national support, my group had conducted a research project on data-driven animation for five years from 1999 to 2004. The objective of this project was to develop on-line, real-time character animation technology for digital contents production. As illustrated in Figure 1, the project covered three different areas in character animation, that is, motion capture and reuse [1, 2, 3, 8, 9, 10, 12, 13, 14, 17, 18, 19, 20, 21, 25, 27, 29], on-line facial animation [15, 16, 22, 23, 24, 28], and image-based navigation [4, 5, 6, 7, 11] to generate motions, facial expressions and background scenes, respectively. At first glance, these three research areas look a little too diverse for a small research group in a university. However, unveiling those areas reveals two common ingredients, that is, an example-based paradigm and computational geometry.

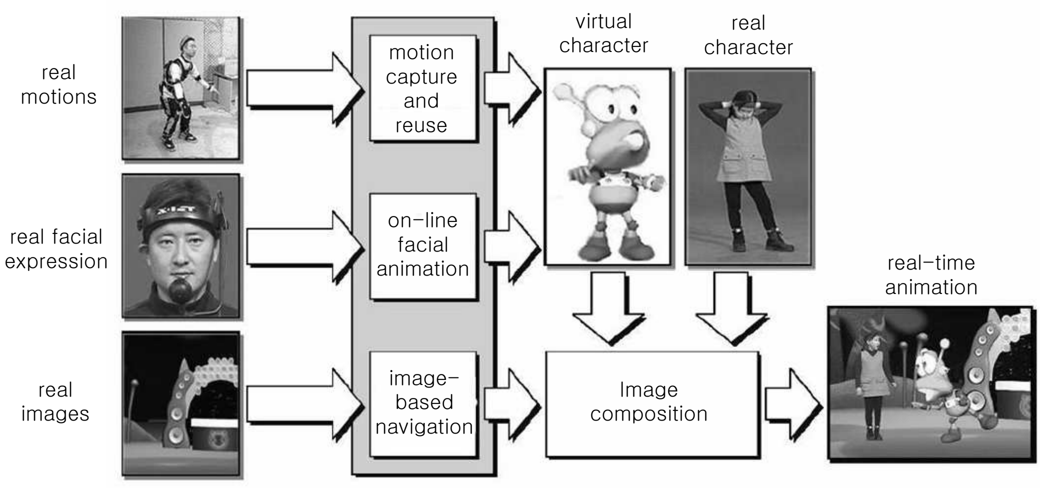

Figure 1. The schematic diagram of The NRL project

UNDERLYING PHILOSOPHY

The example-based paradigm facilitates ”watch the nature” approaches to building, animating, and rendering things both animate and inanimate in virtual worlds. These approaches are driven by data captured from the real world, that is, the masterpiece of God according to my interpretation. Computer graphics people dream of creating their own virtual worlds as God created the universe. However, it is absolutely impossible to simulate the almighty power of God simply with human-made equations. Believe or not, God created the whole universe moving at his will in a week. God can render the whole earth, if not the whole universe, instantaneously.

Realizing the limitations as human-being, our strategy is to mimic what is done by God rather than to simulate how things done by him. We first take samples from the real world and parameterize them. Then, we construct a solution space S(x) by interpolating the samples. By the solution space, we mean the space that hopefully gives an instance of God’s masterpiece when evaluated at a specific parameter vector. Finally, we solve a constrained optimization problem to compute the parameter vector, at which the solution space S(x) is evaluated. The objective function f(·) can be chosen depending on what to optimize, for example, forces, torques, or distortions depending on applications. The constraints are specified to further characterize a desirable parameter vector.

$$\min_{x} f(s(x))$$
$$\text{subject to } C(s(x)) = 0$$

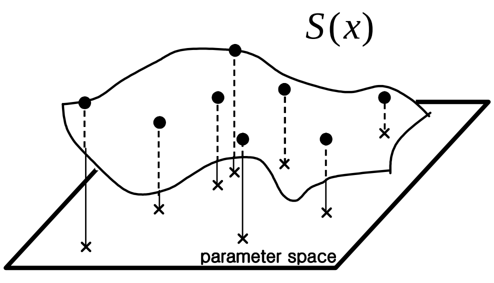

Figure 2. Solution space S(·)

Computational geometry is the other ingredient of our researches. According to Shamos [26], computational geometry is intended to recast classical geometry within the framework of algorithm analysis and designs. Our objective is to design algorithms of geometric nature based on a solid theoretical background, which will eventually push computer graphics toward science.

