Journal Subline

LNCS 7380

Transactions on Computational Science XVI

Marina L. Gavrilova · C. J. Kenneth Tan

Editors-in-Chief

Springer

Lecture Notes in Computer Science 7380

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Marina L. Gavrilova C.J. Kenneth Tan (Eds.)

Transactions on Computational Science XVI

Springer

Volume Editors

Marina L. Gavrilova
University of Calgary, Department of Computer Science
2500 University Drive N.W., Calgary, AB, T2N 1N4, Canada
E-mail: marina@cpsc.ucalgary.ca

C.J. Kenneth Tan
Exascala Ltd.
Unit 9, 97 Rickman Drive, Birmingham B15 2AL, UK
E-mail: cjtan@exascala.com

ISSN 0302-9743 (LNCS)
e-ISSN 1611-3349 (LNCS)
ISSN 1866-4733 (TCOMPSCIE)
e-ISSN 1866-4741 (TCOMPSCIE)
ISBN 978-3-642-32662-2
e-ISBN 978-3-642-32663-9
DOI 10.1007/978-3-642-32663-9
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: Applied for

CR Subject Classification (1998): H.5, I.2, H.3-4, C.2, I.3, I.4

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

LNCS Transactions on Computational Science

Computational science, an emerging and increasingly vital field, is now widely recognized as an integral part of scientific and technical investigations, affecting researchers and practitioners in areas ranging from aerospace and automotive research to biochemistry, electronics, geosciences, mathematics, and physics. Computer systems research and the exploitation of applied research naturally complement each other. The increased complexity of many challenges in computational science demands the use of supercomputing, parallel processing, sophisticated algorithms, and advanced system software and architecture. It is therefore invaluable to have input by systems research experts in applied computational science research.

Transactions on Computational Science focuses on original high-quality research in the realm of computational science in parallel and distributed environments, also encompassing the underlying theoretical foundations and the applications of large-scale computation. The journal offers practitioners and researchers the opportunity to share computational techniques and solutions in this area, to identify new issues, and to shape future directions for research, and it enables industrial users to apply leading-edge, large-scale, high-performance computational methods.

In addition to addressing various research and application issues, the journal aims to present material that is validated – crucial to the application and advancement of the research conducted in academic and industrial settings. In this spirit, the journal focuses on publications that present results and computational techniques that are verifiable.

Scope

The scope of the journal includes, but is not limited to, the following computational methods and applications:

- Aeronautics and Aerospace
- Astrophysics
- Bioinformatics
- Climate and Weather Modeling
- Communication and Data Networks
- Compilers and Operating Systems
- Computer Graphics
- Computational Biology
- Computational Chemistry
- Computational Finance and Econometrics
- Computational Fluid Dynamics
- Computational Geometry

- Computational Number Theory
- Computational Physics
- Data Storage and Information Retrieval
- Data Mining and Data Warehousing
- Grid Computing
- Hardware/Software Co-design
- High-Energy Physics
- High-Performance Computing
- Numerical and Scientific Computing
- Parallel and Distributed Computing
- Reconfigurable Hardware
- Scientific Visualization
- Supercomputing
- System-on-Chip Design and Engineering

Editorial

The Transactions on Computational Science journal is part of the Springer series *Lecture Notes in Computer Science*, and is devoted to the gamut of computational science issues, from theoretical aspects to application-dependent studies and the validation of emerging technologies.

The journal focuses on original high-quality research in the realm of computational science in parallel and distributed environments, encompassing the facilitating theoretical foundations and the applications of large-scale computations and massive data processing. Practitioners and researchers share computational techniques and solutions in the area, identify new issues, and shape future directions for research, as well as enable industrial users to apply the techniques presented.

The current volume is devoted to the topic of cyberworlds. It is comprised of eleven extended versions of selected papers from Cyberworlds 2011, held in Banff, Alberta, Canada. The topics span areas of haptic modeling, shared virtual worlds, virtual collaborative spaces, shape modeling, computer vision, augmented and virtual reality, human-computer interfaces, e-learning in virtual collaborative spaces, multi-user web games, cybersecurity, social networking and art and heritage in cyberspaces.

We would like to thank all of the authors for submitting their papers to the issue and the associate editors and referees for their valuable work. We would like to express our gratitude to the LNCS editorial staff of Springer, in particular Alfred Hofmann, Ursula Barth and Anna Kramer, who supported us at every stage of the project.

It is our hope that the fine collection of papers presented in this special issue will be a valuable resource for Transactions on Computational Science readers and will stimulate further research into the vibrant area of computational science applications.

March 2012

Marina L. Gavrilova
C.J. Kenneth Tan

LNCS Transactions on Computational Science – Editorial Board

Table of Contents

Function-Based Single and Dual Point Haptic Interaction in Cyberworlds . . . 1
Lei Wei, Alexei Sourin, Zoran Najdovski, and Saeid Nahavandi

Visual 3D Perception of Motion Environment and Visibility Factors in Virtual Space . . . 17
Vladimir Aleshin, Valery Afanasiev, Alexander Bobkov, Stanislav Klimenko, Vitaly Kuliev, and Dmitry Novgorodtsev

Continuous Control of Style and Style Transitions through Linear Interpolation in Hidden Markov Model Based Walk Synthesis . . . 34
Joëlle Tilmanne and Thierry Dutoit

Modeling and Analyzing the Human Cognitive Limits for Perception in Crowd Simulation . . . 55
Vaisagh Viswanathan and Michael Lees

Experiments in Artimetrics: Avatar Face Recognition . . . 77
Roman V. Yampolskiy, Gyuchoon Cho, Richard Rosenthal, and Marina L. Gavrilova

Visualization of Joinery Using Homotopy Theory and Attaching Maps . . . 95
Kenji Ohmori and Tosiyasu L. Kunii

Example of Business Applications Using the Numerical Value and Exponential Calculation of the Cellular Data System . . . 115
Toshio Kodama, Yoichi Seki, and Tosiyasu L. Kunii

Analysis of Inverse Snyder Optimizations . . . 134
Erika Harrison, Ali Mahdavi-Amiri, and Faramarz Samavati

Error-Controllable Simplification of Point Cloud . . . 149
Yichen Li, Mingqiang Wei, Jianhuang Wu, and Mingyong Pang

Physically-Based Haptic Rendering for Virtual Hand Interaction . . . 163
Yang Wenzhen, Pan Zhigen, and Chen Wenhua

The Advanced Open Metaplastic Platform for Cyber Art . . . 179
Gianluca Mura

Author Index . . . 191

Function-Based Single and Dual Point Haptic Interaction in Cyberworlds

Lei Wei[1], Alexei Sourin[2], Zoran Najdovski[1], and Saeid Nahavandi[1]

[1] Centre for Intelligent System Research, Deakin University, Australia
{lei.wei,zoran.najdovski,saeid.nahavandi}@deakin.edu.au
[2] Nanyang Technological University, Singapore
assourin@ntu.edu.sg

Abstract. Polygon and point based models dominate virtual reality. These models also affect haptic rendering algorithms, which are often based on collision with polygons. With application to dual point haptic devices for operations like grasping, complex polygon and point based models will make the collision detection procedure slow. This results in the system not able to achieve interactivity for force rendering. To solve this issue, we use mathematical functions to define and implement geometry (curves, surfaces and solid objects), visual appearance (3D colours and geometric textures) and various tangible physical properties (elasticity, friction, viscosity, and force fields). The function definitions are given as analytical formulas (explicit, implicit and parametric), function scripts and procedures. We proposed an algorithm for haptic rendering of virtual scenes including mutually penetrating objects with different sizes and arbitrary location of the observer without a prior knowledge of the scene to be rendered. The algorithm is based on casting multiple haptic rendering rays from the Haptic Interaction Point (HIP), and it builds a stack to keep track on all colliding objects with the HIP. The algorithm uses collision detection based on implicit function representation of the object surfaces. The proposed approach allows us to be flexible when choosing the actual rendering platform, while it can also be easily adopted for dual point haptic collision detection as well as force and torque rendering. The function-defined objects and parts constituting them can be used together with other common definitions of virtual objects such as polygon meshes, point sets, voxel volumes, etc. We implemented an extension of X3D and VRML as well as several standalone application examples to validate the proposed methodology. Experiments show that our concern about fast, accurate rendering as well as compact representation could be fulfilled in various application scenarios and on both single and dual point haptic devices.

Keywords: Function-based approach, Haptic interaction, Dual-point haptic interaction, Haptic collision detection, Haptic force rendering, Shared Virtual Spaces.

1 Introduction

Haptics refers to the incorporation of touch sense into computer programs by providing force feedback which is delivered by haptic devices. In visual rendering, the most

M.L. Gavrilova and C.J.K. Tan (Eds.): Trans. on Comput. Sci. XVI, LNCS 7380, pp. 1–16, 2012.

frequently used models are polygon-based and point-based models. In most application scenarios, the haptic rendering pipeline is attended after the visual rendering pipeline. This also affects how the haptic rendering deals with model representations. Therefore, both visual and haptic rendering with polygons suffer from the same problem in polygon-based visualization–polygon complexity. After a certain number of polygons participated in the scene, haptic rendering cannot be performed with the required 1000 Hz frequency. Since visual rendering requires as low as 30 Hz for refreshing images, this imposes two orders of magnitude more demand of the processing speed and reduces, respectively, the number of polygons which can be processed haptically. Things get even worse when collaborative haptic interaction is performed.

There are efforts in solving this problem from several perspectives, such as hierarchies of bounding volumes [1], spatial partitioning [2], GPU-based acceleration [3], and a velocity driven haptic rendering approach [4]. Besides, there are efforts in solving this problem by using mathematical functions. For example, it is rather trivial for implicit functions to implement the collision with the contact point membership predicate. In [5, 6], we proposed an approach to haptically collide with implicit functions, which does not need the primitive level input. In [7], another implicit function based approach was proposed for rendering large geometry models at 1000 Hz rate. However, it retrieves surface information from volumetric data for penalty-based force generation. In [8], a solution was proposed to haptically render the volume data using dynamic spline-based implicit functions. However, these approaches usually consider only one type of mathematical functions in haptic collision detection and force rendering, and therefore they have many restrictions and limitations.

On the other hand, drawbacks and restrictions on using single point haptic interaction have been widely identified. In [9], the authors declare that stylus point interaction is limited due to minimized haptic perception for object identification. To achieve haptic operations such as virtual object grasping and twisting, a dual point haptic interface has to be involved [10, 11]. A dual point haptic interface will allow an operator to directly interact with virtual objects through finger contact rather than point interaction through a stylus. In [12] it demonstrates that the use of more than one single point of contact can increase a person's ability to better discriminate the object being contacted. Various forms of dual point haptic interfaces exist. They range from the more straightforward approach utilizing multiple single point haptic interfaces [13], to the more elaborate utilizing a full hand exoskeleton [14]. Although the dual point haptic interface can greatly expand the usability of the typical desktop haptic devices, it literally doubles the requirement for interactive haptic collision detection and force rendering, which makes the polygonal-based approach even more unfeasible.

With all abovementioned problems on haptic collision detection and force rendering, it is clear that a **fast** and **accurate** approach with **compact representation** for both **single** and **dual** point haptic interaction is in need. Motivated by these problems, in this paper, we seek to propose a function-based approach haptic interaction with objects containing various haptic properties, different geometric models, as well as different sizes and locations.

We discuss the challenges and propose our solutions to existing haptic interaction problems in Section 2, and give implementation details for single point haptic interaction in Section 3. In Section 4 we briefly introduce a dual point haptic device system and apply our approach onto it to render force as well as torque. In Section 5, we discuss technical details and show several practical examples to illustrate our approach. Finally, in Section 6 we summarize the completed work.

2 Literature Review

We define geometry, visual appearance and physics properties of the objects using various function definitions given as analytical formulas, function scripts and procedures. The functions in our approach serve both as elements of a descriptive language as well as core parts of the respective rendering algorithms. This approach allows us to be flexible when choosing the actual rendering platform. Mathematically, there are three ways of function definitions: explicit, implicit, and parametric. In our approach, we allow for mixing these three definitions to devise the most efficient representation of the objects. These function-defined objects and parts constituting them (geometry, appearance and physics) can be used together with other common definitions of objects such as polygon meshes, point sets, voxel volumes, etc. The approach allows us to enrich the existing models and open them to practically unlimited opportunities of defining their geometry, 3D colours, geometric textures, and tangible physical properties that can be rendered with haptic devices.

Our work has roots in [15], where F-Rep representation of geometric objects was proposed. In [16, 17], the original idea was extended to solid texturing and hypervolumes. In [18, 19], the authors further expanded the problem scope to shared virtual spaces and metamorphoses.

We propose to use functions to define and implement geometry, visual appearance (3D colours and geometric textures) and various tangible physical properties such as tension, friction, viscosity, and force fields. Since the defining functions are very small compared to polygon meshes and point sets, we can efficiently use them in collaborative virtual environments to exchange between the participating clients.

Geometric curves can be conveniently defined by parametric functions as: $x = f_1(u|,t)$, $y = f_2(u|,t)$, $z = f_3(u|,t)$. Geometric surfaces can be defined by either using implicit function *f(x,y,z,t) = 0* or by parametric functions as: $x = f_1(u|,v|,t)$, $y = f_2(u|,v|,t)$, $z = f_3(u|,v|,t)$. Solid geometry could be defined using explicit function *g = f(x,y,z,t) ≥0*, or defined by parametric functions $x = f_1(u|,v|,w|,t)$, $y = f_2(u|,v|,w|,t)$, $z = f_3(u|,v|,w|,t)$. Addition of time t allows for definition of animated objects.

3D colour can be defined as explicit functions *g = f(x,y,z,t)*, followed by colour interpolation to get the respective RGB values or directly defined by parametric functions $r = \varphi_1(u|,v\,|,w|,t)$, $g = \varphi_2(u|,v|,w|,t)$, $b = \varphi_3(u|,v|,w|,t)$. 3D geometric texture can be defined by distance functions either explicitly as displacement *d = f(x,y,z,t)* or parametrically $x = f_1(u|,v|,w|,t)$, $y = f_2(u|,v|,w|,t)$, $z = f_3(u|,v|,w|,t)$.

Surface physical properties (including tension and friction), as well as density, can be defined as explicit functions *g = f(x,y,z,t)*. For each point on the object surface and inside the object, a corresponding tension, friction and density value will be calculated and rendered. Force field property can be defined using parametric functions $x = f_1(u|,v|,w|,t)$, $y = f_2(u|,v|,w|,t)$, $z = f_3(u|,v|,w|,t)$ at each point inside the force field, a corresponding force vector [*x y z*] will be generated to push the haptic handle to a new point.

3 Algorithm Details for Single Point Haptic Interaction

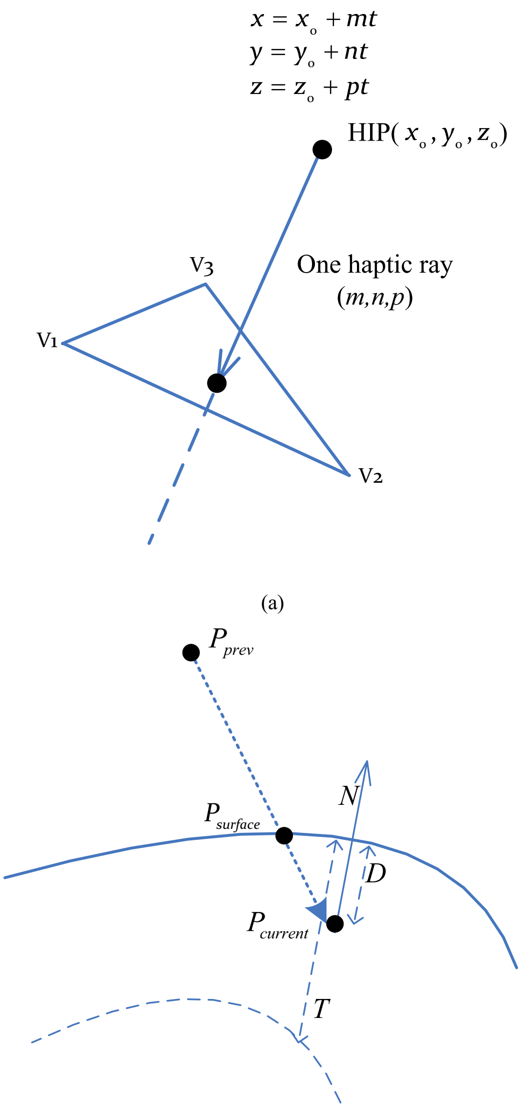
Our algorithm is based on casting multiple haptic rendering rays from the HIP. Each haptic ray cast is tested for possible intersections with haptic containers. The containers can be represented either by commonly used polygon mesh-based surfaces or defined by mathematic functions (parametric and implicit). In case of polygon meshes, a trivial intersection of a parametrically defined ray with the polygons is performed (Fig. 1a). It can be accelerated by many commonly used ways. In case of parametrically defined surfaces of the containers, the polygon mesh is first calculated and then intersected with the rays. In case of implicitly defined surfaces, the haptic ray intersection can be performed at the level of function definitions. **N** is the approximated normal at $P_{current}$, *T* is the surface zone thickness at $P_{current}$, and *D* is the penetrated depth at $P_{current}$. This procedure is similar to that of ray tracing and is performed by iterative binary subdivision steps which eventually result in the coordinates of the intersection point on the surface of the container (Fig. 1b). Alternatively, the implicitly defined surface can be also polygonized first, however the direct evaluation method has a higher precision, requires less computation and hence provides for a higher performance.

HIP position location assumes understanding of whether the HIP is inside or outside the haptic container, and, if inside, how far from the surface it is located. In case of polygonal representation of the containers, the inside/outside predicate is implemented as a dot product

$$g=\mathbf{R}\cdot\mathbf{N} \tag{1}$$

of a haptic ray **R** and a normal vector **N** to the surface at the intersection point (Fig. 2a). The HIP is inside the container if g>0 and outside if g<0. The distance to the surface then is calculated as a minimum of distances to polygon planes for the rays cast $D=\frac{|\mathbf{R}\cdot\mathbf{N}|}{\|\mathbf{N}\|}$ (Fig. 2b).

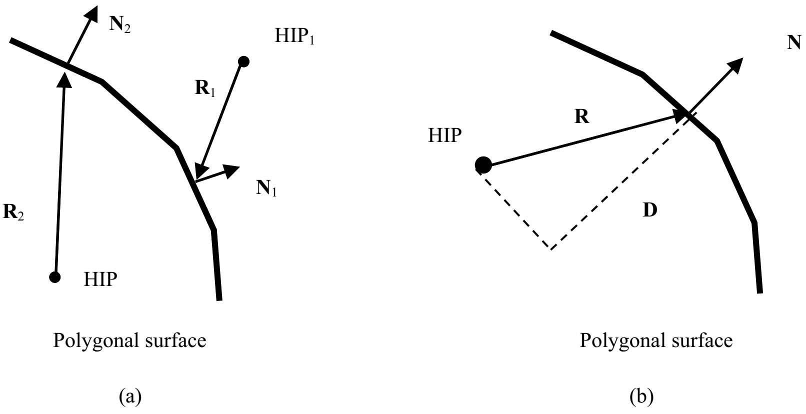
In case of implicitly defined surfaces of the container, the inside/outside predicate is evaluated by analysis of the surface defining function sign which changes on different sides of the surface (Fig. 3). Hence, using function definitions $g=f(x,y,z)\geq 0$ for the containers, HIP is inside if g>0 and outside if g<0. We also use a gradient to approximate the normal at $P_{current}$. Depending on the function definitions, there can be cases of degenerated normal calculated at $P_{current}$. In that case we calculate gradient at $P_{surface}$ (the intersection point of the haptic ray and haptic container) rather than $P_{current}$.



(a)

(b)

Fig. 1. a) Intersection test between multiple haptic rays and polygon-based haptic containers, b) Intersection test between multiple haptic rays and function-based haptic containers



(a)

(b)

Fig. 2. a) HIP position location for polygonal haptic container, b) HIP distance calculation for polygon-based haptic container

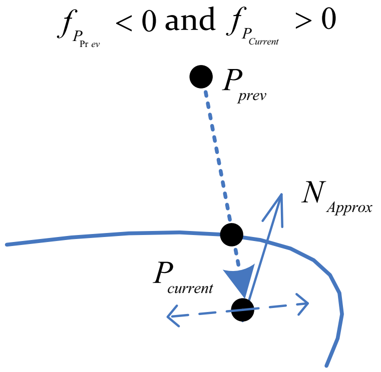


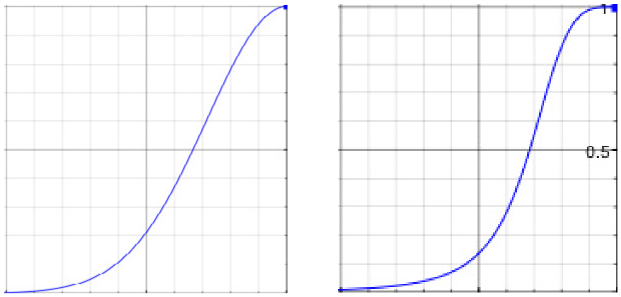
Fig. 3. HIP location and distance calculation for implicit-based haptic containers

The distance is then calculated using the same formula as for the polygonal representation.

Surface properties rendering activates when the HIP is inside the container within its surface zone. Forces such as tension and friction generate in the direction opposite to the HIP movement.

$$\mathbf{F}_{Tension} = T_{clipped} * \mathbf{F}_{Device\_Max} * f(D/L_{SurfaceZone}) \qquad (2)$$

where $f(D/L_{SurfaceZone})$ represents different profile functions which are used for mapping penetrated distance into tension force magnitude (Fig. 4). These mapping profile functions enable users to fine tune the tension force on different object surfaces as well as on different parts of one object surface.



$e^{-(x*2.5)^2}$ $1/(100*t^4+1)$

Fig. 4. Various profile functions for mapping penetrated distance into tension force magnitude

The surface friction is calculated after tension calculation:

$$\mathrm{F}_{Friction} = -\ \mathrm{Friction}_{current\_Pos} * \left|\mathrm{F}_{Tension}\right| .* V_{interpolated} \tag{3}$$

where $\left|\mathrm{F}_{Tension}\right|$ is the magnitude of the tension force and $\mathrm{Friction}_{current\_Pos}$ is the friction value of the HIP.

Inner properties rendering activates when the HIP is inside the haptic container but not within the surface zone. For density rendering, the force will generate in the direction opposite to the HIP movement and proportional to the density value. We obtain the velocity of HIP at the haptic frequency and interpolate it as follows:

$$V_{current\ interpolated} = C_1 * V_{previous} + C_2 * V_{current} \tag{4}$$

where $V_{previous}$ is the velocity at 1/1000 sec before current haptic frame and C_1 and C_2 are interpolation coefficients.

The feedback force is then calculated as follows:

$$F = V_{current\ interpolated} * L \tag{5}$$

where L is the function-defined density evaluated at the HIP.

When a force field within a haptic container is defined, the forces there define the direction of the HIP movement (Fig. 5a). Since the haptic pipeline executes at 1000Hz, the HIP will be continuously guided to move from one point to another (Fig. 5b).

We also proposed an algorithm for haptic rendering of virtual scenes including mutually penetrating objects with different sizes and arbitrary location of the observer, as shown in Fig. 6.

Typical haptic rendering is based on rigid objects, although there are also a number of research works on haptic interaction with deformable objects [20, 21]. The proposed stack-based algorithm is aimed at solving haptic rendering problems with mutually penetrating objects, and it could be applied to both rigid and deformable objects. In real life it corresponds to placement of one solid object inside another solid object, i.e. in medical applications it can be an implant or a lump surrounded by human body tissues.

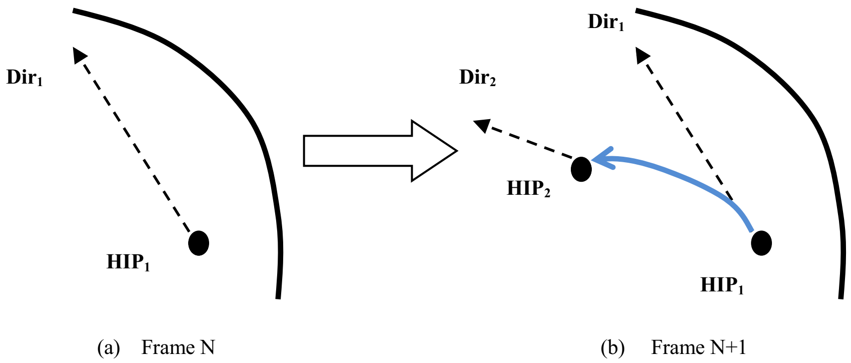


Fig. 5. Changes of movement direction between haptic frames generate smooth HIP moving path

In haptic rendering, most commonly used approaches are based on polygon-based models, which only have object surfaces defined. Therefore the situation of mutually penetrated objects does not need to be considered. On the other hand, function-based models could be defined as both surface and solid models. Since solid models have interior properties defined, the question is raised of how to correctly render haptic effects when two or more solid objects mutually penetrate, as their overlapping parts may contain different haptic effect definitions from several objects. Such a problem also exists in visual rendering when two solid objects with different colors mutually penetrate. The "correct" color for the overlapping part is questionable and usually is set by the programming system.

Hence, mutually penetrating objects require a proper solution and we propose three options for it:

1. The haptic effect for the overlapping part is determined by the last object the haptic handle has penetrated.
2. The haptic effect for the overlapping part is determined by intersection operation of some or all of the involved objects.
3. The haptic effect for the overlapping part is determined by union operation of some or all of the involved objects.

Depending on the content creator's intention, any of the three options could be adopted. We choose to adopt the first option but other options can also be easily incorporated into the existing function-based haptic interaction pipeline.

The algorithm builds a stack to keep track on all the objects colliding with the HIP. When the HIP penetrates a new solid object, the object name will be pushed into the stack; when the HIP leaves the solid object, the object name will be removed from the stack. At any moment, the current rendering haptic effect will be retrieved from the top most object in the stack. This makes sense in most cases, however there can be some confusing exceptions. For example, if the HIP penetrates a series of objects twice from different directions, the last object the HIP penetrates may be different; therefore the rendering result for the most internal object may also vary.

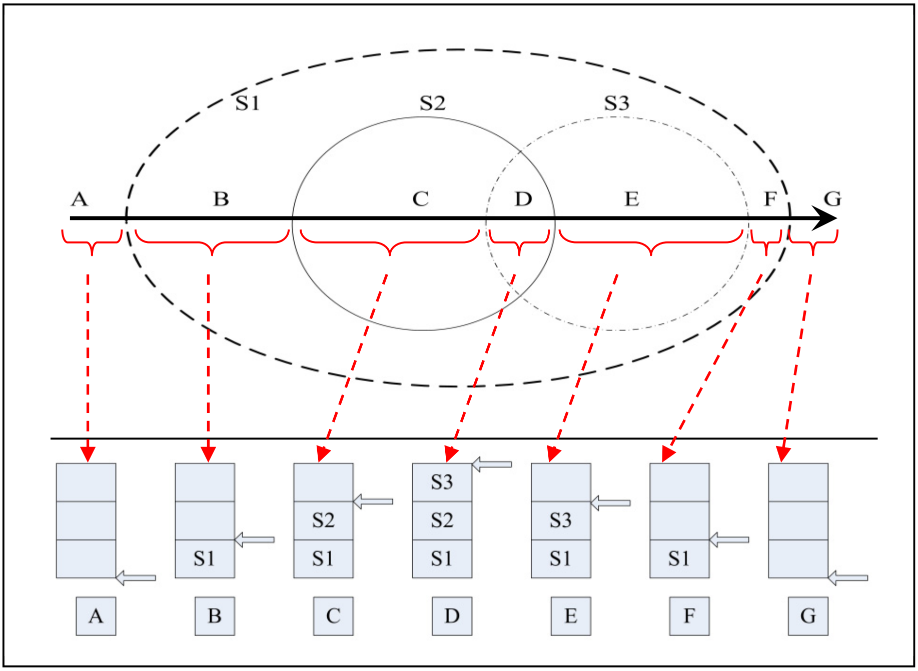


Fig. 6. Inclusion & Intersection graph and stack analysis

In contrast to the commonly used predefined location of the haptic tool and prior knowledge of the scene and haptic effects to be rendered, the stack-based algorithm does not require such prior knowledge and could still identify HIP location and haptic effects to be rendered on the fly. In Fig. 6, there are three objects S1 (dashed line), S2 (solid line) and S3 (dash-dot-dash line) defined with different haptic effects on their surfaces as well as inside them. S1 is the biggest object which includes mutually penetrating S2 and S3. The HIP moves along the direction indicated by the bold line from left to right and collides with the surfaces of the three objects six times. A stack is constructed to record collision history with all objects. A pointer which always points to the stack top is also used to indicate current haptic effects. While the HIP is at segment A, it has not collided with any of the objects, therefore the stack is empty and no haptic effects will be rendered. While the HIP is within segment B, it has penetrated S1 therefore S1 was put into stack. HIP is now inside S1 but still outside S2 and S3, and haptic effects of S1 should be rendered. While HIP is within segment C, it has penetrated S2 and therefore S2 was put into the stack. Current HIP is within both S1 and S2 but S2 is the last penetrated object, therefore haptic effects of S2 interior will be rendered. While HIP is within segment D, it has penetrated all three objects and S3 is the last penetrated object, therefore haptic effects of S3 should be rendered. While HIP is within segment E, the HIP has already left S2 and therefore S2 was removed from the stack. While HIP is within segment F, the HIP has left S3, and S3 was removed from the stack. The pointer is now pointing to S1 with the haptic effects we need to render (the HIP is still within S1). While HIP is within segment G, the HIP has left S1 and the stack was cleared, therefore no haptic effects will be rendered.

Usually, most haptic scenes are still not as complicated as visual scenes. Typically, haptic scenes only have several objects with haptic effects at predefined positions. Therefore mutually penetrating objects were generally avoided or not specifically processed in most haptic collision and rendering algorithms and SDKs.

4 Hardware and Algorithm Details for Dual Point Haptic Interaction

4.1 Brief Introduction to Hardware Design

The dexterity of a user's hand requires a hand interface that will not restrict the natural motion of the fingers throughout any grasping procedure. The dual point haptic interface was designed to follow the natural grasp closure of the operator's hand. This is achieved by use of a serial link structure that allows each finger interface point to move around a central axis and also within a vertical motion as shown in Fig. 7. To aid in the design and analysis of the hand interface, the kinematic relationship between the joint angles and the position of the finger interaction points was determined. This allows for tracking the position of the operator's fingertips in any possible pinch grasp configuration or finger motion.

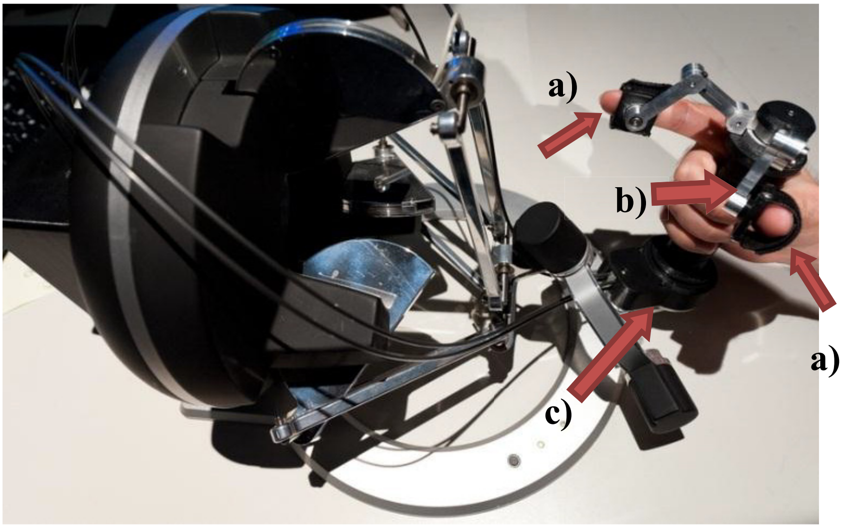


Fig. 7. Dual point haptic interface attached to a Force Dimension haptic interface, where a) is the Finger attachment points, b) is the device holding position, and c) is the interfaces position for commercial haptic devices

The haptic grasping hand interface acts as an extension of the attached haptic interface (Force Dimension haptic device), and consequently the kinematics of the presented device is to be utilized with respect to that device's position and orientation in the workspace.

4.2 Algorithm Details for Dual Point Haptic Interaction

We adopt the same collision detection and force rendering approach for single point haptic device here to both HIPs. Rays are cast from each of the HIPs and seek possible collisions with scenario objects defined in mathematical functions. As the user can only stretch a certain distance between his thumb and index finger, we also introduce one more parameter to control the scaling between the two figures and thus make it adapt to different sizes of application scenarios. The dual point interface is attached to the end-effector position of a commercial haptics device (main device), which enables the forces from the device to be incorporated into the grasping forces of the dual point interface. This leads to gravitational forces being applied to the user when they grasp an object, ultimately being able to feel the object's mass. The approach taken to render grasp forces was achieved using two methods.

We have proposed two approaches of mapping two HIPs to the original HIP as well as new end effectors.

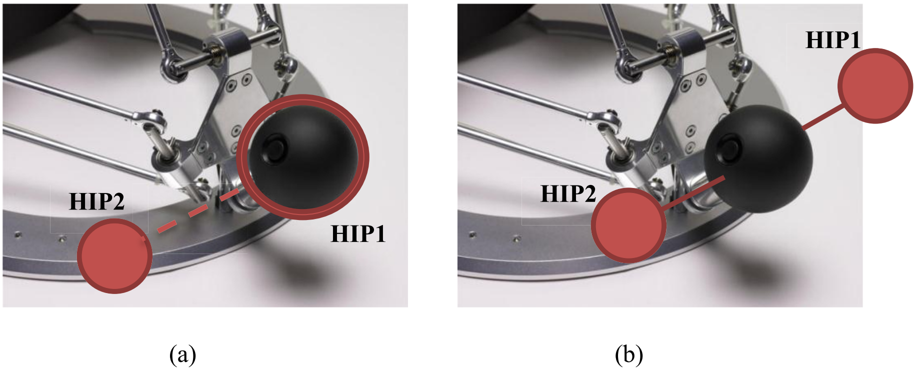


Fig. 8. a) Using original HIP with one new end effector as the second HIP, b) Using two new end effectors as HIPs with original HIP in the center

The first approach (Fig. 8a) utilizes the original end effector on a single point haptic device and connects the second HIP to a newly added end effector, which is further connected to an added force motor. With this approach the main device avatar was used to track the position of the fingers in the virtual world, and also render collision forces that were made in a non-grasping scenario, for example, when the fingers contacted an object without grasping it. When a grasping action was required, the main device's forces were only generated for gravity effects and the grasp forces were solely produced by the dual point device.

The second approach (Fig. 8b) considers two added force motors as the new HIPs, with the original HIP in the middle of them and on the axis of rotation. This approach resulted in less complexity for collision between scenario objects and the HIPs, as switching between the main device force rendering and the dual point device force rendering was removed. This means that collision between the objects and the HIPs was simplified. Any collision that occurred that was not a grasping scenario resulted in the main device generating forces to the user to restrict them from penetrating the virtual object.

We also developed an algorithm to render torque to the user's wrist based on the rotation movement from the grasped object. As the device's finger points rotate centrally to the user's hand, and each finger force can be generated bi-directionally, a torque was rendered as a component of the two independent finger forces. This was achieved in scenarios that required the user to rotate or twist the object, such as a screwdriver, or to feel the effect of inertia on the held object (Fig. 9).

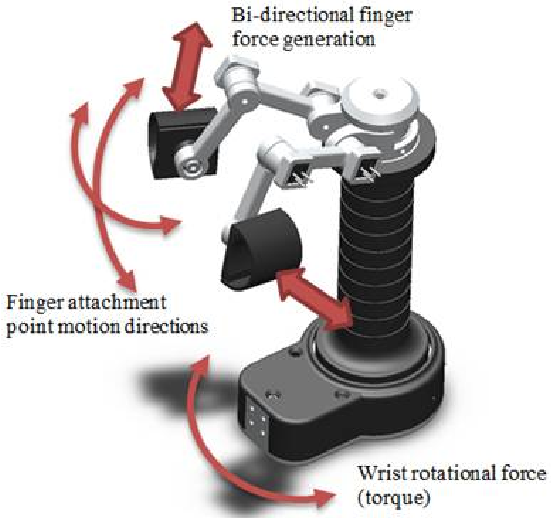

Bi-directional finger force generation
Finger attachment point motion directions
Wrist rotational force (torque)

Fig. 9. 3D CAD model of the haptic grasping hand interface showing bi-directional finger force components, finger motion directions and torque generation

5 Technical Details and Application Examples

We implemented our algorithm as a plug-in to BS Contact VRML/X3D browser which works with MS Internet Explorer and Mozilla Firefox. A specially crafted string in the url field of the Script node of this browser allows it to automatically load and execute functions from a dynamic link library file. The library is written in C++, which generates native codes running on the machine. The parser module of the plug-in is implemented using Yet Another Compiler Compiler (YACC) for C. The input file of YACC is a text file which contains the grammar definition of the programming language which is written in a form similar to Backus Naur Form (BNF). By running the YACC executable, the input file is converted into a C/C++ source file, which parses the grammar of the function definitions.

Implementation with dual point haptic device is based on the OpenHaptics Toolkit and C++ as standalone executable files. The same haptic collision detection and force rendering algorithms are adopted and extended for handling torque rendering as well as relative position scaling between the two HIPs. We take both force and torque into consideration and hardcode different mathematical functions into the source files specific to various scenarios.

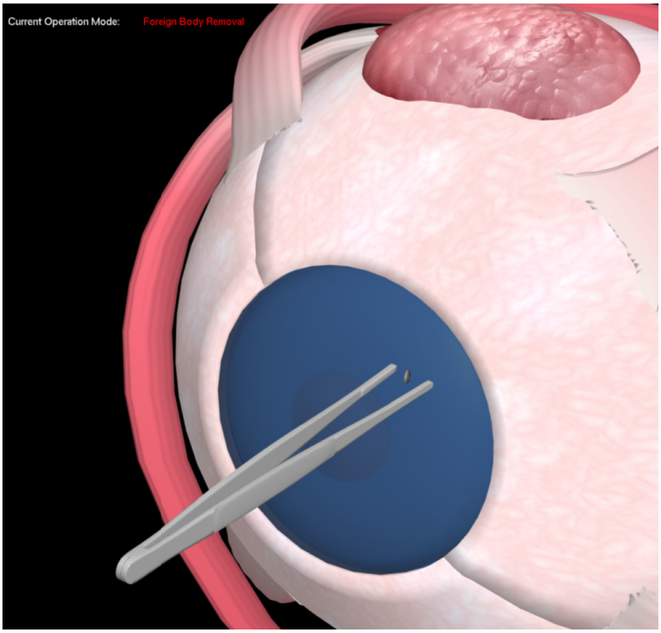


Fig. 10. Function-based collision detection and force rendering of an optometry procedure on an eye model

Fig. 10 illustrates an example of haptic rendering of an optometry surgery procedure (foreign body removal) on an eye model. The model consists of several sub models and some of them are mutually penetrated (i.e. the foreign body, the cornea and the iris). For visual rendering, the eye model is polygonized with a huge number of polygons. However, such a model is prohibitive for haptic collision detection and force rendering. Therefore for collision detection and force generation, different implicit functions were used based on the proposed haptic collision and force rendering algorithm to describe the model haptically. Two implicit functions are used to represent different frictions and tensions on muscles and cornea of the eye. Another implicit function is used to represent variable density inside the eye to ensure that injections from different directions to the eye will have distinct force rendering results. Besides, user interactions such as foreign body removal and corresponding force rendering (attraction force when pull the foreign body out of the eye) have also been implemented in order to provide better immersion.

Fig. 11 illustrates an example of dual point haptic collision detection and force rendering of a virtual assembly procedure. The assembly model is displayed with huge number of polygons for fine details. For haptic interaction, we create an invisible bounding box to trigger the function-based collision detection when the HIPs approach the assembly model. The model surface is defined with implicit functions to describe variant frictions across different sections. By adopting the previously described algorithm to both HIPs, interactive force rendering on both figures could be achieved.

Compared with traditional polygon-based haptic collision detection and force rendering approach, our algorithm is much faster in haptic collision detection and interactive force rendering (1000 HZ) procedures, while still maintaining better accuracy due to the nature of mathematical functions and procedures. These examples also show the advantages of defining various object properties (geometry, visual appearance and physical properties) in separate domains and then merged together to constitute one object.

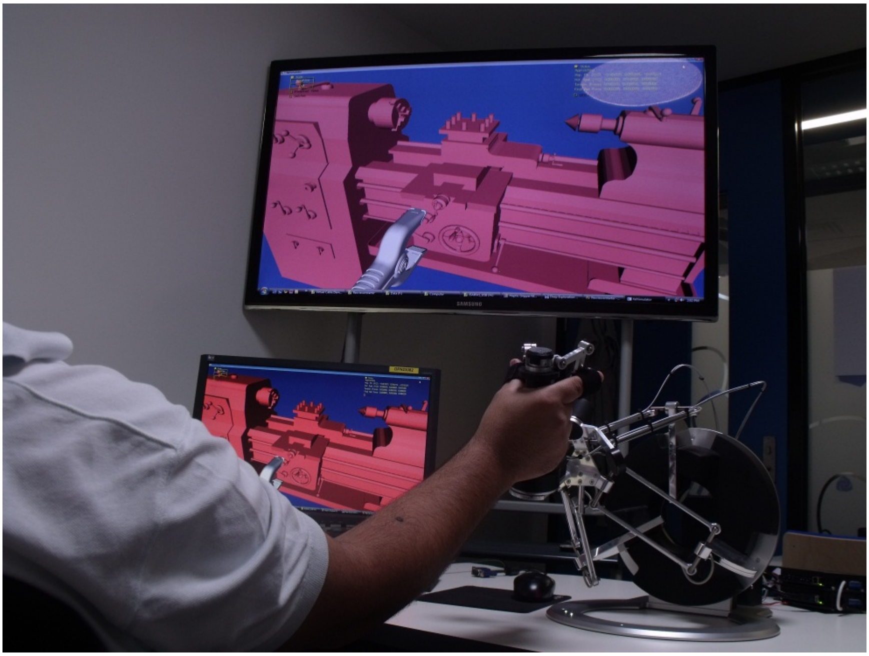


Fig. 11. Function-based collision detection and force rendering of haptic assembly scenario

6 Conclusion

We proposed a function-based approach to define various haptic effects in cyberworlds with mixed geometric models, including polygons meshes, point clouds, image-based billboards and layered textures, voxel models and especially functions-based models of surfaces and solids. We also proposed a way to identify the location of the haptic tool in such haptic scenes as well as consistently and seamlessly determine haptic effects when the haptic tool moves in the scenes with objects having different sizes, locations, and mutual penetrations. We also used implicit function definitions for collision detection and force generation where polygon-based method fails to work. It permitted us to mix models for visual and haptic rendering to achieve the best performance. The proposed haptic collision detection and force rendering algorithms has also been adopted and extended to dual point haptic devices to achieve interaction in complex scenarios.

Acknowledgements. This project is supported by the Singapore National Research Foundation Interactive Digital Media R&D Program, under research Grant NRF2008IDM-IDM004-002 "Visual and Haptic Rendering in Co-Space". This project is also funded by Centre for Intelligent System Research under strategic initiatives.

References

1. Weller, R., Zachmann, G.: A unified approach for physically-based simulations and haptic rendering. In: Proceedings of the 2009 ACM SIGGRAPH Symposium on Video Games, pp. 151–159 (2009)
2. El-Far, N.R., Georganas, N.D., Saddik, A.E.: Collision Detection and Force Response in Highly-Detailed Point-Based Hapto-Visual Virtual Environments. In: Proceedings of the 11th IEEE International Symposium on Distributed Simulation and Real-Time Applications, pp. 15–22 (2007)
3. Knott, D., Pai, D.K.: CInDeR: collision and interference detection in real time using graphics hardware. In: Proceedings of Graphics Interface, pp. 73–80 (2003)
4. Kolčárek, P., Sochor, J.: Velocity driven haptic rendering. In: Proceedings of the 3rd International Conference on Computer Graphics and Interactive Techniques in Australasia and South East Asia, pp. 389–394 (2005)
5. Sourin, A., Wei, L.: Visual immersive haptic rendering on the web. In: Proceedings of the 7th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and Its Applications in Industry (2008)
6. Wei, L., Sourin, A.: Function-based approach to mixed haptic effects rendering. Vis. Comput. 27, 321–332 (2011)
7. Laehyun, K., Kyrikou, A., Sukhatme, G.S., Desbrun, M.: An implicit-based haptic rendering technique. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, vol. 3, pp. 2943–2948 (2002)
8. Hua, J., Qin, H.: Haptics-based volumetric modeling using dynamic spline-based implicit functions. In: Proceedings of the 2002 IEEE Symposium on Volume Visualization and Graphics, pp. 55–64 (2002)
9. Edin, B., Panarese, A.: Human ability to discriminate direction of threedimensional force stimuli applied to the finger pad. J. Neurophysiol. 105, 541–547 (2011)
10. Najdovski, Z., Nahavandi, S.: Characterising a Novel Interface for Event-Based Haptic Grasping. In: IEEE International Symposium on Robot and Human Interactive Communications, Japan (2009)
11. Najdovski, Z., Nahavandi, S.: Extending Haptic Device Capability for 3D Virtual Grasping. In: Ferre, M. (ed.) EuroHaptics 2008. LNCS, vol. 5024, pp. 494–503. Springer, Heidelberg (2008)
12. Jansson, G.: Effects of number of fingers involved in exploration on haptic identification of objects. excerpt from pureform: The museum of pure form; haptic exploration for perception of the shape of virtual objects. EUPURE FORM, Technical report (2000)

13. Bttcher, G., Allerkamp, D., Glckner, D., Wolter, F.-E.: Haptic two-finger contact with textiles. The Visual Computer 24, 911–922 (2008)
14. Mourad Bouzit, G.P., Burdea, G., Boian, R.: The rutgers master iinew design force-feedback glove. IEEE/ASME Transactions on Mechatronics 7, 256–263 (2002)
15. Pasko, A., Adzhiev, V., Sourin, A., Savchenko, V.V.: Function representation in geometric modeling: concepts, implementation and applications. The Visual Computer 11, 429–446 (1995)
16. Pasko, A., Adzhiev, V., Schmitt, B., Schlick, C.: Constructive hypervolume modelling. Graphical Models 63(6), 413–442 (2001)
17. Adzhiev, V., Kartasheva, E., Kunii, T., Pasko, A., Schmitt, B.: Cellular-functional modeling of heterogeneous objects. In: Proc. 7th ACM Symposium on Solid Modeling and Applications, Saarbrücken, Germany, pp. 192–203. ACM Press (2002)
18. Liu, Q., Sourin, A.: Function-defined shape metamorphoses in visual cyberworlds. The Visual Computer 22, 977–990 (2006)
19. Liu, Q., Sourin, A.: Function-defined Shape Metamorphoses in Visual Cyberworlds. The Visual Computer 22(12), 977–990 (2006)
20. Duriez, C., Dubois, F., Kheddar, A., Andriot, C.: Realistic haptic rendering of interacting deformable objects in virtual environments. IEEE Transactions on Visualization and Computer Graphics 12, 36–47 (2006)
21. Barbic, J., James, D.L.: Six-DoF haptic rendering of contact between geometrically complex reduced deformable models: Haptic demo. In: Third Joint EuroHaptics Conference, 2009 and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems. World Haptics 2009, pp. 393–394 (2009)

Visual 3D Perception of Motion Environment and Visibility Factors in Virtual Space

Vladimir Aleshin, Valery Afanasiev, Alexander Bobkov,
Stanislav Klimenko, Vitaly Kuliev, and Dmitry Novgorodtsev

Moscow Institute of Physics and Technology (National Research University),
141700, 9, Institutskii per., Dolgoprudny, Moscow Region, Russia
stanislav.klimenko@gmail.com
http://mipt.ru

Abstract. Visual perception of 3D space is a major source of information for a person. The role of athletes visual perception and corresponding psychomotor actions is very important for adequate actions in sport. Taking into account our team specialization (the virtual environment technology), we decided to investigate the impact of visual 3D perception different parameters on sport results of alpine skiing athletes.

Keywords: virtual reality, human-computer interfaces, induced virtual environment, 3D perception, binocular parallax, vision accommodation, alpine skiing training system, motion parallax, motion parallax inversion.

1 Introduction

Factors influencing the visual perception process are investigated in many different sports, but more often they are studied in ”dynamic” sports in which an athletes success depends on speed and adequate perception of rapidly changing 3D environment.

We investigated 3D visual perception of an alpine skiing athlete who can be considered as a very suitable object for our research. Indeed, a skier is forced to perform visual analysis in extreme conditions with rapidly changing 3D environment and distances between the athlete and objects on the ski course. The importance of the proper skill of “seeing the course” in professional alpine skiing (“Toni” Sailer, http://en.wikipedia.org/wiki/Toni_Sailer) is well known. 3D visual perception is also very important for safety precautions in recreational alpine skiing. In this work, we made an attempt to evaluate the impact of 3D visual perception different factors on sport results in virtual alpine skiing.

2 Related Works

Problems of human 3D visual perception and human-computer interfaces have been investigated for a long time. Two pioneer works in this field are the work of Marr [1] and the work of Rauschenbusch [2]. The book [3] covers motion

M.L. Gavrilova and C.J.K. Tan (Eds.): Trans. on Comput. Sci. XVI, LNCS 7380, pp. 17–33, 2012.

perception in terms of experimental psychology. In particular, neurophysiological tasks of defining a trajectory and motion parameters are considered in this book. One of the first overview books in visual perception and action in sports is [4]. The connection of visual perception and adequate actions of a sportsman has been actively studied in sports psychology, biomechanics, and kinesiology. A large number of works cover this topic for different kinds of sports (e.g. tennis [5][6], table tennis [7], and cricket [8]).

Since recently some scientists use the virtual environment technology to investigate problems related to the perception-action connection. The Bunraku (INRIA) team studies perception, decision and action of real and virtual humans in virtual environments. In this project, virtual reality can be defined as a set of dedicated hardware and software techniques that allow one or several users to interact in a natural way with numerical data sensed by the way of sensory channels [9]. [10] covers the topic of using virtual reality to analyze sports performance. The virtual environments technology was used for studying the perception-action connection in the handball sport [11], in soccer [12], and in other team ball sports [13]. In these works, virtual reality was used for experiments and training sessions in team sports. We didn't find any analogue works for alpine skiing.

3 Methods

For all experiments we used the next-generation visual-force training complex which is described in [14]; its improved version is described in [15]. The binocular passive projection stereo system was used as the virtual reality visual system with polarizing filters for splitting visual channels. We studied the following factors:

1. Disparity caused by binocular parallax
2. Accommodation
3. Factors hindering visual perception and influencing on skiing course visibility — fog, snow, rain
4. Influence of myopia on perceiving a skiing course

Besides, we noticed that one more factor can possibly influence on dynamical 3D scenes perception. This factor can be called "inverse motion parallax". It is strange that inverse motion parallax is still ignored in developing of 3D visualization equipment and mathematical justification. The training system functioning is illustrated in Figures 1, 2.

Video of real alpine ski course descends and virtual models of these courses were used to solve the problems of synchronization and to improve physical adequacy and virtual 3D scene realism.

The mechanical part of our training system fails to reproduce the real downhill skiing process realistically because the carriage with a ski binding is oriented horizontally. That is why we decided to simplify the dynamical model of motion using the approximate carving turn model. We used the results of P. Federolf [16] for turn trajectory approximation. The motion model simplification is based

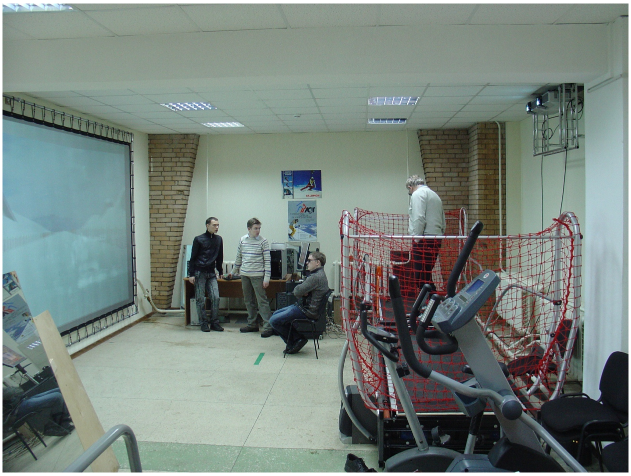


Fig. 1. The training system

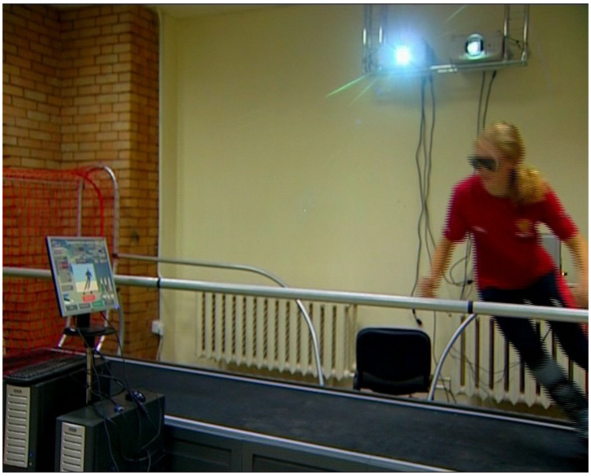


Fig. 2. Professional alpine skier testing the training system

on the fact that the local turn curvature depends first of all on the edging angle $\varphi(t)$ cosine. Additionally it depends on the ski side cut and geometry, as well as on the elastic constant tensor and rigidity of snow. All these dependences were approximately represented by the empirical monotonous function $f(x)$, which is close to a linear dependence. A virtual skier on a skiing course moves synchronically with a real skier-tester. This synchronization is provided through the edging angle $\varphi(t)$ and the right-left sign *turn*. These parameters are received from the mechanical part of the training system. Consequently, the virtual skier descending trajectory is defined as follows.

To calculate the new skier's position, we used the following formula:

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} + dt \cdot \begin{pmatrix} V_x \\ V_y \end{pmatrix}, \tag{1}$$

where $\begin{pmatrix} x' \\ y' \end{pmatrix}$ is the new position, $\begin{pmatrix} x \\ y \end{pmatrix}$ is the previous position, $\begin{pmatrix} V_x \\ V_y \end{pmatrix}$ is the skier's velocity, dt represents a small interval of time.

The calculation (rotation) of the skier's velocity new direction is the following:

$$\begin{pmatrix} V_x' \\ V_y' \end{pmatrix} = \begin{pmatrix} \cos(\alpha \cdot turn) & -\sin(\alpha \cdot turn) \\ \sin(\alpha \cdot turn) & \cos(\alpha \cdot turn) \end{pmatrix} \cdot \begin{pmatrix} V_x \\ V_y \end{pmatrix}, \tag{2}$$

where $\begin{pmatrix} V_x' \\ V_y' \end{pmatrix}$ is the new direction, $\begin{pmatrix} V_x \\ V_y \end{pmatrix}$ is the previous direction, α is the rotation angle, *turn* represents the sign of the right (1) or left (-1) turn.

The rotation angle is calculated as follows:

$$\alpha = K \cdot v \cdot dt, \tag{3}$$

where is the turn arc curvature and v is the velocity module.

The turn arc curvature depends on the skis edging angle (which is received from the training system) and is given by:

$$K = f(\frac{1}{\cos \varphi(t)}), \tag{4}$$

where $f(x)$ is a monotonous function which binds the edging angle $\varphi(t)$ with the turn curvature.

4 Experiment Results

The main goal of our experiments was to investigate how visibility factors influence on skiers reactions in complex rapidly changing virtual scene. At the same time, we tried to evaluate how the skiing testers sensation on a virtual course corresponds to the sensation on a real skiing course in mountains. In the conducted experiments, we estimated the impact of weather effects (fog, snow, rain) on ski course visibility and the point spread function (myopia) on skiers descent time in a virtual environment. It was revealed that accommodation affects visual 3D course perception greatly. Preliminary statistics is represented in the following tables.

4.1 Influence of Fog on Virtual Skiing Descent Time

The following equation calculates the color of fog:

$$C = f \cdot C_p + (1 - f) \cdot C_f, \tag{5}$$

where C_p is the RGBA value of the processed pixel (fragment) color, C_f is the RGBA value of the fog color, and f is calculated by the following rule:

$$f = e^{-d \cdot z}, \tag{6}$$

where e is the Euler number, z is the distance from the point of view to the pixel (fragment), d is the fog density factor. The rendered virtual ski course with fog is illustrated in Figure 3, and the experimental results are represented in Table 1.

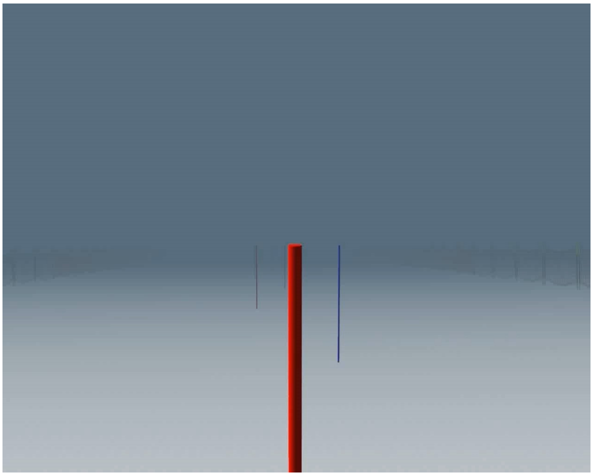


Fig. 3. The virtual ski course with fog density $d = 0.10$

Table 1. Influence of the fog density (d) on the relative time of virtual descent

d	0.01	0.05	0.10	0.15	0.18	>0.2
dT/T %	0.5	0.7	1.1	1.9	>2.5	DC

- T is the descent time without visibility scattering affects
- T+dT is the descent time with a visibility scattering affect (fog, snow, rain)
- dT/T is the ratio (percentage)
- DC stands for disqualification (the skiing tester didn't manage to finish the course)

4.2 Influence of Snow on Virtual Skiing Descent Time

The snow was modeled using the system of virtual particles. Dependences of snow particles' parameters on the intensity parameter I are the following:

1. The particle velocity is $V = -0.75 - 0.25I$
2. The particle size is $s = 0.02 + 0.03I$
3. The particle color is $C = 0.85 - 0.1I$
4. The maximum density is $D = 8.2I$

The experimental results are represented in Table 2, and the virtual ski course example with snow is shown in Figure 4.

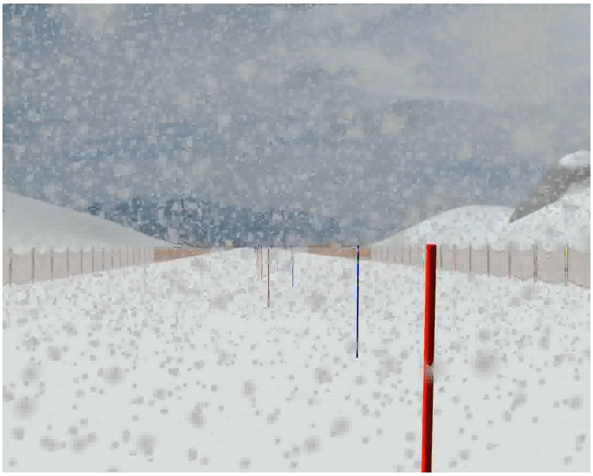


Fig. 4. The virtual ski course with the snow intensity $I = 1.5$

Table 2. Influence of the snow intensity (I) on the relative time of virtual descent

I	1.00	1.50	2.00	2.5	3.00	>3.4
dT/T %	0.6	0.9	1.8	1.9	>2.9	DC

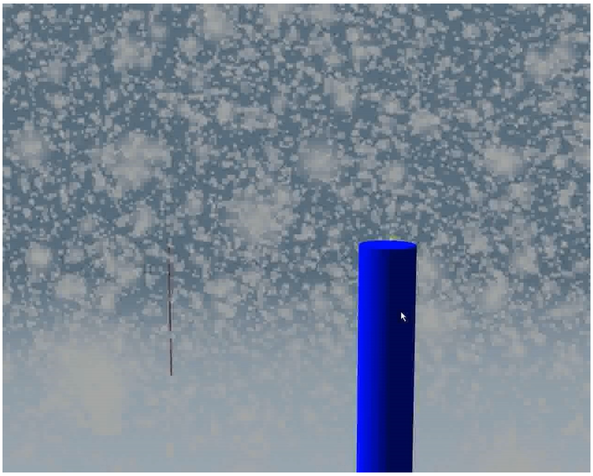


Fig. 5. The virtual course with the snow intensity $I = 2.0$ and the fog density $d = 0.08$

4.3 Influence of Fog and Snow on Virtual Skiing Descent Time

Figure 5 shows the virtual scene with snow and fog. Table 3 illustrates the experimental results with snow and fog.

