Center for Human Modeling and Simulation
Quarterly Progress Report
No. 53

Norman I. Badler
Director, HMS
Department of Computer and Information Science
University of Pennsylvania
Philadelphia, PA 19104-6389
Third Quarter 1994

November 21, 1994

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	November 1994	

4. TITLE AND SUBTITLE
CENTER FOR HUMAN MODELING AND SIMULATION QUARTERLY PROGRESS REPORT NO. 53

5. FUNDING NUMBERS

6. AUTHOR(S)
Dr. Norman I. Badler

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
University of Pennsylvania
Computer & Information Science Department
Philadelphia, PA 19104-6389

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U.S. Army Research Office
P.O. Box 12211
Research Triangle Park, NC 27709-2211

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
The views, opinions and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy, or decision, unless so designated by other documentation.

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

This Quarterly Report includes descriptions of various projects underway at the Center for Human Modeling and Simulation during July through September.

14. SUBJECT TERMS
Human Modeling and Simulation

15. NUMBER OF PAGES
36

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRA
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

MEMORANDUM OF TRANSMITTAL

U.S. Army Research Office
ATTN: AMXRO-RT-IPL (Hall)
P.O. Box 12211
Research Triangle Park, NC 27709-2211

____ Reprint (15 copies) XX Technical Report (40 copies)

____ Manuscript (1 copy) ____ Final Report (40 copies)

____ Thesis (1 copy)

____ MS ____ PhD ____ Other________

CONTRACT/GRANT NUMBER DAAL03-89-C-0031

TITLE: Center for Human Modeling and Simulation Quarterly Progress Report No. 53

is forwarded for your information.

SUBMITTED FOR PUBLICATION TO (applicable only if report is manuscript):

Sincerely,

Dr. Norman I. Badler
Professor, CIS and Director, HMS
University of Pennsylvania
Computer & Information Science Department
Philadelphia, PA 19104-6389

Contents

1 Introduction: Norman I. Badler

With this issue we would like to introduce two of our newest staff members in the Center for Human Modeling and Simulation. Karen Carter joined us in July as our Associate Director. She assumed the position previously held by Dawn Becket. (Dawn entered the Wharton MBA program at the University of Pennsylvania.) Karen has many years experience in the Computer and Information Science Department, being the Administrative Assistant and Office Manager for CIS while Norm Badler was the Chair of CIS. Our other new staff member is Pei-Hwa Ho, who should be familiar to our readers through his frequent reports on human body modeling in this publication. Pei-Hwa will be the *Jack* customer service representative. He will be the technical contact for *Jack* users and maintain a "Frequently Asked Questions" list. Over the next few months, information requests should be streamlined through a *Jack* users electronic newsgroup and a FAQ database.

This Quarterly Report includes descriptions of various projects underway in the Center for Human Modeling and Simulation during July through September 1994.

These reports include:

- Motion System packaged as a library for tighter integration with TTES.
- Design and implementation of a proof-of-concept demo for the ARPA MediSim project.
- SASS improvements.
- Improvements to human body scaling in *Jack.*
- Extensions of pipeline rendering.
- Motion planing algorithm for human reaching motions.
- Implementation of a forward dynamics algorithm for articulated figures.
- "Duck" sensor completed.
- Progress in the animation of fluid phenomena.
- Qualitative models of respiratory dynamics.
- Progress on the modeling of the respiratory system.
- 2-D lung model.
- Discussion of efficient techniques for hierarchical radiosity.
- Investigation of a distributed multi-level radiosity solution for complex environments.

There are also three appendices:

- *A Review of Human, Robotic, and Simulated Grasping Literature:* Brett J. Douville. This is part of a report produced for the AirForce DEPTH project.

- *Sight and Sound: Generating Facial Expressions and Spoken Intonation From Context:* Catherine Pelachaud and Scott Prevost. This paper appeared in the proceedings of the 1994 ESCA/IEEE workshop on speech synthesis in New Paltz, NY.
- *Automatically Generating Conversational Behaviors in Animated Agents:* Justine Cassell, Catherine Pelachaud, Norman Badler, Mark Steedman. This abstract was invited for presentation at a Microsoft-sponsored workshop on Animating Lifelike Agents, 1994.