REPRESENTATIVE RESULTS

From now, we introduce some representative results. First, we begin with video-guided animation. The objective of this work is to reconstruct a plausible 3D motion sequence from a monocular video. In theory, it is hard, if not impossible, to reconstruct 3D motions only with a single video stream. Our solution is to use a motion library that contains a repertoire of densely-sampled 3D motions. A reference motion is first selected from the library, the projection of which is similar to the input video and then is deformed, guided by the input video. This can be formulated as a constrained optimization problem. We find a motion, which is as similar to the reference motion as possible, while satisfying 2D constraints specified by the input video.

Second, we show rhythmic motion synthesis, that is, synthesizing novel rhythmic motion sequences from unlabeled example motions while preserving their rhythmic patterns. In pre-processing, out approach first analyzes the unlabeled example motions to extract their rhythmic patterns called motional beats and then labels the motion data automatically to construct a motion transition graph, by exploiting the extracted motional beats. A node of the graph represents a set of blendable basic motions, that is, a set of motion segments of an identical structure, and an edge represents a motion transition. At runtime, this graph is traversed from node to node in an on-line manner, while blending the motions at each node to synthesize a novel motion, guided by an input sound signal. The optimization problem becomes much more complicated because of the temporal constraints imposed by the rhythmic patterns. To my knowledge, our approach is not only the first work for automatic synchronization of motions with music but also the first successful attempt to automatic 3D motion labeling, although restricted to rhythmic motions.

Next, we describe facial animation, in particular, facial expression cloning, that is, transferring facial expressions from a source face model to a target face model. We adopt a blend-shape scheme, based on scattered data interpolation, which is similar to the motion blending scheme. We blend facial expressions rather than example motions. To synthesize more diverse expressions with fewer key-models, we segment a face model automatically into regions, apply the blend-shape scheme to the segmented regions separately, and combine them.

Finally, we discuss image-based navigation, in particular, tour into the picture to show how to model background scenes solely based on images. We use images as examples in place of motions and facial expressions. Exploiting a vanishing line rather than vanishing points, our scene model is much simpler and yet more general than that of the original TIP by Horry and others. This scheme can be generalized easily for panoramic images and videos.

CONCLUSIONS

In summary, we have discussed on-line character animation. In the future, we would like to enhance our example-based paradigm by incorporating ideas from other approaches: for example, from physically-based approaches for reduction of sample size, from biomechanical data-driven approaches for plausibility and efficiency, and from perception-based approaches for visual effectiveness. Our focus has mainly been on a single character control although we have some work done for multiple characters. In the future, our efforts will be concentrated on multiple character control including two-character motions and crowd simulation.

REFERENCES

1. M.Brand, “Coupled Hidden Markov Models for Modeling interacting Processes,” Technical Report, 405, MIT Media Lab, MIT, 1996.
2. M.G.Choi, J.Lee, and S.Y.Shin, “Planning Biped Locomotion using Motion Capture Data and Probabilistic Roadmaps,” *ACM Transaction on Graphics*, vol.22, no.2, pp. 182-203, 2003.
3. M.Gleicher, “Retargeting motion to new characters,” *Computer Graphics (Proc.SIGGRAPH ’98)*, vol.32, pp. 33-42, July. 2003.
4. K.Horry, K.Anjyo, and K.Arai, “Tour into the picture : Using the Spidery Mesh Interface to Make Animation from a Single Image,” *Computer Graphics (Proc.SIGGRAPH 97).*, vol. 31, pp. 225-232, Aug. 1997.
5. H.W.Kang, S.H.Pyo, K.Anjyo, and S.Y.Shin, “Tour Into the Picture using a Vanishing Line and Its Extension to Panoramic Images,” *Computer Graphics Forum.*, vol. 20, no. 3, pp. 132-141, 2001. (Also presented at *EuroGraphics* 2001)
6. H.W.Kang, and S.Y.Shin, “Tour Into Video : Image-based Navigation System for Video Sequences of Dynamic Scenes,” *In Proc. ACM VRST Symposium.*, pp. 73-80, 2002.
7. H.W.Kang, and S.Y.Shin, “Creating Walk-through Images from a Video Sequence of a Dynamic Scene,” *Presence-Teleoperators and Environments.*, 2005. (in press)
8. H.K.Kim, L.Guibas, and S.Y.Shin, “Efficient collision detection among moving spheres with unknown trajectories,” *Algorithmica.*, 2005 (in press)
9. D.Kim, H.K.Kim, and S.Y.Shin, “An Event-Driven Approach to Crowd Simulation with Example Motions,” Technical Report, CS-TR-2002-186, Computer Science Department, KAIST, 2002.
10. T.H.Kim, S.I.Park, and S.Y.Shin, “Rhythmic-Motion Synthesis based on Motion-Beat Analysis,” *ACM Transactions on Graphics.*, vol. 22, no. 3, pp. 392-401, 2003. (Also presented at *SIGGRAPH* 2003)
11. S.Y.Lee, K.Y.Chwa, S.Y.Shin, and G.Wolberg, “Image Matamorphosis Using Snakes and Deformations,” *Computer Graphics (Proc.SIGGRAPH ’95).*, vol. 29, July, 1995.
12. J.Lee, and S.Y.Shin, “A hierarchical approach to interactive motion editing for human-like figures,” *Compuer Graphics (Proc.SIGGRAPH ’99)*, vol.33, pp. 395-408, Aug. 1999.
13. J.Lee, and S.Y.Shin, “A coordinate-invariant approach to multi-resolution motion analysis,” *Graphical Model.*, vol. 63, no. 3, pp. 1-19, 2001.
14. J.Lee, and S.Y.Shin, “General construction of time-domain filters for orientation data,” *IEEE TVCG.*, vol. 8, no. 2, pp. 119-128, 2002.
15. J.Noh and U.Neumann, “Expression Cloning,” *Computer Graphics (Proc.SIGGRAPH 01).*, vol. 30, pp. 277-288, July. 2001.
16. B.C.Park, H.J.Chung, J.H.Ma, and S.Y.Shin, “A Feature-Based Approach to Facial Expression Cloning,” Technical Report, CS-TR-2004-197, Computer Science Division, KAIST, 2004.

