CASA 2009

International Conference on Computer Animation and Social Agents

SHORT PAPER AND POSTER PROCEEDINGS OF THE TWENTY-SECOND ANNUAL CONFERENCE ON COMPUTER ANIMATION AND SOCIAL AGENTS

Amsterdam, June 17-19, 2009

Anton Nijholt, Arjan Egges, Herwin van Welbergen and Hendri Hondorp (eds.)

CIP GEGEVENS KONINKLIJKE BIBLIOTHEEK, DEN HAAG

Nijholt, A., Egges, A., van Welbergen, H., Hondorp, G.H.W.

Conference on Computer Animation and Social Agents (CASA) 2009
Short Paper en Poster Proceedings of the twenty-second
Annual Conference on Computer Animation and Social Agents
A. Nijholt, A. Egges, H. van Welbergen, G.H.W. Hondorp (eds.)
Amsterdam, Universiteit Twente, Faculteit Elektrotechniek, Wiskunde en Informatica

ISSN 0929–0672
CTIT Workshop Proceedings Series WP09-02

trefwoorden: *Animation Techniques*: Motion Control, Motion Capture and Retargeting, Path Planning,
Physics based Animation, Artificial Life, Deformation, Facial Animation;
Social Agents: Social Agents and Avatars, Emotion and Personality, Virtual Humans, Autonomous Actors,
AI based Animation, Social and Conversational Agents, Gesture Generation, Crowd Simulation;
Other Related Topics: Animation Compression and Transmission, Semantics and Ontologies for
Virtual Humans/Environments, Animation Analysis and Structuring, Anthropometric Virtual Human Models,
Acquisition and Reconstruction of Animation Data, Semantic Representation of Motion and Animation,
Medical Simulation, Cultural Heritage, Interaction for Virtual Humans, Augmented Reality and
Virtual Reality, Computer Games and Online Virtual Worlds.

Book orders:
Ms. C. Bijron
University of Twente
Faculty of Electrical Engineering, Mathematics and Computer Science
P.O. Box 217
NL 7500 AE Enschede
tel: +31 53 4893740
fax: +31 53 4893503
Email: bijron@cs.utwente.nl

Druk- en bindwerk: Ipskamp Drukkers, Enschede.

Preface

These are the proceedings containing the short and poster papers of CASA 2009, the twenty second international conference on Computer Animation and Social Agents. CASA 2009 was organized in Amsterdam, the Netherlands from the 17th to the 19th of June 2009. CASA is organized under the auspices of the Computer Graphics Society (CGS) and is the premier academic conference in the field of computer animation and behavior simulation of social agents. CASA was founded in 1988. Over the last years the conference has been organized in Philadelphia (1998, 2000), Seoul (2001, 2008), Geneva (2002, 2004, 2006), New Jersey (2003), Hong Kong (2005), and Hasselt (2007).

In 2009 the conference was held in Amsterdam, the Netherlands. The organization was done by the Human Media Interaction (HMI) research group of the University of Twente, the Netherlands. The CASA 2009 edition received 123 submissions. Of these submissions 35 full papers have been chosen to appear in revised form in a special issue of the Wiley InterScience online journal Computer Animation and Virtual Worlds. From the remaining submissions 16 short, 10 poster and 3 GATE-session-abstracts papers have been selected for these proceedings. They also contain the list of accepted full papers and the abstracts of the two invited talks by Volker Blanz of the University of Siegen in Germany and Franck Multon of the University of Rennes in France.

We thank all the authors for having submitted their work to this conference and the members of the international program committee and the additional external reviewers for their time and efforts invested in the reviewing process. We would like to thank the editors of Wiley for their support to publishing the selected full papers in a very short time. A particular word of thanks goes to the HMI support team, that is, to Lynn Packwood, Charlotte Byron and Alice Vissers for their support in organizing the conference and taking care of financial and organizational matters. CASA 2009 has been sponsored by the innovation agency IOP-MMI of SenterNovem (Dutch Ministry of Economic Affairs), the Netherlands Organisation for Scientific Research (NWO) and the GATE (Game research for Training and Entertainment) project.

Anton Nijholt, Arjan Egges, Herwin van Welbergen and Hendri Hondorp — Amsterdam, June 2009

Previous CASA Conferences

2003	New Jersey, USA	2006	Geneva, Switserland
2004	Geneva, Switserland	2007	Hasselt, Belgium
2005	Hong Kong, China	2008	Seoul, Korea

Committees CASA 2009

Conference Chair
Anton Nijholt

Program Co-Chairs
Scott King — Nadia Magnenat-Thalmann — Mark Overmars

Local Co-Chairs
Arjan Egges — Hendri Hondorp — Herwin van Welbergen

International Program Committee

Neeharika Adabala	Norman Badler	William Baxter	Massimo Bergamasco
Ronan Boulic	Marc Cavazza	Hwangue Cho	Min-Hyung Choi
Sabine Conquillart	Zhigang Deng	Fabian Di Fiore	Stephane Donikian
Arjan Egges	Petros Faloutsos	James Hahn	Dirk Heylen
Donald House	Eric Jansen	Chris Joslin	Gerard J. Kim
Hyungseok Kim	Arie E. Kaufman	Prem Kalra	Scott King
Taku Komura	Rynson W.H. Lau	John P. Lewis	Seungyong Lee
WonSook Lee	Nadia Magnenat-Thalmann	Carlos Martinho	Franck Multon
Dinesh Manocha	Louis Philipe Morency	Soaia Musse	Ahmad Nasri
Luciana Porcher Nedel	Mark Overmars	Igor Pandzic	Rick Parent
John Patterson	Catherine Pelachaud	Qunsheng Peng	Paolo Petta
Julien Pettre	Rui Prada	Stephane Redon	Skip Rizzo
Zsofi Ruttkay	Isaac Rudomin	Hyewon Seo	Sung Yong Shin
Matthias Teschner	Yiying Tong	Hanqui Sun	Daniel Thalmann
Frank Van Reeth	Luis Velho	Frédéric Vexo	Nin Wang
Herwin van Welbergen	Philip Willis	Enhua Wu	Ying-Qing Xu
Yizhou Yu	Jian Zhang	Job Zwiers	

External Reviewers

Ben van Basten	Alessandro Bicho	Wim Fikkert	Luiz Gonzaga Júnior
Arno Kamphuis	Ioannis Karamouzas	Fernando Marson	Andreea Niculescu
Ronald Poppe	Dennis Reidsma	Bart van Straalen	Ivo Swartjes
Mariët Theune			

Sponsors

[1]

[2]

[3]

[4]

[1] http://www.nwo.nl
[2] http://www.senternovem.nl/
[3] http://gate.gameresearch.nl/
[4] http://hmi.ewi.utwente.nl

Contents

Invited Talks

List of Accepted Full Papers 7

Short Papers

Posters

GATE Session Papers

Invited Talks

CASA 2009

Morphable Models: A Return Trip beyond the Limitations of Linearity

Volker Blanz
Universität Siegen
Hölderlinstr. 3
57068 Siegen, Germany
blanz@informatik.uni-siegen.de

Abstract

Morphable Models represent real-world data, such as 3D scans of human faces, as vectors in a high-dimensional linear space, and synthesize approximated intermediate instances by linear combinations (morphs). This involves statistical and geometrical assumptions that may or may not be appropriate for empirical data. We present a new morphing paradigm for simulated eye movements, and a non-linear model of aging faces, to show how these phenomena can be captured in a generalized Morphable Model framework.

Using Real-time Virtual Humans for Analyzing Interactions in Sports

Franck Multon
M2S, University Rennes2 – Bunraku INRIA
Av. Charles Tillon 35044 Rennes Cedex - France
Franck.Multon@uhb.fr

In many sports, and mainly in team-based games, both one to one interactions between players and the implementation of general game strategy have been identified as key issues. Studying these kinds of problems is highly complex as an individual player's motion depends on many inter-related parameters (e.g. the dynamics of other players and the movement of the ball). In real game scenarios it is impossible to isolate and systematically vary only one of these parameters at a time to scientifically study its influence on an individual player's behaviour. Scientific protocols currently used to study these interactions are generally far removed from real situations which makes it difficult to draw pertinent conclusions about player behaviour that could inform coaching practice.

VR and simulation present a promising means of overcoming such limitations. From a psychologist's perspective a computer-generated space ensures reproducibility between trials and precise control of the dynamics of a simulated event, something that is impossible in real-life. By carefully controlling the information presented in a visual simulation (e.g. speed of player movement, speed or trajectory of the ball) one can see how the perceptual information that is being controlled by the experimenter affects subsequent choices of action (otherwise known as the perception/action loop). For this type of application in sports to work the simulated actions of the virtual humans must contain a high degree of realism. To be more realistic the virtual humans need to be able to take into account in real time several kinds of constraints such as: kinematics (obeying biomechanical laws), dynamics (satisfying the laws of Newton) and physiological (applying appropriate muscular forces). If fast interactions are required, this problem must be solved with fast iterative but inaccurate methods.

This talk will address two complementary topics. Firstly, we will describe a real-time virtual human animation engine that is able to replay motion capture data while taking kinematic and dynamic constraints into account. This engine [3, 4] addresses the following issues: motion retargeting, fast kinematic and dynamics constraints solving, synchronization and blending.

Secondly, we will describe two main applications in sports [5]: handball and rugby. In handball, we have studied the perception-action coupling of a goalkeeper who has to anticipate the trajectory of a ball thrown by an opponent [1, 2]. In that case, the motion of the opponent is simulated on a virtual player which enables us to exactly know the information that is available for the goalkeeper. For this work, we have validated that the motor behaviour of the goalkeeper in a real and simulated situation are similar. We then studied the relevance of some visual cues on the goalkeeper's performance and behaviour.

Secondly, we applied this kind of approach for analyzing the perception-action coupling of rugby players who have to intercept an opponent who perform some deceptive motions. A preliminary biomechanical analysis enabled us to identify the kinematic information that is relevant to anticipate the final direction of the opponent when he is performing deceptive motions. We then evaluate how subjects with various levels of expertise in rugby react to simulated opponents (were they able to use this kinematic information for predicting the correct final direction of the opponent?).

We will finally conclude the talk by giving some perspectives about how using virtual humans and virtual environments in analyzing human-to-human interactions.

Acknowledgements

Parts of the works presented in this talk have been carried-out in collaboration with the psychology school of Queen's University Belfast and IPAB from Edinburgh University.

References

[1] B. Bideau, R. Kulpa, S. Ménardais, F. Multon, P. Delamarche, B. Arnaldi (2003), Real handball keeper vs. virtual handball player: a case study, *Presence*, 12(4):412-421, August 2003

[2] B. Bideau, F. Multon, R. Kulpa, L. Fradet, B. Arnaldi, P. Delamarche (2004) Virtual reality, a new tool to investigate anticipation skills: application to the goalkeeper and handball thrower duel. *Neuroscience letters*, 372(1-2) :119-122.

[3] F. Multon, R. Kulpa, B. Bideau (2008) MKM: a global framework for animating humans in virtual reality applications. *Presence*, 17(1): 17-28

[4] F. Multon, R. Kulpa, L. Hoyet, T. Komura (2009) Interactive animation of virtual humans from motion capture data. *Computer Animation and Virtual Worlds* 20:1-9

[5] B. Bideau, R. Kulpa, N.Vignais, S. Brault, F. Multon, C. Craig (2009) Virtual reality, a serious game for understanding behavior and training players in sport. *IEEE CG&A* (to appear).

List of Accepted Full Papers*

CASA 2009

*The full papers appear in a Special Issue of Computer Animationand Virtual Worlds, Wiley InterScience, Volume 20, 2009

Accepted Full Papers CASA 2009

- *Pressure Corrected SPH for Fluid Animation*
  Kai Bao, Hui Zhang, Lili Zheng and Enhua Wu
- *N-way morphing for 2D animation*
  William Baxter, Pascal Barla and Ken Anjyo
- *Interactive Chroma Keying for Mixed Reality*
  Nicholas Beato, Yunjun Zhang, Mark Colbert, Kazumasa Yamazawa and Charles Hughes
- *Advected River Textures*
  Tim Burrell, Dirk Arnold and Stephen Brooks
- *Perceptual 3D Pose Distance Estimation by Boosting Relational Geometric Features*
  Cheng Chen, Yueting Zhuang and Jun Xiao
- *'Give me a Hug': the Effects of Touch and Autonomy on people's Responses to Embodied Social Agents*
  Henriette Cramer, Nicander Kemper, Alia Amin and Vanessa Evers
- *Time-critical Collision Handling for Deformable Modeling*
  Marc Gissler, Ruediger Schmedding and Matthias Teschner
- *Simulating Attentional Behaviors for Crowds*
  Helena Grillon and Daniel Thalmann
- *Pseudo-dynamics Model of a Cantilever Beam for Animating Flexible Leaves and Branches in Wind Field*
  Shaojun Hu, Tadahiro Fujimoto and Norishige Chiba
- *Real-time Dynamics for Geometric Textures in Shell*
  Jin Huang, Hanqiu Sun, Kun Zhou and Hunjun Bao
- *Chemical Kinetics-Assisted, Path-Based Smoke Simulation*
  Insung Ihm and Yoojin Jang
- *Fast Data-Driven Skin Deformation*
  Mustafa Kasap, Parag Chaudhuri and Nadia Magnenat-Thalmann
- *Symmetric Deformation of 3D Face Scans using Facial Features and Curvatures*
  Jeong-Sik Kim and Soo-Mi Choi
- *Perceptually Motivated Automatic Dance Motion Generation for Music*
  Jae Woo Kim, Hesham Fouad, John L. Sibert and James Hahn
- *Patches: Character Skining with Local Deformation Layer*
  Jieun Lee, Myung-Soo Kim and Seung-Hyun Yoon
- *Performance-driven Motion Choreographing with Accelerometers*
  Xiubo Liang, Qilei Li, Xiang Zhang, Shun Zhang and Weidong Geng
- *Competitive Motion Synthesis Based on Hybrid Control*
  Zhang Liang
- *Development of a Computational Cognitive Architecture for Intelligent*
  Pak-San Liew, Ching Ling Chin and Zhiyong Huang
- *TFAN: A Low Complexity 3D Mesh Compression Algorithm*
  Khaled Mamou, Titus Zaharia and Françoise Pretaux
- *Impulse-based Rigid Body Interaction in SPH*
  Seungtaik Oh, Younghee Kim and Byung-Seok Roh

- *Deformation and Fracturing Using Adaptive Shape*
  Makoto Ohta, Yoshihiro Kanamori and Tomoyuki Nishita
- *Automatic Rigging for Animation Characters with 3D Silhouette*
  Junjun Pan, Xiaosong Yang, Xin Xie, Philip Willis and Jian Zhang
- *Furstyling on Angle-Split Shell Textures*
  Bin Sheng, Hanqiu Sun, Gang Yang and Enhua Wu
- *Angular Momentum Guilded Motion Concatenation*
  Hubert Shum, Taku Komura and Pranjul Yadav
- *Fast Simulation of Skin Sliding*
  Richard Southern, Xiaosong Yang and Jian Zhang
- *Interactive Shadowing for 2D Anime*
  Eiji Sugisaki, Feng Tian, Hock Soon Seah and Shigeo Morishima
- *Dealing with Dynamic Changes in Time Critical Decision-Making for MOUT Simulations*
  Shang Ping Ting
- *Stylized Lighting for Cartoon Shader*
  Hideki Todo, Ken Anjyo and Takeo Igarashi
- *Interactive Engagement with Social Agents: An Empirically Validated Framework*
  Henriette van Vugt, Johan Hoorn and Elly Konijn
- *2D Shape Manipulation via Topology-Aware Rigid Grid*
  Yang Wenwu and Feng Jieqing
- *Real-time fluid simulation with adaptive SPH*
  He Yan
- *Compatible Quadrangulation by Sketching*
  Chih-Yuan Yao, Hung-Kuo Chu, Tao Ju and Tong-Yee Lee
- *CSLML: A Markup Language For Expressive Chinese Sign Language Synthesis*
  Baocai Yin, Kejia Ye and Lichun Wang
- *Fireworks Controller*
  Hanli Zhao, Ran Fan, Charlie C. L. Wang, Xiaogang Jin and Yuwei Meng
- *A Unified Shape Editing Framework Based on Tetrahedral Control Mesh*
  Yong Zhao

Short Papers

CASA 2009

A Framework for Evaluating the VISSIM Traffic Simulation with Extended Range Telepresence

Antonia Pérez Arias and Uwe D. Hanebeck
Intelligent Sensor-Actuator-Systems Laboratory (ISAS)
Institute for Anthropomatics
Universität Karlsruhe (TH), Germany
aperez@ira.uka.de, uwe.hanebeck@ieee.org

Peter Ehrhardt, Stefan Hengst, Tobias Kretz, and Peter Vortisch
PTV Planung Transport Verkehr AG
Stumpfstraße 1, D-76131 Karlsruhe, Germany
{Peter.Ehrhardt | Stefan.Hengst | Tobias.Kretz | Peter.Vortisch}@PTV.De

Abstract

This paper presents a novel framework for combining traffic simulations and extended range telepresence. The real user's position data can thus be used for validation and calibration of models of pedestrian dynamics, while the user experiences a high degree of immersion by interacting with agents in realistic simulations.

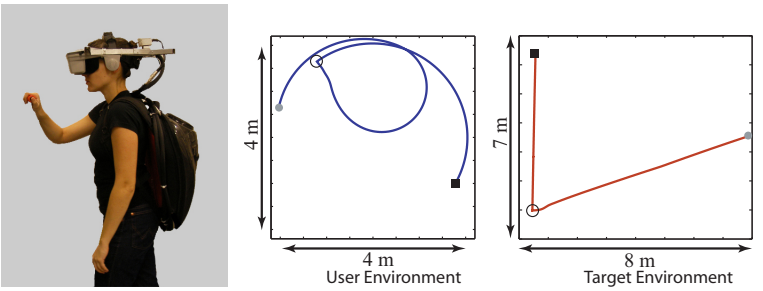
Keywords: Extended Range Telepresence, Motion Compression, Traffic Simulation, Virtual Reality

1 Introduction

The simulation of traffic flow was an early application of computer technology. As the computational effort is larger for the simulation of pedestrian flows, this followed later, beginning in the 1980s and gaining increasing interest in the 1990s. Today, simulations are a standard tool for the planning and design process of cities, road networks, traffic signal lights, as well as buildings or ships.

Telepresence aims at creating the impression of being present in a remote environment. The feeling of presence is achieved by visual and acoustic sensory information recorded from the remote environment and presented to the user on an immersive display. The more of the user's senses are telepresent, the better is the immersion in the target environment. In order to use the sense of motion as well, which is specially important for human navigation and way finding (Darken et al., 1999), the user's motion is tracked and transferred to the *teleoperator* in the *target environment*. This technique provides a suitable interface for virtual immersive simulations, where the teleoperator is an avatar instead of a robot (Rößler et al., 2005). As a result, in extended range telepresence the user can additionally use the proprioception, the sense of motion, to navigate the avatar intuitively by natural walking, instead of using devices like joysticks, keyboards, mice, pedals or steering wheels. Fig. 1(a) shows the user interface in the presented telepresence system.

Our approach combines realistic traffic simulations with extended range telepresence by means of Motion Compression (Nitzsche et al., 2004). We will first shortly sketch the two subsystems: Motion Compression and the traffic simulation. Then, an overview of the system as a whole will be presented, as well as a short experimental validation. Finally, the potentials of the system in various fields of application will be discussed.



(a) (b)

Figure 1: (a) User interface in the extended range telepresence system. (b) The corresponding paths in both environments.

2 Motion Compression

In order to allow exploration of an arbitrarily large target environment while moving in a limited user environment, Motion Compression provides a nonlinear transformation between the desired path in the target environment, the *target path*, and the *user path* in the user environment. The algorithm consists of three functional modules.

First, the *path prediction* gives a prediction of the desired target path based on the user's head motion and on knowledge of the target environment. If no knowledge of the target environment is available, the path prediction is based completely on the user's view direction.

Second, the *path transformation* transforms the target path into the user path in such a way, that it fits into the user environment. In order to guarantee a high degree of immersion the user path has the same length and features the same turning angles as the target path. The two paths differ, however, in path curvature. The nonlinear transformation found by the path transformation module is optimal regarding the difference of path curvature. Fig. 1(b) shows an example of the corresponding paths in both environments.

Finally, the *user guidance* steers the user on the user path, while he has the impression of actually walking along the target path. It benefits from the fact that a human user walking in a goal oriented way constantly checks for his orientation toward the goal and compensates for deviations. By introducing small deviations in the avatar's posture, the user can be guided on the user path. More details can be found in (Nitzsche et al., 2004; Rößler et al., 2004).

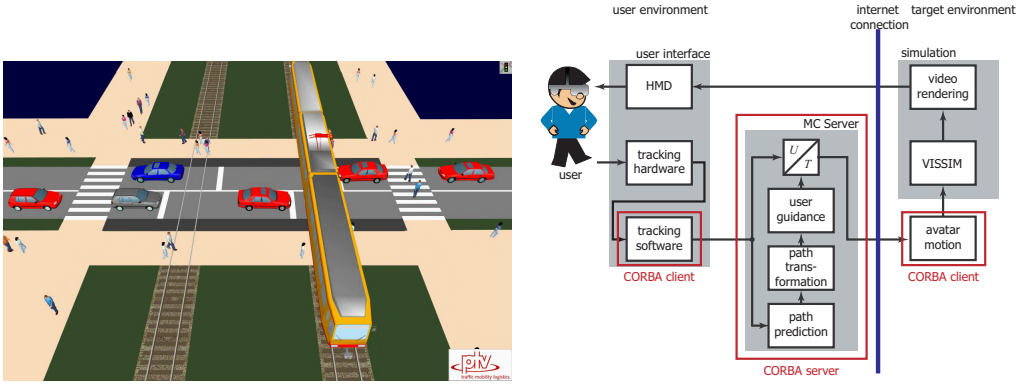
3 VISSIM

VISSIM (Fellendorf and Vortisch, 2001; PTV, 2008) is a multi-modal microscopic traffic flow simulator (fig. 2(a)) that is widely used for traffic planning purposes like designing and testing signal control (see (Fellendorf, 1994) as an example) and verifying by simulation that an existing or planned traffic network is capable of handling a given or projected traffic demand as in (Keenan, 2008).

Recently the simulation of pedestrians has been included in VISSIM. The underlying model is the Social Force Model (Helbing and Molnar, 1995; Helbing et al., 2000).

4 Connecting VISSIM to the Extended Range Telepresence System

The integration of telepresence and VISSIM has been made to exchange data, such that the scene shown to the user is populated with agents (pedestrians) from the VISSIM simulation. The user is blended into the VISSIM simulation such that the agents in the simulation react on him and evade him. In order to be able to control the avatar according to the user's head motion, the user's posture is recorded and fed into the Motion Compression server. Every time an update of the user's posture is available, the three steps of the algorithm are executed as described in section 2. Motion Compression transforms finally the user's posture into the avatars's desired head posture.

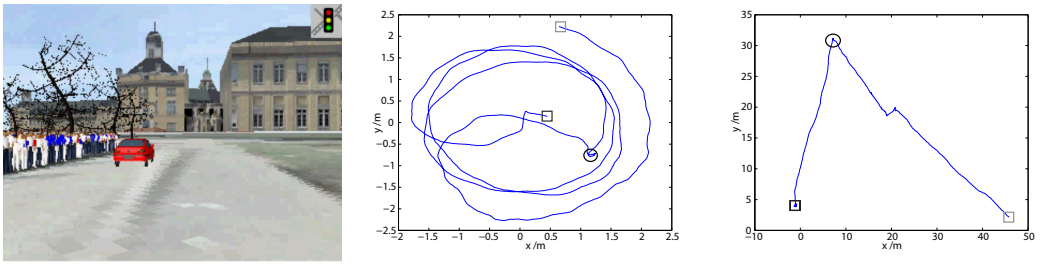


(a) (b)

Figure 2: (a) A snapshot from a VISSIM animation. (Animation online at (PTV, 2008)). (b) Data flow in the proposed telepresence system.

The desired head posture is now sent to VISSIM through an internet connection. The simulation constantly captures live images, which are compressed and sent to the user. Fig. 2(b) shows the whole data flow.

5 Experimental Evaluation



(a) (b) (c)

Figure 3: (a) Impression of the tested scenario. (b) Path in the user environment. (c) Path in the target environment.

The setup uses a high quality head-mounted display of 1280×1024 pixels per eye and a field of view of 60°. The user's posture, i.e., position and orientation, is estimated by an acoustic tracking system that provides 50 estimates per second (Beutler and Hanebeck, 2005). For testing the framework, an environment known to the users was chosen for the VISSIM Simulation. The users were asked to walk from the Karlsruhe Schloss to the ISAS lab's building. The completion time was very similar to the time needed for walking the real path. It is remarkable that the users' velocity increased during the experiment. This indicates that after a couple of minutes of adjustment the user adapts to the system and is able to navigate intuitively through the target environment. An example of the recorded paths in both environments during a test run is shown in fig. 3(b) and 3(c).

6 Conclusions and Outlook

The presented setup is the first step demonstrating the possibilities of the complete system, which provides a unit for first person simulation testing. The extended range telepresence system can be also used for experiments on pedestrian dynamics, since much less data is available for pedestrians

than for vehicles, especially highway traffic, and the currently available data is by far not sufficient for validation and calibration of models of pedestrian dynamics. These experiments in the virtual environment are not only cheap to be set up, but also quick to evaluate, as all positions of the user are available in the system. Having one real person moving through a crowd of simulated agents might also be a good supplement to the validation method proposed in (Hoogendoorn and Daamen, 2007), where one agent is simulated in an environment of data of real pedestrians' movements.

Applications of extended range telepresence in pedestrian simulations include visiting virtual museums and virtual replications of cities or historic buildings. An application with particular focus on gaining spacial knowledge is the simulation of emergency evacuations, where people are trained to find the way out of buildings.

References

Beutler, F. and Hanebeck, U. D. (2005). Closed-form range-based posture estimation based on decoupling translation and orientation. In *Proceedings of IEEE Intl. Conference on Acoustics, Speech, and Signal Processing (ICASSP 2005)*, pages 989–992, Philadelphia, Pennsylvania.

Darken, R. P., Allard, T., and Achille, L. B. (1999). Spatial orientation and wayfinding in large-scale virtual spaces II. *Presence*, 8(6):3–6.

Fellendorf, M. (1994). VISSIM: A microscopic simulation tool to evaluate actuated signal control including bus priority. In *Proceedings of the 64th ITE Annual Meeting*, Dallas, Texas.

Fellendorf, M. and Vortisch, P. (2001). Validation of the microscopic traffic flow model VISSIM in different real-world situations. In *Proceedings of the Transportation Research Board*, Washington, DC.

Helbing, D., Farkas, I., and Vicsek, T. (2000). Simulating dynamical features of escape panic. *Nature*, 407:487–490.

Helbing, D. and Molnar, P. (1995). Social force model for pedestrian dynamics. *Phys. Rev. E*, 51:4282–4286.

Hoogendoorn, S. and Daamen, W. (2007). Microscopic calibration and validation of pedestrian models: Cross-comparison of models using experimental data. In Schadschneider, A., Pöschel, T., Kühne, R., Schreckenberg, M., and Wolf, D., editors, *Traffic and Granular Flow '05*, pages 329–340. Springer-Verlag Berlin Heidelberg.

Keenan, D. (2008). Singapore kallang-paya lebar expressway (KPE) phase 1: A tunnel congestion management strategy derived using VISSIM. In *Proceedings of the 3rd Intl. Symposium on Transport Simulation (ISTS 2008)*, Queensland, Australia. (eprint).

Nitzsche, N., Hanebeck, U. D., and Schmidt, G. (2004). Motion compression for telepresent walking in large target environments. *Presence*, 13(1):44–60.

PTV (2008). *VISSIM 5.10 User Manual*. PTV Planung Transport Verkehr AG, Stumpfstraße 1, D-76131 Karlsruhe. http://www.vissim.de/.

Rößler, P., Beutler, F., Hanebeck, U. D., and Nitzsche, N. (2005). Motion compression applied to guidance of a mobile teleoperator. In *Proceedings of the IEEE Intl. Conference on Intelligent Robots and Systems (IROS 2005)*, pages 2495–2500, Edmonton, Canada.

Rößler, P., Hanebeck, U. D., and Nitzsche, N. (2004). Feedback controlled motion compression for extended range telepresence. In *Proceedings of IEEE Mechatronics & Robotics (MechRob 2004), Special Session on Telepresence and Teleaction*, pages 1447–1452, Aachen, Germany.

Hybrid Motion Control combining Inverse Kinematics and Inverse Dynamics Controllers for Simulating Percussion Gestures

Alexandre Bouënard *† Sylvie Gibet *‡ Marcelo M. Wanderley †

* SAMSARA/VALORIA, Université de Bretagne Sud, France
† IDMIL/CIRMMT, McGill University, Qc., Canada
‡ Bunraku/IRISA, Université de Rennes I, France

Abstract

Virtual characters playing virtual musical instruments in a realistic way need to interact in real-time with the simulated sounding environment. Dynamic simulation is a promising approach to finely represent and modulate this interaction. Moreover, capturing human motion provides a database covering a large variety of gestures with different levels of expressivity. We propose in this paper a new data-driven hybrid control technique combining Inverse Kinematics (IK) and Inverse Dynamics (ID) controllers, and we define an application for consistently editing the motion to be simulated by virtual characters performing percussion gestures.

Keywords: Physics-based Computer Animation, Hybrid Motion Control

1 Introduction

Playing a musical instrument involves complex human behaviours. While performing, a skilled musician is able to precisely control his motion and to perceive both the reaction of the instrument to his actions and the resulting sound. Transposing these real-world experiences into virtual environments gives the possibility of exploring novel solutions for designing virtual characters interacting with virtual musical instruments.

This paper proposes a physics-based framework in which a virtual character dynamically interacts with a physical simulated percussive instrument. It enables the simulation of the subtle physical interactions that occur as the stick makes contact with the drum membrane, while taking into account the characteristics of the preparatory gesture. Our approach combines human motion data and a hybrid control method composed of kinematics and physics-based controllers for generating compelling percussion gestures and producing convincing contact information.

Such a physics framework makes possible the real-time manipulation and mapping of gesture features to sound synthesis parameters at the physics level, producing adaptative and realistic virtual percussion performances[1].

2 Related Work

Controlling adaptative and responsive virtual characters has been intensively investigated in computer animation research. Most of the contributions have addressed the control of articulated figures using robotics-inspired ID controllers. This has inspired many works for handling different types of motor tasks such as walking, running (Hodgins et al, 1995), as well as composing these tasks (Faloutsos et al, 2001) and easying the hard process of tuning such controllers (Allen et al, 2007).

[1] More details about sound synthesis schemes, as well as our system architecture can be found in (Bouënard et al, 2009).

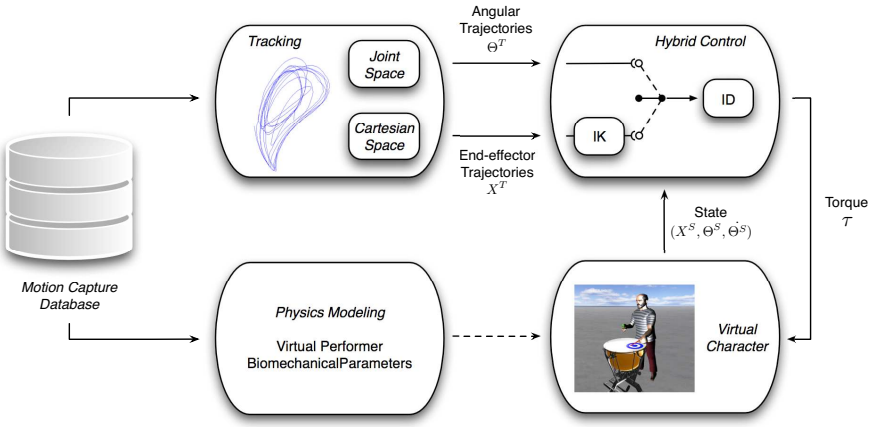


Figure 1: Physics-based motion capture tracking, either in the *Joint Space* from angular trajectories, or in the *Cartesian Space* from end-effector trajectories. The *Hybrid Control* involves the combination of *IK* and *ID* controllers.

More related to our work are hybrid methods, based on the tracking of motion capture data performed by a fully dynamically controlled character. The specificity of our contribution lies in the integration and the collaboration of IK and ID controllers, rather than handling strategies for transtioning between kinematic and dynamic controllers (Shapiro et al, 2003; Zordan et al, 2005). IK has also been used as a pre-process for modifying the original captured motion and simulating it on a different character anthropometry (Zordan and Hodgins, 1999). We rather use IK as a basis of our hybrid control method for specifying the control of a dynamic character from end-effector trajectories. This hybrid collaboration is particularly consistent for the synthesis of percussive gestures, which is not taken into account in previous contributions (Zordan and Hodgins, 1999; Bouënard et al, 2008-a).

3 Data-driven Hybrid Motion Control

A motion capture database contains a set of various percussion performances including different drumstick grips, various beat impact locations and several musical playing variations. We propose two ways for achieving the motion control (Figure 1), either by tracking motion capture data in the Joint space (angular trajectories), or tracking end-effector trajectories in the 3D Cartesian space. Tracking motion capture data in the Joint space requires ID control, whereas tracking in the end-effector (Cartesian) space requires both IK and ID (hybrid) control.

In the latter case, end-effector targets (X^T) in the 3D Cartesian space are extracted from the motion capture database, and used as input for the IK algorithm to compute a kinematic posture Θ^T (vector of joint angular targets). We chose the Damped Least Squares method (Wampler, 1986) equation (1), a robust adaptation of the pseudo-inverse regarding the singularity of the Inverse Kinematics problem. J^+_Θ is the damped adaptation of the pseudo-inverse of the Jacobian, and X^S represents the current end-effector position of the system to be controlled. Other traditional IK formulations may be equally used, as well as learning techniques (Gibet and Marteau, 2003).

Angular targets Θ^T and current states (Θ^S, $\dot{\Theta^S}$) are then used as inputs of the ID algorithm, equation (2), for computing the torque (τ) to be exerted on the articulated rigid bodies of the dynamical virtual character. This one is composed of rigid bodies articulated by damped springs parameterized by damping and stiffness coefficients (k_d, k_s).

$$\Delta\Theta^T = \lambda.J^+_\Theta.(X^T - X^S), \quad \Theta^T = \Theta^S + \Delta\Theta^T \tag{1}$$

$$\tau = k_s.(\Theta^S - \Theta^T) - k_d.\dot{\Theta^S} \tag{2}$$

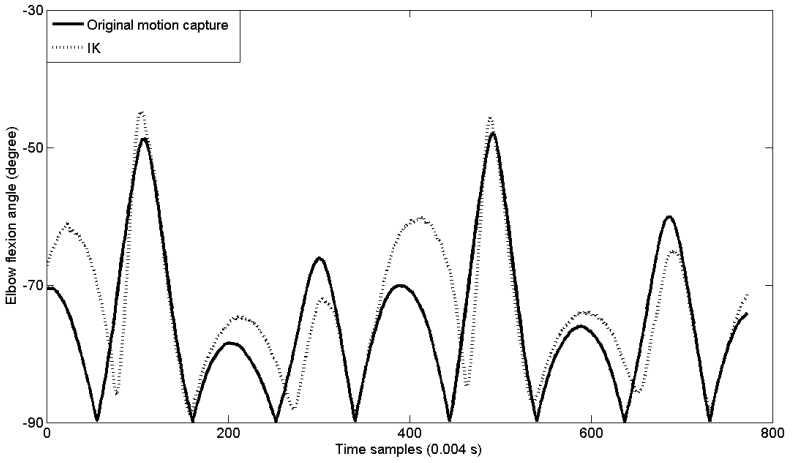


Figure 2: Comparison of elbow flexion angle trajectories: original motion capture data vs. data generated by the IK algorithm.

This hybrid approach enables the manipulation of physically simulated motion capture data in the 3D Cartesian space (X^T) instead of the traditional angular space (Θ^T). It is indeed more consistent and intuitive to use end-effector trajectories for controlling percussion gestures, for instance drumstick extremities obtained from the motion capture database.

4 Results

The results obtained by the two tracking modes are compared, keeping the same parameterization of the damped springs composing the virtual character. We ran the simulation on a set of percussion gestures (French grip, legato) recorded at a sample rate of 250 Hz for capturing the whole body of the performer, as well as the drumsticks. The hybrid control scheme tracks one percussion gesture for synthesizing whole arm movements solely from the specification of drumstick tip trajectories.

Figure 2 presents the comparison between raw motion capture data and data generated by the IK process. It shows that data generated by the IK formulation are consistent with real data, especially for the elbow flexion angle that is one of the most significant degree of freedom of the arm in percussion gestures, especially during preparatory phases (Bouënard et al, 2008-b).

Finally, we present the comparison of the two control modes (ID control only and hybrid control) in Figure 3. One interesting issue is the accuracy of the hybrid control mode compared to the simple ID control. This observation lies in the fact that the convergence of motion capture tracking is processed in the Joint space in the case of ID control, adding and amplifying multiple errors on the different joints and leading to a greater error than processing the convergence in the Cartesian space for the hybrid control. The main drawback of this improvement is however the additional computationnal cost of the IK algorithm which is processed at every simulation step. It nevertheless provides a more consistent and flexible motion edition technique for controlling a fully physics-based virtual character.

5 Conclusion

We proposed in this paper a physically-enabled environment in which a virtual character can be physically controlled and interact with the environment, in order to generate virtual percussion performances. More specifically, the presented hybrid control mode combining IK and ID controllers leads to a more intuitive yet effective way of editing the motion to be simulated only from drumstick extremity trajectories. Future work includes the extension and improvement of our hybrid control technique for editing and simulating percussion motion in the 3D Cartesian space.