Table 3. Influence of the fog density (d) and the snow intensity (I) on the relative time of virtual descent

d+I	0.01+1.00	0.08+1.20	0.08+1.5	0.15+>2.00
dT/T %	0.8	1.3	3.2	DC

4.4 Influence of Rain on Virtual Skiing Descent Time (It Is Actual for SOCHI 2014)

Dependence of rain particles' parameters on the intensity parameter $I_{rain}(I)$ are the following:

$V = -2.0 - 5.0I$, $s = 0.01 + 0.02I$, $C_{rain} = 0.6 - 0.1I$, $D = 8.5I$.

Figure 6 presents the virtual scene with rain. Table 4 shows the experimental results of skiing on the training system in the rainy conditions.

Table 4. Influence of the rain intensity (I) on the relative time of virtual descent

I	1.00	1.50	2.00	2.5	>3.00
dT/T %	0.7	1.1	2.1	2.5	DC

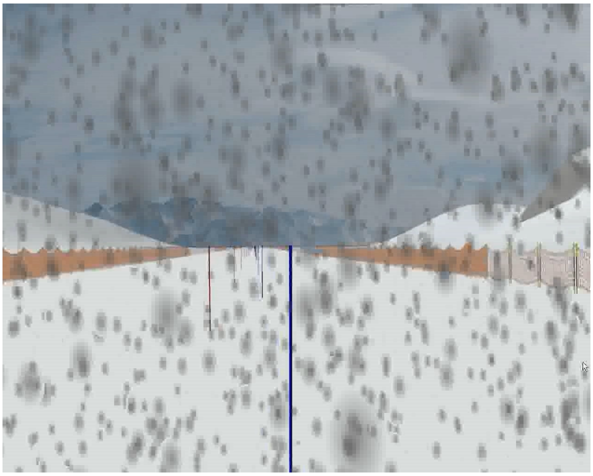


Fig. 6. The virtual ski course with the rain intensity $I = 2.5$

4.5 Influence of Myopia on Virtual Ski Course Perception

The influence of myopia on visual perception [17] can be represented by the well-known convolution equation [18]:

$$i_s(r) = \int i(\xi) \cdot h(r - \xi) d\xi, \tag{7}$$

where r is the two-dimensional coordinate of a pixel, $i(r)$ is the non-aberrated image of the virtual course, $h(r)$ is the point spread function which defines characteristics of an eye with myopia, $i_s(r)$ is the aberrated image of the virtual course, ω is the two-dimensional spatial frequency, $I(\omega)$ is the Fourier transform of the non-aberrated virtual course image, $I_s(\omega)$ is the Fourier transform of the aberrated virtual course image, $H(\omega)$ is the Fourier transform of the point spread function (the transfer function),

$$F\{g(r)\} = \int g(r) \cdot e^{-j2\pi\omega r} dr, \tag{8}$$

$$I(\omega) = F\{i(r)\}, \tag{9}$$

$$I_s(\omega) = F\{i_s(r)\}, \tag{10}$$

$$H(\omega) = F\{h(r)\}, \tag{11}$$

$$I_s(\omega) = I(\omega) \cdot H(\omega), \tag{12}$$

where $F\{g(r)\}$ is the Fourier transform.

Figure 7 shows the non-aberrated virtual course image and the pixels intensity distribution histogram. For this histogram, the mathematical average value m is 176.474, and the standard deviation *sigma* is 44.415.

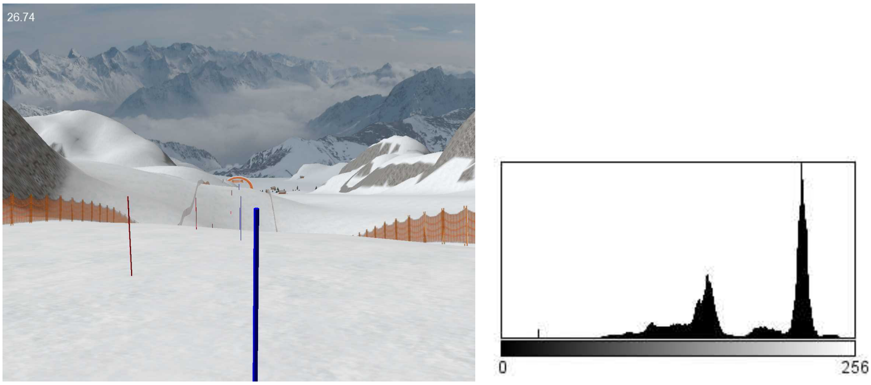


Fig. 7. The non-aberrated virtual course image $i(r)$ and the pixels intensity distribution histogram ($m = 176.474$, $sigma = 44.415$)

Figure 8 illustrates the Fourier spectrum modulus of the non-aberrated virtual course image and the spatial frequency distribution histogram. The histogram parameters are $m = 40.940$ and $sigma = 17.005$.

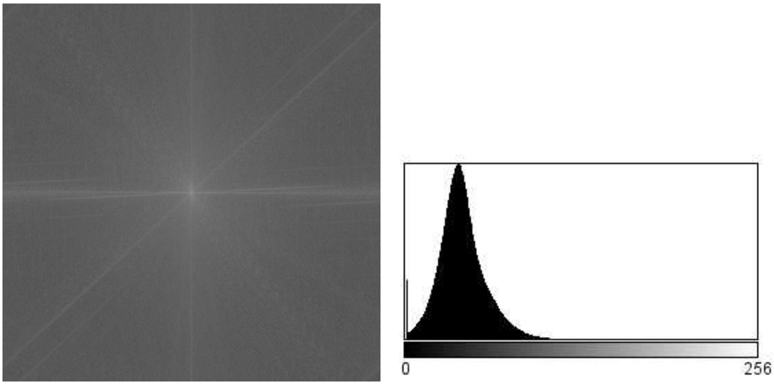


Fig. 8. The Fourier spectrum modulus of the non-aberrated virtual course image $|I(\omega)|$ and the distribution histogram with parameters $m = 40.940$ and $sigma = 17.005$

Figure 9 shows the aberrated virtual course image and the pixels intensity distribution histogram ($m = 176.504$, $sigma = 42.773$). The point spread function effective width is 10 pixels.

Figure 10 illustrates the Fourier spectrum modulus of the aberrated virtual course image and the spatial frequency distribution histogram. The histogram parameters are $m = 8.253$ and $sigma = 13.990$.

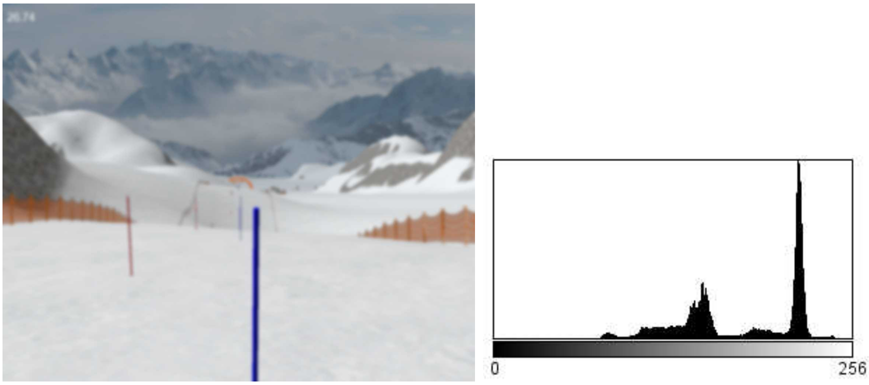


Fig. 9. The aberrated virtual course image $i_s(r)$ and the pixels intensity distribution histogram (m = 176.504, *sigma* = 42.773)

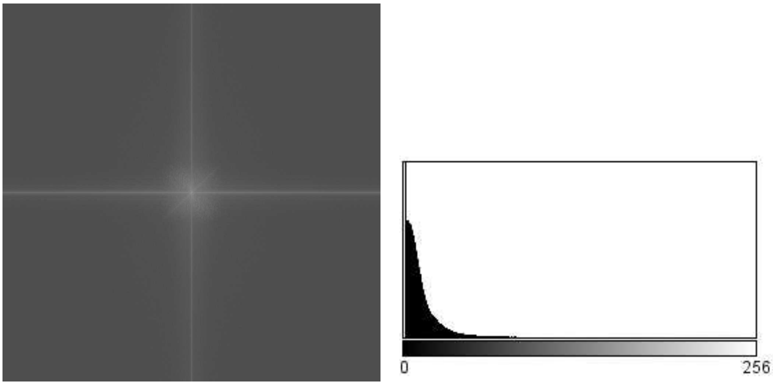


Fig. 10. The Fourier spectrum modulus of the aberrated virtual course image $|I_s(\omega)|$ and the distribution histogram (m = 8.253, *sigma* = 13.990)

If we compare figure 7 and figure 9, we will see that the intensity distribution histograms of the non-aberrated and aberrated images are quite similar (m = 176.474, *sigma* = 44.415 and m = 176.504, *sigma* = 42.773 respectively). At the same time, figures 8 and 10 show that the spatial frequency distribution histograms differ considerably (m = 40.940, *sigma* = 17.005 for the non-aberrated image, and m = 8.253, *sigma* = 13.990 for the aberrated image). Myopia leads to the loss of high spatial frequencies in the image. The conducted experiments showed that myopia affects perception of a rapidly changing surrounding environment greatly. The skiing tester wasnt able to finish the virtual course.

It means that clear boundaries of objects and, consequently, high spatial frequencies play a very important role in the 3D perception of a rapidly changing virtual scene.

During the training system testing, it became clear that the skier perceives turning poles distances incorrectly. The visual ability to perceive the world in three dimensions and distance of objects is provided by depth perception cues. Depth perception cues are described in [19]. Our visual stereoscopic system allows reproducing the following depth cues which are used in two-dimensional images: **perspective, texture gradient, details, occlusions, lighting and shading, motion parallax**.

There are other depth cues that are not enabled in 2D images:

- Binocular disparity. Binocular viewing of a scene creates two slightly different images of the scene in the two eyes due to the eyes' different positions on the head. These differences, referred to as binocular disparity, provide information that the brain can use to calculate depth in the visual scene. Besides disparity (which is considered to be the main factor of stereoscopic vision) there are two more very important depth cues:
- Vergence (the simultaneous inward movement of both eyes toward each other, usually in an effort to maintain single binocular vision when viewing an object).
- Accommodation (the change of the eye lens optical power to maintain a clear image on an object as its distance changes).

4.6 Joint Role of Disparity, Vergence and Accomodation

Three factors (disparity, vergence and accommodation) form the ”main triad” of stereoscopic vision. In fact, there are many other depth cues. We should note one very powerful depth cue. This factor is the reflex connection of accommodation and vergence. This connection has been known for a long time. However, in recent years much more attention is paid to the conflict of accommodation and vergence in induced 3D virtual environments [20]. This conflict becomes apparent in the near zone visualization (3-5 meters around the observer).

Developers of present virtual reality interface devices try to avoid or neutralize the connection of accommodation and vergence in order to avoid the perception conflict. For this purpose, the LEEP-optics is often used. This optical system moves 3D virtual scene objects back from the observer. The drawback of this system is that it distorts the visual perception constancy and the 3D-environment kinesthetic image.

Full reproduction of all depth perception cues of 3D space is possible in holographic displays only by recording the wave front from an object and then reproducing it. Nowadays, holographic displays are currently under development and far from being complete. However, it is possible to reproduce wave front parameters which provide adequate visual perception of 3D objects with the help of adaptive optics [21][22].

We intend to develop a “two channels” variant of the system which will combine binocular perception and accommodation (other variants are possible). The first channel will reproduce objects in the nearest to the observer zone (the nearest turning pole) with the help of the active spherical mirror (OKOTECH) in accordance with this object distance in the virtual scene. The second channel will reproduce the virtual 3D scene without the nearest objects.

While performing a descent on a high speed, the skier focuses on the nearest turning pole. Considering this fact, we developed geometrical parameters for the optic-mechanical experimental setup for studying the adaptive optics system functioning and its operation when the accommodation plane changes its depth. A preliminary scheme of the experimental setup is shown in Figure 11. At the present time, our team develops this experimental setup using all our experience in virtual environments and astronomic adaptive optics.

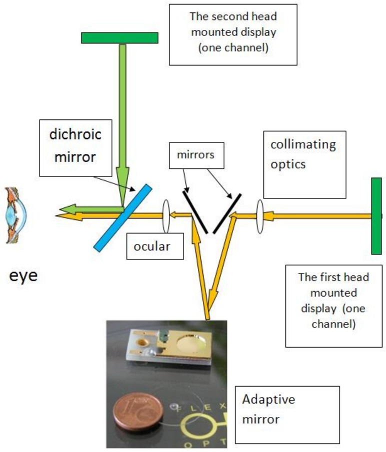

The second head mounted display (one channel)
dichroic mirror
mirrors
collimating optics
ocular
eye
The first head mounted display (one channel)
Adaptive mirror

Fig. 11. The experimental setup for testing the adaptive optics operation when observing virtual objects at different distances

Let us talk briefly about the optical phenomenon which we call inverse motion parallax. The motion parallax phenomenon has been known for about 3 centuries, but it still attracts a lot of attention [23][24][25]. The widely known geometrical model of motion parallax is greatly simplified and is often considered in the conditions of fixed optic axes of visual sensors (in the special case, the axes are parallel). In the most common model, the observer’s eyes are focused at infinity. With this condition, when the observer moves relative to the 3D space, he has a visual sensation that the space rotates around a point at infinity (vanishing point).

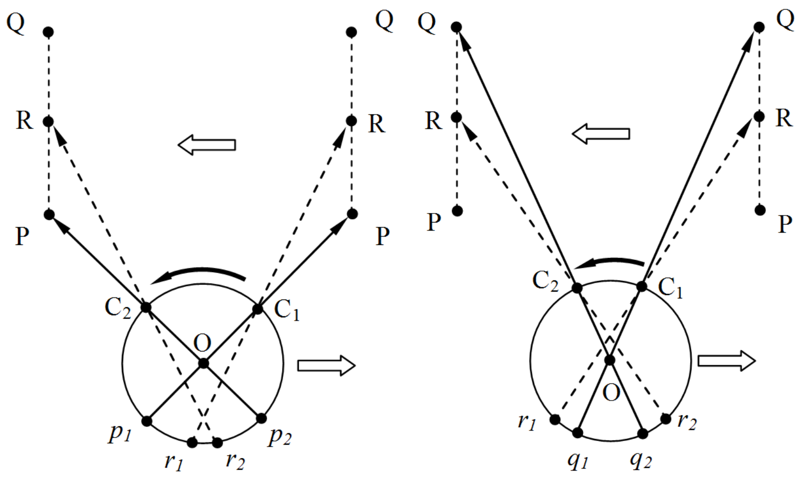


Fig. 12. Geometrical scheme of motion inverse parallax (a simplified model of an eyeball horizontal cut is used). Left — tracking point P, right — tracking point Q.

At the same time, the geometrical projection scheme analysis shows that the image on the retina changes its behavior significantly when the point on which the observer's eyes are focused changes its depth as shown in Figure 12. This figure contains a horizontal cut of the spherical camera obscura with different optic axes directions which correspond to different target points. Points C_1 and C_2 show different positions of the camera obscura projection center which corresponds to different optic axis's positions. Points P, Q and R (P and Q are the nearest and furthest target points correspondingly, R is the reference point) are situated along a straight line which is orthogonal to the line of camera obscura motion from left to right. The camera rotates counterclockwise around its rotation center O and targets at point P first, then at point Q (as it shown on fragments a and b correspondingly). Points r_1 and r_2 represent the reference point projections' positions, while points p_1, p_2 and q_1, q_2 represent positions of the target points projections. The main conclusion is that the direction of the reference point R projection movement depends on the depth of the camera obscure target point.

You can see this phenomenon more clearly in Fig. 13 and Fig. 14 (consider each vertical row separately). These figures contain the 3D scene screenshots taken while a more habitual cubical camera obscura moved. In the first figure, the camera was targeted at the object in the foreground (a blue pole in the left vertical row of figures), and on the second figure at the object in the background (a cyan pole in the right vertical row of figures). Note that the reference point (a red pole) moves in opposite directions in Fig. 13 and Fig. 14.

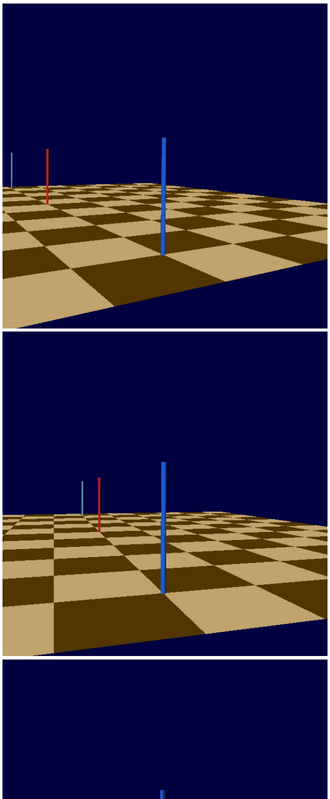


Fig. 13. The camera is targeted on the nearest object (blue pole). The projection of the reference object (red pole) moves from left to right.

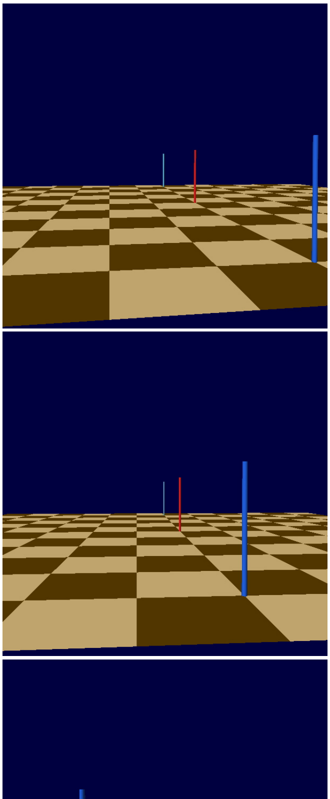


Fig. 14. The camera is targeted on the furthest object (cyan pole). The projection of the reference object (red pole) moves from right to left.

5 Discussion

The role of the accommodation depth cue in 3D environment depth perception is underestimated. To investigate the accommodation role, we propose a optic system which visualize the nearest turning pole. The proposed system modulates the wave front with the help of a spherical mirror or a flexible liquid-crystal adaptive lens to bring into accord the power of accommodation and the target object depth.

6 Conclusion

The main results of the work:

1. The designed visual-force training complex allows to study 3D visual perception of the skiing course inside a laboratory.
2. We conducted several experiments to estimate the influence of weather effects (fog, snow, rain) on skiing descent time.
3. We studied the impact of myopia on the virtual skiing course perception. High spatial frequencies in an image play a very important role in the adequate perception of complex rapidly changing virtual scenes.
4. The role of accommodation in 3D visualization of skiing courses was defined.
5. We proposed the new visualization system on the base of adaptive optics to resolve the problem of vision accommodation.
6. We investigated conditions of the inverse motion parallax emergence.

In further studies, we intend to investigate the joint action of disparity, vergence and accommodation in rapidly changing 3D scenes. In particular, it is interesting to explore the hierarchical relationship between these three factors in extreme high-speed conditions. It is also interesting to estimate the impact of the accommodation ability of a skier on his athletic performance (including the role of myopia and hyperopia).

Acknowledgments. We express our thanks to Professor A. Sourin, Professor M. Gavrilova and postgraduate student M. Borodina for their assistance. We express our thanks to Russian Foundation of Basic Research for funding our project (grants 08-07-00399 and 10-07-00513).

References

1. Marr, D.: Vision: a computational investigation into the human representation and processing of visual information, pp. 1–385. Freeman, San Francisco (1982)
2. Rauschenbach, B.V.: Perspective systems in art. The general theory of perspective, Moscow, Nauka, pp. 1–256 (1986) (in Russian)
3. Sekuler, R., Blake, R.: Perception. McGraw-Hill, New York (2001)

4. Williams, A.M., Davids, K., Williams, J.G.: Visual perception and action in sport, pp. 1–468. E & FN Spon, London (1999)
5. Williams, A.M., Ward, P., Smeeton, N.J., Allen, D.: Developing Anticipation Skills in Tennis Using On-Court Instruction: Perception versus Perception and Action. Journal of Applied Sport Psychology 16, 350–360 (2004)
6. Shim, J., Carlton, G., Chow, J.W., Chae, W.S.: The Use of Anticipatory Visual Cues by Highly Skilled Tennis Players. Journal of Motor Behavior 37(2), 164–175 (2005)
7. Bianchi, B., Domini, F., Agostini, T., et al.: Service and return in table tennis: Visual cues and selection of the right information. In: 30th European Conference of Visual Perception, August 27-31 (2007)
8. Davids, K., Renshaw, I., Glazier, P.: Movement models from sports reveal fundamental insights into coordination processes. Exerc. Sport Sci. Rev. 33(1), 36–42 (2005)
9. Team Bunraku: Perception, decision and action of real and virtual humans in virtual environments and impact on real environments, INRIA, Activity report (2009)
10. Bideau, B., Kulpa, R., Vignais, N., Brault, S., Multon, F., Craig, C.M.: Virtual reality, a serious game for understanding performance and training players in sport. IEEE Computer Graphic Applications 30, 14–21 (2010)
11. Vignais, N., Bideau, B., Craig, C., Brault, S., Multon, F., Kulpa, R.: Virtual environments for sport analysis: perception-action coupling in handball goalkeeping. The International Journal of Virtual Reality 8(4), 43–48 (2009)
12. Craig, C.M., Bastin, J., Montagne, G.: Immersive Interactive Virtual Reality How information guides movement: Intercepting curved free kicks in soccer. Human Movement Science 30(5), 931–941 (2011)
13. Frank, T., Zaal, J.M., Bootsma, R.J.: Virtual Reality as a Tool for the Study of Perception-Action: The Case of Running to Catch Fly Balls. Presence 20(1), 93–103 (2011)
14. Aleshin, V., Klimenko, S., Manuilov, M., Melnikov, L.: Alpine skiing and snowboarding training system using induced virtual environment. In: Proc. of the 4th ICSS 2007, pp. 137–144 (2007)
15. Aleshin, V., Klimenko, S., Astakhov, J., Bobkov, A., Borodina, M., Volegov, D., Kazansky, I., Novgorodtsev, D., Frolov, P.: 3D scenes simulation, animation, and synchronization in training systems with force back-coupling. In: Proc. of the 19th International Conference GraphiCon 2009, Moscow, Russia, pp. 166–170 (2009)
16. Federolf, P.A.: Finite element simulation of a carving snow ski. PHD Thesis (2005)
17. Gross, H. (ed.): Handbook of Optical Systems: vol. 4 Survey of Optical Instruments (36, Human Eye). WILEY-VCH Verlag GmbH and Co. KGaA, Weinheim (2008)
18. Goodman, J.W.: Introduction to Fourier Optics. McGraw-Hill (1968)
19. Reichelt, S., Haussler, R., Futterer, G., Leister, N.: Depth cues in human visual perception and their realization in 3D displays. In: Proc. SPIE 7690 (2010)
20. Hoffman, D.M., Girshick, A.R., Akeley, K., Banks, M.S.: Vergence–accommodation conflicts hinder visual performance and cause visual fatigue. Journal of Vision 8(3), 1–30 (2008)
21. Schowengerdt, B.T., Seibel, E.J.: True 3D scanned voxel displays using single or multiple light sources. Journal of the SID 14/2, 135–142 (2006)
22. Schowengerdt, B.T., Seibel, E.J., Kellya, J.P., Silvermana, N.L., Furness, T.A.: Binocular retinal scanning laser display with integrated focus cues for ocular accommodation. In: 15th Annual Symposium on IS&T/SPIE's (2003)

23. Watt, S.J., Bradshaw, M.F.: The visual control of reaching and grasping: Binocular disparity and motion parallax. Journal of Experimental Psychology: Human Perception and Performance 29(2), 404–415 (2003)
24. Ono, H.: Perception of Depth, Motion, and Stability with Motion Parallax. In: Second International Symposium on Universal Communication, ISUC 2008, December 15-16, pp. 193–198 (2008)
25. Nawrot, M., Ratzlaff, M., Leonard, Z., Stroyan, K.: Modeling perceived depth from motion parallax with the motion/pursuit ratio. Journal of Vision 11(11), 705 (2011)

Continuous Control of Style and Style Transitions through Linear Interpolation in Hidden Markov Model Based Walk Synthesis

Joëlle Tilmanne and Thierry Dutoit

Numediart Institute / TCTS Lab.
University of Mons (UMons)
Mons, Belgium
joelle.tilmanne@umons.ac.be

Abstract. We present a Hidden Markov Model (HMM) based stylistic walk synthesizer, where the synthesized styles are combinations or exaggerations of the walk styles present in the training database. Our synthesizer is also capable of generating walk sequences with controlled style transitions. In a first stage, Hidden Markov Models of eleven different gait styles are trained, using a database of motion capture walk sequences. In a second stage, the probability density functions inside the stylistic models are interpolated or extrapolated in order to synthesize walks with styles or style intensities that were not present in the training database. A continuous model of the style parameter space is thus constructed around the eleven original walk styles. Qualitative user evaluation of the synthesized sequences showed that the naturalness of motions is preserved after linear interpolation between styles and that evaluators are sensitive to the interpolation factor.

Keywords: HMM, gait, style, control, synthesis, motion capture.

1 Introduction

The character animation field covers a lot of different domains, ranging from the coarse motions seen in video games to the precise humanlike motions of the last generation of 3D films, and including fields like virtual reality or character animation for human-computer interactions. In the framework of character animation, several approaches are available to produce realistic humanlike motion. There are currently three main kinds of techniques which are used for motion production in the animation field: traditional keyframe animation, procedural techniques in which a software based on a set of rules helps the animator for motion production, and motion capture based approaches where the animator uses real motions captured on an actor.

Producing natural looking animations of virtual humans is a very challenging task as human eyes are natural experts of human motion and naturalness. This is why motion capture, which consists in capturing human motion in the real world to transfer it to the virtual world under a mathematical form usable by

M.L. Gavrilova and C.J.K. Tan (Eds.): Trans. on Comput. Sci. XVI, LNCS 7380, pp. 34–54, 2012.

computers, is the only way to obtain truly realistic human motion [21]. Motion capture opens a huge field of study and of potential applications, and has received growing interest in the last years, especially since it is becoming more affordable and reliable. However, this technology suffers from several drawbacks. Motion capture data have a high dimensionality, the choice of the parameterization is not straightforward and motion is highly variable in general. Furthermore, motion capture is not very flexible. It is hard to reuse recorded motion capture segments and it is also very difficult to modify natural motion without loosing their naturalness, especially since the human eyes will be very sensitive to any inconsistency in motions. In this article, we address this last problem, by interpolating and extrapolating between and beyond the motion styles present in our motion capture training database. Finally, natural looking transitions are another challenge of motion capture sequences. For instance, generating a smooth transition between two separate walk styles is not straightforward if the transition has not been recorded. This issue is also addressed in this article, as our model is capable of handling controllable and natural looking transitions between distinct style models.

Two different approaches coexist about using motion capture data for producing animations: the "template-based" and the "model-based" approaches. In the "template-based" approach, a large database of motion sequences is built and algorithms are developed to retrieve, edit and blend together motion parts, to produce new sequences [8]. The "model-based" approach consists in training models based on motion capture data. The models can later be used to synthesize new motion sequences without resorting to the database initially used for training [9,1,31]. If the model is properly trained, the information contained in the database is summarized in the model parameters. This model gives then more freedom to the user for producing new plausible sequences, even if they were not present in the training database. This approach has been used for years in speech processing for example, first for recognition and more recently for synthesis [14]. Our work falls in the latter category, with the use of model-based techniques and more precisely of Hidden Markov Models (HMMs) [25], for the modeling and synthesis of humanlike motion. HMMs have been used for motion modeling and more especially motion recognition for a long time. The statistical nature of HMMs makes them perfect candidates for the modeling of spatio-temporal time series like human motion where both the tempo and the space trajectory can vary for several realizations of the same motion.

This approach is inscribed at the crossing between motion capture and procedural methods, as we use statistical learning techniques to automatically extract the underlying rules of human motion, without any prior knowledge, directly from training on 3D motion data. The position of our current work in the character animation field is illustrated in Figure 1.

In addition to template- or model-based approaches, other distinctions can be made between the methods applied to modify motion capture data. Parametric motion synthesis is a method that can be applied to both approaches and which consists in producing new motion by interpolating between motions that are

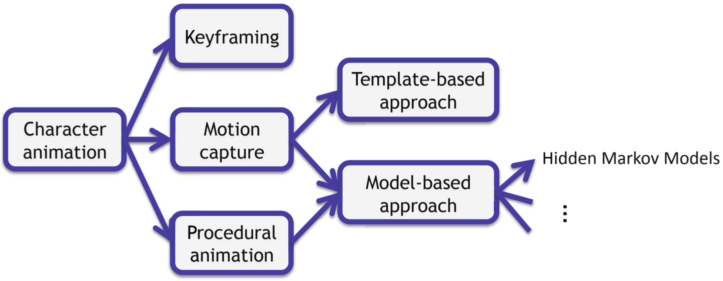


Fig. 1. Position of the HMM-based motion synthesis in the character animation hierarchy

visually similar and correspond to the same logical action [24]. In this article, our approach is parametric and model-based as we aim not only to synthesize a plausible human walk but also to take into account some kind of "style" component. "Style" can refer to emotions, but also to speed, gender (male or female), age (children or elderly,...), specific contexts, etc. The "style" studied in this paper consists of both emotions (like sadness for instance) or particular contexts (like "topmodel" walk). In this paper, style-specific models of walks are first trained over a training motion capture database. The parameters of these style-specific models are then used to produce new models corresponding to style combinations or style exaggerations not present in the training database. Furthermore, the parameters of the style-specific models can be used to build style-transition models and generate a sequence of walk presenting a smooth style transition, without jumps nor abrupt and disturbing changes.

The remainder of the paper is organized as follows. Section 2 presents related work. The motion capture training databases are presented in Section 3. Section 4 describes how the style-specific HMM models of walk were trained, and how these models can be used to synthesize new walk sequences. Section 5 explains how new style characteristics can be obtained by interpolation and extrapolation of the style-specific walk models. The results are presented in Section 6, and Section 7 describes the qualitative user evaluation. Conclusion and future works are addressed in Section 8.

2 Related Work

One of the main problems associated with the use of motion capture data to build animations is that using only the recorded motions can be very limitative. Even if the sequence of motion wanted by the animator can be found in the database, this sequence might not present the required style. Recording a database with all the style options for all the motions is impossible, and recording new motions each time a new animation has to be produced is very costly and often not materially possible. The goal is to find ways of parameterizing the style component of the motion independently from the functional part of the same motion, in order to give the user some kind of interactive control on the style of the output motion.

This approach is similar to approaches encountered in other aspects of human biometrical characteristics modeling or synthesis. Wecker et al. [42] for instance decompose iris images into basic iris properties and individual biometric characteristics, and use this decomposition in order to synthesize new plausible irises images not present in the training database. Lu et al. [20] start from a generic 3D face model and project 2D images of real faces on it. From one single image of each person, using their generic 3D model, their method generates new 2D images of the same person with different expressions or lighting conditions.

People have taken several approaches to address this problem of decomposition into a functional basis and variable characteristics. We will focus the present review of related work on research addressing 3D motion problems.

One approach is to use signal processing techniques to tackle the problem. Bruderlin and Williams [2] use a multiresolution filtering that decomposes the motion into several frequency bands whose amplitude can be modified in order to change the motion style. Unuma et al. [37] apply Fourier transform on the motion data and can modify the aspect of the motion by changing the weights of the Fourier transform in the frequency domain. Unfortunately, these approaches are not easy to use for style control, as changing the weights in the frequency domain is not an intuitive way of controlling the style of a motion.

The underlying principle of most works, including ours, is to use statistical learning techniques on a set of stylistic motion capture data. These techniques separate automatically, without any prior knowledge, the style component from the fundamental function of the motion. The statistical models can then give the user some kind of control on the style parameter, and synthesize motion according to the user's commands. If the principle is the same, several statistical approaches have been tested in the last years.

In [26], Rose et al. decompose motion into verb (fundamental of motion, like "walking") and adverb (which modifies the basic motion according to emotion, gender, "uphill" or "downhill", etc.). They propose a technique for real-time interpolation of motion sequences, based on radial basis functions and low order polynomials representation of the motion. Glardon et al. [9], Troje [36], Min et al. [22] and Tilmanne et al. [31] all use principal component analysis (PCA) not only for reducing the dimensionality of the problem, but to separate the influence of style from the functional part of the motion. In a similar way, Shapiro et al. [27] use Independent Component Analysis (ICA) for the same purpose. Min et al. [23] conduct a multilinear motion analysis to extract separately style and individuality variations, after time warping and PCA for dimensionality reduction.

Another interesting approach is to use Hidden Markov Models (HMMs) to synthesize motion, and to integrate a style variable into the HMM parameterization. One of the advantages of using Hidden Markov Models is that they exempt from using time warping procedures, needed in most approaches in order to align sequences prior to analyze them or extract the style component among them. HMMs integrate directly in their modeling the time variability of the motion. In their work, Wang et al. [41,40] present an HMM that can be trained as a parametric HMM incorporating a "style" parameter in the probability density functions (these densities are represented by SOMN (self organizing mixture networks) in [41] and by mix-SDTG (stylized decomposable triangulated graphs) in [40]). Brand and Hertzmann [1] include a style variable which is automatically extracted during the training process of the HMM and that can vary during the synthesis of a motion sequence. However, in their "Style Machine", the style variable is not explicit, and changes some intrinsic style-related parameters which can make it hard to use as a style controller. In a work closely related to ours, Yamazaki et al. [46] model walk using a Hidden Semi-Markov Model (HSMM). Their model takes into account speed and stride length as a "style" variation using multiple regression. However, this method can only be use to model quantitative variations, and is thus not suited to model emotions or expressivity that can hardly be described by numerical values. In their approach, the whole training has to be done again if one wants to add a new style in the model.

3 Databases

In this work, two databases recorded with an inertial motion tracking system (IGS-190 from Animazoo [15]) were used. Our two databases, respectively called "eNTERFACE'08 3D" and "Mockey", were recorded with the same motion capture suit but with different aims, subjects and settings.

The eNTERFACE'08 3D database is described in details in [33]. This database contains, among others, three sequences of straight "free" walk for 41 different subjects. In the "free" walk, subjects were invited to walk at their usual comfort speed. In the present work, the three free walk sequences of the 41 subjects were used to train our "neutral" walk model. In that database, the motion was captured at a frame rate of 60 frames per second (fps).

The Mockey Database aims to study the "expressivity" of walk [31]. All the different walks were performed by the same actor. He was given instructions about the "walking style" he had to act before each walk sequence recording. The eleven different acted styles were the following: proud, decided, sad, top-model, drunk, cool, afraid, tiptoeing, heavy, in a hurry and manly. Our "style" component consists thus in exaggerated variations that can be far from a plain walk. In this second database, motion was recorded at a frame rate of 30 fps. These eleven styles were arbitrarily chosen as they all have a recognizable influence on walk, as illustrated in Figure 2.

The walk sequences were manually segmented into left and right steps. The boundaries of the steps were arbitrarily defined as the moment the heel touches the ground. Depending on the style of walk performed and its corresponding step length, a different number of walk steps was recorded for each style. Each motion file contains two parts: the skeleton definition and the motion data. In the motion data part, the first three values of each frame give the 3D position of the

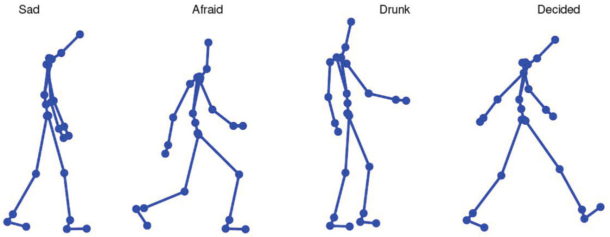


Fig. 2. Four example postures taken from the motion capture database (sad, afraid, drunk and decided walks)

root of the skeleton. They were discarded, as they depend on the displacement and orientation of the walk and can be recalculated given the foot contact with the ground and the leg segments lengths. The pose of the skeleton at each frame is then described by 18 tridimensional joint angles, which gives 54 values per frame to describe the motion.

No dimensionality reduction was conducted on the original set of 54 parameters in order to build a generic style interpolation procedure in which any new walk style could be added in the future without having anything to modify in the previously trained models. This would not have been the case if principal component analysis (PCA) had been used for dimensionality reduction for instance, since the PCA space would depend on the styles present in the training database. By keeping the 54 original motion observations, we avoid our modeling procedure from being dependent on the set of styles present in the training database.

In this paper, our 3D angles were converted from their original Euler angle representation into the exponential map angle parameterization which is locally linear and where singularities can be avoided [11,16]. Exponential maps represent each 3D rotation by 3 values.

4 Style Models Training

Our approach for stylistic model training is to start from a procedure originally developed for speaker adaptation in speech synthesis and to adapt it to our motion problem. This HMM-based procedure is presented in more details in [32] and is based on functions originally implemented for speech within the "HMM-based Speech Synthesis System" (HTS) framework, publicly available on the HTS website [14]. The dynamical aspect of the data is taken into account by integrating the first and second derivatives of our parameters both for reference and stylistic model training and for synthesis [35]. By adding these derivatives to our 54 original parameters, we obtain a 162 dimensional vector of observations to model. The time spent in each state of the HMM is explicitly modeled in duration probability density functions thanks to Hidden Semi-Markov Models

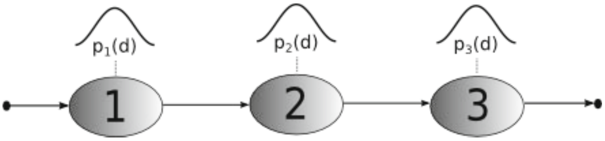


Fig. 3. A three-states HSMM with no skips (with $p_i(d)$ representing the density probability of the duration d of state i)

(HSMM) [47]. The schematic representation of the state duration modeling in a three-states HSMM is represented in Figure 3.

The first stage of the procedure consists in the training of a multi-walkers "reference" model, with the "neutral" walk sequences from the 41 subjects of the "eNTERFACE'08 3D" database. That reference HSMM model is trained using the HTS framework, with a procedure called SAT (for "speaker adaptive training" as it was designed for speech processing [44]) that removes the influence of the subject-specific variations in the final model.

Both steps (left or right) are modeled by separate left-right five-states HSMMs with no skips. Furthermore, contextual factors related to the position of the step in the whole walk sequence were taken into account during the training, thereby multiplying the number of models to train. We made the contextual distinction between five positions in the walk sequence for each step: the first, second, last, last-but-one steps of the sequence, and all the other steps. The training began thus with ten models to train (five for each step). These ten models can be clustered using decision trees. The decision tree is a binary tree, and in each of its nodes, a question splits contextual models into two groups. All possible contextual combinations can be found by traversing the tree. Once the decision tree is constructed, unseen contexts can be prepared and leave nodes containing little or very similar data can be merged (for more information on how the trees are built and used, please refer to [48]). After the decision tree clustering, six out of the ten originally possible contextual HSMMs remained to model the reference walk.

The reference model is used in the second stage of the training, as a basis from which the adaptive training is conducted in order to adapt the reference model to a specific style, using the stylistic walks from the Mockey database. The adaptive training is a method from the latest results of the HMM speech synthesis field, adapted to our motion synthesis problem [32]. Using this adaptation procedure for each style of the Mockey database, we obtain eleven style-specific HSMM models, with each stylistic walk model containing 6 contextual HSMMs.

Using these models, an HMM-based walk synthesis can be conducted and new sequences of walk can be calculated for each one of our eleven walk styles. First a model of the new sequence is obtained by concatenating HMMs corresponding to the succession of steps to be synthesized. This complete model is then used to calculate the corresponding optimal observation sequence, taking into account the dynamics of the synthesized data (thanks to the first and second derivatives of the parameters), which ensures the continuity of the synthesized sequence.

The model gives us joint angles and the cartesian coordinates of the root of the skeleton can then be computed. Using our knowledge of the boundaries of the synthesized walk cycles and calculating the height of each foot thanks to the known leg segment lengths, we determine which foot is in contact with the ground. From that fixed 3D position, we calculate the position of the whole body until the other foot becomes the reference, and so on for the whole sequence. This method enables us to ensure that no foot sliding effect can occur, as the displacement of the whole body is driven by the foot contact point with the ground.

5 Interpolation

Once our eleven style-dependent HSMM models are built, we can synthesize as many stylistic walk sequences as we want, for each one of the eleven styles. So far, each style is modeled separately, and in the synthesis step, the user's control on the output sequence is only the choice of one among the eleven possible styles. In this work, we want to go further in the style modeling and study how the styles can be controlled with more freedom and how the models behave outside of the styles trained from the database. In the model training stage presented in Section 4, each stylistic walk is modeled by six five-states HSMMs, and each HSMM model contains both state duration and 162-dimensional observation modeling. For each state of each HSMM, duration is modeled by one Gaussian probability density function (mean and variance) and observations are modeled by single Gaussian probability density functions (multidimensional Gaussian with diagonal covariance matrix). The set of parameters of one whole style model consists thus in 9780 parameters for 4890 probability density functions (pdfs), as each pdf consists in a Gaussian model (mean + variance) and $(\ 162\ (observation\ parameters)\ +\ 1\ (duration\ of\ one\ state)\ )$
$*\ 5\ (number\ of\ states)\ *\ 6\ (number\ of\ HSMMs) = 4890\ pdfs.$

Among these 4890 probability density functions, 4860 pdfs model the observations and 30 pdfs model the state durations.

In this work, we use these model parameters as a basis for obtaining new or exaggerated styles by extrapolation or interpolation. To do so, a simple procedure was applied. New models were obtained by linear combinations of the means of the probability density functions of each style-specific model. The 4890-dimensional model pdfs space is thus considered as a continuous space in which new styles can be produced by taking points in the space between two existing styles or slightly beyond these styles. Since the new style is obtained through a simple linear interpolation between two vectors, the high dimensionality of the model parameters is not an issue. The known styles are used as landmarks in the continuous model parameter space, around which new style possibilities can be produced by going further away or coming closer to another known style.

In this continuous space, combinations of styles can be obtained, but the intensity of the eleven styles can also be controlled in a continuous manner, even though no control was available in the eleven style models themselves. An average walk model was calculated by taking the mean model parameters over the eleven style models, and used as a basis for controlling the intensity of each style separately. This average model corresponds to some kind of "neutral" style for the actor recorded in the stylistic database. Figure 4 illustrates the linear operations in the 4890-dimensional model space. In our previous work [32], eleven separate style models were obtained and could be used separately (Figure 4.A). In this work, we generalize the motion space by interpolating and extrapolating between two styles (Figure 4.B) or by continuously controlling the amplitude of a given style by interpolation and extrapolation with respect to the average style (mean of our eleven styles, (Figure 4.C)).

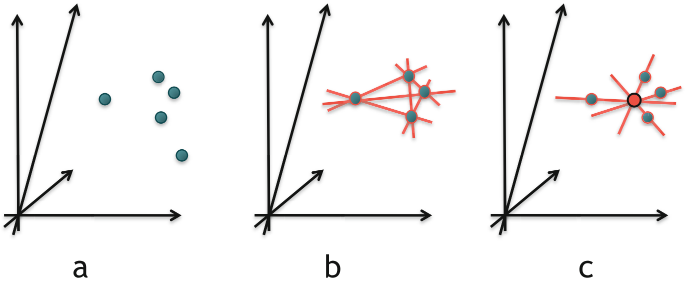


Fig. 4. Schematic representation of the interpolation/extrapolation procedure in the 4890-dimensional model space. Green dots represent the separate stylistic models, red lines the interpolation between models and the red dot is the averaged style model.

6 Results

Our first approach in building new models consists in linearly interpolating the whole set of 9780 parameters between two styles:

$$pdf_{new} = pdf_{style1} + interp * (pdf_{style2} - pdf_{style1}). \tag{1}$$

Where *pdf* is the 4890-dimensional vector of the probability density function means (i.e. the parameters of the model), and *interp* is a scalar value that gives the interpolation ratio between *style* 1 and *style* 2. The difference between *style* 2 and the reference *style* 1 is thus multiplied by a factor *interp* before being added to *style* 1. Figure 5 shows an example of interpolation between the original sad and decided walks, and illustrates that both postures and durations are interpolated. This approach can easily be generalized to a linear combination of any number of original styles.

The second approach aims at controlling the intensity of expression for each style separately. Style interpolation and exaggeration were obtained by using the averaged walk model as a reference (*style* 1 in equation 1), and the style whose

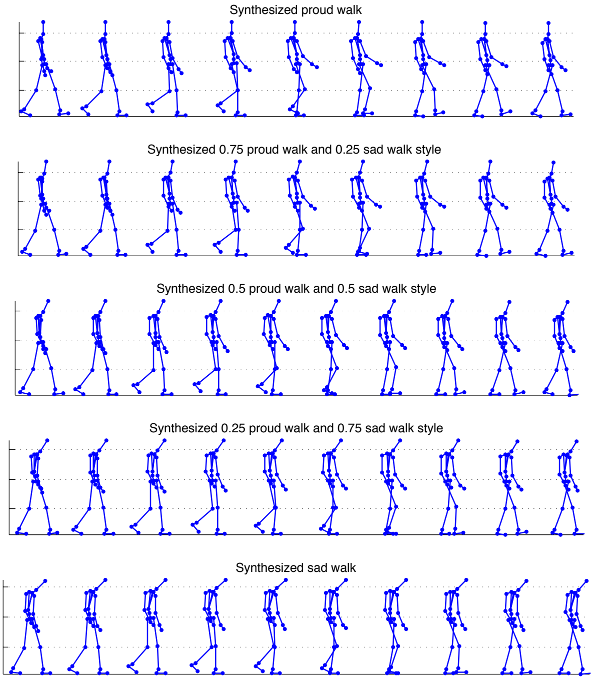


Fig. 5. Synthesized left step for original proud walk (first subfigure, *interp* = 1), 0.75 proud and 0.25 sad walk (second subfigure, *interp* = 0.75), half sad/half proud walk (third subfigure, *interp* = 0.5), 0.25 proud and 0.75 sad walk (fourth subfigure, *interp* = 0.25), original sad walk (last subfigure, *interp* = 0), . Synthesized skeleton poses are displayed every 0.1 second.

intensity we want to control (*style* 2) as an interpolation direction. The strength of the style can be diminished by decreasing the *interp* value from 1 to 0. For values of *interp* greater than 1, the style is exaggerated. Our synthesized walk sequences remained natural for values of *interp* lower than 2. For values around 2 and above, the quality of the synthesized walk depended on the original style as some exaggerated styles were affected by impossible movements (knees bending backwards or awkward bending of the spine for instance). These problems could be avoided in future studies by adding constraints to the joint angles so that they cannot take values beyond what is physically possible for a human being.

Another interesting result is obtained by giving negative values to the *interp* parameter. The difference between the controlled style and the average walk model is substracted to that average walk instead of being added. The obtained style then presents characteristics at the opposite of the controlled style. For instance the sad walk which is slower than the average model and where the pose of the character tends to bend inwards with respect to the average posture will give an opposite model where the character walks faster and looks much more “open” by its posture. We are thus able to synthesize styles that do not appear in our database but that show style characteristics that are opposite to the ones of recorded styles. A left step synthesized with the original sad walk model and with three new style intensities obtained by interpolation or extrapolation is illustrated in Figure 6. The figure shows that both poses and durations are affected by the interpolation process.

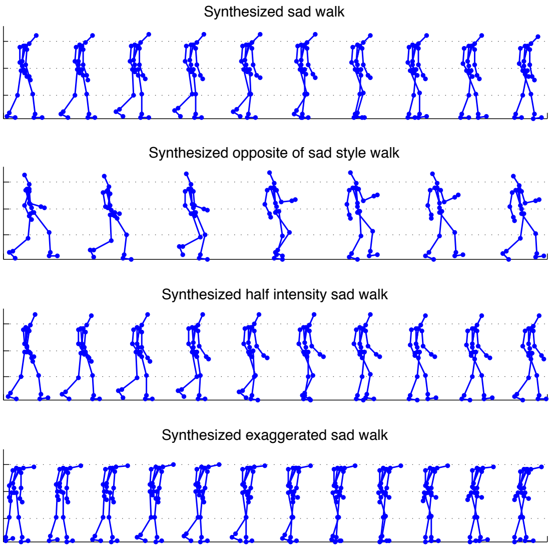


Fig. 6. Synthesized left step for original sad walk (first subfigure, *interp* = 1), interpolated opposite model (second subfigure, *interp* = -1), half sadness intensity (third subfigure, *interp* = 0.5) and exaggerated sad walk (last subfigure, *interp* = 2). Synthesized skeleton poses are displayed every 0.1 second.

In the first two approaches presented, a new model is calculated by interpolation of the original style models. The interpolated model represents a single new style and is used to synthesize a sequence of walk with a style not present in the original database. The HMM model takes into account the dynamics of the data, and the continuity between the steps of the walk sequence is hence ensured. But it also enables us to synthesize a sequence presenting style transitions. During the synthesis process, step models are concatenated in order to form a complete model of the entire walk sequence. If the concatenated step models represent different walks, the resulting synthesized sequence will display the corresponding style change. As the synthesis process takes into account the dynamics of the data to calculate the optimal parameter sequence corresponding to the concatenated walk model, the style transition will occur without abrupt jerks. This result is similar to the blending procedures applied to smooth the transition between two different mocap files over a few frames as it is usually done in pure mocap approaches. However, even if no jerks are present in the angle data and that the motion continues smoothly when the character goes from a step with style A to a step with style B, the transition will not appear as "natural" for all style transitions. If they are not surprised by something, humans do not abruptly change the style of their walk from one step to the next one. The change will rather occur continuously over a few steps.

Rather than concatenating directly a sequence of "style 1" step models to "style 2" step models, our interpolation procedure is able to produce step models corresponding to styles between style 1 and style 2. The style transition can thus be distributed over as many steps as the user wants. The steps concatenated in the whole sequence correspond to gradual interpolations, starting from style 1 and going to style 2. The walk sequence produced this way is thus a smooth and gradual transition between style 1 and style 2, transiting by in-between styles that were not present in the original database. The resulting transition looks more natural than a brutal style change occurring between one step and the next one, as illustrated in Figure 7.

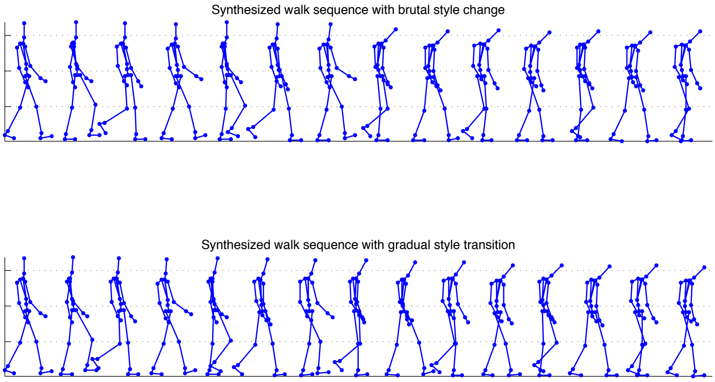


Fig. 7. Synthesized walk sequence with abrupt proud to sad style transition (first subfigure), and synthesized walk sequence with gradual proud to sad style transition (second subfigure). Synthesized skeleton poses are displayed every 0.5 second.

7 User Evaluation

Evaluating the quality of motion sequences is an open problem common to the whole character animation field, independently from the studied approach. Results are often illustrated with a video displaying some motion examples, but most works are presented without any kind of subjective evaluation by the user.

In [32], we proposed three tests to assess the quality of the synthesis results. These tests evaluated the naturalness of the synthesized motion, the style recognition of our eleven original styles and the comparison between original motion capture and synthesized examples. In one of the tests, users were asked to classify the original walk styles (without any modification) among 11 possibilities, and the results showed that the styles were correctly perceived most of the time, even if some close styles were sometimes switched. In this paper, a similar evaluation approach was chosen. Our evaluation consisted in two tests, evaluating respectively the quality of the interpolation between two styles (Section 7.1) and the style intensity control (Section 7.2).

Participants accessed to the evaluation tests through a web browser. They were presented one video at a time and asked to evaluate its content. Once their answer was selected and saved, they could not come back to previous videos. If they did not complete the test thoroughly, they could come back later, but the participant's results were saved even if the two tests were not completely finished. They had to start the video themselves by clicking on it, and could watch it as many times as they wanted. In the video sequences, motion was performed by a basic blue stick-figure character as shown in the previous figures.

Fifty-two naive evaluators took part in the evaluation, 22 females and 30 males, from 16 to 66 years old, with an average and standard deviation of 32 and 11 years, respectively. Every evaluator was presented a set of 10 videos or couples of videos for each test. Those videos were randomly picked by the evaluation program, and the evaluated set was thus different for each evaluator. The result of each test for each evaluator were taken into account only if the user completed all the evaluations of the given test. The final number of evaluators taken into account is not the same for both test as some users dropped the evaluation after taking the first test.

7.1 Style Interpolation Evaluation

In the first test, the evaluator was presented one video displaying three walk sequences with different styles, in a row. In the video, the first sequence was a walk with style A, the second one was a walk with style B, and the third sequence was an interpolation between styles A and B. The evaluation sequences corresponded to five possible pairs of A and B styles, and to five interpolation factors (0, 0.25, 0.5, 0.75 or 1) for each A and B pair. The set of evaluation videos for this first test thus contained 25 videos from which ten were randomly picked for each user to evaluate.

The user was asked to position the interpolated walk between style A and style B, by choosing between five possible answers:

- identical to style A (100% A + 0% B)
- close to style A (75% A + 25% B)
- in the middle between style A and style B (50% A + 50% B)
- close to style B (25% A + 75% B)
- identical to style B (0% A + 100% B)

Table 1. Confusion matrix of interpolation factor recognition test. The recognition rate is expressed in percents of the actual interpolation factor sequences presented to the evaluators, rounded to the unit.

		Evaluators classification (%) of interpolation factor				
		0	0.25	0.5	0.75	1
Actual interpolation factor	0	31	47	17	3	2
	0.25	3	48	39	8	2
	0.5	2	13	41	39	5
	0.75	3	5	22	46	24
	1	1	1	8	43	47

He was also asked to assess the naturalness of the interpolated walk by choosing if it seemed “Real”, “Synthetic”, or “I don’t know”.

Fifty-two participants completed this first test. Table 1 presents the confusion matrix of the interpolation factor recognition.

The interpolation factor was properly recognized by the evaluator in 42.7% of the cases, much higher than the 20% of mere chance. And in 46% of the cases, the interpolation factor was misidentified for one of its direct neighbors (for instance 0.5 or 0 instead of 0.25). In 60.6% of the cases, the evaluator was not capable of recognizing that the same file was displayed twice (when $interp = 0 \ or \ 1$), which demonstrates that even if the evaluators were capable of recognizing the trend (more like A or more like B), he is not very good at assessing small style variations. This poor recognition of the original styles can also be explained by the fact that even if the answers “identical to style A (or B)” were proposed to the evaluators, our formulation of the question asked to position the interpolated walk “between” styles A and B which might have mislead some people.

The mean value of the interpolation factor evaluated by the user for each one of the actual interpolation factor is presented in Figure 8, along with a 95% confidence interval. It can be observed that the confidence interval of the different interpolation factors do not overlap. Even if the exact interpolation factor was not always recognized, especially at the extremities ($interp = 0 \ or \ 1$), the global trend of the interpolation is very clear to the user who can make the distinction between each one of the interpolation factors even if he does not evaluate them at their exact value.

Figure 9 presents the results of the naturalness evaluation of the interpolated sequence. It can be observed that the value of the interpolation factor did not influence the naturalness of the sequence as perceived by the user, and our interpolated style models were seen in the same way as models of the original style ($interp = 0 \ or \ 1$).

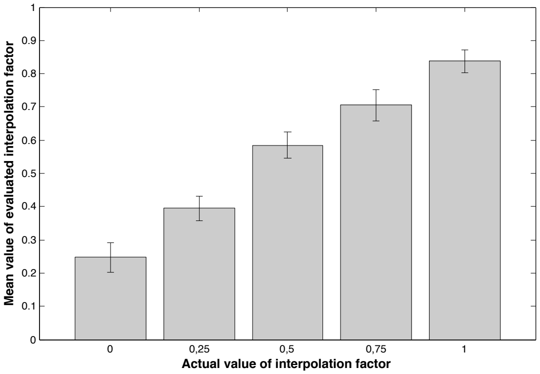


Fig. 8. Mean value of the evaluated interpolation factor (with .95 confidence intervals) for each one of the actual interpolation factor

7.2 Style Intensity Control Evaluation

In this second test, the displayed video contained two walk sequences in a row. The first sequence was a walk with style A, and the second sequence was a walk presenting a variation of style A. The evaluation sequences corresponded to five possible "A" styles, and to five interpolation factors between the style and the averaged style (-1, 0, 0.5, 1 or 2). The set of evaluation videos contained thus 25 files, from which ten were randomly picked for each user. The evaluator was asked to position the gradation of the style intensity of the interpolated walk in comparison to the original style A. The five possible answers he had to choose from were:

- Opposite of style A
- Neutral style
- Half intensity of style A
- Identical to style A
- Exaggeration of style A

The participant was also asked to assess the naturalness of the style intensity variation sequence, in the same manner as in the first test.

Forty-one users completed this second test. Table 2 presents the confusion matrix of the style intensity factor recognition.

The style intensity factor was properly recognized by the evaluator in 69% of the cases, which is much higher than the 20% of mere chance and even better than the result obtained in the first evaluation test. These results show that it was easier for the evaluator to quantify the intensity of a given single style than to evaluate the interpolation factor between two styles that might have been an improbable mix. In 27% of the cases, the interpolation factor was misidentified for one of its direct neighbors (for instance 0.5 or -1 instead of 0). Figure 10 presents the average intensity factor evaluated by the participants for each one of

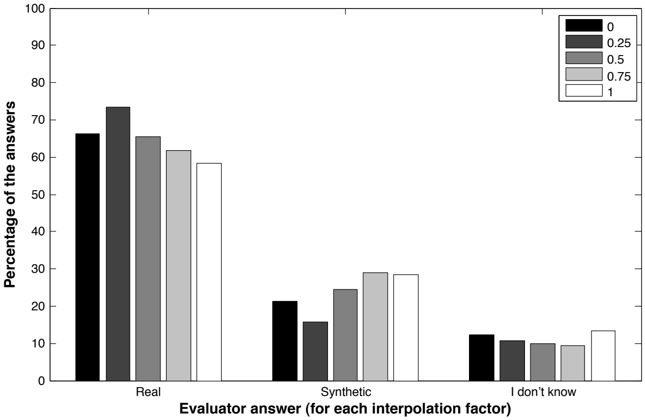


Fig. 9. Results of the naturalness test comparing the perception (real, synthetic or I dont know) of interpolated style sequences with each interpolation factor

Table 2. Confusion matrix of style intensity factor recognition test. The recognition rate is expressed in percents of the actual intensity factor sequences presented to the evaluators, rounded to the unit.

		Evaluators classification (%) of interpolation factor				
		-1	0	0.5	1	2
Actual interpolation factor	-1	92	6	0	0	2
	0	16	31	47	4	2
	0.5	0	4	52	45	0
	1	0	0	5	89	6
	2	11	0	3	4	82

the actual style intensities. It can be noticed that even if the exact interpolation factor was not always recognized, the global trend of the interpolation is very clear to the user, as it was also observed in the first test.

Figure 11 presents the results of the naturalness evaluation of the style intensity control. It can be observed that as long as the style intensity factor stayed in the 1 to 0 range, it did not influence the naturalness of the sequence as perceived by the user. Our averaged sequence (corresponding to $interp = 0$) seems thus as natural to the user as styles from the original database. However, the perceived naturalness decreases dramatically when we exaggerate the style or when we take its opposite. This can easily be explained as, by taking styles outside of the range of value of the walk present in our database (which were already exaggerated styles performed by an actor), the synthesis gives angles values that are outside of the range of possible humans movements and completely ruin the

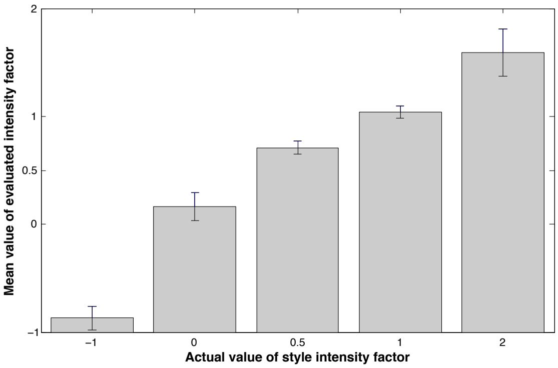


Fig. 10. Mean value of the evaluated style intensity factor (with .95 confidence intervals) for each one of the actual style intensity factors

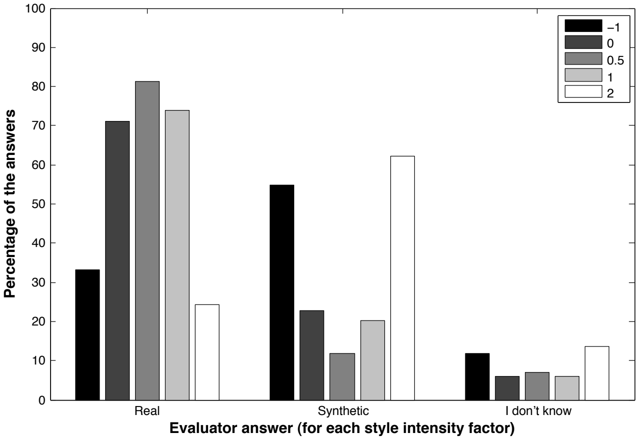


Fig. 11. Results of the naturalness test comparing the perception (real, synthetic or I dont know) of interpolated style sequences with each style intensity factor

naturalness of the motion. This issue will have to be further investigated in the future, and adding rules that constrain the angle values to stay in a plausible human range might be a way of dealing with this problem.