17. M.J.Park, M.G.Choi, and S.Y.Shin, “Human Motion Reconstruction from Inter-Frame Feature Correspondences of a Single Video Stream Using a Motion Library,” *In Proc. ACM SIGGRAPH Symposium n Computer Animation.*, pp. 113-120, 2002.

18. M.J.Park, and S.Y.Shin, “Example-based motion cloning,” *Computer Animation & Virtual Worlds.*, vol. 15, pp. 245-257, 2004. (Also presented at *CASA* 2004.)

19. S.I.Park, T.Kwon, H.S.Shin, and S.Y.Shin, “Analysis and synthesis of interactive two-character motions,” Technical Report, CS-TR-2004-194, Computer Science Division, KAIST, 2004.

20. S.I.Park, H.J.Shin, T.H.Kim, and S.Y.Shin, “On-line motion blending for real-time locomotion generation,” *Computer Animation & Virtual Worlds.*, vol. 15, pp. 125-138, 2004. (Also presented at *CASA* 2004.)

21. S.I.Park, H.J.Shin, and S.Y.Shin, “On-line Locomotion Generation Based on Motion Blending,” *In Proc. ACM SIGGRAPH Symposium on Computer Animation.*, pp. 105-111, 2002.

22. H.W.Pyun, W.S.Chae, Y.J.Kim, H.W.Kang, and S.Y.Shin, “An Example-based Approach to Text-driven Speech Animation with Emotional Expressions,” Technical Report, CS-TR-2004-200, Computer Science Division, KAIST, 2004.

23. H.W.Pyun, Y.J.Kim, W.S.Chae, H.W.Kang, and S.Y.Shin, “An Example-based Approach for Facial Expression Cloning,” *In Proc, ACM SIGGRAPH Symposium on Computer Animation.*, pp. 161-171, 2003.

24. H.W.Pyun, H.J.Shin, and S.Y.Shin, “On Extracting Wire Curves from Multiple Face Models for Facial Animation,” *Computer Graphics.*, vol. 28, no. 5, pp. 757-765, 2004.

25. C.Rose, M.Cohen, and B.Bodenheimer, “Verbs and Adverbs : Multidimensional motion interpolation,” *IEEE CG&A.*, vol. 18, no. 5, pp. 32-40, 1998.

26. M Shamos, “Computational Geometry,” Ph.D. thesis, Department of Computer Science, Yale University, 1978.

27. H.J.Shin, J.Lee, M.Gleicher, and S.Y.Shin, “Computer Puppetry : An importance-Based Approach,” *ACM Transactions On Graphics.*, vol. 20, no. 2, pp. 67-94, 2001.

28. K.Singh and E.Fium, “Wires : A Geometric Deformation Technique,” *Computer Graphics (Proc.SIGGRAPH 98).*, vol. 27, pp. 405-414, July. 1998.

29. P.Sloan, C.Rose, M.Cohen, “Shape by Example,” *In Proc. ACM Symposium on Interactive 3D Graphics.*, pp. 135-144, 2001.