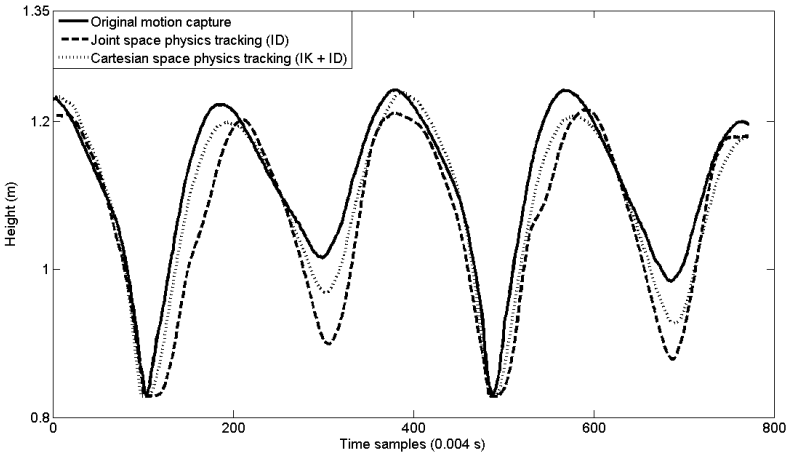


Figure 3: Comparison of drumstick trajectories: original motion capture data vs. Joint space (ID) physics tracking vs. Cartesian space (IK + ID) physics tracking.

References

Bouënard, A., Gibet, S. and Wanderley, M. M. (2009). Real-Time Simulation and Interaction of Percussion Gestures with Sound Synthesis. Technical Report, in *HAL Open Archives*.

Hodgins, J., Wooten, W., Brogan, D. and O'Brien, J. (1995). Animating Human Athletics. In *SIGGRAPH Computer Graphics*, pages 71–78.

Faloutsos, P., van de Panne, M. and Terzopoulos, D. (2001). Composable Controllers for Physics-Based Character Animation. In *Proc. of the SIGGRAPH Conference on Computer Graphics and Interactive Techniques*, pages 251–260.

Allen, B., Chu, D., Shapiro, A. and Faloutsos, P. (2007). On the Beat!: Timing and Tension for Dynamic Characters. In *Proc. of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 239–247.

Shapiro, A., Pighin, F., and Faloutsos, P. (2003). Hybrid Control for Interactive Character Animation. In *Proc. of the Pacific Conference on Computer Graphics and Applications*, pages 455–461.

Zordan, V., Majkowska, A., Chiu, B. and Fast, M. (2005). Dynamic Response for Motion Capture Animation. In *Transactions on Graphics*, 24(3):697–701. ACM.

Zordan, V. and Hodgins, J. (1999). Tracking and Modifying Upper-body Human Motion Data with Dynamic Simulation. In *Proc. of Computer Animation and Simulation*, pages 13–22.

Bouënard, A., Gibet, S. and Wanderley, M. M. (2008-a). Enhancing the Visualization of Percussion Gestures by Virtual Character Animation. In *Proc. of the International Conference on New Interfaces for Musical Expression*, pages 38–43.

Wampler, C. (1986) Manipulator Inverse Kinematic Solutions based on Vector Formulations and Damped Least Squares. In *IEEE Trans. on Systems, Man and Cybernetics*, 16(1):93–101. IEEE Press.

Gibet, S. and Marteau, P. F. (2003). Expressive Gesture Animation based on Non-Parametric Learning of Sensory-Motor Models. In *Proc. of the International Conference on Computer Animation and Social Agents*, pages 79–85.

Bouënard, A., Wanderley, M. M. and Gibet, S. (2008-b). Analysis of Percussion Grip for Physically Based Character Animation. In *Proc. of the International Conference on Enactive Interfaces*, pages 22–27.

Laughing, Crying, Sneezing and Yawning: Automatic Voice Driven Animation of Non-Speech Articulations*

Darren Cosker
Department of Computer Science
University of Bath
D.P.Cosker@cs.bath.ac.uk

James Edge
Centre for Vision, Speech and Signal Processing, University of Surrey
J.Edge@surrey.ac.uk

Abstract

In this paper a technique is presented for learning audio-visual correlations in non-speech related articulations such as laughs, cries, sneezes and yawns, such that accurate new visual motions may be created given just audio. We demonstrate how performance accuracy in voice driven animation can be related to maximizing the models likelihood, and that new voices with similar temporal and spatial audio distributions to that of the model will consistently provide animation results with the lowest ground truth error. By exploiting this fact we significantly improve performance given voices unfamiliar to the system.

Keywords: Voice Driven Facial Animation

1 Introduction

In this paper we propose a data-driven HMM based method for learning correlations between non-speech related audio signals – specifically, laughing, crying, sneezing and yawning – and visual facial parameters. Unlike previous work dealing with the audio-visual modeling of this class of signals (DiLorenzo et al., 2008), our data is observed from recorded motions of real performers as opposed to a pre-defined physical model. Unlike previous audio-driven HMM based synthesis work (e.g. (Brand, 1999)), we also attempt to specifically address person independence in our framework. We concentrate on several common non-speech related actions – laughing, crying, sneezing and yawning. A major challenge when using automatic audio driven systems is that of achieving reliable performance given a variety of voices from new people. We demonstrate our approach in a number of speaker-independent synthesis experiments, and show how animation error in voice driven animation has a relation to the proximity of audio distributions for different people and well as similarities between their temporal behaviour. By exploiting these facts we consistently improve synthesis given voices from new people. We implement this improvement using a pre-synthesis classification step. In sum, our approach potentially increases the reusability of such a model for new applications (e.g. online games), and can reduces the need to retrain the model for new identities. Our approach initially requires example audio-visual performances of the action of interest for training: e.g. several laughs, cries, sneezes or yawns. A HMM framework then encodes this audio-visual information. The framework may be trained using any number of desired non-speech action types.

2 Audio-Visual Data Acquisition

Our data set consisted of four participants (2 male and 2 female) captured performing approximately 6-10 different laughs, cries, sneezes and yawns using a 60Hz Qualysis optical motion-capture system. We captured audio simultaneously at 48KHz. We placed 30 retro-reflective markers on each person in order to capture the visual motion of their face while performing the different

*Thanks to the Royal Academy of Engineering and EPSRC for partially funding this work.

actions. We remove head pose from our data set using a least-squares alignment procedure. We then pick one identity from the data set as the base identity and normalise the remaining three identities such that their mean motion-capture vector is the same as the mean for the base. Finally, we perform PCA on the data to reduce its dimensionality, and use the notation **V** to refer this data set. We represent audio using Mel-Frequency Cepstral Coefficients (MFCCs), and use the notation **A** to refer to this data.

3 Modelling Audio-Visual Relationships

Observing audio-visual signals for different non-speech related articulations reveals evidence of a temporal structure. We therefore decided to model this behaviour using HMMs (Rabiner, 1989). We first consider a traditional HMM trained using visual data. Let us consider this data to be a set of example non-speech sounds from **V**. After training, the HMM may be represented using the tuple $\lambda_v = (\mathbf{Q}, \mathbf{B}, \pi)$, where **Q** is the state transition probability distribution, **B** is the observation probability distribution, and π is the initial state distribution. In our model, each of the K states in a HMM are represented as a Gaussian mixture $G_v = (\mu_v, \sigma_v)$, where μ_v and σv are the mean and covariance. Each state therefore represents the probability of observing a visual vector.

Given an example visual data sequence, we may calculate the visual HMM state sequence most likely to have generated this data using the Viterbi algorithm. However, we wish to slightly modify the problem such that we may estimate the visual state sequence given an *audio* observation instead. This is our animation goal, i.e. automatic animation of visual parameters given speech. We can do this by remapping the visual observations to audio ones using the learned HMM parameters, i.e. for each G_v we calculate the distribution $G_a = (\mu_a, \sigma_a)$ based on the audio **A** corresponding to the visual vectors **V** used in HMM training.

Using the Viterbi algorithm, we may now estimate the most probable visual state sequence using an audio observation. More formally, we can estimate via the HMM the post probable hidden sequence of Gaussian distribution parameters μ_v and σ_v corresponding to the observation sequence of MFCC vectors. We next consider what visual parameters $\vec{v}_t$ to display at output for each state.

We first partition the visual parameter distribution used to train the HMM into distinct regions based on the proximity of a visual parameter to each gaussian. Using μ_v and σ_v, we calculate the Mahalanobis distance between each observation $\vec{v}_i$ and each of the K states and assign a visual parameter to its closest state. This results in K partitions of the parameter training set, and given an audio observation we may now state that the visual parameter to display at time t given $\vec{a}_t$ is taken from the visual parameter partition associated with the state at time t. In order to find an optimal output visual parameter sequence, we again utilise the Viterbi algorithm.

Figure 1 gives an overview of visual synthesis, and defines it in terms of two levels: High-Level Re-synthesis, and Low-Level Resynthesis. The High-Level stage is concerned with initially selecting the visual state sequence through the HMM given the audio input. This results in a sequence of visual parameter partitions – one for each time t. The Low-Level stage then uses the Viterbi algorithm to find the most probable path through these partitions given the observed audio. Resulting visual parameters are converted back in to 3D visual motion vectors by projecting back through the PCA model. An RBF mapping approach (Lorenzo et al., 2003) is then used to animate a 3D facial model for output using this data.

3.1 Speaker Independence via Best Matching Person Selection

It is often highly desirable for a voice driven system to be robust to a wide range of different voices. Several design options exist in this case, including: (1) a single HMM trained with the knowledge of multiple people, or (2) one of several HMMs where each contains audio-visual data for a specific person. We concentrate on the latter case for now, so our problem is therefore to select one of several HMMs where each encodes information from a specific identity. It turns out that this is equivalent to determining the probability that a specific HMM generated the observation. Calculating this probability may be achieved by estimating the log-likelihood that a HMM could

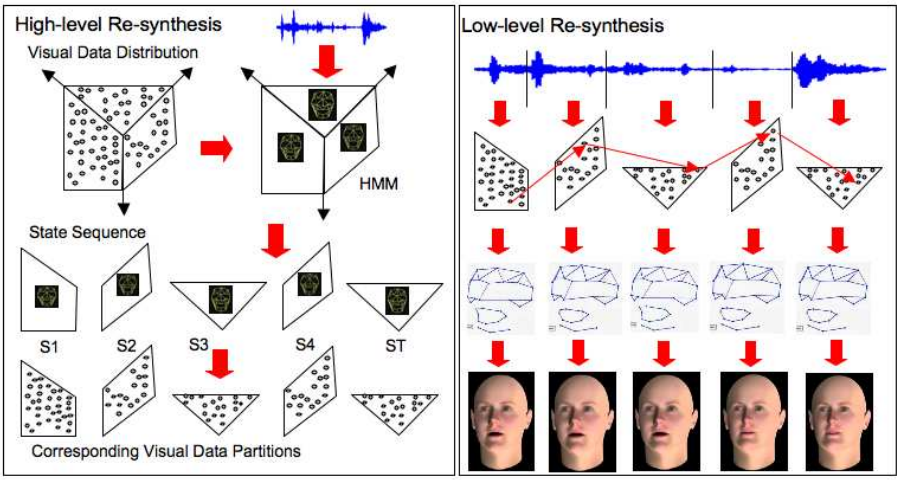


Figure 1: Animation production may be visualised as a high-level state based process followed by a low-level animation frame generation process.

have generated the persons input audio (Rabiner, 1989). We show in our results how selecting a HMM with a higher log-likelihood consistently leads to a lower overall animation error.

4 Experimental Results and Future Directions

We first consider person and action specific synthesis of animations. We trained audio-visual HMMs for a range of specific non-speech actions – laughing, crying, sneezing and yawning – for each of our four performers. Each HMM was trained using approximately 4 different actions, and approximately 4 more were left out for the test cases. Audio corresponding to the test cases was then used to synthesise new 3D animation vectors which were compared to the motion-capture ground truth. Example animations may be found in the video, and RMS errors in millimeters may be found in Table 1.

Person	Laugh			Cry			Sneeze			Yawn		
	Min	Max	Mean	Min	Max	Mean	Min	Max	Mean	Min	Max	Mean
P1	0.7	1.42	0.95	0.89	2.3	1.36	0.6	1.5	1.99	1.8	5.1	2.49
P2	2.68	4.7	3.68	1.96	2.42	2.12	3.8	5.6	4.56	1.99	4.13	2.8
P3	0.93	1.49	1.19	1.57	2.25	1.96	0.6	0.92	0.92	ND	ND	ND
P4	1.75	2.16	1.92	1.11	1.4	1.24	1.57	2.52	2	3.74	5.8	4.55

Table 1: Action Specific HMM animation: Min, Max and Mean RMS errors (millimetres) for average synthesised 3D coordinates versus ground truth 3D coordinates.

Person	Laugh			Cry			Sneeze			Yawn		
	Min	Max	Mean	Min	Max	Mean	Min	Max	Mean	Min	Max	Mean
P1+P2+P3+P4	1.15	2.76	1.75	1.29	3.61	2.01	1.6	5.96	3.52	1.77	6.15	3.52

Table 2: Animation with HMMs encoding multiple actions: Min, Max and Mean RMS errors (millimetres) for average synthesised 3D coordinates versus ground truth 3D coordinates.

We next tested combining data from multiple people performing a specific non-speech action inside the same HMM. This assesses the models ability to generalise data for different people within the same model. Again, we left out part of the data for each performer to use as a test-set and calculated RMS errors as shown in Table 2.

	Laugh		Cry		Sneeze		Yawn	
	B / W E	B/W L	**B / W E**	B / W L	**B / W E**	B / W L	**B / W E**	B / W L
P1	**2 / 3.3**	-1033/-1126	**2.08/2.45**	-704/-851	**2.4/2.46**	-749/-1105	**2.8/6.6**	-607/-1239
P2	**2.3/2.5**	-422/-777	**1.3/2.2**	-662/-1130	**3.4/3.66**	-748/-1006	**3.5/5.4**	-1081/-1281
P3	**1.6/2.8**	-763/-2558	**1.1/2.1**	-857/-3519	**2.4/2.9**	-1050/-2352	ND	ND
P4	**1.7 / 3**	-1085/-2039	**0.8 / 2.1**	-770/-1804	**1.5/2.7**	-902/-1627	**1.8/2.8**	1073/1215

Table 3: Average 3D vector animation error (millimeters) given best and worst matching (log-likelihood) HMMs. (B/W E = best/worst error, B/W L = best/worst log-likelihood)

We now test the case where the model has no prior knowledge of a persons voice For each performer we trained four separate HMMs – one for each action. Given input audio for an action, the HMM with the best log-likelihood was selected for synthesis – thus taking into account match between input audio distribution and those of the trained HMMs. Table 3 shows the results, and Figure 2 gives side-by-side comparisons between ground truth video data of a performer, reconstructed 3D vectors, and an animated 3D facial model. Our results clearly show that a HMM with a higher log-likelihood always gives a lower average error reconstructions error. This shows that a high log-likelihood appears correlated with a low animation error. Future work will involve automatically discriminating between non-speech sounds and normal speech, with the eventual aim of animating faces from entirely natural and unconstrained input audio.

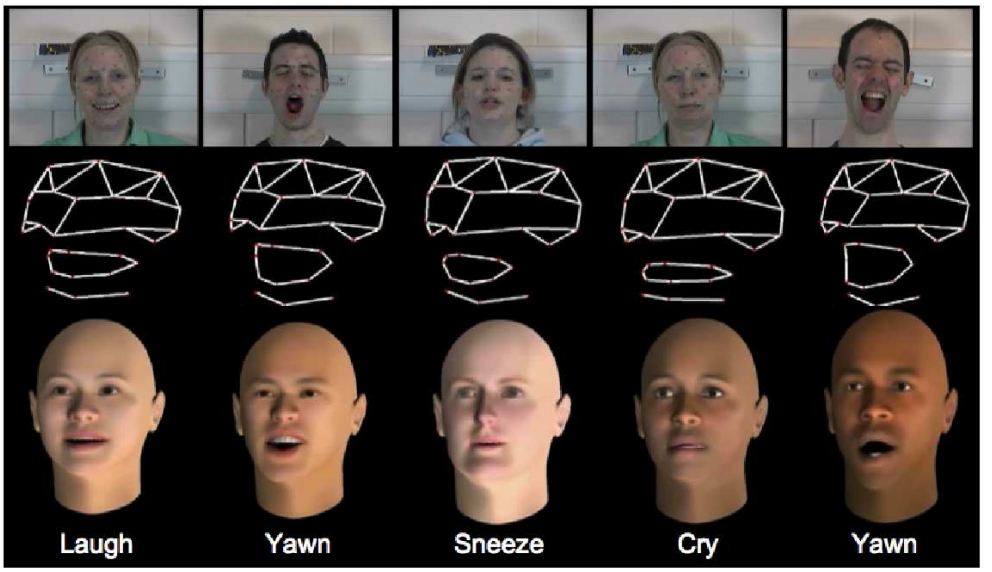


Figure 2: Example Animation Frames. (Top) Ground truth video. (Middle) Corresponding 3D Motion vectors automatically synthesised from speech. (Bottom) A 3D head model animated using the motion vectors using an RBF mapping technique.

References

Brand, M. (1999). Voice puppetry. In *Proc. of SIGGRAPH*, pages 21–28. ACM Press.

DiLorenzo, P., Zordan, V., and Sanders, B. (2008). Laughing out loud: Control for modelling anatomically inspired laughter using audio. *ACM Trans. Graphics*, 27(5).

Lorenzo, M. S., Edge, J., King, S., and Maddock, S. (2003). Use and re-use of facial motion capture data. In *Proc. of Vision, Video and Graphics*, pages 135–142.

Rabiner, L. R. (1989). A tutorial on hidden markov models and selected applications in speech recognition. *Proc. of the IEEE*, 77(2):257–285.

Mixed-Initiative Authoring for Augmented Scene Modeling

Carles Fernández, Pau Baiget, Jordi Gonzàlez
Computer Vision Centre – Edifici O, Campus UAB, 08193, Bellaterra, Spain
perno@cvc.uab.es

Abstract

This contribution proposes a virtual storytelling interface that augments offline video sequences with virtual agents. A user is allowed to describe behavioral plot lines over time using natural language texts. Virtual agents accomplish the given plots by following (i) spatiotemporal patterns learnt from recordings, and (ii) behavioral models governed by an ontology. Such behavioral models are also modified and extended online, permitting the user to adjust them for a desired performance. The resulting interactions among virtual and real entities are visualized in augmented sequences generated online. Several experiments of different nature have been conducted in an intercity traffic domain, to account for the flexibility and interaction possibilities of the presented framework. Such capabilities include defining complex behaviors on-the-fly, adding naturalism to the goal-based realizations of the virtual agents, or providing advanced control towards final augmented sequences.

1 Introduction

Both virtual storytelling and augmented reality constitute emerging applications in fields like computer entertainment and simulation. The main concern of virtual storytelling is to provide flexible and natural solutions that produce generally complex sequences automatically. On the other hand, a challenge for augmented reality consists of providing the generated virtual agents with autonomous or complex behaviors. One of the main current challenges on these fields consists of bringing complex high-level modeling closer to the users, so that it becomes both intuitive and powerful for them to author mixed scenes.

Following [5], some of the most clear future challenges in creating realistic and believable Virtual Humans consist of generating on-the-fly flexible motion and providing them with complex behaviors inside their environments, as well as making them interactive with other agents. On the other hand, interaction between real and virtual agents has been little considered previously [3]. Gelenbe et al. [3] proposed an augmented reality system combining computer vision with behavior–based non–human agents. Zhang et al. [7] presented a method to merge virtual objects into video sequences recorded with a freely moving camera. The method is consistent regarding illumination and shadows, but it does not tackle occlusions with real moving agents. Existing works on virtual storytelling typically use AI-related approaches such as heuristic search and planning to govern the behaviors of the agents. Cavazza et al. [1] describe an interactive virtual storytelling framework based on Hierarchical Task Networks. Lee et al. [4] describe a Responsive Multimedia System for virtual storytelling, in which external users interact with the system by means of tangible, haptic and vision-based interfaces.

We propose a framework in which a virtual storytelling interface allows users to author original recordings, extending them with virtual agents, by introducing goals for them at specific points along the video timeline. Once a plotline is given, virtual agents follow it according to a defined scene model. Additionally, our approach offers two interesting contributions: first, a user can model and extend virtual agent behaviors in a flexible way, being enabled to define arbitrarily complex occurrences easily. In the second place, to improve the naturalness of the virtual agents, we base their concrete spatiotemporal realizations on patterns learnt from real agents in the scenario. Our solution achieves mixed-initiative authoring and advanced scene augmentation, and provides benefits to fields such as simulation or computer animation.

2 Real Scene Analysis

Virtual agents must be aware of real occurrences, in order to decide for reactions. Instantaneous real world information is analyzed in 3 steps: (i) tracking relevant scene objects and extracting spatiotemporal data; (ii) qualifying these data in terms of low-level predicates, using a rule-based reasoning engine; and (iii) inferring higher-level patterns of behavior by applying inductive mechanisms of decision.

The tracking algorithm has been implemented following [2], which describes an efficient real-time method for detecting moving objects in unconstrained environments. In order to carry out further analyses over real world data, the spatiotemporal statuses are conceptualized. To do so, we use the Fuzzy Metric Temporal Logic (FMTL) formalism proposed in [6], which incorporates conventional logic formalisms and extends them by fuzzy and temporal components. The instantaneous values for a target Id are encoded into temporally-valid predicates of the form t ! $has\_status(Id, x, y, \theta, v, a, \alpha)$, stating its 2D-position, orientation, velocity, action, and current progression within the action cycle, at time-step t. These quantitative values are fuzzified and, after that, an FMTL reasoning engine processes every *has_status* predicate, and derives goal-oriented predicates such as $has\_velocity(Id, V)$ or $is\_standing(Id, Loc)$.

The conceptual knowledge about agent behavior is encoded in a set of rules in FMTL and organized within a behavior modeling formalism called *Situation Graph Tree* (SGT) [2]. SGTs build behavioral models by connecting a set of defined situations by means of prediction and specialization edges. When a set of conditions is asserted, a high-level predicate is produced as an interpretation of a situation. In this work, instead of inferring high-level situations from low-level information, we use SGTs to decompose abstract and vague linguistic explanations into concrete sequences of low-level actions. More information can be found in [2].

3 Ontologically-based Linguistic Analysis

The main motivation for the use of ontologies is to capture the knowledge involved in a certain domain of interest, by specifying conventions about its implied content. In our case, the behavioral models introduced by a user must conform to the chosen domain; also, input texts refer to entities that the system should identify. An ontology has been created for our pursued domain, unifying the possible situations, agents, semantic locations, and descriptors that constrain the domain, and establishing relationships among them. For instance, a `Theft` situation links a thief `Agent` with a victim `Agent` through a stolen `PickableObject`. Two additional ontological resources are considered: an *episodical database*, which accounts the history of instantiated situations to enable retrieval capabilities; and an *onomasticon*, a dynamic repository that maintains the set of identifiers that are used by different processes to refer to active entities in the scene.

NLU (Natural Language Understanding) is regarded as a process of hypothesis management that decides for the most probable interpretation of a linguistic input. Following this idea, the NLU module links plotline sentences to their most accurate domain interpretations, in form of high-level predicates. Input sentences are analyzed through a sequence of 3 basic processes: a *morphological parser*, which tags the sequence of words and identifies those ones linked to relevant concepts of the domain; a *syntactic/semantic parser*, which recursively builds dependency trees out of the tagged sentence; and finally, a *predicate assignment* process, which compares the resulting tree of highlighted concepts with a list of tree patterns, by computing a semantically-extended Tree Edit Distance. Each pattern tree is linked to a conceptual predicate that interprets it. The predicate of the closest pattern tree is selected as the most valid interpretation of the input sentence. Further lexical disambiguation is accomplished by relying on the WordNet lexical database [1] to retrieve lists of closely related words, using semantic metrics based on relationships such as synonymy and hypernymy. New candidates are evaluated to determine the ontological nature of an unknown word; as a result, the word is linked to a number of domain concepts that can explain it.

[1] http://wordnet.princeton.edu/

4 Conceptual Planner

Each plotline predicate produced by the NLU module instantiates a high-level event, which must be converted into a list of explicit spatiotemporal actions. At this point, we are interested in providing the user with a device to define behavioral patterns for the agents, still keeping it an intuitive solution with interactive operability. The proposed conceptual planner is based on the reasoning engine and the situation analysis framework already described.

Each high-level predicate is decomposed into a temporal sequence of lower-level objectives. For instance, we may want to define a pedestrian situation *"P1 meets P2"* as the sequence (i) *"P1 reaches P2"*, and (ii) *"P1 and P2 face each other"*, or translated into FMTL predicates:

$$meet(P1, P2) \vdash go(P1, P2) \rightarrow faceTowards(P1, P2) \vee faceTowards(P2, P1) \quad (1)$$

The SGT framework facilitates encoding such information in an easy way. We define a situation s as a pair formed by a set of conditions and a set of reactions, $s = \langle \mathcal{C}, \mathcal{R} \rangle$. Then, a behavior b is encoded as a linear sequence of defined situations, $b = \{s_1, \dots s_N \mid s_{i-1} \prec s_i\}, \forall i = 0 \dots N$, where $\prec$ is the temporal precedence operator.

5 Path Manager

The final step of the top-down process decides detailed spatiotemporal realizations of the virtual agents. Storytelling plots cannot fully specify agent trajectories. Instead, we take advantage of the observed footage of real agents, in order to extract statistical patterns that suggest common realizations. A trajectory τ is a time-ordered sequence of ground plane positions, $\tau = \{x(t)\}$. A training set $\mathcal{T} = \{\tau_n\}, n \in 1 \dots N$ contains all trajectories observed by the trackers. Each trajectory τ starts at an entry point a and ends at an exit point b; when several trajectories follow similar patterns, common entry and exit areas A and B can be identified. Depending on the tracking accuracy and the scenario conditions, trajectories might lack of smoothness, being noisy or non-realistic representations of the actual target motion. To solve this, a continuous cubic spline $s(\tau)$ is found to fit each trajectory $\tau \in \mathcal{T}$. Finally, a sequence of K equidistant control points is sampled from each spline, obtaining $\tilde{s}(\tau_n) = \delta_k \cdot s(\tau_n) = \{\tilde{x}_n^1, \dots, \tilde{x}_n^K\}$.

6 Experimental Results

Several recordings of 3 intercity traffic scenarios containing real actors have been provided for the experiments. An external user provides the plots and receives the augmented scene, and is allowed to interactively change the plots or models towards a desired solution. Fig. 1 includes some snapshots from the augmented sequences that were automatically extracted from the three plots tested. In the *Discorteous bus* scene, a complex behavior "missing a bus" is defined by a small number of simple situations. The plot used for this scene is: "*An urban bus appears by the left. It stops in the bus stop. A pedestrian comes by the left. This person misses the bus.*" The framework also allows testing and correcting the many possible ways of modeling such an ambiguous behavior, until reaching a convincing result for the user.

The *Anxious meeting* sequence states how the real world influences the decisions of the virtual agents: depending on the behavior of a real police agent, who gives way to vehicles or to pedestrians, a virtual agent can either wait in the sidewalk or directly enter the crosswalk, in order to meet somebody in the opposite sidewalk. The plot used for one of the sequences is as follows: "*A person is standing at the upper crosswalk. A second pedestrian appears by the lower left side. He meets with the first pedestrian.*". The moment at which the agent appears, or its velocity, determine the development of the story and affect to further occurrences.

Finally, the *Tortuous walk* sequence includes several pedestrians walking around an open scenario. The system has learnt from observed footage of real agents in the location, so that virtual agents know typical trajectories to reach to any point. Virtual agents select the shorter learnt path, or the one that avoids collisions with agents. The plot tells the agent to go to different zones A, B, or C at different moments of time. Last snapshot of Fig. 1(c) shows the trajectories performed by virtual agents.

Figure 1: Selected frames from *Discorteous bus* (top), *Anxious meeting*, and *Tortuous walk*.

7 Conclusions

We have presented a framework to author video augmentations of domain-specific recordings by inputting natural language plotlines. The proposed framework accomplishes behavior-based scene augmentation by means of a two-fold strategy, in order to (i) enable the user to model high-level behaviors interactively, and (ii) automatically learn spatiotemporal patterns of real agents, and use them for low-level animation. The experiments carried out demonstrate advantages of this approach, such as the control of the user over unexpected or time-dependent situations, automatic learning of regular spatiotemporal developments, or the reaction of the virtual agents to the real scene occurrences.

8 Acknowledgements

This work is supported by EC grants IST-027110 for the HERMES project and IST-045547 for the VIDI-video project, and by the Spanish MEC under projects TIN2006-14606 and CONSOLIDER-INGENIO 2010 MIPRCV CSD2007-00018.

References

[1] Cavazza, M., Charles, F., and Mead, S. (2001). Agents interaction in virtual storytelling. *Intelligent Virtual Agents, Springer LNAI 2190*, pages 156–170.

[2] Fernández, C., Baiget, P., Roca, X., and Gonzàlez, J. (2008). Interpretation of complex situations in a semantic-based surveillance framework. *Signal Processing: Image Communication*.

[3] Gelenbe, E., Hussain, K., and Kaptan, V. (2005). Simulating autonomous agents in augmented reality. *Journal of Systems and Software*, 74(3):255–268.

[4] Lee, Y., Oh, S., and Woo, W. (2005). A Context-Based Storytelling with a Responsive Multimedia System (RMS). In *ICVS 2005*, Strasbourg, France. Springer.

[5] Magnenat-Thalmann, N. and Thalmann, D. (2005). Virtual humans: thirty years of research, what next? *The Visual Computer*, 21(12):997–1015.

[6] Schäfer, K. (1997). Fuzzy spatio-temporal logic programming. In Brzoska, C., editor, *Proc. of 7th Workshop in Temporal and Non-Classical Logics (IJCAI'97)*, pages 23–28, Nagoya, Japan.

[7] Zhang, G., Qin, X., An, X., Chen, W., and Bao, H. (2006). As-consistent-as-possible compositing of virtual objects and video sequences. *CAVW*, 17(3-4):305–314.

Real-Time Simulation of Pedestrian Groups in an Urban Environment

Murat Haciomeroglu, Robert G. Laycock and Andy M. Day
University of East Anglia
University of East Anglia Norwich NR4 7TJ UK
{muratm|rgl|amd}@cmp.uea.ac.uk

Abstract

Populating an urban environment realistically with thousands of virtual humans is a challenging endeavour. Previous research into simulating the many facets of human behaviour has focused primarily on the control of an individual's movements. However, a large proportion of pedestrians in an urban environment walk in groups and this should be reflected in a simulation. This paper, therefore, proposes a model for controlling groups of pedestrians by adjusting the pedestrians' speeds.

Keywords: real-time crowd simulation, virtual pedestrian groups.

1 Introduction

The majority of real-time crowd simulations largely treat pedestrians as individual entities and do not consider simulating pedestrians in groups. Groups of pedestrians in an urban environment are created for a variety of reasons resulting in different group dynamics that should be simulated. Johnson et al. (1994), defined a group as being one of four types: primary, secondary, nested primary or nested secondary. Primary groups contain group members with primary relationships such as friendship or family ties, whereas the secondary groups are composed of group members with weaker ties.

The main contribution of this paper is a speed controller engine capable of simulating both primary and nested secondary group behaviours for pedestrians in an urban environment. The speed controller is able to keep group members together in a realistic and efficient manner. A number of surveys have been undertaken in the fields of social psychology and transportation to improve the understanding of the interactions between group members, (Willis et al. (2004)) and these are used to ensure that the resulting pedestrian simulation is realistic.

2 Related Work

To obtain a better visualization of a coherent group structure many researchers used leader follower techniques (Bayazit et al. (2002) and Loscos et al. (2003)). These local navigation approaches suffered from members of the group becoming separated and consequently the structure of the group has the potential to be lost. Recently Silveira et al. (2008) proposed a physically based group navigation technique using dynamic potential field maps. The formation is obtained by aligning the agents with a deformable template for the arrangement of the group members. However, the deformation of the template is undertaken only when attempting to navigate obstacles and therefore will not model the natural movement of groups of pedestrians in an urban setting.

3 System Overview

The presented technique for group behaviour is demonstrated by integrating it into an existing behaviour system, Haciomeroglu et al. (2007), which is capable of simulating ten thousand individuals traversing an urban environment. In the existing system, each pedestrian moves through

Figure 1: Screen shots from the real-time simulation; (a) a group consists of several primary groups, (b) several independent primary groups and (c) a mix of nested groups.

the scene by travelling along the edges of a pre-calculated navigation graph. Initially a group is assigned a destination node in the navigation graph and a path planning component is utilized to determine the routes the group members take. As the pedestrians traverse the edges of the graph, the two level steering model described in Section 4 is used.

4 Two Level Steering Module

Let the combination of steer away force from the closest possible collision and the goal vector be $\vec{DV}$, the number of nearby pedestrians be n, the current pedestrian position be Ap, the position of the nearby pedestrian i be P_i and the vector from Ap to P_i be $\vec{ApP_i}$. The other pedestrians repulsive force is calculated by using Equation 1. Finally $\vec{PV}$ and $\vec{DV}$ are averaged and the final motion vector $\vec{M}$ is determined. β in Equation 1 scales the physical repulsion forces and it is set to the value of 1.8 in order to allow people to pass more compactly.

$$\vec{PV} = \sum_{i=0}^{n} \left(\frac{-\vec{ApP_i}}{(\beta \|\vec{ApP_i}\|)^2} (1 + \vec{DV} \cdot \vec{ApP_i}) \right) \tag{1}$$

5 Speed controller stage 1: Maintaining the distance to a subgroup

This stage of the speed controller ensures that the distances between primary groups in nested secondary groups are maintained. It is achieved by allowing a non-leading primary groups' members to attempt to adjust their speed to converge on the desired gap between themselves and the subgroup in front. The members of a leading primary group adjust their speed to the subgroup behind them. The speed is adjusted using an error term ϵ_{SG}. This is calculated by determining the distance between a pedestrian and the centre of mass of the group in front of them minus the desired gap between subgroups. For this simulation the value of two metres is selected. ϵ_{SG} is a signed distance which is dependent on whether a group is in front or behind,

Initially each pedestrian, i, is assigned a $locomotionSpeed_i$, which states the speed of movement for a pedestrian given the particular mode of scene traversal such as walking or running. This value may alter during the simulation if an agent changes his mode of locomotion. In this simulation all group members have the same value for $locomotionSpeed_i$. In order to reduce the error, ϵ_{SG}, the $locomotionSpeed_i$ will need to be adjusted over successive frames of the simulation. This is achieved using Equation 2 where a new $subgroupSpeed_i$ is calculated for each pedestrian.

$$subgroupSpeed_i = locomotionSpeed_i - (\epsilon_{SG} \times locomotionSpeed_i) \tag{2}$$

Incidentally, if the pedestrian is not a member of a subgroup then their $subgroupSpeed_i$ will equal their original $locomotionSpeed_i$, since there does not exist a group in front of them to introduce an error term.

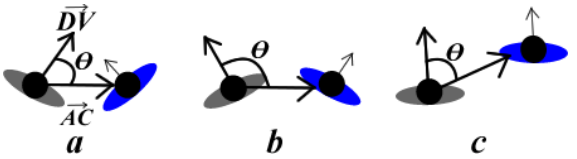


Figure 2: Both agents determine that its companion is in front in (a) behind in (b) and one infront of the other in the ideal case (c). In cases (a) and (b) ϵ indicates a zero distance.

6 Speed controller stage 2: Maintaining inter-personal distances

Pedestrians use speed controller engine stage 2 for the synchronization of their positions with their fellow primary group members. Initially a set of groups are created with each pedestrian in the group following the same path to a common destination. Every pedestrian searches for the nearest primary group member and considers it their companion during the simulation. In every simulation step, a group member determines the signed Euclidean distance, d, from its companion recorded from the previous simulation step. Let the direction vector of an agent, A, be $\vec{DV}$, position of A be Ap, its companion's position be Cp and the vector from Ap to Cp be $\vec{AC}$. If the smallest angle, computed using the dot product, between $\vec{DV}$ and $\vec{AC}$ is less than 90 degrees then his companion is considered to be in front of A otherwise it is behind. However, there are two special cases which are illustrated in Figure 2 to be considered. In Figure 2 (a) each agent determines that they are behind the other and in (b) they both determine that they are in front of the other. In these two special cases the simulation alters the agents' speeds as if the distance between them was zero. If these two special cases were not handled, then both agents would either speed up or slow down continuously and this would result in inconsistent behaviour.

Utilizing the signed distance, d, does not distinguish between whether the pedestrians are side by side (as intended) or in a queue. Therefore, the signed distance, d, is modified to promote side by side walking. Equation 3 calculates the modified distance, d', which will be the error for stage 1.

$$d' = d \times (1 + |\vec{DV} \cdot \vec{AC}|) \tag{3}$$

d' is bound between $\pm\beta$ metres and normalized by dividing it by β, (β is 2 metres in our simulation). The normalized value for d' becomes the primary group error value ϵ_{PG} that is used to control the pedestrian's speed. For example, if a pedestrian is behind his companion then the pedestrian's speed will be increased. The desired speed for an agent in the next simulation step is calculated by using Equation 4 to reduce the ϵ_{PG} error term.

$$ds_i = subgroupSpeed_i - (\epsilon_{PG} \times subgroupSpeed_i) \tag{4}$$

Where ds_i is the desired speed of the pedestrian and is defined in order for the pedestrian to stay in close proximity with their companions. However, it is not possible to set this speed directly to the pedestrians since the time passed between frames, Δt, has to be taken into account to ensure that pedestrians smoothly accelerate or decelerate. To achieve this, first an unconstrained speed, $Uspeed_i$, is determined using Equation 5. The $speed_i$ variable in Equation 5 is the current speed of the pedestrian, which is computed from the result of a speed controller calculation in the previous frame and is initialised using the $locomotionSpeed_i$.

$$Uspeed_i = speed_i + ((ds_i - speed_i) \times \Delta t) \tag{5}$$

To ensure that the pedestrians do not alter their speeds abruptly and cause visible discontinuities in their scene navigation, each pedestrian is assigned a maximum acceleration value, AR, and a maximum deceleration value, DER. These may differ between pedestrians and they store the maximum difference in speed that can occur in one second. During scene navigation a pedestrian will move from his current position to his next position along a final motion vector $\vec{M}$, as described in Section 4. The vector is not always of unit length resulting in a pedestrian's actual speed of

traversal being different from the pedestrian's speed value calculated in Equation 5. Therefore the acceleration terms are used to clamp the speed based on the actual speed the agents were moving at in the previous frame, PAS, and the actual speed for the pedestrians in the next frame, $NAS = Uspeed_i \times \left|\vec{M}\right|$.