8 Conclusion

In this work, a set of eleven stylistic walk models based on Hidden Semi Markov Models was used as a basis for style interpolation and extrapolation, giving the user continuous control on the style of the synthesized motion while preserving its naturalness. New walk styles not present in the original database could be synthesized by interpolating the model parameters between different original styles. An average style model was also calculated by computing the mean of the parameters of the eleven style models. By taking this average model as a reference and changing the values of the interpolation factor, we were able to control the intensity of the style expression (values of *interp* between zero and one), to exaggerate the controlled style (values of *interp* greater than one), and to obtain new styles with characteristics at the opposite of the controlled style (values of *interp* lower than zero). Our model also enabled us to synthesize smooth and natural looking transitions between two different styles by progressive interpolation. Some examples of walk sequences synthesized with our method can be found at http://tcts.fpms.ac.be/~tilmanne. Qualitative user evaluation assessed that the trend of the interpolation factor was perceived by the user and that the naturalness of the motion was preserved for styles between the original styles.

In future works, constraints on the range of variation of the angles should also be added for style extrapolation, so that the synthesized styles remain physically plausible. The interpolation and extrapolation was conducted on 4890 parameters without any feature selection, but the influence of these parameters on the stylistic variations could be investigated, as some of them might be of lesser or greater influence than others on the perceived style. Our next step will be to implement our continuous style control and HMM synthesis procedure into a real-time framework, giving the user the possibility to control the synthesized walk in real time. We will also study how the style characteristics could be added directly on plain motion capture walk sequences.

Acknowledgment. This project was partly funded by the Ministry of Région Wallonne under the Numediart research program (grant N0716631). Joëlle Tilmanne was supported by the "Fonds pour la formation à la recherche dans l'industrie et l'agriculture" (FRIA) during part of this work. We gratefully acknowledge Alexis Moinet for his help in designing the online evaluation procedure and all our evaluators for their participation.

References

1. Brand, M., Hertzmann, A.: Style machines. In: Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques, pp. 183–192 (2000)
2. Bruderlin, A., Williams, L.: Motion signal processing. In: SIGGRAPH 1995 Proceedings, pp. 97–104 (1995)
3. Calinon, S., Guenter, F., Billard, A.: On Learning, Representing, and Generalizing a Task in a Humanoid Robot. IEEE Transactions on Systems, Man and Cybernetics 37(2), 286–298 (2007)
4. Chiu, C., Marsella, S.: A style controller for generating virtual human behaviors. In: Proceedings of AAMAS 2011, The 10th International Conference on Autonomous Agents and Multiagent Systems, vol. 3, pp. 1023–1030 (2011)
5. Elgammal, A., Lee, C.S.: The Role of Manifold Learning in Human Motion Analysis Human Motion Understanding, Modeling, Capture and Animation, pp. 1–29 (2008)

6. Forbes, K., Fiume, E.: An efficient search algorithm for motion data using weighted PCA. In: Proceedings of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pp. 67–76 (2005)
7. Forsyth, D.A., Arikan, O., Ikemoto, L., O'Brien, J., Ramanan, D.: Computational Studies of Human Motion: Part 1, Tracking and Motion Synthesis. Foundations and Trends in Computer Graphics and Vision 1(2/3) (2006)
8. Geng, W., Yu, G.: Reuse of Motion Capture Data in Animation: A Review. In: Kumar, V., Gavrilova, M.L., Tan, C.J.K., L'Ecuyer, P. (eds.) ICCSA 2003. LNCS, vol. 2669, pp. 620–629. Springer, Heidelberg (2003)
9. Glardon, P., Boulic, R., Thalmann, D.: PCA-based walking engine using motion capture data. In: Computer Graphics International, pp. 292–298 (2004)
10. Glardon, P., Boulic, R., Thalmann, D.: A Coherent Locomotion Engine Extrapolating Beyond Experimental Data. In: Proceedings of Computer Animation and Social Agent (CASA), Geneva, Switzerland, pp. 73–84 (2004)
11. Grassia, F.S.: Practical parameterization of rotations using the exponential map. Journal of Graphics Tools 3, 29–48 (1998)
12. Grudzinski, T.: Exploiting Quaternion PCA in Virtual Character Motion Analysis. In: Bolc, L., Kulikowski, J.L., Wojciechowski, K. (eds.) ICCVG 2008. LNCS, vol. 5337, pp. 420–429. Springer, Heidelberg (2009)
13. Hsu, E., Pulli, K., Popovic, J.: Style Translation for Human Motion. In: SIGGRAPH 2005 Proceedings, pp. 1082–1089 (2005)
14. HTS working group: The HMM-based speech synthesis system (HTS) Version 2.1, http://hts.sp.nitech.ac.jp/ (accessed 2010)
15. IGS-190. Animazoo (2010), http://www.animazoo.com
16. Johnson, M.P.: Exploiting quaternions to support expressive interactive character motion. PhD Thesis (2002)
17. Jolliffe, I.T.: Principal Component Analysis, 2nd edn. Springer Series in Statistic. Springer, New York (2002)
18. Lau, M., Bar-Joseph, Z., Kuffner, J.: Modeling Spatial and Temporal Variation in Motion Data. ACM Transactions on Graphics (SIGGRAPH ASIA) 28(5), 171 (2009)
19. Li, Y., Wang, T., Shum, H.: Motion texture: a two-level statistical model for character motion synthesis. In: Proc. of SIGGRAPH 2002, New York, NY, USA, pp. 465–472 (2002)
20. Lu, X., Hsu, R.-L., Jain, A.K., Kamgar-Parsi, B., Kamgar-Parsi, B.: Face Recognition with 3D Model-Based Synthesis. In: Zhang, D., Jain, A.K. (eds.) ICBA 2004. LNCS, vol. 3072, pp. 139–146. Springer, Heidelberg (2004)
21. Menache, A.: Understanding motion Capture for Computer Animation and Video Games. Morgan Kauffman Publishers Inc., San Francisco (1999)
22. Min, J., Chan, Y., Chai, J.: Interactive generation of human animation with deformable motion models. ACM Trans. Graph. 29(1), 9:1–9:12 (2009)
23. Min, J., Liu, H., Chai, J.: Synthesis and editing of personalized stylistic human motion. In: Proceedings of SI3D, pp. 39–46 (2010)
24. Pejsa, T., Pandzic, I.S.: State of the Art in Example-Based Motion Synthesis for Virtual Characters in Interactive Applications. Computer Graphics Forum 29(1), 202–226 (2010)
25. Rabiner, L.R.: A tutorial on hidden markov models and selected applications in speech recognition. Proc. of IEEE 77(2), 257–286 (1989)
26. Rose, C., Cohen, M.F., Bodenheimer, B.: Verbs and Adverbs: Multidimensional Motion Interpolation. IEEE Comput. Graph. Appl. 18(5), 32–40 (1998)

27. Shapiro, A., Cao, Y., Faloutsos, P.: Style components. In: Proceedings of Graphics Interface, Quebec, Canada, pp. 33–39 (2006)
28. Shoemake, K.: Animating Rotations with Quaternion Curves. In: Proc. of SIGGRAPH 2005, San Francisco, vol. 19(3), pp. 245–254 (1985)
29. Tanco, L.M., Hilton, A.: Realistic synthesis of novel human movements from a database of motion capture examples. In: Proc. of the Workshop on Human Motion (HUMO 2000), Washington DC, USA, pp. 137–142 (2000)
30. Taylor, G.W., Hinton, G.E.: Factored conditional restricted Boltzmann Machines for modeling motion style. In: ICML 2009 Proceedings of the 26th Annual International Conference on Machine Learning, pp. 1025–1032 (2009)
31. Tilmanne, J., Dutoit, T.: Expressive Gait Synthesis Using PCA and Gaussian Modeling. In: Boulic, R., Chrysanthou, Y., Komura, T. (eds.) MIG 2010. LNCS, vol. 6459, pp. 363–374. Springer, Heidelberg (2010)
32. Tilmanne, J., Moinet, A., Dutoit, T.: Stylistic gait synthesis based on hidden Markov models. EURASIP Journal on Advances in Signal Processing, 72 (2012)
33. Tilmanne, J., Sebbe, R., Dutoit, T.: A Database for Stylistic Human Gait Modeling and Synthesis. In: Proceedings of the eNTERFACE 2008 Workshop on Multimodal Interfaces, Paris, France, pp. 91–94 (2008)
34. Toda, T., Tokuda, K.: A Speech Parameter Generation Algorithm Considering Global Variance for HMM-Based Speech Synthesis. IEICE-Transactions on Information and Systems 90(5), 816–824 (2007)
35. Tokuda, K., Yoshimura, T., Masuko, T., Kobayashi, T., Kitamura, T.: Speech parameter generation algorithms for HMM-based speech synthesis. In: Proc. of ICASSP (June 2000)
36. Troje, N.F.: Retrieving information from human movement patterns. In: Understanding Events: How Humans See, Represent, and Act on Events, pp. 308–334. Oxford University Press (2008)
37. Unuma, M., Anjyo, K., Takeuchi, R.: Fourier principles for emotion-based human figure animation. In: SIGGRAPH 1995 Proceedings, pp. 91–96 (1995)
38. Urtasun, R., Glardon, P., Boulic, R., Thalmann, D., Fua, P.: Style-based Motion Synthesis. Computer Graphics Forum 23(4), 799–812 (2004)
39. Wang, Y., Liu, Z., Zhou, L.: Automatic 3D Motion Synthesis with Time-Striding Hidden Markov Model. In: Yeung, D.S., Liu, Z.-Q., Wang, X.-Z., Yan, H. (eds.) ICMLC 2005. LNCS (LNAI), vol. 3930, pp. 558–567. Springer, Heidelberg (2006)
40. Wang, Y., Liu, Z., Zhou, L.: Learning Style-directed Dynamics of Human Motion for Automatic Motion Synthesis. In: IEEE Conference on Systems, Man, and Cybernetics 2006, Taiwan, pp. 4428–4433 (2006)
41. Wang, Y., Xie, L., Liu, Z., Zhou, L.: The SOMN-HMM Model and Its Application to Automatic Synthesis of 3D Character Animation. In: IEEE Conference on Systems, Man, and Cybernetics 2006, Taiwan, pp. 4948–4952 (2006)
42. Wecker, L., Samavati, F., Gavrilova, M.: A multiresolution approach to iris synthesis. Computers & Graphics 34(4), 468–478 (2010)
43. Yamagishi, J., Nose, T., Zen, H., Ling, Z.H., Toda, T., Tokuda, K., King, S., Renals, S.: Robust speaker-adaptive HMM-based text-to-speech synthesis. IEEE Transactions on Audio, Speech, and Language Processing 17(6), 1208–1230 (2009)
44. Yamagishi, J., Kobayashi, T.: Average-voice-based speech synthesis using HSMM-based speaker adaptation and adaptive training. IEICE TRANSACTIONS on Information and Systems 90(2), 533–543 (2007)

45. Yamagishi, J., Kobayashi, T., Nakano, Y., Ogata, K., Isogai, J.: Analysis of speaker adaptation algorithms for HMM-based speech synthesis and a constrained SMAPLR adaptation algorithm. IEEE Transactions on Audio, Speech, and Language Processing 17(1), 66–83 (2009)
46. Yamazaki, T., Niwase, N., Yamagishi, J., Kobayashi, T.: HumanWalking Motion Synthesis Based on Multiple Regression Hidden Semi-Markov Model. In: 2005 International Conference on Cyberworlds (CW 2005), pp. 445–452 (2005)
47. Yoshimura, T., Tokuda, K., Masuko, T., Kobayashi, T., Kitamura, T.: Duration modeling for HMM-based speech synthesis. In: Fifth International Conference on Spoken Language Processing (ICSLP), pp. 29–32 (1998)
48. Young, S., Evermann, G., Gales, M., Hain, T., Kershaw, D., Liu, X., Moore, G., Odell, J., Ollason, D., Povey, D., Valtchev, V., Woodland, P.: The HTK Book (for HTK Version 3.4) (2009)

Modeling and Analyzing the Human Cognitive Limits for Perception in Crowd Simulation

Vaisagh Viswanathan and Michael Lees

School of Computer Engineering, Nanyang Technological University, Singapore
vaisagh1@e.ntu.edu.sg, mhlees@ntu.edu.sg

Abstract. One of the major components of Agent Based Crowd Simulation is motion planning. There have been various motion planning algorithms developed and they've become increasingly better and more efficient at calculating the most optimal path. We believe that this optimality is coming at the price of realism. Certain factors like social norms, limitations to human computation capabilities, etc. prevent humans from following their optimal path. One aspect of natural movement is related to perception and the manner in which humans process information. In this paper we propose two additions to general motion planning algorithms: (1) Group sensing for motion planning which results in agents avoiding clusters of other agents when choosing their collision free path. (2) Filtering of percepts based on interestingness to model limited information processing capabilities of human beings.

Keywords: Agent-Based Model, Sensing, Crowd Simulation, Motion Planning, Visual Cognition, Group Based Perception, Information, Collision Avoidance.

1 Introduction

Crowd simulation is a field that has recently been gaining significant attention because of its usefulness in various applications, ranging from simulation of emergency evacuation to animation of large crowds in movies and games. There are a number of different approaches which are typically applied to modeling of human crowds. These include: flow models [21], force-based models [12] and agent-based models [23]. All models offer different ways of describing human motion and make different assumptions about how interacting individuals affect one anothers motion.

In this paper we focus on agent-based models of crowds; one key aspect of which is *navigation*. Navigation is defined as the process or activity of accurately ascertaining one's position and planning and following a route. In the context of crowd simulation, navigation is generally considered to be the process of planning a route towards a destination and following this route. We refer to the former as *path planning* and the latter as *motion planning*. The higher level path planning is typically done using A-star or other similar algorithms and deals with the static aspects of the environment. Motion planning is a term borrowed from

M.L. Gavrilova and C.J.K. Tan (Eds.): Trans. on Comput. Sci. XVI, LNCS 7380, pp. 55–76, 2012.

robotics which originally means detailing a task into discrete motions. In the context of crowd simulation, we use the term motion planning to refer to the task of finding a collision free velocity to get from the current point to the next waypoint in a planned path.

There are a number of existing motion planning methods that can effectively and efficiently calculate trajectories that avoid all collisions for agents, even in very dense environments. For robots and computer games, this might be the ideal goal: perfect, smooth and efficient motion. However, for applications like simulation of emergency evacuations the goal is obtaining realistic motion and not smooth and efficient motion. While we all thrive to be mechanically efficient, this is hardly always the case. There exist, among other things, social norms and limits to mental processing capabilities that prevent individuals from following their ideal preferred path. Also, humans do not necessarily use optimality (in any sense) to determine their preferred path. Our approach is a more naturalistic one [19] in that we feel the navigation models should explicitly consider and model human inadequacies and limitations.

The agent-based models (ABMs) we consider, consist of large numbers of heterogeneous, autonomous entities inhabiting a spatially explicit, partially observable environment; macro-level dynamics are said to emerge through the asynchronous interactions among these entities [2,6]. Each of these individual entities will iterate through a sense-think-act cycle, where agents obtain information from their environment through *sensing*, make a decision through *thinking* and finally carry out their decision by *acting*. In many application areas in which ABMs have been applied, including crowd simulation, the emphasis is generally on describing thought processes accurately via rules. However, sensing is a critical aspect in the modeling process and can greatly impact both the individual and emergent properties of the system. The terms perception and sensing are often used interchangeably in the simulation literature. For clarity in explanation we use the term *perception* to define the complete process of obtaining a set of (possibly filtered) percepts from the environment. *Sensing*, on the other hand, we define as the process of obtaining raw information from the environment. In this definition, and in our model, sensing is a part of perception.

Miller's seminal work [24] on human cognition revealed two important characteristics of human cognition: 1. Humans constantly group together similar data into *chunks* of information. 2. At any given time, a human can only process a limited amount of information. In this paper, we make the assumption that this limited capacity results in humans being attracted towards certain kinds of information, e.g. a bright light or a celebrity; this, in turn, results in other information in the environment being unnoticed. By organizing information into chunks, humans are able to use their limited information processing capability more efficiently. This ability can manifest itself in different ways. We assume that during motion planning, humans will process a group of people coming towards them as a single obstacle rather than many individuals. This grouping not only helps the person make use of his limited information processing capacity more efficiently, it also helps him/ her conform to social norms that instruct him/ her that walking through a group of interacting people would be rude.

In this paper, we propose an alternative information based naturalistic perception system, which does not focus on explicit vision, but rather treats the entire human perception system as an information processing entity. We do eventually plan to extend the use of this *information based perception* for higher level path planning and decision making. However, this is beyond the scope of this paper. The remainder of this paper is organized as follows: Sect. 2 gives an overview of the existing work; the theoretical basis of the proposed model is explained in Sect. 3; in Sect. 4, we show some simulation results that illustrate the effects of implementing the proposed theory; and finally, Sect. 5 concludes this article and gives a brief overview of possible future directions of work.

2 Related Work

This section of the paper is divided into two parts: In the first section, we present some of the existing work in motion planning for virtual crowds and in the following section, we present some of the work on whose basis the presented limited information model for agents was developed.

2.1 Motion Planning

There are various different approaches to motion planning in agent based models with non-discrete space. The earliest agent based approach to collision avoidance was proposed in Reynolds' seminal paper [29] on a model of the flocking behavior of birds. There are various simple approaches to modeling human motion like Klein and Köster's [20] use of an electric potential based model; positive charges are assigned to goals and negative charges to obstacles and agents. Okazaki and Matsushita [25] uses a similar approach of using magnetic poles instead of columbic charges. There are also slightly more complicated approaches like the one proposed by Pettré et al [28] which considers the effect of errors in perception on motion planning. In this section, only two of the most popular models for motion planning and collision avoidance used in agent based models, viz. the Social Forces model and the Reciprocal Velocity Obstacle (RVO) model are presented.

The social forces model was first introduced in Helbing's paper [13]. In this model, each agent is modeled as a particle that has multiple forces acting on it. Repulsive forces help in collision avoidance and attractive forces model goal directed and grouping behavior. Over the years, this model has been extended and combined with other higher level behavior models. For example, in [17]more complicated group movement was modeled with an underlying social forces model for collision avoidance. In his thesis, Still [31] criticized the heavily mathematical approach which, according to him, is too complicated to be the natural way in which humans try to avoid crowds.

Another ABM that is increasingly becoming popular for collision avoidance is based on the idea of using the relative motion of objects to determine their time to collision. A velocity is then selected which maximizes this time. This algorithm, based on Reciprocal Velocity Obstacles (RVO) was first extended for use with multi agent systems in [1]. Since then there have been several modifications and improvements to the system but the underlying algorithm still remained the same. CLEARPATH [10] which mathematically optimized RVO was the first to introduce a change in the underlying algorithm. Later, Guy et al. [8] introduced an entirely new approach to RVO that was based on computational geometry and linear programming. They called this new approach *RVO2*. This method further improved the efficiency and smoothness of the system. In [11], the introduction of a personal space factor and an observation delay made the algorithm more appropriate for virtual humans.

In[9], Guy et al. introduced an extension to RVO in the form of a higher level navigation based on the principle of least effort. While it is obvious that rational humans would prefer taking the path of least effort, as was explained in Sect. 1, humans do not have perfect knowledge or perfect calculation. Also, it is arguable whether humans are always rational enough to choose least effort as their goal. Another important optimization that was introduced in this paper was using the idea of clustering very distant objects into KD-trees to reduce computational cost. While this might sound similar to the idea that is suggested in our paper, there are two fundamental reasons why this is different from our algorithm: Firstly, we use multiple levels of clustering which will be explained in more detail in Sect. 3.1. Secondly, the motivation and hence design is significantly different: we use clustering as a reflection of how agents perceive their environment and not an optimization for collision avoidance.

2.2 Limits of Human Perception

Most motion planning systems focus on optimality of motion. This is either in terms of selecting a path which attempts to avoid collisions with minimal deviation, or in the sense that they are capable of obtaining accurate information about the environmental state. While certain extensions have been previously suggested for making models more realistic, there hasn't been any approach to bring about a human like perception system in multi agent based crowd simulation.

Miller [24] proposed the idea that humans process 7 ± 2 chunks of information. Cowan [5] argued that humans can actually only cognitively process 4 ± 1 chunks at any given time. Also, others [15] have shown that humans try to group together similar information so that information can be encoded in the simplest possible format. This is called the simplicity principle [15]. Based on these ideas, we propose a method that will emulate how humans perceive groups whenever possible and propose a system in which the agents avoid groups rather than individuals. We refer to this perception system as Group Based Perception (GBP). We have done this using the Evolving Clustering Method (ECM) [30] and computational geometry based RVO2 [8]. But our approach can, in principle, use almost any clustering and collision avoidance algorithms.

Some studies[16,26,32] have shown that humans only pay attention to certain salient features in the objects that they perceive. This results in them not noticing changes in items that are not of interest to them. In[26], the authors classify elements as either central interest or marginal interest elements and prove that the internal representation of the visual world is rather sparse and essentially contains only central interest information and not information of objects of marginal interest. The factors that influence how interesting a particular object is is extensively discussed in[16]. In the present paper, we do not propose to model all the complexities of human perception and visual cognition, we would rather like to propose an agent based model for crowds which can not only show a basic implementation of these ideas but can be easily extended when required, to model more complicated visual cognition.

Broadbent [3] has extensively discussed the idea of using information theory for modeling human perception. He introduced various studies that indicate that humans have an upper bound on their capacity for holding information for perception. For a single dimension, this limit is roughly estimated to be about 5-6 percepts. For more than one dimension, the number of discernible alternatives is larger but not as large as would be expected if each dimension was completely independent. The idea of humans being able to process only a limited amount of information is not new to computer animation either. Hill [14] was one of the first to introduce the importance of cognition in sensing. Courty et al. [4] used a saliency map based approach and Kim et al[18] used cost-benefit analysis in a decision theory based approach to determining the interest points. Grillon and Thalmann [7] automated this process of interest point determination. They used criteria like proximity, relative speed, relative orientation and periphery to determine the interestingness of various features. Similar criteria are used in this paper. However, our application being particularly collision avoidance, the information which the agents perceive are dynamic obstacles, i.e. other agents or groups of agents. The proposed Group Based Perception system for humans differs from traditional perception systems in that, rather than using a level of perception limited by distance or occlusion of sight, we limit the amount of information, or number of obstacles, which the agents can cognitively process.

3 The Theory

This section explains the agent perception system that is proposed in this paper. The sense-think-act cycle of agents was introduced in Sect. 1. Figure 1 illustrates how motion planning works in an agent in terms of this sense-think-act cycle. An agent's perception can be described by a function $f : Env \rightarrow p*$, where $p*$ is the set of percepts. Each percept p is then processed by the agent in its decision making process, which in turn will determine an appropriate action for collision avoidance. In our case, the motion planning module is passed a set of percepts which consists of neighboring agents and static obstacles which it

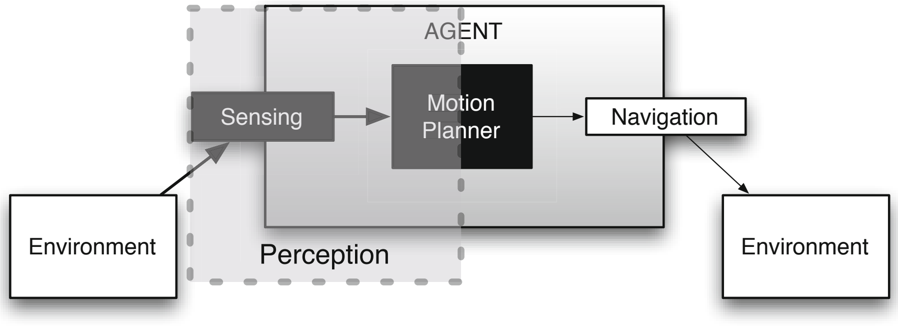


Fig. 1. An agent perceives and then acts

processes to find the optimal or most appropriate velocity for reaching the goal. Typically, this list of neighbors is a set of agents within some cone of vision or some distance away from the agent. In this paper we propose a modification to the perception procedure such that it takes place in three phases: *clustering*, *sensing* and *filtering*. Figure 2 gives an overview of the complete process that is detailed in the following subsections.

3.1 Clustering

Central to our information based perception system is the definition of *information units*. In traditional crowd simulation each individual agent or obstacle is considered as a percept, i.e. as an entity which should be processed by the motion planning system. The first assumption of our approach is that percepts can be both individuals and groups of other pedestrians. Whether an individual considers a group or individual is related to the *coherence* of the group and also the distance of the perceiving agent from the group. In order to achieve this, we perform a global clustering across the entire environment of agents. We create n_l layers within the environment, each layer identifies and stores groups of a particular size, with increasing layer numbers storing groups of increasing size. The criteria which determines what actually constitutes a group is itself unknown and probably highly dependent on the individual. We make the assumption that only the proximity of the individuals to one another determine whether a collection of people is perceived as a *group*.

For reasons of efficiency we simplify things by performing a single clustering (for each level) for all agents at every time-step, the consequence is that we are implicitly assuming all agents have the same notion of what constitutes a group. In reality this assumption may be too strong, different people may have different criteria for what they perceive as groups.

While there are various clustering techniques that could be used for grouping agents, we chose to use ECM [30] because: 1. It does not require the number of clusters to be predefined and 2. It can restrict the maximum radius of a cluster.

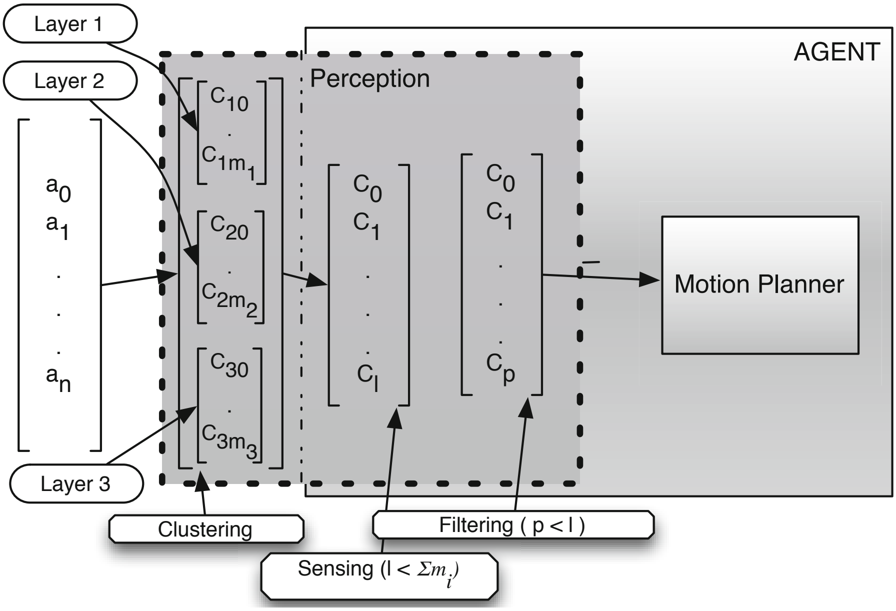


Fig. 2. Perception in agents takes place through three stages: 1. Clustering is done at a global level. The dotted line indicates this separation. Agents $a_{0\cdots n}$ form m clusters $c_{i,0\cdots m_i}$ in layer i where $m \leq n$ and $i \leq K$ where K is the predetermined number of layers. 2. Sensing is the process by which the agents perceive only a subset $(c_{0\cdots l})$ of these clusters $(c_{0,0\cdots m_0}, \ldots, c_{K,0\cdots m_K})$. 3. Filtering further reduces the size of this list and models human visual cognition.

It is also important to remember that this clustering is done dynamically at *each step* and not as a one time calculation of groups.

First the number of clustering layers iInteresting and well written paper on navigation and optimality,taking into account cognitive aspects. While the main premises on which the paper is based might be somewhat schematic, the work is very sounds decided. In the Fig 3, we illustrate information based perception using two layers. The algorithm starts by initializing a single agent as the first cluster, the maximum clustering radius for layer i, r^i_{max} is fixed (3 and 4). Each subsequent agent is then compared with every existing cluster to assess its suitability for addition to that cluster. Suitability is determined by the distance of the agent from the cluster. If the agent lies within an existing cluster, it is simply added to that cluster without updating either the radius or the cluster center. Otherwise, the cluster whose center is closest to the agent is determined. If the agent can be added to this cluster, without exceeding the maximum allowed radius for the cluster, then the agent is added to the cluster and the cluster's radius, center and velocity are updated. On the other hand, if adding the agent violates the maximum radius criteria, then a new cluster is created at the location of the agent.

Once this process is completed for layer i, the process is repeated for layer $i+1$ until the clusters for all the layers are determined. This process is illustrated figuratively in Fig. 2. The clustering function for layer i, cf_i allocates one and only one cluster for each agent in each later. This can be represented mathematically as shown below:

$$\forall a_k \in A \; \exists j \in [1, m] \quad cf_i : a_k \rightarrow C_{ij} \text{ where } 1 \leq m \leq n \tag{1}$$

$$\forall a_j \in A \quad C_{0j} = a_j \tag{2}$$

$$r^1_{max} = 2\alpha * a_r \tag{3}$$

$$\forall i \geq 2 \quad r^i_{max} = 2\alpha * r^{i-1}_{max} \tag{4}$$

Here a_r is the average radius of an agent[1] in A which is the set of all agents; C_{ij} indicates cluster j in layer i; m is the number of clusters and n is the number of agents. α is a parameter that determines the size of clusters and the range of each region (Fig. 3). Through experimentation we found the most pleasing results with $\alpha = 2$.

To correct certain undesirable behavior produced by ECM clustering, a modification was made to the algorithm. With large values of r_{max}, there is a chance that distant agents might be grouped into sparse clusters. To counter this problem, we define a *checking circle* as a circle of radius $2\alpha a_r$. If there are no agents within this checking circle, then the cluster is considered sparse and the cluster is removed. The sparseness check is done five times. First with the circle centered at the center of the cluster and subsequently with the checking circles centered at a distance equal to half the distance from the center of the cluster along each of the coordinate axes.

3.2 Sensing

Once the agents have been clustered, the next step is to make use of these clusters for motion planning. As previously explained, existing motion planning algorithms need a list of nearby agents and obstacles to determine the most appropriate velocity. The sensing module of our proposed perception mechanism uses the set of n_l layers created in the clustering module. The list of things to avoid will now consist of agents, obstacles and groups of agents. This list of nearby objects is now calculated from the multiple clustering layers as shown in Fig. 3.

From each cluster layer (explained in Sect. 3.1) a ring shaped *perception region* pr_i is defined for each agent. This region can be considered as a modification of the sensor range which is used in most ABM. In the first region (pr_0), immediately surrounding the agent performing the sensing, the agent perceives other individual agents from the clustering layer 0. This region extends to a distance $r_{pr_0} = 2\alpha * a_r$ from the agent's current location. For each subsequent region, the

[1] In the experiments in this paper, it is assumed that all agents have the same radius. Hence, the radius of every agent is the same as the average radius.

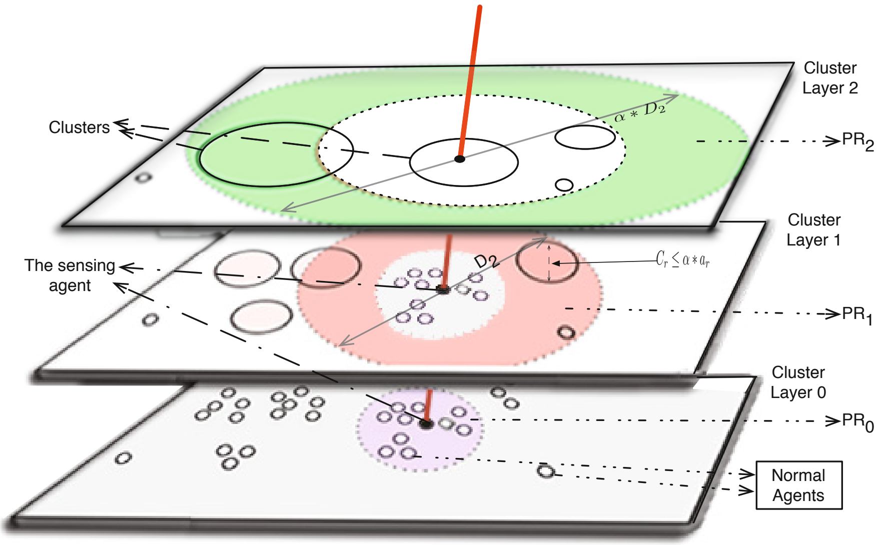


Fig. 3. The figure illustrates how the opaque agent senses objects using 2 clustering layers. The bottom layer is the original environment and the two planes above show the two clustering layers. Clusters in layer 2 are generally bigger than in layer 1. Solid lined circles indicate the normal agents and the clustered agents. The dotted lines show the regions of perception.

ring shaped region of sensing is from the boundary of the previous layer's region to the boundary of a circle of radius 2α times the radius of the preceding region. So for region pr_1 the agent perceives groups of maximum size r^1_{max} (Fig. 3). If the nearest edge of their minimum enclosing circle is within a distance d, such that $r_{pr_0} < d \leq r_{pr_1}$. The result is a list of obstacles which consists of clusters of various sizes and individual agents.

3.3 Filtering

As explained in Sections 1 and 2.2, a human being does not cognitively process every single object or obstacle that is within its vision. In other words, an agent can only process a limited amount of information. The information that is processed will be that which is deemed most interesting or important to the agent. So each object in the list obtained from perception is assigned an interestingness score of between 0 and 1 (1.5 for exceptional cases). During the sensing process each agent is given an *information limit* a_{IL}, indicating the total amount of information that can be processed by the agent. This limit is a parameter than can change as the stress level or other characteristics of the agent changes [27].

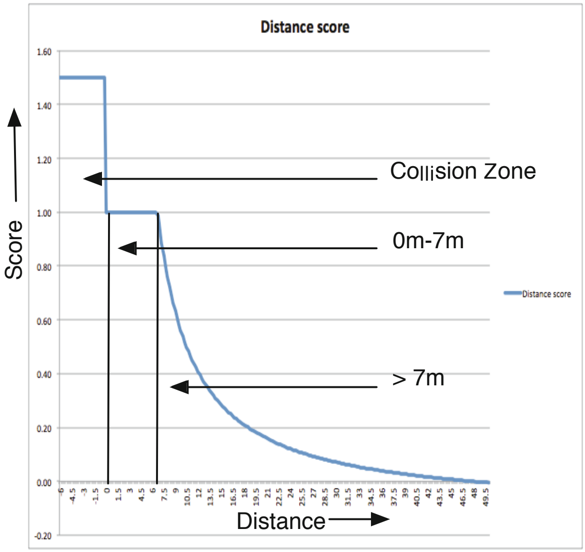


Fig. 4. This graph shows the variation of distance score with distance (in metres) used in experiments. A score of 1.5 if a collision has already occured, a score of 1 if it is within 7m and an exponentially decreasing score beyond that distance.

For this paper we assume that interestingness of an object depends on two criteria: 1. The distance of the object from the agent. 2. The angle that the object currently forms with the direction of motion of the agent. A third factor indicating the innate interestingness of the object being perceived can also be used; this can represent a lot of other properties related to interestingness. For example, an object's speed, color, action or something more subjective, i.e. it is of interest only to this agent because of certain properties of the agent. For e.g., for a thirsty agent, a water cooler would be interesting whereas it is unlikely to catch the attention of someone else. A more exact definition of interestingness is not the focus of this paper, but the general model here should be able to adapt to more sophisticated definitions.

Based on these criteria, a score is given to each agent. A distance score of 1.5 is given if the distance between two agents is less than or equal to zero. This is to ensure that in high density scenarios where a collision does occur, a collision recovery mechanism is forced on the objects regardless of what angle or how interesting the object is. For other distances the following equation is used to calculate the score for a distance d. γ and k are parameters which were fixed at 5.0 and 1.11 respectively to get a curve as in Fig 4.

$$S_d = max(min(1.0, e^{\gamma/d} - k), 0.1) \tag{5}$$

An angle score of 1.0 is given to all objects forming an angle of less than a_{min} with the agent's direction. For all agents that form an angle of more than a_{max}

with the agent's direction, a score of $(1-\beta)$ is given. For our experiments a β value of 0.9 was used and this is illustrated in Fig. 5. For all angles in between, the angle score linearly decreases to $(1-\beta)$ from 1. This is assigned based on the following equation (Fig. 5). All angles are in radians:

$$S_\theta = 1.0 - (\beta * (a - a_{min})/(a_{max} - a_{min})) \tag{6}$$

The final score for the object is calculated as the product of the S_θ and S_d (as long as distance score is not 1.5). This list of objects is then sorted on the basis of the score that is determined. Objects are then removed from the head of this list in turn and added to the final list of perceived objects as long as the cumulative score of all the perceived objects does not exceed the information limit for the agent, a_{IL}. All the remaining objects are dropped from the list of objects sensed and the final list of percepts $p*$ is obtained. In case two objects have the same score, the objects that are moving towards the perceiving agent are given precedence, subsequently closer objects are given preference.

For the implementation in this paper we pass the shortened neighbor list to RVO2 [8] for calculating the velocity at each time step. Our hypothesis is that the 3-step perception process proposed by us in this paper provides an improvement in two ways: Firstly, there are fewer neighbors and hence, fewer constraints for a given sensor range. Secondly and more importantly, more human like results can be obtained as will be illustrated in Sect. 4.

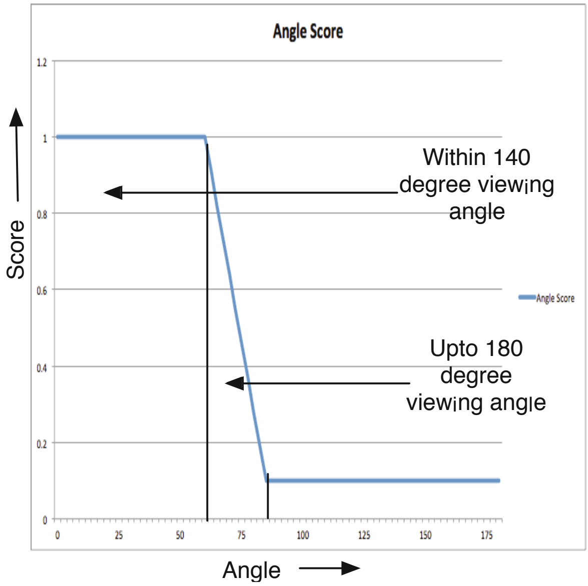


Fig. 5. This graph shows the variation of angle score with the angle(in radians) formed by the object with the agent used in experiments. For objects forming an angle of less than 70° (viewing angle 140°, a score of 1 is given. For objects forming an angle of up to 90°, the score linearly decreases to 0.1 which is the angle score for all remaining obstacles.

4 Results

We are currently working towards gathering real world data that would ideally be used for validation of the proposed model. Nevertheless, in the following sections, we use the ideas introduced in Sects. 1 and 2.2 as the basis for validating different aspects of the proposed model. Two quantitative measurements are used to analyze the model: *Effort Expended* and *Inconvenience Cost*. In proposing their least effort based approach to motion planning [9], Guy et al. used a measure of effort expended to demonstrate the usefulness of their model. This effort was calculated as follows:

$$E = m \int (e_s + e_w |\boldsymbol{v}|^2)\, \mathrm{d}t^{2} \tag{7}$$

In this paper, we use the same measure of effort to analyze and validate our model. For simplicity, we take all agents to have the same average mass of 70 Kg. However, this only measures the mechanical effort involved. To measure the amount of effort spent in decision making, we introduce *inconvenience cost*. The inconvenience cost is the number of time steps in which the agent chose a velocity other than its preferred velocity i.e., the number of times they have to avoid a collision.

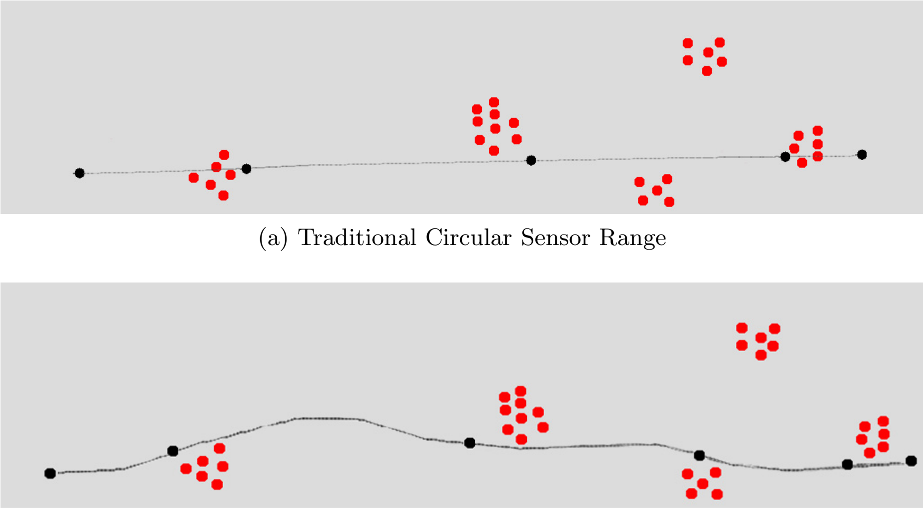
We consider four different scenarios which we consider to be a good way to evaluate the overall performance. First, we simply demonstrate the effects that Group Based Perception can have on the trajectory of an agent both in visual and in quantitative terms. Next, we present the benefits of using a multiple layers of clustering. Following this, we conduct an experiment to demonstrate how group based perception is essential if we are to model a human being's information processing limits. In the final experiment we analyze the effects

4.1 Group Based Perception

In this experiment we compare the results of using RVO2 with a traditional simple circular sensor range against RVO2 with a Group Based Perception system. The intention is to show the effect of perceiving agents as groups. Our hypothesis is that by perceiving groups as obstacles the simulation will generate more visually natural motion. In Fig. 6, there is a single black agent moving towards the right, and a number of groups of red agents moving towards the left. The black trail shows the path that is taken by the black agent. It can be seen that in Fig. 6a where GBP was not used, the agent walked through other groups. Since RVO2 enforces each agent to do half the work to avoid collision, the agents within the group individually give way through its center for the oncoming agent to pass.

At present we base our argument on the discussion in Sects. 1 and 2.2, due to social norms and the human tendency to group information together people generally try to move around an entire group rather than walking directly through

[2] $e_s = 2.23 \frac{J}{Kgs}$ and $e_w = 1.26 \frac{Js}{Kgm^2}$ for an average human [33].



(a) Traditional Circular Sensor Range

(b) Group Based Perception

Fig. 6. Experiment 1: The effect of Group Based Perception

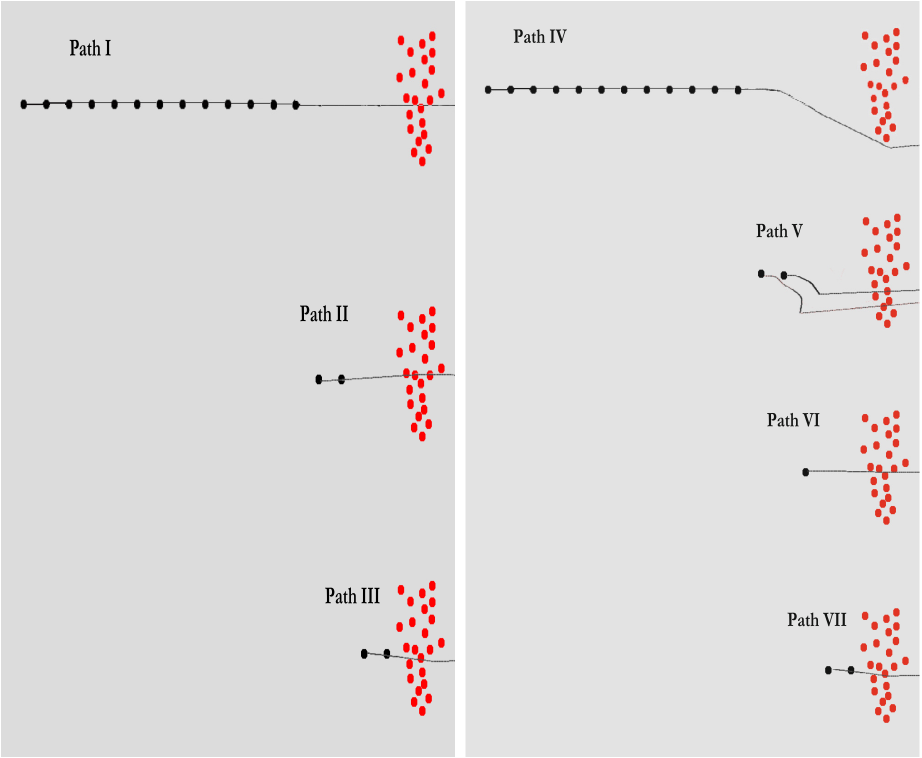
Table 1. Quantitative analysis of Group Based Perception

Agent Considered	Effort ($*10^5$)		Inconvenience Cost	
	Without GBP	With GBP	Without GBP	With GBP
Black Agent	71730	71726	120	148
All other agents (average)	1884	1880	14.28	6.56

a group. As shown in Fig 6b our perception algorithm is capable of generating motion which avoids entire groups.

An analysis of the effort expended and the inconvenience cost gives some interesting results. Since the simulation is executed for a given number of time steps, the effort expended is normalized with the progress towards the agent's goal. This is to avoid slow or stationary agents from being considered to be more efficient despite traveling a lesser distance. On comparing the normalized effort in the two scenarios of the black agent, it is found that despite having a much longer path, the GBP enabled agent expends slightly lesser (practically the same) amount of effort than the other. This is because the non-GBP agent has to slow down to wait for the other agents to give way before it can proceed and thus progresses less towards the goal.

The inconvenience cost comparison gives another interesting, though not surprising, result. The inconvenience cost to the black agent of using Group Based



(a) Using traditional sensing

(b) Using Group Based Perception

Fig. 7. Experiment 2: Effect of multi layered clustering

Perception is higher because of the more indirect path that it has to take. However, the average inconvenience caused to all the other agents is significantly lesser. This conforms with the general human reluctance to inconvenience others. It also gives the interesting idea that even if the same amount of mechanical effort is expended in following two different paths, the amount of decision making required for each path might be significantly different.

4.2 Effects of Multi Layered Clustering

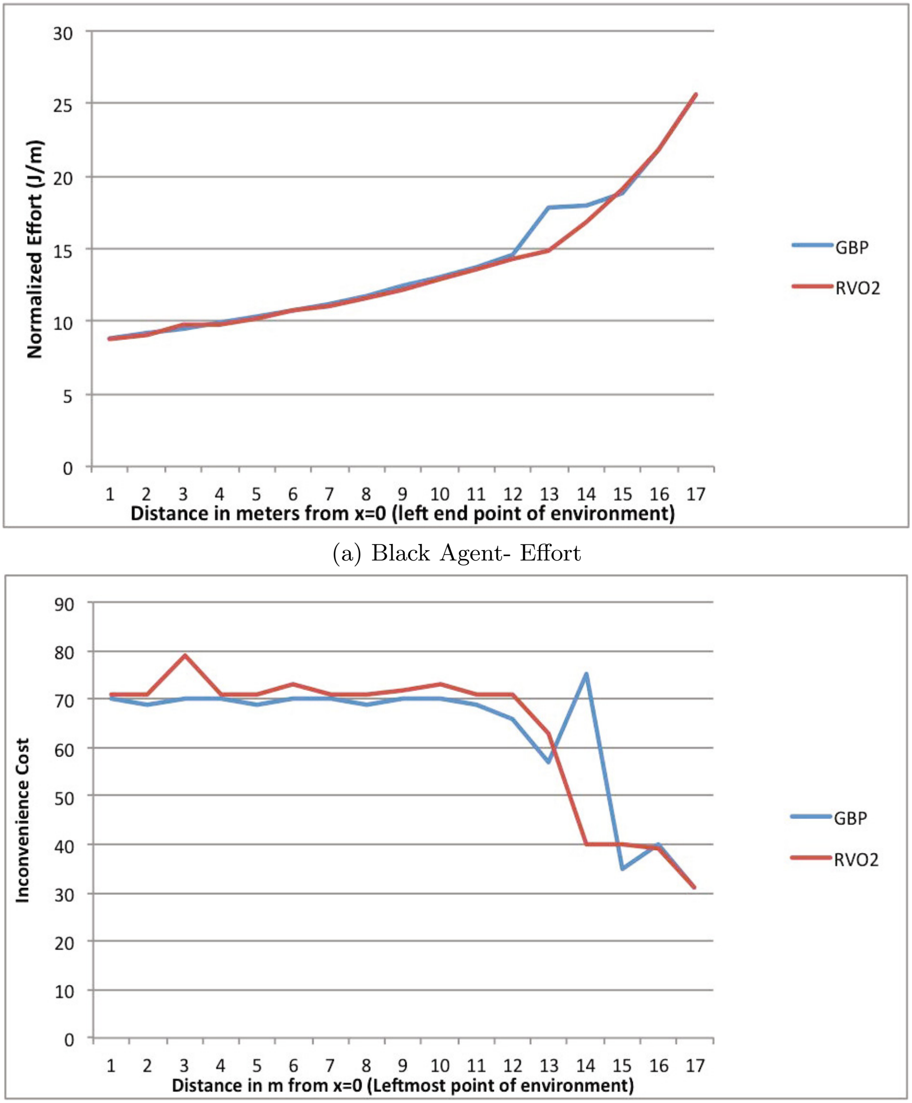
In this experiment, we studied the simple scenario where a single (black) agent had to get past a big group of agents to get to its goal. The same experiment was performed by keeping the agent at different distances from the group. The objective of this experiment is two-fold. Firstly, it demonstrates the importance and the working of the multi-layered clustering (Sect. 3.2) used. Secondly, it demonstrates that when agents are very close to each other, where RVO2 already performs well, the Group Based Perception does not interfere with RVO2's functioning.

To recap, the multiple layers are used to describe groups of varying size at varying ranges of perception. This means agents will perceive other agents as groups or individuals depending on the distance; as an agent moves towards a group it will start to perceive the group as individual agents.

When GBP isn't used, the path followed does not change significantly with distance. The agents in the path of the black agent, give way to the agent, and the black agent just proceeds straight through the center of the large group (Path A in Fig. 7a). In the last few cases (Paths B and C), the path is slightly different because the black agent does not have enough time to plan for a smooth, straight path and hence there is a slight deviation. Also, similar to the experiment in Sect. 4.1 it is forced to slow down in the process.

The result produced when GBP is used is more varied. Four distinctly different paths (labeled D, E, F and G in Fig. 7b) are produced based on how far the oncoming black agent is from the big group. At distances between x-y m away, the agent has enough time to perceive the group and avoid it completely (Path D). At distances between y-z m away, due to the size of the group, the agent gets too close to the group such that it then perceives the group as individuals. At this time (as described in Fig. 3) the agent performs motion planning on all the individual agents and as a consequence moves through the group, shown by path E. Path F is obtained in a similar fashion; however, the black agent is too close to the group to discern the effect of GBP. At distances closer than this, the path followed by the agent (Path G) is exactly the same as that followed by the agent not using Group Based Perception (Path C). We argue that this type of flexibility in the perception of groups is critical to creating more natural behavior, humans will adapt what they perceive based on success or failure of their attempt to avoid larger groups.

Figures 8a and 9a show a comparison of the effort expended by the black agent and the average effort expended by all the remaining agents while using a traditional sensor range and Group Based Perception. As in the previous experiment (Sect. 4.1) there is hardly any difference in the effort expended in both scenarios (except for a slight increase for trail E). However, an interesting pattern can be observed in the inconvenience measurement. Firstly, the inconvenience for the rest of the group, is always lesser when GBP is used and almost the same for the black agent when path D is followed. However, when path E or F is followed there is a spike in the inconvenience curve. This can be explained by considering the fact that in both path E and F, the black agent changes its planned path suddenly and decides to go through the group, thus not only increasing its own inconvenience but also the inconvenience caused to others in the group who have to move to give way to the agent. Finally, when path G is followed both the effort and inconvenience count are exactly the same as for path C.



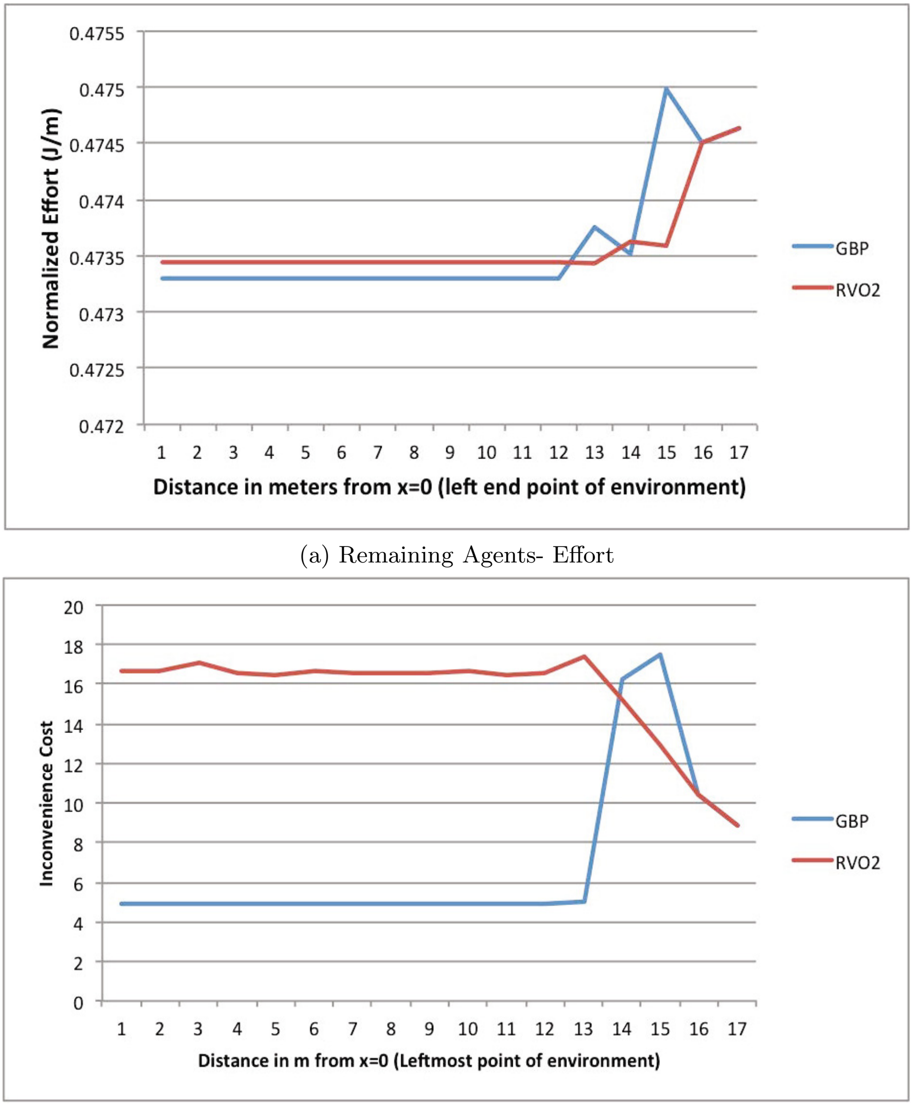
(a) Black Agent- Effort

(b) Black Agent- Inconvenience

Fig. 8. Experiment 2: Quantitative Analysis - Black Agent

4.3 Filtering Necessitates Group Based Perception

In Sects. 1 and 2.2, the fact that humans have limited information processing capacity was explained. In this experiment, we demonstrate that if we are to model a human being's limited information processing capability, it is also necessary to use Group Based Perception. This is done by observing the simple scenario of an

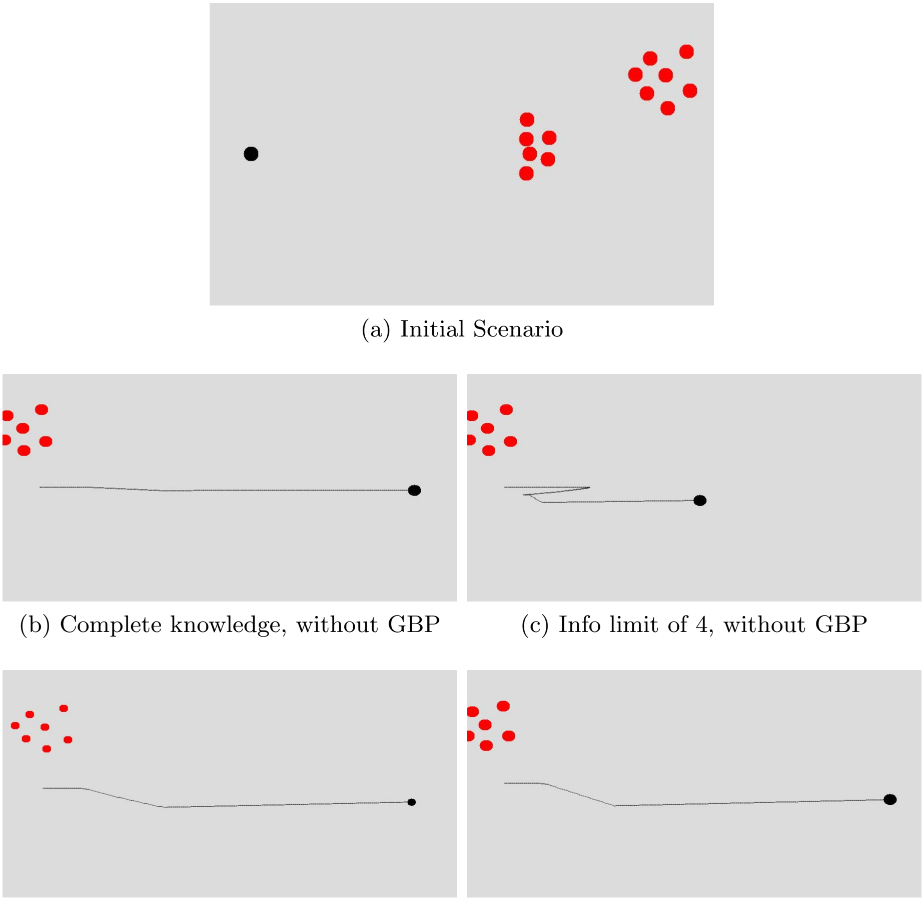


(a) Remaining Agents- Effort

(b) Remaining Agents- Inconvenience

Fig. 9. Experiment 2: Quantitative Analysis - Remaining Agents

agent moving towards two groups of other agents (Fig. 10). When no information limit is imposed on the agent, and a normal circular sensor range is used, the agent, as expected, follows a nice straight path through the center of the group. However, when an information limit of $a_{IL} = 4$ is imposed on the agent, the black agent, does not perceive all the individual agents in the group and as a result it is forced to reconsider its path mid-route. As a result, the irregular trail shown in Fig. 10c is obtained. However, in the same situation, when Group



(a) Initial Scenario

(b) Complete knowledge, without GBP

(c) Info limit of 4, without GBP

(d) Info limit of 4, with GBP

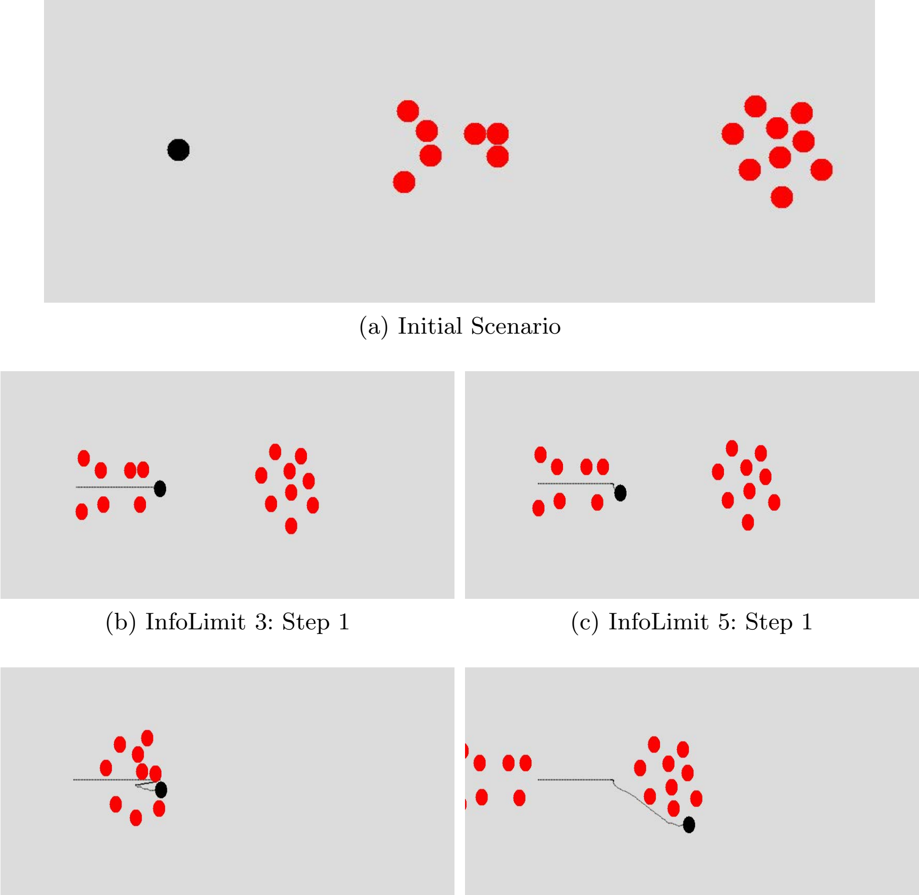
(e) Info limit of 1, with GBP

Fig. 10. Experiment 3: The necessity of Group Based Perception

Based Perception is used, the agent smoothly avoids the whole group (Fig. 10d). In fact, this smooth path is obtained for as low a limit as $a_{IL} = 1$ (Fig. 10e).