Now the final constraint checks can be undertaken on the actual pedestrian speeds. If the difference between NAS and PAS is larger than $AR \times \Delta t$ then NAS is reduced to the maximum acceleration amount of $PAS + AR \times \Delta t$. If the difference between NAS and PAS is smaller than $DAR \times \Delta t$ then NAS is increased to the maximum de-acceleration amount of $PAS + DAR \times \Delta t$. Finally the output speed is calculated by dividing the NAS by $\left|\vec{M}\right|$ in order to make the output speed independent from the motion vector size.

In summary, every pedestrian attempts to synchronize their position with their companion and other group members. By using the speed controller engine described in this paper the pedestrians accurately synchronize their positions, allowing the pedestrians to walk side by side with their neighbours, akin to their real life counterparts.

7 Results and Conclusion

The system has been tested for 10 minutes with 2530 pedestrians including 1264 of them in groups. There were 117 nested groups and 534 single level groups. Every pedestrian in the simulation has the same $locomotionSpeed_i$ of $(1.7ms^{-1})$, however their actual speed differs because of both the steering and the speed controller engines. The single level group's and the individual's average speeds are similar $(1.55ms^{-1}$ and $1.58ms^{-1}$ respectively) but the nested groups navigate slightly more slowly $(1.51ms^{-1})$. This is natural because in nested groups the individuals will try to be in close proximity with each other and the other subgroups.

Both the inter-personal distances between members in nested groups and single level groups have been recorded. The single level group's average inter-personal distance was $0.95metres$ and the nested group's average inter-personal distance was $0.94metres$. Members in a nested group will be combined more tightly than members in a single level group, because other individuals cannot easily break a larger group apart; instead these individuals will prefer to steer away from the large oncoming group.

The overhead for the group calculations is 2.3%. The additional steering calculations are neccessary, since the number of pedestrians navigating in close proximity increases when many groups are in the environment.

References

Bayazit, O. B., Lien, J.-M., and Amato, N. M. (2002). Roadmap-based flocking for complex environments. In *PG '02: Proceedings of the 10th Pacific Conference on Computer Graphics and Applications*, page 104, Washington, DC, USA. IEEE Computer Society.

Haciomeroglu, M., Laycock, R., and Day, A. (2007). Distributing pedestrians in a virtual environment. In *Cyberworlds 2007*, pages 152–159, Hannover, Germany.

Johnson, N. R., Feinberg, W. E., and Johnston., D. M. (1994). Microstructure and panic: The impact of social bonds on individual action in collective flight from the beverly hills supper club fire. In *In Disasters, Collective Behavior, and Social Organization*.

Loscos, C., Marchal, D., and Meyer, A. (2003). Intuitive crowd behaviour in dense urban environments using local laws. page 122.

Silveira, R., Prestes, E., and Nedel, L. P. (2008). High-level path specification and group control for virtual characters in interactive virtual environmetns. In *CGI*, pages 100–107, Turkey.

Willis, A., Gjersoe, N., Havard, C., Kerridge, J., and Kukla, R. (2004). Human movement behaviour in urban spaces: implications for the design and modelling of effective pedestrian environments. In *Environment and Planning B: Planning and Design*, volume 31(6), pages 805–828.

Plausible Virtual Paper for Real-time Applications

Young-Min Kang Heng-Guang Zhang
Tongmyong University
Busan, 608-711, Korea
ymkang@tu.ac.kr

Hwan-Gue Cho
Pusan National University
Busan, 609-735, KOREA
hgcho@pusan.ac.kr

Abstract

We propose an adaptive mesh animation techniques for virtual paper simulation. The proposed method can be applied to arbitrary triangular mesh structures and efficiently produces wrinkles and creases on the paper surface with stable numerical integration and deformation-based mesh refinement.

Keywords: virtual paper, physically-based modelding

1 Introduction

Since Terzopoulos et al. (1987) simulated deformable object in computer graphics literature, Baraff and Witkin (1998); Choi and Ko (2002); Meyer et al. (2001); Volino and Magnenat-Thalmann (2001) proposed various techniques for soft object animation, and Ma and Baciu (2006) devised a method to generate seams and wrinkles for realistic appearance. Grinspun et al. (2003) introduced a discrete shell model for describing thin objects. In this method, however, adaptive mesh restructuring was not taken into account. Although the method was improved for origami simulation in Burgoon et al. (2006), their work did not consider the arbitrary crumpling with external forces in interactive applications. In this paper, we propose an adaptive and stochastic mesh reconstruction method for simulating the behavior of soft and thin virtual paper objects in an interactive application. Inextensible thin objects can be represented with stiff mass-spring models, and Baraff and Witkin (1998) proposed an implicit integration scheme for stiff differential equation. In an implicit integration scheme, the mass-spring simulation can be expressed as $\Delta\mathbf{v}^{t+h} = h(\mathbf{M} - h^2\frac{\partial \mathbf{f}_\sigma}{\partial \mathbf{x}} - h\frac{\partial \mathbf{f}_\delta}{\partial \mathbf{v}})^{-1}(\mathbf{f}_\sigma^t + \mathbf{f}_\delta^t + h^2\frac{\partial \mathbf{f}_\sigma}{\partial \mathbf{x}}\mathbf{v}^t)$ where $\mathbf{x}$, $\mathbf{v}$, $\mathbf{f}_\sigma$, and $\mathbf{f}_\delta$ are the vectors of locations, velocities, spring forces, and damping forces respectively, and $\mathbf{M}$ denotes the mass matrix. We employed an approximate integration proposed in Kang and Cho (2004). However, the resulting animation does not look like paper because the static mesh structure cannot generate any crumples on the surface. Therefore, we employed an adaptive mesh techniques which was first introduced in Kang and Cho (2008).

2 Adaptive Structure with Breakable Springs

Fig.1 illustrates the breakable spring model. The compressed spring is broken into two distinct spring edges in order to maintain the original length. Two additional springs are inserted to maintain the triangular structure, and an auxiliary spring is added to preserve the damage. The total mass cannot be changed at any condition. After a new mass m_n is inserted, the neighboring masses m_0, m_1, m_2, and m_3 are adjusted as m_0', m_1', m_2', and m_3' to preserve the total mass. Let us consider two extreme cases: (a) a new particle is placed exactly on a neighboring mass point i, and (b) a neighboring mass point i is extremely far from the new mass. In the first case, the mass of the newly added particle should be $m_i/2$. In the second case, adding the particle should not decrease m_i. With this consideration, we adjust masses as follows:

$$m_l' = \left(\frac{d_l}{2\sum_{k\in N} d_k} + \frac{1}{2}\right) m_l, \quad m_n = \frac{\sum_{k\in N} m_k}{2} - \frac{\sum_{k\in N} d_k m_k}{2\sum_{k\in N} d_k} \tag{1}$$

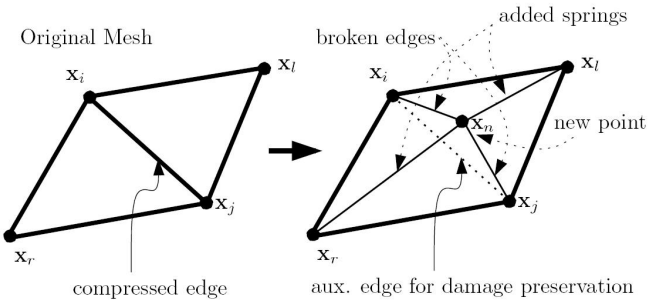


Figure 1: Breakable spring model

where d_k is the distance from the new mass to another mass point k in the undeformed configuration, and N is the set of the neighboring masses.

The momentum should be also preserved by specifying the velocity of the new mass point as follows:

$$\mathbf{v}_n = (\sum_{k \in N} m_k \mathbf{v}_k - \sum_{k \in N} m'_k \mathbf{v}_k)/m_n \tag{2}$$

3 Stochastic Edge Break Model

In our approach, the fracture of an edge occurs stochastically. Probability of the fracture is proportional to the contraction ratio $(1 - l^t_{ij}/l_{ij})$ where l^0_{ij} and l^t_{ij} denote the rest length and the current length of the spring between $\mathbf{x}_i$ and $\mathbf{x}_j$ respectively. To take the horizontal and vertical curvatures into account, we consider two neighbor vertices $\mathbf{x}_r$ and $\mathbf{x}_l$. The distance between the neighbors is l_{rl}. We denote the normal vectors at the mass-point $\mathbf{x}_i$, $\mathbf{x}_j$, $\mathbf{x}_r$, $\mathbf{x}_l$ as $\mathbf{n}_i$, $\mathbf{n}_j$, $\mathbf{n}_r$, $\mathbf{n}_l$. The curvature along the edge increases the probability of the fracture while the curvature across the edge decreases it. Therefore, we can model the fracture probability to be proportional to $(1 - n_{ij})/2$ and $(1 + n_{rl})/2$ where $\mathbf{n}_{ij} = \mathbf{n}_i \cdot n_j$. In order to make it possible for a flat object to be folded, we employed a control parameter ϕ to scale the dot product of the normal vectors. Based on these observations, the probability of fracture was actually computed as follows:

$$\mathbf{P}_{ij} = \frac{1}{4l^0_{ij}}(l^0_{ij} - l^t_{ij})(1 - \phi \cdot n_{ij})(1 + \phi \cdot n_{rl}) \tag{3}$$

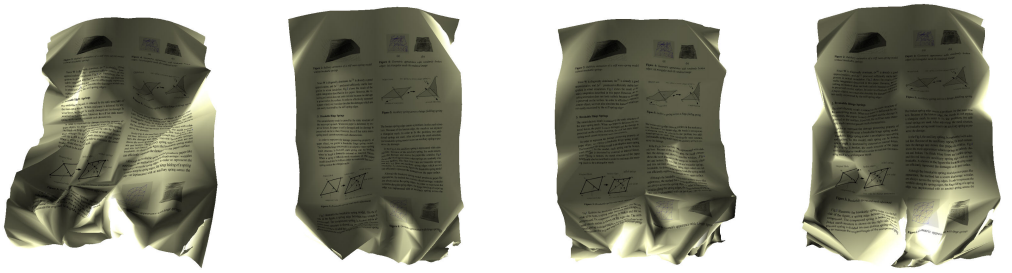
We adjust the location of the new mass-point along the surface normal vector. The magnitude of the adjustment ϑ can be easily computed as follows:

$$\vartheta = \frac{\psi}{2}\sqrt{(l^0_{ij})^2 - (l^t_{ij})^2} \tag{4}$$

where ψ is a parameter that controls length of broken edges.

4 Bending Energy based Edge Recovery

In some cases, some broken edges have to be recovered. When an edge e_{ij} is broken into two different springs e_{in} and e_{nj}, the original spring e_{ij} becomes an unbreakable auxiliary spring, i.e., $e_{ij \cdot breakable} = false$. The unbreakable auxiliary spring contains pointers to newly inserted vertex $\mathbf{x}_n$, and newly added edges e_{in}, e_{nj}, e_{rn}, and e_{nl}. When we need to remove the crumple across the broken edge e_{ij}, we can simply change the breakability property of the edge, i.e., $e_{ij \cdot breakable} \leftarrow true$, and remove one vertex $\mathbf{x}_n$ and four edges, e_{in}, e_{nj}, e_{rn}, and e_{nl}. In a more



(a) $\phi = 0.0, \psi = 1.0$ (b) $\phi = 0.7, \psi = 1.0$ (c) $\phi = 0.5, \psi = 0.0$ (d) $\phi = 0.5, \psi = 0.7$

Figure 2: Effect of control parameter ϕ and ψ: (a) $\phi = 0.0, \psi = 1.0$ (b) $\phi = 0.7, \psi = 1.0$ (c) $\phi = 0.5, \psi = 0.0$ (d) $\phi = 0.5, \psi = 0.7$

complicated situation, the broken edge e_{ij} is broken into e_{in} and e_{nj}, and the edge e_{in} or e_{nj} can be broken again. In such a case, the edge e_{ij} cannot be recovered immediately. In the proposed method, the recovery is applied to the edges which have been broken only once. Although we can recover the broken edges into an original edge, there is no criteria to determine which edges to be recovered. In our method, we exploited the bending energy of a broken edge to decide whether it would be recovered or not. A broken edge produces a crease line on the surface. The bending of the crease line recovers the broken edge. The bending energy of an edge can be conveniently computed with the model described in Grinspun et al. (2003). In their model, the bending energy $\mathbf{W}_b(e)$ of an edge e was computed as $(\theta_e - \bar{\theta_e})^2||\bar{e}||$ where θ_e and $\bar{\theta_e}$ denote the corresponding complements of the dihedral angle of the edge e measured in the deformed and undeformed configuration respectively, and $||\bar{e}||$ is the length of the edge e. In fact, the angle θ_e can be easily computed by measuring the angle between the normal vectors of two triangles incident to the edge. However, our model cannot use this angle. In order to efficiently compute the bending energy of the broken edge, we simply exploited the normal vectors at the left and right neighbors. In our model, the bending energy of a broken edge $\mathbf{W}_b(e)$ was computed as follows:

$$\mathbf{W}_b(e_{ij}) = (1 - n_{rl})/2 \tag{5}$$

where r and l denote the right and the left neighbor vertices of the edge e_{ij} respectively, and $\mathbf{n}_r$ and $\mathbf{n}_l$ are the normal vectors at those vertices.

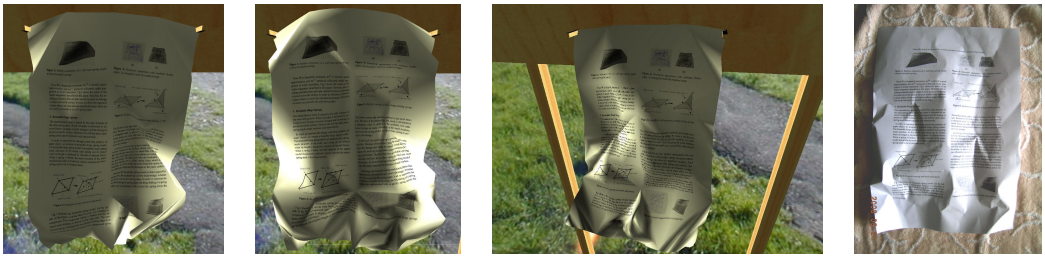
If $\mathbf{W}_b$ is larger than a given threshold ϵ, the edge is selected to be recovered. It is obvious that ϵ controls the tendency of the edge recovery. When a vertex is removed, the masses and the velocities of neighboring vertices should be adjusted to conserve the total mass and the linear momentum as follows:

$$m_i' = m_i + \frac{(\sum_{k\in N} d_k - d_i)m_n}{(|N| - 1)\sum_{k\in N} d_k}, \qquad \mathbf{v}_i' = \mathbf{v}_i + \frac{(\sum_{k\in N} m_k - m_i)(m_n\mathbf{v}_n)}{(|N| - 1)\sum_{k\in N} m_k} \tag{6}$$

where $|N|$ denotes the number of linked neighbors of removed mass point.

5 Experiments

Fig.2 shows the effect of the control parameters ϕ and ψ. As shown in the figure, a larger ϕ value generates stiffer paper appearance. The parameter ψ could be successfully used for controlling the crumples on the paper surface. The bending energy based strategy for the recovery of broken edges successfully works in an actual interactive animation as shown in Fig.3 (a) and (b). With a small ϵ, a virtual paper object easily recovers its broken edge. On the other hand, a large ϵ prevents the model from frequently recovering its broken edges, and it produces a crumpled surface. Fig.3 (c) and (d) compares the virtual paper generated with the proposed method and a real paper object. As shown in the figure, the proposed method plausibly reproduces the appearance of the paper object.



(a) $\epsilon = 0.1$ (b) $\epsilon = 1.5$ (c) virtual paper (d) real paper

Figure 3: The effect of ϵ and comparison with real paper: (a) $\epsilon = 0.1$, (b) $\epsilon = 1.5$, (c) a virtual paper produced with the proposed method, and (d) photo of real paper

6 Conclusion

An intuitive and efficient method for simulating virtual paper in an interactive application was proposed. The proposed method modeled the virtual paper as highly damped stiff mass-spring with breakable edges, and a stochastic edge fracture and recovery models were applied to adaptively change the mesh structure. The mass and the momentum of the adaptive mesh were preserved regardless of the structure. The experimental result shows the proposed method can efficiently control the properties of virtual paper and the result can be interactively animated in realtime applications.

Acknowledgment

This research was supported by Ministry of Knowledge and Economy, Republic of Korea, under the ITRC (Information Technology Research Center) support program supervised by IITA(Institute for Information Technology Advancement). (IITA-2009-C1090-0901-0004).

References

Baraff, D. and Witkin, A. (1998). Large steps in cloth simulation. *Proceedings of SIGGRAPH 98*, pages 43–54.

Burgoon, R., Wood, Z. J., and Grinspun, E. (2006). Discrete shells origami. In *Computers and Their Applications*, pages 180–187.

Choi, K.-J. and Ko, H.-S. (2002). Stable but responsive cloth. *ACM Transactions on Graphics: Proceedings of SIGGRAPH 2002*, pages 604–611.

Grinspun, E., Hirani, A., Desbrun, M., and Schröder, P. (2003). Discrete Shells. In *ACM SIGGRAPH / Eurographics Symposium on Computer Animation*, pages 62–67.

Kang, Y.-M. and Cho, H.-G. (2004). Real-time animation of complex virtual cloth with physical plausibility and numerical stability. *Presence - Teleoperators and Virtual Environments*, 13(6):668–680.

Kang, Y.-M. and Cho, H.-G. (2008). A simple and effective model for interactive paper folding. In *Poster Proceedings of Pacific Graphics 2008*.

Ma, L. and Baciu, J. H. G. (2006). Generating seams and wrinkles for virtual clothing. *Proceedings of ACM International Conference on Virtual Reality Continuum and Its Applications*, pages 14–17.

Meyer, M., Debunne, G., Desbrun, M., and Bar, A. H. (2001). Interactive animation of cloth-like objects in virtual reality. *The Journal of Visulalization and Computer Animation*, 12:1–12.

Terzopoulos, D., Platt, J., Barr, A., and Fleischer, K. (1987). Elastically deformable models. *Computer Graphics (Proceedings of SIGGRAPH 87)*, 21(4):205–214.

Volino, P. and Magnenat-Thalmann, N. (2001). Comparing efficiency of integration methods for cloth simulation. *Proc. of Computer Graphics International 2001*, pages 265–272.

Imaginary Wall Model for Efficient Animation of Wheeled Vehicles in Racing Games

Young-Min Kang
Tongmyong University
Busan, 608-711, Korea
ymkang@tu.ac.kr

Hwan-Gue Cho
Pusan National University
Busan, 609-735, KOREA
hgcho@pusan.ac.kr

Abstract

Racing game requires plausible physics model that can be simulated in realtime. We propose an efficient and effective "imaginary wall" model for racing games. The method can be easily implemented because of the simplicity of the physical model used, and the result of the simulation is realistic enough for the racing games.

Keywords: racing game, physically-based modelling, realtime animation, impulse model

1 Introduction

Hung and Orin (2001); Shiang-Lung Koo and Tomizuka (2007) proposed dynamic models for wheeled vehicles in robotics and automation literature. de Wit and Horowitz (1999); Claeys et al. (2001) also proposed sophisticated models for simulating the tire/road friction. However, those methods are too complex to be easily implemented in game applications. The tire simulation in realtime is considered a difficult problem so that semi-empirical models are usually employed as in Deák (1999). However, the parameters for the industry standard models such as *Model of Pacejka* are unfortunately too complex to be modified intuitively. To avoid the expensive cost, a simple method that computes the angular velocity of a wheeled vehicle has been widely used as in Monster (1993). Let l be the distance between the front and the rear wheels. The angle between the direction of the front wheel and that of the vehicle is δ. The vehicle rotates about a single pivot point $\mathbf{c}$. Let C and r be respectively the circumference and the radius of the circular path of the front wheel. It is obvious that C is $2\pi r$ where r is $l/\sin\delta$. The time required for a vehicle with velocity $\mathbf{v}$ to complete the rotation is $2\pi r/|\mathbf{v}|$. Therefore, the angular velocity ω can be computed as $|\mathbf{v}|/r$. However, it is not actually easy to compute the radius because it can be extremely large when δ is very small. In other words, it will be the most difficult for the simple method to deal with the most usual case where the vehicle is moving *almost* straight.

2 Simulating Tire Friction with Impulse-based Contact Force

We propose an impulse-based method that introduces *imaginary wall model*. In our model, the wheels contact with ground and imaginary walls, and low friction is applied to the wheel direction so that the vehicle can be easily accelerated or decelerated along the driving path. When a wheel is, however, sliding aside from the driving path, strong frictional force should be produced to immediately hold the wheels. This kind of strong forces can be considered impulses from the imaginary walls. Let the masses of the object a and b be m_a and m_b respectively. The objects are colliding at $\mathbf{p}_c$, and $\mathbf{x}_a$ and $\mathbf{x}_b$ denote the mass centers. The vector from $\mathbf{x}_a$ to the collision point is denoted as $\mathbf{r}_a$, and $\mathbf{r}_b$ is also defined similarly. $\hat{\mathbf{n}}$ is the collision normal. The velocities of the objects at the collision point are denoted as $\mathbf{v}_a^{\mathbf{p}_c}$ and $\mathbf{v}_b^{\mathbf{p}_c}$. Baraff (1994) and Mirtich and Canny (1995) proposed methods for efficient and effective computation of impulses between non-penetrating rigid bodies. Let $\mathbf{K}_a$ denote $\mathbf{E}/m_a + \mathbf{r}_a^{*T}\mathbf{I}_a^{-1}\mathbf{r}_a^*$ where $\mathbf{E}$ is an identity matrix, $\mathbf{I}_a$ is the inertia tensor of object a, and $\mathbf{r}^*$ denotes the cross product matrix of the

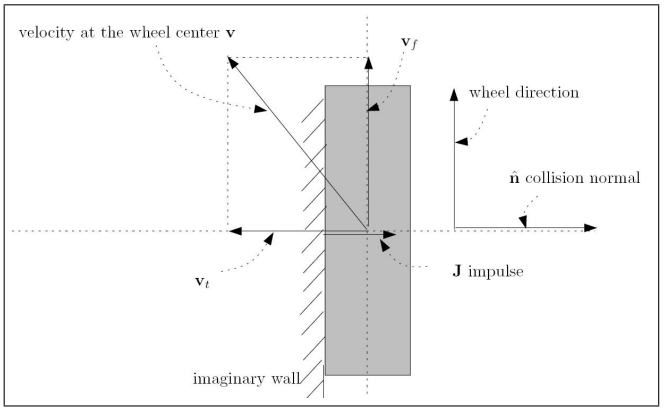


Figure 1: Impulse model with an imaginary wall

vector $\mathbf{r}$. $\mathbf{K}_b$ is defined similarly. The impulse $\mathbf{J}_b$ applied to object b can then be computed as $\mathbf{J}_b = \{-(1+\epsilon)(\mathbf{v}_b^{\mathbf{P}_c} - \mathbf{v}_a^{\mathbf{P}_c}) \cdot \hat{\mathbf{n}}/(\hat{\mathbf{n}}^T(\mathbf{K}_a + \mathbf{K}_b)\hat{\mathbf{n}})\}\hat{\mathbf{n}}$ where ϵ is the coefficient of restitution. Fig.1 shows the visual concept of imaginary wall model. The wheel moves forward and backward along the imaginary wall, and the imaginary wall continuously change its direction according to the wheel direction. The only role of the imaginary wall is to produce impulses that prevent a wheel from penetrating the wall. The velocity at the wheel center $\mathbf{v}$ can be decomposed into two parts: the velocity along the wheel direction $\mathbf{v}_f$ and the perpendicular part $\mathbf{v}_t$. The relative velocity between the contacting wall and the wheel is simply $\mathbf{v}_t$. Therefore, the direction of the impulse from the wall is $-\mathbf{v}_t/|\mathbf{v}_t|$. Because the wheel should not rebound from the wall and the imaginary wall is a static object, the restitution coefficient ϵ and the $\mathbf{K}$ matrix of the wall should be 0 and *zero* matrix respectively. Therefore, the impulse $\mathbf{J}$ from the imaginary wall can be computed as follows:

$$\mathbf{J} = \left(\frac{|\mathbf{v}_t|}{\hat{\mathbf{n}}^{\mathbf{T}}\mathbf{K}_w\hat{\mathbf{n}}}\right)\hat{\mathbf{n}} = \left(\frac{-\mathbf{v}\cdot\hat{\mathbf{n}}}{(\mathbf{K}_w\hat{\mathbf{n}})\cdot\hat{\mathbf{n}}}\right)\hat{\mathbf{n}} \tag{1}$$

where $\mathbf{K}_w$ is the $\mathbf{K}$ matrix at the center of the wheel.

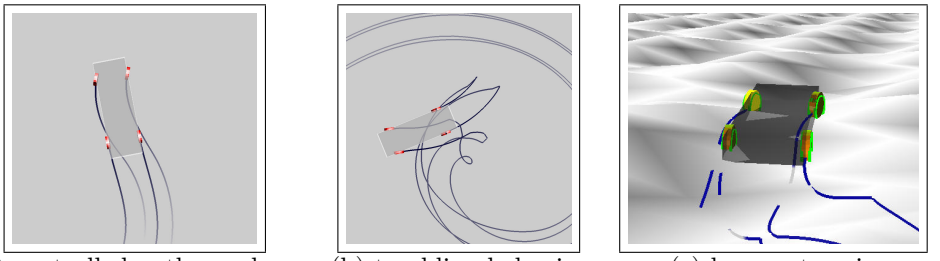
By employing the imaginary walls, the driving path can be perfectly and physically controlled as shown in Fig.2 (a). Only the wheels placed on the ground are affected by the imaginary walls so that our method is expressive enough to reproduce the realistic animation of a vehicle that trembles and tumbles as shown in Fig.2 (b), and moves on a bumpy terrain as shown in Fig.2 (c).

3 Shock-absorbing Imaginary Wall Model for Side Slip Simulation

It is often the case that a high-speed vehicle slips aside toward the outward direction of the turning circle. The side-slip is essential not only for the reality of racing simulation but also for the amusement of the game. The actual side-slip occurs when the frictional forces between tires and wheels are not strong enough. However, the proposed method does not use frictional model for the wheel dynamics. To enable the side-slip, the imaginary walls should allow wheels to penetrate them. The soft wall model can be modeled with a negative restitution coefficient. In order for a intuitive modeling, we employed a side-slip control parameter μ which ranges from 0 to 1. If the parameter μ is 0, the wheels can freely penetrate imaginary walls. The imaginary wall model with penetration can be formulated as follows:

$$\mathbf{J} = \left(\frac{-(1+\epsilon)\mathbf{v}\cdot\hat{\mathbf{n}}}{(\mathbf{K}_w\hat{\mathbf{n}})\cdot\hat{\mathbf{n}}}\right)\hat{\mathbf{n}} = \left(\frac{-\mu\mathbf{v}\cdot\hat{\mathbf{n}}}{(\mathbf{K}_w\hat{\mathbf{n}})\cdot\hat{\mathbf{n}}}\right)\hat{\mathbf{n}} \tag{2}$$

μ was defined to be $\max\{\mu_{max}\,(1-|\mathbf{v}|/v_\theta)\,,0\}$ where μ_{max} is the maximum value for the parameter μ, and v_θ is a specifiable threshold speed where μ parameter becomes zero. If the speed of the vehicle exceeds the limit, μ is enforced to be zero. When we need a vehicle that drifts more easily, we have only to decrease the parameter v_θ.



(a) controlled path on plane (b) tumbling behavior (c) bumpy terrain

Figure 2: The control and expressiveness of the proposed method: (a) a vehicle with moderate speed, (b) a high-speed vehicle with sharp turn, (c) a vehicle on a bumpy terrain

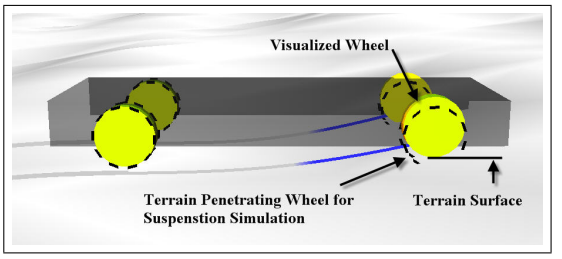


Figure 3: Suspension Simulation: the dashed black circles represent the actual dynamic wheels allowed to penetrate the terrain. Users can observe only the solid wheels lifted on the surface to show the suspension effect.

4 Impulse-based Suspension Simulation

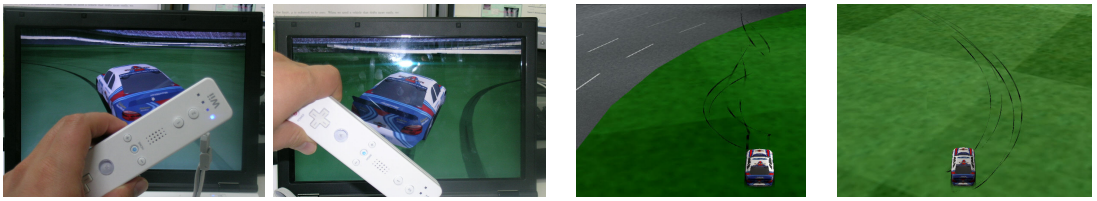
Although the imaginary wall model can be successfully employed for lightweight dynamics for steering a wheeled vehicle, the model does not simulate the suspension mechanism. Because the suspension is one of the most important mechanisms of usual vehicles, plausible racing game requires proper suspension model. The suspension is actually implemented with shock-absorbing stiff springs. However, the stiff spring model often causes stability problem. For the suspension simulation, we also used an impulse model. The suspension should be considered only when a wheel collides with ground. While the usual impulse model prevents wheels from penetrating the terrain, our impulse-based suspension model allows the wheels to penetrate the terrain surface. Let $\mathbf{p}_w$ be the contact position of a wheel that touches the ground, and $\mathbf{p}_g$ be the contact position of the ground. $\wp_\tau$ and $\wp_p$ denote the maximum penetration allowed and the actual penetration(i.e. $|\mathbf{p}_g - \mathbf{p}_w|$ respectively. The penetration ratio $\wp$ can be calculated as $\wp_p/\wp_\tau$. Let $\mathbf{J}_g$ be the impulse from the ground to wheel. We scale the impulse according to the magnitude of the impulse $|\mathbf{J}_g|$ and the penetration ratio $\wp$, and the scaled impulse is applied to the wheels. The scaled impulse $\mathbf{J}_s$ is computed as follows:

$$\mathbf{J}_s = \frac{1}{2}(e^{-\xi|\mathbf{J}_g|} + \wp)\mathbf{J}_g \tag{3}$$

Fig.3 shows the suspension effect of our model. The dashed black circle shows the actual wheel that penetrates the ground surface while the yellow solid wheel is lifted on the ground to provide plausible visualization to users.

5 Experimental Results

The proposed method produced physically plausible animation of wheeled vehicles in interactive applications such as game. Fig.4 (a) demonstrates realtime performance of the proposed method. The driving path was plausibly controlled in an experimental game applications. Since the impulse model makes it possible for the wheels to be independently simulated, trembling or tumbling



(a) Realtime Performance (b) Tumbling (c) Side slip

Figure 4: The effective path control of the proposed method: (a) realtime control (b) tumbling vehicle(c) side-slip of high speed vehicle

motions of vehicles can also be easily expressed as shown in Fig.4 (b), which cannot be simulated with kinematic approaches. The shock absorbing imaginary wall model for side-slip simulation described in Eq.2 was also tested, and the model produced realistic motion of high-speed vehicles as shown in Fig.4 (c).

6 Conclusion

A realtime approach to the vehicle wheel simulation was proposed. The method is plausible enough for racing game application because the behavior of each wheel is physically simulated with impulse. Moreover, the simplicity of the proposed method enables experienced game developers to easily implement a racing game in a short time. Because the proposed method efficiently generates plausible results, the method can be successfully employed for developing high quality racing game running on CPU-limited computing environments.

Acknowledgment

This research was supported by Ministry of Knowledge and Economy, Republic of Korea, under the ITRC support program supervised by IITA (IITA-2009-C1090-0901-0004).

References

Baraff, D. (1994). Fast contact force computation for nonpenetrating rigid bodies. In *Proceedings of ACM SIGGRAPH 1994*, pages 23–34.

Claeys, X., Yi, J., Alvarez, L., Horowitz, R., and de Wit, C. C. (2001). A dynamic tire/road friction model for 3d vehicle control and simulation. In *Proceedings of IEEE Transportation Systems Conference*, pages 483–488.

de Wit, C. C. and Horowitz, R. (1999). Observers for tire/road contact friction using only wheel angular velocity information. In *Proceedings of the 38th Conference on Decision and Control*, pages 3932–3937.

Deák, S. (1999). Dynamic simulation in a driving simulator game. In *Proceedings of The 7th Central European Seminar on Computer Graphics*, pages 3932–3937.

Hung, M.-H. and Orin, D. E. (2001). Dynamic simulation of actively-coordinated wheeled vehicle systems on uneven terrain. In *Proceedings of the 2001 IEEE International Conference on Robotics and Automation*, pages 779–786.

Mirtich, B. and Canny, J. F. (1995). Impulse-based simulation of rigid bodies. In *Symposium on Interactive 3D Graphics*, pages 181–188.

Monster, M. (1993). Car physics for games. *http://home.planet.nl/ monstrous.*

Shiang-Lung Koo, H.-S. T. and Tomizuka, M. (2007). Impact of tire compliance behavior to vehicle longitudinal dynamics and control. In *Proceedings of the 2007 American Control Conference*, pages 5736–5741.

Motion analysis to improve virtual motion plausibility

Barbara Mazzarino and Maurizio Mancini
Infomus Lab, DIST
Università degli Studi di Genova, Italy
[barbara.mazzarino|maurizio.mancini]@dist.unige.it

Abstract

By understanding the mechanisms of human-human communication, developers are trying to better address expressive communication in virtual subjects, such as agents, and consequently to improve Human Computer Interaction. The work presented here is focused on the evaluation of human motion features. The algorithms we present can be applied to virtual humanoids in order to determine if the expressive information codified in the synthesized motion are comparable to the motion of real humans acting with the same intent.

Keywords: Motion Analysis, Virtual humanoid plausibility

1 Introduction

In the last few years one of the key issues of the Human Computer Interaction framework is the design and creation of a new type of interfaces, able to adapt HCI to human-human communication capabilities. In this direction the ability of computers to detect and synthetize emotional state is becoming particularly relevant, that is, computers must be equipped with interfaces able to establish an *Affect Sensitive* interaction with the user, in the sense defined by Zeng et al. 2009.

To synthesize correctly the emotional expressive information to be conveyed by virtual humanoids motion, it is necessary to study the mechanism used by humans to use and read this high level content. In human-human interaction the communication of emotional expressive content takes different channels of communication that includes also full-body motion.

In this paper we present a method for measuring two motion features: impulsivity and smoothness. Impulsivity indicates wether or not movement presents sudden and abrupt changes in energy. For example, an unexpected danger like a car approaching a person crossing the street may induce a sudden and impulsive reaction in the person movement, due to the emotion of fear/terror. Smoothness identifies the continuity/fluency of movement. Happy and relaxed persons usually communicate their state by producing body movements that are very fluent and continuous. Instead, angry and tensed persons perform quick and short body movements exhibiting abrupt changes in limbs curvature/speed.

2 Impulsivity

2.1 Definition

In our context *Impulsivity* can be seen as a "a short time perturbation of the subject motion state". Referring to physics we focus on the Impulsive Momentum Theorem, where an impulse can be considered as a variation in the momentum of an object to which an external force is applied. If Force and Mass are considered as constants then the following rule is true: $I = F \triangle t = m \triangle v = \triangle p$ and knowing the starting and the ending velocities: $\triangle p = m(vf - vi)$.

The underlying concept of this theorem considers the impulse as a variation of the momentum, that is, *a perturbation of the state*. In psychology Impulsivity is an important aspect to consider for evaluating some specific pathologies. In this area we found the following definition: "actions that are poorly conceived, prematurely expressed, unduly risky, or inappropriate to the situation and that often result in undesirable outcome ". From this definition we can observe that an impulsive behavior or gesture lacks of premeditation, that is, it is performed *without a significant preparation phase*. In the work of Heiser et al. 2004

on Hyperkinetic Disorders we found a characterization of the impulsive motion that "was 3.4 times as far, covered a 3.8-fold greater area, and had a more linear and less complex movement pattern". Heiser concludes that an impulsive motion can be read as *linear, without complex pattern*. From the work of Wilson et al. 1996 on the structure of the natural gesture we highlight analogies between impulsive gestures and beat gestures, characterized by short duration and high magnitude. The main research that helps us in our definition is the Theory of Effort by Laban and Lawrence 1947, that identified four Effort Qualities in human movement: Flow, Weight, Time, and Space.
By integrating all the approaches found in our overview we obtained the following definition of impulsive gestures: gestures performed without premeditation, i.e. looking to the motion phases with a very short or absent preparation phase; gestures performed with a simple pattern, i.e. simple shape performed; gestures characterized by short duration and high magnitude; gestures performed with Time = sudden and Flow =free in Laban terms.

2.2 ALGORITHM

The algorithm for the automatic evaluation of impulsivity, has been implemented in the EyesWeb software platform (www.eyesweb.org) using the EyesWeb Expressive Gesture Library Camurri et al. 2004 to extract motion features as for example energy, called Quantity of Motion (QoM). To identify the gesture duration we use motion segmentation based on the motion bells identified by thresholding the QoM. In this work we set the threshold of the normalized energy equal to 0.02, since we work in a controlled environment. A motion with energy higher then such threshold can be considered with "high magnitude". To respect the characteristic of "fast"execution, we fixed (considering also the state of the art in this context) a time duration $dt = 0.45sec$ to discriminate the impulsive gesture, with an attack phase of $dta \leq 0.15sec$. In Figure 1 there is an example of Impulsive gesture in term of QoM and Time duration.

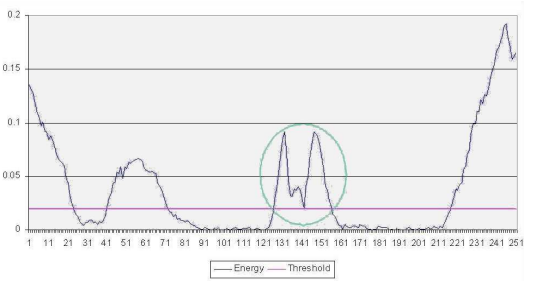


Figure 1: This is the graph of the energy motion feature with respect to the threshold value. In the green circle is highlighted the motion bell related to the impulsive gesture.

From empirical considerations, to rapidly modify actual motion, it is necessary to rapidly modify posture and in particular to modify the body occupation of space, that is Eyesweb corresponds to the Contraction Index motion cue. The calculation of this cue can be performed during the gesture execution, without introduce additional delay in the final evaluation.
The algorithm can then be written as:

let $\triangle t = 0.45sec$ and let gesture $threshold = 0.02$;
if $(energy \geq threshold)$ then evaluate the GestureTimeDuration dt;
If $dt \geq 0$ and $dt \leq \triangle t$ then $ImpulsivityIndex = \triangle CI/dt$;

3 SMOOTHNESS

3.1 DEFINITION

From English dictionary, *smooth*: "free from or proceeding without abrupt curves, bends, etc.; allowing or having an even, uninterrupted movement or flow". In mathematics, smoothness is linked to the speed of variation, that is, a smooth function is a function that varies "slowly" in time; more precisely, smooth functions are those that have derivatives of all orders. In music smoothness corresponds to articulation in

music performance, as for example stated in DiPaola and Arya 2004. In psychopathology, smoothness of human movement could allow one to diagnose psychological disorders, for example schizophrenia: patients movements are described "staccato-like, jerky and angular", while they become "smooth and rounded" after successful therapy, as reported in Wallbott 1989. Gallaher 1992 refers to smooth and fluid movements: "an individual high on this factor has a smooth voice, flowing speech and gestures, and a fluid walk; such a person would appear graceful and coordinated ". Smooth/fluid movements are often associated with slow, sluggish and lethargic movements, in contrast with large and energetic body movement. Slowness in movements corresponds to the definition of smooth functions as slowly varying functions in mathematics.

Wallbott measured displacement of hand in psychiatric patients behavior and found four main movement characteristics: space, which describes the extension of movement; hastiness, which is related to speed and acceleration; intensity, which describes the energy of a movement; fluency-course, which is related to the quality between the beginning and the end of a movement. Wallbott states that smoothness is a possible value for the fluency-course characteristics, thus demonstrating the importance of such parameter in describing movement quality.

3.2 ALGORITHM

Research work reported by Todorov and Jordan 1998 demonstrates a correspondence between (i) smooth trajectories performed by human arms, (ii) the minimization of the third-order derivative of the hand position (called *jerk* in physics) and (iii) the correlation between hand trajectory curvature and velocity[1]. In our work we use an approach similar to (iii) to check whether a trajectory is smooth or not by computing the trajectory curvature and velocity. Other researchers like Sezgin et al. 2006 investigated the same characteristics in sketch recognition algorithms, to determine the corners of a curve, that is, the points in which curvature is high and velocity is low.

The input to our system consists of video frames frames at 60 Hz showing a moving person. During the preprocessing phase, for each video frame the system extracts the 2D position (x,y) of the barycenter of a green marker placed on the person right or left hand and stores it in a buffer consisting of 60 samples, while the oldest element of the buffer is discarded. The hand position buffer is then provided as input to the smoothness computation algorithm: for every sample (x,y) in the buffer we compute curvature k and velocity v as:

$$k = \left| \frac{x'y'' - y'x''}{(x'^2 + y'^2)^{\frac{3}{2}}} \right| \quad v = \sqrt{x'^2 + y'^2} \tag{1}$$

Where x', y', x'' and y'' are the first and second order derivatives of x and y. To compute them from the buffer of samples (x,y) we apply a Savitzky-Golay filter (Savitzky and Golay 1964). This type of filter provide as output both the filtered signal and an approximation of the $n-th$ order smoothed derivatives.

Then we compute the correlation between trajectory curvature and velocity. However, k and v are computed over a "short" time window, so we could approximate the covariance $\sigma_{log(k),log(v)}$ with 1, as the k and v variate (or not) approximately at the same time. In this way the correlation coefficient can be computed by the following formula:

$$\rho' = \frac{1}{\sigma_{log(k)}\sigma_{log(v)}} \tag{2}$$

We use ρ' to determine the amount of human hand trajectory *Smoothness Index*, as shown in Figure 2. The upper part of the Figure shows the trajectories of the performer hand: a continuous smooth circle on the left and a square shape performed with movements exhibiting sharp direction variations. The bottom part reports the information provided as output by our system EyesWeb in realtime: the trajectory as it was detected by the program and the trajectory Smoothness Index computed as explained above. As shown, the index is high for the circular smooth trajectory, while it drops to very low values for the square shape. These empirical results demonstrate show that our algorithm is able to correctly distinguish between smooth and angular movements, even if further refinement tests should be performed in future.

[1]M. Mancini would like to acknowledge D. Glowinski at InfoMus for his collaboration on developing the Smoothness Algorithm.

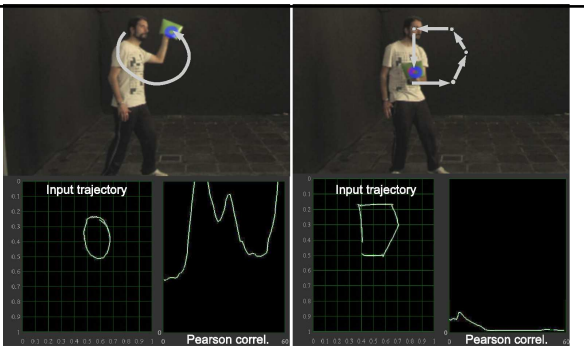


Figure 2: Round and a square trajectories: Smoothness Index is high when computed on the round smooth trajectory (left) and is approximately zero when computed on the square non-continuous one (right).

4 CONCLUSION

The main focus of the presented work is on motion analysis. We highlight how humans use motion to communicate expressive information with the aim of implementing them in virtual agents. The algorithms proposed are related to the evaluation of two motion feature in the real human full-body motion: impulsivity and smoothness. Future work includes the validation of the proposed methods using our video corpus of recorded motions performed by professional dancers, martial arts experts and students. The aim is to compare the data computed by our algorithms with subjects rates, and to refine the automatic features extraction. After this validation, the algorithms will be applied to the motion of virtual humanoid since they work on video streams in real-time.

Performed and future work are addressed in the framework of the EUICT Project SAME (www.sameproject.eu) and the EU Culture 2007 project CoMeDiA (www.comedia.eu.org).

REFERENCES

Camurri, A., Mazzarino, B., and Volpe, G. (2004). Analysis of expressive gesture: The eyes web expressive gesture processing library. *LECTURE NOTES IN COMPUTER SCIENCE*.

DiPaola, S. and Arya, A. (2004). Affective communication remapping in musicface system. In *Electronic Imaging & Visual Arts*.

Gallaher, P. E. (1992). Individual differences in nonverbal behavior: Dimensions of style. *Journal of Personality and Social Psychology*, 63(1):133–145.

Heiser, P., Frey, J., Smidt, J., Sommerlad, C., M.Wehmeier, P., J.Hebebrand, and Remschmidt, H. (2004). Objective measurement of hyperactivity, impulsivity, and inattention in children with hyperkinetic disorders before and after treatment with methylphenidate. *European Child & Adolescent Psychiatry*, 13(2):100–104.

Laban, R. and Lawrence, F. C. (1947). *Effort*. Macdonald & Evans, USA.

Savitzky, A. and Golay, M. J. E. (1964). Smoothing and differentiation of data by simplified least squares procedures. *Analytical chemistry*, 36(8):1627–1639.

Sezgin, T., Stahovich, T., and Davis, R. (2006). Sketch based interfaces: early processing for sketch understanding. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Courses*, page 22, New York, NY, USA. ACM.

Todorov, E. and Jordan, M. I. (1998). Smoothness maximization along a predefined path accurately predicts the speed profiles of complex arm movements. *Journal of Neurophysiology*, 80(2):696–714.

Wallbott, H. G. (1989). Movement quality changes in psychopathological disorders. *Normalities and Abnormalities in Human Movement. Medicine and Sport Science*, 29:128–146.

Wilson, A., Bobick, A., and Cassell, J. (1996). Recovering the temporal structure of natural gesture. In *Proc. of the Second Intern. Conf. on Automatic Face and Gesture Recognition*.

Zeng, Z., Pantic, M., Roisman, G., and Huang, T. (2009). A survey of affect recognition methods: audio, visual and spontaneous expressions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(1).

Generating Concise Rules for Retrieving Human Motions from Large Datasets

Tomohiko Mukai and Ken-ichi Wakisaka and Shigeru Kuriyama
Toyohashi University of Technology
1-1 Hibarigaoka, Tenpaku-cho, Toyohashi, 441-8580 Aichi, Japan
{mukai@|k_wakisaka@val.|kuriyama@}ics.tut.ac.jp

Abstract

This paper proposes a method for retrieving human motion data with concise retrieval rules based on the spatio-temporal features of motion appearance. Our method first converts motion clip into a form of clausal language that represents geometrical relations between body parts and their temporal relationship. A retrieval rule is then learned from the set of manually classified examples using inductive logic programming (ILP). ILP automatically discovers the essential rule in the same clausal form with a user-defined hypothesis-testing procedure. All motions are indexed using this clausal language, and the desired clips are retrieved by subsequence matching using the rule. Such rule-based retrieval offers reasonable performance and the rule can be intuitively edited in the same language form.

Keywords: motion capture, motion indexing, motion appearance feature, inductive logic

1 Introduction

Automated retrieval methods of human motion data have been proposed using some feature analysis techniques. However, existing methods have one major problem with query formulation. The numerical methods (Chiu et al., 2004) use a short motion clip as a query, but the similarity measure between motions should be manually defined with fine parameter tunings. The learning-based methods (Arikan et al., 2003) implicitly obtain a motion classifier which can not be modified after the learning. The template-based methods (Müller and Röder, 2006) use many binary symbols so as to enable them to represent a wide variety of human movement, making the notation often redundant for retrieval problem. Although the heuristic method (Müller et al., 2008) eliminates the redundancy of the template, the conventional method does not guarantee the minimality of the resulting template and requires high computational cost.

In the fields of artificial intelligence, a general induction technique has been developed to discover an effective solution to multiple classification problems. The induction method often uses a logical language such as symbolic and clausal language to represent the training data, and discovers a concise classification rule using logical programming, called inductive logic programming (ILP) (Muggleton, 1995). ILP analyzes an essential rule presented in the explicit logical language, and provides a programmable framework based on the same logical language to control its learning procedure.

We propose a rule generation technique for human motion retrieval using ILP. Our method first computes a set of spatio-temporal features of motion appearance in the form of a multivalued logical expression. An ILP framework then discovers an essential classification rule, which is composed of a few logical expressions, by analyzing an intrinsic difference among the set of training motion clips. The desirable segments are retrieved from a database using the discovered rule by specifying the name of the motion class. Moreover, such a retrieval rule can be easily edited in the form of logical language to improve the retrieval accuracy. Consequently, our system provides flexible motion retrieval with semi-automated rule generation.

2 Algorithm

We here explain how to discover the retrieval rule. Given training data, they are manually segmented into clips of unit movement and classified into multiple semantic classes. One class is then chosen as a positive class and others are used as a negative class. Each training clip is represented by a set of clauses corresponding to the spatio-temporal motion features. The inductive learning discovers a retrieval rule consisted of as few clauses as possible so that the resulting rule explains common features of the positive examples and no features of the negative ones.

2.1 Spatio-temporal features of motion appearance

Given a training motion clip, several key-poses are first extracted from the training data to reduce the computational cost of the learning. After selecting the first key-pose at the first frame of the motion sequence, the next key-pose is sequentially searched until the pose distance to the previous key-pose exceeds a given threshold. Next, the multivalued spatial features are computed at each key-frame and then represented in the clausal form like $has\_sf(f_i[l_i])$, where f_i and l_i denote a name and quantization index of a spatial feature, respectively. This clause means that the training data has a pose indicating a spatial feature f_i with the quantization index l_i, where we omit $[l_i]$ for binary features for simplicity. Our definition of spatial feature includes 31 geometrical features proposed in (Müller and Röder, 2006) and 4 additional customized features.

We also define two types of temporal features that explain the duration of a spatial feature and a temporal relation between different spatial features. The duration is represented by two clauses: $long(f_i[l_i])$ and $short(f_i[l_i])$, which indicate longer and shorter duration than 0.5 sec, respectively. The temporal relation is represented by a clause: $after(f_i[l_i], f_j[l_j], l_t)$, for the spatial feature $f_j[l_j]$ appearing after $f_i[l_i]$ with a quantized delay l_t. The time delay index l_t is represented by three symbols: *short* ($[0, 0.25)$ sec), *middle* ($[0.25, 0.5)$ sec), and *long* ($[0.5, 1.0)$ sec), where these time ranges are experimentally optimized.

2.2 Simplification of retrieval rules

Given the clauses of spatio-temporal feature of a training data, the ILP framework discovers the retrieval rule for each motion class. We use a public ILP system, called Progol (Muggleton, 1995), which uses a programmable hypothesis-testing procedure to discover an essential rule. It uses both positive and negative examples to discover a rule that is obeyed by the positive examples and is excluded by the negative examples. This learning model often results in too strict a retrieval rule, which can be reduced by relaxing the tolerance of the quantization error of multivalued feature.

The learning criterion of ILP is the minimality of the clauses used in retrieval rules. Multiple clauses can often be substituted with a simpler clause based on a syllogism and other reasoning. ILP introduces the substitution procedure with user-defined logical expressions represented in the clausal form for discovering the retrieval rule that consists of as few clauses as possible. We define the subsumption relation of multivalued feature, which is modeled by a combinational structure. The basic component of the structure is the quantization index of multivalued feature. The ILP system then selects the most appropriate subset to best describe the feature of training data.

2.3 Subsequence search with space windows

By specifying the name of motion class, motion segments are retrieved by a subsequence search using the retrieval rule associated with the specified class. Our system sequentially searches the subsequence that includes all constituent clauses of the retrieval rule from the motion sequence. The discrete representation of spatial feature often decreases retrieval accuracy because its quantization index is computed by regularly discretizing the geometrical distance between body parts. The space window is therefore introduced for tolerating the small variation of simple quantization of multivalued feature. If a quantization index l_f is assigned to the interval $[d_i, d_{i+1})$, where d_i and d_{i+1} are the geometrical distance between body parts, the retrieval process uses a wider range $[d_i - \alpha, d_{i+1} + \alpha)$ for discriminating the region of the quantization index l_f. The margin α is experimentally optimized by the quantization interval $\alpha = 0.5|d_{i+1} - d_i|$.

Table 1: Retrieval rules discovered from training dataset, where the number in [] represents a quantization index of multivalued features.

Class	Retrieval rule
Cartwheel	long(lhand_up[2]) & long(gradient) & long(move_upward)
ElbowToKnee	long(larm_bent[2])
	long(larm_bent[1]) & after(larm_bent[0], lfoot_up[0], middle)
	short(rarm_bent[1]) & has_sf(lleg_bent[0])
JumpingJack	long(move_upward) & short(lhand_up[1]) & after(rhand_up[2], lhand_up[0], short)
Lie	long(lying)
Sit	long(move_upward) & long(gradient) & long(body_bent[1])
Squat	long(rhand_up[0]) & has_sf(body_bent[0])
	long(lhand_up[1]) & long(rhand_up[1]) & has_sf(body_bent[0])
Toss	long(move_upward) & short(larm_bent[0]) & after(larm_bent[1], rhand_up[1], middle)

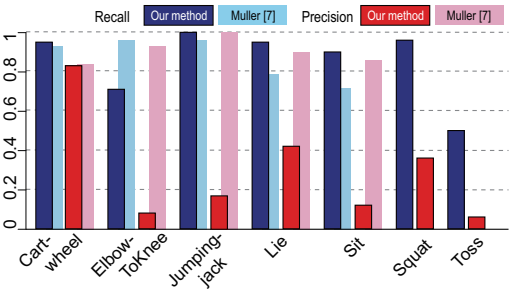


Figure 1: Retrieval results of seven motion classes. The dark-colored bars and light-colored ones represents the performance of our method and existing method (Müller et al., 2008), respectively.

3 Experimental result

The retrieval performance of our method is compared with the existing heuristic method (Müller et al., 2008) under almost the same experimental condition. We experimentally retrieved motion segments from a large public collection of motion capture data (http://www.mpi-inf.mpg.de/resources/HDM05). We manually segment a whole motion sequence of 120 minutes into 5481 clips of unit movement and arranged them into 99 motion classes. The training dataset consists of 7 motion classes; *Cartwheel*(6/21), *ElbowToKnee*(17/58), *JumpingJack*(15/52), *Lie*(6/20), *Sit*(6/20), *Squat*(16/56), and *Toss*(4/14)), where the two numbers in () denote the number of training data of each class and the total number of motion clips, respectively.

3.1 Discovery of retrieval rules

Table 1 shows the retrieval rules for the seven motion classes which are discovered using the training dataset. It shows that a motion clip is classified as a cartwheel motion if the actor bend his/her body and raises his/her left hand, and moves upward for a long period. The number of constituent clauses is determined according to the uniqueness of movements in comparison with other motion classes. For example, the rule of *Lie* motions only has one clause because the spatial feature of *Lying* appears only in the *Lie* motion class. On the other hand, the ILP framework discovers multiple retrieval rules for *ElbowToKnee* and *Squat*, and the desired segment is retrieved using all rules. This indicates that these motion classes can be respectively divided into subclasses. In fact, the training dataset of *ElbowToKnee* includes symmetric motions.

3.2 Retrieval by discovered rule

The statistics of the retrieval performance is shown in Figure 1. Average computational time of the retrieval is about 10 milliseconds, which is fast enough for practical usage. High recall indicates that the subsequence matching with a retrieval rule successfully retrieves almost all relevant motions. The low recall of *Toss* is probably caused by the overfitting problem; the ILP generalizes the small common part of the example motions which do not appear in other *Toss* motions that are not used in the learning. On the other hand, the precision values are remarkably lower than those of the existing method except for *Cartwheel*. This means that the discovered rules can not exclude the non-relevant motions because the number of training data is too small to generalize the retrieval rule for the size of the entire database. However, a large number of examples often lead to a failure in learning because of the limited memory capacity. We consider that the accuracy of our retrieval method becomes acceptable for the practical motion database because the retrieval performance could be improved by a manual editing.

3.3 Extensions of rule-based retrieval

The manual editing can make the retrieval rule more distinctive and often improves the retrieval accuracy. For example, we modify the second clause of the rule of *Cartwheel* from *long(gradient) & long(lhand_up[2])* to *long(somersault)*, based on our knowledge that cartwheel motion includes a handstanding pose. This modification increases the precision and recall from 0.83 to 1.0 and 0.95 to 1.0, respectively. This improvement is attributed to the constraint of *somersault* stricter than that of the *gradient* where both features appear in most cartwheel motions. Such an artificial decision can be integrated into the rule by simple text editing.

Our rule-based method can also use a motion clip for a retrieval key. Given a query clip, every retrieval rule is checked if it categorizes the query motion into one of the given class, and the validated rules are then used for retrieving the similar motion segments. We use the short motion clip composed of several types of gymnastic movements for the retrieval query. Our system validates that the query motion clip consists of subsequences categorized as *ElbowToKnee* and *Squat*. The related motion clips are then retrieved from the database using the two corresponding retrieval rules. This approach enables the retrieval of a semantically similar motion with a large difference in appearance. This property can overcome the limitations in existing numerical techniques.

4 Conclusions

This paper has proposed a rule generation technique for motion retrieval using ILP. The clausal formulation provides a meaningful representation of human motion and its retrieval rules. The retrieval rules are efficiently learned within the ILP framework from a set of manually classified training data. The discovered rule is directly edited in the clausal form. By specifying the name of a motion class, motion segments are efficiently retrieved from a large database using the rule assigned to the motion class with the space windows. Our system also retrieves the motions using a short motion clip for the retrieval query, which actually uses the retrieval rule associated with the query clip.

The major limitation of our method is that the retrieval rule can not be incrementally learned. Another limitation is that our method requires fine adjustment of many numerical parameters. Furthermore, the manual segmentation of the training motions often affects the retrieval accuracy, which is a general issue in example-based motion retrieval techniques. These problems could be alleviated by statistically optimizing the thresholds or using a fuzzy representation in the logical expression. Our future work also involves the investigation of the adaptive sampling method for selecting training data essential to rule generation.

Acknowledgement

This work was supported by the MEXT Grant-in-Aid for Young Scientists (B) 19700090 and Scientific Research (B) 18300068.

References

Arikan, O., Forsyth, D. A., and O'Brien, J. F. (2003). Motion synthesis from annotations. *ACM Transactions on Graphics*, 22(3):402–408.

Chiu, C.-Y., Chao, S.-P., Wu, M.-Y., Yang, S.-N., and Lin, H.-C. (2004). Content-based retrieval for human motion data. *Journal of Visual Communication and Image Representation (Special Issue on Multimedia Database Management Systems)*, 15(3):446–466.

Muggleton, S. (1995). Inverse entailment and progol. *New Generation Computing*, 13:245–286.

Müller, M., Demuth, B., and Rosenhahn, B. (2008). An evolutionary approach for learning motion class patterns. In *Symposium of the Germann Association for Pattern Recognition*.

Müller, M. and Röder, T. (2006). Motion templates for automatic classification and retrieval of motion capture data. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation 2006*, pages 137–146.

Intelligent switch:* An algorithm to provide the best third-person perspective in augmented reality

P. Salamin, D. Thalmann, and F. Vexo
Virtual Reality Laboratory - EPFL
EPFL-IC-ISIM-VRLab Station 14
1015 Lausanne, Switzerland
email: patrick.salamin@epfl.ch
www: http://vrlab.epfl.ch

Abstract

Augmented reality (AR) environments are suffering from a limited workspace. In addition, registration issues are also increased by the use of a mobile camera on the user that provides a first-person perspective (1PP). Using several fixed cameras reduces the registration issues and, depending on their location, the workspace could also be enlarged. In this case of an extended workspace, it has been shown that third-person perspective (3PP) is sometimes preferred by the user. Based on the previous hypotheses, we developed a system working with several fixed cameras that can provide 3PP to a user wearing a video see-through HMD. Our system uses an "intelligent switch" to propose our "best view" to the user, i.e. avoiding markers occlusion and taking into account user displacements. We present in this paper, such a system, its decision algorithm, and the discussion of obtained results that seem to be very promising within the AR domain.

Keywords: Augmented reality, User context awareness, Third-Person Perspective, Best view, Video see-through HMD

Introduction

During our augmented reality experiments with a video see-through Head-Mounted Display (HMD), we always try to provide the best view to the user. As shown in [7], the best perspective depends on the performed action: first-person perspective (1PP) for manipulations and third-person perspective (3PP) for moving actions. In order to propose a 3PP to the user, we need at least a second camera that follows him/her when he/she moves in the environment. Moreover, within the framework of augmented reality, it has been proven that fixed cameras avoid lots of registration issues [2]. Finally, if there are multiple cameras, occlusion problems can also be reduced [5].

Based on the previous researches, we propose in this paper a system with several fixed cameras combined with a mobile one on the user to provide the different perspectives to the user who wears a video see-through HMD. We such a kind of system, we should have better results and provide a better comfort in almost every simulation with augmented reality. Moreover, working with several cameras allows us to enlarge the work area to a building (or at least two rooms in this paper). In order to manage this system, we implemented an "intelligent switch" that chooses the "best view" depending on the user context (location, movement, performed action, and occlusions).

*This research has been partially supported by the European Coordination Action: FOCUS K3D (http://www.focusk3d.eu).

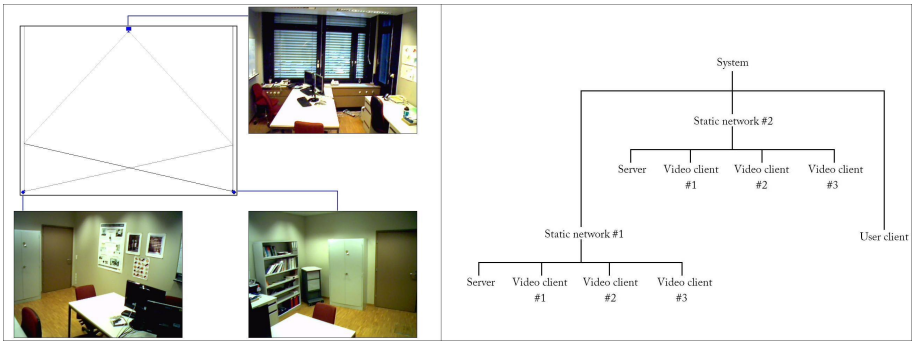


Figure 1: Left: Schema of the cameras (in blue) disposition in the room in order to cover the whole space (their angle of view is represented by the line in different colors). Each camera is coupled with a picture representing a snapshot of its video flow sent onto the network; Right: Schema of the system architecture.

Related work

A well-known way to reduce the marker occlusions consists in working with multiple cameras[3]. Indeed, even if the marker is hidden for one camera, the other cameras (due to their strategic position) should still be able to detect the marker. The second issue, the registration, is one of the main issues in AR software and may be a source of motion sickness.Notice that it has been proven in [2] that the use of a fixed camera considerably reduces this lack.

The researches cited above propose a system providing for augmented reality experiments into a static user working on a very small area like a desktop in indoor, or for a user carrying a camera outdoor[1]. Unfortunately, in this last case, users are carrying a camera, which is then not at a fixed location. And as shown above, this induces registration issues.

We will then use several fixed cameras to provide 3PP when the user is moving, and 1PP for the fine manipulation with a camera coupled on the users' HMD.

Description of the system

The goal of our system is to provide the "best view" to a user who can move in several rooms and manipulate objects in augmented reality. Based on previous studies of Salamin et al. [7][6], we know that these two actions require different perspective: third-person perspective (3PP) for navigation tasks, respectively first-person perspective (1PP) while manipulating an object with the hands. In our case, instead of having a camera that follows the user for the 3PP, we decided to have multiple fixed cameras. Consequently, the user will not need to matter about collisions of a cumbersome backpack with wall, ceiling, doors, etc.

As there are multiple cameras, we need a system that will automatically detect which camera needs to be activated for the user best view. For this simulation, we will work in an area of two adjacent rooms in which we already put three cameras at strategic positions (see left pic of Figure 1). Our system (right pic of Figure 1) considers that there are two networks of three cameras (one network per room). Our system then first have to localize the user, i.e. in which room he/she is currently. Once done, depending on the visibility of the markers and on the displacement of the user, our system will choose which video stream to provide to the user.

Video clients are linked to a TRUST Webcam (30fps at 1280x1024). They have three tasks to perform in parallel: acquire webcam video flow with the help of the Digital Signal Video Library (DSVL) and process it with ARToolKit to detect if there are visible markers; stream via RTP this JMF-processed video flow onto their network in a continuous way; and connect to the server of their network to transmit the markers visibility status (ARToolKit).

The servers also have three main tasks to perform: detect and accept all the three video clients (plus the user) of its network to receive their information about their markers visibility status;

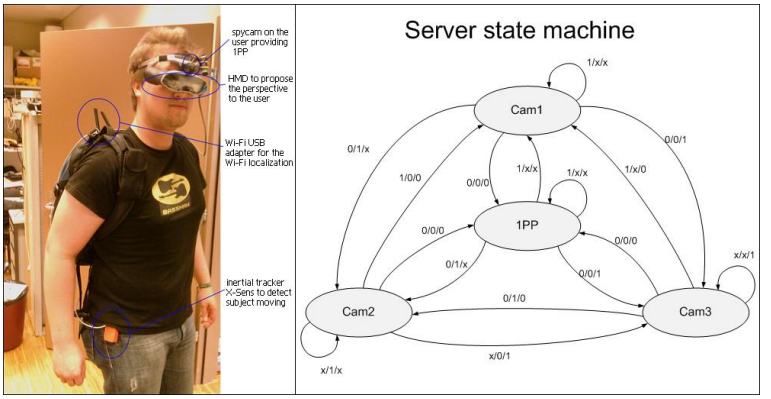


Figure 2: Left: User fully equipped; Right: Schema defining the switch of perspective in a room. The states correspond to the active camera (providing the video flow to the user) and the values on the link between them represent if they "can see" the ARToolKit marker ('1') or not ('0'). (Notice that the value 'X' means that any value can fit)

choose the best video stream to indicate to the user client; and inform user client the video stream to connect to.

The user client is composed of a notebook (Windows Vista for network tricks with a Wi-Fi antenna to connect to the network server) linked to a Trust Webcam, an XSens inertial tracker (user movements), a Wi-Fi USB adapter (user location), and a SONY Glasstron video see-through HMD (user video feedback).

Here are the main rules of our algorithm(presented in left pic of Figure 2): no movement detected by the inertial tracker on the user directly leads to choose 1PP anddetected movement leads to propose 3PP.Notice that there is a preference for a camera located behind the user in order to avoid "mirror effect" that may introduce biases like the right-left inversions.

Obviously, our system must avoid changing too many times the chosen perspective. Indeed we hypothesize that each camera change would perturb the user.

Experimentation

We tested our system with 12 naive and voluntary participants (10 males and 2 females). They were all between 20 and 35 years old. The equipment previously described is shown in right pic of Figure 2.

Protocol

There are four main steps in the experimentation described in this paper. As written above, our working space is extended to two rooms. This means that one step will be to move from a room into the other one.

Another step of this experiment is to stay in a room with no displacement and to turn on oneself (use of 1PP).

A third step consists in walking in the room (use of a chosen 3PP).

The last step is the manipulation of an augmented object. This action can be performed while walking or staying, which means a change of perspective (3PP, respectively 1PP).

All the four steps cited above are performed several times in different orders during the experimentation. The experimentation usually lasts around twenty minutes.

Questionnaire

Once the experiment performed, we proposed a SUMI-like (Software Usability Measurement Inventory) questionnaire [4] to the users. This questionnaire is composed of two parts: user profile

and software evaluation (50 statements).

Results

Globally, most of users enjoyed the system. Every step was performed by every user, even if some of them needed more time to adapt to the system. They walk a lot in the rooms looking for augmented objects and trying the perspectives. We will now first present the users' answers tothe questionnaire proposed to them after the experiment.

Our adapted SUMI questionnaire was filled by every participant. Its first part, concerning the users' profile, reveal us that twenty minutes for training was widely enough all the participants but one.

The questions of the second part reveal us that our software is very accurate and fast to leave the perspective current when the augmented object disappears. But it also informs us that the reconnection to another video flow (couple of seconds) can be very perturbing at the beginning.

Our system was then considered as very attractive and intuitive enough, even if improvements can be done for a future version.

Conclusion and further works

The obtained results confirm our hypotheses. User comfort does not suffer from the changes of perspective; some of the users even play at forcing the perspective change during the experiment. Working with an augmented environment larger than a desktop seems to be very promising for future researches in this domain.

Participants, who already took part to previous 3PP experiments with a camera coupled to a backpack on their body, especially appreciate the change of perspective that avoids occlusions.

An improvement would be to reduce the time needed for the perspectives switch and improve the image quality. A solution to this problem would be to send only the video flow chosen by the system to the access point. This would allow us to send video flows with a higher resolution onto the network.

References

[1] Cheok, A. D., Goh, K. H., Liu, W., Farbiz, F., Fong, S. W., Teo, S. L., Li, Y., and Yang, X. (2004). Human pacman: a mobile, wide-area entertainment system based on physical, social, and ubiquitous computing. *Personal Ubiquitous Comput.*, 8(2):71–81.

[2] Collins, R., Lipton, A., Fujiyoshi, H., and Kanade, T. (2001). Algorithms for cooperative multisensor surveillance. In *Proceedings of the IEEE*, volume 89, pages 1456–1477.

[3] J., K., I., C., and G., M. (2003). Continuous tracking within and across camera streams. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 267–272.

[4] McSweeney, R. (1992). Sumi – a psychometric approach to software evaluation. Unpublished MA (Qual) thesis in Applied Psychology, University College Cork, Ireland.

[5] Mittal, A., Larry, and Davis, S. (2003). M2tracker: A multi-view approach to segmenting and tracking people in a cluttered scene using region-based stereo. In *International Journal of Computer Vision*, pages 189–203.

[6] Salamin, P., Thalmann, D., and Vexo, F. (2008). Improved third-person perspective: a solution reducing occlusion of the 3pp? In *VRCAI 2008, the 7th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and its Applications in Industry.*

[7] Salamin, P., Vexo, F., and Thalmann, D. (2006). The benefits of third-person perspective in virtual and augmented reality? In *ACM Symposium on Virtual Reality Software and Technology (VRST '06)*, pages 27–30.

Simulating Self-forming Lane of Crowds through Agent Based Cellular Automata

Mankyu Sung
Electronics and Telecommunications Research Institute(ETRI)
161 Kajung-dong, Yusung-gu, Daejeon, Korea
mksung@etri.re.kr

Abstract

This paper presents a simple model for simulating self-forming lane phenomenon among crowds. This model combines the agent-based scheme with CA model to achieve both smooth motion trajectory and real time simulation of self-forming lane at the same time. Our model also has an advantage in memory use because our model computes crowd density and flow directions, which are two important values for self-forming lane, using big grid structure with interpolation technique. The proposed model has been validated through several experiments.

Keywords: *Crowd Simulation, Cellular Automata*

1. INTRODUCTION

In simulating crowds, the most evident collective behavior that crowds show is so-called self-forming lane behavior [1][2]. Basically, this phenomenon shows a natural flow of crowds by causing segregation in crowds even when they are moving in opposite direction along a path and then forming natural lane by themselves. This behavior is quite important because many transportation researches and computer simulations use this behavior as a measure to check the accuracy of crowd simulation [2][4][5]. Also, it is important to provide collision-free natural flow of crowds even when the density of crowd is pretty high just like the case of real life.

This paper presents a simple algorithm to simulate the self-forming lane behavior by combining cellular automata (CA) with the agent-based scheme [6]. Unlike the traditional CA algorithm where all environments and agents move in a discrete space, our algorithm can move the agents with arbitrary speed and directions because agents are simulated independently, which results in smooth motion trajectories. Our CA model, on the other hand, has a 2D field that contains the regional density of crowds and overall crowd direction information in its relatively large grid structure. Because we adopt the 4-way interpolation technique to guess the values inside the grid, our model does not require small grid size, which is required for smooth trajectory for CA model. This advantage leads to the reduction of memory use significantly. Memory efficiency is especially a very important problem for simulating large number of crowds. Our major contribution is to improve the existing CA model with Agent-based technique for better quality of individual agent motion.

This paper consists of following chapters. Section 2 presents the related work, section 3 explains our algorithm in detail, and section 4 shows the experiment and results. Finally, we conclude this paper with discussion in section 5.

2. RELATED WORK

Crowd simulation has drawn a lot of attentions from computer graphics researchers in recent years. Treuille *et al* proposed a dynamic potential field method based on density-dependent velocity terms for modeling the "*lane formation*" in crowds [1]. Essentially, this method models a crowd as a collection of identical particles. Therefore, each agent's personal difference and characteristics are ignored.

Agent-based systems [3], on the other hand, are able to represent the agent-specific properties easily because they simulate each individual independently but it is hard to for them to simulate the aggregate behaviors which need fine level of control between agents [3].

Most currently, several statistical physicists propose pedestrian dynamic models using cellular automata theory [2][4][5]. Although those models can simulate aggregate behaviors such as self-forming lane, they have drawbacks in motion quality and memory efficiency. That is, complex individual motions are not guaranteed.

Our method combines the agent-based approach with existing CA model. As a result, agent's motion trajectories are smooth yet produce the realistic aggregate behavior like a self-forming lane.