4.4 Effect of Filtering of Percept Information

The final experiment (Fig. 11) demonstrates the effect of filtering, i.e. having limits on the information processing capabilities of the agents. The scenario consists of an agent moving towards a collection of individuals (moving towards the agent) followed by a group of agents behind the set of individuals. In the first case we set an information limit of $a_{IL} = 5$ so that the agent is continually capable of perceiving a larger number of other agents and groups. In the second scenario we use a lower limit of $a_{IL} = 3$ such that the agent isn't initially



(a) Initial Scenario

(b) InfoLimit 3: Step 1

(c) InfoLimit 5: Step 1

(d) InfoLimit 3: Step 2

(e) InfoLimit 5: Step 2

Fig. 11. Experiment 4: Effect of filtering of percept information

capable of perceiving the group behind the individuals. Figure 11c shows how agents perceive the cluster that is farther away, even when there is an immediate collision to avoid. Figure 11e shows that the agent manages to move around this group because it had a head start in planning - i.e. , it considered the group early when avoiding collisions. In the second scenario we gave a much lower information limit, such that it could process a maximum of 3 or 4 percepts at any given time. Due to this, as seen in Fig. 11b, the agent cannot see beyond the immediate obstacles in front and does not prepare in advance to avoid the larger group. Once the agent finally perceives this group, it is too late to move around this group as it perceives the group as individuals and then moves through the group as in Fig 11d.

This experiment illustrates how small differences in the information limit can generate different forms of behavior in the agents[3]. Clearly the value of the limit is critical to behavior, we also propose that this limit will change with personal characteristics and the emotional state of the agents. In fact we feel that this varying limit of perception is an important factor for collisions in crowds, this is especially relevant in emergency egress scenarios where stress and collisions are critically important to safety planning. We plan to attempt to quantify this information limit through experimentation in future work.

5 Conclusion

In this paper, we have proposed a model of perception based on perceived information rather than spatial distance. We argue that this is a more appropriate model of human perception for crowd and egress simulation. We have illustrated the behavior of this system through experiments and have shown and argued that this creates more realistic group avoidance behavior. We also presented a perception model which incorporated the idea that humans have limited perception capacity such that they only process certain obstacles more relevant to collision avoidance, which in turn will result in a reduction in efficiency of collision avoidance. Critical to the model is the quantification of information limits and appropriate definitions of interest; we plan to conduct real world experiments to attempt to quantify these parameters.

A more naturalistic system like this is essential for the development of an accurate model of crowd evacuation in emergencies. In emergency situations, according to [27], humans start perceiving cues in the environment differently. In the future, we plan to extend this model by first adding more features to the measure of interestingness score and also by modeling different cues and their effect on the agent's information processing capabilities as suggested in [22]. The third criteria which we mentioned in section 3, i.e. the inherent interestingness of the object has not been elaborated on in this paper. Also, in this paper we have not implemented any memory for the agent. To accurately simulate virtual humans' and their motion, the fact that they can remember the positions of objects should also be taken into consideration. Virtual humans can also extrapolate the movement of agents that they have perceived previously but are not in their field of vision at the current time. In this paper, we have considered the effect that perception can have on cognition. However, as mentioned in papers like [14] there is also a reciprocal effect of cognition on perception where agents would turn towards objects of more interest, we plan to incorporate this in future versions of the model.

Acknowledgment. This research has been funded by the NTU SU Grant M58020019.

[3] Interestingly, the info limit of 3 and 5 correspond to Cowan's [5] finding that all humans can cognitively process only 3-5 chunks of information at any given time.

References

1. van den Berg, J., Patil, S., Sewall, J., Manocha, D., Lin, M.: Interactive navigation of multiple agents in crowded environments. In: 2008 Symposium on Interactive 3D Graphics and Games. University of North Carolina (2008)
2. Bonabeau, E.: Agent-based modeling methods and techniques for simulating human systems. In: Arthur M. Sackier Colloquium of the National Academy of Sciences, pp. 7280–7287. Icosystem Corporation, 545 Concord Avenue (2002)
3. Broadbent, D.E.: Applications of Information Theory and Decision Theory to Human Perception and Reaction. Progress in Brain Research 17, 309–320 (1965)
4. Courty, N., Marchand, E., Arnaldi, B.: A New Application for Saliency Maps: Synthetic Vision of Autonomous Actors. In: Proceedings of the 2003 International Conference on Image Processing, ICIP 2003 (2003)
5. Cowan, N.: The magical number 4 in short-term memory: A reconsideration of mental storage capacity. Behavioral and Brain Sciences 24(01), 87–114 (2001)
6. Epstein, J.M.: Agent-based computational models and generative social science. Complexity 4(5), 41–60 (1999)
7. Grillon, H., Thalmann, D.: Simulating gaze attention behaviors for crowds. Computer Animation and Virtual Worlds 20(2-3), 111–119 (2009)
8. Guy, S., van den Berg, J., Lin, M.: Geometric methods for multi-agent collision avoidance. In: Proceedings of the 2010 Annual Symposium on Computational Geometry, pp. 115–116. University of North Carolina, Utah (2010)
9. Guy, S.J., Chhugani, J., Curtis, S., Dubey, P., Lin, M., Manocha, D.: PLEdestrians: a least-effort approach to crowd simulation. In: SCA 2010: Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation. Eurographics Association, University of North Carolina (July 2010)
10. Guy, S.J., Chhugani, J., Kim, C., Satish, N., Lin, M., Manocha, D., Dubey, P.: ClearPath: highly parallel collision avoidance for multi-agent simulation. In: SCA 2009: Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation. ACM Request Permissions (August 2009)
11. Guy, S.J., Lin, M.C., Manocha, D.: Modelling Collision Avoidance Behavior for Virtual Humans. In: 9th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2010), Toronto, Canada, pp. 575–582 (May 2010)
12. Helbing, D., Farkas, I., Vicsek, T.: Simulating Dynamical Features of Escape Panic. Physical Review E cond-mat.stat-mech (September 2000)
13. Helbing, D., Molnar, P.: Social Force Model for Pedestrian Dynamics. Physical Review E cond-mat.stat-mech, 4282–4286 (1995)
14. Hill, R.W.: Modeling Perceptual Attention in Virtual Humans. In: Computer Generated Forces and Behavioral Representation, Orlando, pp. 1–11 (May 1999)
15. Hochberg, J., McAlister, E.: A quantitative approach, to figural "goodness". Journal of Experimental Psychology 46(5), 361–364 (1953)
16. Itti, L., Koch, C.: Computational modeling of visual attention. Nature 2, 194–203 (2001)
17. Kamphuis, A., Overmars, M.H.: Finding paths for coherent groups using clearance. In: Proceedings of the 2004 ACM SIGGRAPH Symposium on Computer Animation, Utrecht University (2004)
18. Kim, Y., van Velsen, M., Hill Jr., R.W.: Modeling Dynamic Perceptual Attention in Complex Virtual Environments. In: Panayiotopoulos, T., Gratch, J., Aylett, R.S., Ballin, D., Olivier, P., Rist, T. (eds.) IVA 2005. LNCS (LNAI), vol. 3661, pp. 266–277. Springer, Heidelberg (2005)

19. Klein, W., Köster, G., Meister, A.: Towards the Calibration of Pedestrian Stream Models. In: Wyrzykowski, R., Dongarra, J., Karczewski, K., Wasniewski, J. (eds.) PPAM 2009, Part II. LNCS, vol. 6068, pp. 521–528. Springer, Heidelberg (2010)
20. Klein, W., Köster, G., Meister, A.: Towards the Calibration of Pedestrian Stream Models. In: Wyrzykowski, R., Dongarra, J., Karczewski, K., Wasniewski, J. (eds.) PPAM 2009, Part II. LNCS, vol. 6068, pp. 521–528. Springer, Heidelberg (2010), http://dl.acm.org/citation.cfm?id=1893586.1893650
21. Klüpfel, H., Schreckenberg, M., Meyer-König, T.: Models for crowd Movement and egress Simulation. Traffic and Granular Flow (2005)
22. Kuligowski, E.D.: The Process of Human Behavior in Fire. Tech. rep., National Institute of Standards and Technological (May 2009)
23. Luo, L., Zhou, S., Cai, W., Low, M.Y.H., Tian, F., Wang, Y., Xiao, X., Chen, D.: Agent-based human behavior modeling for crowd simulation. Computer Animation and Virtual Worlds 19(3-4), 271–281 (2008)
24. Miller, G.: The magical number seven, plus or minus two: some limits on our capacity for processing information. Psychological Review (1956)
25. Okazaki, S., Matsushita, S.: A study of simulation model for pedestrian movement with evacuation and queuing. Engineering for Crowd Safety (1993)
26. O'Reagan, J.K., Rensink, R.A., Clark, J.J.: Change-blindness as a result of 'mud-splashes'. Nature 398, 34 (1999)
27. Ozel, F.: Time pressure and stress as a factor during emergency egress. Safety Science 38, 95–107 (2001)
28. Pettré, J., Ondřej, J., Olivier, A.H., Cretual, A., Donikian, S.: Experiment-based modeling, simulation and validation of interactions between virtual walkers. In: SCA 2009: Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation. ACM Request Permissions (August 2009)
29. Reynolds, C.: Flocks, herds and schools: A distributed behavioral model. Computer Graphics 21(4), 25–34 (1987)
30. Song, Q., Kasabov, N.: Ecm - a novel on-line, evolving clustering method and its applications. In: Posner, M.I. (ed.) Foundations of Cognitive Science, pp. 631–682. The MIT Press (2001)
31. Still, G.K.: Crowd dynamics. Ph.D. thesis, University of Warwick, University of Warwick, Department of Mathematics (August 2000)
32. Triesch, J., Ballard, D.H., Hayhoe, M.M., Sullivan, B.T.: What you see is what you need. Journal of Vision 3, 1–9 (2003)
33. Whittle, M.: Gait Analysis. An Introduction, 4th edn. Butterworth Heinemann (December 2006)

Experiments in Artimetrics: Avatar Face Recognition*

Roman V. Yampolskiy[1], Gyuchoon Cho[1],
Richard Rosenthal[1], and Marina L. Gavrilova[2]

[1] Dept. of Computer Engineering and Computer Science, University of Louisville, USA
[2] Dept. of Computer Science, University of Calgary, Canada
{roman.yampolskiy,g0cho001,rarose03}@louisville.edu,
marina@cpsc.ucalgary.ca

Abstract. The Artimetrics, a field of study that identifies, classifies and authenticates virtual reality avatars and intelligent software agents, has been proposed as a tool for fighting crimes taking place in virtual reality communities and in multiplayer game worlds. Forensic investigators are interested in developing tools for accurate and automated tracking and recognition of avatar faces. In this paper, we evaluate state of the art academic and commercial algorithms developed for human face recognition in the new domain of avatar recognition. While the obtained results are encouraging, ranging from 53.57% to 79.9% on different systems, the paper clearly demonstrated that there is room for improvement and presents avatar face recognition as an open problem to the pattern recognition and biometric communities.

Keywords: artimetrics, avatar, biometric, face, recognition, robot, synthesis.

1 Introduction

Quick investigation of the Second Life (SL) virtual world shows that it is populated by organizations posing security risks, including international terrorist groups associated with al-Qaeda and local groups of radicals such as Second Life Liberation Army [28]. Similar to the 9/11 terrorists who practised flying planes on simulators in preparation for their deadly attack on civilian buildings and the Pentagon, security experts believe some of the people behind the SL virtual terror campaign are real-world terrorists who are rehearsing for future strikes against non-virtual targets [22, 16]. Terrorist organizations are well known for sending their recruits to training camps in Pakistan and Afghanistan, but with the US forces present on the ground in those countries new training options being investigated by the terrorist networks include training in virtual worlds. Virtual worlds can be used to create an exact replica of a real world target and utilized to rehearse an entire attack online, including

* This paper is based on a previous conference report "Evaluation of Face Detection and Recognition Algorithms on Avatar Face Datasets" presented at International Conference on Cyberworlds (CW2011), Banff, Alberta, Canada. October 4-6, 2011 by Roman V. Yampolskiy, Gyuchoon Cho, Richard Rosenthal and Marina L. Gavrilova.

M.L. Gavrilova and C.J.K. Tan (Eds.): Trans. on Comput. Sci. XVI, LNCS 7380, pp. 77–94, 2012.

monitoring the response and ramifications [28]. Additionally, virtual worlds are extremely attractive for the run-of-the-mill criminals interested in conducting identity theft, fraud, tax evasion, illegal gambling and other traditional crimes in the new "Wild Wild West" known as the World Wide Web. As the economies of virtual communities continue to grow into billions of real dollars, any instability or terrorist activity in virtual communities may have a direct impact on the real world economy, similar to the aftershock created by the 9/11 attack [22, 16].

This makes researchers and officials very concerned over the unprecedented growth of online virtual communities populated by avatars. Such communities are creating security vulnerabilities which "offer the opportunity for religious/political extremists to recruit, rehearse, transfer money, and ultimately engage in information warfare or worse with impunity" [29]. The seriousness of the situation is such that the U.S. Congress has held hearings about the potential use of virtual worlds for money laundering operations by Al-Qaida and other terrorist groups [33]. The government's growing concerns are likely to make online virtual worlds the next frontier in the battle over the limits on the government's right to improve security via data collection and analysis and the surveillance of commercial computer systems [29]. The wide availability of virtual worlds has created new challenges for government surveillance because commercial services do not have to keep records of communications among virtual world avatars, or even record their true identities [22, 16]. As more people join virtual communities and create avatars, it will become more difficult to identify potential criminals and terrorists [22]. According to R. O'Harrow: "As in the real world, one of the central difficulties is establishing the identity of individuals" [29].

It is worth spending a few moments to motivate our work by analyzing news reports of crimes reported to have been committed in the virtual communities. The examples given are by no means exhaustive as any type of real crime has a virtual equivalent [22]:

Virtual Rape – "Brussels police have begun an investigation into a citizen's allegations of rape in Second Life" [24].

Theft of Virtual Property – "Netherlands Teen Sentenced for Stealing Virtual Goods" [13].

Virtual Prostitution, Strip Clubs and Pornography – "Escorts, the Second Life equivalent of phone-sex operators or prostitutes, are quite common in Second Life" [42]**.**

Virtual Gambling – "FBI checks gambling in Second Life virtual world" [31].

Virtual Money Laundering – "Second Life and Other Online Sites Targeted by Criminals" [38].

Virtual Fraud – "…the "bank" vanished, and depositors say their money did, too**"** [40].

Identity Theft – "Second Life charges for real names, increases identity theft risk" [41].

Illegal Content (Child Porn) – "Second Life 'child abuse' claim" [4].

Virtual Murder – "Woman arrested after virtual murder" [23].

Virtual Counterfeit Goods and Intellectual Property – "As … more companies open virtual stores, the problem of trademark violations and counterfeit products takes on a whole new form" [1].

Stalking, Cyber Bullying and Harassment – “People who are bullied at work or in school may not be able to escape their tormentors in the virtual world, researchers warned today” [25].

Due to the growing number of virtual crimes, avatar tracking and recognition is becoming an important problem which security professionals and forensic investigators face every day. To address that problem authors have proposed a new field of research: ***Artimetrics*** – a field of study that identifies, classifies and authenticates virtual reality avatars, robots and intelligent software agents [45, 49, 48, 16, 22, 46, 50, 2, 7, 17, 47, 12, 3, 26].

2 Survey of Robots and Avatars

This section presents a quick overview of different types of robots and avatars and identifies those which are in the domain of artimetrics research. Here we are interested in robots which could be placed into the following general categories [32]:

- **Industrial Robots:** Robots optimized for repetitiveness, accuracy, endurance, speed, and reliability and utilized on fully automated production lines.
- **Mobile Robots:** They are used for transporting material in hospitals, container ports, or warehouses.
- **Tele Robots:** Robots which are remotely controlled by a human operator. Many military and space exploration robots fall under this category.
- **Service Robots:** Robots used for professional tasks such as underwater exploration or cleaning up hazardous waste.
- **Personal Robots:** Automated caregivers for disabled and the elderly, pet robots, house cleaning and entertainment robots.

In the context of artimetrics we are most interested in humanoid and socially interactive robots [14], [36] as subcategory of embodied robots. Social robots are said to exhibit the following human-like characteristics [14]: express and perceive emotions, communicate via high-level language, recognize model of other agents, establish social relationships, use natural cues such as gaze or gestures, exhibit distinctive personality traits, and ability to develop social competencies. All of the above characteristics can be used in the context of behavioral artimetrics [6, 39].

From the robots’ body design point of view, four broad categories of hardware robots exist [14]:

- Anthropomorphic – human like;
- Zoomorphic – animal like;
- Caricatured – cartoon character like;
- Functional – optimized for a specific task (ex. handle bars, cargo space, wheels, etc.)

In case of identically built industrial or service robots, focusing on individual facial features or body parts is not helpful in recognition task, however behavioral profiling is still possible by a) individual observation of robot actions and interactions with humans, other robots or physical environment, or by covert observation of combined behavior of team of industrial robots – looking for anomalies, operational delays, abnormal behavior etc. with the goal of recognizing faulty, compromised or malicious identity. For industrial robots recognizing "normal" expected behavior could be extremely simple: such a robot might perform a simple sequence of elementary operations, repeated with specific time interval. Thus behavior can be classified as NORMAL or ABNORMAL. For a personal household robot (such as Rumba, for example) behavior could be more complex, and depend on physical space boundaries, environment, operational settings etc. For social robots the range of interactions, activities and patterns can be very complex and resemble humans - thus behavioral biometric models used in human recognition (gait, voice, speech pattern, etc.) can be used in this case [20, 19].

Now, let us look at avatar classification. According to [37], the following types of avatars exist based on preferences and behavior of its human creator:

Odd/shocking avatars are unusual, strange, or bizarre; Abstract avatars may be represented by abstract art; Billboard avatars are announcements of some kind; Lifestyle avatars depict a significant aspect of a person's life; Matching avatars are designed to accompany each other; Clan avatars are worn by members of the same social group; Animated avatars contain motion; Animal avatars are typically associated with person's pets or self-association with nature; Cartoon avatars are based on famous drawn characters; Celebrity avatars tend to follow trends in popular culture; Evil avatars are scary looking; Real Face avatars are uploaded pictures of the actual users; Idiosyncratic avatars are strongly associated with a specific user; Positional avatars are designed by the member to be placed into specific locations; Power avatars are symbols of omnipotence; Seductive avatars are scantily clothed figures.

Identification of such avatars can be carried out through analysis of their appearance, attributes, behavioral patterns, frequency and type of changes, using a combination of traditional image pattern recognition techniques and biometric behavioral identifiers. Classifying further the types of behaviors that avatars might exhibit as well as drawing parallels between avatar and human behavior are mainly unexplored problems solving which can assist significantly in the task of avatar authentication.

Robot and avatar "faces", on the other hand, represent the fundamental physical artimetric trait. Robot facial development and research have a relatively long history and are approaching human-like representations coupled with facial expression and realistic emotion representation [6, 39]. While the variety of different robot "faces" is truly astonishing as can be seen from Figure 14 [37], only those robots with a face very similar to human faces would be recognizable by the state of the art face recognition systems without a complete modification of the algorithm. Consequently, it is extremely important to design and develop systems capable of recognizing faces corresponding to more types of robots which will likely begin to appear in ever greater numbers in our workplaces, homes and public locations such as museums, libraries and government buildings. Comparing variability of emotions that can be exhibited by robot faces to human faces is at present another open area of research.

In this paper, in order to establish a baseline for future research in avatar face recognition we conducted experiments aimed at the evaluation of current academic and commercial software for human face recognition on avatar face datasets. The paper also proposes some improvements to the current models.

We begin our analysis by looking at VeriLook SDK released by NeuroTechnology which is intended for biometric systems developers. It allows quick production of biometric software using functions available in the VeriLook library [27]. We conclude our experiments with testing and analysis of Picasa, a state-of-the-art commercial software from Google [43].

3 Advanced Face Localization

The utilized algorithm implements Advanced Face Localization (AFL). It was tested on a PC with Intel Core 2 processor running at 2.66 GHz. VeriLook SDK enables both one-to-one and one-to-many matching modes, for processing and identification, with a comparison speed of 100,000 faces per second [27]. The algorithm also has built in tolerance for variations in roll, pitch and yaw (see Figure 1 and Table 1). Head roll (tilt) is within ±180 degrees (configurable); ±15 degrees are recommended as the fastest setting which is usually sufficient for most near-frontal face images [27]. Both head pitch (nod) and head yaw (bobble) are within ±15 degrees from frontal position. All face templates need to be loaded into the computer's RAM before identification can begin; consequently, the maximum face template database size is restricted by the amount of available RAM [27].

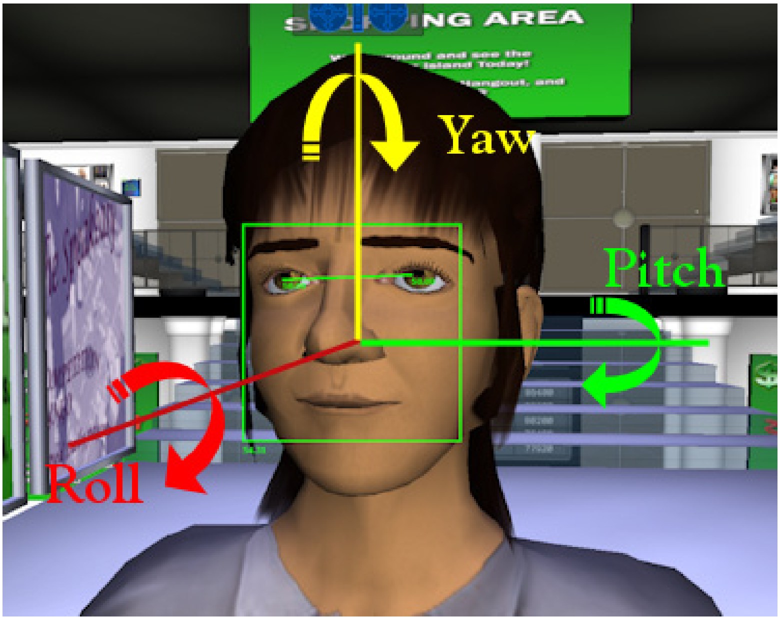


Fig. 1. Tolerances of Roll, Pitch, and Yaw

Table 1. Results of Reliability Test Using FRGC Database.

Reliability Test Using FRGC database	Maximized template size	Medium Temple Size	Minimum Template Size
Detection time for all faces in a frame (±15º head roll tolerance)	10 milliseconds		
Detection time for all faces in a frame (±180º head roll tolerance)	135 milliseconds		
Single face template extraction time	111	62	31
Matching speed (face records per second)	24,000	44,000	200,000
Template size in database (bytes)	20,440	11,368	2,296

Face template extraction is done after the faces are detected in a particular frame [27]. The extraction time does not depend on image size, but only on the defined template size. The probe template is defined to contain large face record, whereas the gallery templates can contain small, medium or large face records. Template size increases proportionately to the number of face records stored in the template [27].

Face Recognition Grand Challenge (FRGC), is a dataset that contains 50,000 images for evaluating face recognition technology [5]. It may also be used for structural feature detection, such as eyes and the position of several important points such as nose tip, and the inner corners of the eyes [5]. In the FRGC database, the landmark information is stored in the XML format. Since the images in the databases are of various scales, a Root Mean Square (RMS) error analysis could be used. In order to calculate a meaningful measure, an algorithm due to Beumer et al. was utilized [5]:

a) Translate, scale and rotate the ground truth data so that the eye landmarks are on a horizontal line.
b) Register the shape found to the respective ground truth shape.
c) Calculate the Euclidian distance between each landmark and its ground truth.
d) Remove the bias created by the different labeling policies in the databases.
e) Calculate the RMS value of the difference between the found shape and the ground truth shape, obtaining percentage of the inter-ocular distance [5].

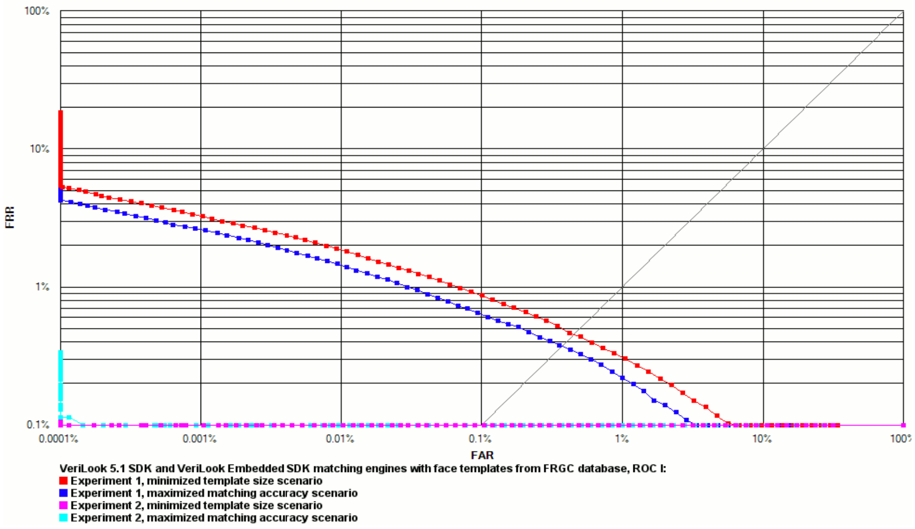


Fig. 2. VeriLook algorithm on face templates from FRGC database

Experiment 1 and Experiment 2 were performed according to the FRGC protocol. Experiment 1 measured performance of the recognition of frontal facial images taken under controlled illumination. The biometric samples in the target and query sets consist of a single controlled still image at high resolution. Experiment 2 is designed to examine the effect of multiple still images on performance (see Figure 2). The biometric samples in the target and query sets are composed of the four controlled images of each subject.

4 Color Structure Descriptor

Color Structure Descriptor (CSD) is based on color histogram but, the main goal of CSD is image to image matching [9]. In some cases when our system recognizes an avatar's face it is actually a false positive. This descriptor is used to overcome false results by using pair-wise distance between query image and a set of similar images. The CSD is implemented in Matlab 2009. CSD in our research is used only for still image retrieval, in order to rearrange queried images. CSD obtains a local color structure by utilizing a structuring element based on several image sub-samples [9]. The CSD is a generalization of the color histogram that captures the color distribution characteristics of an image. It is defined in the Hue Max Min Diff (HMMD) color space using non-uniform color quantization to up to 256 colors [21]. When our system extracts an avatar's face image, the CSD will quantize the image using a median-cut algorithm. The structuring element, *8 x 8* in Figure 3, calculates values of each pixel by visiting all locations of the image, then incrementing each color value in CSD's bin.

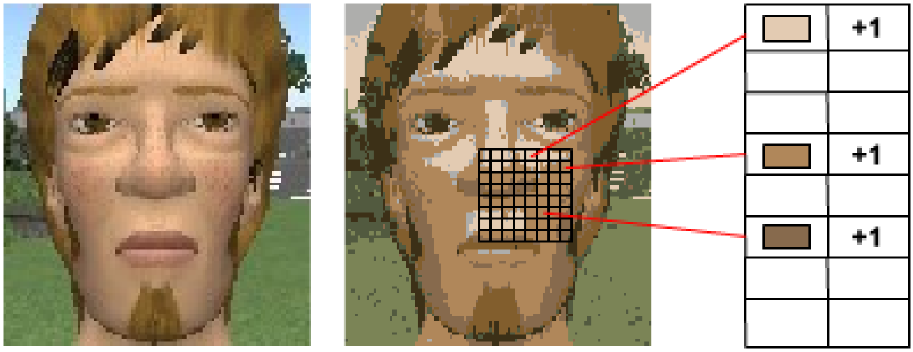


Fig. 3. Color structure descriptor

HMMD is based on hue, shade, tone, tint and brightness of color, along with a quantizing method (see Figure 4) [21]. The HMMD model treats the colors adjacent to a given color as the neighboring colors, and suses the indices of those neighboring colors as the partial values according to their respective distances in color space [21].

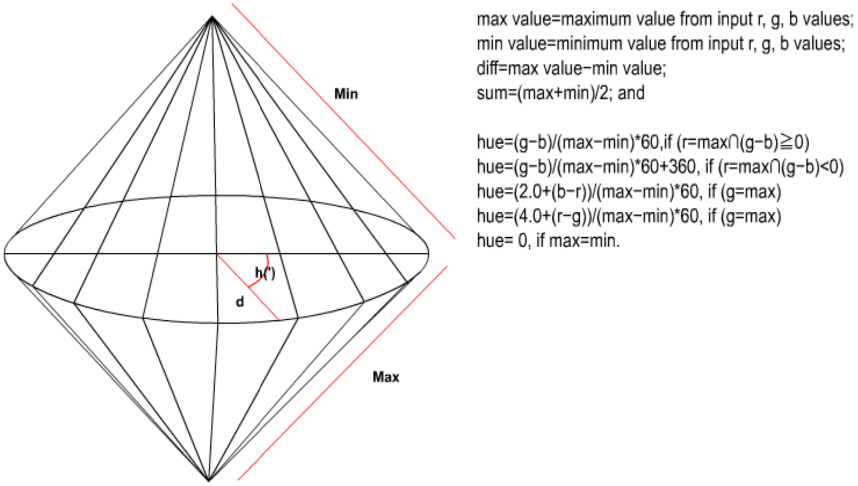

Min
h(°)
d
Max
max value=maximum value from input r, g, b values;
min value=minimum value from input r, g, b values;
diff=max value−min value;
sum=(max+min)/2; and
hue=(g−b)/(max−min)*60,if (r=max∩(g−b)≧0)
hue=(g−b)/(max−min)*60+360, if (r=max∩(g−b)<0)
hue=(2.0+(b−r))/(max−min)*60, if (g=max)
hue=(4.0+(r−g))/(max−min)*60, if (g=max)
hue= 0, if max=min.

Fig. 4. A color model HMMD [21]

5 Edge Histogram Descriptor

Histograms are frequently used to represent global features of an image. They are invariant to both translation and rotation operations, and normalization of the historgram leads to scale invariance [44]. Consequenlty, histogram is a great descriptor for the indexing and retrieval of images from a dataset. An edge historgram descriptor reflects the underlying directionality and brightness in the image [44]. The main objective of using EHD in our experiments is to find the same avatar's faces by matching their characteristics, such as hair style, or the shape of their eyebrows, or face. The input image is divided into *4 x 4* sub-images (as in Figure 5). All 16 image partitions converge to the same dimensions, regardless of the size of the input image.

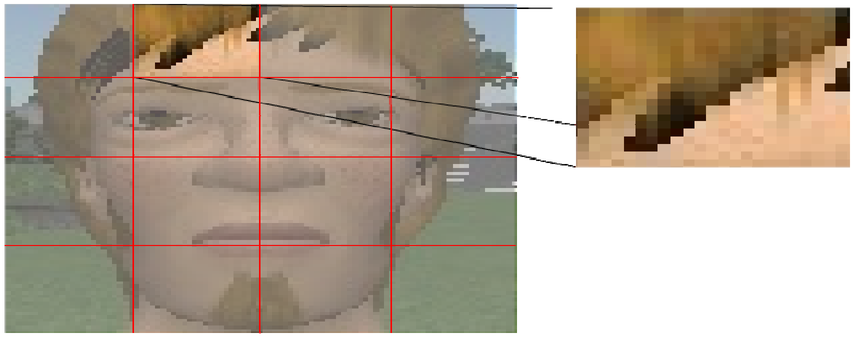


Fig. 5. Sub-image grid of input image

Edges in the sub-images can be grouped into five types: vertical, horizontal, 45 degree diagonal, 135 degree diagonal and non-directional edges, like the one shown in Figure 6 [44]. The histogram for each sub-image represents the relative frequency of occurrence of the five types of edges in the corresponding sub-image [44]. EHD produces 80 bins because there are 16 sub-images in the input image and 5 edge types.

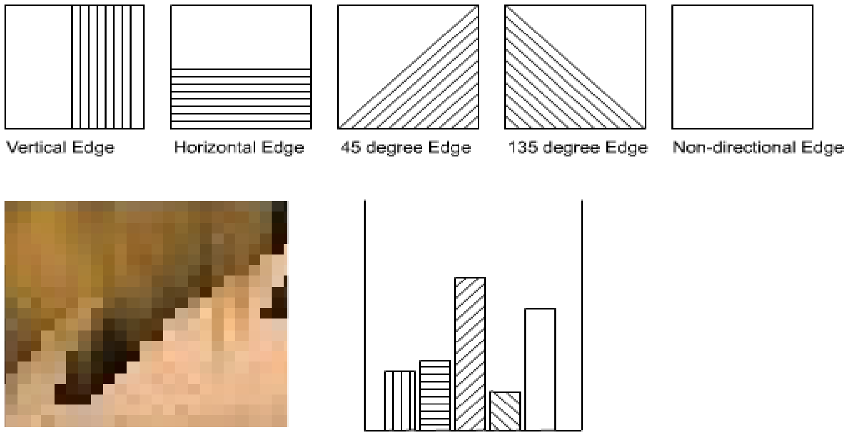


Fig. 6. Five types of edge bins for sub-image and its histogram [44]

The similarity matching is computed for the two image histograms by using the following formula [44]:

$$D(A,B) = \sum_{i=0}^{79} |A[i] - B[i]| \tag{1}$$

6 VeriLook Results

The system searches avatar faces and enrolls them into a database using SQLite. We used 700 avatar images with each avatar represented by ten different head shots. Then given a query image our system retrieves highest matched images from the dataset, with CSD and EHD utilized to improve the accuracy. An avatar face is considered to be correctly recognized if one of the other nine head shots for this avatar is returned by the system as a top match. The percentage of successfully recognized face images is 79.9%; 559 out of 700. Total processing time is 1259.33 seconds. The diagram in Figure 7 shows the overall system's work flow.

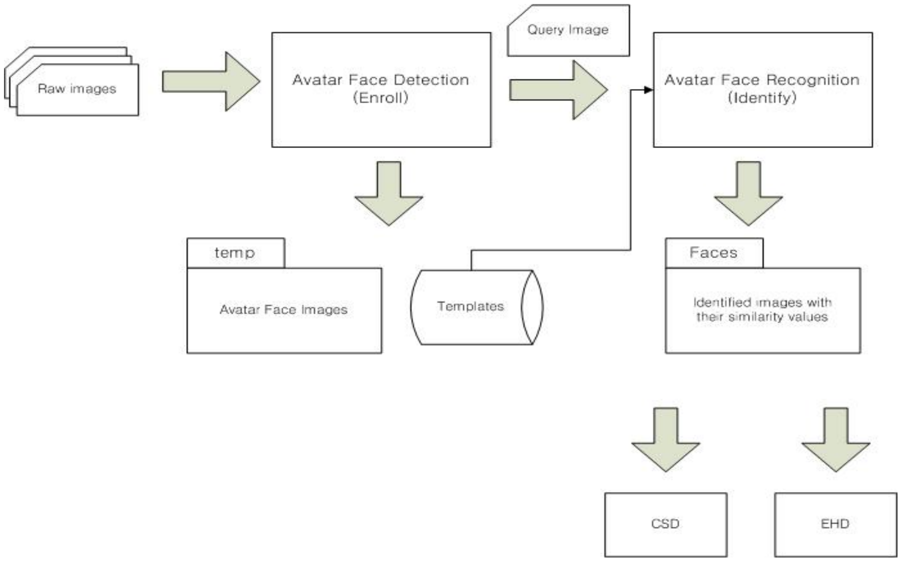

Raw images
Avatar Face Detection (Enroll)
Query Image
Avatar Face Recognition (Identify)
temp
Avatar Face Images
Templates
Faces
Identified images with their similarity values
CSD
EHD

Fig. 7. System's work flow

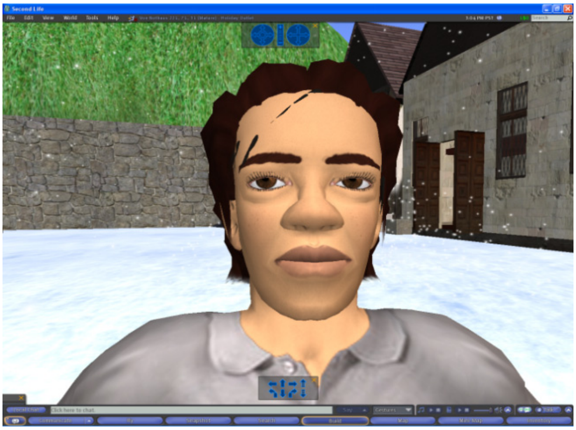
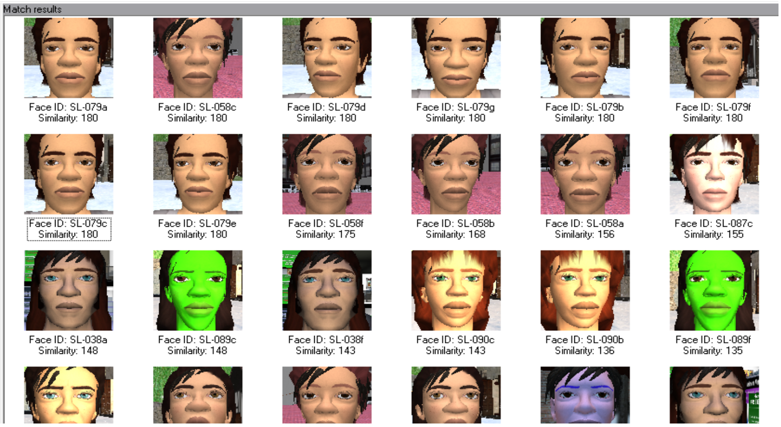


Fig. 8. Query image

Without descriptors, some query image (see Figure 8) may lead to retrieval of incorrect images, based on a higher matching score. For example the second image, SL-058c (in Figure 10, second from left in the top row) was an incorrect match. However, CSD improves image retrieval based on query image's color (see Figure 11). Theoretically, we expected that EHD would return results based on face shape similarity, but since most avatars have similar face shapes, EHD could not improve results compared to CSD (see Figure 12). Overall, using a descriptor can improve results, but it may require more processing time. If an avatar's face has a unique shape, it may be reliably recognized without a need for use of special descriptors, as seen in Figure 13.



```
Match results
Face ID: SL-079a    Face ID: SL-058c    Face ID: SL-079d    Face ID: SL-079g    Face ID: SL-079b    Face ID: SL-079f
Similarity: 180     Similarity: 180     Similarity: 180     Similarity: 180     Similarity: 180     Similarity: 180
Face ID: SL-079c    Face ID: SL-079e    Face ID: SL-058f    Face ID: SL-058b    Face ID: SL-058a    Face ID: SL-087c
Similarity: 180     Similarity: 180     Similarity: 175     Similarity: 168     Similarity: 156     Similarity: 155
Face ID: SL-038a    Face ID: SL-089c    Face ID: SL-038f    Face ID: SL-090c    Face ID: SL-090b    Face ID: SL-089f
Similarity: 148     Similarity: 148     Similarity: 143     Similarity: 143     Similarity: 136     Similarity: 135
```

Fig. 9. Results without descriptors

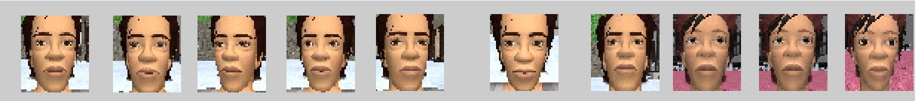


Fig. 10. Results using CSD with 64 bits

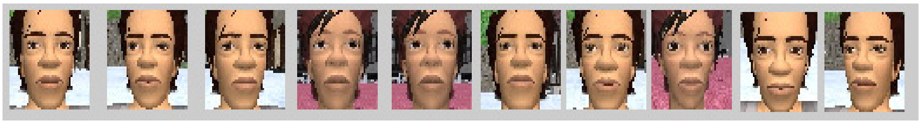
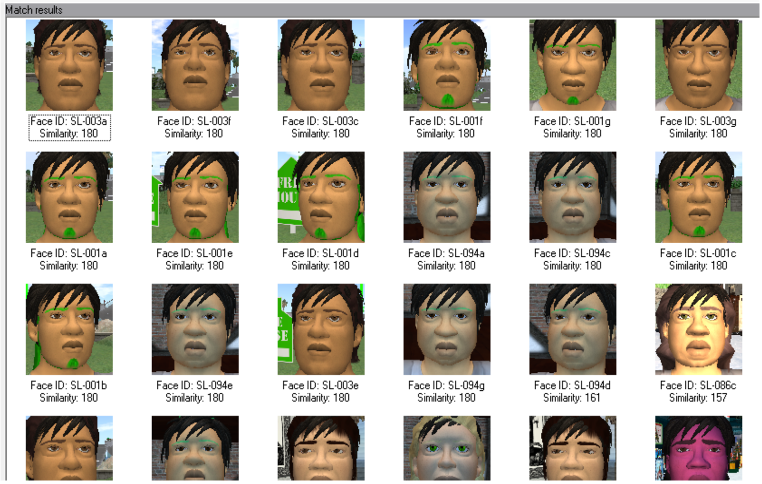


Fig. 11. Results using EHD with 80 bins



```
Match results
Face ID: SL-003a    Face ID: SL-003f    Face ID: SL-003c    Face ID: SL-001f    Face ID: SL-001g    Face ID: SL-003g
Similarity: 180     Similarity: 180     Similarity: 180     Similarity: 180     Similarity: 180     Similarity: 180
Face ID: SL-001a    Face ID: SL-001e    Face ID: SL-001d    Face ID: SL-094a    Face ID: SL-094c    Face ID: SL-001c
Similarity: 180     Similarity: 180     Similarity: 180     Similarity: 180     Similarity: 180     Similarity: 180
Face ID: SL-001b    Face ID: SL-094e    Face ID: SL-003e    Face ID: SL-094g    Face ID: SL-094d    Face ID: SL-086c
Similarity: 180     Similarity: 180     Similarity: 180     Similarity: 180     Similarity: 161     Similarity: 157
```

Fig. 12. Query results without descriptors on a uniqe face

7 Picasa Experiments and Results

Picasa is an image editor and viewer, originally created by Idealab in 2002 and purchased by Google in 2004. It features an automated face recognition and classification software allowing, for automated sorting of pictures "by person." The face recognition software was originally developed by Neven Vision and acquired by Google in 2006. Neven Vision's face recognition algorithm was among the top finalists in FERET 1997 and FRVT 2002 independent tests comparing face recognition software [43].

We have chosen to use Picasa because it utilizes state-of-the-art face recognition software produced by one of the industry leaders. In order to perform the experiment to establish baseline capability of Picasa in recognizing avatar faces a group of avatars was collected from the virtual community - Second Life [30]. There were 22 avatars in the set, and 20 pictures of each. These were then divided into a control and experiment group – 60% (12 of each avatar) of the images were put into the control group, and the other 40% (8 of each avatar) were put into the test group. Then the program was "trained" on the control group – this process consisted of loading the images into Picasa, letting it run its facial recognition comparisons on them, naming the groups it found, confirming or denying its "suggested faces," until each of the 22 avatar's had its own named group with its designated 12 pictures (see Figure 9). During this process, however, it was found that Picasa was not able to recognize all of the faces. Out of the whole control group of 251 images, it was only able to recognize 209 (83.27%). This was most likely due to the camera angle of the pictures.

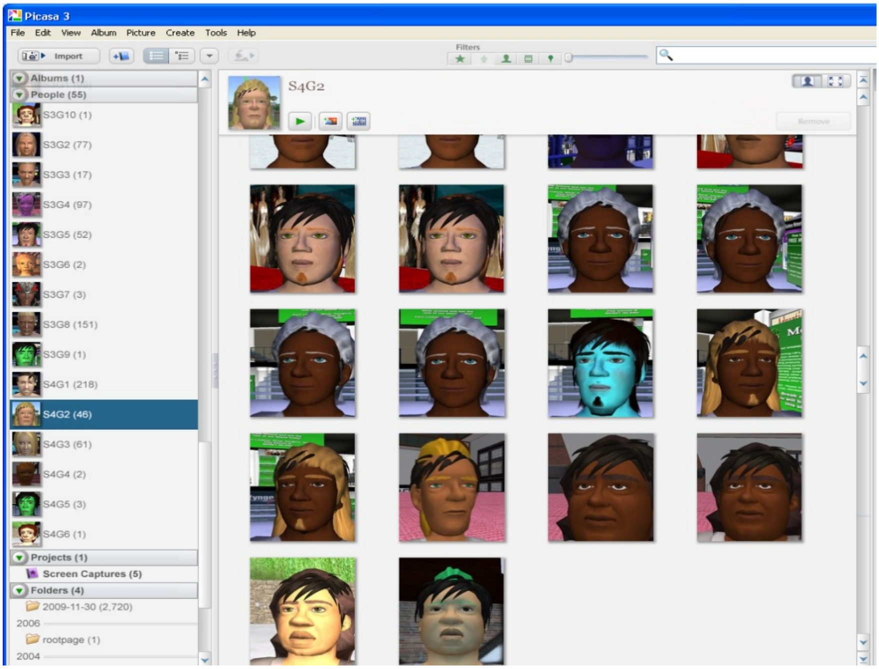


Fig. 13. Picasa software classifying avatar faces

The next step was to add in the test group images. Once loaded into Picasa, it ran its facial recognition algorithms on the faces. As with the control group, not all of the faces were recognized, however it still managed to find 95.24% of the faces. It was then recorded whether or not the images from the experiment group were assigned to the correct name groups created from the control set. Only counting the number of faces it correctly matched, Picasa had an overall accuracy rate of 59.92%; that is, it correctly grouped 50 out of the 84 faces in the test group. There were very few false positives; only five incorrect guesses. Subtracting the number incorrect from the number it got correct, it comes to an accuracy rate of 53.57%. Some of the test-set's images were not fully rendered, specifically, 5 of the 84 faces. However, it doesn't seem to have impacted the results, as 4 out of the 5 improperly rendered images were correctly recognized and grouped, despite the problem.

Fig. 14. A visual survey of robots by Frank Hegel (reproduced with permission) [18]

8 Future Directions: Multimodal Artimetrics

Biometric system based solely on a single biometric may not always identify the entity (human or robot) in the most optimal or precise way. The problem is common for human and robot authentication: some robots might not possess certain trait that is being recognized, i.e. industrial robots might have similar facial features but different voice, gait or behavior. Thus, multibiometric system research is emerging as a trend which helps to overcome limitations of a single biometric solution [35]. This is especially useful in the presence of complex patterns, conflicting or misleading behavior, abnormal data samples, intended or accidental mischief etc. A reliable and successful multibiometric system normally utilizes an effective fusion scheme to combine the information presented by multiple matchers.

Over the last decade, researchers tried different biometric traits with sensor, feature, decision, and match score level fusion approaches to enhance the security of a biometric system [35], thus enhancing security and performance of authentication system. Multimodal biometric approaches improve overall system accuracy and address issues of non-universality, spoofing, noise, and fault tolerance. Multimodal biometrics or Multibiometrics can referrer to a number of different approaches such as [34]:

- **Multi-Sensor** – employ multiple sensors to capture a single biometric trait.
- **Multi-Algorithm** – utilize a number of feature extraction or matching algorithms on the same data.
- **Multi-Instance** – utilize data from multiple instances of the same trait such as multiple fingerprints or two irises
- **Multi-Sample** – collect multiple instances of the same trait via a single sensor.
- **Multi-Modal** – utilize multiple biometric traits (ex. face and fingerprint and voice).

Most common approach in multibiometrics currently relies on multi-modal system, where numerous strategies for decisions making are employed. The most successful ones are based on rank-level, decision level and match score level fusion [15]. Other approaches (multi-sensor, multi-algorithm, multi-instance, multi-sample) are not as popular due to overhead associated with either multiple devices, multiple samples stored in database or extra time required to run different algorithms.

First paper combining face and fingerprint human identification as a true multi-modal system was based on match level fusion introduced by A. Jain. Authors of this article postulate that in a similar manner, combining behavioral and physical artimetrics in robot authentication or in the virtual worlds can be utilized as a part of a Physoemotional Artimetric system which is a multimodal system. In addition, another concept of Multi-Dimensional system, crossing over between virtual and real world, can be explored. The multi-dimensional authentication is the visual authentication of avatar through its creator authentication, and vice versa. Research in this domain is emerging, with a number of projects being conducted at BTLab, University of Calgary. No final published results however are available at the time of this article preparation.

While multi-modal system research became very popular over last few years, it carries certain challenges. One is amount of information that needs to be processed and as such associated technological and management challenges of obtaining, securely storing and accessing multiple databases. Another is increased cost of developing and maintaining such a system and slightly increased processing time, mainly due to the addition of a fusion module. Finally, in the presence of multi-source data processing and decision making, certain dimensionality reduction techniques are necessary to ensure real-time system performance.

9 Conclusions

This paper represents one of the first contributions in the domain of Artimetrics – recognition of non-biological agents. Specifically, we report results of evaluating state-of-the-art academic and commercial algorithms developed for human face recognition in the new domain of avatar recognition. Obtained accuracy rates range from 53.57% (Picasa) to 79.9% (VeriLook) and clearly demonstrate that the room for improvement exists. Consequently, the recognition of avatar faces is an open problem for the pattern recognition and biometric communities.

Potential directions for future Artimetrics research include the investigation of other visual and behavioral approaches to avatar security based, on the appearance of new characteristics and abilities in the virtual characters of tomorrow [16]. Even today it is possible to expand artimetrics beyond faces and vocabulary to intelligent software agents which mimic higher order human intelligence such as composing inspiring music [11], drawing beautiful paintings [10], and writing poetry [8]. As artificial intelligence and virtual reality research progresses, it will in turn stimulate creation of new security solutions to identity management across both human and artificial entity worlds [16].

The difficulties faced by artimetrics researchers are similar to those in the field of biometrics such as: chaotic and noisy environments, varied lighting conditions, subject occlusion, system spoofing by well-trained adversaries, etc. These are mainly unexplored areas of research. Improving system accuracy and authenticating both biological (human being) and non-biological entities (robots) are key future research directions, with the goal of providing a complete security solution to identity management.

References

[1] Adamczyk, S.: Second Life – A Whole New World of Trademark Infringement? ChillingEffects.org (October 27, 2006), `http://www.chillingeffects.org/weather.cgi?WeatherID=561`

[2] Ajina, S., Yampolskiy, R.V., Essoukri Ben Amara, N.: Evaluation of SVM Classification of Avatar Facial Recognition. In: Liu, D., Zhang, H., Polycarpou, M., Alippi, C., He, H. (eds.) ISNN 2011, Part III. LNCS, vol. 6677, pp. 132–142. Springer, Heidelberg (2011)

[3] Ali, N., Hindi, M., Yampolskiy, R.V.: Evaluation of Authorship Attribution Software on a Chat Bot Corpus. In: 23rd International Symposium on Information, Communication and Automation Technologies (ICAT 2011), Sarajevo, Bosnia and Herzegovina, October 27-29 (2011)

[4] Bbc, Second Life ’child abuse’ claim, BBC News (May 9, 2007), http://news.bbc.co.uk/2/hi/technology/6638331.stm

[5] Beumer, G.M., Tao, Q., Bazen, A.M., Veldhuis, R.N.J.: A Landmark Paper in Face Recognition. In: 7th International Conference on Automatic Face and Gesture Recognition (FGR 2006), April 2-6 (2006)

[6] Blow, A.M., Dautenhahn, K., Appleby, A., Nehaniv, C.L., Lee, D.: The art of designing robot faces: dimensions for human-robot interaction. In: 1st ACM SIGCHI/SIGART Conference on Human-Robot Interaction, Salt Lake City, Utah, USA, pp. 331–332 (2006)

[7] Boukhris, M., Mohamed, A.A., D’souza, D., Beck, M., Ben Amara, N.E., Yampolskiy, R.V.: Artificial human face recognition via Daubechies wavelet transform and SVM. In: 16th International Conference on Computer Games (CGAMES), Louisville, KY, pp. 18–25 (2011)

[8] Boyd-Graber, J.: Semantic Poetry Creation Using Lexicographic and Natural Language Texts, http://www.cs.princeton.edu/~jbg/documents/poetry.pdf (retrieved July 2, 2006)

[9] Buturovic, A.: MPEG 7 Color Structure Descriptorfor visual information retrieval project VizIR (December 8, 2010), http://cbvr.ims.tuwien.ac.at/user-documentation/csd-desc-1.pdf

[10] Cohen, H.: How To Draw Three People In a Botanical Garden (1988), http://crca.ucsd.edu/~hcohen/cohenpdf/how2draw3people.pdf

[11] Cope, D.: Virtual Music: Computer Synthesis of Musical Style. The MIT Press, Cambridge (2001)

[12] D’souza, D., Yampolskiy, R.V.: Avatar Face Detection Analysis using an Extended Set of Haar-like Features, Kentucky Academy of Science, Annual Meeting, Bowling Green, Kentucky, November 12-13 (2010)

[13] Feldmann, E.: Netherlands Teen Sentenced for Stealing Virtual Goods, PC World (October 23, 2008), http://www.pcworld.com/businesscenter/article/152673/netherlands_teen_sentenced_for_stealing_virtual_goods.html

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

[15] Gavrilova, M., Monwar, M.: Pattern Recognition and Biometric Fusion. In: Wang, P. (ed.) Pattern Recognition, Machine Intelligence and Biometrics (PRMIB), in memory of late Prof. King-Sun Fu. Springer (2011)

[16] Gavrilova, M.L., Yampolskiy, R.: Applying Biometric Principles to Avatar Recognition. In: International Conference on Cyberworlds (CW 2010), Singapore, October 20-22 (2010)

[17] Gavrilova, M.L., Yampolskiy, R.: Applying Biometric Principles to Avatar Recognition. In: Gavrilova, M.L., Tan, C.J.K., Sourin, A., Sourina, O. (eds.) Transactions on Computational Science XII. LNCS, vol. 6670, pp. 140–158. Springer, Heidelberg (2011)

[18] Hegel, F., Lohse, M., Wrede, B.: Effects of Visual Appearance on the Attribution of Applications in Social Robotics. In: 18th IEEE International Symposium on Robot and Human Interactive Communication, Toyama, Japan, September 27-October 2 (2009)

[19] Jain, A.K., Dass, S.C., Nandakumar, K.: Can soft biometric traits assist user recognition? In: Proceedings of SPIE Defense and Security Symposium, Orlando, FL (April 2004)

[20] Jain, A.K., Dass, S.C., Nandakumar, K.: Soft Biometric Traits for Personal Recognition Systems. In: Zhang, D., Jain, A.K. (eds.) ICBA 2004. LNCS, vol. 3072, pp. 731–738. Springer, Heidelberg (2004)

[21] Kim, H.J., Lee, J.S.: HMMD Color Space and Method for Quantizing Color Using HMMD Space and Color Spreading, United States Patent # 6633407, January 29 (1999)

[22] Klare, B., Yampolskiy, R.V., Jain, A.K.: Face Recognition in the Virtual World: Recognizing Avatar Faces, MSU Technical Report, MSU-CSE-11-2 (January 2011), http://biometrics.cse.msu.edu/Publications/Face/KlareYampolskiyJain_FRVirtualWorld_RecognizingAvatarFaces_TechRep2011.pdf

[23] Leach, B.: Woman arrested after virtual murder. Telegraph (October 25, 2008), http://www.telegraph.co.uk/news/newstopics/howaboutthat/3257876/Woman-arrested-after-virtual-murder.html

[24] Lynn, R.: Virtual Rape Is Traumatic, but Is It a Crime? Wired (May 4, 2007), http://www.wired.com/culture/lifestyle/commentary/sexdrive/2007/05/sexdrive_0504

[25] Mail, D.: Warning Over Cyber-Bullying in Second Life. Daily Mail (May 2007), http://www.dailymail.co.uk/sciencetech/article-458743/Warning-cyber-bullying-Second-Life.html

[26] Mohamed, A., Yampolskiy, R.V.: An Improved LBP Algorithm for Avatar Face Recognition. In: 23rd International Symposium on Information, Communication and Automation Technologies (ICAT 2011), Sarajevo, Bosnia and Herzegovina, October 27-29 (2011)

[27] Neurotechnology, VeriLook SKD, http://www.neurotechnology.com/vl_sdk.html?gclid=CPOWz8OB3aUCFeYD5QodOHDe2A

[28] O'brien, N.: Spies watch rise of virtual terrorists. The Australian (July 31, 2007), http://www.news.com.au/spies-watch-rise-of-virtual-terrorists/story-e6frfkp9-1111114075761

[29] O'harrow, R.: Spies' Battleground Turns Virtual, The Washington Post (February 6, 2008), http://www.washingtonpost.com/wp-dyn/content/article/2008/02/05/AR2008020503144.html

[30] Oursler, J.N., Price, M., Yampolskiy, R.V.: Parameterized Generation of Avatar Face Dataset. In: 14th International Conference on Computer Games: AI, Animation, Mobile, Interactive Multimedia, Educational & Serious Games, Louisville, KY (2009)

[31] Pasick, A.: FBI checks gambling in Second Life virtual world. Reuters (April 4, 2007), http://www.reuters.com/article/idUSHUN43981820070405

[32] Putatunda, R.: Types of Robots, http://www.buzzle.com/articles/types-of-robots.html (retrieved January 2, 2010)

[33] Reuters, A.: US Congress launches probe into virtual economies. Reuters (October 15, 2006), http://secondlife.reuters.com/stories/2006/10/15/us-congress-launchs-probe-into-virtual-economies/

[34] Ross, A.: An Introduction to Multibiometrics. In: 15th European Signal Processing Conference (EUSIPCO) Poznan, Poland (September 2007)

[35] Ross, A., Jain, A.: Information fusion in biometrics. Pattern Recognition Letters 24, 2115–2125 (2003)

[36] Shin, E., Kwak, S.S., Kim, M.S.: A Study on the Elements of Body Feature Based on the Classification of Social Robots. In: 17th IEEE International Symposium on Robot and Human Interactive Communication, Munich, Germany, August 1-3 (2008)

[37] Suler, J.: The Psychology of Cyberspace (2009), On-line book, http://psycyber.blogspot.com

[38] Sullivan, K.: Virtual Money Laundering and Fraud - Second Life and Other Online Sites Targeted by Criminals. Bank Info Security (April 3, 2008), http://www.bankinfosecurity.com/articles.php?art_id=809

[39] Takahashi, Y., Hatakeyama, M.: Fabrication of simple robot face regarding experimental results of human facial expressions. In: International Conference on Control, Automation and Systems (ICCAS 2008), Atsugi, Japan, October 14-17, pp. 1641–1646 (2008)

[40] Talbot, D.: The Fleecing of the Avatars. MIT Technology Review (January 2008), http://www.technologyreview.com/Infotech/19844/

[41] Turner, A.: Second Life charges for real names, increases identity theft risk. IT Wire (Thursday 5, 2007), http://www.itwire.com/opinion-and-analysis/seeking-nerdvana/11110-second-life-charges-for-real-names-increases-identity-theft-risk

[42] Wagner, M.: Sex in Second Life. Information Week (May 26, 2007), http://www.informationweek.com/news/software/hosted/showArticle.jhtml?articleID=199701944

[43] Wikimedia, Picasa, http://en.wikipedia.org/wiki/Picasa (retrieved April 3, 2011)

[44] Won, C.S., Park, D.K., Park, S.J.: Efficient Use of MPEG-7 Edge Histogram Descriptor. ETRI Journal 24(1), 23–30 (2002)

[45] Yampolskiy, R.V.: Behavioral Biometrics for Verification and Recognition of AI Programs. In: 20th Annual Computer Science and Engineering Graduate Conference (GradConf 2007), Buffalo, NY (2007)

[46] Yampolskiy, R.V., Cho, G., Rosenthal, R., Gavrilova, M.L.: Evaluation of Face Detection and Recognition Algorithms on Avatar Face Datasets. In: International Conference on Cyberworlds (CW 2011), Banff, Alberta, Canada, October 4-6 (2011)

[47] Yampolskiy, R.V., Cho, G., Rosenthal, R., Gavrilova, M.L.: Evaluation of Face Recognition Algorithms on Avatar Face Datasets. In: International Conference on Cyberworlds, Banff, Canada (2011)

[48] Yampolskiy, R.V., Govindaraju, V.: Behavioral Biometrics for Recognition and Verification of Game Bots. In: The 8th Annual European Game-On Conference on Simulation and AI in Computer Games (GAMEON 2007), Bologna, Italy, November 20-22 (2007)

[49] Yampolskiy, R.V., Govindaraju, V.: Behavioral Biometrics for Verification and Recognition of Malicious Software Agents. In: SPIE Defense and Security Symposium on Sensors, and Command, Control, Communications, and Intelligence (C3I) Technologies for Homeland Security and Homeland Defense VII, Orlando, Florida, March 16-20 (2008)

[50] Yampolskiy, R.V., Klare, B., Jain, A.K.: Face Recognition in the Virtual World: Recognizing Avatar Faces. In: SPIE Defense and Security Symposium on Biometric Technology for Human Identification IX, Baltimore, Maryland, USA, April 23-27 (2012)

Visualization of Joinery Using Homotopy Theory and Attaching Maps

Kenji Ohmori[1] and Tosiyasu L. Kunii[2]

[1] Hosei University, Computer and Information Sciences
3-7-2 Kajino-cho, Koganei-shi, Japan
ohmori@hosei.ac.jp
[2] Morpho Inc.
Iidabashi First Tower 31F
2-6-1 Koraku, Bunkyo-ku, Tokyo 112-0004 Japan
kunii@ieee.org

Abstract. Visualization plays an important role to understand complicated phenomena. Joinery is complicated combination of woods. The mechanism of joinery, in particular the difference between the tenon-mortise and lap joints, is analyzed using homotopy theory. Then, designing a simple house is visually described using the incrementally modular abstraction hierarchy that starts from the homotopy level and ends at the physical level. At the cellular structured level that plays an important role in design, the frame represented by a fundamental group of homotopy theory is transformed into CW-complex consisting of cells, where the concept model is transferred to the physical model. Visualization of designing joinery is realized at each level of the abstraction hierarchy to help a designer understand joinery intuitively.