3. AGENT BASED SCHEME

In our system, agents update their position and velocity sequentially at each time step [3]. Every individual agent has his own property including character size and default speed. Update depends on so-called "*driving force*", V, which is a 3D vector that forces the agent to move to a specific direction. The bigger the length of the vector, more powerful the driving force is. Determining the driving force should take several factors into account; it should consider the avoidance of the collision with static obstacles in the environment as well as other moving agents; it should drive the agent to his target position if he has one. In addition, if a path is specified to the agent, he should stay on the path as long as possible. In our approach, each factor proposes a local driving force vector, $V_{local}$and then final driving force $V$is computed by summing up all local driving forces:

$$V = \sum_{k=1}^{n} W_k \cdot V_{local}(k)$$

where n is the number of local forces and $W_k$are weight values. W_k are computed automatically depending on the urgency. For example, collision avoidance with static or moving obstacles should have a high weight value.

Special care must be taken when we deal with collision between agents. In our approach, the local force for collision avoidance is obtained by computing the perpendicular vector against the vector that is a multiplication of neighbor agents' direction (forward) vectors. The neighbor agents can be identified through the lattice-bin method. Static obstacles are approximated by 2D circles and we let each circle emit a repulsive local driving force depending on the distance from the obstacle, which is a lateral force that turns the agent towards edge of the obstacle. The lattice-bin method assumes that entire environment is subdivided into grid and each cell contains a list of agents inside. When an agent enters one of the cells, the agent list is updated and all existing agents in the list correspond to neighbor agents.

Once we get the driving force, it is adjusted and truncated by the predefined maximum speed. Afterwards, to get the smooth agent movement, our model computes the adjusted acceleration vector through Newton's law first and then interpolates it with the current velocity vector. Finally, agent's position is obtained through the Euler integration method. This results in smooth velocity changes and consequently smooth positional changes.

Although update rules above produce smooth and realistic agent motion, they are not necessary adequate for aggregate behavior such as self-forming lane because they do not consider the inter-agents relationship. The required information includes the density and direction of crowd flow. In the next section, we are going to introduce a simple method that integrates the CA model with the agent-based model to produce the aggregate behaviors as well as to keep the smooth motions.

4. CA FIELD

The self-forming lane behavior occurs when agents detect the crowd density and velocity of crowd flow ahead and reflect those information when they compute the driving force. In practice, agents should check the density of crowd in front of them and decide whether or not they should follow other agents' direction. For this goal, our model use a 2D CA field structure and embeds all required information into four corners of the cell so that, when an agent enters one cell, he or she receives the information automatically. This is similar to the MAC structure proposed in [1]. The cell structure is represented in Fig 1. The $\delta 1 - \delta 4$ are the values of distance from the agent to the four vertices.

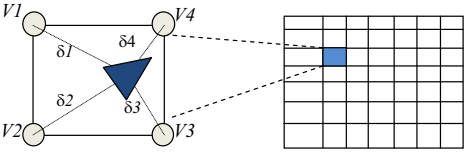


Fig.1 Cell Structure

Essentially, we make four vertices (*v1-v4)* of the cell have a set of values associated with each cell. The values of the vertices are registered and updated when a new agent enters the cell. Two important information that each vertex contains are the density of the crowd, *d*, and the velocity of the flow, *f*.

- $d \in \mathbb{R}$: Density of crowd
- $f \in \mathbb{R}^2$: Velocity of crowd flow

Since one vertex is in general shared by four cells (north, south, west, and east) except boundary vertices, the update of the information of each vertex can happen four times at maximum rate.

The density of crowd is a real value that is calculated by the following equation:

$$d = \frac{\sum d_v(i)}{4}$$

$$d_v(i+1) = d_v(i) + \left(1 - \frac{\delta(i)}{\sum_{k=1}^{4} \delta(k)}\right) \cdot P$$

Intuitively, the density *d* of the cell is an average value of four vertex density $d_v(i)$, which is increased and decreased as agents enter or exit the cell. Given a real value of single agent's contribution to the cell, *P*, when one agent enters a cell, four values of $d_v(i)$ are computed depending on the distance from the agent. So, the closer to the vertices the agent is, the bigger value is given to the vertex. Because it may take several time steps to cross one cell, the distance from the agent to the four corners is also changing at each time step. Therefore, the agent's contribution to four corners is changing while the agent stays in the cell.

Because our goal is to attract following agents to form a lane, we keep *m* number of previously visited cells in agent history, and also compute the density value of those cells with difference *P* value, which is decaying from 1 to 0.01 with rate *r*. This process leaves a sort of "*trace*" behind the agent, which provides information such as density change over the short period of time. Because *P* is gradually decreased, the influence of an agent to the density of cells he has already visited is also gradually decreased.

On the other hand, the crowd flow vector is a 2D vector representing the velocity of crowd at a particular region of the environment. This value is also need to be registered in the grid structure in almost same way as crowd density. When an agent enters a cell, his current velocity (f_c) is decomposed into four little vectors ($f_v(i)$) whose length depends on the distance between the agent and the corner ($w(i)$). Then, the final crowd flow vector (*f*) is obtained by averaging four vectors.

$$f_v(i+1) = f_v(i) + \left(1 - \frac{\delta(i)}{\sum_{k=1}^{4} \delta(k)}\right) \cdot f_c$$

$$f = avg(f_v(i)), 1 \leq i \leq 4$$

5. SIMULATING SELF-FORMING LANE

Based on the density of crowd and crowd flow velocity, crowd computes a new local driving force, V_d, and add this on the final driving force$V$. One important point is that crowd forms a lane only when the crowd flow velocity is relatively big due to the high density of crowd [1]. This makes sense because people do not tend to join a crowd flow when there are only a small number of people. Our model has a predefined threshold density value *T* to decide whether or not to turn the crowd flow velocity into local driving force. If the crowd density is bigger than *T*, then the crowd flow velocity is treated as a driving force and added to the final driving force V. Otherwise, it is discarded at the time step.

$$\begin{cases} v_d = f & d > T \\ v_d = 0 & otherwise \end{cases}$$

6. EXPERIMENTS AND RESULT

We have tested our model on the following situation: we first create 50 agents and then divide them into two groups. Two groups are separated in distant locations and all members of a group have a target position in the other side of the group, which makes both groups cross in the middle (Fig 2). As we can see in Fig 2, crowds predict the collisions ahead first (2) and then check the density of crowd, and, separated from their group (3), start to form lanes by themselves (4). This results in "no stuck" among the crowds, which produces the natural crowd flow.

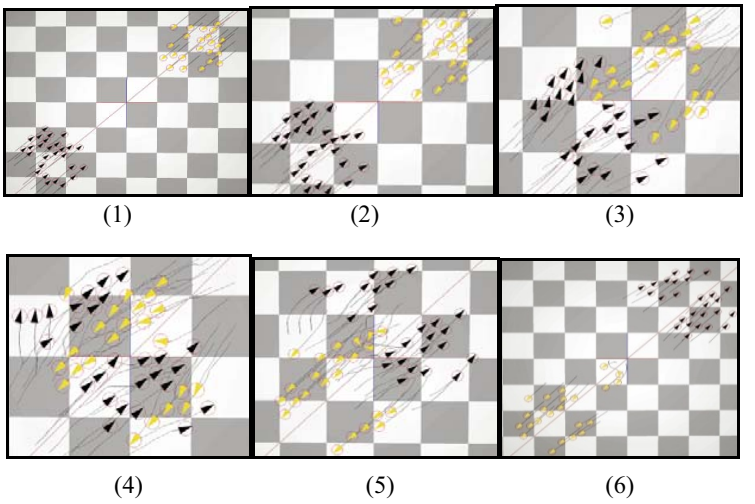


Fig.2 Snapshots of self-forming lane phenomenon

7. CONCLUSION

This paper presents a simple and efficient algorithm for simulating self-forming lane phenomenon in real time. Our model combines the agent based approach with CA model to produce the smooth trajectory of agents as well as simulating the realistic lane formation in real time. Our future work focuses on implementing the human motion data to the agent trajectory.

REFERENCES

[1] Treuille, A. Cooper, S. Popović, Z.,"*Continuum crowds*", ACM Transactions on Graphics 25(3) (SIGGRAPH 2006)

[2] Yamamoto, K. Kokubo, S Nishinari, K, "*Simulation for pedestrian dynamics by real-coded cellular automata(RCA)*", Physica A ,379(2007), 654-660.

[3] Sung, M. Kovar, L. Gleicher, M, "*Fast accurate goal-directed motion synthesis for crowds*",ACM/Eurographics Symposium on computer animation(SCA) 2005.

[4] Jian, L. Lizhong, Y. Daoliang, Z. "*Simulation of bi-direction pedestrian movement in corridor*", Physica A ,354(2005), 619-628.

[5] Kirchner, A. Nishinari, K. Schadschneider, A. "*Friction effects and clogging in a cellular automation model for pedestrian dynamics*", Physical Review E 67, 056122, 2003

[6] Nagel, K. Schreckenberg, M."*A Cellular automation model for freeway traffic*", J. Phy, I France 2, 2221-2229, 1992

Towards Realistic Simulation of Skin Deformation by Estimating the Skin Artifacts

Dr. Y.M. TANG
The Hong Kong Polytechnic University,
Department of Industrial and Systems Engineering, Hong Kong SAR, China;
mfymtang@inet.polyu.edu.hk

Dr. K.L. YUNG
The Hong Kong Polytechnic University,
Department of Industrial and Systems Engineering, Hong Kong SAR, China;
mfklyung@polyu.edu.hk

Abstract

Simulating human skin deformation has wide application in the areas of computer graphics. Nowadays, human biomechanics are usually determined by the marker-based motion capture system. Nevertheless, it is known that the computation of human biomechanics is significantly affected by the displacement of the markers due to the skin artifacts. In order to obtain the prior skin parameters for the accurate deformation of the skin layer, we propose to quantify the skin movement artifacts in terms of the probability density using a set of skin markers.

We demonstrate our works by measuring the skin artifacts of the lower limb of the human body. In order to estimate the skin artifacts, a set of skin markers was attached to the thigh. The least-square minimization was adopted to determine the rigid motion of the thigh segment. The determined motion of the thigh was then used to estimate the position of the markers at the bony landmarks. The displacement of the markers was computed by the distance between the measured and the estimated markers. We estimate the skin artifacts at the thigh during walking motion. It was found that the skin artifacts can be reasonably approximated with a Gaussian function.

Keywords: skin artifacts, motion capture system, probability density function, root means square displacement.

1 INTRODUCTION

Simulating human skin deformation has wide application in the areas of computer graphics. Nowadays, geometry based and physics based approaches are adopted to simulate the human skin deformation. Geometry based approaches such as free-form deformation (FFD) [1], [2] employs purely geometric techniques to model deformation. These approaches provide flexibility to for the users to control the deformation. However, it relies on the skill of the users for accurate simulation of the model. Recently, physics based approaches are becoming more popular. The most popular one is the mass-spring system [3], [4] because of its simplicity and capability to achieve real-time performance. Another approach to simulate human skin deformation is the finite element method (FEM) [5]. Despite numerous approaches have been proposed to simulate the skin deformation, the methods are still difficult to model skin deformation to be applied in biomedical applications because prior information are usually required for determining the deformation of the skin layers.

In this article, we propose to obtain the prior information of the human skin artifacts by using the skin markers. The magnitude of the skin artifacts is quantified in terms of the probability density measured by a cluster of skin markers. The proposed technique makes use of a set of skin markers to estimate the motions of the rigid segments. The least-square minimization is adopted to determine the motions of the rigid segments. The position of the markers was estimated based on the computed motions of each segment. Then, the displacement of the markers is computed by determining the distance between measured and estimated markers. We demonstrated our experiment results with walking motion.

2 METHOD

The proposed method determines the amount of displacement of the markers at the bony landmarks in two stages. The transformation between the first and the subsequent frames is firstly determined by the least-square minimization technique. The determined transformation is used to estimate the position of the markers at the bony landmarks. Then, the displacement of the markers is determined by the distance between the estimated and measured markers. A computer program developed using the Matlab (The MathWorks, Inc.) was adopted for computations.

A subject age 30 years old, 63 kg and 182 cm height was participated in this test. In our experiment, 7 markers were attached to the thigh. In which, 3 markers were attached to the bony landmarks including lateral epicondyle, medial epicondyle and greater trochanter. The rest of the markers were evenly distributed at the thigh segment. The 3D coordinates of each reflective marker was captured using the motion analysis system (Motion Analysis, Santa Rosa, USA). The system consisted of 8 cameras connected to a controlling computer. The capturing frequency was 120 Hz. The cameras were firstly calibrated before capturing the human dynamic motions. The trajectories of each reflective marker were the output of the system.

2.1 *ESTIMATING THE POSITION OF THE MARKERS AT THE BONY LANDMARKS*

Suppose a set of N reflective markers ($N > 3$) is evenly attached to the skin of a rigid segment of the human body. The marker set includes the markers at the bony landmarks that we are going to estimate. The number of makers at the bony landmarks N_b is less than the total number of reflective markers at the segment, such that $N > N_b$. The estimated rotation matrix $\mathbf{R}$ and translation vector $\mathbf{t}$ of the marker set between the 1st and the j^{th} frame is firstly determined by minimizing the typical least-square equation [6, 7].

Let $\left\{\mathbf{y}_1^j,...,\mathbf{y}_{N_b}^j\right\}$ be the three-dimensional (3D) position of the markers at the bony landmark at the j^{th} frame of the motion sequence. The estimated 3D position of the k^{th} marker at the bony landmark at the j^{th} frame, $\mathbf{y}_k^j$ is computed by

$$\mathbf{y}'^j_k = \mathbf{R}\mathbf{y}_k^j + \mathbf{t}. \quad (1)$$

2.2 *COMPUTING THE SKIN ARTIFACTS*

Denote the position vector of the k^{th} estimated and measured marker at the bony landmark as $\mathbf{y}'^j_k = \begin{bmatrix} x'^j_k \\ y'^j_k \\ z'^j_k \end{bmatrix}$ and $\mathbf{y}_k^j = \begin{bmatrix} x_k^j \\ y_k^j \\ z_k^j \end{bmatrix}$ respectively. The displacement of the k^{th} marker in the x-, y- and z-directions at the j^{th} frame between the estimated and the measured markers at the bony landmark is computed by

$$\mathbf{e} = \begin{bmatrix} e_x^k \\ e_y^k \\ e_z^k \end{bmatrix} = \begin{bmatrix} x'^j_k - x_k^j \\ y'^j_k - y_k^j \\ z'^j_k - z_k^j \end{bmatrix}. \quad (2)$$

Then, the magnitude of the displacement of the k^{th} marker due to the skin artifacts is determined by the norm of **e**

The root mean square displacement (RMSD) of the markers at the bony landmark are computed by

$$\text{RMSD} = \sqrt{\frac{\sum_{i=0}^{N_b} \|e_i\|^2}{N_b}} . \quad (3)$$

3 RESULTS

We demonstrated our method by estimating the displacement of the markers at the lower extremity in a walking cycle. Figure 1 illustrates the displacement of the markers at the bony landmark in the *x*-, *y*- and *z*-directions in terms of the probability density in a walking cycle. It was found that the maximum magnitudes of the markers displacement including greater trochanter, lateral epicondyle, medial epicondyle were 7.99 mm, 11.83 mm and 12.43 mm respectively. The RMSD of the markers were 5.12 mm, 7.11 mm and 6.05 mm respectively. The average maximum displacement in the thigh was 10.75 mm.

Table 1: The maximum displacement and the RMSD of the markers at the thigh (greater trochanter: GT, lateral epicondyle: LE, medial epicondyle: ME) and shank (lateral tibial plateau: LTP, medial tibial plateau: MTP, lateral maleoli: LM, medial maleoli: MM) during walking. All values are in millimetres (mm).

		Max. displacement (mm)	RMSD (mm)
Thigh	GT	7.99	5.12
	LE	11.83	7.11
	ME	12.43	6.05

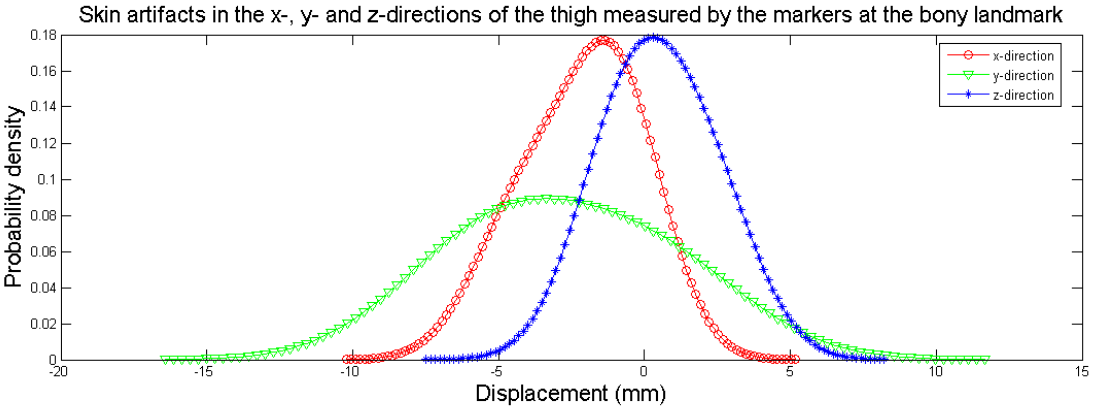


Figure 1: The probability density of the markers displacement during walking

4 DISCUSSION AND CONCLUSION

In order to obtain the prior information of the human skin artifacts for realistic skin deformation, a method that estimates the displacement of the markers due to the skin artifacts during walking has been presented. The method makes use of the skin markers instead of the bony markers as proposed in most of the literatures [8, 9] to estimate the skin artifacts. The setup of the experiment of the proposed method is simple and easy for implementation.

The propose method was validated by comparing with the results of a number of existing works [10, 11, 12]. Our results have indicated that the displacement of the markers due to the skin artifacts was approximately Gaussianly distributed. As illustrated in [12], the maximum magnitude of displacement was 10 mm and 6 mm respectively with up to 60 degrees of knee angle during a walking motion. The results agreed with our experimental results. In addition, the RMSD and the displacement of the markers at each bony landmark including the greater trochanter, lateral epicondyle, lateral malleolus etc. have been compared with the figures in [13] showing similar experiment results. In spite of our results the prior information of skin deformation have been compared with the results in the published articles, the results of the skin artifacts have not been applied in simulating the deformation of the skin artifacts. In the future, the measured skin artifacts should be applied to simulate the deformation of the skin layers towards an accuracy that can be applied in various biomedical applications.

REFERENCES

[1] Sederberg, T. W. and Parry, S. R.. Free-Form Deformation of solid Geometric Models. SIGGRAPH, 20(4): pp.151-159, 1986.

[2] Barr, A.H. Global and Local Deformations of Solid Primitives, *Computer Graphics* (SIGGRAPH '84 Proceedings), pp. 21-30.

[3] Mathieu Desbrun, Peter Schröder, and Alan Barr. Interactive Animation of Structured Deformable Objects. Proceedings of Graphics Interface, 1-8, 1999.

[4] A. Luciani, S. Jimenez, J.-L. Florens, C. Cadoz and O. Raoult. Computational Physics: a Modeler-Simulator for Animated Physical Objects. Eurographics, Elsevier, Vienne (Autriche), 1991.

[5] J.P. Gourret, N.M. Thalmann and D. Thalmann. Simulation of object and human skin deformations in a grasping task. Computer Graphics, 23: 21-29, 1989.

[6] Soderkvist I., Wedin P.A. Determining the movements of the skeleton using well-configured markers. Journal of Biomechanics, 26:1473-1477, 1993.

[7] Zhuang H.Q., Sudhakar, R. Simultaneous Rotation and Translation Fitting of 2 3-D Point Sets. IEEE trans. on SMC-B, 27(1):127-131, 1997.

[8] Benoit D.L., Ramsey D.K., Lamontagne M., Xu L., Wretenberg P., Renstrom P. Effect of skin movement artifact on knee kinematics during gait and cutting motions measured in vivo. Gait Posture, 24:152-164, 2006.

[9] Manal K., McClay I.S., Stanhope S.J., Richards J., Galinat B. Comparison of surface mounted markers and attachment methods in estimating tibial rotations during walking: An in vivo study. Gait & Posture, 11(1):38-45, 2000.

[10] Reinschmidt, C., van den Bogert A.J., Nigg B.M., Lundberg A., Murphy N., Stacoff A., Stano A. Tibiofemoral and tibiocalcaneal motion during walking: external vs. skeletal markers. Gait and Posture, 6:98–109, 1997.

[11] Reinschmidt, C., van den Bogert A.J., Nigg B.M., Lundberg A., Murphy N. Effect of Skin Movement on the Analysis of Skeletal Knee Joint Motion During Running. Journal of Biomechanics, 30(7):729–732, 1997.

[12] Sangeux M., Marin F., Charleux F., Dürselen L., Tho M.H.B. Quantification of the 3D relative movement of external marker sets vs. bones based on magnetic resonance imaging. Clin Biomech, 21(9):984-991, 2006.

[13] Angeloni C., Cappozzo A., Catani F., Leardini A. Quantification of relative displacement of skin- and plate-mounted markers with respect to bones. Journal of Biomechanics, 26:864, 1993.

3D Characters that are Moved to Tears*

Wijnand van Tol and Arjan Egges
Games and Virtual Worlds group
Utrecht University, the Netherlands
wijnandvantol@gmail.com, egges@cs.uu.nl

Abstract

Displaying facial motions such as crying or laughing is difficult to achieve in real-time simulations and games. Not only because of the complicated simulation of the physical characteristics such as muscle motions or fluid simulations, but also because one needs to know how to control these motions on a higher level. In this paper, we propose a method for simulating realistic tears in real-time. The tear simulation is based on the Smoothed Particle Hydrodynamics technique, which we optimized for real-time tear generation and control. Additionally, our method works independently of the graphics and physics engines that are used.

Keywords: Fluid simulation, animation, emotion, virtual character

1 Introduction

Virtual characters in games and simulations are becoming more realistic, not only through more detailed geometry but also because of better animations. Due to motion capturing and automatic blending methods, both facial and body motions can be displayed convincingly. One of the main goals of using virtual characters in 3D environments is to make an affective connection with the user. An important aspect of establishing this connection between the virtual character and the user is by using emotions. Virtual characters are already capable of displaying a variety of emotions, either by employing different body motion styles (Egges and Magnenat-Thalmann, 2005) or facial expressions (Garchery and Magnenat-Thalmann, 2001). However, the range of emotions for virtual characters is still quite limited. For instance, more extreme expressions such as laughing out loud, screaming in anger, or crying are rarely seen in games. The absence of these extreme expressions results in a less emotionally involving experience, as opposed to movies, where such expressions are used regularly to entice the viewer. In this paper, we propose a system for automatically generating and displaying crying motions in real-time. We achieve this by using a real-time fluid simulation method, which we optimized for crying synthesis.

This paper is organized as follows. In the next section, we will discuss related research. Section 3 describes our real-time crying simulation method. Then, some results of our approach are shown in Section 4. Finally, we present our conclusions and recommendations for future work.

2 Related Work

In order to accurately reproduce crying motions for virtual characters, we first need to determine what the phenomena of crying are and when crying occurs. There have been several studies that investigate why people cry, and these have been summarized by Vingerhoets et al. (2000). In particular, we would like to mention Borgquist (1906) who did a study among students, which pointed out three mood states in which crying occurs: anger, grief or sadness, and joy. He also pointed out accompanying physical states such as fatigue, stress and pain. Other phenomena include an increased heart rate and flow of blood to the head, as researched by Ax (1952).

*This research has been supported by the GATE project, funded by the Netherlands Organization for Scientific Research (NWO) and the Netherlands ICT Research and Innovation Authority (ICT Regie).

A major drawback of any existing facial animation engine is that they only allow for geometrical deformation of a face mesh (Garchery and Magnenat-Thalmann, 2001). For displaying a crying motion, a more elaborate animation model is required, involving a physical simulation of fluids. Fluid simulation can be done in different ways, the most common approaches are grid-based (Eulerian) and particle-based methods. Grid-based simulations (Chen et al., 1997) use a grid of points connected by springs. For each point at each time step, the next position is calculated using Euler integration. Grid based simulations for liquids usually involve Navier-Stokes equations to calculate the momentum of the fluid, and need additional formulas to conserve the mass and energy of the system. These computations can be quite complex. Additionally, the grid-based approach is not suitable for simulating drops of water, since multiple meshes are required.

The Smoothed Particle Hydrodynamics (SPH) approach (Monaghan, 1992) uses particles with a fixed mass. Each particle represents a volume, which is calculated by dividing the mass by the density of the particle. SPH is a very general method which can be used for any application where field quantities have to be calculated. Müller et al. (2003) proposed a method where he applies SPH to fluid simulation. While the physical fluid behavior is simulated by the particle system, the visualization step is generally done by applying a point-splatting technique (Zwicker et al., 2001) or a marching cubes algorithm (Lorensen and Cline, 1987). SPH is a suitable approach for simulating smaller bodies of water (such as drops or tears). However, a challenge lies in integrating this method into a facial animation engine, in a way that allows an animator to control the fluid behavior as a part of the facial animation. Also, existing visualization techniques do not take interaction with the skin into account. Finally, real-time performance is a requirement for such an integrated system.

In the next section, we will propose a novel SPH approach, optimized for the simulation of crying motions. Our tear simulation technique takes the skin into account by defining an additional skin adhesion force. Furthermore, we propose a realistic tear trail synthesizer for visualizing the interaction between tears and the skin.

3 Real-time Crying Simulation

In this section, we will present our SPH-based real-time crying engine. First we will discuss the generic SPH approach and our added extensions. Next, we will discuss the visualization of the fluid. Finally, we show a simple method for simulating the trail of a teardrop.

3.1 Smoothed Particle Hydrodynamics

The SPH method (Müller et al., 2003) consists of a few basic steps to be executed for each frame. First, the densities of the particles need to be calculated based on their current position. Then, the pressure and viscosity forces of the Navier-Stokes equations have to be calculated for each particle. An additional surface tension force is defined for better fluid stability. Finally the external forces are applied to the particles. Originally, these consist of a gravity force and a collision force. A problem with this approach is that it works best with a large number of particles, which puts a strain on the real-time requirement of the fluid simulation. Furthermore, the method does not take friction into account. We have addressed this issue by defining a **skin adhesion** force for the particles that are in contact with the surface. The direction of this force is in the opposite direction of the normal of the surface (thereby attracting the particles to the surface), and its size is dependent on the surface volume:

$$\mathbf{f}_i^{\text{adhesion}} = -\alpha \frac{\mathbf{n}_s}{\rho_i}.$$

where $\mathbf{n}_s$ is the normal of the surface and α is a coefficient which determines how 'sticky' the fluid is.

3.2 Mesh generation

In the previous section we discussed how the physical behavior of tears are simulated using the SPH method, extended with a skin adhesion force. The second step is to visualize the fluid by

generating a mesh. Based on the location of the particles, a scalar field is defined to estimate the surface and the normal of the simulated fluid. Following Müller et al. (2003), we will call this scalar field the *color field* in the remainder of this paper. The color field is then visualized using a mesh generation algorithm, such as marching cubes Lorensen and Cline (1987) or point-splatting Zwicker et al. (2001). The point splatting technique is more efficient than the marching cubes approach, but it requires an extension of the renderer. In order to provide an implementation that is independent of the rendering engine, we have opted to use the marching cubes algorithm for visualizing the isosurface of the fluid. This algorithm traverses the scalar field through a grid. At each point on this grid, the algorithm samples 8 neighboring locations, forming an imaginary cube, then moves to the next 'cube' on the grid. Each of the points on a cube has a value in the scalar field. The isosurface is determined by an iso-value. If the value of a point is higher than the iso-value it is inside the surface, otherwise it is outside the surface. So a cube can be fully inside the surface, if all the points have a higher value than the iso-value, fully outside it, or partially inside it. Through rotations and reflections there are only 15 unique polygon configurations for a cube Lorensen and Cline (1987), formed by the points determined by the iso-values. The next step is calculating the correct normals for each polygon. Since the mesh is created for the isosurface of the smoothed color field c_s, we can use the gradient of the smoothed color field at the location of the vertices as the normal. By traversing the grid and creating the right polygons and normal for each cube a mesh is generated that resembles the isosurface.

3.3 Fluid trail synthesis

In real life, a water drop leaves a trail as it slides down a surface. This is partly because of the adhesive force. Water molecules stick to the surface, decreasing the volume of the drop. We have developed a method that simulates the trail and the decreasing volume of a tear. Every step, we identify the different tear drops and which particles are part of which drop. For each drop, we identify the left-most en right-most position. We also take the middle point of those two. We will call those the boundaries of the teardrop. The middle point is elevated a small amount in the direction of the normal of the skin and the left and right are moved in the opposite direction. With these three points, their normals and those points and their normals from the previous frame, we create four triangles, which simulate a small part of the trail. First, at every frame, we compute the new boundaries of the tear. We store the boundaries of this tear only if the distance d between the current and previously stored center is larger than a user-set value d_{min}. This ensures a limited number of triangles to be created. A result of a drop leaving a wet trail, is that it reduces in size because of the water molecules that cling to the surface. In order to simulate this behaviour, we alter the mass of the particles in the drop, based on how far the drop has traveled over the surface.

4 Results

We ran our simulation on a Pentium 4 Duo CPU 2.66 Ghz with 4 Gb RAM. We generated tears consisting of 100 particles and achieved an average frame rate of 36 fps (see Figure 1). For each eye, a particle source is defined. The position of these sources can be modified, to allow for tears flowing from either the middle or the side of the eye.

5 Conclusions

We have presented a real-time crying simulation framework, by using an extended SPH approach, optimized for crying fluid simulation. Our framework integrates with an existing facial animation system, and it is independent of the renderer and physics engine that is used. The shape of the fluid and the material used create a convincing simulation of tears. By adding a skin adhesion force, the motion of the tear neatly follows the skin without falling off. Finally, the addition of a wet trail greatly improves the realism of the tear.

Although we have set the first steps in the direction of creating a more expressive face, a lot of work still needs to be done. In this paper we have mainly focused on generating fluids to simulate tears, but our current method for simulation tear-skin interaction is limited. Second,

Figure 1: A few frames from a tear rolling over the character's face. Both the fluid and trail are clearly visible. Also note the reduction in tear volume over time.

extreme expressions such as laughing and crying are difficult to simulate convincingly without properly modeling muscle motions. In the future, we are going to look at what muscle motions are important in crying and laughing animations in order to improve our simulation, while retaining the real-time constraint.

References

Ax, A. F. (1952). The physiological differentiation between fear and anger in humans. *Psychosomatic Medicine*, 14,(5):433–442.

Borgquist, A. (1906). Crying. *American Journal of Psychology*, 17:149–205.

Chen, J. X., da Vitoria Lobo, N., Hughes, C. E., and Moshell, J. M. (1997). Real-time fluid simulation in a dynamic virtual environment. *IEEE Computer Graphics and Applications*, 17(3):52–61.

Egges, A. and Magnenat-Thalmann, N. (2005). Emotional communicative body animation for multiple characters. In *First International Workshop on Crowd Simulation (V-Crowds)*, pages 31–40.

Garchery, S. and Magnenat-Thalmann, N. (2001). Designing mpeg-4 facial animation tables for web applications. In *Multimedia Modeling 2001*, pages 39–59.

Lorensen, W. E. and Cline, H. E. (1987). Marching cubes: A high resolution 3d surface construction algorithm. In *SIGGRAPH '87: Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, pages 163–169, New York, NY, USA. ACM.

Monaghan, J. J. (1992). Smoothed particle hydrodynamics. *Annual review of astronomy and astrophysics*, 30:543–574.

Müller, M., Charypar, D., and Gross, M. (2003). Particle-based fluid simulation for interactive applications. In *SCA '03: Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 154–159, Aire-la-Ville, Switzerland, Switzerland. Eurographics Association.

Vingerhoets, A. J. J. M., Cornelius, R. R., van Heck, G. L., and Becht, M. C. (2000). Adult crying: a model and review of the literature. In *Review of General Psychology, Vol. 4, No. 4*, pages 354–377. Educational Publishing Foundation.

Zwicker, M., Pfister, H. P., van Baar, J., and Gross, M. (2001). Surface splatting. In *SIGGRAPH 2001*, pages 371–378.

Adaptive Behavioral Modeling for Crowd Simulations

Cagatay Turkay Emre Koc
Sabanci University, Turkey
{turkay|emrekoc}@su.sabanciuniv.edu

Kamer Yuksel
Technische Universitat Berlin / DAI-Labor, Germany
kamer.yuksel@dai-labor.de

Selim Balcisoy
Sabanci University, Turkey
balcisoy@sabanciuniv.edu

Abstract

In this study, we design an adaptive behavioral model for a dynamic virtual environment. We model the dynamic environment with behavior maps which are constructed with information theory quantities. These maps are capable of capturing the dynamic nature of the environment by changing temporally and spatially. Subsequent to building this model, agents' responses to these maps are represented with a set of formulations. In our test studies, we have observed that our model successfully produces realistic and diverse behaviors by incorporating effects of the environment.

Keywords: crowd simulation, information theory, behavioral modeling

1 Introduction

Realistic behavior of agents in crowd simulations is still a challenging problem due to the number of factors determining agents' behavior which are not easy to represent mathematically. In agent-based behavioral models, an agent responds to other agents and events using static and predefined behavior rules, whatever the environment conditions are. However, dynamic conditions which are inherent in the environment greatly effect an agents' behavior and existing models are not capable of adapting themselves to these conditions.

In order to model the effects of the crowd on agents, we need a quantification to represent the activities of the crowd and model the effects of the crowd on individual agents. A good model has to be adaptive to changes in the dynamics of the crowd both spatially and temporally. In our model, we develop *behavior maps* which convey information on probabilistic and statistical properties of agents' activities. Information theory quantities, i.e. *information entropy* and *Kullback-Leibler divergence* are used to produce behavior maps. As behavior maps are updated spatially and temporally; agents adaptively respond to their environment with the contribution of these maps. This contribution is represented with numerical entities and agents' responses are calculated with a set of formulations.

2 Related Work

There have been many studies on agent-based crowd models to create human-like behaviors. Seminal works of Reynolds used behavioral models considering local rules to create emergent flocking (Reynolds, 1987) behaviors. Pelechano et al. (2007) created an improved model by using psychological and geometrical rules with a social and physical forces model. There are studies which

model the virtual environment as maps to guide agents' behaviors. Shao and Terzopoulos (2007) modeled the environment with topological, perception and path maps to generate autonomous agents.

3 Behavior Maps

A behavior map spans over the virtual environment, and records all the agents' activities. This map, B, is a 2D grid, consisting of w rows and h columns. Physical properties of an agent, a_i, can be described as $a_i = \{u, \vec{v} \ : u, \vec{v} \in \mathbb{R}^2\}$ where u defines the position and $\vec{v}$ defines the velocity. The activity of an agent is described by its position, the direction and the magnitude of its velocity. Activities of an agent is mapped to the corresponding cell in B. We compute behavior maps by using probabilistic analysis methods incorporating quantities from information theory. *Information entropy* from information theory field (Shannon, 1948) provides an insight about how likely a system produces varied outcomes. Namely, it is a measure of uncertainty of a random variable. The other concept we have utilized is *Kullback-Leibler divergence* (KL) (Kullback, 1997) which is a non-symmetric metric expressing the difference between two probability distributions. Figure 1 displays behavior map construction.

We need to have probability mass functions (*pmf*) regarding to the activities in the scene to make use of information theory quantities. The first *pmf*, $P_{\hat{\vec{v}}}$, defines the velocity direction distribution. Consider an agent with normalized velocity $\hat{\vec{v}}$, then this vector is added as a sample to one of the n bins of $P_{\hat{\vec{v}}}$, where the value of n effects quantization resolution. The second *pmf*, $P_{\|\vec{v}\|}$, defines the velocity magnitude distribution. Agents' speed is quantized into m categories and speed of an agent is added as a sample to one of these categories.

3.1 Entropy Map

Entropy values represent behavioral patterns of the crowd. Entropy values denote whether agents move independently or in a pattern. To build the entropy map, E, we begin by considering a random variable, $X_{i,j}$ (i,j indicating location on E), drawn according to *pmf* $(P_{\vec{v}}{}^{(t-n\Delta t)\rightarrow t})_{i,j}$. Then, E can be defined as;

$$E^t = \{H(X_{i,j}) \ ; \ 0 \leq i < w, \ 0 \leq j < h\} \tag{1}$$

, where $H(X_{i,j})$ is the entropy of $X_{i,j}$.

3.2 Expectance Map

In order to quantize the expectance of the current activities on the scene, we compare the current probability distribution on the scene with $P_{\vec{v}}$. We employ Kullback-Leibler divergence to compute the difference between two probability distributions. KL calculations are called as *expectance maps*. Let $P_{\vec{v}}{}^t$ define the current probability distribution of the crowd, and $P_{\vec{v}}{}^{(t-n\Delta t)\rightarrow(t-\Delta t)}$ define the cumulative distribution of activities, expectance map KL is defined as;

$$KL^t = \{(D(P_{\vec{v}}{}^{(t-n\Delta t)\rightarrow(t-\Delta t)} \| P_{\vec{v}}{}^t))_{i,j} \ ; \ 0 \leq i < w, \ 0 \leq j < h\} \tag{2}$$

A high KL value represents that current activities taking place at that location can be regarded as *surprising*, while in areas with lower KL values, the current status of the crowd is as *expected*.

4 Response to Behavior Maps

In our crowd simulation engine, agents have internal properties called *behavioral constants* and *behavioral state*. Behavioral constants can be regarded as personality attributes of an agent and behavioral state determines at what level behavioral constants effect agent's behavior. Throughout the simulation, behavioral state is altered adaptively by behavior maps. These two properties modify agents' responses through certain formulations.

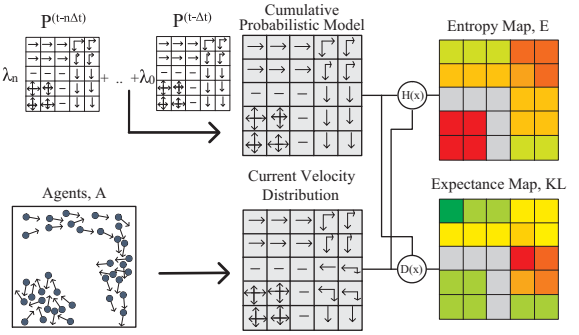


Figure 1: Behavior map construction.

To display diversity in agents' behaviors, we incorporate composite agents into our model. A composite agent is a special agent equipped with a proxy agent, r_i, to model a number of emergent behaviors realistically (Hengchin et al., 2008). Definition of an agent has to include internal properties, in addition to the physical properties. Definition of an agent a_i is extended as:

$$a_i = \{type, u, \vec{v}, \vec{v}_p, d, s, r_i, \delta, \langle f_1, .., f_n \rangle, \beta : \vec{v}_p \in \mathbb{R}^2; f_n, \beta, d, s \in [0, 1]\} \tag{3}$$

These parameters are: 1) *type*: Indicates whether an agent is composite agent or proxy agent. 2) $\vec{v}_p$: Indicates the velocity an agent prefers to move. 3) d: Distance to set between $r_i[u]$ and u. The longer the distance, the further a_i can proceed with less collisions. 4) s: Size of the area r_i occupies 5) δ: Indicates the range in which an agent considers possible collisions. 6) f_n: Indicates a behavior constant. Each constant can be utilized to mimic certain personality attributes. 7) β: Indicates the behavior map value.

Agents' responses to behavior maps are formulated as;

$$\begin{gathered} a_i[\beta] = kB_{i,j} \;;\; f_0^{'} = \beta f_0;\; f_1^{'} = \beta(1 - f_0) \\ a_i[\vec{v}_p] = (\widehat{\vec{v}_p^{\,o} + f_1^{'}\vec{v}_b}) f_0^{'} m_0 \\ a_i[d] = f_0^{'} d_0 \;;\; a_i[s] = f_0^{'} s_0;\; a_i[\delta] = 1/\beta\delta_0 \end{gathered} \tag{4}$$

$f_0^{'}$ value modifies the proxy agent r_i, as $f_0^{'}$ gets higher, d and s values are amplified. $f_0^{'}$ value also increases the speed of an agent in areas with high B value. $f_1^{'}$ value, on the other hand, amplifies the deviation from optimal velocity $\vec{v}_p^{\,o}$. The final response is the change in δ value that is inversely proportional to β values.

5 Results & Test Cases

We tested our methods through a number of scenarios. Our test environment is built on top of the modified version of multi-agent simulation system called RVO proposed by Van Den Berg et al. (2008). We create a crowd containing 200 agents. Crowd contains three groups of agents with specific behavioral constants. First of these groups constitutes of 20 agents with high f_0 and low f_1 values which can be considered either as *aggressive* agents. Second of these groups contain 20 agents with low f_0 and high f_1 values which can be considered as *calm* agents. The last group consist of 160 standard agents which do not display any adaptive behavior. Figure 2 displays snapshots from test scenarios.

In our tests with entropy maps, aggressive agents move directly to their goal, as their preferred velocity is not effected from higher β values. However, in areas with lower entropy all the agents behave identical and do not search for a more "safe" (collision-free) velocity in areas with low entropy, where δ values are lowered by β. In simulations that are run with KL maps; aggressive

Figure 2: Screenshot from our test environment.

agents clear their way aggressively and move directly to their goal in areas with high KL value. On the other hand, agents in the second group behave unexpectedly and this response mimics panicking behavior when an unexpected event happens. To address the general case for a dynamic virtual environment, we combined all three models into a single one. We calculate a weighted average response by adjusting the contribution of each behavior map.

6 Conclusion

In this paper, we proposed an adaptive behavioral model for crowd simulations. Our model incorporates the dynamics of a virtual environment through building an analytical model of crowd's activities and formulates agents' responses. We ran our model over a number of scenarios and observed that agents' behaviors are adaptively altered under certain environmental conditions. Results show that our methods add complexity and diversity in agents' behaviors, thus improve realism. These methods can be integrated into either scripted behavioral models to increase their behavioral variation or autonomous agent systems to improve their realism.

References

Hengchin, Y., Sean, C., Sachin, P., Jur, v. d. B., Dinesh, M., and Ming, L. (2008). Composite agents. In *Symposium on Computer Animation - SCA'08*.

Kullback, S. (1997). *Information Theory and Statistics (Dover Books on Mathematics)*. Dover Publications.

Pelechano, N., Allbeck, J. M., and Badler, N. I. (2007). Controlling individual agents in high-density crowd simulation. In *Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 99–108. Eurographics Association.

Reynolds, C. (1987). Flocks, herds and schools: A distributed behavioral model. In *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, pages 25–34.

Shannon, C. E. (1948). A mathematical theory of communication. *Bell Systems Technical Journal*, 27:623–656.

Shao, W. and Terzopoulos, D. (2007). Autonomous pedestrians. *Graphical Models*, 69(5-6):246–274.

Van Den Berg, J., Lin, M., and Manocha, D. (2008). Reciprocal Velocity Obstacles for real-time multi-agent navigation. In *Robotics and Automation, 2008.*, pages 1928–1935.

An Animation Framework for Continuous Interaction with Reactive Virtual Humans

H. van Welbergen, D. Reidsma, J. Zwiers, Zs. Ruttkay, and M. ter Maat
Human Media Interaction - University of Twente
P.O. Box 217, 7500 AE Enschede
{welberge|dennisr|zwiers|ruttkay|maatm}@ewi.utwente.nl

Abstract

We present a complete framework for animation of Reactive Virtual Humans that offers a mixed animation paradigm: control of different body parts switches between keyframe animation, procedural animation and physical simulation, depending on the requirements of the moment. This framework implements novel techniques to support *real-time continuous interaction*. It is demonstrated on our interactive Virtual Conductor.

Keywords: Virtual Humans, Interactivity, Gesture Generation, Conducting Motion

Figure 1: The Virtual Conductor, Photo: Henk Postma, Stenden Hogeschool

1 Introduction

Virtual Humans (VHs) often interact with users using a combination of speech with gestures in a conversational setting. They tend to be developed using a turn-based interaction paradigm in which the interlocutor and the VH take turns to talk (Thiebaux et al., 2008). If the interaction capabilities of VHs are to become more human-like and they are to function in social settings, their design should shift from this turn-based paradigm to one of *continuous interaction* in which all partners in an interaction perceive each other and express themselves continuously and in parallel (Nijholt et al., 2008). We present in this paper the design and implementation of a framework for building Reactive Virtual Humans (RVHs) that are capable of exhibiting this kind of continuous interaction. Continuous interaction needs an immediate adaptation to external events (in the environment and from the user). This requires re-timing of already planned behavior to match with these events, and re-planning or re-scheduling of the planned behavior on short notice.

In our previous work we have introduced mixed paradigm animation using procedural motion on selected body parts and physical simulation on the remaining body parts (van Welbergen et al., 2009), which allows us to combine the physical integrity of physical simulation with the precision of procedural animation. In this paper we show the applicability of physical simulation for *secondary* motion, that is, motion such as balancing or eye blinking that one wants the VH to display, but does not want to have to specify in detail. We present the implementation of *switches* between physical or kinematic control of motion on different joints, depending on the focus of the animation task at any moment.

We explain the design and implementation of the architecture using our implementation of a Virtual Conductor (Reidsma et al., 2008) that can interactively conduct an ensemble of human musicians, listen to the music they play, and reactively adapt its conducting behavior and the timing thereof when the musicians need to be corrected (See Figure 1).

2 Architecture of Our Animation Framework

We base our architecture (Figure 2) on the SAIBA Framework (Kopp et al., 2006), which contains a three-stage process: *communicative intent planning*, multi modal *behavior planning*, resulting in a BML stream, and *behavior realization* of this stream. Our architecture encompasses the behavior planning and realization stages. A feedback loop between these two stages allows flexible (re)planning of behavior. We zoom in on our implementation for the planning and realization of animation of our system (see (Nijholt et al., 2008) for its initial design).

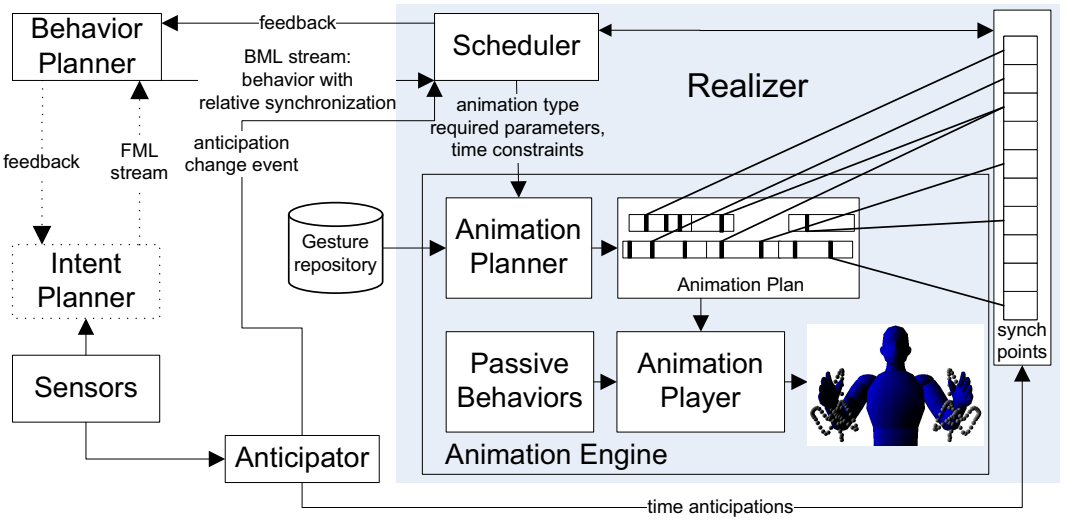


Figure 2: Architecture

The realization stage starts with a specification, in a BML stream, of a set of 'behaviors' on different modalities, synchronized to each other and possibly to predicted/anticipated world events. For example: the BML stream could specify that the timing of conducting gestures is initially determined by *synchronization points* derived from the tempo indications in the music score. Interaction with the world is achieved through *Anticipators*. An Anticipator is a module that takes as input perceptions of events in the real world, extrapolates them into predictions of the timing of future events, and uses these predictions to update the timestamps of synchronization points. The conducting anticipator, for example, sets – and dynamically updates – the exact timing with which the conducting beats should actually be executed, making use of the score of the piece (intended tempo), the detected tempo of the music played by the musicians and knowledge on how to make musicians play on time (Reidsma et al., 2008). BML$^{\mathrm{T}}$, our extension of BML, allows the specification of alignment to events from an Anticipator.

2.1 The Organization of Motion

We organize motion in *motion units*. A motion unit has a predefined function (for instance: a 3-beat conducting gesture, a walk cycle) and acts on a selected set of joints. A set of parameters can be used to adapt the motion unit (for instance: amplitude for the conducting motion unit, or desired pelvis height and joint stiffness for a physical balancing motion unit). Motion units contain one or more *motion phases*, separated by *keys*. Each key is assigned a predefined canonical time value $0 \leq \alpha_i \leq 1$ that indicates where it is located within the motion unit. Given the current set

of parameter values and some canonical time $0 \leq \alpha \leq 1$, a motion unit can be executed, typically by rotating some joints of the RVH.

Procedural motion units rotate joints over time as specified by mathematical expressions that take α as well as a vector $\mathbf{a} \in \Re^n$ as parameters. These expressions can be used to directly steer Euler angles components of joint rotation, to position the root or to position the wrists and ankles using analytical inverse kinematics. The parameter values $\mathbf{a}$ can be changed in real time, changing the motion shape or timing. All mathematical expressions are evaluated using the Java Math Expression Parser.[1] Arbitrary custom function macros can be designed. We defined such macros for hermite splines, TCB splines (Howe et al., 2006) and perlin noise. Our design allows arbitrary mathematical formulas and parameter sets to be used for motion specification and is therefore more flexible than, but still compatible with traditional procedural animations models that define motion in terms of splines or other *predefined* motion formulas and use *fixed* parameter sets (Chi et al., 2000; Howe et al., 2006). In our work, keyframe animation is regarded as a specialized procedural motion unit.

Secondary motion units, such as eye blinking or balancing, are activated activated using the BML stream. Secondary physical motion units are executed by physical controllers. We have implemented several balancing controllers, that offer different trade-offs between balancing stability and movement naturalness (van Welbergen et al., 2009). Our pose controllers loosely keep body parts in their desired position, while still being effected by the forces on the body. Controllers use techniques from control theory to steer the VH's 'muscles' in real time. The input to such a controller is the desired value of the RVH's state, for example desired joint rotations or the desired position of the VH's center of mass (CoM). Such controllers can, to a certain extent, cope with external perturbation and move the body using Newtonian dynamics, taking friction, gravity, and collisions into account.

Transition motion units are used to automatically create movement from the pose at which they are activated to a predefined pose. Currently this is done using a simple slerp interpolation.

2.2 Motion Planning, Re-planning and Execution

The animation planner creates instances of motion units (called *timed motion units*) and inserts them in the *Animation Plan*, as specified by the scheduler. Timed procedural motion units are instantiated from a gesture repository. Secondary motions are enabled and disabled as prescribed by the BML stream.

We infer the continuously changing mix of kinematically and physically steered joints from the active procedural motion units and secondary motions. A *switch* from kinematical to physical control on a body part is implemented by setting up the appropriate physical representation and applying the current velocity and position to the matching body parts in the new physical representation. A switch from the physical to kinematic control simply removes the physical representation of the body part from the physical body of the RVH. For example, when the conductor just indicates the beat, he conducts with his right hand lets the left hand hang down loosely. This is implemented using a physical representation of the lower body and left arm, steered by respectively a balancing controller and a pose controller. The right arm is steered by a procedural conducting animation. To use the left arm for an expressive conducting gesture, we disable the pose controller and plan the expressive gesture as a procedural motion. This automatically executes a switch, removing the physical representation of the left arm from the physical body.

The keys of the timed motion units are linked to the synchronization points in the realizer, as specified in the BML. Synchronization between keys in different timed motion units is achieved by linking them to the same synchronization point.

The Anticipator notifies the *Scheduler* whenever its predictions change, and updates the synchronization points within the Realizer. Many of such updates are minor and do not require a change in the Animation Plan. Since the keys of timed motion units are symbolically linked to

[1] Singular Systems, `http://sourceforge.net/projects/jep/`

the synchronization points, the timing update is handled automatically. More significant prediction updates might require an update of the Animation Plan, which is handled by the Animation Planner and the Scheduler. Such an update typically involves re-timing of behavior on several modalities to generate a more natural behavior execution plan, as suggested in Nijholt et al. (2008). If the Scheduler can not generate a (multi modal) execution plan that satisfies the new time predictions, the Scheduler omits the behaviors that cannot be executed and notifies the Behavior Planner. It is then up to the Behavior Planner to update the behavior plan.

The animation player executes the active timed procedural motion units. The generated motion is then combined with the currently enabled secondary motions, using our system that mixes motion on physically steered body parts with (procedural) motion on kinematically controlled body parts, taking the forces generated by the kinematically steered joints into account (van Welbergen et al., 2009).

3 Results and Discussion

We presented a complete framework for animation of Reactive Virtual Humans that implements novel techniques to support tightly synchronized real-time continuous interaction using a mixed animation paradigm that switches the control of different body parts between procedural animation and physical simulation, depending on the requirements of the moment. The system offers an adjustable balance between ease-of-use and flexibility by allowing motion specification through both high level behavioral primitives and (at the same time) a detailed specification on those aspects for which the user needs it. Some demonstration movies of our mixed paradigm animation and procedural motion system can be found online.[2]

Acknowledgments

This research has been supported by the GATE project, funded by the Dutch Organization for Scientific Research (NWO) and the Dutch ICT Research and Innovation Authority (ICT Regie).

References

Chi, D. M., Costa, M., Zhao, L., and Badler, N. I. (2000). The EMOTE model for effort and shape. In *SIGGRAPH*, pages 173–182, New York, USA. ACM Press/Addison-Wesley Publishing Co.

Howe, N. R., Hartmann, B., Leventon, M. E., Mancini, M., Freeman, W. T., and Pelachaud, C. (2006). Implementing expressive gesture synthesis for embodied conversational agents. In *Gesture in Human-Computer Interaction and Simulation*, volume 3881 of *LNCS*, pages 188–199. Springer.

Kopp, S., Krenn, B., Marsella, S., Marshall, A. N., Pelachaud, C., Pirker, H., Thorisson, K. R., and Vilhjálmsson, H. H. (2006). Towards a common framework for multimodal generation: The behavior markup language. In *IVA'06*, volume 4133 of *LNCS*, pages 205–217. Springer.

Nijholt, A., Reidsma, D., van Welbergen, H., op den Akker, H. J. A., and Ruttkay, Z. M. (2008). Mutually coordinated anticipatory multimodal interaction. In *Nonverbal Features of Human-Human and Human-Machine Interaction*, volume 5042 of *LNCS*, pages 70–89, Berlin. Springer.

Reidsma, D., Nijholt, A., and Bos, P. (2008). Temporal interaction between an artificial orchestra conductor and human musicians. *Computers in Entertainment*, 6(4):1–22.

Thiebaux, M., Marshall, A. N., Marsella, S., and Kallmann, M. (2008). Smartbody: Behavior realization for embodied conversational agents. In *Autonomous Agents and Multiagent Systems*, pages 151–158.

van Welbergen, H., Zwiers, J., and Ruttkay, Z. (2009). Real-time animation using a mix of dynamics and kinematics. *Submitted to Journal of Graphics Tools*.

[2] http://hmi.ewi.utwente.nl/casa09_files/demo_conductor/mixed.html

Extracting Reusable Facial Expression Parameters by Elastic Surface Model

Ken Yano
Graduae School of Engineering
Hiroshima University
1-4-1 Kagamiyama
Higashi Hiroshima, Hiroshima, Japan
d064016@hiroshima-u.ac.jp

Koichi Harada
Graduae School of Engineering
Hiroshima University
1-4-1 Kagamiyama
Higashi Hiroshima, Hiroshima, Japan
harada@mis.hiroshima-u.ac.jp

Abstract

We introduce a novel parameterization of facial expressions by using elastic surface model. The elastic surface model has been used as a deformation tool especially for nonrigid organic objects. The parameter of expressions is retrieved from existing reference face models. The obtained parameter can be applied on target face models dissimilar to the source model to make similar expressions. The parameterization can be easily adapted to facial animation by shape blending and also can be used for realtime key-framed facial animation system.

Keywords: facial animation, facial parameterization, expression cloning

1 INTRODUCTION

Mesh deformation plays a central role in computer modeling and animation. Animators sculpt facial expressions and stylized body shapes. Despite the tremendous amount of artistry, skill and time dedicated to crafting deformations, there are few techniques to help with reuse.

Expression cloning (Yong and Ulrich, 2001) reuses the dense 3D motion vectors of the source model to create similar animations on a new target model. Animation of completely new characters can be based on existing libraries of high-quality animations created for many different models. In (W.Summer and Popovic, 2004), deformation of a source mesh is transfered to a target mesh through the correspondence map between the two models by solving an optimization problem. In this paper, a novel parameterization of facial expressions is introduced. The parameters can be learned from existing reference face models or created from scratch. The obtained parameters can be applied on target face models dissimilar to the source model to generate similar expressions on them

2 ELASTIC FACIAL SKIN MODEL

In our proposed method, a facial skin is assumed to behave like a physical skin that stretches and bends as forces are acting on it. Mathematically this behavior can be captured by the energy functional that penalizes both stretching and bending. Let d is the displacement function defined on the surface and k_s and k_b are the parameters to control the resistance to stretching and bending respectively, the elastic energy E is defined as;

$$E(d) = \int_{\Omega} k_s(\| d_u \|^2 + \| d_v \|^2) + k_b(\| d_{uu} \|^2 + 2 \| d_{uv} \|^2 + \| d_{vv} \|^2) du dv \quad (1)$$

where the notations d_x, d_{xy} are defined as $d_x = \partial d/\partial x$, $d_{xy} = \partial^2 d/\partial x \partial y$.

In a modeling application one would have to minimize the elastic energy in Equation 1 subject to the user-defined constraints. By applying variational calculus, the corresponding Euler-Langrange equation that characterizes the minimizer of Equation 1 can be expressed as

$$-k_s \triangle_s d + k_b \triangle_S^2 d = 0 \tag{2}$$

The Laplacian operator in 2 corresponds to the Laplace-Beltrami operator (Taubin, 1995). Using the famous cotangent discretization of the Laplace operator, the Euler-Lagrange PDE turns into a sparse linear system:

$$\begin{aligned} -k_s \triangle d(p_i) + k_b \triangle^2 d(p_i) &= 0, \qquad p_i \notin H \cup F, && \text{(3a)} \\ d(p_i) &= d_i, \qquad p_i \in H, && \text{(3b)} \\ d(p_i) &= 0, \qquad p_i \in F, && \text{(3c)} \end{aligned}$$

Where H is the handle vertices and F is the fixed vertices. In general, the order k of Laplacian operator corresponds to the C^{k-1} continuity across the boundaries. For the facial skin deformation, we use the pure bending $(k_s = 0, k_b = 1)$ surface model because the model can retain the C^1 continuity around the handle vertex H which is proved to be a good approximation of the skin deformations of various expressions from our test results.

3 Facial parameter estimation

The facial parameters are calculated so that parameters obtained are precise enough to approximate the facial deformation due to expressions. In order to compute facial parameters, reference face models with neutral and a set of expressions are required. To match up every vertices, the models must share the same number of vertices and triangles, and have identical connectivity. The equation 3 can be expressed in matrix form as follows by transferring the columns which correspond to the constrained vertices to the right hand side (LHS) of Equation 3

$$Ld(p) = \sum_{i=1}^{m} b_i \times d(p_i), \qquad p_i \in H \tag{4}$$

b_i is the column vector of matrix L that corresponds to handle vertex p_i of Equation 3. The solution of Equation 4 can be explicitly expressed in terms of the inverse matrix L^{-1}. We observe that the explicit solution of Equation 4 is

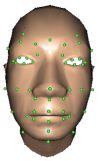
$$\begin{aligned} d(p) &= L^{-1} \sum_{i=1}^{m} b_i \times d(p_i) && \text{(5a)} \\ &= L^{-1} b_1 \times d(p_1) + L^{-1} b_2 \times d(p_2) + \ldots + L^{-1} b_m \times d(p_m) && \text{(5b)} \\ &= B_1 \times d(p_1) + B_2 \times d(p_2) + \ldots + B_m \times d(p_m) && \text{(5c)} \\ & && \text{(5d)} \end{aligned}$$

where m is the number of handle points and B_i is the basis functions which is uniquely defined from the mesh structures. Note that the basis functions $\{B_i\}$ can be precomputed once the handle points are fixed and they can be reused for all the expressive face models of the same subject. The matrix L is symmetric positive definite and we solve the linear system by using direct method (TAUCS, 2001). As a precomputation, we apply Cholesky factorization to the matrix L and the result of the factorization is used to compute each basis function. Once all the basis function are computed, the left hand side of Equation 5 is computed by subtracting the neutral face from the corresponding expressive face model. The facial parameter $\mathbf{FP_i}$, e.g., three dimensional displacement vector at each handle vertex, is computed by solving Equation 6 using Least Square method for each coordinate, namely x, y, and z.

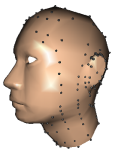
$$\mathbf{d}(p) = \sum_{i=1}^{m} B_i \times \mathbf{FP_i} \tag{6}$$

In order to obtain the basis functions $\{B_i\}$, handle region H corresponding to the facial control points need to be defined. We adopt a subset of facial control points defined in MPEG-4 standard (ISO, 2003), which are distributed symmetrically over a entire front face. The total number of control points is forty-seven for our test model and they are shown in Figure 1 (a). The fixed region F is empirically defined on the vertices which are static under the change of expressions. In order to search for the fixed vertices, we let R to be the Euclidean distance between the tip of the nose and the center of the forehead, then if the Euclidean distance r between the vertex and the tip of nose is greater than the threshold value defined as $1.5R$ (for our test models), we set the vertex as fixed and put it in the fixed region F.

Figure 2 shows the generated face models along with the original models.

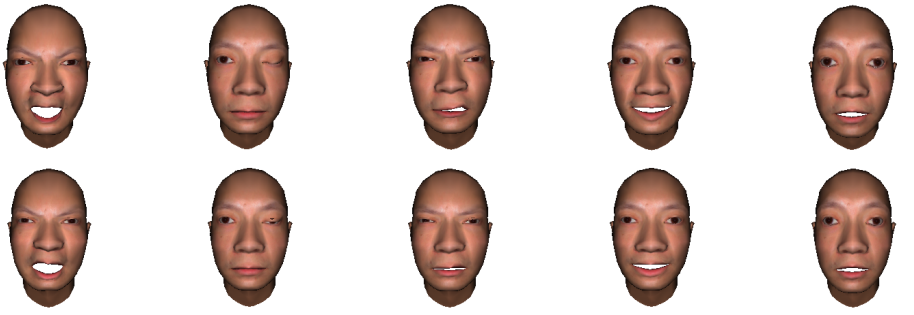


(a) control points (handle region H)



(b) fixed points (fixed region F)

Figure 1: The control points are a subset of MPEG-4 FPs(feature points) and some additional points. The total number of control points is empirically chosen as forty-seven. Fixed region is also defined on the undeformed facial area and head.



(a) anger (b) blink (c) disgust (d) smile (e) surprise

Figure 2: Extraction of facial parameters. First row shows the original face model. Second row shows expressions reconstructed using the facial parameters extracted from the original model

Facial expression blending is a common technique for facial animation to create a novel expression by blending existing expressions. Given the set of facial parameters $\{\mathbf{FP_i}\}$ generated for each expression, a novel expression can be created by simply blending the facial parameter for each expression.

$$\mathbf{d}(p) = \Sigma_k w_k \begin{pmatrix} B_1 & B_2 & \dots & B_m \end{pmatrix} \begin{pmatrix} \mathbf{FP_1}^k & \mathbf{FP_2}^k & \dots & \mathbf{FP_m}^k \end{pmatrix}^T \tag{7a}$$

$$= \begin{pmatrix} B_1 & B_2 & \dots & B_m \end{pmatrix} \begin{pmatrix} \Sigma w_k \mathbf{FP_1}^k & \Sigma w_k \mathbf{FP_2}^k & \dots & \Sigma w_k \mathbf{FP_m}^k \end{pmatrix}^T \tag{7b}$$

where $\{w_k \mid 0.0 \leq w_k \leq 1.0\}$ is the blending weight for each expression.

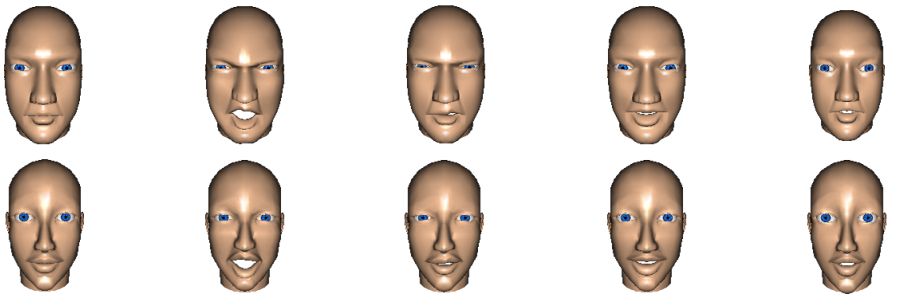
4 Facial Expression Cloning

Expression cloning is a technique that copies expressions of a source face model onto a target face model. The mesh structure of the models need not to be the same.

The first step selects the facial control points on the target model, they have to exactly correspond to the control points on the source model. It takes no more than twenty minutes to select all facial control points on the target model. The second step computes the basis functions $\{B'_i\}$ for the target model as we did in

the previous section. The time took for computing the basis functions depends on the complexity of the target model. In the third step, we copy the expressions on the target model. Given the facial parameters $\{\mathbf{FP_i}\}$ for each expression of the source model and the basis functions $\{B_i'\}$ obtained from the target model, the displacement vectors for each expression of the target model are computed by using Equation 6.

Note that in order to compensate for the scale difference between the source and the target model, the facial parameters $\{\mathbf{FP_i}\}$ are normalized such that each vector $\mathbf{FP_i}$ is divided by the scalar,e.g., FAPU (Facial Action Parameter Unit). FAPU is commonly set as the distance between the inner corners of the eyes of the face model. We also assume that models are aligned so that the y axis points through the top of head , x axis points through the left side of head and looking in the positive z axis direction. If target models are not aligned with source model, they are aligned before deformation is applied then moved back to the original alignment after the deformation. Figure 3 shows the results of expression cloning. The expressions on the source model are copied on three different target models dissimilar to the source mode.



(a) neutral (b) anger (c) disgust (d) smile (e) surprise

Figure 3: Facial parameters extracted from the reference face models are applied to target models with neutral expression to generate similar expressions on them.

5 Conclusion

In this paper, novel facial parameters are learned from existing reference face models with various expressions by assuming that each expressions is generated by elastic surface. The obtained facial parameters can be applied on other target face models even if the mesh structure (number of vertices, number of triangles, and connectivity) is different from the source model. The size of the total parameter depends on the number of handle points defined on facial surface. With only forty-seven control points, various expressions can be duplicated on target models as shown in the results. Our facial parameter can be easily adapted to the shape blending and possibly used for realtime key-framed facial animation system.

Proposed parameterization of facial expressions enables the user to reuse existing expression data. The time took to select control points and to compute basis functions on target models can be paid off by exact duplication of expressions from the reference model thus greatly lesses the skills needed for making expressive face models from scratch.

References

ISO (2003). Mpeg-4 international standard, moving picture experts group.

Taubin, G. (1995). A signal processing approach to fair surface design. In *Proceeding of the 22nd annual conference on Computer graphics and interactive techniques*, pages 351 – 358.

TAUCS (2001). Taucs: A library of sparse linear solvers. http://www.tau.ac.il/ stoledo/taucs/.

W.Summer, R. and Popovic, J. (2004). Deformation transfer for triangle meshes. In *Proceedings of International Conference on Computer Graphics and Interactive Techniques*, pages 399 – 405.

Yong, J. and Ulrich, N. (2001). Expression cloning. In *Proceeding of 28th annual conference on Computer graphics and interactive techniques*, pages 277–288.

Poster Papers

CASA 2009

Phoneme-level External and Internal Articulator Dynamics for Pronunciation Learning

Hui Chen Lan Wang Jian-Jun Ouyang Yan Li Xiao-Hua Du
Shenzhen Institute of Advance Integration Technology,
Chinese Academy of Sciences / The Chinese Univ. of Hong Kong, SIAT, Shenzhen, China
{hui.chen, lan.wang, yan.li, xh.du}@siat.ac.cn, jj.ouyang@sub.siat.ac.cn

Abstract

A low cost data-driven three-dimensional talking head is established. External and internal articulations are defined and calibrated from video streams and videofluoroscopy to a common head model. Confusable phonetic pairs are displayed to validate the effectiveness of distinguishing the articulations between English phonemes.

Keywords: internal articulator movements, computer-assisted pronunciation training, talking head

1 INTRODUCTION

Recent studies [1] have shown that given additional visual information of internal articulator dynamics can do improve the interpretation of speech production, affect speech comprehension and improve the pronunciation skill. The videofluoroscopic images of X-rays [2] was used to record the movements of tongue body, lips, teeth, mandible, and uvula etc. in real-time. The individual articulators on the images are not easy to identify, and the speakers maybe put into hazards by X-ray radiation during the long-term recording. However this approach was in principle the most appropriate method to determine the actual tongue shapes of articulations [1]. While in Electro-Magnetic Ariculography (EMA) recording method [3], receptor coils are attached to the subject's tongue or lips to acquire internal motions. However EMA has the drawback with poor spatial resolution related to the limited number of points, besides the cost of the device and motion capturing is high.

In this work, a low cost data-driven three-dimensional talking head was established to reveal the phoneme-level speech production. The external and internal articulators were recorded by two video streams and videofluoroscopy. Articulator movements of each phoneme were then traced and calibrated with the established 3D head model. Three different deformable modes in relation to pronunciation characteristics of different articulators are integrated. Time warping and shape blending functions according to natural speech input are synthesized in an utterance. A set of confusable phonetic pairs were displayed to validate the effectiveness of the proposed talking head model.

2 PHONEME-LEVEL EXTERNAL AND INTERNAL ARTICULATOR DYNAMICS

The system outline consists of a preprocessing stage constructing speech articulators' feature database and an on-the-fly stage with talking head animations shown in Figure 1.

2.1 CONSTRUCTION OF PHONEMES FEATURE DATABASE

To derive articulator motions of all phonemes in American English, a native speaker is invited to pronounce phonemes, words and sentences respectively. Facial articulator movements are recorded synchronously by VCRs set from front and lateral views. Internal articulator movements are segmented from existed videofluoroscopy of X-ray films (http://psyc.queensu.ca/~munhallk/05_database.htm). Considering the dynamics of mouth and tongue, all the 45 phonemes in International Phonetic Alphabet are selected as the basic units of visual speech pronunciations.

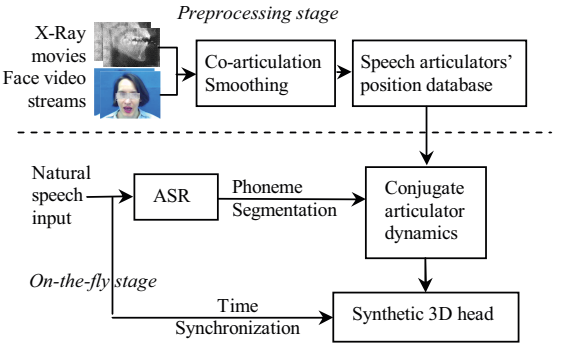


Figure 1. The flowchart of synthetic 3D talking head for a pronunciation training system.

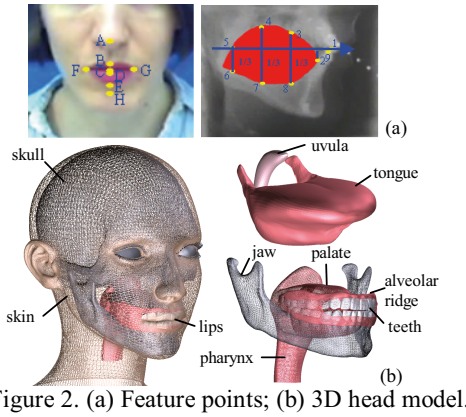


Figure 2. (a) Feature points; (b) 3D head model.

Phoneme segmentation is generated by automatic speech recognition (ASR) system in all video streams for synchronization. Feature points of facial appearance on front and lateral views include six points on the lips, nose tip, and middle chin point; while internal features are seven points on the tongue, upper, and lower teeth in Figure 2(a). A familiar three-dimensional talking head model in Figure 2(b) is established based on the templates of MRI slices. Articulators of lips, jaw, teeth, palate, alveolar ridge, tongue, uvula and pharynx have been recreated. Accordingly feature positions on 3D head model are evaluated and adjusted via registration of all above images.

2.2 PHONEME-LEVEL DYNAMICS OF ARTICULATORS

Dynamic pronunciation movements require the cooperation of all articulators.The lips and tongue are both muscular hydrostats in that they are composed entirely of soft tissue and moved under local deformation. Let feature points move under displacements of each phoneme, constraint deformations are applied to the adjacent points on facial skin or tongue. An elliptic adjacent area is defined via distance function, and multiple ellipses can easily approximate to the shape of lips or tongue with given feature points. Therefore, the points inside the adjacent area of feature point are deformed under cosine function, while the points outside the influence keep constant. The deformation of jaw and linked lower teeth & chin skin is mainly controlled in six degrees of freedom. The position relative to the skull are defined with three orientation angles of yaw, pitch, &roll and three displacements of horizontal, vertical, &lateral. The skull and linked upper teeth and facial skin are set as fixed constraint during pronunciation, while the movement of the fixed part can be used to simulate the head motion during speech.

The overall animation of multi articulators are integrated into a single display. Since the video of facial appearance and videofluoroscopy of internal organs are recorded separately, individual actions of external and internal articulators are combined based on synchronization of phone waveform segmented among start/peak/end frames of lips and tongue. Time warping is applied to transfer the time intervals of each phoneme from videofluoroscopy to accordingly videos. Given natural speech input utterance, ASR is conducted to generate the phone sequences as linguistic constrains for coarticulation synchronization. Words or sentences speech of talking head are then synchronized via linear shape blending of motion vectors between individual phonemes. Movements span peaks of the involved phonemes from the start to the end pronunciation of words or sentences, during which they ramp up gradually at the start of speech and fall off gradually at the end. The mean square errors of feature points at the silence interval during speaking are computed to recalibrate the reference stage of talking head.

3 EXPERIMENT

The confusable phonemes (/s/ *vs.* /th/, /eh/ *vs.* /ae/) and words (sing *vs.* thing, het *vs.* hat) are tested. Figure 3 is the synchronized speaking talking head of phonemes /s/ *vs.* /th/, /ae/ *vs.* /eh/. We can observe that the motions of internal tongue are of large difference such as between /s/ and /th/. The differences of the confusable phonemes are shown more clearly with the additional internal tongue movements thus can help in pronunciation learning.

4 CONCLUSION

A data-driven 3D talking head with external and internal articulator movements has been established and tested in this work. In future, we will focus on the learning results of foreigners with the proposed system.

5 ACKNOWLEDGEMENT

This work is supported by the National Nature Science Foundation of China (No. 60703120 & No. 60772165).

REFERENCES

[1] Tarabalka, Y., Badin, P., Elisei, F., Bailly, G. (2007). Can you read tongue movements? Evaluation of the contribution of tongue display to speech understanding, *in ASSISTH2007*, pages: 187-193, 2007.