Keywords: joinery, woodworking, attaching map, homotopy, homotopy lifting properties.

1 Introduction

Joinery [8], which is a part of woodworking that involves jointing together pieces of woods, is sophisticated work. As the pieces of woods that have been manufactured to construct a house, furniture and toys have complicated shapes, a considerable amount of time and effort is necessary to recognize how these pieces are jointed together. Visualization and animation help us understand the complicated process of joining pieces of woods if the jointing shape is not complex. Otherwise, visualized deformation without losing generality is useful. Homotopy theory [1], [9], [10], which is the most modern mathematics and belongs to the field of topology, is a useful tool for deformation as well as visualization since it combines algebra with geometry.

Jointing and disjointing are the essential concept in cyberworlds. These phenomena are observed in many cases. A mobile telephone, which is a key tool

M.L. Gavrilova and C.J.K. Tan (Eds.): Trans. on Comput. Sci. XVI, LNCS 7380, pp. 95–114, 2012.

in cyberworlds, dynamically changes its base stations when it is moving. Merging and separation of companies also characterize the industrial activities in cyberworld age. Therefore, jointing and disjointing are worthwhile events to describe cyberworlds. However, set theory accommodating intersection and union as binary operations cannot describe the property of jointing and disjointing accurately since set theory does not have capability of attaching.

An attaching map, which accommodates disjoint union of two entities with the feature of identifying an attaching area where a piece of wood attaches another piece, gives a mathematical tool for visualizing the process of jointing together pieces of woods [6]. The paper introduces a virtual attaching space to connect indirectly one piece of a wood with another one. By introducing a virtual attaching space, it is avoided to consider the complex relation of two pieces and focused on considering the attaching area of each piece by deformation while preserving essential properties. As the attaching area dynamically changes its shape in the process of jointing, the attaching map also varies continuously. Homotopy theory, introduced to visualizing and modeling cyberworlds in this paper, explains the continuous and dynamic change of attaching maps.

Homotopy theory has been applied to software development [5], an accounting system [4], kaleidoscope behavior [3] and logical thinking [7]. The research mainly treats the methodologies of design, analysis and implement by applying two important properties of homotopy to top-down and bottom-up approaches. On the other hand, this paper emphasizes visualization to which homotopy theory is applied.

Jointing process consists of three situations: before-joining, under-joining and after-joining. As homotopy theory describes changes of shapes, the situation of under-joining, which increases the attaching area continuously, can be described completely. However, transition from the situation of before-joining to one of under-joining and from the situation of under-joining to one of after-joining are discontinuous, which are reflected in homotopy theory in the form of changing from one fundamental group to another different one. These transitions are deformed to the abstract level of homotopy and visualized to understand easily what is happening in jointing together pieces of woods.

Joinery has many different techniques to joint pieces of woods. The difference in jointing techniques appears as difference in fundamental groups. The paper describes two different basic techniques of the lap and tenon-mortise joints and explains how these techniques are different in the visual abstract level. Through visualization, essential properties of each techniques are clarified, which is very important to analyze, model and implement cyberworlds.

Finally, a simple house utilizing tenon-mortise and lap joints is designed using the incrementally modular abstraction hierarchy. The modular abstraction hierarchy is a methodology for designing, modeling and testing. Using this hierarchy, the house is designed from the homotopy level to the physical level by providing a visualization method at each level.

2 Joint Classification

Many types of jointing techniques are used in joinery. Among them, the lap and tenon-mortise joints are typical techniques. The lap joint attaches two pieces of woods by overlapping each other and the tenon-mortise joint by fitting a tenon tightly into a mortise hole. Fig. 1 shows difference in an essential property at the abstract level of homotopy theory between two joint techniques. The top of this figure shows the most abstract description of joining two pieces of woods. Two pieces are attached together by an attaching map f, which will be explained in the following section. Then, jointing structures are classified in accordance with joint techniques. In this figure, it is divided into two groups for the tenon-mortise and lap joints. The figure shows how two pieces are connected in each technique. Then, attaching areas are shown in case of under-jointing and after-jointing for the two techniques. In the case of the tenon-mortise joint, the attaching area of under-jointing is the shape of a cylinder, which is homotopy equivalent to a ring. On the other hand, the attaching area of the lap joint is one of a rectangular, which is homotopy equivalent to a point. Therefore, the tenon-mortise joint is different from the lap joint in at the most abstract level. The difference is also visualized. After joining, the attaching area for the tenon-mortise joint is the shape of a hat, which is homotopy equivalent to a point, and the lap-joint is one of a chair, which is also homotopy equivalent to a point. Therefore, two pieces are completely attached, so that the two techniques give the same results at the homotopy level. The final results at the most abstract level are also visualized.

3 Attaching Maps

As attaching maps are main tools in this paper, the property of attaching map is described at first. When explaining attaching maps, equivalence relations play important roles to identify entities. Equivalence relations are defined as follows.

3.1 Equivalence Relations

For a binary relation $R \subseteq X \times X$ on a set X, if R is :
1 reflexive if $(\forall x \in X)[xRx]$: reflexivity,
2 symmetric if $(\forall x, y \in X)[xRy \Rightarrow yRx]$: symmetry, and
3 transitive if $(\forall x, y, z \in X)[[xRy \Rightarrow yRz] \Rightarrow xRz]$: transitivity,
R is called an equivalence relation (in a notation $\sim$).
Given $x \in X$, a subset of X defined by

$$x/\sim = \{y \in X : x \sim y\} \tag{1}$$

is called the equivalence class of x. Here a class actually means a set; it is a tradition, and hard to be changed at this stage. The set of all the equivalence classes $X/\sim$ is called the quotient space or the identification space of X.

$$X/\sim = \{x/\sim \in 2^X | x \in X\} \subseteq 2^X. \tag{2}$$

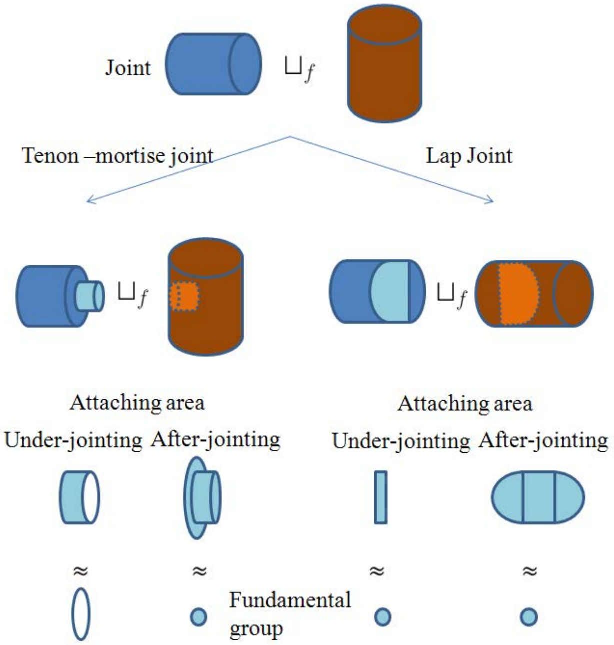


Fig. 1. Joint classification

From the transitivity, for each $x \in X, x/\sim \neq \phi$, the followings hold:

$$x \sim y \Leftrightarrow x/\sim = y/\sim, \tag{3}$$

$$x \not\sim y \Leftrightarrow x/\sim \cap\, y/\sim = \phi. \tag{4}$$

This means a set X is partitioned (also called decomposed) into non-empty and disjoint equivalence classes. This property plays a central role to make them modular.

When an equivalence relation is introduced, the elements that are defined as the same construct a class, which is called a quotient space. A quotient space is defined as follows.

3.2 A Quotient Space

Let X be a topological space. Let f be a surjective (onto) and continuous mapping called a quotient map (often also called an identification map) that maps each point $x \in X$ to a subset (an equivalence class $x/\sim \in X/\sim$) containing x. Then, $f : X \to X/\sim$. $X/\sim$ is called a quotient space (or an identification space) by a quotient map (or an identification map) f. There is a reason why a quotient space is also called an identification space. It is because, as stated before, a quotient space is obtained by identifying each element (an equivalence class) $x/\sim \in X/\sim$ with a point $x \in X$ that is contained in $x/\sim$.

An attaching map shows mathematically how two spaces are merged. It is defined as follows.

3.3 An Attaching Space

Let us start with a topological space X and attach another topological space Y to it. Then,

$$Y_f = Y \sqcup_f X = Y \sqcup X / \sim \tag{5}$$

is an attaching space (an adjunction space, or an adjoining space) obtained by attaching (gluing, adjuncting, or adjoining) Y to X by an attaching map (an adjunction map, or an adjoining map) f (or by identifying each point $y \in Y_0 | Y_0 \subseteq Y$ with its image $f(y) \in X$ by a continuous map f). $\sqcup$ denotes a disjoint union (another name is an "exclusive or" and often a + symbol is used instead. Attaching map f is a continuous map such that $f : Y_0 \to X$, where $Y_0 \subseteq Y$. Thus, the attaching space $Y_f = Y \sqcup X / \sim$ is a case of quotient spaces

$$Y \sqcup X / \sim = Y \sqcup_f X = Y \sqcup X / (x \sim f(y) | \forall y \in Y_0). \tag{6}$$

The identification map g in this case is

$$g : Y \sqcup X \to Y \sqcup_f X = Y_f = Y \sqcup X / \sim = (Y \sqcup X - Y_0) \sqcup Y_0. \tag{7}$$

A continuous function, which appears in the previous sentence, is important concept in mathematics. It is defined as follows.

3.4 A Continuous Function

A function $f : X \to Y$, where X and Y are topological spaces, is continuous if and only if the inverse image

$$f^{(-1)}(V) = \{x \in X | f(x) \in V\} \tag{8}$$

is open for every open set $V \subseteq Y$.

A matchmaking party gives a good example to explain how separate groups are merged together using an attaching map. There are a group of girls, which is represented by a set

$$G = \{Betty, Michel, Alice, Chelsea, Jackie\}, \tag{9}$$

and another group of boys described by a set

$$B = \{Tom, Mike, Peter, Jack, Alex\}. \tag{10}$$

A member of girls has never met a member of the boy group. Suppose that these groups have a matching party. The situation before the meeting is represented by the disjoint union of two groups. The situation after the meeting is described using an attaching map, which describes new partners that have been made at the party.

The disjoint union of these two sets is represented by

$$G \sqcup B = \{(Betty, f), (Michel, f), (Alice, f), (Chelsea, f), (Jackie, f), (Tom, m), (Mike, m), (Peter, m), (Jack, m), (Alex, m)\}. \quad (11)$$

At the meeting, Betty and Mike like each other and become a partner. Also, Chelsea and Peter become another partner as shown in Fig. 2. If a partner is considered to be identical by an attaching map f, then,

$$G \sqcup_f B = \{\{(Betty, f), (Mike, m)\}, (Michel, f), (Alice, f), \{(Chelsea, f), (Peter, m)\}, (Jackie, f), (Tom, m), (Jack, m), (Alex, m)\}, (12)$$

where $f : G_0 \to B$ and $G_0 = \{(Betty, f), (Chelsea, f)\}$.

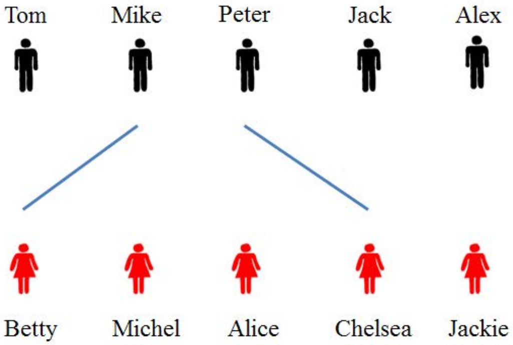


Fig. 2. A matching party

4 How to Employ Homotopy in Cyberworlds

Homotopy is defined in mathematics as follows.

4.1 Homotopy

A homotopy between two continuous functions f and g from a topological space X to a topological space Y is defined to be a continuous function $H : X \times [0,1] \to Y$ from the product of the space X with the unit interval $[0,1]$ to Y such that, if $x \in X$ then $H(x,0) = f(x)$ and $H(x,1) = g(x)$. If the second parameter of H is considered as time then H describes a continuous deformation of f into g where we have the function f at time 0 and the function g at time 1.

As cyberworlds are often employed in the set of discrete elements, mathematical terms have to be interpreted to adjust discrete characteristics of cyberworlds. The set of discrete elements is transformed into a topological space by introducing the trivial space or the discrete space. When the unit interval is discrete, the elements of the unit interval are treated as the components of a totally ordered set.

Fig. 3 is an example of homotopy in cyberworlds. X and Y are the sets of discrete elements. Y is divided into the subset of $\{Y_0, Y_1, Y_2, Y_3, Y_4\}$. It is considered that X is mapped to Y_i at time t_i. This example can be used to describe how a certain area has changed in history or how the financial state of a company has changed in recent years. Now, we will introduce topological spaces to X and Y and define a homotopy H. The topological space (X, T_x) is defined by introducing the subset $T_x = \{\phi, X\}$ of X. The interval is defined as follows.

$$t_i \in I, T_I = \{[t_i, t_j] | t_i, t_j \in I, i \leq j\}. \tag{13}$$

I is a totally ordered set. The topological space (Y, T_y) is also defined by introducing the power set T_y of $\{Y_0, Y_1, Y_2, Y_3, Y_4\}$. Then, the homotopy H is defined by

$$H(x, t_i) : x \in X, t_i \in I \to y \in Y. \tag{14}$$

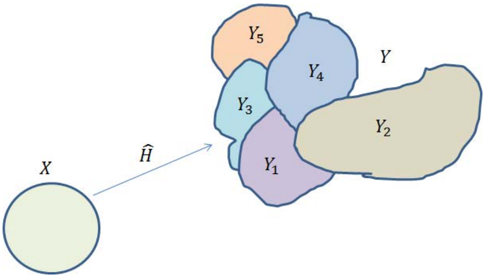


Fig. 3. Homotopy employed in cyberworlds

5 The Homotopy Lifting Property in a Discrete Space

5.1 The Homotopy Lifting Homotopy

The homotopy lifting property is defined as follows. Given any commutative diagram of continuous maps as shown in Fig. 4, the map $p : E \to B$ has the homotopy lifting property if there is a continuous map $\hat{H} : Y \times I \to E$ such that $p \circ \hat{H} = H$. The homotopy $\hat{H}$ thus lifts H through p.

Fig. 4 also includes an example of the homotopy lifting property. E is the surface of a grove and B is the projected space of B. X is a line which is mapped to E. However, the mapped lines are continuously changed along the time interval I as shown in Fig. 4.

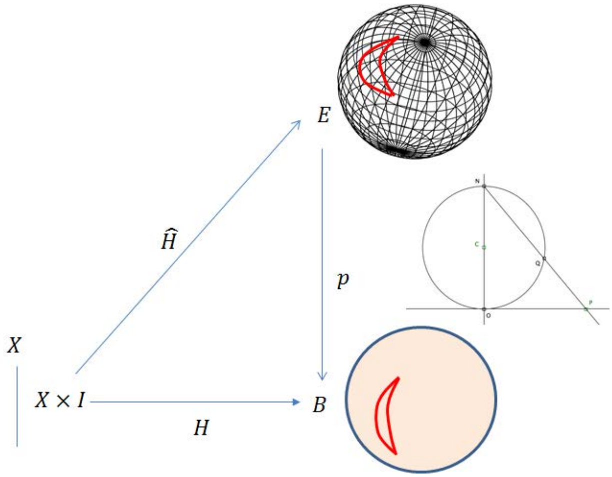


Fig. 4. The homotopy lifting property

5.2 A Fiber Bundle

A fiber bundle is a quadruple $\xi = (E, B, F, p)$ consisting of a total space E, a base space B, a fiber F, and a bundle projection that is a continuous surjection called $F-$bundle $p : E \to B$ such that there exists an open covering $\mathcal{U} = \{U\}$ of B and, for each $U \in \mathcal{U}$, a homeomorphism called a coordinate chart

$$\varphi_U : U \times F \to p^{-1}(U) \tag{15}$$

exists such that the composite

$$U \times F \to p^{-1}(U) \to U \tag{16}$$

is the projection to the first factor U. Thus the bundle projection $p : E \to B$ and the projection $p_B : B \times F \to B$ are locally equivalent. The fiber over $b \in B$ is defined to be equal to $p^{-1}(b)$, and it is noted that F is homeomorphic to $p^{-1}(b)$ for every $b \in B$, namely $\forall b \in B, F \cong p^{-1}(b)$.

A Mobius strip gives a good example to explain a fiber bundle as shown in Fig. 5. If a Mobius strip and its center circle described by the dot line are considered as E and B of a fiber bundle and any point of E is projected to the intersection of the center circle with the perpendicular line from the point of E to the center circle, then, F is a straight line segment vertical to the center circle.

Though a cylinder is different from a Mobius strip in shape, these objects are alike in homotopy. When constructing a fiber bundle for a cylinder, the cylinder has the same B and H as a Mobius strip. B of the cylinder is also a circle. The fibers are also straight line segments vertical to the center circle. However, the fibers of the Mobius strip are rotating along the center circle. On the other

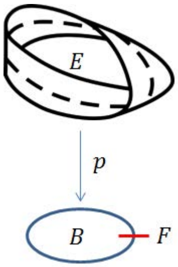


Fig. 5. A mobius strip

hand, the fibers of the cylinder are always straight, which causes the difference in shape. B is considered as a common property or invariant among similar objects in homotopy. A cylinder is deformed to the circle by moving any point perpendicularity to the point of the center circle. A Mobius strip is also deformed to the circle by the same moving. In the abstract level, a cylinder and a Mobius strip are the same. The invariance of two objects is explicitly represented by B.

6 Describing the Process of Jointing Two Pieces of Woods by the Homotopy Lifting Property

The process of jointing together pieces of woods consists of three situations: 1) before-jointing; 2) under-jointing; and 3) after-jointing as shown in Fig. 6. As was shown in the previous section, two pieces are completely attached together after joining them regardless of joint techniques. As a result, the attaching area after jointing is homotopy equivalent to a point. However, while jointing together pieces of woods, the abstract description of an attaching area is different in joint technique.

The visual abstract model for the tenon-mortise joint is described using the homotopy lifting property.

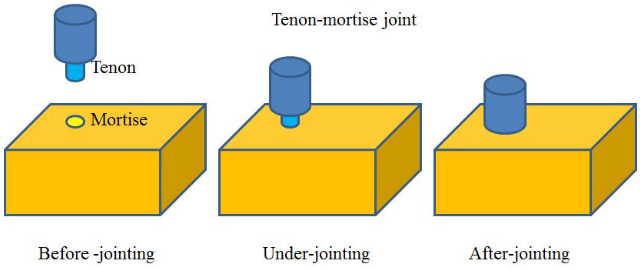


Fig. 6. The process of jointing together pieces of woods

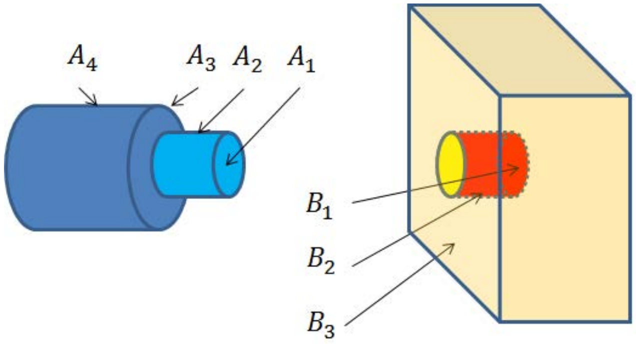


Fig. 7. The subspaces for two pieces of woods

Let's consider to define the process of jointing woods in the abstract level. Suppose that two pieces A and B of woods are provided and their subspaces are named as shown in Fig. 7 so that $A = \{A_1, A_2, A_3, A_4\}$ and $B = \{B_1, B_2, B_3\}$. Before joining, two pieces A and B of woods are separated. Under joining, parts of two pieces of woods are attached. In this case, part of A_2 is attached to part of B_2. After jointing, the jointing areas of both woods are completely attached. In Fig. 7, the whole area of A_2 is attached to the whole area of B_2. In addition, A_1 is completely attached to B_1 and A_3 to part of B_3.

Time is given to each situation so that 1) $t \leq 0$ for before jointing, 2) $0 < t < 1$ for under jointing, and 3) $t \geq 1$ for after jointing. For $t < 0$, the relation of A and B is represented by $\{A, B | A \sqcup B\}$. At $t = 0$, two woods are prepared for jointing so that the boundary of A_2 is slightly touched to the boundary of B_2. A virtual attaching space Y_2, which is homotopy equivalent to a cylinder, is provided so that Y_2 is attached to the boundary of A and also to one of B. As a result, both boundaries of A and B are indirectly attached together through Y_2. This relation is represented by $\{A, B | Y_2 \sqcup_{f_0} \partial A_2, Y_2 \sqcup_{g_0} \partial B_2\}$ where ∂ is a boundary.

As jointing process advances, that is, for $0 < t < 1$, the attaching areas of A_2 and B_2 are increasing. This phenomenon is explained using homotopy maps. The attaching areas of A_2 and B_2 through Y_2 are described by homotopy maps from Y_2 to A_2 and B_2 so that

$$\hat{H}_1(y, t) = f_t(y), \tag{17}$$
$$\hat{H}_1(y, 0) = f_0(y), \tag{18}$$
$$\hat{H}_1(y, 1) = f_1(y), \tag{19}$$
$$\hat{H}_2(y, t) = g_t(y), \tag{20}$$
$$\hat{H}_2(y, 0) = g_0(y), \tag{21}$$
$$\hat{H}_2(y, 1) = g_1(y) \tag{22}$$

and $y \in Y_2$ as shown in Fig. 8.

These relations are represented by two diagrams of the homotopy lifting properties. For Y_2 and A, A is assigned to E of the diagram, Y_2 to X, and t to I as

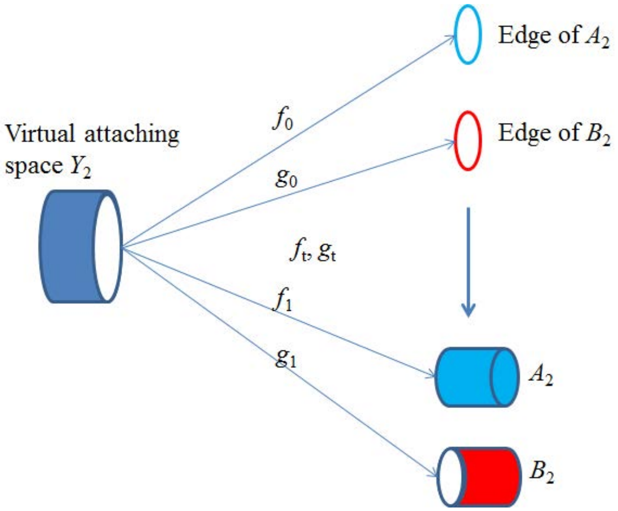


Fig. 8. An attaching map

an element as shown in Fig. 9. B is a ring that is homotopy equivalent to Y_2. $\hat{H}_1$ is assigned to $\hat{H}$ of the diagram. If F is the topological space of a line segment, then (E, B, F, p) constitutes a fiber bundle.

In the same way, the other diagram is defined for Y_2 and piece B of the other wood.

At $t = 1$, A_1 and A_3 are attached to the part of B_1 and the whole area of B_3, respectively. In the same way as $t = 0$, virtual attaching spaces Y_1 and Y_3 are provided. The attaching maps h_{11}: $Y_1 \to A_1$, h_{12}: $Y_3 \to A_3$, h_{21}: $Y_2 \to B_1$ and h_{22}: $Y_3 \to B_3$ are defined. The relation of A, B is described by

$$\begin{aligned}\{A, B | Y_2 \sqcup_{f_1} A_2, Y_2 \sqcup_{g_1} B_2, Y_1 \sqcup_{h_{11}} A_1,\\ Y_1 \sqcup_{h_{21}} B_1, Y_3 \sqcup_{h_{12}} A_3, Y_3 \sqcup_{h_{22}} B_3\}.\end{aligned} \tag{23}$$

For $t > 1$, the relation of A, B is the same as at $t = 1$.

The jointing process is visualized and formalized by homotopy theory and attaching maps.

7 Hierarchical Construction of a House

Lets consider constructing a simple house using an incrementally modular abstraction hierarchy.

7.1 An Incrementally Modular Abstraction Hierarchy

The incrementally modular abstraction hierarchy (IMAH) gives a designer a design methodology to construct a system according to different abstraction

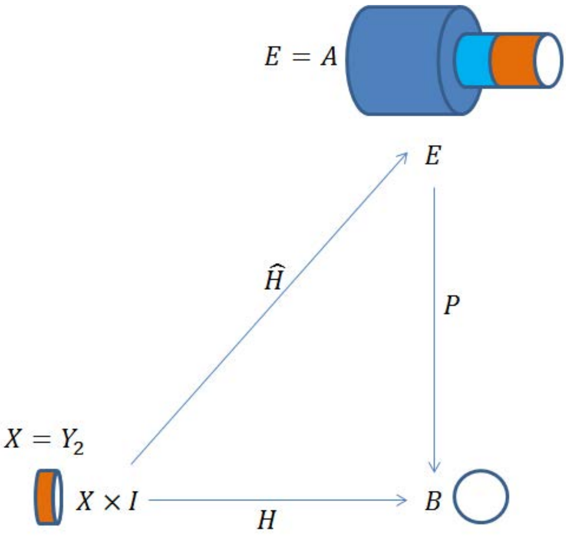


Fig. 9. The homotopy lifting property for the tenon-mortise joint

levels. A designer can design a system by climbing down abstraction hierarchy in a bottom-up way or by climbing up abstraction hierarchy in a top-down way. Or a designed can climb up and down abstraction hierarchy for a developing a system by an error-and-trial approach. IMAH consists of the following levels. 1. The homotopy (including fiber bundles) level; 2. The set theoretical level; 3. The topological space level; 4. The adjunction space level; 5. The cellular structured space level; 6. The presentation (including geometry) level; 7. The view (also called projection) level.

At the most abstract level, the most important properties are defined. These properties, called invariants, are kept while climbing down the abstraction hierarchy. At the second most abstract level, the second most important properties are defined and added to the most important properties. By climbing down, new properties are added lineally while keeping original properties. In the top down method of the IMAH, as the invariants defined at a general level is kept at a specific level, design faults and errors are avoided. On the contrary, when climbing up abstraction hierarchy, more important properties are extracted. By the bottom-up method, the ideas are put together.

7.2 The Homotopy Level

According to the IMAH, lets consider to design a simple wooden house that is composed of four posts, four roof beams and four floor beams, which is the simplest form of a Japanese house shown in Fig. 10. At the homotopy level, the implementation procedure for a house is defined using the homotopy lifting property as shown in Fig. 11. The model of the house is defined by $X \times I$. X is a

Fig. 10. A typical Japanese house

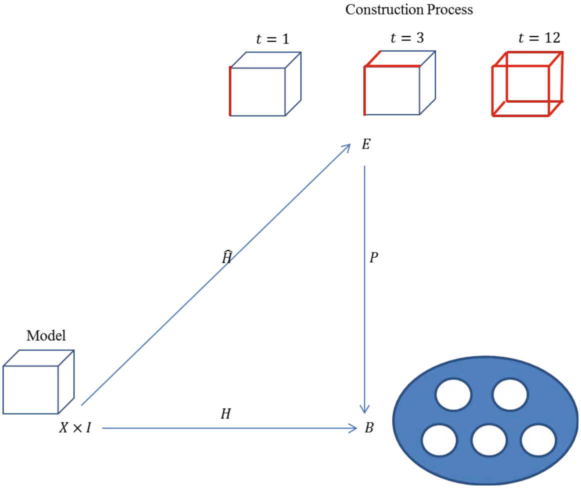


Fig. 11. A homotopy lifting property for constructing a house

space describing the model. I is a sequence of implementation steps. $t \in I$ is the t^{th} step in implementation. The sequence of implementation is represented in E. In Fig. 11, the constructing house at the first, third and final steps are depicted. The homotopy fundamental group of the house is described in B. In any case, the constructing house has to be homotopy equivalence to this fundamental group. Otherwise, a different house from the model will be constructed. B is a disk with five holes, which is homotopically equivalent to a four post house.

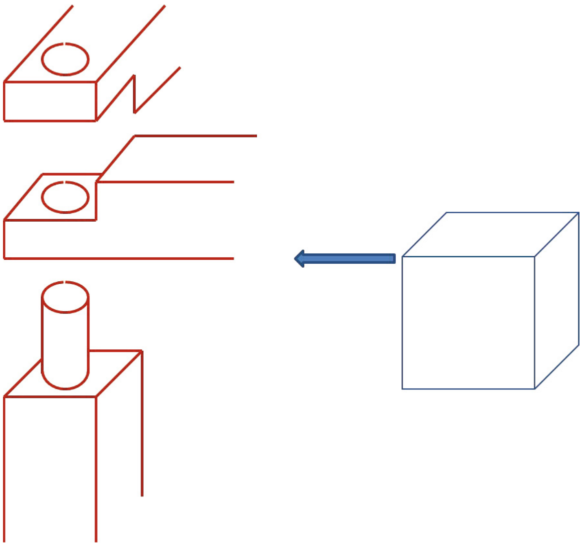


Fig. 12. A corner attached by lap and tenon-mortise joints

7.3 The Set Theoretical and Topological Space Levels

As a four post house H consists of four post p_1, p_2, p_3, p_4, four roof beams r_1, r_2, r_3, r_4 and four floor beams f_1, f_2, f_3, f_4, the house is described as a set of woods.

$$H = \{p_1, p_2, p_3, p_4, r_1, r_2, r_3, r_4, f_1, f_2, f_3, f_4\}. \tag{24}$$

The topological space is induced to the set as the discrete topology on H, which is the power set of H.

7.4 The Adjunction Space Level

At this level, it is defined how the components for the whole is connected using an attaching map. To construct a house, posts are connected to beams. At each corner of the four post house, one post and two beams have to be connected. Suppose that at each corner, two beams are attached by a lap joint and a post and the two beams are attached by a tenon-mortise joint as shown in Fig. 12.

As a beam has two holes for a tenon-mortise joint, it is homotopy equivalent to a two connected rings or a disk with two holes, which is described as $S^1 \vee S^1$. As a post has no holes, it is homotopy equivalent to a disk described as D^1. Suppose that two beams are firstly attached and a post is then connected to them. This process is described using attaching map. Suppose that r_2 is attached to r_3 using an attaching map f_1. As one ring of r_2 is attached to one of r_2, the connected beams have three rings and are homotopy equivalent to $S^1 \vee S^1 \vee S^1$. Then, a post is connected to the beams by an attaching map f_2, so that the ring connecting the beams is closed by the post. Therefore, the following result is obtained for the connected woods,

$$r_2 \sqcup_{f_1} r_3 \sqcup_{f_2} p_1 \sim S^1 \vee S^1 \tag{25}$$

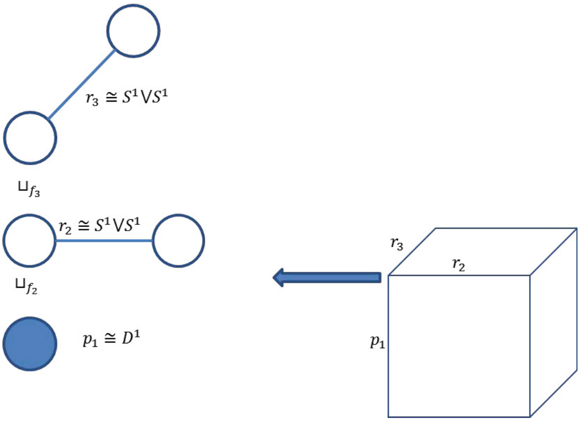


Fig. 13. Attaching process using attaching maps

The attaching process for the woods is shown in Fig. 13.

7.5 The Cellular Structured Space Level

At the homotopy level, no dimension is defined. However, a house and woods has dimensions. At a concrete level, dimension is introduced. At the cellular structured space level, woods are represented a set of cells with a dimension for constructing CW-complex.

Fig. 14 shows cell construction for a beam. At first, a tree-dimensional close ball $\mathcal{B}$ is prepared. To make a hole, two boundaries of the ball is attached together using an attaching map $g_1 : \partial\mathcal{B} \to \partial\mathcal{B}$. Then, another hole is configured by attaching map $g_2 : \partial\mathcal{B} \to \partial\mathcal{B}$, so that the ball with two holes are provided. To obtain a beam, the ball with two holes is transformed while keeping topological equivalence to the ball. The process obtaining a beam is depicted in Fig. 14. From the final result, cells are provided. Before describing this process, filtration has to be discussed.

A Filtration Space. A filtration space is a sequence of cells used to represent a topological space. It is defined as follows. For any topological space X, we can obtain a finite or infinite sequence of skeletons X^p, where p is an integer, such that

$$X = \cup_{p \in Z} X^p, \tag{26}$$

$$X^0 \subset X^1 \subset .... \subset X^p .... \subset X. \tag{27}$$

A skeleton X^p consists of cells whose dimensions do not exceed n. A cell is a topological space, homeomorphic to an n-dimensional open ball $Int\mathcal{B}^n$, where n is an arbitrary integer. A sequence of skeletons is called a filtration. If it is finite, it becomes a CW-complex.

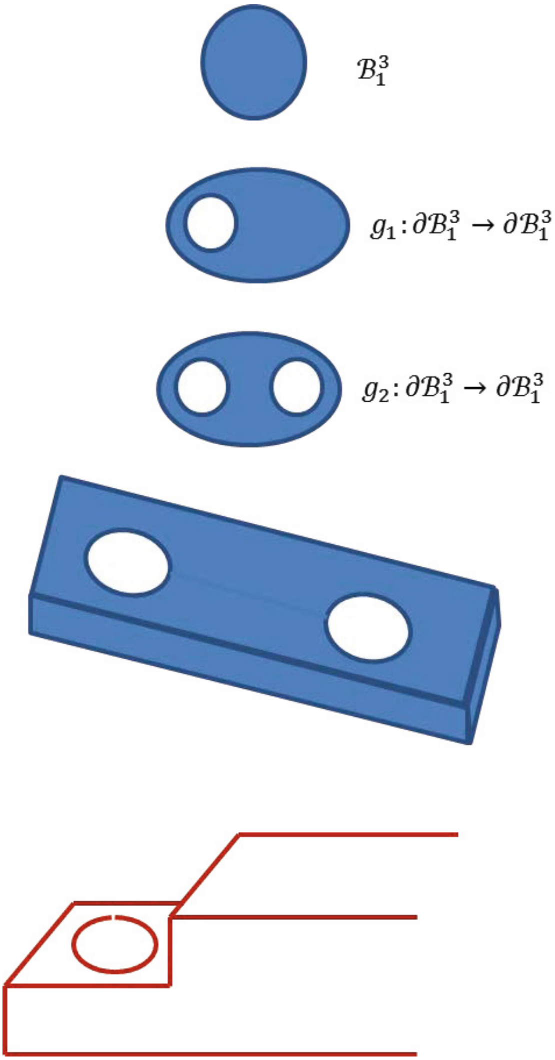

$\mathcal{B}_1^3$
$g_1 : \partial\mathcal{B}_1^3 \rightarrow \partial\mathcal{B}_1^3$
$g_2 : \partial\mathcal{B}_1^3 \rightarrow \partial\mathcal{B}_1^3$

Fig. 14. Topological deformation from a ball to a wood

Using the filtration space, a topological space is constructed as a CW-complex or a cellular structured space. A skeleton X^p is obtained from the skeleton X^{p-1} by attaching $\sqcup_i \mathcal{B}_i^p$ to X^{p-1} through an attaching map f where $\mathcal{B}_i^p$ is a p-dimensional closed ball. The attaching map is a surjective and continuous function.

$$f : \sqcup_i \partial\mathcal{B}_i^p \rightarrow X^{p-1}, \tag{28}$$

$$X^p = X^{p-1} \sqcup (\sqcup_i \mathcal{B}_i^p)/(x \sim f(x) | x \in \sqcup_i \partial\mathcal{B}_i^p). \tag{29}$$

The Cellular Structured Space for a Tetrahedron. Before constructing the cellular structured space of the tenon-mortise joint, it is shown how a cellular structured space is organized using the filtration space [2]. A function $\mathcal{F}$, called a continuous function, is introduced to transfer an internal space of a closed ball into a cell, which is an open set, of a cellular structured space. If a filtration space is represented by using polygons, then

$$X^p = \mathcal{F}(\sqcup_i \mathcal{B}_i^p). \tag{30}$$

If $Int\mathcal{B}_i^p$ is transformed to a cell e_i^p and X^p is attached to X^{p-1} by an attaching map f, then

$$\begin{aligned} X^p &= \mathcal{F}(\sqcup_i \mathcal{B}_i^p) && (31)\\ &= \mathcal{F}((Int \sqcup_i \mathcal{B}_i^p) \sqcup (\partial \sqcup_i \mathcal{B}_i^p)) && (32)\\ &= \{e_1^p, e_1^p, , e_n^p, X^{p-1}\}, && (33) \end{aligned}$$

where $f : \partial \sqcup_i \mathcal{B}_i^p \to X^{p-1}$ and $\partial\mathcal{B}$ is the boundary of $\mathcal{B}$.

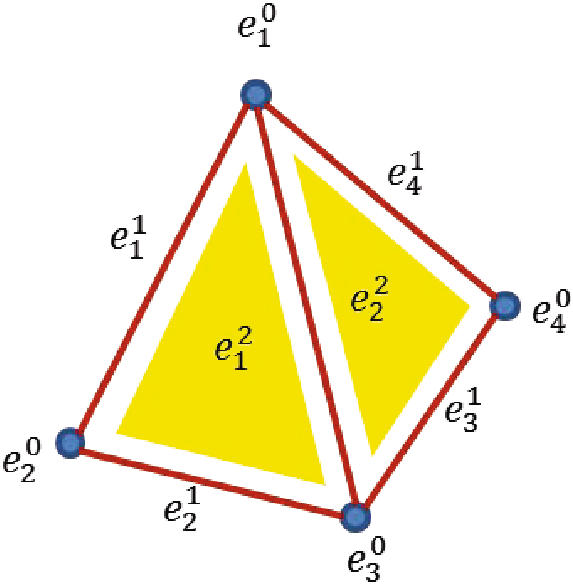


Fig. 15. The cellular structure model of a tetrahedron

The filtration space of the tetrahedron of Fig. 15 is obtained as follows. As the tetrahedron has four vertexes, it is represented by four 0-dimensional cells as follows,

$$X^0 = \mathcal{F}(\sqcup_i \mathcal{B}_i^0) \tag{34}$$
$$= \{e_1^0, e_2^0, e_3^0, e_4^0\}. \tag{35}$$

The tetrahedron has six edges. The boundary of each edge is connected to two vertexes. Each edge is represented by a 1-dimensional cell as follows,

$$X^1 = \mathcal{F}(\sqcup_i \mathcal{B}_i^1) \tag{36}$$
$$= \mathcal{F}((Int \sqcup_i \mathcal{B}_i^1) \sqcup (\partial \sqcup_i \mathcal{B}_i^1)) \tag{37}$$
$$= \{e_1^1, e_2^1, e_3^1, e_4^1, e_5^1, e_6^1, X^0\}. \tag{38}$$

where $f : \partial \sqcup_i \mathcal{B}_i^1 \to X^0$ means that the boundary of $e_1^1$is attached to e_1^0 and e_2^0, e_2^1 to e_2^0 and e_3^0, and so on.

The tetrahedron has four surfaces, each of which is represented by a 2-dimensional cell. The boundary of a surface is attached to three edges and three vertexes as follows,

$$X^2 = \mathcal{F}(\sqcup_i \mathcal{B}_i^1) \tag{39}$$
$$= \mathcal{F}((Int \sqcup_i \mathcal{B}_i^2) \sqcup (\partial \sqcup_i \mathcal{B}_i^2)) \tag{40}$$
$$= \{e_1^2, e_2^2, e_3^2, e_4^2, X^1\} \tag{41}$$

where $f : \partial \sqcup_i \mathcal{B}_i^2 \to X^1$ means that the boundary of e_1^2 is attached to e_1^0, $e_1^1, e_2^0, e_2^1, e_3^0$ and e_5^1, and so on.

$$X^3 = \mathcal{F}(\mathcal{B}_1^3) \tag{42}$$
$$= \mathcal{F}(Int\mathcal{B}_i^3 \sqcup \partial\mathcal{B}_i^3) \tag{43}$$
$$= \{e_1^3, X^2\} \tag{44}$$

where $f : \partial(\mathcal{B}_1^3) \to X^2$ means that the boundary of e_1^3 is attached to all elements of X^2.

The Cellular Structured Space for a House. The cellular structured space for a beam is obtained in the same way as a tetrahedron as shown in Fig. 16 and Fig. 17. At first, the 0-dimensional cells are obtained as follows,

$$X^0 = \{e_1^0, e_2^0, ...\} \tag{45}$$

where $e_1^0, e_2^0, e_3^0, e_4^0$ are attached points when a hole is constructed.

Then, the 1-dimensiona cells are provided as follows,

$$X^1 = \{e_1^1, e_2^1, ...\} \tag{46}$$

where $e_1^1, e_2^1, e_3^1, e_4^1$ are attached lines when a hole is constructed.

In the same way, X^2, X^3 is constructed.

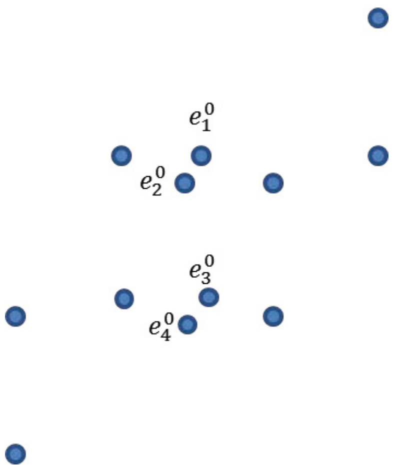


Fig. 16. 0-dimensional cells

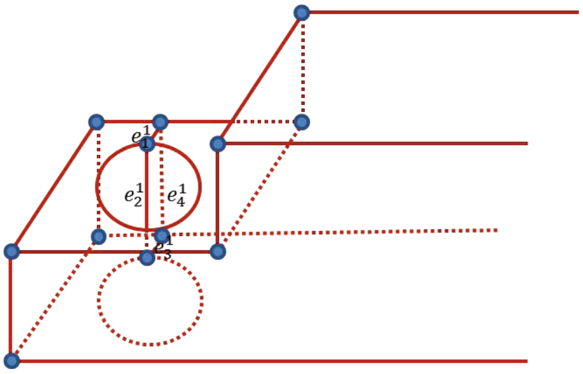


Fig. 17. 1-dimensional cells

7.6 The Presentation and View Level

Each cell composing the house is instantiated by giving position, length, area and volume. The designed house will be described precisely using CAD software and observed.

8 Conclusions

Visualization is a very important tool for understanding complicated and sophisticated phenomena. The jointing of pieces of woods is one example of such cases. We have succeeded in visualizing the process of jointing using homotopy theory and attaching maps. Attaching maps have been used to describe how entities are attached. Though the paper does not describe how two jointing entities are disjointed, it can be easily realized by the inverse map of an attaching map.

Two pieces of woods are jointed together by changing the structure or relation of them, which may change the fundamental group that the two pieces of woods belong to. The joining structure depends on a jointing technique. The paper has showed that the tenon-mortise joint gives the fundamental group of $\mathbb{Z}$ equivalent to a ring and the lap joint does 0 to a point. These properties as invariants should be kept through implementing an animation form the abstract model in homotopy to the specific models in the cellular structured level and the physical presentation one.

The changes of an attaching area have been successfully described using the homotopy lifting property. In using the property, two pieces of woods are indirectly attached through a virtual attaching space. As the relation of two pieces is sometimes very complicated, the unnecessary description of this relation should be avoided. The virtual attaching space is effective to describe the process of jointing. The virtual attaching space guarantees the appropriate attaching of two pieces by equivalence relation.

Joinery plays an important role when building a wooden house. The paper describes how the incrementally modular abstraction hierarchy is applied for designing a house in a top-down way so that the invariants defined at an abstract level are preserved at a concrete level so that design faults and errors are avoided.

As homotopy theory combines algebra and geometry, the application area for visualization is not limited in joinery but is expanded wide areas of cyberworlds.

References

1. Dodson, C., Parker, P.E.: A User's Guide to Algebraic Topology. Kluwer Academic Pub., Boston (1997)
2. Kunii, T.L.: Valid computational shape modeling: Design and implementation, pp. 123–133. World Scientific (December 1999)
3. Kunii, T.L., Ohmori, K.: Cyberworlds: A kaleidoscope as a cyberworld and its animation: Linear architecture and modeling based on an incrementally modular abstraction hierarchy. Computer Animation and Virtual Worlds 17(3-4), 145–153 (2006)
4. Ohmori, K., Kunii, T.L.: Development of an accounting system. In: ICEIS 2007, pp. 437–444 (June 2007)
5. Ohmori, K., Kunii, T.L.: Designing and modeling cyberworlds using the incrementally modular abstraction hierarchy based on homotopy theory. The Visual Computer 26(5), 297–309 (2010)
6. Ohmori, K., Kunii, T.L.: Visualized deformation of joinery to understand jointing process by homotopy theory and attaching maps. In: Int. Conf. on Cyberworlds 2011, pp. 203–211 (October 2011)
7. Ohmori, K., Kunii, T.L.: Visualizing logical thinking using homotopy: A new learning method to survive in dynamically changing cyberworlds. In: The 2011 Int. Conf. on Frontiers in Education: Computer Science and Computer Engineering (July 2011) (to be appeared)
8. Rogowsky, G.: The Complete Illustrated Guide to Joinery. The Taunton Press, Newton (2002)
9. Sieradski, A.J.: An introduction to topology and homotopy. PWS-Kent Publishing Company, Boston (1992)
10. Spanier, E.H.: Algebraic topology. Springer, New York (1966)

Example of Business Applications Using the Numerical Value and Exponential Calculation of the Cellular Data System

Toshio Kodama[1,2], Yoichi Seki[3], and Tosiyasu L. Kunii[4]

[1] School of Engineering, University of Tokyo,
7-3-1 Hongo, Bunkyo-ku, Tokyo 113-0033, Japan
[2] Maeda Corporation, 2-8-8 Sarugaku-cho, Chiyoda-ku, Tokyo 101-0064, Japan
[3] Software Consultant, 3-8-2 Hino-shi, Tokyo 191-0001, Japan
[4] Morpho, Inc., Iidabashi First Tower 31F, 2-6-1 Koraku, Bunkyo-ku, Tokyo 112-0004, Japan
kodama@ken-mgt.t.u-tokyo.ac.jp, kodama.ts@jcity.maeda.co.jp,
gamataki@hotmail.co.jp, kunii@ieee.org

Abstract. In the era of cloud computing, data is processed in a “Cloud”, and data and its dependencies between systems or functions progress and change constantly within the cloud, as a user’s requirements change. Such information worlds are called cyberworlds. We need a more powerful mathematical background which can model the cyberworlds in the "cloud" as they are. We consider the Incrementally Modular Abstraction Hierarchy (IMAH) to be appropriate for modeling the dynamically changing cyberworlds by descending from the most abstract homotopy level to the most specific view level, while preserving invariants. We have developed a data processing system called the Cellular Data System (CDS) based on IMAH. In this paper, we introduce numerical value calculation, exponential calculation and the processing maps on the presentation level of IMAH into CDS. This function is quite effective in business application development because numerical values and exponential identifiers can be put into formulas and be calculated. We show its effectiveness through examples of a calculate system for 1. cellular phone charges, and 2. BOMs (bills of materials) used in manufacturing.

Keywords: incrementally modular abstraction hierarchy, formula expression, numerical value calculation, exponential calculation.

1 Introduction

Cyberworlds are information worlds formed in a cloud either intentionally or spontaneously, with or without design. As information worlds, they are either virtual or real, and can be both. In terms of information modeling, the theoretical ground for the cyberworlds is far above the level of integrating spatial database models and temporal database models. They are more complicated and fluid than any other previous worlds in our history, and are constantly evolving. The number of organizations that conduct business in a cloud is increasing and the market is growing remarkably. On the other

M.L. Gavrilova and C.J.K. Tan (Eds.): Trans. on Comput. Sci. XVI, LNCS 7380, pp. 115–133, 2012.

hand, in general business application system, as the scale of systems becomes larger and the specifications of systems changes more frequently, development and maintenance becomes more difficult, leading to higher costs and delays. In some cases, a huge system as the mainstay system in a large company, where the number of program steps is the hundreds of millions, needs several years to develop. Increases in development and maintenance cost squeeze management. Such situations arise because of combinatorial explosions. The era of cloud computing requires a more powerful mathematical background to model the cyberworlds and to prevent combinatorial explosions. In the cloud, every business object and business logic should be expressed in a unified form to eliminate discontinuity between systems or functions and to meet changes in user requirements. The needed mathematical mechanisms are: 1. disjoint union of spaces by an equivalence relation; 2. change in spaces to preserve invariants; 3. attachment of different spaces by an equivalence relation; 4.space with dimensions as a special case. We consider the Incrementally Modular Abstraction Hierarchy (IMAH) that one of the authors (T. L. Kunii) proposes able to satisfy the above requirements, as it models the architecture and the dynamic changes of cyberworlds from a general level (the homotopy level) to a specific one (the view level), preserving invariants while preventing combinatorial explosion [1]. It also benefits the reuse of information, guaranteeing modularity of information based on the mechanism of disjoint union. Unlike IMAH, other leading data models do not support the disjoint union or the attaching function by equivalence relation. In this research, one of the authors (Y. Seki) proposed a finite automaton called Formula Expression as a development tool to design spaces in IMAH. One of authors (T. Kodama) has actually designed spaces and implemented the data processing system called the Cellular Data System (CDS) using Formula Expression. In this paper, first, we design numerical value and exponential calculations to put numerical values and exponential identifiers in topological spaces on the presentation level, and next implemented them. Using this new design and processing, numerical values are written and calculated in a formula with the other functions of Formula Expression. We demonstrate the effectiveness of the functions by developing general business applications for core processing of calculation system for cellular phone charges and BOMs used in manufacturing, there by abbreviating the process of developing application programs.

2 Literature Review

The distinctive features of our research are the application of the concept of topological process, which deals with a subset as an element, and that the cellular space extends the topological space, as seen in Section 2. Relational OWL as a method of data and schema representation is useful when representing the schema and data of a database [2][5], but it is limited to representation of an object that has attributes. Our method can represent both objects: one that has attributes as a cellular space and one that does not have them as a set or a topological space.

Many works applying other models to XML schema have been done. The motives of most of them are similar to ours. The approach in [8] aims at minimizing document revalidation in an XML schema evolution, based in part on the graph theory. The X-Entity model [9] is an extension of the Entity Relationship (ER) model and converts XML schema to a schema of the ER model. In the approach of [6], the conceptual and logical levels are represented using a standard UML class and the XML represents the physical level. XUML [10] is a conceptual model for XML schema, based on the UML2 standard. This application research concerning XML schema is needed because there are differences in the expression capability of the data model between XML and other models. On the other hand, objects and their relations in XML schema and the above models can be expressed consistently by CDS, which is based on the cellular model. That is because the tree structure, on which the XML model is based, and the graph structure [3][4][7], on which the UML and ER models are based, are special cases of a topological structure mathematically. Entity in the models can be expressed as the formula for a cellular space in CDS. Moreover, the relation between subsets cannot in general be expressed by XML.

3 IMAH and Formula Expression

3.1 The Incrementally Modular Abstraction Hierarchy

The following list is the Incrementally Modular Abstraction Hierarchy to be used for defining the architecture of cyberworlds and their modeling:

1. The homotopy (including fiber bundles) level
2. The set theoretical level
3. The topological space level
4. The adjunction space level
5. The cellular space level
6. The presentation (including geometry) level
7. The view (also called projection) level

In modeling cyberworlds in cyberspaces, we define general properties of cyberworlds at the higher level and add more specific properties step by step while climbing down the Incrementally Modular Abstraction Hierarchy. The properties defined at the homotopy level are invariants of continuous changes of functions. The properties that do not change by continuous modifications in time and space are expressed at this level. At the set theoretical level, the elements of a cyberspace are defined, and a collection of elements constitutes a set with logical calculations. When we define a function in a cyberspace, we need domains that guarantee continuity such that the neighbors are mapped to a nearby place. Therefore, a topology is introduced into a cyberspace through the concept of neighborhood. Cyberworlds are dynamic. Sometimes cyberspaces are attached together, an exclusive union of two cyberspaces where attached areas of two cyberspaces are equivalent. It may happen that an attached space is obtained. These attached spaces can be regarded as a set of equivalent spaces called a quotient space that is another invariant. At the cellular structured level, an inductive dimension is introduced into each cyberspace. At the presentation level, each space is represented in a form which may be imagined before designing cyberworlds. At the view level, the cyberworlds are projected onto view screens. An example of an adjunction space level is shown in Fig. 1.

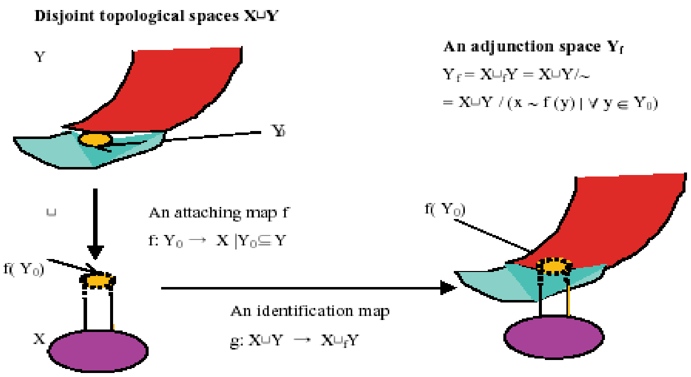


Fig. 1. An example of e-manufacturing on an adjunction space level

3.2 Formula Expression

Outline. Formula Expression is a finite automaton as a communication tool that has been developed in order to guarantee universality in communication between subjects by expressing states of things in a formula. This is very effective in solving the frequent problems arising from misunderstandings between providers and suppliers in business application system development. Formula Expression was invented over a number of years by pursuing the greatest simplicity possible. As a result, spaces could be designed on each level of the cellular model using Formula Expression, as shown in the previous paper [9], while this was not possible using other tools. It is thought that Formula Expression, by following the levels of the cellular model, can reflect any space that humans create and their operations. In other words, it reflects human thought.

The Definition. Formula Expression in the alphabet is the result of finite times application of the following (1)-(7).

(1) a ($\in \Sigma^*$) is Formula Expression
(2) unit element ε is Formula Expression
(3) zero element φ is Formula Expression
(4) when r and s are Formula Expression, addition of r+s is also Formula Expression
(5) when r and s are Formula Expression, multiplication of r×s is also Formula Expression
(6) when r is Formula Expression, (r) is also Formula Expression
(7) when r is Formula Expression, {r} is also Formula Expression

Strength of combination is the order of (4) and (5). If there is no confusion, ×, (), { } can be abbreviated. Unit element ε is expressed as "()" and zero element φ as "{ }" in

Formula Expression, but the characters *ε* and *φ* are used in this paper to prevent misunderstandings.

The Generating Grammar. The grammar which generates Formula Expression is the following.

We assume that Σ_L is a set of ideograms and its element is w ($\epsilon\ \Sigma_L$).

$G = (\{E, T, F, id\}, \Sigma_L \cup \{\varepsilon, \varphi, +, \times, (,), \{, \}\}, P, E)$,
$P = \{E \rightarrow T | E+T, T \rightarrow F | T \times F, F \rightarrow (E) | \{E\} | id, id \rightarrow w\}$

Here, *E* is called a *formula*, *T* is called a *term*, *F* is called a *factor*, *id* is called an *identifier*; + is called a *separator*, which creates a disjoint union (= addition operation) and is expressed as Σ specifically, and × is called a *connecter* which creates a direct product (= multiplication operation) and is also expressed as Π. The Parentheses () mean a set where an order of elements is not preserved and braces { } an ordered set where the order of elements is preserved. In short, you can say "a formula consists of an addition of terms, a term consists of a multiplication of factors, and if () or { } is added to a formula, it becomes recursively the factor".

The Meaning of Formula Expression. The language L(*r*) ($\epsilon\,\Sigma^*$) that Formula Expression *r* expresses is defined as:

(1) $L(a) = \{a\}$ ($a \ \epsilon\, \Sigma^*$)
(2) $L(\varepsilon) = \{\varepsilon\}$
(3) $L(\varphi) = \emptyset$
(4) $L(r+s) = L(r) \cup L(s)$
(5) $L(r \times s) = L(r) \times L(s)$

The Algebraic Structure of Formula Expression. Formula Expression *r, s, t, u* follow the following algebraic structure.

(1) $r+(s+t) = (r+s)+t$, $r\times(s\times t) = (r\times s)\times t$
(2) $r+s = s+r$
(3) $r\times\varepsilon = \varepsilon\times r = r$
(4) $r\times\varphi = \varphi\times r = \varphi$, $r+\varphi = r$
(5) $r\times(s+t) = r\times s+r\times t$, $(r+s)\times t = r\times t+s\times t$

The Examples. Simple examples of spaces created by Formula Expression in the topological space, the adjunction space and the cellular space level, which are more characteristic in the cellular model, are shown with their images in Fig. 2. The spaces of a tree structure, an inverted tree and a graph are a kind of a topological space which deals with a subset as an element.

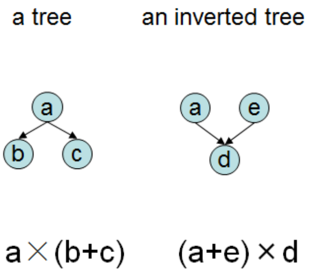


Fig. 2. Spaces on the topological space level

Next, in the adjunction space level, the adjunction space is created after attaching topological spaces by an equivalence relation. Fig. 3 shows that the two spaces in Fig. 2 are attached to create an adjunction space by the equivalent identifier "*a*".