[2] Tye-Murray, N., Kirk, K.I., Schum, L. (1993). Making typically obscured articulatory activity available to speechreaders by means of videofluoroscopy, *NCVS Status and Progress Report*, 4: 41-63, 1993.

[3] Fagel, S., Clemens, C. (2004). An articulation model for audiovisual speech synthesis-determination, adjustment, evaluation. *Journal of Speech Communication*, 44:141-154, 2004.

Figure 3. Synchronized speaking talking head of phonemes: left: /s/ *vs.* /th/, right: /ae/ *vs.*

Example Based Caricature Synthesis

Wenjuan Chen[1)], Hongchuan Yu[2)], Jianjun Zhang[2)]
1) Animation school, Communication University of China ,beijing cwj@cuc.edu.cn
2) National Centre for Computer Animation, Bournemouth University, United Kingdom
{hyu, jzhang}@bournemouth.ac.uk;

ABSTRACT

The likeness of a caricature to the original face image is an essential and often overlooked part of caricature production. In this paper we present an example based caricature synthesis technique, consisting of shape exaggeration, relationship exaggeration, and optimization for likeness. Rather than relying on a large training set of caricature face pairs, our shape exaggeration step is based on only one or a small number of examples of facial features. The relationship exaggeration step introduces two definitions which facilitate global facial feature synthesis. The first is the *T-Shape* rule, which describes the relative relationship between the facial elements in an intuitive manner. The second is the so called *proportions*, which characterizes the facial features in a proportion form. Finally we introduce a similarity metric as the likeness metric based on the Modified Hausdorff Distance (MHD) which allows us to optimize the configuration of facial elements, maximizing likeness while satisfying a number of constraints. The effectiveness of our algorithm is demonstrated with experimental results.

Keywords: Caricature Synthesis, Exaggeration, Likeness, Modified Hausdorff Distance, Texture Style Transferring

1 INTRODUCTION

This paper presents a new technique for the synthesis of novel human face caricatures learning from existing examples. There are three elements essential to caricatures: exaggeration, likeness, and statement [1]. A caricaturist must decide *which* features to exaggerate, and the *scale* of the exaggeration. The likeness emphasizes the visual similarity of the caricature to the subject. Statement allows the artist to add some personality to the subject by editorializing the caricature. Statement is an artistic process and cannot be emulated by a computer. In this paper we address exaggeration and likeness with the aim to create exciting caricatures by learning from available examples.

Example based learning methods usually need a large training set from a particular artistic tradition, such as [2]. In practice, however, it is impossible to get a large training set of caricatures that have the same style or from the same artist. Commonly only a small number of caricatures from the same caricaturist or the same artistic tradition are available, making these conventional example-based learning approaches ineffective.

Facial features (e.g. facial contour, eyes and nose etc.) are essential elements of a caricature. Different caricaturists and artistic traditions draw them differently which give caricatures a distinct style. Therefore a new caricature can be created by taking these individual elements from several caricature examples. For instance, one may want to exaggerate a face with a narrow facial contour and short nose. If both features are present in different examples, the solution is to pick up the necessary features from the respective example caricatures. However, because the facial features are from different examples, harmonious arrangement of these features is essential.

2 ALGORITHM

- **Shape exaggeration:** The shape exaggeration of individual face elements is computed based on only one or a small number of examples. Let $\{X_0, X_0^*\}$ be a given face image-caricature pair, where X_0 denotes the original natural face while X_0^* denotes the caricatured one. In terms of the training set $\{X_i\}$, one can build an eigenface space. It is therefore expected to build a mapping between $X_i^* - X_0^*$ and $X_i - X_0$,which is described as follows:

$$X_i^* - X_0^* = U_k[\lambda_k]U_k^T(X_i - X_0), \quad (1)$$

where $[\lambda_k]$ denotes the approximation coefficients in a diagonal matrix form. We formulate this problem of seeking λ over the training set $\{X_i\}$ as a minimization problem with respect to λ as follows,

$$\min_{\lambda} \sum_{i=0}^{n} \left\| X_0^* - U[\lambda]U^T X_i \right\|^2 . \quad (2)$$

Once λ is yielded, one can select the first k principal components to compute the deformed X_i^* with Eq.(1).

• **Relationship Exaggeration:** Most present approaches exaggerate the difference from the mean[3], although the results of this approach have often been criticized. In fact caricaturists tend to emphasize only one or two salient features in a caricature[4]. In this paper, the proportion description of the features is introduced. Normalizing the proportion differences of a feature by using its mean is viewed as an expression of the feature distinctiveness.

Caricatures think that all the facial features relate to one another fundamentally, and we cannot make a change to one feature without it affecting the others. This is one of the few constants you can rely on with respect to drawing caricatures: Action and Reaction**.** The T-Shape rule [1] are utilized to exaggerate the relationships between the facial features, which can be stated as follows: if the eyes move apart from each other, the nose should be shortening; whereas, if the nose is lengthening, the eyes should move closer to each other. It proves both simple and intuitive.

• **Likeness:** In existing methods, "likeness" was seldom considered for caricature synthesis due to lack of a "likeness" metric. We introduce the Modified Hausdorff Distance (MHD) [5]to measure the visual similarity. Based on this metric the likeness is incorporated into the integral caricature by optimizing the configuration of the facial elements, ensuring the resulting caricature resembles the original subject.

Some line-drawing caricatures are produced by three steps above : the shape exaggeration, relationship exaggeration and likeness optimization shown in Fig 1.

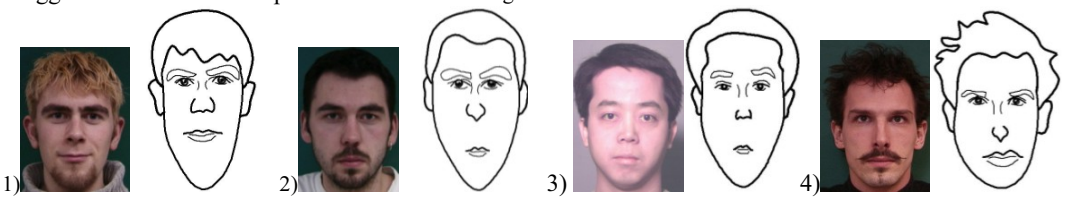


Fig. 1. The generated caricatures using our approach.

3 CONCLUSION

In this paper, we present an example based caricature synthesis approach. Unlike other published approaches, our new shape exaggeration method is based on only one or a few examples of facial components. So it's easy to change caricature style. Regarding the measurement of face likeness, there was little work done in the area of caricature synthesis. Our MHD based similarity metric definition attempts to tackle this issue.

However, there remain a number of issues in our current development, which will be investigated in the future. The hair style and head shape have not been considered due to the problem of hair occlusion. In addition, we will also try to incorporate different texture styles, such as watercolor and pencil sketch, a process known as *texture style transferring*. Combining a caricature with versatile texture styles is likely to produce more fantastic exaggeration effects.

REFERENCES

[1] RICHMOND T.: How to draw caricature. http://www.tomrichmond.com, 2008

[2] LIANG L., CHEN H., XU Y. Q. and SHUM H. Y.: Example-based Caricature Generation with Exaggeration. *In Proc. of 10th Pacific Conf. on Computer Graphics and Applications* (2002), China, pp. 386–393.

[3] BRENNAN S. E.: Caricature Generator: The Dynamic Exaggeration of Faces by Computer. *Leonardo* (1985), Vol.18, No.3, pp.170-178.

[4] The American Heritage Dictionary of the English Language (Fourth Edition). Houghton Mifflin, 2000.

[5] DUBUISSON M. P. and JAIN A. K.: A modified Hausdorff Distance for Object Matching. *In Proc. of Int'l Conf. on Pattern Recognition* (1994), Jerusalem, Israel, pp566-568.

A GPU-based Method for Massive Simulation of Distributed Behavioral Models with CUDA

Ugo Erra
University of Basilicata
ugo.erra@unibas.it

Bernardino Frola
University of Salerno
ber.frola@gmail.com

Vittorio Scarano
University of Salerno
vitsca@dia.unisa.it

Abstract

This work reports the results of a GPU-based approach for the massive simulation of a distributed behavioral model. In this model an agent has a local perception of the world and then it moves by coordinating with the motion of its neighbors. This carries a very high computational cost in the so-called nearest neighbors search. By leveraging the parallel processing power of the GPU and its new programming model called CUDA, we implemented a spatial hashing where a partitioning of the space is used to accelerate the neighbors search. Through extensive experiments, we demonstrate the effectiveness of our GPU implementation when simulating the motion of high-density agent groups.

Keywords: massive simulation, distributed behavioral models, graphics processing unit

1 Introduction and Motivation

Agent-based simulation is a common way to implement autonomous characters or agents to create crowds and other flock-like coordinated group motion. The number of agents involved in such collective motion can be huge, from several hundred birds to millions of fish schooling.

Reynolds (1987), presented the first behavioral model for computer animation applications. In this model every agent, called boid, takes decision using a local behavior model and it moves by coordinating with the motion of a limited number of its neighbors. Then, in order to obtain interactive results each agent must be able to identify efficiently neighbors among all existing agents in the world. This problem was already pointed out by Reynolds as a bottleneck, and suggested spatial hashing as a solution to avoid to the $O(n^2)$ of the brute force approach.

In this work, we present a GPU-based approach for massive simulation of distributed behavioral models by using CUDA (2008) framework for the graphics card NVIDIA G8x series. We adopt the GPU processing power for implementing a uniform data grid to support local perception in the nearest neighbors search. The simulation uses grid cells to keep track of the agents' position in the space and leverage the GPU offloads the sorting to build up the data grid structure to the GPU. The sorting is performed inside the cells to optimize the search and then to quickly obtain information about neighbors for each agent in parallel. Then, the GPU calculate for each agent the steering forces and update its position according to the Reynolds model.

2 The GPU-based Method

In order to avoid the $O(n^2)$ complexity of the neighbors search due to the communication of each agent with every other agent in the world, we adopt a common strategy based on the assumption that interaction of steering behavior drops off with distance. Then, we are interested only to compute efficiently a limited amount of neighbors agents. This assumption alleviates the computational effort required by the neighbors search as well as the difficult to manage dynamic data structures which are not trivial to implement on the GPU.

A video of this work is available at http://isis.dia.unisa.it/projects/behavert/.

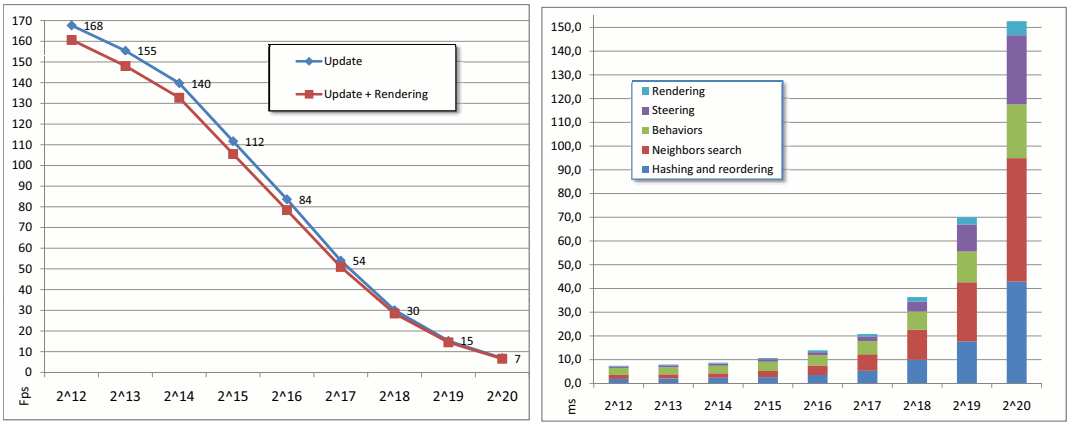


Figure 1: Performances with increasing number of agents. On the left, simulation time in fps with and without rendering (we used a simple impostor to render a boid). On the right, computational times in milliseconds broken down into principal phases. We considered the average values on 1000 frames.

Our implementation is inspired by the work on a particle system of Green in Nguyen (2007) which uses a static uniform grid data structure to compute the list of neighbors particle from a given particle. In fact, behavioral models have an important common point with particle systems; each agent is independent and, once computed the neighbors, they can be simulated in parallel.

Then, in order to accomplish this task, a static grid subdivides the world in cubic cells of the same size. Each cell in the world has an unique id. To aggregate in the GPU memory all the agents inside the same cell we assign a hash value to each agent based on its center position. In particular, given an agent position the hash value is the id of its cell. At the end of this step, the GPU performs a radix sort based on id cells. This reordering allows to identify quickly all agents inside the same cell as well as to increase the cache hit rate during neighbors search. In fact, to find all agents within a given region it is sufficient to consider agents inside the cells that overlap a region of interest.

3 Performance Evaluation and Conclusion

We performed a series of experiments in order to measures the performance of our approach. The tests were executed on an Opteron 252 2.6Ghz equipped with 2GB RAM and a GeForce 8800GTS 512MB. All the kernels were written in CUDA 2.1 and the application in C++. In all the tests, the agents are modeled by using the three basic steering behaviors of Reynolds. In the Figure 1, we report a simulation considering only 7 nearest neighbors while each boid steers and a cell size equals to 9. With these values it is possible to simulate about 100K boid agents at 60 fps including both simulation and graphics and up to 1M of boids with interactive performances.

The experiment results showed that this approach can be very effective in implementation of the basic Reynolds model. By exploiting the GPU processing power, we expect in the future that it will be possible to simulate more complex models or integrate features like the obstacle avoidance and the path planning.

References

CUDA (2008). *NVIDIA CUDA Programming Guide 2.0*.

Nguyen, H. (2007). *Gpu gems 3*. Addison-Wesley Professional.

Reynolds, C. W. (1987). Flocks, herds and schools: A distributed behavioral model. In *SIGGRAPH '87: Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, pages 25–34, New York, NY, USA. ACM.

Creating Your Own Facial Avatars

Yujian Gao
Computer Laboratory
University of Cambridge
yg259@cam.ac.uk

T. M. Sezgin
IUI Laboratory
Koc University
mtsezgin@ku.edu.tr

N. A. Dodgson
Computer Laboratory
University of Cambridge
nad10@cam.ac.uk

Abstract

We propose a method for generating muscle-based 3D facial avatars from static facial models. Our method eliminates the requirement of anatomical knowledge for building facial avatars, and it adapts easily to individual faces. A muscle model is automatically constructed based on the 3D landmarks labeled on the facial geometry, and then it can be animated using muscle activation data or by real-time performance. 3D facial avatars have wide utility in many applications in games, entertainment and communication in virtual environment.

Keywords: 3D Facial Avatar, Muscle-based Facial Animation, 3D Landmarks

1 Introduction

3D facial avatar is a key application of facial animation, it is widely used in entertainment, virtual communication, and low bandwidth teleconferencing for users to represent themselves. But for most of the current applications, the avatars are all created by skilled artists with expert anatomical knowledge, therefore it is difficult for users to customize them.

We address this problem and investigate ways in which this non-trivial task could be simplified. Our main contribution is a novel method for constructing deformable facial muscles interactively. We use regression to learn the mapping between 3D facial landmarks and muscle positions from examples. The muscle model for a new face can then be automatically created using this mapping. Our method makes it straightforward for ordinary users to generate muscle-based animatable faces without expert knowledge, and allows them customize their own avatars using static models obtained from internet.

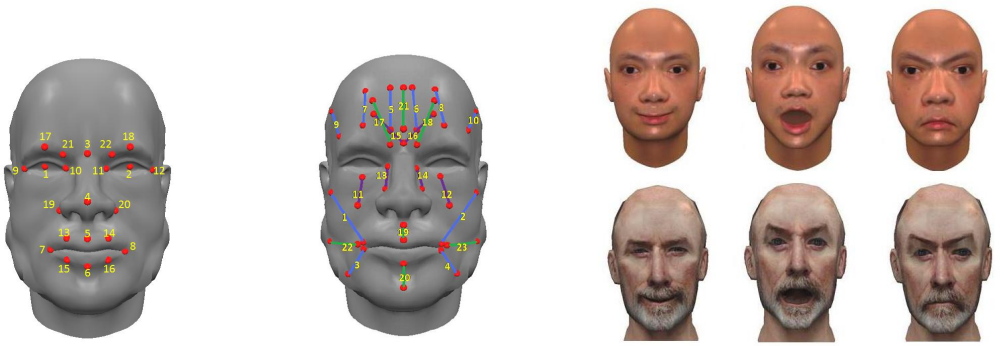
2 Related Work

Anatomically based muscle modeling is a non-trivial task. Several techniques have been proposed for modeling physically accurate muscles effeciently. For example, Kähler et al. (2001) introduced an editing interface to interactively specify mass-spring muscles to 3D facial geometry, but it still aimed for the usage of artists. A muscle mapping approach was presented by Zhang et al. (2003), but it needed an expert to mark the muscle end points on a *facial muscle image*. Yang and Zhang (2006) proposed an automated method for constructing basic muscle structure of human body, but it is not applicable to the human face. All in all, expert anatomical knowledge is still required for current facial muscle modeling techniques. Our method focuses on automating this procedure.

3 Method

The muscle model we used is based on Water's (Waters, 1987). It consists of 23 vector muscles and a sphincter muscle. To learn the mapping between landmarks and muscle positions, we first collected a set of 50 facial models, which cover a wide range of human facial variation. Then we manually labeled the 22 predefined 3D landmarks for each model (see figure 1(a)). The 23 vector muscles for each model were also created and tuned manually by artists (see figure 1(b)).

Each of the 23 vector muscles has two endpoints, i.e. 46 muscle points for each face. We construct a 46×50 muscle matrix, M, and a 22×50 landmark matrix, F. (The elements of M



(a) 22 Predefined Landmarks (b) 23 Vector Muscles (c) Facial Animation Results

Figure 1: Left: 22 predefined landmarks. Middle: 23 vector-based muscles. Right: Faical animation results using our technique.

and F are vectors representing 3D points, M_j^i is the jth muscle point position of the ith model, and F_j^i is the jth landmark position of the ith model.) Then the mapping is represented using a 46×22 transformation matrix, T, which can be obtained by solving the error minimization problem:

$$\min_T \left(\sum_{i=1}^{50} \sum_{j=1}^{46} \left\| M_j^i - (TF)_j^i \right\|^2 \right) \tag{1}$$

This equation represents the total spatial error of the 46 predicted muscle points for all 50 facial models. We use least-squares minimization to calculate T. Effectively each muscle point is computed as a linear combination of all the feature points.

For any new facial model, we first label the landmarks on the model manually or using automated method, then construct the 23 vector muscles M_{new} from the landmark matrix F_{new} by:

$$M_{new} = TF_{new} \tag{2}$$

Finally, we construct the sphincter muscle of the facial model. The sphincter muscle can be modelled as a parametric ellipsoid. The epicentre of the sphincter muscle P_{epi} is taken as the geometrical mean of the eight landmarks around the mouth. The semi-major axis, a, and the semi-minor axis, b, of the sphincter muscle ellipsoid are computed as:

$$a = \frac{\|P_{epi} - F_{new}(7)\| + \|P_{epi} - F_{new}(8)\|}{2} \quad b = \frac{\|P_{epi} - F_{new}(5)\| + \|P_{epi} - F_{new}(6)\|}{2} \tag{3}$$

Facial expression can then be animated using muscle activation data. We tested the universality of our method by applying it to models of different types. Figure 1(c) shows some results on two models.

References

Kähler, K., Haber, J., and Seidel, H. (2001). Geometry-based muscle modeling for facial animation. In *Proc. Graphics Interface '01*, pages 37–46, Toronto, Ont., Canada.

Waters, K. (1987). A muscle model for animation three-dimensional facial expression. In *Proc. SIGGRAPH '87*, pages 17–24, New York, NY, USA. ACM.

Yang, X. and Zhang, J. (2006). Automatic muscle generation for character skin deformation. *Computer Animation and Virtual Worlds*, 17(3-4):293–303.

Zhang, Y., Prakash, E. C., and Sung, E. (2003). Efficient modeling of an anatomy-based face and fast 3d facial expression synthesis. *Computer Graphics Forum*, 22:159–169.

Non monotonic approach of text-to-scene systems

Nicolas KAMENNOFF[1,2]
[1]ACSEL/L.E.R.I.A.
24, rue Pasteur
94270 Le Kremlin-Bicêtre

[2]LIPN – UMR CNRS 7030
99, avenue J.B. Clément,
93430 Villetaneuse, France
kamennoff.nicolas@lipn.univ-paris13.fr

Abstract

This document proposes an architecture for text-to-scene systems to deal with the problems of animation generation, lack of information from the text and errors due to literal interpretation. It suggests using a revisable approach based on a strong knowledge representation.

Keywords: Text-to-scene, Animation, non-monotonic logic.

1 TEXT-TO-SCENE SYSTEMS

A text-to-scene system aims commonly to automatically render a graphical scene from a textual description in natural language. This work is related to the ASMV project (French acronym for Semantical Analysis, Model-making and Visualization).

The main challenges a text-to-scene system has to deal with are:

- The fact that an action verb could correspond to different resulting motions for an object, depending on the context.
- The many pieces of information that are necessary for an animation, but are generally not given in non-specific texts, such as speeds, distances or number of lanes on the road and so on.
- The errors that are resulting from a literal interpretation of a text [1].

This research work studies a solution using knowledge representation and non-monotonic reasoning.

2 KNOWLEDGE REPRESENTATION

Object are representated by four main topics:

- Common information: to organize the structure of the system's knowledge database and to retrieve it. If applicable, identifier, name, type of the object and its component.

- Graphical properties: allows to render the object in the scene. If applicable, 3D model or information to create it, textures and material.

- Physical properties: required by the physical engine [2]. If applicable, weight, center of weight, velocity, adherence and dimensions.

- Basic capacities: used to generate animations for the object (see 3). If applicable, perceptual functions, basic motions and complex actions.

3 COMPLEX ACTIONS

Complex actions describe advanced capacities of an object; they are expressed in two parts:

- General description: a simple expression that summarize the action at a high level of generality. This description will be used by the validation system (see 5). It uses a set of operators that objects have to specify. For example, an overtake will be described by:
  - Overtake(A,B): T0 = A < B
    T1 = A > B
- Specific description: that describes how to create the action from the inherent capacities of the specified object. These descriptions use basic capacities, perceptive functions and previously defined complex actions. It is expressed using non-monotonic logic. This kind of logic allows to reason with uncertainty. We use here Reiter's default logic that allows to infer conclusion from incomplete elements and constraints.

4 INFORMATION FILLING

Due to lack of information in the initial text, system is required to fill in the blanks by its own knowledge. These new information is the output of the following process:

- Default knowledge: system needs to maintain default values in its own knowledge base. When the system computes the scenario (initial state of the scene and an ordered list of basic motions), it fills all information using default knowledge. These resulting values may be modified if a conflict is detected after the render validation (see 6).
- Constraint satisfaction: some information cannot be set by default. For example, the exact time needed to complete an action depends on the overall scene. The issue appears here to be a constraint satisfaction problem, and it is solved as the last step of the scenario generation.

5 RENDER VALIDATION

Literal interpretation of a text often leads to errors. To manage this, and other errors that can appear due to the usage of default knowledge, the system implement a run-time validation of the animation. Indeed, most 3D engine offers functionalities to check collision and relative positions of objects. Using the general description of action along with the rules of default behavior (for example, driving manuals for cars) allows the system to check the correctness of the result.

6 ADDITIONAL INFORMATION

Render validation allows the system to detect unwanted collisions, bad positions or failure in complex actions behavior. This detection can be used as an original means to find the correct interpretation of ambiguous part of the text. Moreover, the correction of these errors create additional information used to regenerates the scene, and links between errors and additional information have to be learned by the system from logs of a human expert to improve the overall performance of the text-to-scene system.

REFERENCES

[1] Kayser, D. (1997). *La sémantique lexicale est d'abord inférencielle*. Langue française, volume 113, pages 92-102, 1997.

[2] Bourg, D.M. (2002). *Physics for game developers*, O'Reilly, 2002.

[3] Reiter, R. (1980). *A logic for default reasoning*, Artificial intelligence journal, volume 13 (1-2), pages 81-132, 1980.

Study of Presence with Character Agents Used for E-Learning by Dimensions

Kweon, Sang-Hee
Sungkyunkwan University
Jongro, Seoul, Korea
skweon@skku.edu

Cho, Eun-Joung
Sungkyunkwan University
Jongro, Seoul, Korea
putyourhope@gmail.com

Kim, Eun-Mi
Sungkyunkwan University
Jongro, Seoul, Korea
emtf1984@hanmail.net

Cho, Ae-Jin
Sungkyunkwan University
Jongro, Seoul, Korea
holymars@nate.com

Abstract

This study examines factors of presence using the experimental method. We tried to analyze presence through the dimensions of character agents - text, voice, character dimension and gender - for e-learning platforms that use new technology-based content. There were 232 participants in the study. We measured their cognitive awareness of presence by agent dimensions and types of users. Correlation between types of users and presence showed significant results, but there were not significant results on dimensions. However, there were significant differences on character gender, voice or non-voice, text or non-text and character dimensions.

Keywords: presence, agent, character agent, e-learning, dimension, online lecture.

1 INTRODUCTION

The field of online education, however, has developed content specifically designed for the new technology. Online education applications enable people to feel like they are studying with human beings rather than with computers. Character agents have played a significant role in accelerating motivation and solutions for online education. Recently, scholars have studied the different characteristics of presence between standard definition television (SDTV) and HDTV. However, most of those studies have focused on the size or the definition of the display. In studies about character agents, most have measured simple effects, such as gender or facial expressions. This paper will study the effects of presence with character agents on e-learning. We will focus on various dimensions, such as subtitles, voice, character dimension and gender One of the most popular character agents is 'Clippy,' the friendly Microsoft character agent that tries to solve user problems. Microsoft spent a lot of money to develop the character, including employing psychologists as part of the development department. However, most users view Clippy as annoying, rather than helpful. Finally, the Clippy, was fired as quote from Microsoft. Development of character agents often requires the investment of large amounts of money; therefore, substantive studies of character agents are advisable. Studies have shown that character agents are effective at arousing motivation or adaptation [1], creating positive judgment and experience [2] and increasing achievement and efficacy [3]. However, the most significant results have come from studies focused on gender [4, 5, 6].

2 CONCLUSION

Table 1 shows that there are significant correlations between types of users and presence for the online lecture. However, the dimension of cognitive desire does not have significant correlation with some factors, perhaps due to sensory and cognitive being different dimensions. The results of the correlation analysis indicate that feeling presence when users join an online lecture is connected with their basic views about online lectures.

Table 1 Correlation between types of users and presence

	Cognitive	Sensory Fidelity	Interface Quality	Social Presence	Learning Knowledge	Enjoy/ Communicating	Getting Information
Sensory Fidelity	0.58**						
Interface Quality	0.52**	0.43**					
Social Presence	0.79**	0.53**	0.43**				
Learning Knowledge	0.44**	0.24**	0.19**	0.41**			
Enjoy/Communicating	0.44**	0.27**	0.21**	0.42**	0.74**		
Getting Information	0.48**	0.36**	0.38**	0.49**	0.76**	0.67**	
Cognitive Desire	0.18**	0.08	0.30**	0.15*	0.11	0.09	0.18*

Table 2 shows that firstly, there is no significant difference by character agent gender; character agent gender, whether male or female, has no effect on feeling presence. Secondly, there is no significant difference by voice or non-voice on the standard of .05. However, user gender shows significant difference on the standard of .10 (F=2.88, p=.09). Although it may not be a significant difference on the strict standard, it could be that users have different feelings about voice according to their gender. The factor that shows some difference is sensory fidelity; it increases the possibility.

Table 2 Presence by agent dimensions

	Factor	F	sig.		Factor	F	sig.
User * Agent gender	Cognitive	1.68	0.19	User gender * Voice or non-voice	Cognitive	2.71	0.10
	Sensory fidelity	1.85	0.16		Sensory fidelity	2.88	0.09
	Interface quality	0.04	0.96		Interface quality	0.02	0.89
	Social presence	1.09	0.34		Social presence	0.70	0.40
User gender * Text of non-text	Cognitive	2.71	0.10	User gender * Agent dimension	Cognitive	1.95	0.12
	Sensory fidelity	2.88	0.09		Sensory fidelity	0.87	0.46
	Interface quality	0.02	0.89		Interface quality	0.25	0.86
	Social presence	0.70	0.40		Social presence	2.40	0.07

This study has implications for use of character agents in software as well as hardware. Many DVDs and television lectures pay lots of money to develop and use character agents. However, this study indicates that these character agents do not have a large effect on users' experience of presence. The results mean that the content of the lecture is more important than the technological dimension of the online lecture experience.

REFERENCES

[1] Conati C. & Zhao X. (2004), Building and Evaluating an Intelligent Pedagogical Agent to Improve the Effectiveness of an Educational Game, *International Conference on Intelligent Use Interfaces*, pp.6-13.

[2] Park, Joo Yeon (2007), The Generation of Closeness with Interface Agent, *Journal of Korea Journalism and Broadcasting*, 51(2).

[3] Han, Keun Woo, Eun Kyoung Lee, & Young Jun Lee (2007), Computer Education Curriculum and Instruction: The Effects of a Peer Agent on Achievement and Self-Efficacy in Programming Education, *Journal of Korea Computer Education*, 10(5).

[4] Hone, K. (2006), Empathic agents to reduce user frustration: The effects of varying agent characteristics, *Interacting with Computers*, 18(2), pp.227-245.

[5] Nass, C. & Moon, Y. (2000), Machines and mindlessness: Social responses to computers, *Journal of Social Issues*, 56(1), pp.81-103.

[6] Picard, R. W. (1997), *Affective Computing*, MIT Press, Cambridge, MA.

An improved visibility culling algorithm based on octree and probability model*

Xiaohui Liang, Wei Ren, Zhuo Yu, Chengxiao Fang
State Key Lab. of Virtual Reality Technology and Systems
School of Computer Science,Beihang University, Beijing, 100191, China
lxh@vrlab.buaa.edu.cn

Yongjin Liu
Department of Computer Science and Technology
National Lab. for Information Science and Technology, Tsinghua University
liuyongjin@tsinghua.edu.cn

Abstract

This paper studies the visibility culling problem and proposes an improved coherent hierarchical culling (CHC) algorithm. Octree structure is adopted to organize the complex scenes and a novel probability model is proposed to handle the redundancy and unnecessary occlusion culling in the classical CHC. In the probability model, to improve the query strategy of CHC algorithm, the expected values of occlusion query time is calculated and compared to the rendering time. Experimental results with several typical complex scenes are presented. The comparison results with three classical culling algorithms show that our algorithm has superior performance and can render high-depth complex scenes containing a large number of objects in real time.

Keywords: Complexly Dynamic Scene; Real Time Rendering; Visibility Culling; Coherent Hierarchical Culling; Probability Model

1 Introduction

Many visibility culling methods have been developed in the last decade.Both good scene structures and an efficient culling algorithm are of great significance in visibility culling. Cohen-Or (2003) presented a comprehensive survey of visibility culling methods. Bittner (2004) presented the classical CHC algorithm which makes use of both the spatial and the temporal coherence of visibility. CHC algorithm reduces the node set needed for occlusion query to the terminating node set. Terminating nodes are divided into visible leaf nodes and invisible internal nodes. CHC algorithm only arranges occlusion query for visible leaf nodes and invisible internal nodes in the terminating set.The redundant and unnecessary occlusion queries introduced in the classical CHC algorithm have the following disadvantages:

1) All visible nodes in terminating set are continuing to execute redundant occlusion queries. So a lot of occlusion queries have been wasted. This will increase the rendering cost.

2) The CHC algorithm does not analyze specific cases of each node. If the rendering cost is less than occlusion query cost, the node does not need to execute the occlusion queries no matter whether the node is visible or not.

To offer a solution to the above problems, this paper presents a probability model and improves the query strategy of occlusion culling of CHC algorithm. As a result, rendering efficiency are improved and the number of occlusion queries is reduced.

*National Natural Science Foundation of China (60873159), Program for New Century Excellent Talents in University (NCET), and National High Technology Project(2006AA01Z333)

2 An improved visibility Culling Algorithm

The algorithm presented in this paper includes data reorganization, probability optimization computing. First of all, the original data of complex scenes is organized using a octree because this structure can easily be subdivided in memory and easily be loaded for out-of-core model. Based on this structure, in order to reduce the complexity in rendering scenes, we cull invisible objects through probability optimization computing and then send visible objects to the graphics pipeline.

In our algorithm. the octree is constructed through top-down approach by adjusting the size of the bounding volume of the root node. In order to handle out-of-core model, the leaf node of the octree stores the index of geometry model. To make the structure can be used to handle moving object in the scene. We consider the object information in leaf node and adopt loose subdivision mechanism. That is, for the complex scene that contains some small moving objects, the octree is not constructed using strict splitting approach. When a node contains complete(or most of) object information, the node will be treated as a leaf node and not subdivided. In the runtime, when the object moves, only the index in the node will be changed.

It is meaningless to arrange occlusion queries for those objects whose rendering time is less than occlusion query time. How to estimate the occlusion query time and the rendering time is an additional question that we must resolve. We can use the state of nodes to represent the state of objects in the node. The objects in the scene have two states: visible and invisible. Let O_i denote the object, V denote the state is visible and $\bar{V}$ denote the state is invisible. Let $P_i(V)$ denote the visibility's probability of O_i, $P_i(\bar{V})$ denote the invisibility's probability of O_i, C_r denote the rendering time of O_i and C_q denote the occlusion query time of O_i. The object's rendering method may proceed through directly rendering (A1) or rendering after queries, depending on whether the result of occlusion query is visible after finishing occlusion query (A2). For A1, the time cost is C_r; for A2, the time cost is $C_q + C_r$ when the object is visible and the time cost is C_q when the object is invisible. The expectations of time cost of an object in different states are calculated using the following formulae:

$E(A1) = P_i(V) \times C_r + P_i(\bar{V}) \times C_r = C_r$

$E(A2) = P_i(V) \times (C_r + C_q) + P_i(\bar{V}) \times C_q = P_i(V) \times C_r + C_q$

If the time cost is computed before occlusion query, we can only select nodes with $E(A1) > E(A2)$to do occlusion query so as to further reduce the number of occlusion queries and make them more reasonable. However, the calculation of time cost cannot be too complicated, otherwise most rendering time will be spent on it. Therefore, This paper further give a way to determine C_r, $C_q$and $P_i(V)$.

3 Conclusion

In this paper, an improved coherent hierarchical culling algorithm based on probability model is proposed. The scene organization can reduce the cost of dividing the border objects so as to overcome many shortcomings of the traditional octree. This paper also presents a probability model and improves the efficiency of occlusion query by comparing the mathematical expectations of direct rendering time and the rendering time after occlusion query of objects. We use teapot and power plant scene to test our algorithm and get good results.

References

Cohen-Or D., Chrysanthou Y., Silva C., Durand F. (2003). A survey of visibility for walkthrough applications. In *IEEE Transactions on Visualization and Computer Graphics Vol.9 No.3*, pages 412–431. IEEE Computer Society.

Bittner J., Wimmer M., Piringer H., Purgathofer W.(2004). Coherent hierarchical culling:hardware occlusion queries made useful. In *Computer Graphics Forum, 2004, Vol.23, No.3*, pages 615–624,The Publisher.

Using Motion Capture Data to Optimize Procedural Animation

Chang-Hung Liang
Computer Science Department
National Chengchi University
g9606@cs.nccu.edu.tw

Tsai-Yen Li
Computer Science Department
National Chengchi University
li@nccu.edu.tw

1 INTRODUCTION

Procedural animation [1][3] has the advantage of being able to generate flexible animations according to high-level parameters while motion capture-based (MoCap-based) methods [2] have the advantage of creating realistic animations. In this paper, we attempt to use the data from motion capture to automatically find an optimal set of parameters that can be used to produce realistic animations for procedural animation. These sets of procedural parameters can then be interpolated to produce desired animations by taking into account the constraints in a specific scenario. The experimental results show that the proposed method can not only produce plausible animations but also accommodate environmental constraints by adjusting high-level procedural parameters.

2 SYSTEM OVERVIEW

We model the problem of generating natural-looking motions with animation procedures as an optimization problem by comparing the procedure-generated motion with the motion-captured motion. We use a walking procedure as an example conduct our experiments. The walking motion is modeled with four keyframes. A total of 194 motion attributes are used to define the keyframes and inbetweens. Twenty six of them are defined for keyframes such as step length, step height, and arm swaying angle. The remaining parameters (168) are used to control the timing (ease in/out) of the motions between every two keyframes.

An overview of the system is shown in Figure 1. The auto-tuner module is the main component of the system used to search for the optimal set of motion parameters in the animation procedure for the creation of realistic motions. The objective function for the optimization process is defined as

$$f(A) = \sum_{i=1}^{n} d(P_A[i], M[i]), \tag{1}$$

where A is a given set of procedural parameters, $P_A[i]$ and $M[i]$ are the ith frame in the animations generated by the procedure and the MoCap data, respectively, and d is a distance function computing the weighted sum of the discrepancies on the corresponding joint locations between two motions.

We have used the Simulated Annealing (SA) algorithm to solve the optimization problem. The search space is defined by the 194 procedural parameters, each of which is bounded in a reasonable interval to confine the search space. The goal of the search is to find the optimal solution minimizing $f(A)$ of eq. (1). An example (left hip angle) of comparison between the motions generated by the procedure and by the MoCap data before and after the optimization is shown in Figure 2.

The objective of finding the optimal set of procedure parameters is to make the generated motion look as natural as the MoCap motions. Nevertheless, the main advantage of generating animations with procedures is on the flexibility of creating a good variety of motions according to the requirements of a given scenario. For example, the system should be able to generate the walking motion according to the step length and step height required in a specific scene. However, the parameters in the animation procedure are not likely to be independent. Varying only some parameters often results in an unnatural motion.

In order to account for the possible unnaturalness resulting from varying parameters, we propose to apply the method to many sample motions of the same type from the MoCap database and then interpolate the obtained parameter sets according to the given independent parameters such as step length and step height. Assume that the parameter set for walking is defined as $P_i(a_1, a_2, a_3, \ldots, a_{194})$, where i (=1..m) is the index of a sample motion in the

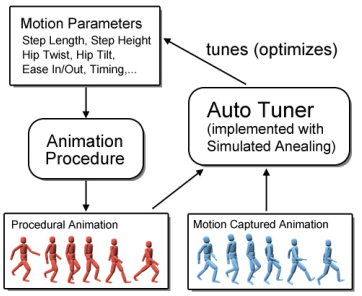
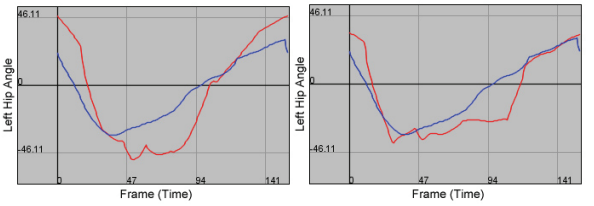


Figure 1. System overview



(a) (b)

Figure 2. Comparison of a joint angle (red: procedural, blue: MoCap) (a) before optimization and (b) after optimization

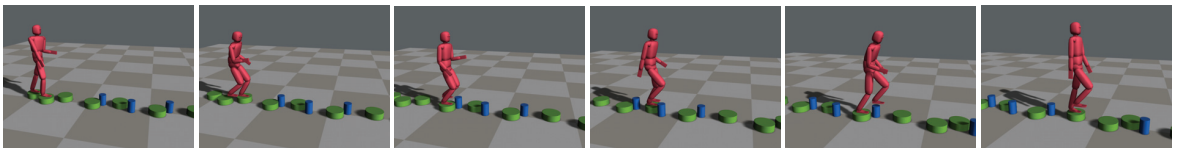


Figure 3. An example animation showing how adaptive motions can be generated according to the requirement of the environmental constraints

MoCap database for walking. Suppose that the parameters a_1 and a_2 represent step length and step height, respectively. We hope to compute the dependent parameters (a_3, … , a_{194}) by interpolating on the given independent parameters (a_1, a_2). In order to find the closest samples of a given query point for interpolation, we first perform Delaunay Triangulation on the sample points. Then we find the triangle where the query point is located and use the three vertices of the triangle to linearly interpolate the dependent parameters.

In Figure 3, we show an example of walking motion generated by the procedure with the need of using different step lengths and step heights along the path to satisfy the environmental constraints. When the desired step lengths and step heights are determined in a given scenario, the system computes the remaining parameters according to this specification by interpolating nearby sample points.

3 CONCLUSIONS

Different approaches to the generation of character animations have their own pros and cons. In this work, we attempt to combine the advantages of the MoCap-based approach and the procedure-based approach. We use MoCap data to enhance the realism of the motions generated by procedural animation in an automatic fashion. The enhancement is modeled as an optimization problem on the animation parameters. The obtained parameters for different motion samples in a MoCap database can be further interpolated to obtain the desired procedural parameters for a target specification. The tedious and time-consuming parameter tuning in the design of an animation procedure can therefore be avoided while the flexibility of procedural animation is kept.

REFERENCES

[1] Bruderlin, A. and Calvert, T.W. (1996). Knowledge-Driven Interactive Animation of Human Running. *Proceedings of the Conference on Graphics Interface 1996*, pages 213-221, 1996.

[2] Kovar, L., Glercher, M., and Pighin. F. (2002). Motion Graphs. *Proceedings of ACM SIGGRAPH 2002*, pages 473-482, 2002.

[3] Perlin, K. (2003). Building Virtual Actors Who Can Really Act. *Proceedings of the 2nd International Conference on Virtual Storytelling*, pages 127-134, 2003.

Stepping Off the Stage

Brian Mac Namee and John D Kelleher
DIT AI Group, Dublin Institute of Technology,
Kevin St, Dublin 8, Ireland
{Brian.MacNamee|John.Kelleher}@comp.dit.ie

Abstract

Mixed-reality virtual agents are an attractive solution to the problems associated with human-robot interaction, allowing all the expressiveness of virtual characters to be married with the advantages of a physical artifact which exists in a shared environment with the user. However, common approaches to achieving this restrict the virtual characters appearing on top of, or encompassing the robot. This paper describes the *Stepping Off the Stage* system in which mixed-reality agents are allowed to step off the robot stage and move to other parts of the environment, offering compelling new interaction possibilities.

Keywords: Mixed reality, robotics, human-robot interaction, intelligent virtual characters

1 Introduction

Because human interactions with robots must be fundamentally different from our interactions with more mundane machines there has been considerable research effort put into making interactions with robots more engaging (for a good overview see (Fong et al., 2003)). However, it is difficult to build hardware devices capable of the subtleties of expression required for engaging interactions. To overcome these difficulties, animated virtual agents displayed on screens mounted on robots are used as interfaces (Bruce et al., 2001). However, this approach denies the user the opportunity to share an interaction space with the virtual agent and has been found to be limited in terms of the level of engagement that is achieved.

More recently *mixed-reality* approaches have been used. In this approach, users view robots, and their environment, through viewing devices that allow the robots to be augmented with virtual agents which appear to sit on top of or encompass them (Dragone et al., 2007). This marries the expressive capabilities of a virtual agent with the advantage that the user and the agent share a common physical interaction space.

The question posed in this paper is: *why restrict the virtual agent to acting as an interface to the robot?* In many application scenarios - such as guiding, tutoring or selling - it is more appropriate to treat the mobile robot as a vehicle that a virtual agent can use to move around an environment and then, when appropriate, step off the robot stage and interact with other artifacts. The remainder of this paper will describe the *Stepping Off the Stage* (SOTS) system which has been developed to achieve this.

2 The Stepping Off The Stage System

The SOTS system is based on the metaphor of virtual agents moving between *stages*. When an agent moves from one interaction area to another it is said to step from one stage to another. On moving to a new stage the agent is informed of the physical limitations of the area in which it can now operate, and the kinds of behaviour that are now suitable (as this can differ greatly from one stage to another).

The high level architecture of the SOTS system is shown in Figure 1. The *Stage Manager* is at the centre of the architecture, and is responsible for managing all objects within the execution environment. Its key tasks are: determining the availability of stages to agents through the

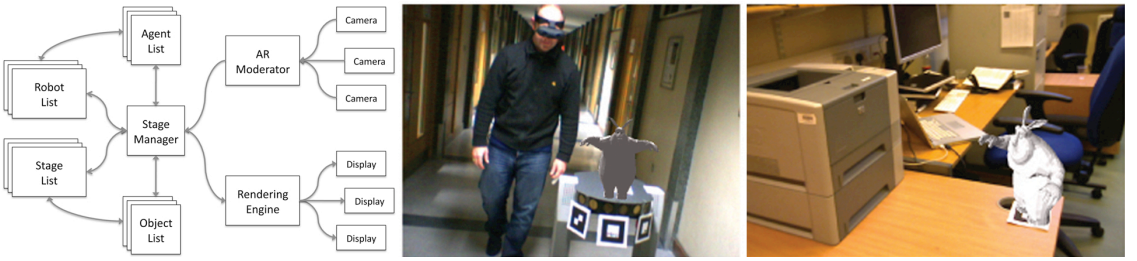


Figure 1: The SOTS architecture; a SOTS agent being followed by a user; and describing a printer.

recognition of fiducial markers, which is performed by the *AR Moderator* using the the open source API ARToolkit; summoning robots as requested by agents; managing agents whose behaviours are driven using a combination of finite state machines and the *role-passing* technique of Mac Namee et al. (2003), which is based on *situational intelligence* and allows agents behave believably in a wide range of different situations; and informing the *Renderer* which virtual augmentations should be drawn. The remaining components of the architecture store lists of the key protagonists in a SOTS execution environment, namely: *objects* (which can be both real and virtual), *stages*, *character agents*, and *robots*.

To demonstrate the SOTS system a working prototype has been implemented in which a new employee arriving on the first day at their new place of work is greeted at the foyer by Pixel the virtual rabbit who guides the employee to their new office and instructs them in how to use their new office equipment. Screenshots of this prototype are shown in Figure 1. In these images Pixel is shown being followed by the user, at which time Pixel inhabits the robot-top stage (this image is a composite generated to illustrate both the user and what they see); and telling the user all about their new printer, at which time Pixel inhabits a desktop stage. This basic prototype demonstrates all of the components of the SOTS system working together.

3 Conclusions & Future Work

The work described in this paper begins from the position that mixed-reality characters offer a very attractive solution to solving the problem of human-robot interaction. However, there is no need to stop here. By adopting the metaphor of stages, and using the technique of role-passing, SOTS agents can step off their robotic stages and into the user's environment to offer new engagement interaction possibilities. In the near future we intend to improve this work through the addition of improved rendering, more interactive characters, and a more elaborate application scenario. All of this work is moving towards a larger goal of a creating a mixed reality character that appears to share environments with human users.

References

Bruce, A., Nourbakhsh, I., and Simmons, R. (2001). The role of expressiveness and attention in human–robot interaction. In *Proceedings of the AAAI Fall Symposium Emotional and Intelligent II: The Tangled Knot of Society of Cognition*.

Dragone, M., Holz, T., and O'Hare, G. (2007). Using mixed reality agents as social interfaces for robots. In *Proceedings of the 16th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN 2007)*, Jeju Island, Korea.

Fong, T., Nourbakhsh, I., and Dautenhahn, K. (2003). A survey of socially interactive robots. *Robotics and Autonomous Systems*, 42:143–166.

Mac Namee, B., Dobbyn, S., Cunningham, P., and O'Sullivan, C. (2003). Simulating virtual humans across diverse situations. In *Proceedings of Intelligent Virtual Agents '03*, pages 159–163.

Creative Approaches to Emotional Expression Animation

Mr Robin J.S. Sloan
University of Abertay
Bell Street, Dundee, UK
r.sloan@abertay.ac.uk

Mr Brian Robinson
University of Abertay
Bell Street, Dundee, UK
b.robinson@abertay.ac.uk

Dr Malcolm Cook
University of Abertay
Bell Street, Dundee, UK
m.cook@abertay.ac.uk

Abstract

In facial expression research, it is well established that certain emotional expressions are universally recognized. Studies into observer perception of expressions have built upon this research by highlighting the importance of particular facial regions, actions, and movements to the recognition of emotions. In many studies, the stimuli for such studies have been generated through posing by non-experts or performances by trained actors. However, character animators are required to craft recognizable, believable emotional facial expressions as a part of their profession. In this poster, the authors discuss some of the creative processes employed in their research into emotional expressions, and how practice-led research into expression animation might offer a new perspective on the generation of believable emotional expressions.

Keywords: Emotion and Personality, Facial Animation

1 EMOTIONAL EXPRESSION ANIMATION

Emotional facial expressions have been examined in great detail. Early research into human perception of expressions has led to the identification of universal expressions, which in turn has informed artistic training and practice. However, while the drive to create more authentic and accurate representations of facial expressions has a strong foothold in computer science and psychology research, the practice of character animation can afford an added insight into how expressions are generated and perceived. Expressions can be carefully modified and varied to determine their effectiveness in conveying a desired emotional state of a character.

Early Disney animation made use of skilled artists who developed a range of procedures, techniques, and principles which can be applied to produce effective character animations [3]. For animators, an effective animation is not only one that the audience can read easily, but also one that they will find believable. It seems straight forward enough to create an expression animation that an audience will recognize as ‘happiness’, yet subtle cues within that animation can shatter audience suspension of disbelief.

Animation principles which imply the application of caricature (such as exaggeration or anticipation) can make characters seem more lifelike. It can be argued that the overriding principle in character animation is that effective character animation is the creative *imitation* - rather than the strict *replication* - of life.

Emotional Avatars is an interdisciplinary research project which draws upon the knowledge of animation, computer arts, and psychology research. The aim of the project is to investigate the generation and perception of emotional expression animation to determine whether the nuances of emotional facial expression can be choreographed more effectively [2]. This poster examines the practice-led research methods utilized by the project members.

2 GENERATING AND EVALUATING ANIMATION

Animations were produced that covered the six universal expressions at three levels of emotional intensity. The eighteen animations were created and refined by the lead author. Reference materials were drawn from prominent studies of facial expressions and from video footage of acted and naturalistic expressions [1]. For the most part, however, the animation was produced iteratively through the application of artistic judgment calls. The animator

initially blocked out the basic animations before making adjustments to sequence, timing and duration. This resulted in a visual log of animation production, both in terms of iterations of animation and sketchbook development.

Throughout the course of the project, the animator focused on enhancing the believability of the expressions by employing core animation principles. For example, the expression of surprise was manipulated through the application of *anticipation* – that is, a preliminary action which sets up a major action. Before the eyes opened fully at the peak of the surprise expression (the major action), a blink was used so that the audience would anticipate this event (preliminary action). Instead of replicating life, a creative imitation of life was used to enhance the recognition and believability of the expression.

Through evaluation of the animation process, it was possible to determine which spatial-temporal attributes the animator felt were most appropriate for the emotion being animated. Adjustments to features and timing are clear from both the development of animation and the sketchbooks. However, this process is highly subjective when only one animator is involved. To gauge the opinion of other animators and a wider audience, the animations were shown to colleagues in the latter stages of development and displayed at an exhibition upon completion.

3 REVIEW AND EXPOSITION OF ANIMATION

To assess the quality of the animation, videos of the expressions were reviewed by the project team. A review procedure was developed which required the team to analyze the animations in subtle detail. The reviewers interrogated the quality of the animation using five primary measures (identification of emotion, emotional clarity, emotional intensity, sincerity, and authenticity) applied to several factors (facial regions, expression onset, expression apex, expression offset, expression duration, and feature timing). The opinions and comments of the reviewers were added to the review document, which was later used to generate the final iterations of the expression animations.

The finished animations were initially incorporated into a controlled experiment which looked to measure observer perception of expressions [2]. While the output from this experiment provided crucial feedback for the animators, the feedback was limited and did not provide in-depth critique from observers. In order to gather detailed responses from observers, an exposition of the animation was designed.

Unlike a standard exhibition of work, the animation exposition was designed to gather qualitative data from visitors through interaction and visitor contribution. Previously, subjective criticism by the animator reflecting in practice and peer criticism from expert colleagues informed the production of animation. By exposing the work to a wider audience and encouraging contributions, the exposition effectively represented a third level of observer criticism. Visitors to the exposition were able to contribute in a number of ways, including; voting for and ranking animations using a ballot box, interacting with an animated avatar, leaving detailed comments in visitor books, creating sketches of facial expression on the wall, and conversing with the animator.

4 CONCLUSION

Scientific research into expression movement and perception will enhance the authenticity of animated characters. However, some of the most believable animated characters are those which are created and refined iteratively by skilled and knowledgeable artists. Further research into how artists produce such performances could improve our understanding of what makes an animated expression recognizable and, perhaps more importantly, believable.

REFERENCES

[1] Sloan, R.J.S., Robinson, B., Cook, M., and Bown, J. (2008). Dynamic emotional expression choreography: perception of naturalistic facial expressions. *SAND Conference Proceedings*, Swansea, UK 24-28 November 2008. Swansea Metropolitan University: Swansea

[2] Sloan, R.J.S., Cook, M., Robinson, B. (2009). 'Considerations for believable emotional facial expression animation'. *2nd International Conference on Visualization,* Barcelona, Spain. (Submitted for publication February 2009).

[3] Thomas, F., Johnston, O. (1981). *The illusion of life: Disney animation*. New York: Disney Editions.

GATE* Session Papers

CASA 2009

*This research has been supported by the GATE project, funded by the Dutch Organization for Scientific Research (NWO) and the Dutch ICT Research and Innovation Authority (ICT Regie)

Abstracting from Character Motion*

B.J.H. van Basten
Center for Advanced Gaming and Simulation
Utrecht University
basten@cs.uu.nl

Abstract

Natural motion of virtual characters is crucial in games and simulations. The naturalness of such motion strongly depends on the path the character walks and the animation of the character locomotion. Therefore, much work has been done on path planning and character animation. However, the combination of both fields has received less attention.

Combining path planning and motion synthesis introduces several problems. A path generated by a path planner does not contain details of the character motion, nor does it contain the smaller details of a movement. This raises the question which (body) part of the character should follow the path generated by the path planner and to what extent it should closely follow the path, because the path generated by the planner often does not contain smaller details of a natural movement. In this paper we will compare several methods to abstract from the pelvis (root) trajectory of the character. The results of this experiment show that the resulting animations generated using the presented path abstraction methods are significantly better than those generated by the standard method.

Keywords: animation, path planning, motion planning, filtering, motion synthesis

1 Introduction

Character locomotion plays a very important role in many games and simulations. The quality of the character locomotion is highly dependent on the path the character chooses and the animation generated for walking along this path. The generation of character locomotion is often divided into two phases, such as in Pettré et al. (2003). First, a path planner generates a path from A to B. Second, an animation system generates an animation that walks along this path.

1.1 Problem

Unfortunately, the path generated by a path planning algorithm is just a parametric curve with no information on how the character following that curve should be animated. Second, such a path is often a simplification of the motion that the character needs to follow. In general, it is not clear which (body) part of the character should follow this path. In many systems, the path is interpreted as the desired trajectory of the projection of the pelvis on the plane. However, when one takes a closer look at the trajectory of the pelvis during locomotion it appears that the pelvis oscillates during such a motion, as can be seen in Figure 1. When enforcing the pelvis to closely follow the desired path, one looses this natural oscillation and will end up with a unnatural motion, without this natural *wiggle*.

In this paper, we will show that synthesizing motion based on a *motion abstraction* instead of the traditional pelvis (or root) trajectory will lead to more natural motions.

*This research has been supported by the GATE project, funded by the Netherlands Organization for Scientific Research (NWO) and the Netherlands ICT Research and Innovation Authority (ICT Regie).

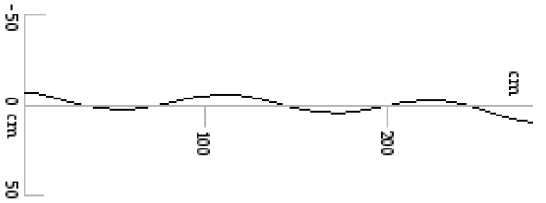


Figure 1: The trajectory of the pelvis during locomotion along a straight line.

1.2 Related Work

Some research has been done on combining path planning and character animation. Choi et al. (2003) create a map of footprints based on a probabilistic path planning technique, then search this roadmap to find a collision-free path after which they use displacement mapping to adjust the motions so they fit the target footprints. Sung et al. (2005) also plan a path using probabilistic path planning techniques and generate a motion (using a motion graph variant) that approximately follows the path. This motion is then adjusted to follow the path more accurately. Lau and Kuffner (2005) create a high-level motion graph (considered as a finite-state machine) and searches over this graph for navigation. This method implicitely uses the animation data to define the navigation space.

2 Motion Synthesis

In order to reuse motion captured data, one can concatenate existing clips by transitioning from one motion to the other. However, the transition frames need to resemble each other, otherwise visual artifacts may occur. Manually selecting transitions between two motion clips is time consuming. Therefore techniques have been developed to automate this process.

One such technique is the *motion graph*. Such a graph encapsulates all motions and all possible transitions between these motions. Each edge is a clip of motion, each node is a frame. One can automatically create a motion graph, as is shown by Kovar et al. (2002). Many variants of motion graphs exist, mostly optimizing the complexity of the graph or the search algorithm.

A path in a motion graph represents an animation and by doing a graph search, animations can be generated that adhere to given constraints. An example of such a constraint is the input path that the generated animation should follow. A search algorithm searches through the graph to find pieces of motion that, when concatenated, will result in a motion that follows the path. It does so by comparing the *pelvis trajectory* to the desired path, which leads to the problems described above.

We can enlarge a graph by increasing the threshold for the allowed distance between transition frames. Doing so will generate a bigger graph, which is able to generate more animations, but we also might introduce more visual artifacts. Many distance metrics for keyframes exist. van Basten and Egges (2009) showed that using joint-angle-based metrics results in the least path deviation. However, as said, enforcing the pelvis trajectory to closely follow the path will result in an unnatural wiggle. Therefore we will use this metric to evaluate the motion abstraction techniques which we will describe below.

3 Motion Abstraction

We have evaluated 5 techniques that provide a motion (or path) abstraction:

Root Trajectory: The standard method, where the pelvis (root) trajectory is considered to be the global motion.

Joint Combination: Uniform interpolation of several joints (foot, ankle, knee).

Joint Orientation Extrema: The global path follows the midpoints of two consecutive local extrema of the joint paths.

Gaussian Smoothing: Gaussian smoothing to filter out pelvis oscillations.

B-Spline Filtering: Approximate the path abstraction by a B-spline, as is done by Gleicher (2001).

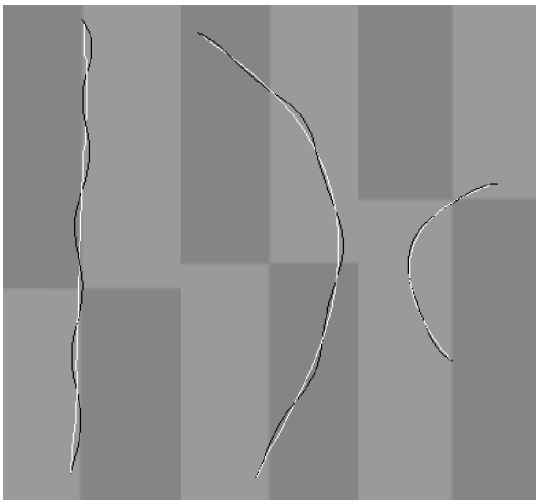


Figure 2: Motion abstraction using a Gaussian filter

We evaluate these techniques by having the motion graph algorithm select the motion clips based on comparing the desired path with the motion abstraction, instead of the standard pelvis trajectory. This results into different animations. Then, we evaluate the wiggle of the generated animations by comparing 3 quality measures with real recorded motion (of a subject walking the same path).

The most obvious visual artifact is foot skating, we determine foot skating using the method by van Basten and Egges (2009). Second, the pelvis oscillation can be considered a *wave*. Therefore we evaluate wave-specific properties such as average amplitude and average wavelength of the pelvis oscillation.

4 Results

For all 3 test paths, using a motion abstraction instead of the standard pelvis trajectory results into significantly better animation: the wiggle properties are closer to recorded motion than when using the pelvis trajectory. Figure 3 shows the quality measures for the real recorded motion, the standard method and the results of using other motion abstractors for a straight path. It is clear that using path abstraction leads to animations that resemble recorded motion. This also holds for curvy paths.

5 Conclusion

Overall, the path abstraction methods generate good approximations of the global motion of the character. We can conclude that it is better to use one of the path abstraction methods to approximate the global path than to use the pelvis trajectory. Motion synthesis using motion abstraction instead of the pelvis trajectory will result in more natural animations, due to a more natural oscillation of the pelvis.

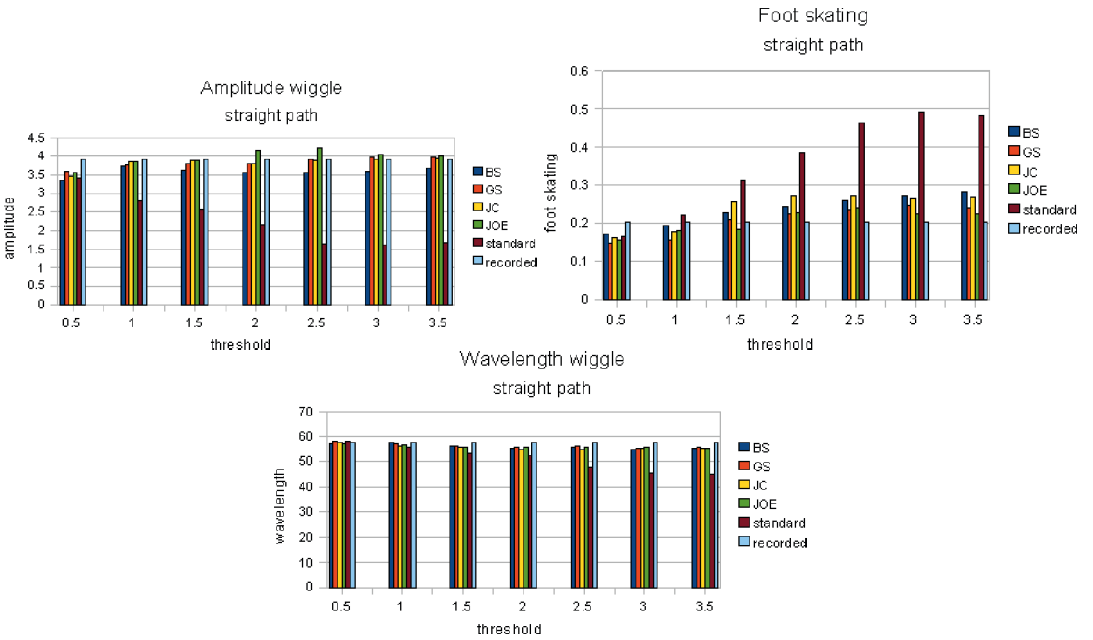


Figure 3: The resulting animations when using a path abstraction are resembling original motion more than the standard method

References

Choi, M. G., Lee, J., and Shin, S. Y. (2003). Planning biped locomotion using motion capture data and probabilistic roadmaps. *ACM Trans. Graph.*, 22(2):182–203.

Gleicher, M. (2001). Motion path editing. In *I3D '01: Proceedings of the 2001 symposium on Interactive 3D graphics*, pages 195–202, New York, NY, USA. ACM.

Kovar, L., Gleicher, M., and Pighin, F. (2002). Motion graphs. In *SIGGRAPH*, pages 473–482, New York, NY, USA. ACM Press.

Lau, M. and Kuffner, J. J. (2005). Behavior planning for character animation. In *SCA '05: Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 271–280, New York, NY, USA. ACM.

Pettré, J., Laumond, J.-P., and Siméon, T. (2003). A 2-stages locomotion planner for digital actors. In *SCA '03: Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 258–264, Aire-la-Ville, Switzerland, Switzerland. Eurographics Association.

Sung, M., Kovar, L., and Gleicher, M. (2005). Fast and accurate goal-directed motion synthesis for crowds. In *SCA '05: Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 291–300, New York, NY, USA. ACM.

van Basten, B. J. H. and Egges, A. (2009). Evaluating distance metrics for animation blending. In *FDG '09: Proceedings of the 4th International Conference on Foundations of Digital Games*, pages 199–206, New York, NY, USA. ACM.

The GATE Project: GAme research for Training and Entertainment*

Prof. Dr. Mark Overmars
University of Utrecht,
Scientific Director of the GATE Project
m.h.overmars@cs.uu.nl

Abstract

It is clear that the possibilities of gaming will rapidly increase over the coming years. Equipment is getting more powerful all the time. New graphics and physics cards allow for increased visual realism but this must be accompanied by increased behavioral realism of game characters. New interface technology will enable a different, more natural form of communication and control. Gesture recognition, tactile feedback, and possibly even direct brain connections will become possible. Games will also not only happen on a screen but can influence other actuators in the house. And high-speed broadband connections and wireless access leads to new forms of collaboration and to new types of games, like large online game communities and mobile gaming, each with its own research challenges. These developments will have a huge impact on both entertainment games and on training and educational use of gaming and simulation. It is already a reality that people take part of their driving lessons in simulators. Games are used in training safety procedures and crisis management. Similar developments will happen in decision and policy making. In education gaming will offer ample possibilities for personalized learning, long distance learning, and lifelong learning. To advance the state-of-the-art in gaming, to facilitate knowledge transfer to companies, and to show the potential of gaming in public sectors, the government has funded the GATE project with a total budget of 19 million Euros. The project runs from 2007 till 2012 and involved eight partners: Utrecht University, Utrecht School of the Arts, TNO, Twente University, Delft University of Technology, Waag Society, NederlandBreedbandLand and Thales.
The ambition of the GATE project is to develop an international competitive knowledge base with respect to game technology, and to train the talent required to enhance the productivity and competitive edge of small and medium-sized creative industrial companies. The project will substantially improve the competitiveness of companies producing (tools for) games and simulations by providing direct access to new technology and by technology transfer projects. This will lead to larger companies, encourage the founding of new companies, and attract companies from other countries to the Netherlands. The project will make people aware of the possibilities of gaming in public sectors such as education, health, and safety by performing pilots in these areas. As a result gaming and simulation will become more commonly applied in these sectors, leading to quality improvements and cost reductions.

*The GATE project is funded by the Netherlands Organization for Scientic Research (NWO) and the Netherlands ICT Research and Innovation Authority (ICT Regie)

Modeling Natural Communication*

Bart van Straalen
University of Twente,
Human Media Interaction,
Enschede, The Netherlands
straalenb@ewi.utwente.nl

Abstract

With the increasing amount of virtual, intelligent agents in serious games and in training and educational simulations, there is also an increasing demand for more natural and realistic behaviors. As such, one of the objectives in the GATE project is to study how the cognitive behavior of such agents can be modeled. At the university of Twente we are trying to create an accurate model of the cognitive processes that are involved in natural communication, focusing on how, why and which cognitive processes direct the production and realization of appropriate verbal and non-verbal behavior by virtual characters in a conversation with the user.

*This research has been supported by the GATE project, funded by the Netherlands Organization for Scientic Research (NWO) and the Netherlands ICT Research and Innovation Authority (ICT Regie)

User evaluation of the Movement of Virtual Humans

Herwin van Welbergen
Human Media Interaction
University of Twente
Enschede, the Netherlands
H.vanWelbergen@ewi.utwente.nl

Sander E.M. Jansen
TNO Human Factors, the Netherlands
Utrecht University, the Netherlands
sander.jansen@tno.nl

Virtual humans (VHs) are employed in many interactive applications, including (serious) games. The motion of these VHs should look realistic. We use the term *naturalness* for such observed realism(van Welbergen et al., 2009a). Furthermore, VH animation techniques should be flexible, to allow interaction with its surroundings and other (virtual) humans in real time. Physical controllers offer physical realism and (physical) interaction with the environment. Because they typically act on a selected set of joints, it is hard to evaluate their naturalness in isolation.

We propose to augment the motion steered by such a controller with motion capture, using a mixed paradigm animation(van Welbergen et al., 2009b) that creates coherent full body motion. A user evaluation of this resulting motion assesses the naturalness of the controller in isolation. This is done by comparing the augmented motion with full body motion capture of the same movement.

Methods from Signal Detection Theory (Macmillan and Creelman, 2004) provide us with the bias-independent sensitivity metric d' that can be compared among these different test setups, observers and motions (see Figure 1). This metric indicates how well two motions can be discriminated. We use the d' of motion captured motion compared with the model based motion as a naturalness measure for the model based motion. Additionally, a naturalness rating test is used to directly asses naturalness.

We discus different test paradigms and asses their efficiency. An efficient test-paradigm has a d' with a low variance within each test condition and large differences between d'-s measured in different test conditions, so that it is easy to make significant observations on discrimination differences in different test conditions.

We demonstrate our approach by evaluating the naturalness of a balance controller (Wooten and Hodgins, 2000) that acts on the legs and trunk, in comparison to motion captured motion and motion in which all trunk and leg movement is omitted. We also asses the effect of several presentation factors on naturalness. Details of the test setup and a full analysis of the results can be found in (Jansen and van Welbergen, 2009).

Acknowledgments

This research has been supported by the GATE project, funded by the Dutch Organization for Scientific Research (NWO) and the Dutch ICT Research and Innovation Authority (ICT Regie).

References

Jansen, S. E. M. and van Welbergen, H. (2009). Methodologies for the user evaluation of the motion of virtual humans. In *Submitted to Interactive Virtual Agents*.

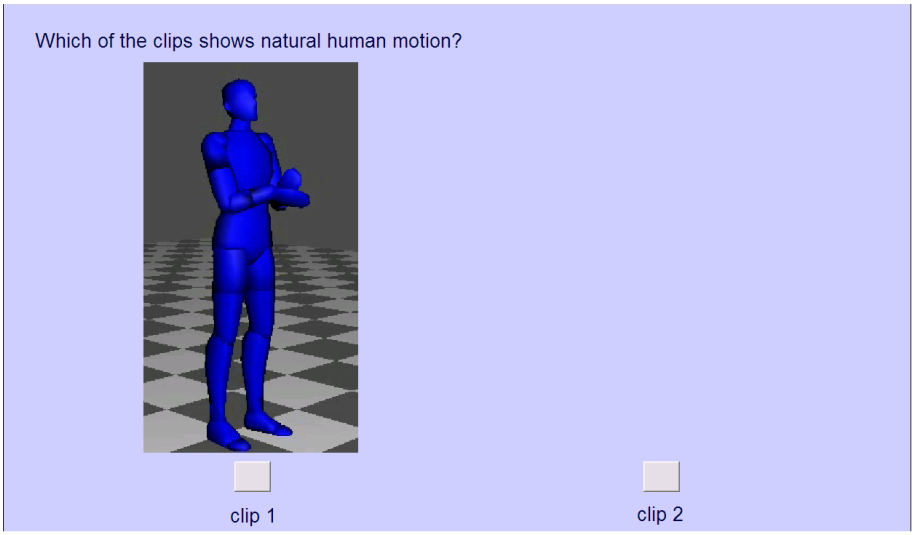
Macmillan, N. A. and Creelman, D. C. (2004). *Detection Theory: A User's Guide*. Lawrence Erlbaum, 2 edition.

van Welbergen, H., van Basten, B. J. H., Egges, A., Ruttkay, Z., and Overmars, M. H. (2009a). Real Time Animation of Virtual Humans: A Trade-off Between Naturalness and Control. In

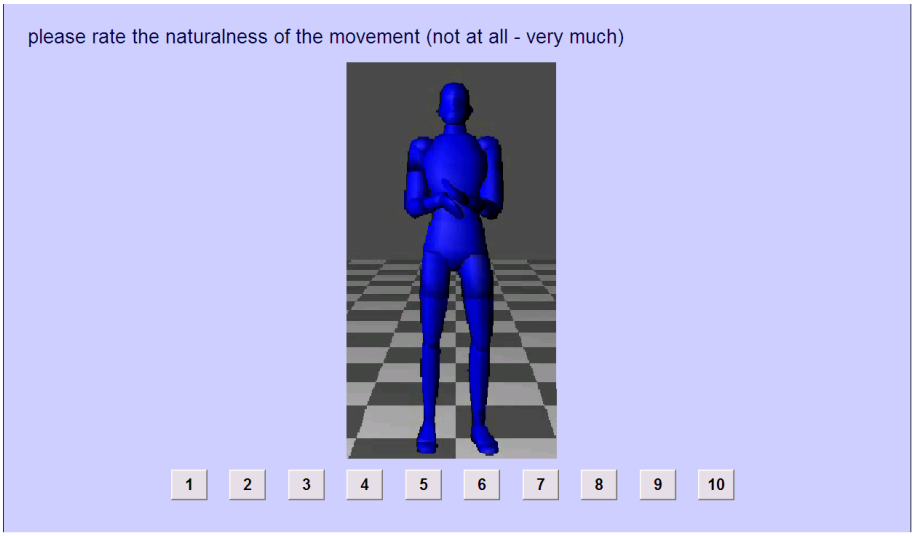
Pauly, M. and Greiner, G., editors, *Eurographics 2009 - State of the Art Reports*, pages 45–72, Munich, Germany. Eurographics Association.

van Welbergen, H., Zwiers, J., and Ruttkay, Z. (2009b). Real-time animation using a mix of dynamics and kinematics. *Submitted to Journal of Graphics Tools*.

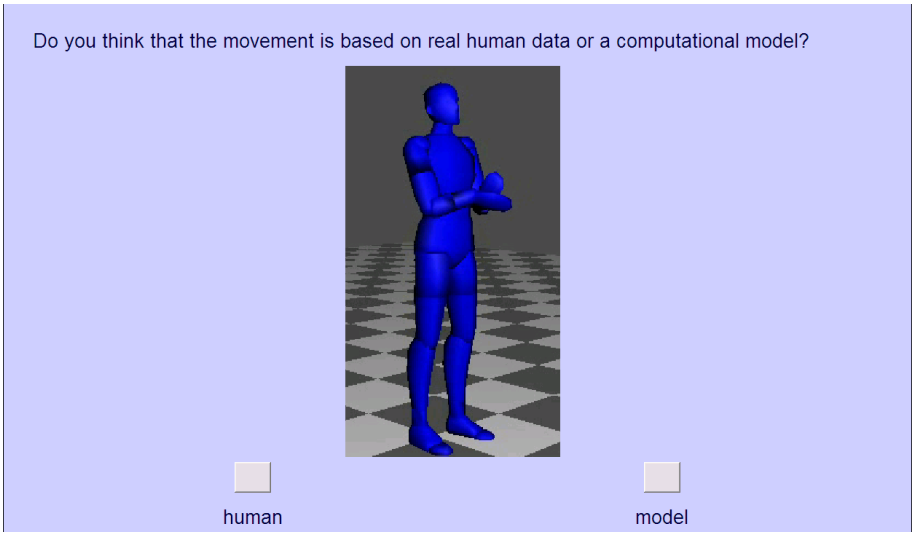
Wooten, W. L. and Hodgins, J. K. (2000). Simulating leaping, tumbling, landing, and balancing humans. In *International Conference on Robotics and Animation*, pages 656–662.



(a) 2 Alternatives Forced Choice test



(b) Rating test



(c) Yes/No test

Figure 1: Test paradigms and viewing angles used in the evaluation study.

List of authors