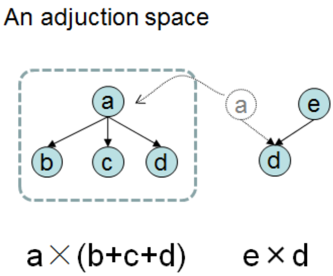
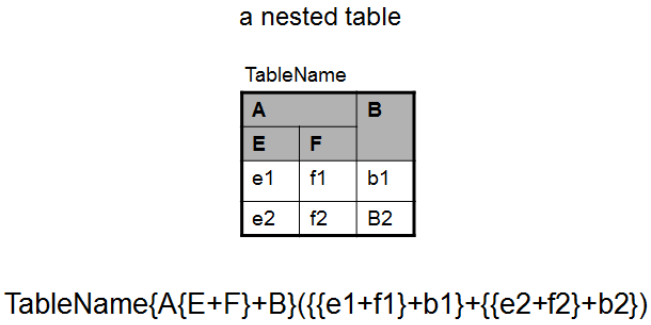


Fig. 3. A space on the adjunction space level

Lastly, in the cellular space level, dimension information, such as the fields of an object or the columns of a table, is designed in the space. The space of a nested table in Fig. 4 is a cellular space.


a nested table
TableName

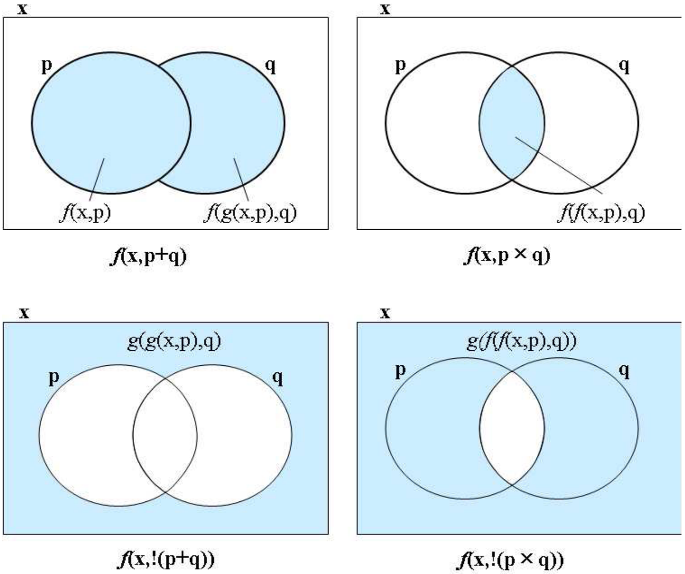
A		B
E	F	
e1	f1	b1
e2	f2	B2

TableName{A{E+F}+B}({{e1+f1}+b1}+{{e2+f2}+b2})

Fig. 4. An example of a space on the cellular space level

3.3 The Condition Formula Processing Map

In Formula Expression, several basic maps were defined in the previous paper [9]. A function for specifying conditions defining a condition formula utilizing basic maps is supported in CDS. This is one of the main functions, and the map is called a condition formula processing map. A formula created from these is called a condition formula. "!" is a special factor which means negation. Recursivity by () in Formula Expression is supported, so that the recursive search condition of a user is expressed by a condition formula. The condition formula processing map *f* is a map that gets a disjoint union of terms that satisfies a condition formula from a formula. When condition formula processing is considered, the concept of a remainder of spaces is inevitable. A remainder acquisition map *g* is a map that has a term that doesn't include a specified factor. Fig. 5 shows each image by the condition formula processing map *f*.



f(x,p+q)

f(x,p × q)

f(x,!(p+q))

f(x,!(p × q))

Fig. 5. Images by the condition formula processing map *f*

4 Numerical Value Calculation and Exponential Calculation on the Presentation Level

4.1 The Properties of Numerical Value Calculation and Exponential Calculation

If we assume that *p, q, r* are arbitrary numerical factors, and that *s, t, u* are arbitrary factors, the numerical value calculation in Formula Expression has the following properties:

(1) $\mathrm{s}\times 1 = \mathrm{s}$
(2) $\mathrm{s}\times 0 = \varphi$
(3) $p\mathrm{s} = \mathrm{s}\times p$
(4) $\mathrm{s}\times p+\mathrm{s}\times q = \mathrm{s}\times(p+q)$
(5) $\mathrm{s}\times p\times \mathrm{t}\times q = \mathrm{s}\times \mathrm{t}\times(p*q)$
(6) $\mathrm{s}\times p = p\times \mathrm{s}$
(7) $\mathrm{s}\times p\ (\mathrm{t}\times q+\mathrm{u}\times r) = \mathrm{s}\times \mathrm{t}\times(p*q)+\mathrm{s}\times \mathrm{u}\times(p*r)$
(8) $(\mathrm{s}\times p+\mathrm{t}\times q)\mathrm{u}\times r = \mathrm{s}\times \mathrm{u}\times(p*r)+\mathrm{t}\times \mathrm{u}\times(q*r)$

And the exponential calculation in Formula Expression has the following properties:

(1) $\mathrm{s}^{0} = \varepsilon$
(2) $\mathrm{s}^{1} = \mathrm{s}$
(3) $/\mathrm{s}^{q} = \mathrm{s}^{-q}$
(4) $\mathrm{s}^{p}\times \mathrm{s}^{q} = \mathrm{s}^{(p+q)}$
(5) $(\mathrm{s}^{p})^{q} = \mathrm{s}^{p\times q}$

4.2 The Numerical Value and Exponential Calculation Map

The numerical value and exponential calculation map *f* is defined based on the above mentioned properties. If you assume the entire set of formulas, including the numerical factors, to be A, *f*: A → A and *f* is the followings:

f: $\mathrm{s} \rightarrow \mathrm{s}\times 1$
f: $p\mathrm{s} \rightarrow \mathrm{s}\times p$
f: $\mathrm{s}\times p+\mathrm{s}\times q \rightarrow \mathrm{s}\times(p+q)$
f: $\mathrm{s}\times p\times \mathrm{t}\times q \rightarrow \mathrm{s}\times \mathrm{v}\times(p*q)$
f: $\mathrm{s}\times p\ (\mathrm{t}\times q+\mathrm{u}\times r) \rightarrow \mathrm{s}\times \mathrm{t}\times(p*q)+\mathrm{s}\times \mathrm{u}\times(p*r)$
f: $(\mathrm{s}\times p+\mathrm{t}\times q)\mathrm{u}\times r \rightarrow \mathrm{s}\times \mathrm{u}\times(p*r)+\mathrm{t}\times \mathrm{u}\times(q*r)$
f: $\mathrm{u}\times p+\mathrm{v}\times q \rightarrow \mathrm{u}\times p+\mathrm{v}\times q$
f: $\mathrm{s}^{0} \rightarrow \varepsilon$
f: $\mathrm{s}^{1} \rightarrow \mathrm{s}$
f: $/\mathrm{s}^{q} \rightarrow \mathrm{s}^{-q}$
f: $\mathrm{s}^{p}\times \mathrm{s}^{q} \rightarrow \mathrm{s}^{(p+q)}$
f: $(\mathrm{s}^{p})^{q} \rightarrow \mathrm{s}^{p\times q}$

And if we assume that T is an arbitrary term, and that E is an arbitrary formula, *f* is:

$f(\mathrm{T}*\mathrm{T}) = f(\mathrm{T})*f(\mathrm{T})$
$f(\mathrm{T}+\mathrm{T}) = f(\mathrm{T})+f(\mathrm{T})$
$f((\mathrm{E})) = (f(\mathrm{E}))$

An example of the map *f* is shown below.

f (cat+dog+rabit+dog+cat+rabit+dog+rabit+mouse)
= cat×2+dog×2+rabit×3+mouse×1

As a further example, when three boxes of cigarettes (which are \$5 per box) are bought, the formula for calculation and processing by the map *f* is:

$$f(\text{cigarette}\times\$5\times/(1\text{box})\times 3\times\text{box})$$
$$=\text{cigarette}\times\$\times 5\times 1\times\text{box}^{-1}\times 3\times\text{box}^{1}$$
$$=\text{cigarette}\times 15\times\$$$

4.3 Implementation

This system is a web application developed using JSP and Tomcat 5.0 as a Web server. The client and the server are the same machine. (OS: Windows XP; CPU: Intel Pentium 3, 1.2GHz; RAM: 1.1Gbyte; HD: 20GB) The following is the coding for the calculation of numerical value and exponential calculation. The focus is the recursive process (line 7, in bold) that is done if a coming numerical calculation is of the type (). The explanation is abbreviated due to space limitations.

```
1    term = null; factor = null;
2    while(factor is not null){
3      term = getTerm(factor);
4      while(term is not null){
5              factor = getFactor(term)
6              if(factor is of the type ()){
7                      factor = calculate(the contents);
               }
8              factor = getNumericalFactor(factor);
9              LetteFactor = getLetteFactor(factor);
10             newNF = newNF×NumericalFactor;
11             newLF = newLF×LetteFactor;
       }
12     newTerm = newNF + newLF;
13     newFormula = newFormula + newTerm;
     }
14   return newFormula;
```

5 Development of a Cellular Phone Charge Calculation System

5.1 Outline

We take up the simple example of a cellular phone charge calculation system to secure generality, because charge calculation systems are generally developed in most industries. The most important thing with a charge calculation system is to be able to deal with unexpected changes of users' charge plans or of user contract data. If a cellular phone company is going to introduce new charge plans for the cellular phone to further their strategy of gaining more cellular phone subscribers, the development of an application program to calculate charges will cost a lot and take a lot of time, because the program needs to be consistent with previous data/programs, and user contract data will change according to the new charge plans. To solve these difficulties, we apply CDS to the development of core processing of the cellular phone charge calculation system. Firstly, each formula for contract data and plan type data is designed as a topological space. Secondly, formulas from January to December of a year, where basic charges or discount rates are expressed using numerical values and exponential calculation, are created according to the design. Every time charge plan or user contract data is changed over time, another topological space is created and added, forming a disjoint union of topological spaces. Thirdly, each charge for use of the cellular phone can be calculated using the maps of CDS, such as the condition formula processing map (3.3) or the numerical value and exponential calculation map (4.1,4.2). In these designs, numerical value calculation, exponential calculation, and the calculation map are used to express unit price per month, duration of use, discount rates and the calculations among them. Here, actual data and functions are simplified to focus on verifying development of core processing without losing generality.

5.2 The Design of Topological Spaces

We design a formula for topological spaces for (1) the user contract data, and (2) the charge plan data. Numerical value identifiers are used to express unit price per month, duration of use, and discount rates. The formulas for (1) and (2) are designed as follows:

(1) CONTRACT$\{\Sigma$ *user id*$_i\}\{\Sigma$ *charge plan type*$_j\}(\Sigma$ *month*$_k)$

user id$_i$: a factor which identifies a cellular phone user
charge plan type$_j$: a factor which shows a charge plan type of a cellular phone
month$_k$: a factor which shows a month of use of a cellular phone

(2) PLAN$\{\Sigma$ *charge plan type*$_i\}\{\Sigma$ *unit price*$_j$ *per month*$\}(\Sigma$ *month*$_k)$

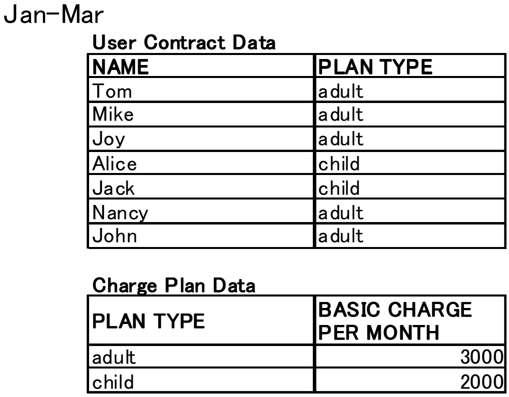
unit price$_j$ *per month*: a factor which shows unit price per month of charge

5.3 Data Input According to the Design

-January to March of the Year-

First, for the period January–March, assume that seven users (Tom, Mike, Joy, Alice, Jack, Nancy, John) sign contracts for the use of cellular phones and that there are two plan types: 1. “adult” whose basic fee is “3000yen” per month and 2. “child” whose basic fee is “2000yen” per month, as seen in Fig. 6. You create the following formula for the topological space (formula C-1), according to the above design in B, and add it to data storage.

formula C-1:
CONTRACT{Tom+Mike+Joy+Alice+Jack+Nancy+John}{adult+adult+adult+child+child+adult+adult}(Jan+Feb+March)+PLAN{adult+child}{3000yen/m+2000yen/m}(Jan+Feb+March)



Jan–Mar

User Contract Data

NAME	PLAN TYPE
Tom	adult
Mike	adult
Joy	adult
Alice	child
Jack	child
Nancy	adult
John	adult

Charge Plan Data

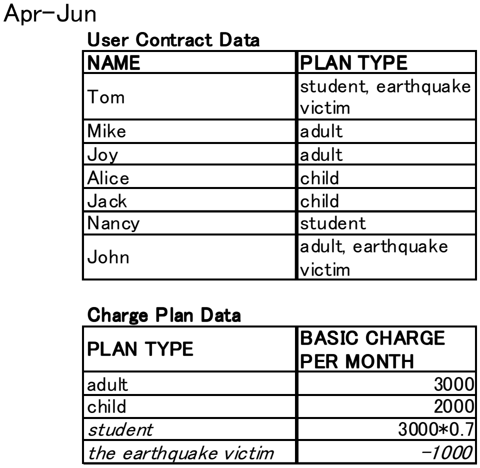
PLAN TYPE	BASIC CHARGE PER MONTH
adult	3000
child	2000

Fig. 6. User contract data and charge plan data from January through March

-From April through June-

Next, for the period April-June, assume that a cellular phone operator introduces a new charge plan type, "student", whose basic fee is 30% off the "adult" fee, and that Tom and Nancy qualify as students, as seen in Fig. 7. Moreover, assume that a special discount (1000yen off per month) for earthquake victims is offered, and that Tom and John qualify. You create the following formula for the topological space (formula C-2), according to the above design, and add it to the previous formula.

formula C-2:
(formulaC-1)+CONTRACT{Tom×*0.5*+Mike+Joy+Alice+Jack+Nancy+John×*0.5*}
{*student*+adult+adult+child+child+*student*+adult}(April+May+June)+PLAN{adult+
child+student}{3000 yen /m+2000 yen /m+3000yen/m×*0.7*}(April+May+June)



Apr–Jun

User Contract Data

NAME	PLAN TYPE
Tom	student, earthquake victim
Mike	adult
Joy	adult
Alice	child
Jack	child
Nancy	student
John	adult, earthquake victim

Charge Plan Data

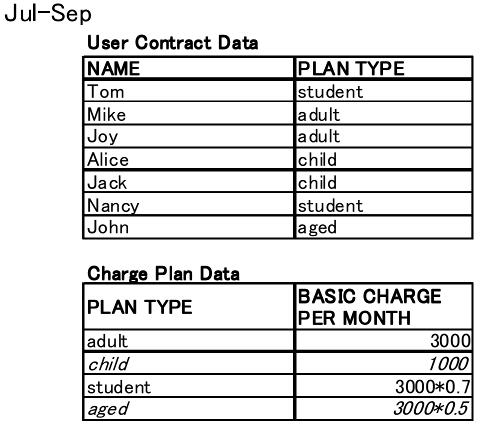
PLAN TYPE	BASIC CHARGE PER MONTH
adult	3000
child	2000
student	3000*0.7
the earthquake victim	*-1000*

Fig. 7. User contract data and charge plan data from April through June

-From July through September-

Next, for the period July-September, assume that cellular phone operator introduces a new charge plan type, “*aged*”, for elderly people, whose basic fee is 50% off the “adult” fee, that John qualifies for the plan, and that the fee for the “child” plan changes from “2000yen/m” to “1000yen/m” , as seen in Fig. 8. You create the following formula for the topological space (formula C-3), according to the design, and add it to the previous formula.

formula C-3:
(formula C-2)+CONTRACT{Tom+Mike+Joy+Alice+Jack+Nancy+John}{student+adult+adult+child+child+student+aged}(July+August+September)+PLAN{adult+child+student+aged}{3000yen/m+*1000yen/m*+3000yen/m×0.7+3000yen/m×0.5}(July+August+September)



Jul-Sep

User Contract Data

NAME	PLAN TYPE
Tom	student
Mike	adult
Joy	adult
Alice	child
Jack	child
Nancy	student
John	aged

Charge Plan Data

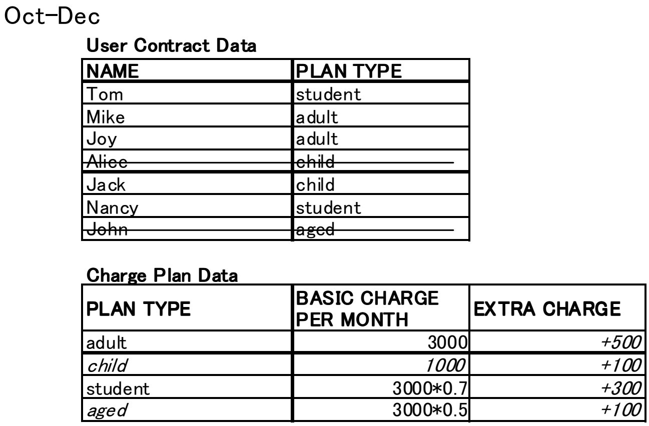
PLAN TYPE	BASIC CHARGE PER MONTH
adult	3000
child	*1000*
student	3000*0.7
aged	*3000*0.5*

Fig. 8. User contract data and charge plan data from June through September

-From October through December-

Finally, for the period October-December, assume that a cellular phone operator adds an extra charge for all users, as seen in Fig. 9, and that Alice and John cancel their contracts. You create the following formula for the topological space (formula C-4), according to the design, and add it to the previous formula.

formula C-4:
formula C-3)+CONTRACT{Tom+Mike+Joy+Jack+Nancy}{student+adult+adult+child+student}(Oct+Nov+Dec)+PLAN{adult+child+student+aged}{(3000yen/m+500yen/m)+(1000yen/m+100yen/m)+(3000yen/m×0.7+300yen/m)+(3000yen/m×0.5+100yen/m)}(Oct+Nov+Dec)



Oct–Dec

User Contract Data

NAME	PLAN TYPE
Tom	student
Mike	adult
Joy	adult
~~Alice~~	~~child~~
Jack	child
Nancy	student
~~John~~	~~aged~~

Charge Plan Data

PLAN TYPE	BASIC CHARGE PER MONTH	EXTRA CHARGE
adult	3000	*+500*
child	*1000*	*+100*
student	3000*0.7	*+300*
aged	3000*0.5	*+100*

Fig. 9. User contract data and charge plan data from October through December

5.4 Data Output

If a user wants to answer the question “How much does Tom pay in fees during the year?”, firstly you can get the image of formula C-4 by “CONTRACT×Tom×(Jan+Feb+Mar+Apr+May+Jun+July+Aug+Sep+Oct+Nov+Dec)” through the condition formula processing map *f*. The result is below.

formula D-1:
CONTRACT×Tom(*adult×(Jan+Feb+Mar)+(student+earthquake victim)× (Apr+May+Jun)+student×(July+Aug+Sep)+student×(Oct+Nov+Dec)*)

Secondly, from the result, you make the following condition formula.

“Plan×*(adult×(Jan+Feb+Mar)+(student+earthquake victim)×(Apr+May+Jun)+ student×(July+Aug+Sep)+student×(Oct+Nov+Dec))”*

And you get the image of formula C-4 by the condition formula through the map *f* again. The result is below.

formula D-2:
PLAN(adult×3000yen/m×(Jan+Feb+Mar)+(student×3000yen/m×0.7+earthquake victim×-1000yen/m)×(Apr+May+Jun)+student×3000yen/m×0.7(July+Aug+Sep)+ student×(3000yen/m×0.7+300yen/m)(Oct+Nov+Dec))

Thirdly, you replace “(Jan+Feb+Mar)”, “(Apr+May+Jun)”, “(July+Aug+Sep)” and “(Oct+Nov+Dec)” by “3×m”, which means "three months", in formula D-1, and calculate it by the numerical value and exponential calculation map *g*.

g(PLAN(adult×3000yen/m×3×m+(student×3000yen/m×0.7+earthquake victim ×-1000yen/m)×3×m+student×3000yen/m×0.7×3×m+student×(3000yen/m×0. 7+300yen/m)×3×m))

=PLAN(adult×yen×9000+(student×yen×6300+the earthquake victim×yen×-3000)+student×yen×6300+student×yen×7200))
=PLAN{adult+(student+earthquake victim)+student+student}{yen×9000+(yen×6300+yen×-3000)+yen×6300+yen×7200}
=PLAN{adult+(student+earthquake victim)+student+student}×yen×25800

From the result, you can know that the answer is 25,800yen.

Next, if a user wants to answer the question "How much do all users pay in fees in July?", firstly you can get the image of formula C-4 by "CONTRACT×Jul" through the map *f*. The result is below.

formula D-3:
CONTRACT{Tom+Mike+Joy+Alice+Jack+Nancy+John}{student+adult+adult+child+child+student+aged}July

Secondly, in the same way, from the result you make the following condition formula.

"Plan×*(student+adult+adult+child+child+student+aged)July"*

And you get the image of formula D-3 by the condition formula through the map *f* again. The result is below.

formula D-4:
PLAN(student×3000yen/m×0.7+adult×3000yen/m+adult×3000yen/m+child×1000yen/m+child×1000yen/m+student×3000yen/m×0.7+aged×3000yen/m×0.5)×July

Thirdly, in the same way, you replace "July" by "1×m", which means "one month", in formula D-4, and calculate it by the map *g*.

g(PLAN(student×3000yen/m×0.7+adult×3000yen/m+adult×3000yen/m+child×1000yen/m+child×1000yen/m+student×3000yen/m×0.7+aged×3000yen/m×0.5)×(1×m))
=PLAN(student×ye×3000×0.7×2+adult×yen×3000×2+child×yen×1000×2+aged×yen×3000×0.5)
=PLAN{student+adult+child+aged}{yen×3000×0.7×2+yen×3000×2+yen×1000×2+yen×3000×0.5}
=PLAN{student+adult+child+aged}×yen×13700

From the result, you can know that the answer is 13,700yen.

5.5 Considerations

In general, it costs a lot to develop and maintain application programs for the cellular phone charge calculation system because of the complexity and frequent changes in the business services. This example shows that every time the service changes, if you only have to create new formulas for the user contract data and the charge plan data according to the formula design, you can get the output using the processing map, thereby reducing the amount of application development and maintenance. This is mainly because of:

1. the mechanism of the topological processing function of CDS, which deals with subsets as an element

In the above example, from April through June, some users can belong to multiple plan types (Tom qualifies for two plan types, “the earthquake victim” and “student”, and John also qualifies for two plan types, “the earthquake victim” and “adult”).

2. numerical values and exponential identifiers that can be put into formulas and calculated according to the definition of numerical and exponential calculation

In the above example, unit price per month and duration of use are calculated using numerical and exponential calculations.

6 Development of a Core Calculation System of BOM (Bill of Materials) in Manufacturing

6.1 Outline

We take up an example of a core calculation system of BOM (Bill of Materials). A BOM is a list of the parts or components that are required to build a product in manufacturing. When the component configuration of a product changes frequently, such as division of a part, integration of parts or changes in the name of a part, it is difficult to manage configuration information consistently. This is because the application programs of a system have to be developed every time the component configuration changes. To solve the difficulties, we have applied CDS to the development of a core calculation system for BOMs. Firstly we make a general design of a topological space for one part of a product and the changes using CDS. Secondly, formulas for the parts in each phase are created. Thirdly, the formulas are processed by the maps of CDS to meet user requirements. Here, data is also simplified without losing generality.

6.2 The Design of a Formula for a Topological Space for Parts

We design a formula for topological spaces for one part of a product $P_{n,m}$ in phase$_n$. $P_{n,m}$ is defined as the multiplication of a part id ($id_{n,m}$), the disjoint union of topological spaces for the subordinate parts ($\Sigma P_{i,m-1}$) and the quantity of the part $num_{n,m}$ where a numerical value identifier is used. The formula for $P_{1,\, m}$ is designed as follows:

$$P_{1,\, m} = \text{a}\, \textit{part id}_1 \times (\Sigma\, P_{i,m-1}) \times num_n$$
where $P_{1,0} = \text{a}\, \textit{part id}_i$

a part id$_n$: a factor which identifies one part of a product
$num_{\,n}$: a factor which shows the quantity of a part

We take up a simple example of $P_{1,\, 2}$ (the 1st phase and 2 layers), whose part id is A and which consists of the subordinate nodes of three B parts, two C parts, and five D parts, as seen in Fig. 10. The formula is as follows:

$$P_{1,\, 2} = A(B\times 3+C\times 2+D\times 5)\times 1$$

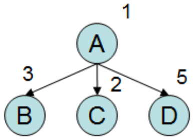


Fig. 10. A component configuration example of a product

Next, we assume that the phase of a product proceeds to the next phase when the configuration of the parts in the produce changes. We define the formula for a topological space for the product $P_{n+1, m}$, removing old nodes and adding new nodes when the nodes (*old nodes*) in P_n change to others (*new nodes*). The formula is as follows:

$$P_{n+1,m} = P_{n,m}+(old\ nodes\times -1+new\ nodes\times 1)\times o$$

We take up a simple example of $P_{2,\,2}$ (the 2st phase and 2 layers), in which the name of the part C in Fig. 10 changes to E, as seen in Fig. 11. The formula is as follows:

$$P_{2,\,2} = A(B\times 3+C\times 2+D\times 5)\times 1+(A\times C\times -1+A\times E\times 1)\times 2$$

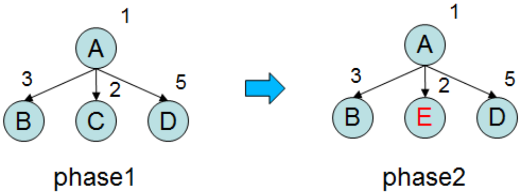


Fig. 11. A change example of a component configuration

6.3 Data Input According to the Design

In phase 1, we assume the component configuration of a product as seen in Fig. 12. According to the above design, the formula for P_1 is created as follows:

$$P_{1,3}=A(B(E1\times o+(E2+E3\times 2)\times p)\times n+C(E1\times r+E2\times s+E3\times 2s)\times l+D\times m)\times 1$$

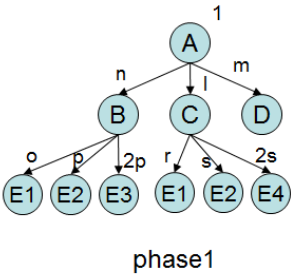


Fig. 12. Component configuration of a product in phase 1

Next, we assume that the parts E2 and E3 are integrated into parts F in phase 2 as seen in Fig. 13. In the same way, the formula for P_2 is created as follows:

$$P_{2,3}=(A(B(E1\times o+(E2+E3\times 2)\times p)\times n+C(E1\times r+E2\times s+E3\times 2s)\times l+D\times m))+ (A\times B\times(E2+E3\times 2)\times -1+A\times B\times F\times 1)\times p\times n$$

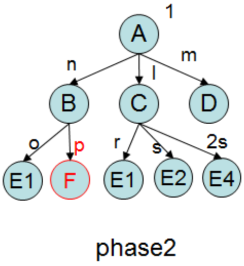


Fig. 13. Component configuration of a product in phase 2

Thirdly, we assume that the parts of F are divided into parts G1, G2 and G3 in phase 3 as seen in Fig. 14. In the same way, the formula for P_3 is created as follows:

$$P_{3,3}=(A(B(E1\times o+(E2+E3\times 2)\times p)\times n+C(E1\times r+E2\times s+E3\times 2s)\times l+D\times m))+(A \times B\times(E2+E3\times 3)\times -1+A\times B\times F\times 1)\times p\times n+(A\times B\times F\times -1+A\times B\times(G1+G2+ G3)\times 1)\times p\times n$$

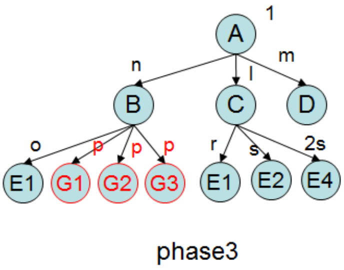


Fig. 14. Component configuration of a product in phase 3

6.4 Data Output

If a user wants to answer the question “What is the structure of parts in phase 3?”, you calculate P_3 using calculation map g (4.2) and get the image of it. The result is below.

$$A(B(E1\times o+(G1+G2+G3)\times p)\times n+C(E1\times r+E2\times s+E3\times 2s)\times l+D\times m)$$

From the result, you know the structure of parts in phase 3.

Next, if a user wants to answer the question "How many pieces of E2 are there in the phase 2?", you cut the last term of P_3 to be returned to P_2 and get the image of P_2 by the factor "*E2*" through condition formula processing map *f* (3.3). Lastly, you calculate it using calculation map *g* (4.2). The result is below.

$$A \times C \times E2 \times (s*l)$$

From the result, you can know that there are ($s*l$) pieces of E2 in phase 2.

6.5 Considerations

In this example, the changes in component configuration are generally designed as a disjoint union of topological spaces using numerical value factors in addition to the design of a topological space for parts using CDS. Therefore, information about the parts in a phase is expressed as the union of information about changes in the parts from an initial value of parts, such that P_n is accumulated information about change information from P_1. The information can then be processed consistently by the maps of CDS, such as the condition formula processing map or the calculation map, according to user requirements. This modeling can reduce the amount of development of application programs and system maintenance.

7 Conclusions

We have developed a data processing system called the Cellular Data System (CDS) based on IMAH. In this paper, we introduced numerical value and exponential calculation and the processing maps on the presentation level of CDS. Using the function in business application development, you can put numerical values and exponential identifiers into formulas as identifiers, and they can be calculated based on definition, so that system development become simpler, business processes more visible, and business applications more flexible to changes in business conditions, as shown in section 5. As a result, use of CDS can make developers more creative, preventing from frequent troubles between the customer side and the supplier side, while preventing combinatorial explosions.

The fruits of this research have already been put to practical use in many companies, and further strides are being made every day. We are sure that CDS has possibilities to bring great social impact in the era of cloud computing.

References

[1] Kunii, T.L., Kunii, H.S.: A Cellular Model for Information Systems on the Web - Integrating Local and Global Information. In: Proc. of DANTE 1999, pp. 19–24. IEEE Computer Society Press (1999)

[2] Antoniou, G., van Harmelen, F.: Web Ontology Language: OWL. In: Handbook on Ontologies, International Handbooks on Information Systems, Part 1, pp. 91–110. Springer (2009)

[3] Lukichev, S.: Improving the quality of rule-based applications using the declarative verification approach. International Journal of Knowledge Engineering and Data Mining 1(3), 254–272 (2011)

[4] Embley, D.W.: Semantic priming in a cortical network model. Journal of Cognitive Neuroscience 21(12), 2300–2319 (2009)

[5] Vysniauskas, E., Nemuraite, L.: Transforming ontorogy representation from owl to relational data. Information Technology and Control 35, 333–343 (2006)

[6] An, Y., Mylopoulos, J., Borgida, A.: Building semantic mappings from databases to ontologies. In: Proc. of AAAI 2006, pp. 1557–1560. AAAI Press (2006)

[7] Dolev, S., Schiller, E.M., Spirakis, P.G., Philippas, P.: Strategies for repeated games with subsystem takeovers implementable by deterministic and self-stabilising automata. International Journal of Autonomous and Adaptive Communications Systems 4(1), 4–38 (2011)

[8] Mlynkova, I., Pokorny, J.: From XML schema to object-relational database –an XML schema-driven mapping algorythm. In: Proc. of IADIS International Conference WWW/Internet 2004, pp. 115–122. IADIS Press (2004)

[9] Lósio, B.F., Salgado, A.C., do Rêgo GalvĐo, L.: Conceptual Modeling of XML Schemas. In: Proc. of WIDM 2003, pp. 102–105. ACM Press (2003)

[10] Mellor, S.J., Balcer, M.J.: Executable UML: A Foundation for Model Driven Architecture. Addison Wesley (2002)

[11] Arni, F., Ong, K., Tsur, S., Wang, H., Zaniolo, C.: The Deductive Database System *LDL*++. In: Theory and Practice of Logic Programming, pp. 61–94. Cambridge University Press (2003)

[12] Kodama, T., Kunii, T.L., Seki, Y.: A New Method for Developing Business Applications: The Cellular Data System. In: Proc. of CW 2006, pp. 65–74. IEEE Computer Society Press (2006)

[13] Ohmori, K., Kunii, T.L.: Designing and modeling cyberworlds using the incrementally modular abstraction hierarchy based on homotopy theory. The Visual Computer: International Journal of Computer Graphics 26(5), 297–309 (2010)

Analysis of Inverse Snyder Optimizations

Erika Harrison, Ali Mahdavi-Amiri, and Faramarz Samavati

University of Calgary, Calgary AB, Canada

Abstract. Modern area preserving projections employed by cartographers and geographers have closed forms when transitioning between the sphere and the plane. Inversions - from the planar map to the spherical approximation of the Earth - are slower, requiring iterative root finding approaches or entirely undetermined. Recent optimizations of the common Inverse Snyder Equal Area Polyhedral projection have been fairly successful, however the work herein improves it further by adjusting the approximating polynomial. An evaluation against the original and improved optimizations is provided, along with a previously unexplored real-time analysis.

Keywords: equal area, projection, optimization, Snyder projection.

1 Introduction

The construction of maps which preserve area has long been discussed within the cartographic community. Such a property is particularly desirable for the scientific community. Researchers exploring migratory or meteorological patterns, disease proliferation and control, and many other fields of study rely on accurate areal and regional information. Even business analysts, exploring optimal market demographics and product distribution benefit from area preserving representations of the Earth. These digital environments act as models, or virtual cyberworlds, upon which important analysis may be performed. As such, when representing their spatially-aware data, it is imperative to preserve area when translating between the spherical and planar maps these cyberworlds maintain.

Traditionally projections transform a point p on the Earth, to a point p' on the map (Figure 1), often preserving a variety of properties. Due to the sphericity of the Earth, area and angle - or shape - are unable to be simultaneously preserved [2]. The objective, then, becomes the absolute preservation of area, while reducing shape distortion. In this way, the resultant data appears to be of the correct form, but guarantees accurate regional calculations. Through the novel use of a polyhedral surface, rather than a flat map, Snyder's [14] polyhedral projection is extremely successful at reducing angular distortion, and has been recommended for equal area projections [8].

The inverse process - transforming from the planar representation to the sphere - is of particular important, especially when working within a digital environment. The visualization process tends to work with the planar maps and images, but prefers a spherical visualization - as is the case for Google Earth [4]. As a result, the inverse projection becomes particularly important.

M.L. Gavrilova and C.J.K. Tan (Eds.): Trans. on Comput. Sci. XVI, LNCS 7380, pp. 134–148, 2012.

Efficiency of both the forward and inverse projection is of extreme importance. For the visualization of a single lake, for example, a thousand points may require projection. If hundreds of lakes are evaluated, over a million points must be transformed. Park boundaries, roads and cities may also be projected - often simultaneously - to meet the needs of businesses, military planners, and scientific researchers. An efficient approach must be taken to meet their real-time needs as motivated by geoscience visualization companies [12]. Consequently, an operationally effective and memory efficient approach for an inverse projection is desired.

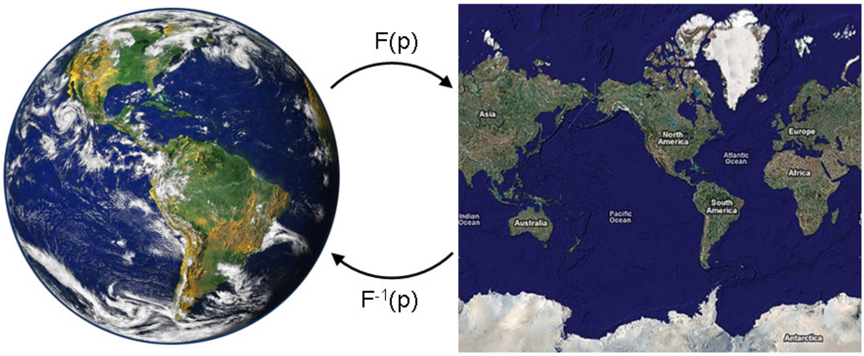


Fig. 1. Projection from the Earth to Planar Map and Inverse (Right: Blue Marble, NASA [11], Left: Mercator Projection, Google Maps [5])

Snyder's equal area approach defines both a forward and inverse projection. His unique approach - employing a polyhedron - and its resultant reduced angular distortion, makes it highly desirable within the modern visualization community. Area is maintained by applying the Lambert Azimuthal Equal Area projection [9] to each of the respective faces, and then adjusted slightly to ensure accurate edge matching. Through a collection of trigonometric equations, function $F(p)$ is defined. Its inverse, $F^{-1}(p)$, is computed by Snyder as a direct reversal of the forward projection. Unfortunately, due to the trigonometric equations, a non-linear system must be solved. Since neither Snyder nor traditional evaluations are able to construct an analytical or closed form, numerical technique is employed.

As a consequence of this non-linear system, the computational time of solving this non-linear system is a large bottleneck. While the closed form of the forward projection has a reliable computation time by calculating each of the equations only once, the numerical technique for finding solutions for the inversion causes an indeterminate number of repeated calculations before converging to a desirable solution. Since these iterations are applied for each of the potentially millions of inverse projection calls, the visualization and analysis process can be immensely impeded.

Few other approaches preserve area, while reducing shape distortion as successfully as Snyder does. Leeuwen et al.'s Slice and Dice projection [10] offer a potential alternative, however it does not define an inverse process, let along a computationally efficient one.

Harrison et al. [7] were able to improve Snyder's inverse process by reducing the iterations to a one dimensional approximating curve. This improved initial estimate reduced the convergence time, while still retaining precise area. An alternative direct replacement of the iterative process with this approximating curve resulted in a 45% faster calculation, with a displacement error of 5.9 m, and areal error of 0.7 km^2. Expanding on this success, we have evaluated alternative approximating curves, and analyzed the real-time improvements.

We start with a brief discussion of the Snyder equal area polyhedral projection, and its inversion. The optimizations from Harrison et al. are reviewed. Our expansions are then discussed. The different polynomial approximating curves are evaluated, a real-time analysis is explored, and finally results are presented accordingly.

2 Background

As computers have improved in power over the years, they have increasingly been incorporated into different fields of study. As late as 1998, former US Vice President, Al Gore proposed a Digital Earth framework [6]. The objective has been to harness the capabilities of computing power to visualize and analyze the Earth's surface directly upon a 3D spherical representation. Consequently, planar input, such as satellite imagery, field surveys and other such 2D data must be transformed into their respective spherical coordinates. While one way to accomplish this is through the employment of accurate Geographic Information System protocol, this is not applicable to large regional input, such as satellite photography. Instead, data must be projected to the appropriate position. These projections, between the planar map and the Earth, range in shape and style, objective and aesthetic, developed continuously over the last two thousand years [15].

These cartographic projections are thereby tasked with overcoming the problems faced with spherical to planar transference. In particular, the inability to simultaneously preserve angle and area results in projections constructed and employed based on the desired qualities of the specific task [2]. Projections are often selected based on the need for preservation of area, preservation of shape, distance and positional accuracy, and an ease of computation [13]. With the prevalence of computers, this ease of computation becomes less critical to cartographers. However, within a Digital Earth framework, requiring immense quantity of data, and a desired real-time analysis, computational complexity remains of high importance.

Researchers and analysts exploring regional-based information, require preservation of area. Equal area projections have been documented since Ptolemy's *Geographia* manuscripts in the second century [15], often incorporating other desirable characteristics. The Werner projection of the 16th century, for example, is able to preserve a collection of longitudinal distances, at the expense of severe shape distortion [15]. The Lambert Equal-Area projections - azimuthal and cylindrical - present the Earth's surface in a less distorted rectangular format. While still employed today [3,16], it is unable to achieve the distance preservance of Werner's. The Mollweide projection retained areal equivalence while reducing the angular distortion within the interior through its ellipsoidal shape [15]. Unfortunately, distortion is still exhibited along the boundaries and non-uniform across the projection.

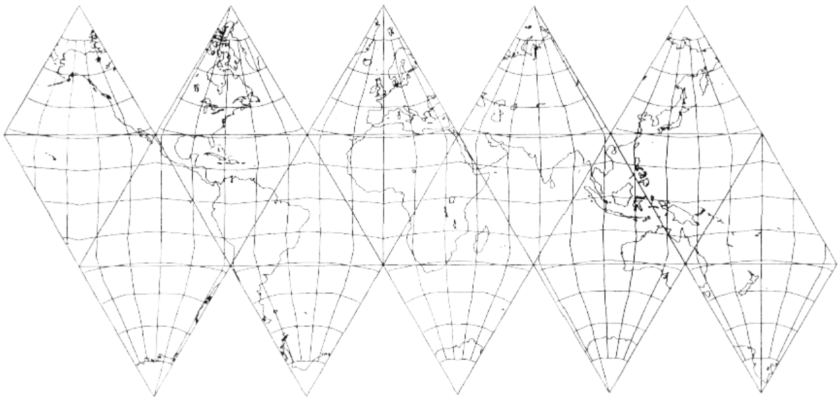


Fig. 2. Icosahedral Mapping Using Snyder Equal Area Projection [14]

As a result of employing a polyhedral surface for his projection, at the expense of a discontinuous flattened map (Figure 2), Snyder [14] was able to reduce the angular distortion due to the polyhedron's close approximation to a sphere. For example, an icosahedron achieves an angular deformation of less than 17.3^o and a scale variation of less than 16.3%. Furthermore, the employment of a polyhedron more readily facilitates its visualization since a polyhedron, or mesh-based, approach is commonly employed within computer visualization.

Alternative approaches have been presented. Leeuwen et al. [10], for example, later demonstrated an alternative equal areal polyhedral projection, more uniformly distributing the angular deformation across the surface. Their Slice and Dice approach is initially constructed comparably to Snyder's, but instead of employing a modified Lambert Azimuthal equal area projection, they partition the surfaces so as to preserve areal ratios. As a result, the distortion is less noticeable, eliminating discontinuities and reducing cusps. Unfortunately, an inverse projection is neither presented nor readily determined - a necessity for computational visualization.

In order to guarantee a real-time visualization, it is important to discuss what it means for a visualization to be real-time. For an interactive system, which responds readily to an individual's input, a visualization must redraw itself at a minimum rate of 24 frames per second (fps). For 24 fps, this corresponds to 0.04167 seconds of screen time per frame. For items that are drawn on the screen, it then becomes a matter of how many items, and for what intensity of computational processing are we able to achieve within the 0.04167 seconds. A review of the original Snyder inversion process is explored, and the number of inverse calls possible within the 0.04167 seconds is determined.

3 Snyder's Polyhedral Projection

An initial discussion of Snyder's polyhedral projection is presented. This is followed by the inversion and its drawbacks.

3.1 Snyder Projection

Snyder's projection defines a function F mapping the spherical point p and computing its position on the polyhedron. To do so, the Lambert Azimuthal Equal-Area projection is centered upon each of the respective faces. In this Lambert Azimuthal Equal-Area projection, a plane is set tangent to the sphere, and points are projected along radial arcs down to this plane. For the Snyder projection, it must be modified for the polyhedral employment to ensure precise edge matching. F is construct by first decomposing the polyhedral faces into their smallest symmetric region - always a right triangle (Figure 3). Then, the area on the plane and on the sphere are made equivalent through a scaling factor between the radius of the sphere, and the radius of the polyhedron's inscribing sphere. The third step defines a triangle on the polyhedron whose area exactly matches that of a spherical triangle bounded by point P. Finally, the new point P' is positioned along this triangle's edge while maintaining areal scale.

Figure 3 visualizes the triangles and their variables. As illustrated, the main face is divided into three subtriangles, and then further halved into right triangles. Such a division may be applied to any regular polygon, and Snyder includes the equations for alternative sphere-circumscribing polyhedra.

Figure 3 additionally illustrates several angles and vertices used through the projection of point P to point P'. Spherical triangle $\triangle ABC$ and polyhedral triangle $\triangle A'B'C'$ are constructed from the underlying polyhedral face with A, A' the centroids, B, B' the vertices, and C, C' each edge's midpoint. During area preservation, radius R of the spherical polyhedron is associated with radius R' of the sphere circumscribing the polyhedron. D is defined using a great circle arc from A through P, intersecting BC. The resulting $\triangle ABD$, with angles $\angle G$, $\angle H$ and $\angle Az$, determine $\triangle A'B'D'$ with the same area. Angle or azimuth $\angle Az'$ is used to determine $\triangle A'B'D'$, which in turn computes point D'. In the final step, the ratios between arc length $q = AD$ and edge length $d' = A'D'$ fix P' in place. It should be noted that angles $\angle \Theta$ and $\angle G$ are fixed for a given polyhedron. These values are listed in Snyder's paper [14].

Having defined these variables, they can then be used to more precisely recreate the Snyder projection. The initial formulation of Snyder's scaling based on the radius R and R' is well discussed by both Snyder [14] and Harrison et al. [7].

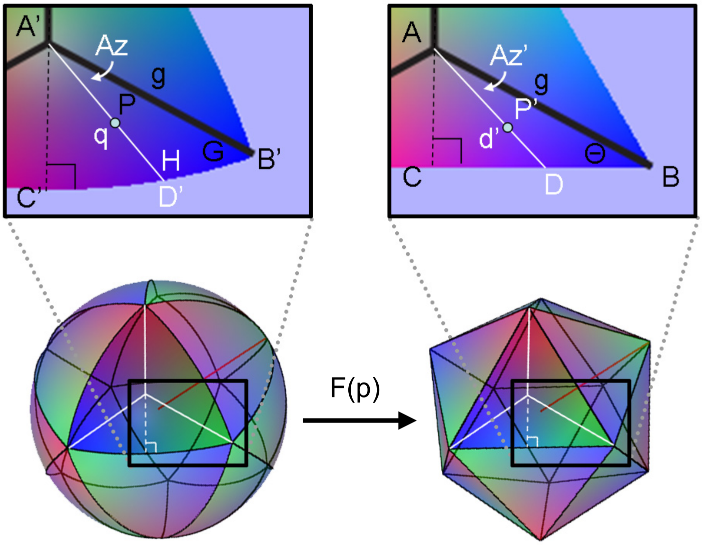


Fig. 3. Spherical and Planar Icosahedron with Symmetric Decomposition (Red line indicates the radius)

The subsequent step ensures the point retains area during throughout the projection, and therefore must position it precisely within the planar triangle. To do so, we must position point D - which may be represented through the calculation of $\angle H$ - from the Spherical Law of Sines and Cosines as follows:

$$\angle H = \arccos(\sin Az \sin G \cos g - \cos Az \cos G), \tag{1}$$

where g is the arclength between AB. Consequently, the area of $\triangle ABD$ is:

$$A_{ABD} = \frac{(Az + G + H - 180^o)\pi R^2}{180^o}. \tag{2}$$

To associate triangle $\triangle A'B'D'$ with its circumscribing radius, and angles of interest, Snyder defines the area as:

$$A_{A'B'D'} = \frac{(R' \tan g)^2 \tan Az'}{2(\tan Az' \cot \Theta + 1)}. \tag{3}$$

Since we need $A_{ABD} = A_{A'B'D'}$, we transform equations 2 and 3 to define our planar azimuth, Az':

$$Az' = \arctan(2A_{ABD}R'^2 \tan^2 g - 2A_{ABD} \cot \Theta).$$

The final step positions point P' along this calculated azimuth, Az' so that it preserves areal scale. The proportionality factor is described in detail by Snyder [14] and Harrison et al. [7]. At this stage, the planar triangle coordinates (x, y) for our projected point is determined.

It should be mentioned that the calculations presented by Snyder are strongly tied to the specific layout of the flattened polyhedron. A visualization of the layout, along with a table of offset coordinates are provided within his paper.

Though this forward projection requires several trigonometric calls, it is otherwise a straightforward closed form.

3.2 Inverse Snyder Projection

The inversion reverses the forward projection, finding the spherical coordinates of P given the coordinates of P' on the polyhedron. From the forward projection, symmetric extraction and radius scaling requires nominal modification. The final reversal of the forward projection - positioning P along great circle arc AD - is also straightforward. The complexity resides within the calculation of Az. In matching the areas of $\triangle ABD$ and $\triangle A'B'D'$, we have Az' and must compute Az. Thus, we can define the area of $\triangle A'B'D'$ as:

$$A_{A'B'D'} = \frac{R'^2 \tan^2 g}{2(\cot Az' + \cot \Theta)}.$$

Setting this equal to the area of $\triangle ABD$, from equation 2, it can be noted that Az is involved linearly and trigonometrically, through the reliance of $\angle H$ on the arccos of $\sin Az$ and $\cos Az$ by equation 1. Solving for Az results in a nonlinear equation. Since a closed form is neither proposed nor easily determined, Snyder suggests the Newton-Raphson iterative approach to deduce an adequate value [14]. This approach computes the derivative and uses it to iteratively find an improved approximate solution. Consequently, the following equations are used:

$$g(Az) = \frac{180^o A_{A'B'D'}}{\pi R^2} - G - H - Az + 180^o \tag{4}$$

$$g'(Az) = \frac{\cos Az \sin G \cos g + \sin Az \cos G}{\sin H} - 1 \tag{5}$$

$$\Delta Az = -\frac{g(Az)}{g'(Az)}. \tag{6}$$

On each iteration, ΔAz is added to Az until ΔAz goes below some pre-determined threshold, ϵ.

These calculations are specified in Algorithm 1.

Algorithm 1. Inverse Snyder Calculation

```
Require: $face$, $Az'$
  $lat \leftarrow 0$
  $lon \leftarrow 0$
  // Convert $Az'$ to symmetric subregion (not shown)
  // Determine initial estimate for Az
  $Az \leftarrow Az'$
  $A_G \leftarrow \frac{R'^2 \tan^2 g}{\cot Az' + \cot \Theta}$
  $\delta \leftarrow 1$
  // Iterate using Newton-Raphson
  while $\delta > \epsilon, \epsilon \sim 0$ do
    $F(Az) \leftarrow \frac{180^o A_G}{\pi R^2} - G - H - Az + 180^o$
    $F'(Az) \leftarrow \frac{\cos Az \sin G \cos g + \sin Az \cos G}{\sin H} - 1$
    $\delta \leftarrow \frac{F(Az)}{F'(Az)}$
    $Az \leftarrow Az + \delta$
  end while
  // Unwrap Az, so it falls in the correct symmetric region of the face (not shown)
```

Due to the non-linear equation, the inverse projection requires a number of iterations to converge on a value within a required accuracy. Furthermore, computations within the iteration process are often repeated and therefore redundant within a formal implementation. These repetitions have been identified and removed by the previous body of optimization work.

4 Optimizations

With the lack of closed form, and resulting indeterminate iterations required for root-finding, numerous optimizations were applied by Harrison et al. [7], to speed up the process. In a graphical application where inversion calls occur millions of times within a single screen of information, slow implementations impede real-time requirements. While an order of magnitude in reduction is preferred, even a constant reduction is beneficial.

These optimizations continue to be applied to an icosahedron. Application to other sphere circumscribing regular polyhedra is possible, but beyond the focus of the work.

Three types of optimizations were performed: operation reduction, curve fitting and, lastly, iteration removal. The operation reductions employed common computer science approaches, involving the identification and temporary storage of repeated calculations. Furthermore, the simultaneous calculation of sin and cos for angles which undergo both processes is improved through *sincos* directives. Such directives are capable of performing both computations in the time it takes to evaluate one separately. The improvements were classified as trivial, and were not, and continue not to be evaluated for improvements in optimization.

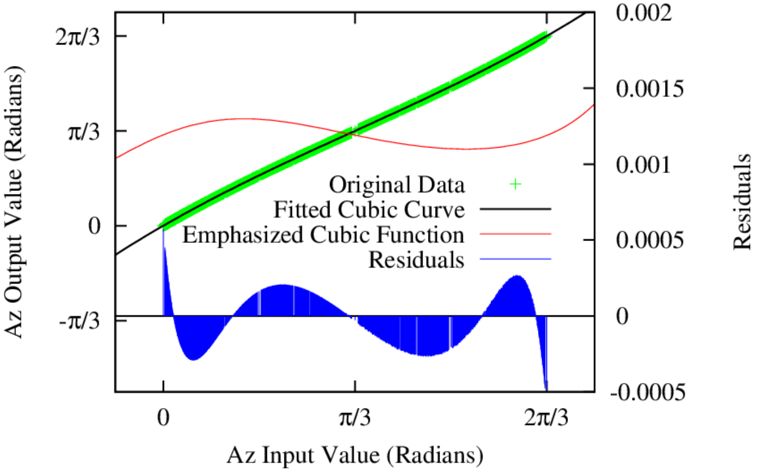


Fig. 4. Azimuth Before Against Azimuth After Iterative Newton Raphson

The more insightful optimization involved the application of a curve fitting approach. In this way, the Netwon Raphson iterations were initiated with a more reliable estimate. It was observed that the formulae applied to Newton Raphson were entirely dependant upon the input azimuth. When six thousand evenly distributed values within the possible range of the azimuth (up to 60^o on an equilateral icosahedron face) were plotted against Newton Raphson's resulting azimuthal value, a smooth curve was constructed (Figure 4). To emphasize its non-linear form, the exponential of its difference is also plotted: $y = e^{5(f(x)-x)}$.

Consequently the data is well suited for applying a polynomial approximation to for curve fitting. While higher degree polynomials were evaluated for the resulting residuals (against the data, Table 1), only a cubic polynomial was considered for analysis. The assumption was that as the degree of the polynomial increased, the operational requirements would offset the benefits of the improved initial estimate. Within this work, the higher degree polynomials are more thoroughly evaluated, and constructed using Horner's rule [1]. It should be noted that when finding higher degree polynomials, coefficients must be non-zero. For many of the even degree polynomials, this resulted in a close to zero coefficient for the dominant term. The operation count, which is reduced through Horner's rule, is included within Table 1.

This improved estimate, through the use of a cubic approximating polynomial, resulted in a 25% reduction in iterations, and a 15% reduction in computational time. These improvements are possible while still ensuring the solution converges to an accurate value.

Upon acquisition of such this improved estimate for the Newton Raphson iterative approach, an evaluation of eliminating iterations entirely was explored. To this end, the result of the polynomial function was directly employed, and its results evaluated. As expected, due to the lack of precision within the framework,

Table 1. Polynomial Approximating Azimuthal Shift

Polynomial	Sum of Squares of Residuals	Variance of Residuals	Operation Count
Degree 1	1.19e+00	2.02e-04	2
Degree 2	9.27e-01	1.66e-04	4
Degree 3	2.30e-04	3.92e-08	6
Degree 4	2.06e-04	3.51e-08	8
Degree 5	2.51e-05	4.28e-09	10
Degree 6	2.20e-05	3.75e-09	12
Degree 7	9.06e-07	1.55e-10	14
Degree 8	7.90e-07	1.34e-10	16
Degree 9	4.79e-08	8.18e-12	18
Degree 10	4.61e-08	7.87e-12	20

positional and areal error was generated, though nominally. The elimination of iterations resulted in a 45% reduction in computational time, at the expense of $0.7km^2$ areal error, and a displacement of 5.9 m. Consequently, for a visualization, or analysis requirement, wherein such errors are negligible for visibility or tolerance, the eliminated approach offers a viable alternative.

5 Supplemental Analysis

Lacking within the analysis of Harrison et al. is the briefly discussed polynomial approximations. While it is assumed that a cubic polynomial is sufficient, evidence is not provided to attest to this situation. As the polynomial improves the estimation of the curve fitted data, it increases the possibility for a reduction in iterations. Furthermore, the additional operational expense, for this improved estimation enables a reduction in errors when iterations are eliminated and the polynomial directly applied.

As such, different polynomial degrees are tested, and evaluated for their ability to support improved estimations. Further evaluation of the resulting real-time support is also explored.

6 Results

Implementation and testing occurred using Qt/C++ on an Intel i7 quad core processor under Ubuntu 10.05.

The original implementation was contrasted against approaches described using a fitted polynomial. Polynomials of degrees one through ten are evaluated. It was observed that a polynomial at degree ten no longer exhibits a time improvement over the original, and as such, higher degrees are not considered. As with Harrison et al., operation reduction is not considered for comparison. Instead, the reduction of iterations through an improved initial estimate, and the elimination of iterations by full use of the polynomial approximation are evaluated. An error analysis for the latter is also provided.

For each of the three approaches - the original, improved polynomial approximation, and iteration elimination - profiling, using *gprof* (v2.17), was performed 100 times for a high resolution, or quality level. A resolution refers to the number of times an icosahedron face is initially divided prior to vertex projection. For example, the 100 × 100 resolution level employed for the tests herein will split the face into 10,000 subtriangles. The vertices of these faces are then projected through the respective inversions.

Numerical results are presented in Table 2. The "Method" column indicates the version explored - the original inverse projection, the improved polynomial approximation, or the eliminated iterations variant. The average iteration reflects the improved estimate reducing the iterative convergence. This is also visualized in Figure 5, where the distribution of iterations across the surface of the sphere coloured accordingly. As the approximation improves, it requires less iterations to converge. The percent time improvement compares the improved and eliminated approaches against the original. Iteration improvement over the original is presented for the improved but not the eliminated approach as no iterations occur.

Error analysis for the elimination approach can be found in Table 3. Here, distance is an absolute value, and the average distance converted to metres. Similarly, the average and maximal error - as a percentage - are presented. The average area error is also converted into m^2, based on the surface area of the Earth being 510,072,000 km^2. The distribution of area error is visualized in Figure 6. Blue indicates an increase in area, whereas purple indicates a decrease.

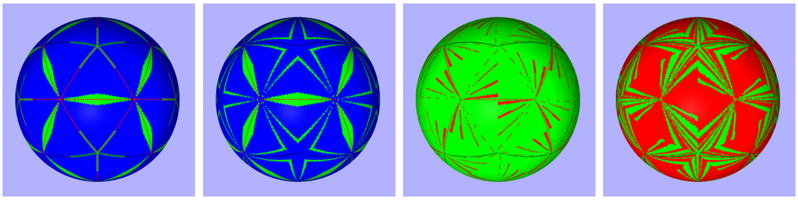

(a) Original (b) Poly. Degree 1 (c) Poly. Degree 5 (d) Poly. Degree 8

Fig. 5. Iteration Distribution. Blue = 4, Green = 3, Red = 2

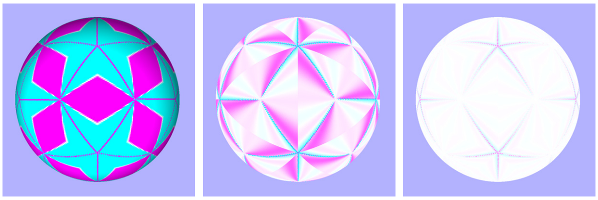
Analyzing the real-time support for the optimizations involves exploring how many points may be projected within the aforementioned 0.04167 seconds. If we assume that the only time consumption during the duration is the projection of points, we observe - based on the average time per call that approximately 240,000 points may be mapped through the original inverse Snyder projection. If we employ the improved iteration technique, this increases to 270,000. Due to the large reduction in time for the eliminated approach, we observe approximately 440,000 points that may be projected in the same time frame. This corresponds to the approximately 44% decrease in computational time. Of course, visualization involves more than just the projection of points, however the values provided

Table 2. Profiling Results

Poly. Degree	Method	Avg Iter.	Avg Time (s)	Std Dev.	Time Improv.	Iter Improv.
	Original	3.82	0.1019	0.0288		
1	Improved	3.82	0.1003	0.0322	1.57%	0.00%
1	Eliminated	0	0.0531	0.0238	47.89%	
2	Improved	2.75	0.0997	0.0302	0.99%	1.70%
2	Eliminated	0	0.0562	0.0226	44.19%	
3	Improved	2.92	0.0939	0.0323	6.38%	23.56%
3	Eliminated	0	0.0550	0.0203	45.16%	
4	Improved	3.26	0.0963	0.0322	3.51%	14.51%
4	Eliminated	0	0.0579	0.0240	41.98%	
5	Improved	2.89	0.0915	0.0299	7.76%	24.19%
5	Eliminated	0	0.0553	0.0248	44.25%	
6	Improved	2.85	0.0944	0.0334	5.22%	25.43%
6	Eliminated	0	0.0577	0.0251	42.07%	
7	Improved	2.43	0.0854	0.0270	13.12%	36.27%
7	Eliminated	0	0.0579	0.0224	41.10%	
8	Improved	2.31	0.0855	0.0293	15.09%	39.57%
8	Eliminated	0	0.0576	0.0230	42.80%	
9	Improved	2.29	0.0830	0.0258	17.17%	40.13%
9	Eliminated	0	0.0558	0.0240	44.31%	
10	Improved	3.74	0.0999	0.0308	4.31%	1.85%
10	Eliminated	0	0.0580	0.0227	44.44%	

Table 3. Error Analysis of Elimination Approach

Poly. Degree	Avg Dist. Error	Max Dist. Error	Avg Dist. Error (m)	Avg Area Error (%)	Max Area Error (%)	Avg Area Error (m^2)
1	4.856e-03	2.193e-02	3.093e+02	-6.665e-04	9.754e-01	-1.700e+04
2	5.121e-03	3.451e-02	3.262e+02	-7.862e-04	9.690e-01	-2.005e+04
3	9.490e-05	6.193e-04	6.044e+00	1.758e-06	1.596e-02	4.485e+01
4	1.538e-03	1.223e-02	9.796e+01	-4.911e-05	9.876e-01	-1.253e+03
5	2.167e-04	1.640e-03	1.380e+01	2.428e-06	1.596e-01	6.193e+01
6	2.456e-05	2.831e-04	1.564e+00	2.237e-06	7.625e-03	5.705e+01
7	5.709e-06	6.889e-05	3.636e-01	5.883e-07	1.813e-03	1.500e+01
8	4.667e-06	6.842e-05	2.973e-01	7.744e-07	2.302e-03	1.975e+01
9	8.099e-06	7.995e-05	5.158e-01	1.366e-07	7.884e-03	3.485e-02
10	1.717e-01	1.820e+00	1.093e+04	-7.657e+00	7.670e-01	-1.953e+08



(a) Poly. Degree 1 (b) Poly. Degree 5 (c) Poly. Degree 8

Fig. 6. Eliminated Distortion - Area Change. Blue - growth, Purple - reduction.

represent a maximal potential given the optimizations. The different quantity of points that may be projected are visualized in the graph of Figure 7. The figure illustrates how the decrease in processing time for the iteration elimination approach increases the number of projected points. Further, the number of projected points for the improved approach gradually increases, over the original, as the degree, and consequently, the accuracy of the polynomial improves. This increase reaches a peak when employing a polynomial of degree 9, whereas the additional calculations for degree 10 begin to inhibit the improved approximation thereby reducing the points that may be projected in real-time.

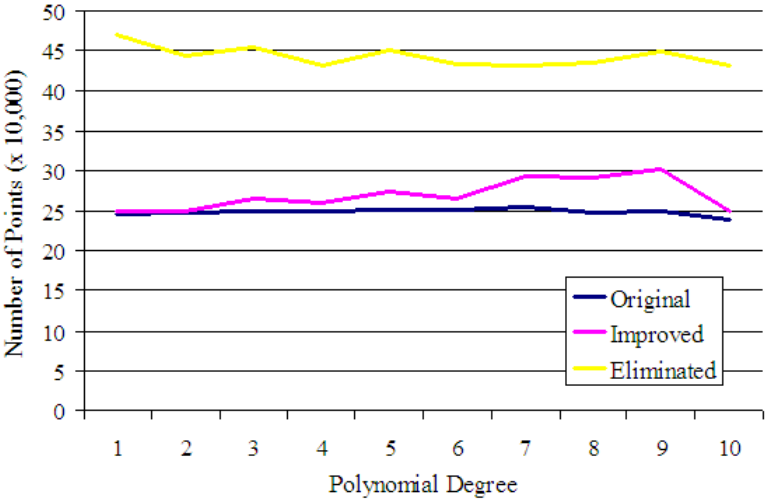


Fig. 7. Number of Points Projected in Real-Time (24 fps)

7 Discussion

Based on the error comparison between the original projection and the eliminated approach, as illustrated in Table 3, distance and areal errors are improved upon as the degree of the approximating polynomial increases. This corresponds to a better approximation for the fitted curve. For example, with a degree of 3, an average area error of 45 m^2 occurs. A polynomial of degree 9, on the other hand, has an average areal error corresponding to 3.5 m^2. This reduction in error increases the resolution upon which the system may employ the faster elimination approach before this error becomes apparent.

The achievement of almost a half a million points, which may be projected within the 24 fps requirement, is a great improvement over the quarter million previously possible. This is particularly important when coupled with a visualization system that exploits level of detail visualization techniques, and therefore have the ability to employ the eliminated approach with simultaneous precision at the high levels of detail.

8 Conclusion

With visualization systems requiring real-time accurate and equal area information of large quantities of data, an effective projection mechanism is imperative. With much data acquired through planar means, area preservation during spherical conversion is important. Such an area preserving quality provides researchers and businesses with accurate analysis of their data. Such analysis, along with the visualization, relies on underlying feature outlines which may span thousands or even millions of points for a single region. The conversion of these feature points to their spherical coordinates requires both accuracy and efficiency. The repeated employment of the inverse Snyder projection, as occurs in industry, can be extremely time consuming - greatly limiting the number of points available for display within a real-time framework. The speeds ups provided by Harrison et al greatly reduce time taken for the iterative root-finding for the non-linear calculations, however an improved approximating polynomial is able to further improve these results.

Whereas Harrison et al found a 25% iteration reduction with a cubic polynomial, upwards of 40% is possible with a degree 9. Furthermore, while the higher degree doesn't significantly improve the speed of the eliminated iteration approach - roughly 45% - it reduces the error ten fold. This improvement increases the amount of data that may be visualized until this erroneous threshold is reached.

Finally, the real-time qualities of these optimizations are obvious and demonstrate the need for their employment. Whereas the original Snyder inversion only supported a quarter million points for projection, the optimized elimination approach supports almost one half million points. This increase in visualization data lends itself to improved visualization, processing and analysis time.

If we consider the Earth as a perfect sphere, it remains of worth to expand on this work further by analyzing the different sphere circumscribing polyhedron. Snyder discovered his projection resulted in the least amount of distortion when projecting with the truncated icosahedron. While the hexagons constructed are challenging to incorporate into a computer graphics visualization, it may be of worth to attempt to work around them to support a representation with reduced shape distortion. In the meanwhile, icosahedral support is employed within industrial applications, and we have demonstrated further optimizations which will benefit the community.

Acknowledgment. This work has been partially supported by PYXIS Innovation and NSERC. We would like to thank Idan Shatz for his thoughtful discussions.

References

1. Borwein, P., Erdélyi, T.: Polynomials and Polynomial Inequalities, pp. 29–41. Springer, New York (1995)
2. Canters, F.: Small-Scale Map Projection Design. Taylor & Francis, NY (2002)
3. Commission for Environmental Cooperation: North America (2004), http://atlas.nrcan.gc.ca/site/english/maps/archives/various/north_america_cec (online accessed May 01, 2011)
4. Google: Google Earth (2011), http://www.google.com/earth/index.html (online accessed November 20, 2011)
5. Google - Imagery: Google Maps (2010), http://maps.google.ca (online accessed December 11, 2010)
6. Gore, A.: The Digital Earth: Understanding our Planet in the 21st Century. California Science Center, Los Angeles, USA (January 31, 1998)
7. Harrison, E., Mahdavi-Amiri, A., Samavati, F.: Optimization of Snyder's Inverse Polyhedral Projection. In: Intl. Conference on CyberWorlds, pp. 1–8. IEEE (2011)
8. Kimerling, A.J., Sahr, K., White, D., Song, L.: Comparing Geometrical Properties of Global Grids. Cartography and Geographic Information Science 26(4), 271–288 (1999)
9. Lambert, J.H.: Anmerkungen und Zusätze zur Entwerfung der Land- und Himmelscharten. Mit 21 Textfiguren (1772)
10. van Leeuwen, D., Strebe, D.: A Slice-and-Dice Approach to Area Equivalence in Polyhedral Map Projections. Cartography and Geographic Information Science 33(4), 269–286 (2006)
11. NASA: Blue Marble (2000), http://visibleearth.nasa.gov/ (online accessed December 11, 2010)
12. PYXIS Innovation Inc: How PYXIS Works (2011), http://www.pyxisinnovation.com/pyxwiki (online accessed May 01, 2011)
13. Snyder, J.P.: Map Projections - A Working Manual: U.S. Geological Survey Professional Paper 1396. U.S. Government Printing Office, Washington, DC (1987)
14. Snyder, J.P.: An Equal-Area Map Projection for Polyhedral Globes. Cartographica 29(1), 10–21 (1992)
15. Snyder, J.P.: Flattening the Earth: Two Thousand Years of Map Projections. University of Chicago Press, Chicago (1997)
16. United States Department of the Interior: Map Projections: From Spherical Earth to Flat Map (2011), http://www.nationalatlas.gov/articles/mapping/a_projections.html (online accessed May 01, 2011)

Error-Controllable Simplification of Point Cloud

Yichen Li[1], Mingqiang Wei[2], Jianhuang Wu[3], and Mingyong Pang[1]

[1] Nanjing Normal University, Nanjing, P.R. China
[2] The Chinese University of Hong Kong, Hong Kong
[3] Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences
yichen.lee.njnu@gmail.com, mqwei@cse.cuhk.edu.hk,
jh.wu@siat.ac.cn, panion@netease.com

Abstract. Point cloud simplification has become a vital step in any point-based surface processing pipeline. This paper describes a fast and effective algorithm for point cloud simplification with feature preservation. First, feature points are extracted by thresholding curvatures; Second, for non-feature points, they are covered by distinct balls, the points in each ball are substituted by an optimized point. Thus, the simplified point cloud consists of extracted feature points and optimized points. This algorithm is able to produce coarse-to-fine models by controlling a general error level. But the error level of each ball may be adaptively adjusted according to the local curvature and density that can avoid holes generation during the simplification process. Finally, the simplified points are triangulated by Cocone algorithm for surface reconstruction. This algorithm has been applied to a set of large scanned models. Experimental results demonstrate that it can generate high-quality surface approximation with feature preservation.

Keywords: Point Cloud Simplification, Surface Reconstruction, Quadric Error Metrics.

1 Introduction

Due to recent advances in point cloud acquisition techniques, 3D object boundary surfaces are now commonly acquired with sub-millimeter accuracy. The initial output of acquisition devices such as laser range scanners therefore generally consists of point clouds of considerable redundancy. Unfortunately, the huge numbers of point clouds will bring us a great deal of trouble in the downstream processing and the data storage, transmission and rendering. By simplifying the point set first, the surface reconstruction from simplified point cloud is accelerated significantly and the mesh simplification step is avoided altogether. Furthermore, with the increasing availability of powerful point-based modelling[1] and visualisation[2] techniques, the simplification of dense point clouds for subsequent point-based rather than polygonal mesh-based processing plays a rather important role by itself. In either case, point cloud simplification represents a vital step in point-based surface processing pipeline.

Obviously the efficiency of point cloud simplification algorithm is essential for the large scale input data. Meanwhile, to guarantee the quality of simplification result for following geometric processing, we must take geometric features into account during simplification process. Generally, clustering method may achieve high efficiency while

M.L. Gavrilova and C.J.K. Tan (Eds.): Trans. on Comput. Sci. XVI, LNCS 7380, pp. 149–162, 2012.

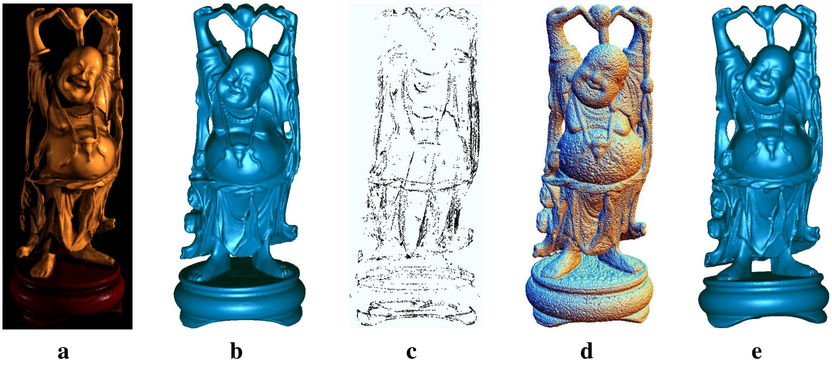


Fig. 1a-e. Simplification and reconstruction of a Buddha model(543652 points). From a raw scan with significant point-sampled data, our algorithm extracts the feature points, and calculates the optimized points for the non-feature points. Those two kinds of points are then blended to produce a simplified version for surface reconstruction. **a** Scanned Model, **b** Input Scan, **c** Feature Points, **d** Coverage Balls, **e** Reconstruction

losing a lot original features, contrarily, particle simulation method could obtain satisfied result while losing some efficiency[12]. So we have to hold the tradeoff carefully between quality and efficiency of simplification.

Our algorithm is an improved version of the algorithm proposed by Othake [20]. In our implementation, we sufficiently consider the feature preservation and can deal with the non-uniform point cloud. We first extract the feature points by thresholding curvatures, and then simplify the non-feature points by solving a collection of error functions. The feature points and optimized points are assembled to reconstruct mesh surface using Cocone [21] algorithm. Fig. 1a-e illustrates the pipeline of our approach. Experimental results demonstrate that our method could reserve enough original feature without loss of efficiency. In the rest of this paper, it is organized as follows. Section 2 gives a brief summary of related work and section 3 and 4 detail the preprocessing and the algorithm. Experimental results and analysis are given in Section 5, followed by concluding remarks in Section 6.

2 Related Work

In recent years, many researchers have focused on how to simplify redundant point clouds efficiently and effectively. The re-sampling method calculates a set of new samples from the original point cloud based on certain rules. Dey et al. [3] present a point cloud simplification approach, and adopt local curvature to detect the redundancy in the input point cloud and to ensure relevant point densities; This is accomplished by exploiting a 3D Voronoi diagram. Alexa et al. [4] uniformly reduce point cloud redundancy by estimating a point's contribution to the moving least squares (MLS) representation of the underlying surface. Those points contributing the least are subsequently removed. This method does not guarantee the absence of insufficiently dense output point sets. Scheidegger et al. [5] extended the MLS approach to simplify point set surfaces and constructing a high-quality triangulation. Kalaiah and Varshney [6] get rid of redundancy through measuring the redundancy of each individual point based on the local geometric properties derived from the surrounding points. Lee et al. [7]presented a simplification method that reduces the number of points using part geometry information. In their work, the points are removed based on their normal vector values using 3D grids. Miao et al. [8] proposed a curvature-aware adaptive sampling method. An adaptive mean-shift clustering scheme is designed to generate non-uniformly distributed sampling points. Bossonnat et al. [9] describe a coarse-to-fine point simplification algorithm that randomly calculates a point subset and constructs a 3D Delaunay triangulation. Moenning et al. [10] propose an intrinsic coarse-to-fine point simplification algorithm that guarantees uniform or feature-sensitive distribution. However, their method requires many computations and a large memory. Wu et al. [11] present a new sub-sampling technique for dense point clouds which is specially adjusted to the particular geometric properties of circular or elliptical surface splats. A global optimization scheme computes an approximately minimal set of splats that covers the entire surface while staying below a globally prescribed maximum error tolerance.

Generally, re-sampling methods can produce high quality of output models while the computation complexity is high. Clustering methods split the point cloud into a number of sub-sets, each of which is replaced by one representative sample. Pauly [12] proposes two types of clustering simplification algorithms. One type of the algorithms are based on incremental region-growing starts from a random seed point and a cluster is built by successively adding nearest neighbors. Such incremental region-growing is terminated when the size of the cluster reaches a maximum bound. The other type of the algorithms are hierarchical clusterings which compute the set of clusters recursively splits the point cloud using a binary space partition. Yu et al. [13] present an Adaptive Simplification Method (ASM) which is an efficient technique for simplifying point-based complex 3D models based upon hierarchical cluster tree structure. Shi et al. [14] also present an ASM by employing the k-means clustering algorithm to gather similar points together in the spatial domain and uses the maximum normal vector deviation as a measure of cluster scatter to partition the gathered point sets into a series of sub-clusters in the feature field. However, the proposed method may generate uniformly distributed sparse sampling points in the flat areas and necessary higher density in the high curvature regions.

The clustering methods usually are simple and fast, however, the simplified models are not satisfied without optimization processing. Another important simplification strategy for point-based surfaces is iterative reducing the number of points using an atomic decimation operator. Decimation operations are usually arranged in a priority queue according to an error metric that quantifies the error caused by the decimation [12]. This kind of method is similar to mesh-based simplification methods mentioned in [16]. Quadric error metrics is applied in [16] to measure the error caused by the mesh contraction. Iterative method can obtain both satisfied quality and speed except a larger memory consumption. In [17], a geometry-images-based simplification algorithm for point-sampled surfaces is proposed and the point set surfaces are simplified according to the curvature and simplified density fastly. Lee et al. [18] adopt discrete shape operator to find the weight of the features of the 3D model and extract the relevant points for a dense input point set. Jong et al. [19] present a novel rapid and effective point simplification algorithm using local coplanar analysis on the basis of an octree data structure. By using the octree data structure, it proposes some hierarchical simplifications and renderings for the base model to suit user demand.

3 Preprocessing

Assuming a piecewise smooth surface Φ that is approximated by a set of sampled points $P = \{p_1, \cdots, p_N\}$, our goal is to fast and effectively simplify it and create a high-quality surface approximation. Typically, raw point cloud data are obtained by encoding a group of overlapped range images and the local density is higher at the regions that correspond to the overlapped regions of the range images. Thus, appropriate weights are assigned to the points for compensating density irregularities. We assign to each point $p_i \in P$ a weight a_i defined by

$$a_i = \frac{1}{K} \sum_{j=1}^{K} \|p_i - p_j\|^2 \tag{1}$$

where $\{p_j\}_{j=1}^{K} \in P$ are the K nearest neighbors of p_i. This weight scheme is sufficient to compensate the density irregularities.

To extract the feature points, calculate the optimized points and compute the 2D convex hull, we need to estimate the unit normals $\aleph = \{n_1, \cdots, n_N\}$ at the points of P. Generally, unit normals can be directly acquired via photometric stereo [22][23]. If the point cloud data provide no normal information, we employ a method relied on covariance analysis [24] to estimate them. The 3×3 covariance matrix C for a sample p_i is given by

$$C = \begin{bmatrix} p_{i,1} - \overline{p} \\ \cdots \\ p_{i,K} - \overline{p} \end{bmatrix}^T \cdot \begin{bmatrix} p_{i,1} - \overline{p} \\ \cdots \\ p_{i,K} - \overline{p} \end{bmatrix}, \overline{p} = \sum_{j=1}^{K} p_{i,j} \tag{2}$$

where $\{p_{i,1}, \cdots, p_{i,K}\}$ is the K nearest neighbors of the point p_i in P. Since C is symmetric and positive semi-definite, all eigenvalues are real-values and all eigenvectors form an orthogonal frame [25]. The eigenvector corresponding to the smallest eigenvalue is taken as the normal vector of the point p_i.

4 Point Simplification Algorithm

To simplify the point cloud, we first extract the feature points $P^f = \{p_1, \cdots, p_m\}(0 \leq m \leq N)$, by thresholding point curvature, and then generate a collection of balls centered at $\{c_1, \cdots, c_n\}$ with adaptive radius $\{r_1, \cdots, r_n\}$ covering the non-feature points. The ball generation used in this paper is an improved version of the approach proposed by Ohtake [20] for creating an approximately minimal set of spheres to cover the whole point-sampled surface. The algorithm is described as follows:

1. Extract the feature points according to curvature; The feature points are protected and do not participate the following processing.
2. Set all non-feature points as uncovered.
3. Select point c from the set of uncovered points without regard to the order of the selection and label this point as covered.
4. For the selected point c, evaluate r, the radius of ball centered at c, and an optimized point v.
5. Calculate local curvature cur and density $dens$ at point c. According to the local properties, adjust the error threshold, and repeat step 4.
6. Project the non-feature set $\{\|p - c\| < r\}$ onto the plane tangent to c. Compute the 2D convex hull of the projections of P_c^r. Label the points of P_c^r which are projected strictly inside the convex hull as covered.
7. Terminate the process if there are no more uncovered points. Otherwise return to step 3.

4.1 Feature Points Extraction

Feature points describe the basic shape of the object and have a significant influence on the quality of surface reconstruction. Thus, in order to avoid geometric information lost, before simplifying the point cloud, we extract the feature points by thresholding curvature. In further simplification process, those feature points will not be dealt with.

There are many kinds of feature points, we mainly focus on silhouettes and corners, and extract them, because they are important visual cues for shape perception [26] and are very effective at conveying shapes [27][28]. More precisely, silhouettes are those points positioned at convex or concave boundaries, and corners are located at sharp regions. If point p is a feature point, the curvature at this point is higher, i.e., the sum of the distances from the k nearest neighbors of p to the tangent plane at p is higher. Otherwise if point p is a non-feature point, the curvature is lower, i.e., the sum of the distances from the k nearest neighbors of p to the tangent plane at p is lower. As seen in Fig. 2, hollow points refer to the k nearest neighbors of point p, the distance sum is higher which means the curvature at point p is higher, thus point p in the left image is a feature point. In contrast, point p in the right image is a non-feature point.

Based on the analysis described above, we construct the measurement function as follows:

$$measure(p) = \frac{1}{K} \sum_{j=1}^{K} |(p - p_j) \cdot n_p| \qquad (3)$$

where $\{p_j\}$ are the k nearest neighbors of point p, n_p is the unit normal at p, $(p - p_j) \cdot n_p$ refers to the distance from the j^{th} nearest neighbor to the tangent plane at p. If $measure(p)$ is higher than a threshold δ, this means point p is considered as a feature point; Otherwise it is a non-feature point. To distinguish feature points and non-feature points, we set threshold δ as follows:

$$\delta = \frac{\alpha}{N} \sum_{i=1}^{N} measure(p_i) \qquad (4)$$

where N is the number of points, α is the adjustment factor. If $measure(p)$ is higher than δ, point p is set as a feature point. Otherwise it is a non-feature point. Fig. 3 and Fig. 4 demonstrate the extracted quality of setting different α for the same models and same α for different models.

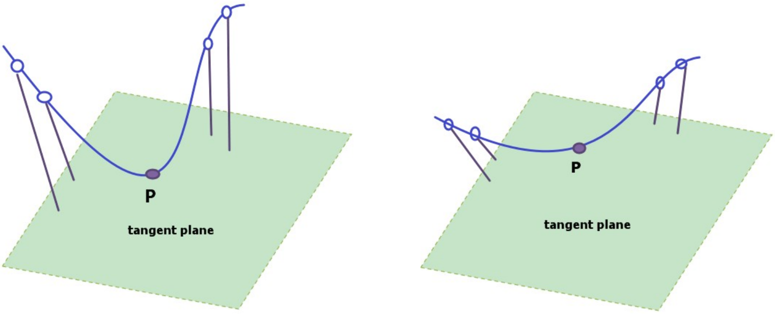


Fig. 2. Measuring feature and non-feature points by curvature

4.2 Point Cloud Density

The extraction methods of point cloud characteristics can be roughly classified into two categories: based on the point-to-point distance method and based on the clustering method. We adopt the method based on point-to-point distance to analyze the density distribution of the point cloud. For non-feature point set $P' = \{p'_1, \cdots, p'_M\}(0 < M < N)$, we denote the distance between point p'_i and p'_j by $dis(p'_i, p'_j)$, local density at point p'_i can be defined by

$$dens_i = \min(dis(p'_i, p'_j)), (1 \leq j \leq M, i \neq j) \tag{5}$$

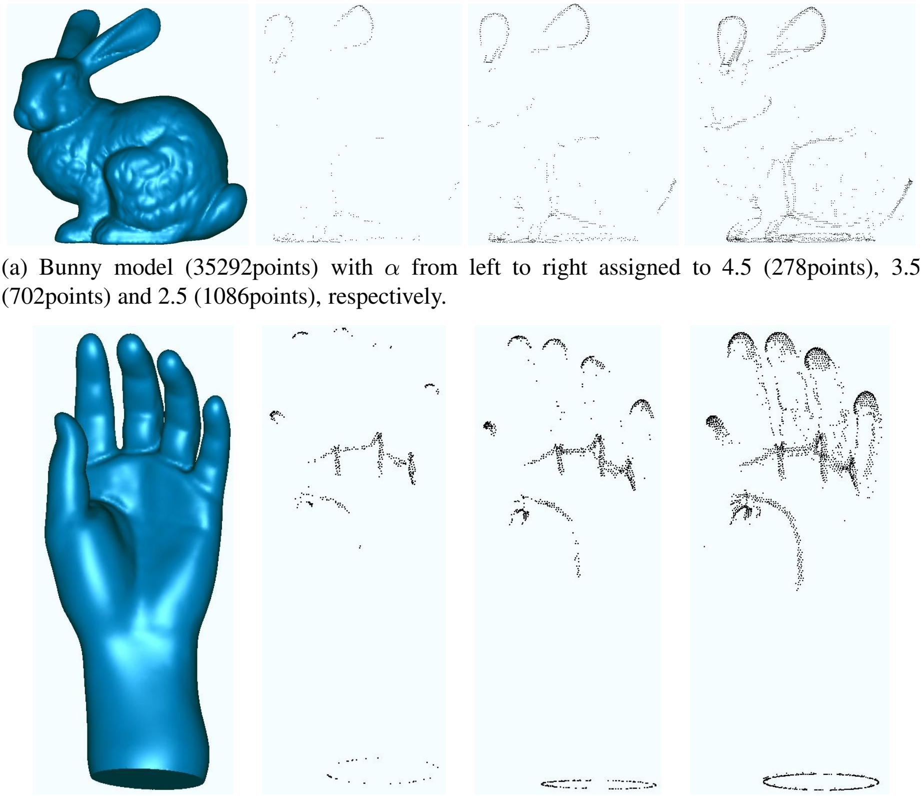
The average density of non-feature point set P' is given by

$$\overline{dens_i} = \frac{1}{M}\sum_{i=1}^{M} dens_i \tag{6}$$

where M is the number of non-feature points. Thus, when we decrease $dens_i$, the algorithm selects more points locally. From the magnified fragment of the left image of Fig. 5, the density is changing in this region, thus, in the process of simplification, we will sufficiently consider such region.

4.3 Curvature Estimation

In order to preserve more details, when simplifying the point cloud, we will sufficiently consider the curvatures of the point cloud. There are many approaches to estimate the curvatures on discrete points, such as Paraboloid Fitting approach, Circular Fitting approach, Guass-Bonnet approach and so on. The paraboloid fitting approach is used



(a) Bunny model (35292points) with α from left to right assigned to 4.5 (278points), 3.5 (702points) and 2.5 (1086points), respectively.

(b) Hand model (39325points) with α from left to right assigned to 4.5 (372points), 3.5 (880points) and 2.5 (2228points), respectively.

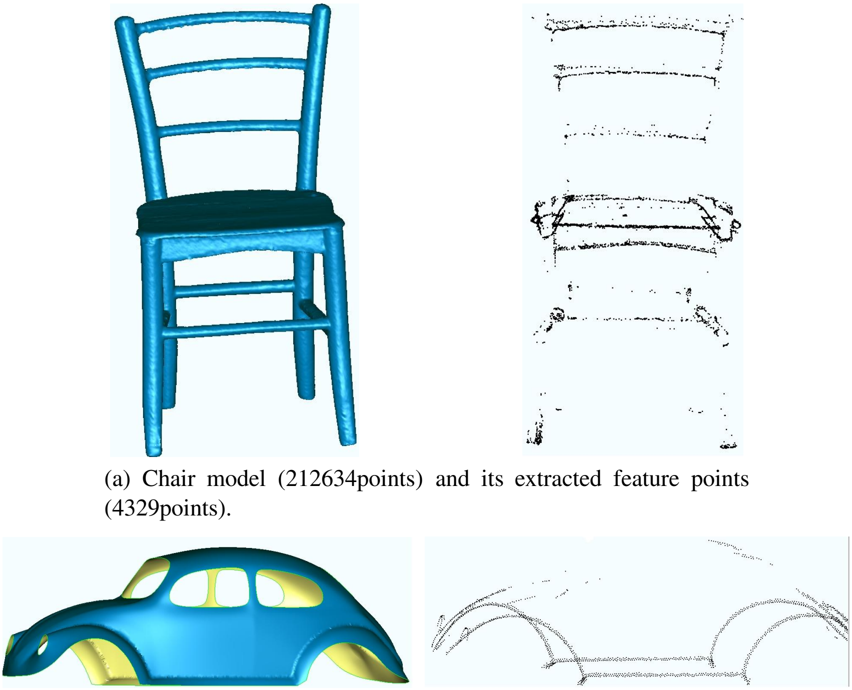
Fig. 3. Different parameter values for the same models

in this paper to estimate the mean curvature, because it is more robust when using paraboloid in the local neighborhood of a point to estimate the curvature, and can result in most optimized results. Assuming paraboloid equation is

$$z = ax^2 + bxy + cy^2 \tag{7}$$

For each non-feature point p, we fit a paraboloid by least square method using point p and its k nearest neighbors. The coefficients a, b and c can be computed by solving a linear equation, i.e., $Ax = b$, where matrix

$$A = \begin{bmatrix} x_1^2 & x_1 y_1 & y_1^2 \\ \vdots & \vdots & \vdots \\ x_{K+1}^2 & x_{K+1} y_{K+1} & y_{K+1}^2 \end{bmatrix}, \tag{8}$$



(a) Chair model (212634points) and its extracted feature points (4329points).

(b) Car model (30024) and its extracted feature points (1855).

Fig. 4. Same parameter value ($\alpha = 3.0$) for different models

$$x = \{a, b, c\}^T, \tag{9}$$

$$b = \{z_1, z_2, \cdots, z_{K+1}\}^T. \tag{10}$$

This overdetermined equation can be solved by householder transformation to obtain coefficients. Thus, the mean curvature at p is

$$H_i = a + c, \tag{11}$$

and the mean curvature of the non-feature point set is

$$\overline{H} = (\sum_{i=1}^{M} H_i)/M. \tag{12}$$

The right image of Fig. 5 shows the mean curvature visualization.

4.4 Ball Radius and Optimized Point

The core of the algorithm is computing a set of coverage balls with adaptive radius $\{r_1, \cdots, r_m\}$ and optimized points $\{v_1, \cdots, v_m\}$ in each ball. The optimized points

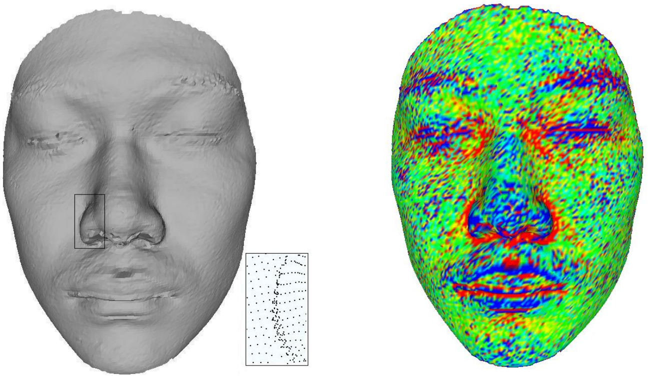


Fig. 5. Face model with varying local density (magnified fragment) and mean curvature visualization using different colors

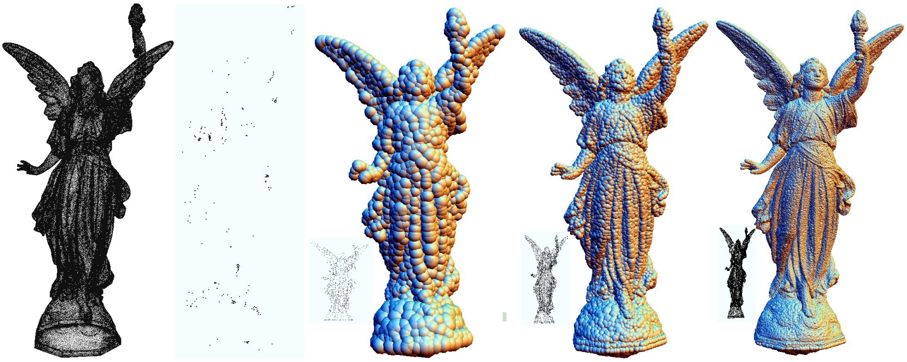


Fig. 6. Lucy model (262909 points) with different initial T_{err}. From left to right, original model, extracted feature points (264 points with $\alpha = 3.5$), coverage balls with $T_{err} = 10^{-4}(4646 points), 10^{-5}(21347 points), 10^{-6}(91019 points)$, respectively.

will substitute the non-feature points in each ball for simplification. The approach is based on local quadric error minimization strategy, and is an improved version of the approach described by Ohtake[20] which set a fixed error threshold. In contrast, our approach will adjust the error threshold according to local curvature and density.

For each selected ball centered at c, we calculate the adaptive radius r and an optimized point v in it. According to[20][16][11], we define a quadric error function for each point c as follows:

$$Q(c, r, x) = \sum_j w_j G_R(\|p_j - c\|)(n_j \cdot (x - p_j))^2 \tag{13}$$

where weights $\{w_j\}$ are defined by (1), p_j is a non-feature point within a bounding sphere centered at c with radius R, n_j is the unit normal at point p_j, x is the potential

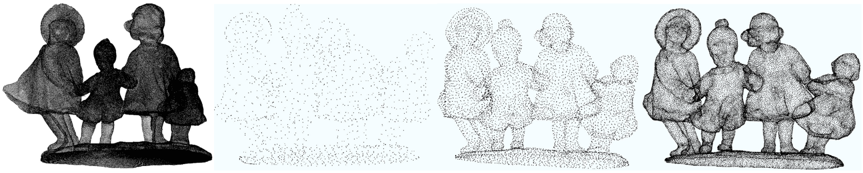


Fig. 7. Children model (724742 points) and simplified model with different error values T_{err}. From left to right, original model, models with $T_{err} = 10^{-4}$, 10^{-5} and 10^{-6}, respectively.

optimized point (we need further judge whether x is an optimized point), $n_j(x - p_j)$ denotes the distance from point x to the tangent plane at point p_j, r is the radius of coveraged ball, $G_R(\| \cdots \|)$ is a Gaussian-like function as defined by

$$G\sigma(\rho) = \begin{cases} exp(-8(\rho - \sigma)^2) & if|\rho| \leq \sigma/2, \\ 16/e^2(1 - |\rho|/\sigma)^4 & if\sigma/2 < |\rho| \leq \sigma, \\ 0 & if\sigma < |\rho|. \end{cases}$$

In practice, we set $R = 2r$. Function $Q(c, r, x)$ computes a weighted sum of the squared distances from point x to the tangent planes at $\{p_j\}$ within spherical region $\|p_j - c\| \leq R$. If r in (13) is fixed, point $x_{min} = x_{min}(r)$, the minimizer of $Q(c, r, x)$ is easily found by solving a system of linear equations. Thus we set error function

$$E(r) = \frac{1}{L}\sqrt{Q(c, r, x_{min})} \tag{14}$$

where L is the length of a main diagonal of the bounding box of the point cloud. As proved in [29], the error function $E(r)$ weighs the curved degree of the reconstructed surface inside sphere $\|x - c\| < R$. We get r by solving the following equation

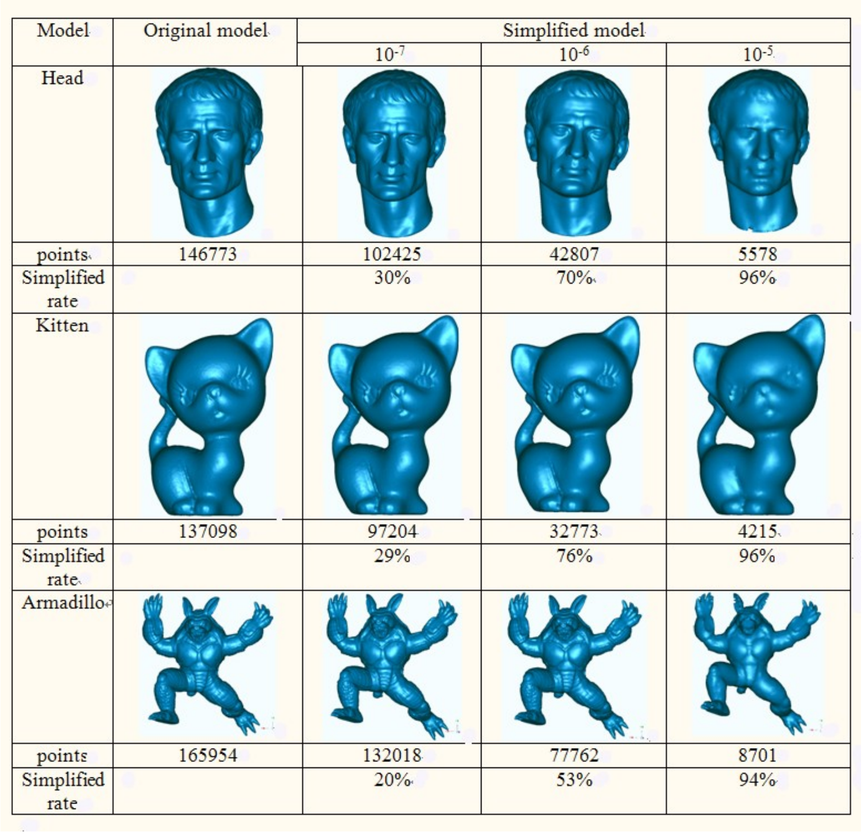
$$E(r) = T_{err} \tag{15}$$

where T_{err} is a user-controlled accuracy. $E(r)$ is monotonically decreasing as $r \to 0$, thus, we can use bisection method to solve (14).

Once r is fixed, we check whether x_{min} lies inside or outside of region $\|x - c\| < r$. If x_{min} is within this region, we use it as the optimized point v associated with the ball centered at c. Otherwise, we set $v = c$, because the surface curvature in this region is large and c there must be a sharp feature.

As seen, for each center c, the above-mentioned error threshold T_{err} is unaltered. Actually, adjusting T_{err} for different center c will be more desirable according to the local curvature and density. Thus, we compute the local curvature and density at point c for regulating the accuracy Terr as follows:

1. If mean curvature H_i at c_i is smaller than $\overline{H}$ and $dens_i$ is larger than $\overline{dens}$, this fact means that this region is relatively flatter, we can increasingly augment T_{err} for extending r to simplify more points.
2. If mean curvature H_i at c_i is larger than $\overline{H}$ and $dens_i$ is smaller than $\overline{dens}$, which indicates that this region contains more features, we decrease T_{err} for shrinking r to save more points.



Model	Original model	Simplified model		
		10^{-7}	10^{-6}	10^{-5}
Head				
points	146773	102425	42807	5578
Simplified rate		30%	70%	96%
Kitten				
points	137098	97204	32773	4215
Simplified rate		29%	76%	96%
Armadillo				
points	165954	132018	77762	8701
Simplified rate		20%	53%	94%

Fig. 8. The reconstruction results produced by Cocone algorithm using the simplified points compared with the original models are demonstrated. As seen, controlling different T_{err} can generate different levels of detail.

As shown in Fig. 6, the error T_{err} controls the approximation accuracy, large T_{err} results in large radius of coverage balls and simplify more points. Otherwise, small T_{err} preserves more points. Fig. 7 demonstrates how different errors T_{err} control the final simplified results.

5 Results and Analysis

5.1 Parameter Settings

To use our method, one must set two parameters beforehand, one is adjustment factor α which determines the number of feature points. If α is larger, the threshold becomes larger simultaneously, the number of feature points will naturally decrease. Otherwise if α becomes smaller, there will be more points considered as feature points. Thus, in the implementation, user can select different factors for different models. The other

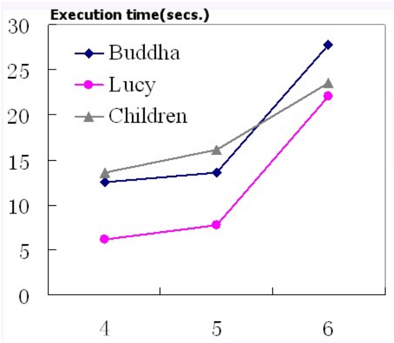
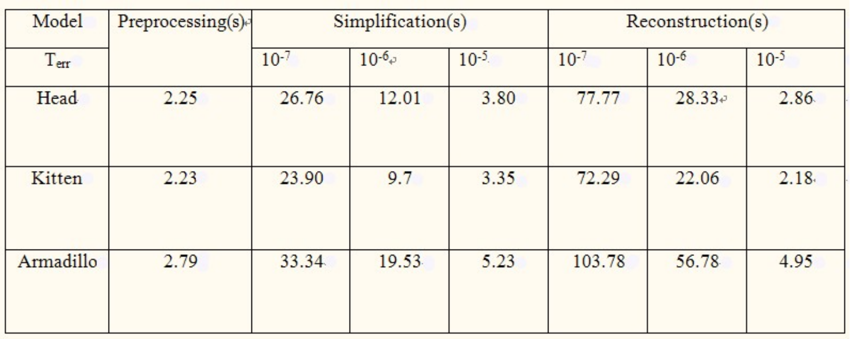


Fig. 9. The execution time of simplification with different error values T_{err} for different models. The digital 4, 5, 6 on the abscissa refer to $T_{err} = 10^{-4}, 10^{-5}$ and 10^{-6}, respectively.



Model	Preprocessing(s)	Simplification(s)			Reconstruction(s)		
T_{err}		10^{-7}	10^{-6}	10^{-5}	10^{-7}	10^{-6}	10^{-5}
Head	2.25	26.76	12.01	3.80	77.77	28.33	2.86
Kitten	2.23	23.90	9.7	3.35	72.29	22.06	2.18
Armadillo	2.79	33.34	19.53	5.23	103.78	56.78	4.95

Fig. 10. Time measurements of three main stages of the suface reconstruction process for the models considered in Fig. 8

parameter is initial error T_{err} which controls the approximation accuracy. It determines the tolerance of the optimized point and the size of radius of coveraged ball. See Fig. 8 for more details.

5.2 Timing and Computational Complexity

The execution time of experimental results is measured on a Pentium 4 (2.99GHz) PC with 2GB of main memory. In the statistical data shown in Fig. 9 and Fig. 10, our simplification is fast in obtaining the preferable results. Since we use $kd - tree$ to find nearest neighbors, the computational complexity is $O(Nlog^N)$, where N is the number of points.

5.3 Simplification Quality

The algorithm first extracts the feature points in the original model, and then generates adaptive balls covered the non-feature points. We use Cocone algorithm to produce meshes, which takes the simplification results as input. Our error-controllable simplification algorithm can produce levels-of-detail results with feature preservation, and the simplification results can be used to generate high-quality meshes using Cocone algorithm.

5.4 Limitations

The proposed algorithm is lack of strict theoretic proof and is basically an application issue. In addition, this algorithm may not be to handle the sparse-point-set case.

6 Conclusions

We present a fast and robust algorithm for point cloud simplification. The algorithm is an improved version of Ohtake's approach. Our algorithm can reconstruct surfaces from uniform and non-uniform point clouds. In the simplification process, we enable the local properties to adjust the error threshold. The simplification results can ensure the surface reconstruction results by using Cocone algorithm.

Acknowledgment. The authors would like to thank the anonymous reviewers for their constructive comments. The authors would like to thank Shengzhou Luo, Seng Wang both from Shenzhen Institutes of Advanced Technology for their valuable suggestions. The authors would like to thank Luming Liang from Colorado School of Mines for his wording modification for this paper. The models used in this paper are courtesy of Stanford 3D Scanning Repository and AIM@SHAPE Repository. This work was supported by the National Natural Science Foundation of China (Grant No. 60803108).

References

1. Linsen, L.: Point cloud representation. CS Technical Report 2001-3. Universität Karlsruhe, Germany (2001)
2. Alexa, M., Behr, J., Cohen-Or, D., et al.: Point Set Surfaces. In: Proc. 12th IEEE Visualization Conf., San Diego, USA, pp. 21–28 (2001)
3. Dey, T.K., Giesen, J., Hudson, J.: Decimating samples for mesh simplification. In: Proc. of 13th Canadian Conference on Computational Geometry, pp. 85–88 (2001)
4. Alexa, M., Behr, J., Cohen-Or, D., et al.: Computer and rendering point set surface. IEEE Transaction on Visaulization and Computer Graphics 9(1), 3–15 (2003)
5. Scheidegger, C., Fleishman, S., Silva, C.: Triangulating point set surfaces with bounded error. In: Eurographics Symposium on Geometry Processing, pp. 63–72 (2005)
6. Kalaiah, A., Varshney, A.: Modeling and rendering of points with local geometry. IEEE Trans. Vis. Comput Graphics 9(1), 30–42 (2003)
7. Lee, K., Woo, H., Suk, T.: Point data reduction using 3D grids. The International Journal of Advanced Manufacturing Technology 18(3), 201–210 (2001)
8. Miao, Y., Pajarola, R., Feng, J.: Curvature-aware adaptive re-sampling for point- sampled geometry. Computer-Aided Design 41(6), 395–403 (2009)
9. Boissonnat, J.D., Cazals, F.: Coarse-to-fine surface simplification with geometric guarantees. In: Proc. of EUROGRAPHICS, pp. 490–499 (2001)

10. Moenning, C., Dodgson, N.A.: Intrinsic point cloud simplification. In: Proc. of 14th GrahiCon (2004)
11. Wu, J., Kobbelt, L.P.: Optimized sub-sampling of point sets for surface splatting. In: Proc. of EUROGRAPHICS, pp. 643–652 (2000)
12. Pauly, M., Gross, M., Kobbelt, L.P.: Eifficient simplifiction of point-sampled surfaces. In: Proc. of IEEE Visualization, pp. 163–170 (2002)
13. Yu, Z., Wong, H., Peng, H., et al.: An adaptive simplification method for 3D point-based models. Computer-Aided Design 42(7), 598–612 (2010)
14. Shi, B., Liang, J., Liu, Q.: Adaptive simplification of point cloud using k-means clustering. Computer-Aided Design 43(8), 910–922 (2011)
15. Song, H., Feng, H.: A global clustering approach to point cloud simplification with a specified data reduction ratio. Computer-Aided Design 40(3), 281–292 (2008)
16. Garland, M., Heckbert, P.: Surface simplification using quadric error metrics. In: Proc. of SIGGRAPH, pp. 209–216 (1997)
17. Wang, R., Hang, S., Ye, X.: A novel simplification algorithm for point-sampled surfaces. In: International Conference on Multimedia and Ubiquitous Engineering, pp. 573–578 (2007)
18. Lee, P., Jong, B.: Point-based Simplification Algorithm. WSEAS Transactions on Computer Research 3(1), 61–66 (2008)
19. Jong, B., Lee, P.: Hierarchical Point Simplification Using Coplanar Criterion. In: Tseng 2006 IEEE Region 10 Conference, pp. 1–4 (2006)
20. Ohtake, Y., Belyaev, A., Seidel, H.-P.: An integration approach to meshing scattered point data. In: ACM Symposium on Solid and Physical Modeling, pp. 61–69 (2005)
21. Dey, T.K., Goswami, S.: Tight Cocone: a water tight surface reconstruction reconstructor. In: Proc. of 8th ACM Sympos. Solid Modeling Appl., pp. 127–134 (2003)
22. Tagliasacchi, A., Zhang, H., Cohen-Or, D.: Curve skeleton extraction from incomplete point cloud. ACM Transactions on Graphics 28(3), 1–9 (2009)
23. Nehab, D., Rusinkiewicz, S., Davis, J., et al.: Efficiently combining positions and normals for precise 3D geometry. ACM Transactions on Graphics 25(3), 560–568 (2005)
24. Hoppe, H., DeRose, T., Duchamp, T., et al.: Surface reconstruction from unorganized points. In: Proc. of SIGGRAPH, pp. 71–79 (1992)
25. Wu, J., Wei, M., Li, Y., et al.: Scale-adaptive surface modeling of vascuar structures. BioMedical Engineering Online 9, 75 (2010)
26. Koenderink, J.: What does the occluding contour tell us about solid shape? Perception 13, 321–330 (1984)
27. Gooch, B., Sloan, P.J., Gooch, A., et al.: Interactive technical illustraction. In: Proc. ACM Symposium on 3D Interactive Graphics, pp. 31–38 (1999)
28. Hertzmann, A., Zorin, D.: Illustrating smooth surfaces. In: Proc. of SIGGRAPH, pp. 517–526 (2000)
29. Heckbert, P.S., Garland, M.: Optimal triangulation and quadric-based surface simplification. Computational Geometry: Theroy and Application 14(1-3), 49–65 (1999)

Physically-Based Haptic Rendering for Virtual Hand Interaction

Yang Wenzhen[1,2,*], Pan Zhigen[1,3], and Chen Wenhua[2]

[1] Department of Computer Science, Zhejiang University, China
[2] Department of Mechanism and Automation, Zhejiang Sci-Tech. University, China
[3] Digital Media and HCI Research Center, Hangzhou Normal University, China
{ywz,chenwh}@zstu.edu.cn, zhigengpan@gmail.com

Abstract. Based on our previous research, this paper integrated four aspects of the haptic rendering for virtual hand interaction, which included the contact, static grasp, movement, and collision of virtual hand. We presented a method of contact force rendering of virtual hand based on the human fingers' force characteristics. A physically-based static grasp force rendering method was used to calculate the grasp force for virtual hand holding objects. Then, we deduced the algorithms to generate the realistic action force of the virtual hand grasping an object and moving in virtual environments. The algorithms were based on the kinematics and dynamics of the robot theory. With the impulse theorem and coefficient of elastic recovery, we analyzed the collision force with friction for virtual hand interaction. A special series of experimental results showed that the physically-based haptic rendering approaches for virtual hand interaction were computationally efficient while retaining a good level of realism.

Keywords: Haptic rendering, Virtual hand, Interaction, Physically-based.

1 Introduction

The virtual hand is a better interactive mode than traditional interactive modes in virtual environments [1]. While virtual hands operating virtual objects, users hope the virtual reality systems could not only provide vivid visual scenes but also render realistic interaction force, which would improve the immersion and authenticity of the virtual environments. The realistic haptic rendering of virtual hand interaction has various applications [2], such as virtual surgery training [3], virtual assembly [4], virtual museums [5] and so on.

We divide the interaction force of virtual hand into three types: contact force[6], grasp force[7],[8] and collision force[9]. Once one or more virtual fingertips touch a virtual object, the contact force of virtual hand is generated. It is useful to evaluate the virtual objects' rigidity and viscosity. When two or more virtual fingers grasp an

[*] Ph.D of Computer Science, associate professor, visiting scholar of George Mason University, research field includes virtual reality and HCI.

M.L. Gavrilova and C.J.K. Tan (Eds.): Trans. on Comput. Sci. XVI, LNCS 7380, pp. 163–178, 2012.

object to keep its static equilibrium, the static grasp force of virtual hand is generated. It is useful to evaluate the virtual objects' weight or motion state etc.

Further, the static grasp force is different when the posture or motion state of the grasped object is changed, even if the grasp positions of the fingers keep fixed. For examples, the static grasp force of the fingers positive holding a basketball is less than upending the basketball at the same grasp positions. It's more strenuous to lift a shot with accelerated motion than with homogeneous velocity motion. So, it is necessary to generate the realistic action force of the virtual hand statically grasping an object and moving in virtual environments. For the collision force of virtual hand, that means when a virtual object grasped by the virtual hand collided with other objects, the realistic impulse force should be acted on the virtual hand.

Based on our previous research, this paper integrates these four aspects of the haptic rendering for virtual hand interaction, which includes the contact, static grasp, movement, and collision of virtual hand.

Haptic rendering algorithms simulating point-contact, such as the proxy [10] and the god-object [11], had been popular for a decade thanks to their computational efficiency. An alternative way of distinguishing the existing haptic rendering techniques was based on the type of haptic interaction: point-based or ray-based. In point-based haptic interactions [11],[12],[13], only the end point of the haptic device, also known as the haptic interface point (HIP) or avatar of a virtual finger, interacted with objects. Each time the user moved the generic probe of the haptic device, the collision detection algorithms checked to see if the end point was inside the virtual object. If so, the depth of indentation was calculated as the distance between the current HIP and a surface point, such as the nearest surface point. Then according to the linear spring law, $F = kx$, where k was the stiffness of the object and x was the depth of indentation, the contact force was calculated. In ray-based haptic interactions [14], the generic probe of the haptic device was modeled as a finite ray whose orientation was taken into account, and the collisions were checked between the ray and the objects. The collision detection algorithms returned the collision point, which was the intersection point between the ray and surface of the object. The distance between the collision point and the HIP along the surface normal at the collision point was taken to be the depth of indentation. Then the contact force was also calculated using the linear spring law.

Borst et al. presented a spring model for virtual grasping that couples tracked hand configuration to a virtual hand model controlled by physical simulation, and also presented a new force rendering equation that uses the forces and torques of the spring model to render forces (or intensity levels) for haptic feedback gloves [15]. Popescu et al. applied a point-based algorithm [16] to grid points of meshes placed on virtual fingertips, summing mesh forces and applying a force mapping step that mapped resulting force to an actuator's line of action [17]. Bergamasco et al. introduced physically-based object response by applying a force vector that included components of static friction, dynamic friction, normal contact forces, and external forces. The calculations were based on grids of control points placed on palmar sides of the virtual hand's phalanges and palm [18]. Tzafestas investigated problems related particularly to the haptic interaction between the human operator and a virtual environment, and proposed a method to render interaction force, which was based on the use of a weighted pseudo-inverse for the solution of the nonlinear optimization problem and the computation of feedback forces (or torques) to be applied on individual fingers/phalanges (or joints) of the human hand [19].

Iwata's system transmitted maximum available torque to fingers that contacted a rigid object, and computed force and moment for palm feedback based on virtual palm and object configurations[20]. For the glove-mounted CyberGrasp system, Immersion developed the VirtualHand software toolkit. Details of its rendering method were not known [21].

These algorithms were useful to render the haptic and interact with objects for virtual hands, but the authenticity of the interaction force had limited their applications. This paper focuses on the topic of realistic haptic rendering for virtual hand interaction, and discusses four aspects: 1) contact force generation with human fingers' force characteristics; 2) physically-based static grasp force generation for virtual hand holding objects; 3) realistic action force generation for virtual hand grasping and moving an object in virtual environments; 4) realistic collision force with friction generation for virtual hand interaction in virtual environments.

2 Contact Force Generation with Human Fingers' Force Characteristics

Both point-based [11] and ray-based [14] techniques didn't take into account the real haptic characteristics of the human finger pad. We presented a realistic contact force rendering method with the human fingertip characteristics [6]. The method had 5 steps. The first step was to measure the human finger pads' contact force and contact area respectively by a force measuring apparatus. The second step was to analyze the intrinsic relationship between them, and deduced a contact force model to describe the real contact force properties of the human fingertips. The third was to build a virtual hand model by emulating the human fingers' physiological tissues. And then an algorithm to calculate the virtual fingertips' contact area was proposed in the fourth step. Based on the contact force model, the virtual contact areas algorithm and the virtual hand model, the fifth was to calculate the contact force while one or more virtual fingertips touching a virtual object suface.

We achieved the contact force model to render the contact force for virtual hand interaction in virtual environments.

$$\ln F = \ln b_0 + b_1 S \quad \text{or} \quad F = b_0 e^{b_1 S} \tag{1}$$

Here, F was the dependent variable of the fingertips' contact force, S was the variable of the fingertips' contact area, b_0 was a constant and b_1 was a regression coefficient.

The architecture of contact force rendering with human fingers' force characteristics was illustrated in Fig. 1.

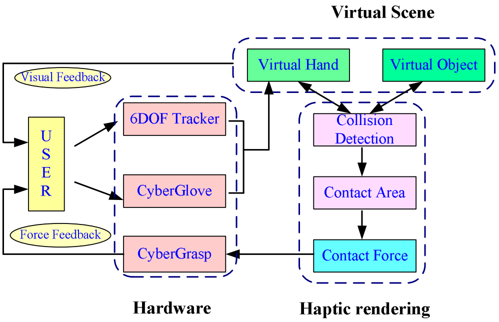


Fig. 1. Architecture of contact force rendering

3 Physically-Based Grasp Force Generation for Virtual Hand Statically Grasping Objects

Based on the theory of robot's hand grasp, a physically-based general force model of virtual hand grasp was deduced. To overcome the solution uncertainty of the general force model, we proposed a minimum force spiral optimized model to solve the realistic grasp force screw [7].

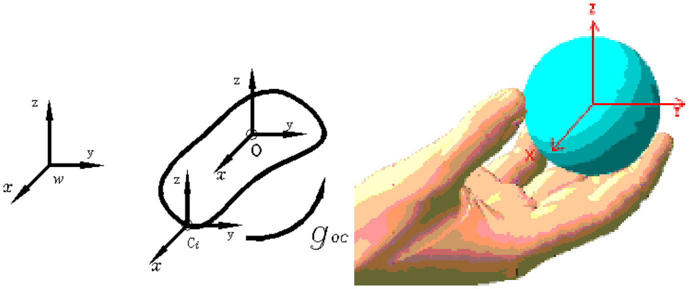


Fig. 2. Coordinate systems of virtual hand interaction

3.1 Coordinate System of Virtual Hand Interaction

There were many elements in a virtual reality system with virtual hand interaction, such as virtual space, virtual objects, virtual hands etc. For laconic and efficient maintaining the virtual reality system, the coordinate systems for virtual hand interaction must be constructed to describe, analyze, and calculate the relationships among virtual environments, virtual hand, and virtual objects etc.

The coordinate systems of virtual hand interaction included the world coordinate system, the virtual object coordinate system, the virtual object initial coordinate system, the virtual hand grasp coordinate system, and collision coordinate system etc. (see Fig.2 and Fig.3). Based on the Cartesian coordinate right-hand rule, the world coordinate system $xyzw$ was static, one and only one in the virtual environment. The virtual object coordinate system $\overline{xyzo}$ was dynamic, its reference frame was $xyzw$, its origin of coordinate was the mass center of the virtual object. The virtual object initial coordinate system $xyzo$ was a posture fixed and position changeable coordinate system, its origin of coordinate was the mass center of the virtual object, its posture always kept the same as $xyzw$, and its position moved with the grasped object together. The virtual hand grasp coordinate system $xyzc_i$ was temporary constructed when the virtual hand grasped an object, and deleted when the virtual hand released an object.

During the virtual hand interacted with the virtual environment, we accomplished the laconic description and highly efficient management of the virtual reality system by constructing, maintaining, modifying, deleting these coordinate systems.

3.2 Minimum Force Spiral Optimized Model

Virtual hand statically grasping object means the force screw equilibrium between the virtual finger force screw and the outside force screw acting on the grasped object. In the 6 D force space, the resultant force screw of the virtual fingers F_o must be equal to the other outside force screw F_e $(F_e^x, F_e^y, F_e^z, M_e^x, M_e^y, M_e^z)$. So, the physically-based general force model for virtual hand grasp was listed:

$$F_o + F_e = 0 \text{ or } Gf_c = -F_e \tag{2}$$

Here, G was the mapping between F_o and F_e. Formula 2 was expanded according to the soft finger contact model:

$$\begin{cases} \sum_{i=1}^{k}(\cos\theta_{i1}\cos\theta_{i2}f_{ci}^x - \sin\theta_{i1}f_{ci}^y + \cos\theta_{i1}\sin\theta_{i2}f_{ci}^z) = -F_e^x \\ \sum_{i=1}^{k}(\sin\theta_{i1}\cos\theta_{i2}f_{ci}^x + \cos\theta_{i1}f_{ci}^y + \sin\theta_{i1}\sin\theta_{i2}f_{ci}^z) = -F_e^y \\ \sum_{i=1}^{k}(-\sin\theta_{i2}f_{ci}^x + \cos\theta_{i2}f_{ci}^z) = -F_e^z \\ \sum_{i=1}^{k}[(-\sin\theta_{i2}y_i - \sin\theta_{i1}\cos\theta_{i2}z_i)f_{ci}^x - \cos\theta_{i1}z_i f_{ci}^y + (\cos\theta_{i2}y_i - \sin\theta_{i1}\sin\theta_{i2}z_i)f_{ci}^z + \cos\theta_{i1}\sin\theta_{i2}m_{ci}^z] = -M_e^x \\ \sum_{i=1}^{k}[(\cos\theta_{i1}\cos\theta_{i2}z_i + \sin\theta_{i2}x_i)f_{ci}^x - \sin\theta_{i1}z_i f_{ci}^y + (\cos\theta_{i1}\sin\theta_{i2}z_i - \cos\theta_{i2}x_i)f_{ci}^z + \sin\theta_{i1}\sin\theta_{i2}m_{ci}^z] = -M_e^y \\ \sum_{i=1}^{k}[(\sin\theta_{i1}\cos\theta_{i2}x_i - \cos\theta_{i1}\cos\theta_{i2}y_i)f_{ci}^x + (\cos\theta_{i1}x_i + \sin\theta_{i1}y_i)f_{ci}^y + (\sin\theta_{i1}\sin\theta_{i2}x_i - \cos\theta_{i1}\sin\theta_{i2}y_i)f_{ci}^z + \cos\theta_{i2}m_{ci}^z] = -M_e^z \end{cases} \tag{3}$$

Unfortunately, the virtual finger force screw f_c in the general force model can only be solved in theory. It is because the general force model (Formula 2) in generalized force space has only 6 equations of equilibrium as shown in Formula 3, which can most solve 6 variables, however, least 2 virtual finger force screws, each has 4 variables (f_i^x , f_i^y , f_i^z , m_i^z) , need to be solved for virtual hand grasping an object. So, we proposed the minimum force spiral optimized model to solve the f_c.

Optimized aim function:

$$\text{Minimal } \sum_{i=1}^{k}\left[\sum_{j=1}^{3}\left(f_{ci}^{j}\right)^{2}+\left(\frac{f_{ci}^{4}}{\gamma}\right)^{2}\right] \tag{4}$$

Constrain conditions:

$$Gf_c=-F_e\text{, }\sqrt{(f_{ci}^{x})^2+(f_{ci}^{y})^2}\le \mu f_{ci}^{z}\text{, } f_{ci}^{z}\ge 0\text{, and } \left|m_{ci}^{z}\right|\le \gamma f_{ci}^{z}$$

Here: $\gamma > 0$ was friction factor of moment, μ was frictional coefficient , and $i=1,2,3.....k, k\in N$.

The minimum aim function meant the resultant force of the virtual fingers should be minimum, and at the same time should keep the static equilibrium of the grasped object.

4 Grasp Force Rendering for Virtual Hand Moving Objects

Referencing the manipulation theory of the dexterous robot hands [22], this session presents a realistic static grasp force rendering method for virtual hand moving objects. The method had 2 parts. The first part was to track the “ghost” point of the grasped object, which was used to judge whether or not the posture change of the grasped object. If the grasped object posture was changed, an algorithm was used to calculate the distribution of the external force imposed on the grasped object in the new posture, and the static grasp force of the virtual hand was regenerated according to the minimum force spiral optimized model [7]. If not, the static grasp force kept the previous value. Based on the kinematics theory, the second part was to track the mass center of the grasped object, which was used to judge whether or not the motion state change of the grasped object. If the grasped object motion state was changed, the resultant force which caused the motion state change of the grasped object could be achieved by the dynamics theory, and the static grasp force of the virtual hand was regenerated according to the minimum force spiral optimized model in the new motion state. If not, the static grasp force kept the previous value. Even the posture and motion state of the grasped virtual object were changed at the same time, the static grasp force of the virtual hand could also be regenerated by the proposed method.

4.1 Judgment for the Posture Change of Grasped Object

We proposed a way to judge whether or not the posture change of the grasped object. If the grasped object posture was changed, an algorithm was used to calculate the distribution of the external force imposed on the grasped object in the new posture, and the grasp force of the virtual hand was regenerated in real time.

Along the coordinate axis oz of the initial coordinate system $xyzo$ of the grasped object, we selected a “ghost point” T in the grasped object. Vector $P_{wT}\,(x_T, y_T, z_T)$ was the "ghost point" T referenced to the world coordinate system. Vector $P_{wo}(x_o, y_o, z_o)$ was the mass center O of the grasped object referenced to the world coordinate system. Vector $\Delta T(\Delta x_T, \Delta y_T, \Delta z_T)$ was the difference between P_{wT} and P_{wo} :

$$\Delta T = P_{wT} - P_{wo} \tag{5}$$

When the virtual hand moved the object from time t to time $t + \Delta t$, the virtual reality system recorded the vector $\Delta T^t(\Delta x_T^t, \Delta y_T^t, \Delta z_T^t)$ at time t and $\Delta T^{t+\Delta t}(\Delta x_T^{t+\Delta t}, \Delta y_T^{t+\Delta t}, \Delta z_T^{t+\Delta t})$ at time $t + \Delta t$ respectively. If $\Delta T^t \neq \Delta T^{t+\Delta t}$, it meant the grasped object's posture was changed, else the grasped object kept the previous posture (see Fig. 3).

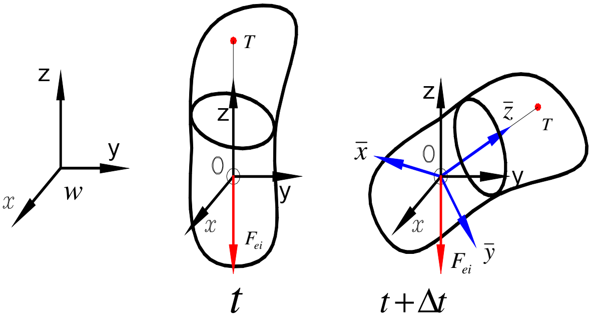


Fig. 3. Posture change of the grasped object

4.2 Calculation of the Grasp Force for Grasped Object in Arbitrary Posture

Without loss of generality, an algorithm was discussed to calculate the static grasp force, when the grasped object's posture changed from the initial posture to an arbitrary posture. The key of this algorithm was to realize the transform of the external force screws acting on the grasped object. All external force screws, except the static grasp force screws, was referenced to the virtual object initial coordinate system $xyzo$. In the arbitrary posture of the grasped object, the external force screws must be transformed to the corresponding virtual object coordinate system $\overline{xyzo}$.

A transformation relation between the $xyzo$ and the $\overline{xyzo}$ was deduced to realize the external force screw $F_{ei}(F_{ei}^x, F_{ei}^y, F_{ei}^z, M_{ei}^x, M_{ei}^y, M_{ei}^z)$ referenced the $xyzo$ transforming to $\overline{F}_{ei}(\overline{F}_{ei}^x, \overline{F}_{ei}^y, \overline{F}_{ei}^z, \overline{M}_{ei}^x, \overline{M}_{ei}^y, \overline{M}_{ei}^z)$ referenced to the $\overline{xyzo}$ [8].

So, the force screw equilibrium constraint condition of the grasped object in arbitrary posture was represented as:

$$\begin{cases}\sum_{i=1}^{k}(\cos\theta_{i1}\cos\theta_{i2}f_{ci}^{x}-\sin\theta_{i1}f_{ci}^{y}+\cos\theta_{i1}\sin\theta_{i2}f_{ci}^{z})=-\sum_{i=1}^{k}\bar{F}_{ei}^{x}\\ \sum_{i=1}^{k}(\sin\theta_{i1}\cos\theta_{i2}f_{ci}^{x}+\cos\theta_{i1}f_{ci}^{y}+\sin\theta_{i1}\sin\theta_{i2}f_{ci}^{z})=-\sum_{i=1}^{k}\bar{F}_{ej}^{y}\\ \sum_{i=1}^{k}(-\sin\theta_{i2}f_{ci}^{x}+\cos\theta_{i2}f_{ci}^{z})=-\sum_{i=1}^{k}\bar{F}_{ei}^{z}\\ \sum_{i=1}^{k}[(-\sin\theta_{i2}y_i-\sin\theta_{i1}\cos\theta_{i2}z_i)f_{ci}^{x}-\cos\theta_{i1}z_i f_{ci}^{y}+(\cos\theta_{i2}y_i-\sin\theta_{i1}\sin\theta_{i2}z_i)f_{ci}^{z}+\cos\theta_{i1}\sin\theta_{i2}m_{ci}^{z}]=-\sum_{i=1}^{k}\bar{M}_{ei}^{x}\\ \sum_{i=1}^{k}[(\cos\theta_{i1}\cos\theta_{i2}z_i+\sin\theta_{i2}x_i)f_{ci}^{x}-\sin\theta_{i1}z_i f_{ci}^{y}+(\cos\theta_{i1}\sin\theta_{i2}z_i-\cos\theta_{i2}x_i)f_{ci}^{z}+\sin\theta_{i1}\sin\theta_{i2}m_{ci}^{z}]=-\sum_{i=1}^{k}\bar{M}_{ei}^{y}\\ \sum_{i=1}^{k}[(\sin\theta_{i1}\cos\theta_{i2}x_i-\cos\theta_{i1}\cos\theta_{i2}y_i)f_{ci}^{x}+(\cos\theta_{i1}x_i+\sin\theta_{i1}y_i)f_{ci}^{y}+(\sin\theta_{i1}\sin\theta_{i2}x_i-\cos\theta_{i1}\sin\theta_{i2}y_i)f_{ci}^{z}+\cos\theta_{i2}m_{ci}^{z}]=-\sum_{i=1}^{k}\bar{M}_{ei}^{z}\end{cases} \quad (6)$$

Equation 6 displaced the corresponding constrain condition of the minimum force spiral optimized model. Calculating the optimization model, the realistic static grasp force of the virtual hand at arbitrary posture was rendered in real time.

4.3 Grasp Force Generation for the Motion State Change

The motion states of the grasped object can be divided into 5 types in virtual environments: 1) Translation with homogeneous velocity, both posture and motion state keep stabilization, and the static grasp force screws are invariable. 2) Translation with variable velocity, its posture keeps stable, its motion state is variable, and the static grasp force screws are variable. 3) Rotation with homogeneous velocity, its posture is variable, its motion state keeps stable, and the static grasp force screws are variable. 4) Rotation with variable velocity, its posture and motion state are variable, and the static grasp force screws are variable. 5) Compound motion, which is the combination of above 4 motion states.

An algorithm was proposed to analyze the static grasp force screw for the translation state change of the grasped object. The translation acceleration of the grasped object was calculated by tracking the mass center of the grasped object, and the resultant force which caused the translation state change could be obtained by the dynamics theory. Transforming the resultant force to the corresponding virtual object coordinate system, the grasp force of the virtual hand was regenerated according to the minimum force spiral optimized model in the new motion state.

During the variable velocity translation of the grasped object, the position information was sampled in real time. Referencing the world coordinate system, $P_o(x_o, y_o, z_o)$ was the position vector of the mass center of the grasped object in t time. In the movement interval $t+\Delta t_a$, the variable velocity translation was divided into two periods (initial movement period and end movement period). In the initial movement period (the interval was Δt_v), we got $P_{os}(x_{os}, y_{os}, z_{os})$, $P'_{os}(x'_{os}, y'_{os}, z'_{os})$ as the starting location and the end location. In the end movement period (the interval was Δt_v), we got $P_{oe}(x_{oe}, y_{oe}, z_{oe})$、$P'_{oe}(x'_{oe}, y'_{oe}, z'_{oe})$ as the starting location and the end location. Here, $\Delta t_a \geq 2\Delta t_v$ .Thus, the grasped object moved from $P_{os}(x_{os}, y_{os}, z_{os})$ to $P'_{oe}(x'_{oe}, y'_{oe}, z'_{oe})$, and its average translation velocity and acceleration were obtained by the follow equations:

$$v_s^* = (\frac{x'_{os} - x_{os}}{\Delta t_v}, \frac{y'_{os} - y_{os}}{\Delta t_v}, \frac{z'_{os} - z_{os}}{\Delta t_v}) \tag{7}$$

$$v_e^* = (\frac{x'_{oe} - x_{oe}}{\Delta t_v}, \frac{y'_{oe} - y_{oe}}{\Delta t_v}, \frac{z'_{oe} - z_{oe}}{\Delta t_v}) \tag{8}$$

$$\begin{gathered} a^* = \frac{v_e^* - v_s^*}{\Delta t_a} = (\frac{(x'_{oe} - x_{oe}) - (x'_{os} - x_{os})}{\Delta t_a \Delta t_v}, \\ \frac{(y'_{oe} - y_{oe}) - (y'_{os} - y_{os})}{\Delta t_a \Delta t_v}, \frac{(z'_{oe} - z_{oe}) - (z'_{os} - z_{os})}{\Delta t_a \Delta t_v}) \end{gathered} \tag{9}$$

By the dynamics theory, the resultant force $F_a(F_a^x, F_a^y, F_a^z)$ producing the translation acceleration was calculated:

$$F_a^x = ma_x^* = m\frac{(x'_{oe} - x_{oe}) - (x'_{os} - x_{os})}{\Delta t_a \Delta t_v} \tag{10}$$

$$F_a^y = ma_y^* = m\frac{(y'_{oe} - y_{oe}) - (y'_{os} - y_{os})}{\Delta t_a \Delta t_v} \tag{11}$$

$$F_a^z = ma_z^* = m\frac{(z'_{oe} - z_{oe}) - (z'_{os} - z_{os})}{\Delta t_a \Delta t_v} \tag{12}$$

It's obvious the resultant force $F_a(F_a^x, F_a^y, F_a^z)$ was referenced to the world coordinate system. If the posture of the variable velocity translation grasped object was the same as its initial coordinate system, $F_a(F_a^x, F_a^y, F_a^z)$ could be added to the force screw equilibrium constraint condition directly.

If the posture of the variable velocity translation grasped object was changed, it's necessary to transform the $F_a(F_a^x, F_a^y, F_a^z)$ to $\overline{F}_a(\overline{F}_a^x, \overline{F}_a^y, \overline{F}_a^z)$, then added to the force screw equilibrium constraint condition as equation 13.

$$\begin{cases} \sum_{i=1}^{k}(\cos\theta_{i1}\cos\theta_{i2}f_{ci}^x - \sin\theta_{i1}f_{ci}^y + \cos\theta_{i1}\sin\theta_{i2}f_{ci}^z) = -\sum_{i=1}^{k}\overline{F}_{ei}^x + \overline{F}_a^x \\ \sum_{i=1}^{k}(\sin\theta_{i1}\cos\theta_{i2}f_{ci}^x + \cos\theta_{i1}f_{ci}^y + \sin\theta_{i1}\sin\theta_{i2}f_{ci}^z) = -\sum_{i=1}^{k}\overline{F}_{ej}^y + \overline{F}_a^y \\ \sum_{i=1}^{k}(-\sin\theta_{i2}f_{ci}^x + \cos\theta_{i2}f_{ci}^z) = -\sum_{i=1}^{k}\overline{F}_{ei}^z + \overline{F}_a^z \\ \sum_{i=1}^{k}[(-\sin\theta_{i2}y_i - \sin\theta_{i1}\cos\theta_{i2}z_i)f_{ci}^x - \cos\theta_{i1}z_i f_{ci}^y + (\cos\theta_{i2}y_i - \sin\theta_{i1}\sin\theta_{i2}z_i)f_{ci}^z + \cos\theta_{i1}\sin\theta_{i2}m_{ci}^z] = -\sum_{i=1}^{k}\overline{M}_{ei}^x \\ \sum_{i=1}^{k}[(\cos\theta_{i1}\cos\theta_{i2}z_i + \sin\theta_{i2}x_i)f_{ci}^x - \sin\theta_{i1}z_i f_{ci}^y + (\cos\theta_{i1}\sin\theta_{i2}z_i - \cos\theta_{i2}x_i)f_{ci}^z + \sin\theta_{i1}\sin\theta_{i2}m_{ci}^z] = -\sum_{i=1}^{k}\overline{M}_{ei}^y \\ \sum_{i=1}^{k}[(\sin\theta_{i1}\cos\theta_{i2}x_i - \cos\theta_{i1}\cos\theta_{i2}y_i)f_{ci}^x + (\cos\theta_{i1}x_i + \sin\theta_{i1}y_i)f_{ci}^y + (\sin\theta_{i1}\sin\theta_{i2}x_i - \cos\theta_{i1}\sin\theta_{i2}y_i)f_{ci}^z + \cos\theta_{i2}m_{ci}^z] = -\sum_{i=1}^{k}\overline{M}_{ei}^z \end{cases} \tag{13}$$

Equation 13 displaced the corresponding constrain condition of the minimum force spiral optimized model. Calculating the optimization model, the realistic static grasp force of the virtual hand in new motion state was rendered in real time.

5 Collision Force Rendering for Virtual Hand Interaction

When a virtual object grasped by virtual hand collided with other objects, impulsive forces should act on the virtual hand. Based on the impulse theorem and the coefficient of elastic recovery, we discussed the collision force rendering for virtual hand interaction in virtual environments. Once a virtual object grasped by virtual hand collided with other objects, we calculated the collection force with the coulomb's friction model in 3D space. Then, the collection force was transferred to the virtual hand by the minimum force spiral optimized model. So, realistic impulsive force acting on the virtual fingers could be rendered [9].

Referring to the physical collision, two assumptions were adopted to simulate the virtual collision.

1) Collision force was larger than general force, such as gravity, spring force.
2) Displacement was not be happened after collision.

Based on the impulsive theorem and physical means of restitution coefficient, the collision forces were calculated [9]:

$$f_1^{\tau} = \sum_{i=1}^{j} f_{1i}^{\tau} = \sum_{i=1}^{j} \frac{m_1(u_{1i}^{\tau} - v_{1i}^{\tau})}{t_1 - t_0} \tag{14}$$

$$f_1^{n} = \sum_{i=1}^{j} f_{1i}^{n} = \sum_{i=1}^{j} \frac{m_1(u_{1i}^{n} - v_{1i}^{n})}{t_1 - t_0} \tag{15}$$

The collision force moment were calculated:

$$L_1^{\tau} = \sum_{i=1}^{j} L_{1i}^{\tau} = \sum_{i=1}^{j} f_{1i}^{\tau} \times r_1^{i} \tag{16}$$

$$L_1^{n} = \sum_{i=1}^{j} L_{1i}^{n} = \sum_{i=1}^{j} f_{1i}^{n} \times r_1^{i} \tag{17}$$

Equation 14-17 displaced the corresponding constrain condition of the minimum force spiral optimized model. Calculating the optimization model, the realistic collision force was rendered in real time.

6 Experiments

A three-layer virtual hand model to handle the motion as well as the deformation was adopted. The virtual hand model was consisted of the skeleton layer, the muscle layer and the skin layer [23]. The skin layer was the polygonal mesh used for displaying, the skeleton model was built based on the anatomic structure of the human hand, and the muscle layer simulated the function of the human hand muscle. For strengthening visual realism, we mapped the images captured from real human hand as textures to the skin layer.

We had implemented the physically-based haptic rendering for virtual hand interaction in a CAVE-based virtual environment using the CAVELib™. The CAVELib™ was a powerful API that provided the cornerstone for creating robust interactive three-dimensional environments [24]. An ascension 6 degrees of freedom (DOF) tracking sensor was used for tracking user's hand position. The CyberGlove® and the CyberGrasp® [16] were used to capture finger motions and provided force feedback respectively. The program ran on an SGI Onxy2 (with 4 CPUs and 2 IR4 graphic pipelines). High-resolution stereo images were projected onto four imaging surfaces of the CAVE by four projectors. CrystalEyes shutter glasses and emitters were used to obtain stereoscopic view. And the arithmetic solver was LINDO API 3.0 [25].

6.1 Grasp Force Rendering for the Grasped Ball

The virtual hand static grasped a 30 N ball, its radius was 2 cm, the frictional coefficient μ was 0.6, and the moment friction factor γ was 60 cm. Except the grasp force, only the gravity acted on the ball. The center of mass and the origin of coordinate were on the mass centre of the ball.

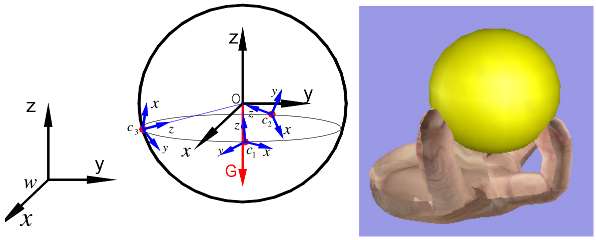


Fig. 4. Three fingers symmetric grasping a ball

Referenced to the ball coordinate system, the grasp positions were $(1.67, 0.97, -0.5)$, $(-1.67, 0.97, -0.5)$, $(0, -1.93, -0.5)$, and the inner normal vectors were $\mathbf{n}_1 = \{-1.67, -0.97, 0.5\}$, $\mathbf{n}_2 = \{1.67, -0.97, 0.5\}$, $\mathbf{n}_3 = \{0, 1.93, 0.5\}$ respectively (see Fig 4.). For testifying our proposed algorithms, the grasp positions and values were specially chosen. At this grasp posture, oy axis was the symmetry axis for the location of the index finger and middle finger. Three fingers static grasp the ball little below its max diameter, and equally located on the same horizontal level.

In this case, the three finger force screws $f_c = (f_{c1}^x, f_{c1}^y, f_{c1}^z, m_{c1}^z, f_{c2}^x, f_{c2}^y, f_{c2}^z, m_{c2}^z, f_{c3}^x, f_{c3}^y, f_{c3}^z, m_{c3}^z)$ were solved by the minimum force spiral optimized model, $f_c = (-7.22, 0, 12.04, 0.27, -7.22, 0, 12.04, 0.27, -7.22, 0, 12.04, -0.54)$. Namely, the resultant force of the thumb, index finger and middle finger $F_{c1} = F_{c2} = F_{c3} = 14.04N$.

6.2 Grasp Force Rendering for the Ball Rotating a Circle around the wx Axis

When the virtual hand grasping the ball rotated around the wx axis, The values of the grasp force were calculated to prove the correctness and feasibility of the proposed methods.

Figure 5 showed the partial scenes of the virtual hand grasping the ball rotating a whole circle around the wx axis. Figure 6 showed each finger's grasp force curve of this experiment respectively. Figure 7 displayed the resultant grasp force for this experiment. Partial values of the grasp force of the ball rotating around the wx axis were listed on table1.

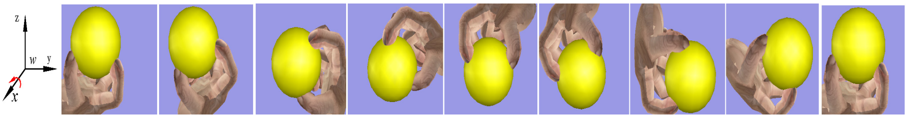
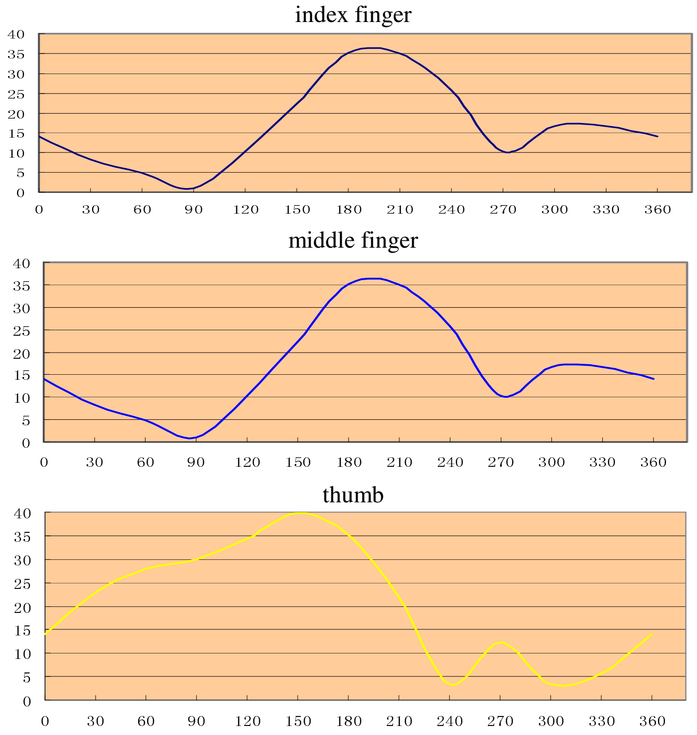


Fig. 5. A ball rotating around the wx axis

Observing figure 6 and figure 7, we discovered some interesting phenomena: 1) The grasp forces of the index finger and middle finger were always the same. 2) When the ball rotated around the wx axis from 0° to 90°, the grasp force of the thumb was continuous increment, but the grasp forces of the index finger and middle finger were continuous decrement. 3) Nearby the 90° grasp position, the grasp force of the thumb almost equated the ball weight, and the grasp forces of the index finger and middle finger approximated to zero. 4) Nearby the 240° grasp position and the 300° grasp position, the grasp force of the thumb was same and almost least, but there existed the distinctive discrepancy of the grasp forces of the index finger and middle finger at the two grasp positions, one was $F_{c1} = F_{c2} = 25.69N$, another was $F_{c1} = F_{c2} = 16.71N$.

These interesting phenomena were analyzed elaborately as follows: 1) The same grasp forces of the index finger and middle finger were caused by their grasp positions which were symmetrical with the oy axis. 2) When the ball rotated around the wx axis from 0° to 90°, the change trend of the grasp force (increment of the thumb and decrement of the index finger and middle finger) was quite accordant as the daily experiences. 3) Nearby the 90° grasp position but absolute not at the 90°, the thumb almost could hold the ball solely without the help of other fingers. It's decided by the grasp position of the thumb. 4) Nearby the 240° grasp position and the 300° grasp position, the action of the thumb was trifling, and the ball kept equilibrium with the index finger and middle finger. Nearby the 240° grasp position, the action of the index and middle finger was to avoid the ball falling down, but nearby the 300° grasp position, the index and middle finger was to lift the ball. It's obvious that the index and middle finger was more laborious to avoid the ball falling down than to lift the ball.


index finger

middle finger

thumb

Fig. 6. Static grasp force curve of each finger (rotating around the wx axis)

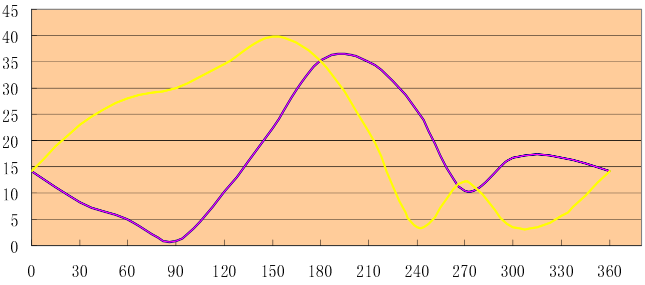


Fig. 7. Curves of the resultant grasp force (rotating around the wx axis)

Table 1. Partial values of the grasp force for rotating around the wx axis (Unit: N)

Rotation angle	Index finger (F_{c1})	Middle finger (F_{c2})	Thumb (F_{c3})
0°	14.04	14.04	14.04
30°	8.264	8.264	22.946
60°	14.904	14.904	27.984
90°	0.912	0.912	29.95
120°	10.257	10.257	34.47
150°	22.38	22.38	39.745
180°	35.254	35.254	35.254
210°	35.06	35.06	21.823
240°	25.69	25.69	3.39
270°	10.35	10.35	12.27
300°	16.71	16.71	3.38
330°	16.74	16.74	5.678
360°	14.04	14.04	14.04

6.3 Collision Force Rendering for Virtual Hand Interaction

We had also carried the experiments to render the collision force. Experiment results showed that using our collision force generation method, the user could sense realistic impulsive forces via the CyberGrasp data glove caused by virtual collision [9].

7 Conclusions

The realistic haptic rendering is a complex phenomenon which is difficult to be emulated accurately in virtual environments. The physically-based haptic rendering for virtual hand interaction was introduced in this paper.

According to the interaction manners of virtual hand (the contact, static grasp, movement, and collision of virtual hand), three type forces (contact force, grasp force and collision force) were generated with physically-based methods. Our experiments showed that we could "reasonably" emulate the action force for virtual hand interaction in a virtual environment.

Many applications require the dynamic grasp force generation in virtual environments. The method is therefore being developed for the dynamic manipulations of virtual hand whilst trying to keep the algorithm clear and efficient.

Acknowledgement. This project is supported by the Zhejiang Science Technology Plan Project, China (No. 2010R50005), the National Fundamental Research Project of China (973 Project) (2002CB312100), the Excellent Young Team of Zhejiang Provincial Education Department (R107725), the Zhejiang Science Technology Plan Project, China (No. 2009C31021), and the Doctor Research of Zhejiang Sci-Tech University(0703667-Y). Thanks go to Prof. Shu Mingao, Dr. Wang Huageng, Dr. Zhu ZhenHua, Mr. Luo Yang for fruitful discussions. The authors also thank the anonymous reviewers for valuable comments and suggestions.

References

1. Moccozet, L., Huang, Z., Thalmann, M.: Virtual Hand Interactions with 3D World. In: Proc. Multimedia Modeling 1997, pp. 307–322. World Scientific (1997)
2. Srinivasan, M., Basdogan, C.: Haptics in Virtual Environments: Taxonomy, Research Status, and Challenges. Computer & Graphics 21, 393–404 (1997)
3. Batteau, L., Liu, A., Maintz, J., Bhasin, Y., Bowyer, M.: A Study on the Perception of Haptics in Surgical Simulation. In: Cotin, S., Metaxas, D. (eds.) ISMS 2004. LNCS, vol. 3078, pp. 185–192. Springer, Heidelberg (2004)
4. Stewart, P., Buttolo, P., Chen, Y.: CAD Data Representations for Haptic Virtual Prototyping. In: Proc. ASME Design Engineering Technical Conferences, pp. 14–17 (1997)
5. Baxter, L., Otaduy, F.: Haptic Interaction for Creative Pocesses with Simulated Media. In: Proc. IEEE Robotics and Automation, vol. 1, pp. 598–604 (2002)
6. Wenzhen, Y., Shuming, G., Huagen, W., Yushen, L., Wenhua, C.: Contact Force Rendering of Virtual Hand Interaction Based on Human fingers' force characteristics. Journal of Zhejiang University 42(12), 2145–2150 (2008)
7. Wenzhen, Y., Shuming, G., Huagen, W., ZhenHua, Z., Yang, L.: Force Generation and Feedback for Physically-Based Virtual Hand Grasp. Chinese Journal of Computer 28(6), 959–964 (2005)
8. Wenzhen, Y., Wenhua, C.: Haptic Rendering of Virtual Hand Moving Objects. In: 2011 International Conference on Cyberworlds, pp. 113–119 (2011)
9. Wenzhen, Y., Shuming, G., Huagen, W., ZhenHua, Z., Yang, L.: Physically-based Haptic Display of Rigid Body Collisions with Virtual Hand Interaction. Chinese Journal of Computer 29(12), 2096–2173 (2006)
10. Ruspini, D., Kolarov, K., Khatib, O.: The Haptic Display of Complex Graphical Environments. In: Proc. 24th Annual Conference on Computer Graphics and Interactive Techniques, pp. 345–352 (1997)
11. Zilles, C., Salisbury, J.: A Constraintbased God-object Method for Haptic Display. In: Proc. IEE/RSJ International Conference on Intelligent Robots and Systems, Human Robot Interaction, and Cooperative Robots, vol. 3, pp. 146–151 (1995)
12. Massie, T., Salisbury, J.: The PHANToM Haptic Interface: a Device for Probing Virtual Objects. In: Pro. ASME Dynamic Systems and Control Division, DSC, vol. 55(1), pp. 295–301 (1994)
13. Salisbury, J., Brock, D., Massie, T., Swarup, N., Zilles, C.: Haptic Rendering: Programming Touch Interaction with Virtual Objects. In: Proc. ACM Symposium on Interactive 3D Graphics, pp. 123–130 (1995)
14. Basdogan, C., Ho, C., Srinivasan, M.: A Ray-based Haptic Rendering Technique for Displaying Shape and Texture of 3D Objects in Virtual Environments. In: Proc. ASME Dynamic Systems and Control Division, vol. 61, pp. 77–84 (1997)
15. Borst, C., Indugula, A.: Realistic Virtual Grasping. In: Proc. IEEE Virutal Reality 2005, pp. 91–98 (2005)
16. Ho, C., Basdogan, C., Srinivasan, M.: Haptic Rendering: Point and Ray-Based Interactions. In: Proceedings of the Second PHANToM Users Group Workshop, pp. 6–10 (1997)
17. Popescu, V., Bouzit, M.: Virtual Reality Simulation Modeling for a Haptic Glove. In: Computer Animation 1999 Conference, pp. 195–200 (1999)
18. Bergamasco, M., Deglapos, P., Bergamasco, M.: A Realistic Approach for Grasping and Moving Virtual Objects. In: IEEE International Conference on Intelligent Robots and Systems (IROS 1994), pp. 717–724 (1994)

19. Tzafestas, C.S.: Whole-hand Kinesthetic Feedback and Haptic Perception in Dexterous Virtual Manipulation. IEEE Trans. on Sys. Man and Cybernatics 33(1), 100–113 (2003)
20. Iwata, H.: Artificial Reality with Force-feedback: Development of Desktop Virtual Space with Compact Master Manipulator. Computer Graphics 24(4), 165–170 (1990)
21. SDKHand, `http://www.immersion.com/products/3d/interaction/cybergrasp.shtml`
22. Murray, L.: Mathematical Introduction to Robotic Manipulation. CRC Press (1994)
23. Huagen, W., Yang, L., Shuming, G., Qunsheng, P.: Realistic Virtual Hand Modeling with Applications for Virtual Grasping. In: Proc. ACM SIGGRAPH International Conference on Virtual Reality Continuum and its Applications in Industry, pp. 81–87 (2004)
24. CAVELib Manual, `http://www.vrco.com/CAVE_USER`
25. LINDDO System Inc. LINDO API User's Manual, `http://www.lindo.com`

The Advanced Open Metaplastic Platform for Cyber Art

Gianluca Mura

Politecnico di Milano University, Milano, Italy
gianluca.mura@polimi.it

Abstract. This paper introduces an open and conceptual platform of metaplastic discipline for the realization of new media and interdisciplinary methodologies of Cyber Arts between reality and virtual realities. It explains the theoretical and artistic background of metaplastic metaspace and virtual worlds evolutions.

Keywords: Metaplastic, Virtual Worlds, Digital Art, Virtual Reality.

1 Introduction

The research is related to the previously published work about the term of Digital Metaplasticity that describes plastic qualities of the metaplastic media configurations and its expressions through the applications of abstract art languages and methodologies to virtual worlds. This paper re-defines the Metaplastic Metaspace model through enhanced social communications and creative possibilities of cultural production offered within the new virtual media.

1.1 Designing Poetics in Imaginary Space

The conceptual activity of designing Metaplastic Virtual Environments (VE) poses a number of problems to solve in different ways, through direct relations between the visitors' psychological being and both the embodying interface and the mediated content. The design's process complexity involves several fields of knowledge like technical and scientific information, social, cultural and psychological factors. The abstraction levels and the construction of meaning for multidisciplinary analysis requires the development of a multidimensional knowledge system[17]. "The medium of 'immersive virtual space',or virtual reality as it is generally known has intriguing potential as an arena for constructing metaphors about our existential being-in the world and for exploring consciousness as it is experienced subjectively,as it is felt"[8]. The research about the human hybridization with the virtual space has been developed in various artistic currents like the Body Art, Fluxus and the most recently NetArt. From these analyses, emerge the need for suitable communication space. Canadian artist Davies realized it with her work a new kind of spaces. "Through which our minds may float among three-dimensionally extended yet virtual forms in a paradoxical combination of the ephemerally immaterial with what is perceived and bodily felt to be real"[8]. The philosopher Gaston Bachelard in his book "The poetics of space" examines the potential of psychological transformation of "real" places like deserts, valleys and deep sea, open spaces different from urban spaces to which we

M.L. Gavrilova and C.J.K. Tan (Eds.): Trans. on Comput. Sci. XVI, LNCS 7380, pp. 179–190, 2012.

have used to: “By changing space,by leaving the space of one’s usual sensibilities, one enters into communication with a space that is psychically innovating. For we do not change place, we change our nature”[2].

Other significant examples could be found in the work of artists as Jeffrey Shaw and Petra Gemeinboek with “Uzume” and many others. Open virtual environments offer individual and collective access with different types of experience construction which stimulate emotions, introspection, reflection and consideration on surrounding space.

Virtual reality immersive systems need new definitions of concepts and space movements, with logical tests and aesthetic values defined within user's interactions. Planning of dynamic processes within virtual environment requires a particular sensibility for dialog construction between user's extended senses and his surrounding immaterial space. This metaphysical space is a metaphor of the post-modern society which is completely immersed into globalized information fluxus. Metaplasticity tries to give an opportunity of redefinition of "user-human scale" condition, becoming an independent actor of its personal political and social knowledge dimension.

2 Ars Metaplastica

Nowadays interactive media contain dialogs with their spectators that are more than simply observers, they have an acting function. The interactive media is created with two actors. The first actor that originates or defines programming rules for (user's)spectator's conditions and the second actor-spectator that introduces the progress of artwork with the goal of acting in its potentiality, differently from the traditional spectator(user) that has no possibility of interaction. The media-work is therefore, constituted of two different semiotic objects: the Actor that is the computer program and the other object, the Spectator(user) with the role of co-authoring or co-acting. The **Plasticity**, concept in opposition with Elasticity, in digital media terms is a characteristic of the user's activities that within its own interaction process can create,modify and perform every form and content of the newer virtual media. The increased plasticity of the post Web 2.0 digital media include the social dimension as another level of potentiality to extend human communicative and creative possibility for the new virtual communities. The term ***metaplasticity*** is defined within the neuroscience or in an algorithmic sense of plasticity which are different points of view from a definition that would be given in the metaplastic discipline. The ***Metaplastic discipline*** is composed with different transdisciplinary fields coming form Art,Design, Architecture, Cybernetics, Psychology, Semiotics, Artificial Intelligence and Computer Sciences. Metaplastic discipline defines proper goals **characteristics** with derivation from their originary disciplines as[17]:

-***Interdisciplinarity*** of existing relations between Art, Design, Science and Technology;

- ***Dematerialization*** of artworks and its processes from their disciplines;

-***Hybridization*** between Aesthetics and Technology(hybridization type1) becomes re-definition of sensible forms production practices;it becomes in socialization through artworks and in different levels of interaction modalities: between artwork, author and spectator(hybridization type 3) and between society,science and technology (hybridization type 2);

-***Interactivity*** as a fundamental paradigm of dynamic relations occurred among author,spectator and artwork;

- ***Synaesthetic immersivity*** of spectator through his sensorial and psychological involvement coordinated in the interactive representation;

- ***Communication of Wisdom*** as a cultural goal to be obtained through the creation of new metaplastic media.

The **Metaplastic Media**, one of discipline's objects[17], within its own aesthetic and semantic codes define a new culture of the representation. Interaction processes defined with metaplastic codes, trace behaviors and plastic multisensorial qualities.

Metaplastic languages or codes are methodologies based on abstract art languages rules applied to digital symbolic systems needed for the construction of Metaplastic media Entities. The following definitions introduce some other necessary levels of theoretical definitions:

- ***Metaplastic Entities,*** *also called* ***Metaplastic Machines,*** are complex objects of any present and future typology, which directly act and interact within the applied metaplastic metalanguage rule configurations;

- ***Metaplastic Virtual Worlds*** are digital spaces typology of Metaplastic Entities;

- ***Metaplastic Metaspace(Open Metaplastic Platform)*** is a cyberspace composed with a network of Metaplastic Virtual Worlds [17] linked between themselves with elements called Balances. Some other new symbolic elements are added to the set of every metaplastic language. They are named as: Arrow, VisualTag, Cube Switcher, Cross sign, Plus sign and Minus sign.

3 The MetaPlastic Virtual Medium

The development of a theoretical meta-model for virtual media definition is based on the metaplastic ontology. This model is a conceptual map described within three fundamental study fields: the information field for the "Construction of knowledge"; the sensorial field for "Emotional involvement" and the area of "Social Participation". Every field mentioned before, was defined with more subfields. The parameters were placed on the conceptual map axis, each for every field through a qualitative evaluation to define a digital communication media. The two external zones were defined, respectively, as the "Private space" and the "Social space", within the use of media and its cultural content. The internal circle's width, visually indicated the conditions of being part of the media included or not included in the social spaces. Defining that media property, we defined different reality conditions: virtual reality, extended reality or mixed reality.

The level that described the "emotional involvement" area, indicated which conditions had improved to create an immersive reality. The application of the methodology to the laboratory projects, defined their quality and made it visible for different categories and various relations levels. The resultant models of the virtual media, offered useful indications about the shared informations; the definition of the user's activity; the interaction modalities with the information and with other connected users. The visual language model had confirmed the importance of the aesthetic value as a meaning element and to emphasize the emotional intensity of the medium.

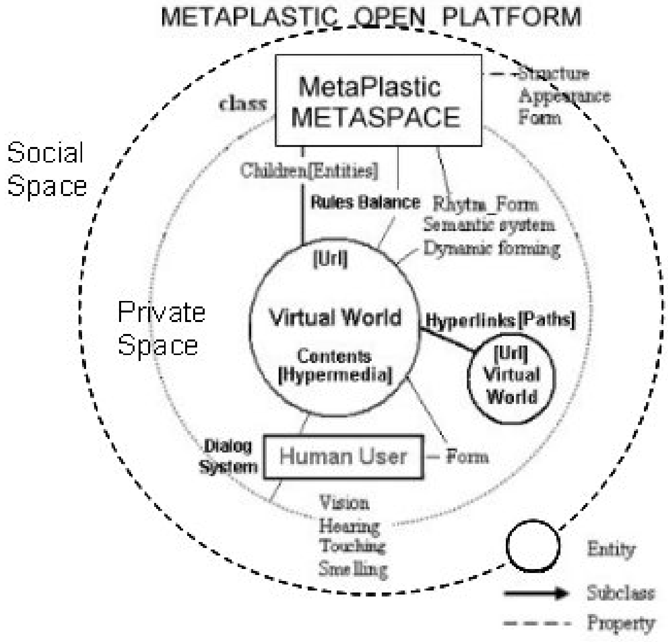


Fig. 1. MetaPlastic Ontology schema

Metaplastic Metaspace properties:

- Name: Name identification or Argument;
- Type: node typology identification;
- Url: node web address or hyperlink;
- World position (GPS or Absolute local coordinates);
- World angles rotation(GPS or Absolute local coordinates);
- Nodes List (Visual/Textual configurations description).

4 MetaSpace Structure Description

The dynamic relations between each network nodes and the user establish the metaspace meaning representation through its shapes and behaviours. The Metaspace acquires knowledge during its processes from every element. At the same time, some part of the ontological knowledge base is shared between the entities. The structure of elements are described in term of group and individual roles hierarchy. Every virtual world (node) has its own node name, type,web address(URL) and a hyperlink network(Node List) that refers to other virtual worlds (subnodes) connected to the previous one and all between each other. ”The visual language is made of related meaning and events dynamics within the metaspace. The movement is determined from the qualitative relations and quantitative elements between the environment, its own network nodes and the user. These dynamic relations establish the equilibrium of the virtual space determining the meaning representation through its shapes and spatial relations”[17]. The Metaspace includes a different kind of symbolic visual elements. Each of these elements have their specific and necessary role.They are described in the following part.

4.1 The Balance

It is a particular element of the structure that links every node to each other of the metaplastic metaspace (artwork). It rules the dynamics of the plastic space and their elements within its own regulation.There are different kind of balances for: Form, Movements and Chromatic elements. The balance function is computed between the x and y state values of the linked space elements as:

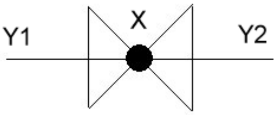


Fig. 2. The Balance

State gradient value (x,y) [0..1] scale; (1)

Balance f(x,y) = y*(gradient x(0)+gradient x(0.5)-gradient x(0/0.5)); (2)

The sum of balances establish the dynamic forming of the complex network of the metaspace [17].

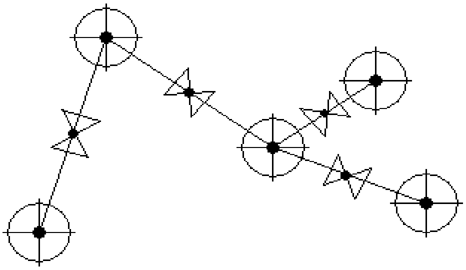


Fig. 3. The metaspace results as a complex balanced structure

4.2 Other MetaSpace Elements

The Arrow It is a sign of movement direction (hyperlink) that guides to another location as a webspace or a local information address. It is visualized as a cone with vertices that indicate the destination versus.

The Visual Tag. It is a sign of static information visualized as a square that contains static or animated interactive multimedia contents (text,image,video,web page,etc.).

This sign could be assigned with an arrow that leads to a specific web location of information.

The Cube Switcher. By Touching or Proximity this sign causes an information state change. It could be assigned to both of the previous kind of signs or directly to metaspace balances or its own properties of form,color or movement.

The Cross. It is a sign for deletion of every current selected element of the metaspace environment.

The Plus. It is a sign for adding a new element of the current selected type in the metaspace environment.

The Minus. It is a sign for subtracting the current selected element from the metaspace environment.

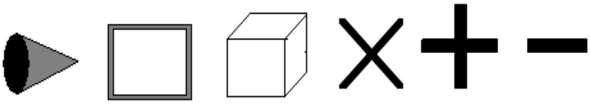


Fig. 4. Set of Metaplastic Visual Interaction Signs

5 The MetaSpace Interaction Processes

The cyberartist can interact with the Metaspace(FSM fuzzy system)and its own interactive elements through the pointing, relocation, clicking, proximity and remote touch functions available from the input system devices. The Virtual environment is driven by previously defined abstract art rule relations that transform sequences of user's sensory inputs,which produce feedback and send it as a response to the output of the system. The sensorial result is a fuzzy truth value [17] that activates the decision making process[17] of the metaplastic space. The artist during his own cyber performance, differ his emotional states with the virtual environment,which change continuosly from “Sensing” to “Feeling” states through recursive cycles of feedback. This interactive process is called Metaplastic Dialog System or Extended Senses[17]. The cyberperformer's immersive experience is produced by the Metaspace simultaneously in three ways: the absorption of the input device into the artist's body image; the integration of the screen interface into the artist's extended body boundaries; and the activation of surrounding space through multisensory and haptic feedback. The state transitions of the metaspace elements are activated generally by proximity through their threshold sensor that defines its own interactive zone with the cyber performer. In the same case, for example if there is a Cube Switcher, it can act independently from their interactive zone. The cyberperformer's five Senses defuzzified values can be used as an expression of the user's stimulation and its new emotional state caused by external environmental inputs and output system values.

The behaviors of the Metaspace environment system are in their processes on the visual rules matrix of relations among dynamic elements. The interaction within the dynamic cycles unifies sensorial states of the virtual environment,creating complex behaviors between itself and its own elements, by assigning meaning through movement codification of interactive forms in the “Red and Black”[17] semantic metaspace.

FSM $\mathrm{Act}_{i} = 1\text{-}f(\mathrm{interp}(S_i(w)\mathrm{vision}, S_i(w)\mathrm{hearing}, S_i(w)\mathrm{touch}, S_i(w)\mathrm{body\ mov.}))_t$; (3)

Metaspace RhytmForm= $\mathbf{\Sigma}_{i}(f\mathrm{Act}_i \,{*}f(\mathrm{interp}(\mathrm{Entity}_k\ \mathrm{state}_i, \mathrm{Entity}_{k+1}\ \mathrm{state}_i))_{t}$; (4)

where: Act = Interaction state graduation; S = Senses function; w = Emotional state weight [0..1]; i = dialogue state; t = action time; interp=interpolation function;
Entity state = Metaspace Entity state.

Rule 1: if sign value - and intensity -100 then w = 0.0
[Relaxed emotional state];

Rule 2: if sign value +/- and intensity 0 then w = 0.20
[Mildly interested emotional state];

Rule 3: if sign value + and intensity 50 then w = 0.50
[Interested emotional state];

Rule 4: if sign value + and intensity 100 then w = 0.75
[Involved emotional state];

Rule 5: if sign value + and intensity 150 then w = 0.85
[High involved emotional state].

The resulting w value from the previous function is used to obtain final Act state graduation for different output channels interaction processes results(vision,body, hearing,touch).

6 Human-Interaction with Real and Virtual Spaces

The Metaplastic Open Metaspace interactive media is a modality that unifies the user's interaction processes within mixed realities spaces. The Multi-plasticity of multi-sensorial and multi-directional communications properties are driven by their own dialog system. The system includes the whole body interaction through the production of multisensorial feedback within both the metaplastic virtual spaces and physical interactive devices. The human haptic interaction in the shared virtual space produces the sensation of tele-presence within the system while the mediated physical feedback with the objects enforce the feeling of extended body senses.

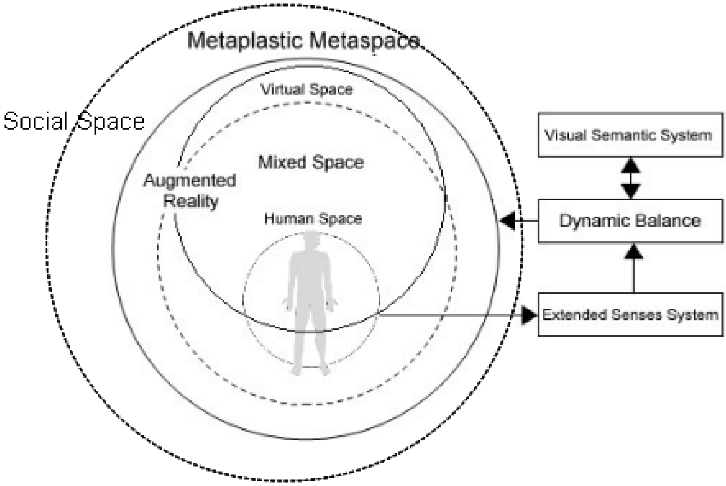


Fig. 5. The Metaspace different realities schema

7 As a Living Art System

The same methodology described above can be useful in applying it to the metaplastic virtual worlds as a living artworks system. Fluid immersive multidimensional space is,in fact, a landscape of knowledge where the user could make his abstract surrounding experience. The interactor or virtual performer redefines his role within its abstract space immersion. This enables him to interact with the environment and make new forms of data creations,also with other connected users. The user in his relation with the synthetic environment realizes different level of participation/inclusion within the system, which changes his role from spectator to interactor and at the end “immersant”,defined with Char Davies's words in her installation “Osmose”[8]. The volume forms develop into a system maps and have a route orientation function.The orientation in the system is possible through signs which indicate to users the possible options. The remote presence of the user's body and his extended senses are indicated within the Klee's man as a virtual pointer of analogue/opposite field polarity in the immaterial-materiality of sound,light,form and colour that produce various space aesthetic effects. The user in these new metaplastic space could explore and interact with the artwork and contribute to change,delete or recreate it in different conceptual landscapes. User's results give important information about the conceptual expression of his artwork and the resulting knowledge map and their dynamics within the virtual environments. Interactions in performance produce aesthetic effects of the virtual environment with transforming the entire system into a new artwork.

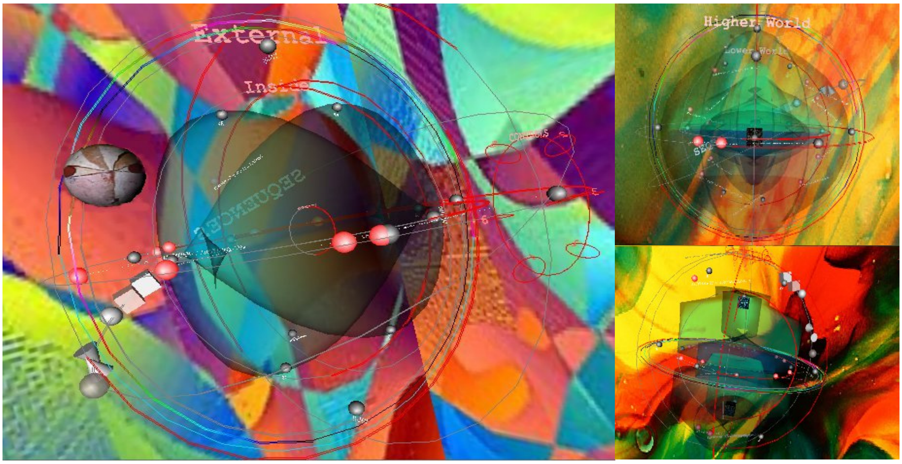


Fig. 6. The Living CyberArt System in action

In this example the picture background is from Paul Klee painting.

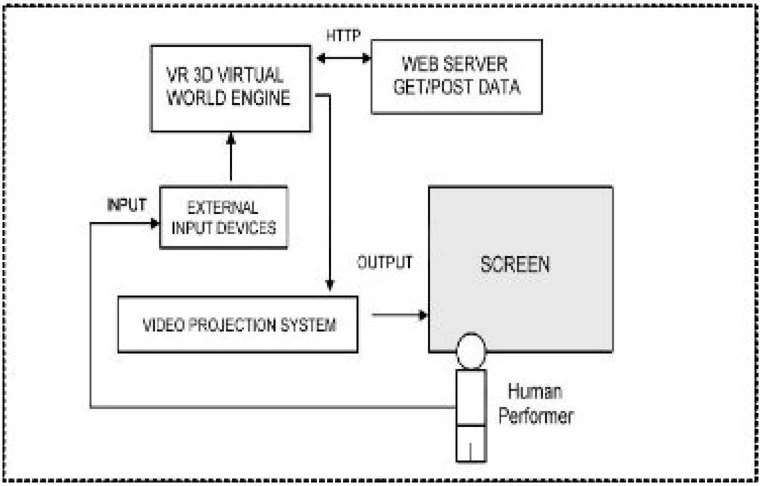


Fig. 7. The virtual performance system schema

8 The MetaPlastic Platform System

The Metaplastic Open Platform system[1] is used as digital environments with different usage, for example: Web 3d, interactive games, virtual museums, digital archives, interface systems, mobile applications, virtual art platform and installations. It is an open source software development environment for creating and deploying deeply collaborative virtual worlds on multiple operating systems and devices. It is, in fact, a multi-platform system that is available for Windows and Java locally, online and mobile uses. The Metaplastic Platform shares its virtual worlds in different typologies for any kind of application through its own SDK. The complete system is available in different configurations for Personal Computers, PDAs, Mobile Phones and other mobile devices. The Platform is shared with the public in an open format in order to use all of its own features.

It is easy to configure with its own 3d interface functions or within simple XML configuration files. Some user's interaction choices or environment settings are possible also to modify with textual scripts during the software running. The user can interact with the system with different external devices as mouse,keyboard,gloves, touch-screens, GPR satellite world localization functions and environmental sensors. The metaplastic system enhance the performance and offers a multi-server architecture with containing the network and every other proper entity's state and characteristics. Entities send their messages to the server regarding their states evaluations.The server computes every network interaction state and decides what message should be sent and to which entity. This process continues from and to every connected server in the network.The network system cooperates online for simultaneous and distributed execution of each of its own metaspace_node between the client and the server through its own software applications. The software engine is a 3d graphic environment based on a specific graphic library that guides the system capabilities to draw the virtual space contents in real-time mode. The scene control uses different low-level functions of the system to properly combine virtual space elements with the user inputs. The Metaplastic Virtual Platform system is suitable for the main operating systems(Windows,Linux and Java) as a stand-alone executable or embedded on web browsers or within mobile devices in single or multi-users interactive environments. It runs with the following step process:

-A. The scene control begins with the user sensorial input analyses through the proximity function of the fuzzy dialog control system;
-B. The Fuzzy state machine (FSM) makes the evaluations, previously defined from membership sets and fuzzy rules;
-C. The Balance function makes its evaluation with previously calculated values.The Balance results activate all interaction processes and behaviours of the entire system;
-D.The resulting values compose the semantic codification of virtual space;
-E. This step provides the 3d calculations to build up the virtual space model;
-F. The virtual space scene is visualized with the system drawing functions.

Within this phase, the system communicates with the web server to update the system states and gives the resulting feedback to the Fuzzy sensorial decision-making process (A phase). Finally, all the output results are combined with the user inputs into a new interaction cycle.

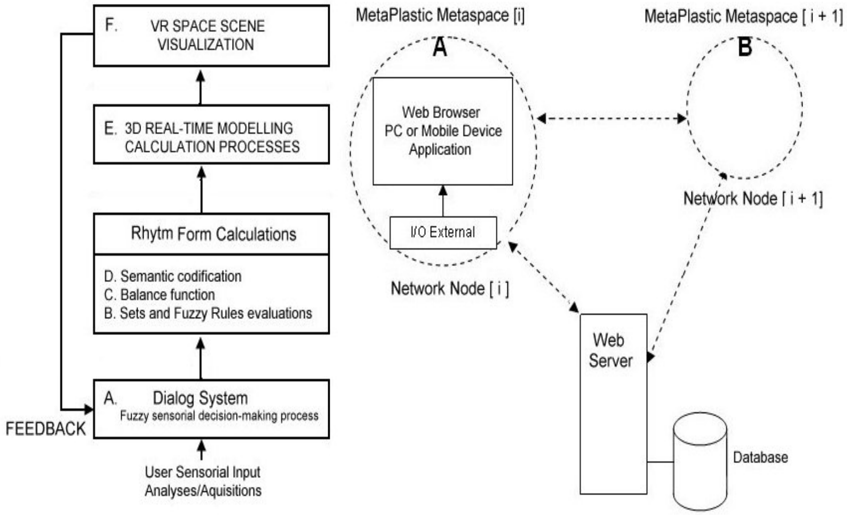

F. VR SPACE SCENE VISUALIZATION
E. 3D REAL-TIME MODELLING CALCULATION PROCESSES
Rhytm Form Calculations
D. Semantic codification
C. Balance function
B. Sets and Fuzzy Rules evaluations
A. Dialog System
Fuzzy sensorial decision-making process
FEEDBACK
User Sensorial Input Analyses/Aquisitions
MetaPlastic Metaspace [i]
A
Web Browser PC or Mobile Device Application
I/O External
Network Node [i]
MetaPlastic Metaspace [i + 1]
B
Network Node [i + 1]
Web Server
Database

Fig. 8. Metaspace real-time rendering and Network System

9 Conclusion and Future Works

In conclusion, the virtual semantic system of the metaplastic virtual worlds opens a possibility for a profound analysis of numerous applications that emerge into social communities and networks on the Web. The metaplastic vision already contains its Klee's original sense and social ethics indicated by Stendhal's homonymous literature masterpiece. The metaplastic virtual world is proposed as a conceptual tool of Contemporary society and indicate the opportunity for social content evaluation on the Net by underlining problems and contradictions through the Red and Black semantics.Within actual globalized panorama the tensions of sustainability, economy and development, is where this semiotic model offers a new possibility of social topics representation and explanation from virtuality to reality.

The practical applications of this theory, its evolutions and new advances,include an open digital platform for different usage(web3d environments, interactive games, virtual museums, digital archives, interface systems, mobile applications, virtual art platform and installations).

References

1. Artsmachine, The Virtual Metaplasticity platform project, `http://www.artsmachine.com`
2. Bachelard, G.: The poetics of space, p. 203. Beacon Press, Boston (1966)
3. Barfield, W., Zelter, D., Sheridan, T.B., Slater, M.: Presence and Performance within Virtual Environments. In: Barfield, W., Furness, T.A. (eds.) Virtual Environments and Advanced Interface Design. Oxford University Press, Oxford (1995)

4. Benedict, M. (ed.): Cyberspace First Step. MIT Press, Cambridge (1991)
5. Bru, C.: L'estetique de l'abstraction. Presses Universitaries de France, Paris (1955)
6. Chomsky, N.: Three models for the description of language in Information and Control, on certain formal properties of grammars. IRE Transactions on Information Theory 2 (1959)
7. Coates, G.: A virtual Show. A multimedia performance work, San Francisco, USA (1992)
8. Davies, C.: Multimedia: from Wagner to Virtual Reality, pp. 293–300. W.W. Norton & Company, New York (2001)
9. Eco, U., Munari, B.: Arte Programmata (Olivetti catalogue), Milano, Italy, Olivetti (1962)
10. Fink, C.A.: Searching for the most powerful behavioral theory: the whole of the Behavior Systems Research Institute and the Behavioral Model Analysis Center. Fink, Falls Church (1979)
11. Goodman, N.: Languages of Art. An approach to a Theory of Symbols. Bobbs-Merril, Indianapolis (1988)
12. Grau, O.: Virtual Art. MIT Press, Cambridge (2003)
13. Greimas, A.J., Courtés, J.: Semiotics and Language: An Analytical Dictionary. Indiana University Press, Bloomington (1982)
14. Holtzman, S.R.: Digital Mantras. MIT Press, Cambridge (1995)
15. Laurel, B.: The Art of Human-Computer Interface Design. Addison-Wesley, Reading (1990)
16. Moles, A.A.: Art et Ordinateur. Blusson, Paris (1990)
17. Mura, G.: MetaPlasticity in Virtual Worlds:Aesthetics and Semantics Concepts. IGI-Global, USA (2010)
18. Mura, G.: Multiplasticity of new media in Multiple Sensorial Media Advances and Applications: new development in MulSe-Media. In: Ghinea, G., Gulliver, S.R., Andres, F. (eds.) IGI-Global, USA (2011)
19. Mura, G.: The Metaplastic Constructor in CAC 2 Computer Art. Europia, Paris (2008)
20. Perrot, X., Mura, G.: Workshop Virtuality in Arts and Design: Virtual exhibition projects, Archives and Museum Informatics (2005), http://www.archimuse.com
21. Thürlemann, F.: Paul Klee. Analyse sémiotique de trois peintures. L'Age d'Homme, Lausanne (1982)
22. Wiener, N.: The Human Use of Human Beings: Cybernetics and Society. Da Capo Press, Cambridge (1988)
23. Zadeh, L.: Biological application of the theory of fuzzy sets and systems. In: The Proceedings of an International Symposium on BioCybernetics of the Central Nervous System, pp. 199–206. Little Brown, Boston (1969)

Author Index

Afanasiev, Valery 17
Aleshin, Vladimir 17

Bobkov, Alexander 17

Cho, Gyuchoon 77

Dutoit, Thierry 34

Gavrilova, Marina L. 77

Harrison, Erika 134

Klimenko, Stanislav 17
Kodama, Toshio 115
Kuliev, Vitaly 17
Kunii, Tosiyasu L. 95, 115

Lees, Michael 55
Li, Yichen 149

Mahdavi-Amiri, Ali 134
Mura, Gianluca 179

Nahavandi, Saeid 1
Najdovski, Zoran 1
Novgorodtsev, Dmitry 17

Ohmori, Kenji 95

Pang, Mingyong 149

Rosenthal, Richard 77

Samavati, Faramarz 134
Seki, Yoichi 115
Sourin, Alexei 1

Tilmanne, Joëlle 34

Viswanathan, Vaisagh 55

Wei, Lei 1
Wei, Mingqiang 149
Wenhua, Chen 163
Wenzhen, Yang 163
Wu, Jianhuang 149

Yampolskiy, Roman V. 77

Zhigen, Pan 163