This research is partially supported by ARO DAAL03-89-C-0031 including U.S. Army Research Laboratory and Natick Laboratory; ARPA AASERT DAAH04-94-G-0362; DMSO DAAH04-94-G-0402; ARPA DAMD17-94-J-4486; U.S. Air Force DEPTH through Hughes Missile Systems F33615-91-C-0001; Naval Training Systems Center N61339-93-M-0843; Sandia Labs AG-6076; NASA KSC NAG10-0122; MOCO, Inc.; National Library of Medicine N01LM-43551; DMSO through the University of Iowa; and NSF IRI91-17110, CISE CDA88-22719.

2 Humans in Distributed Interactive Simulation: John Granieri

I presented our work to date on humans in Distributed Interactive Simulation (DIS) at the 11th DIS Workshop in Orlando, FL. The slides should be published in the DIS proceedings.

I packaged the motion system as a library, for a tighter integration with TTES (and also NPSNET for the AUSA project). More documentation will be forthcoming on the internal structure and API for the motion library, as its implementation settles down.

We submitted a paper to VRAIS '95, entitled *"Off-line Production and Real-time Playback of Human Figure Motion for 3D Virtual Environments"*, which describes the motion system we used for TTES.

We also submitted an abstract for a paper for the SIGGRAPH Symposium on Interactive 3D Graphics, which links the real-time motion generation system with Barry Reich's sensors and Becket's PaT-Nets, to begin building the framework for a behavioral programming regime for real-time agents.

2.1 *Jack* Happenings

Mike Hollick has taken over the main role for putting together the *Jack* 5.9 release. I am focused primarily now on putting together the framework for our *Jack* 6 system. Many features of *Jack* 6 are designed to support current research projects in the Center. It will be the research environment for the next few years. As we finalize some design decisions this quarter, I should have a high level features list for the next report. My overall objective is to leverage off high-quality work done elsewhere, so that we can focus and maximize our efforts in the specific areas of human modeling and simulation.

Graphics: Currently, *Jack* uses the Peabody run-time database and Psurf geometry and drawing utilities (which in turn are based on IrisGL) to create visuals. We've chosen IRIS Performer as the underlying graphics system, as it provides several key features that we do not wish to re-invent ourselves. These are: (1) hierarchical run-time database, with functions for intersections, (2) software rendering pipeline for using multi-processor systems to good effect, (3) level-of-detail management, (4) high-speed graphics library, optimized for each type of SGI machine, (5) database loaders for many different geometry formats.

Extension language: Currently there are several languages used in *Jack*: Peabody, JCL, LISP, and Psurf. We will consolidate these so Peabody and JCL both use a LISP-like syntax, and I'm currently looking into what will be the most suitable extension language for *Jack* and the related tools (i.e. SASS). Current code written to the XLISP API will be upward compatible to the new language (if you don't make use of the XLISP object system).

User interface: *Jack* currently uses a GL-based minimal interface. As mentioned before, we will use a Tk/Tcl-based user interface, for the 2D widgets and components. We can leverage off a lot of User Interface components already built under Tk.

2.2 Next Quarter

I'm working on extending the run-time motion system for the DMSO project, and integrating it into the *Jack* 6 framework. *Jack* 6 will be our research vehicle for this project.

3 MediSim Demo: Mike Hollick

Most of this quarter was spent designing and implementing a proof-of-concept demo for the ARPA MediSim project. This involved integrating *Jack* models and animations into a Performer based renderer (NPSNet - Naval Postgraduate School), and showing basic medical care being performed by soldiers in a DIS environment. The system was completed and demonstrated at the AUSA Conference in Washington, DC in October. A full description will be included in the next Quarterly Report.

4 SASS: Francisco Azuola

- SASS v.2.3

  SASS is now running under IRIX 5.0. Some minor changes were made to improve the screen drawing routines. Also, hand data was included into SASS to support the 1988 Anthropometric Survey: Hand data (Army Natick Tech Report TR-92-011, by Thomas Greiner).

- XSASS

  XSASS project is temporarily on hold. I have concluded, upon examination of what is currently available, that, even though having an X version of SASS is necessary for portability purposes, some redesign is necessary. The problem is not only a cosmetic one, but also deals with SASS's functionality.

- SASS v2.5

  Before an X version of SASS appears, a GL version 2.5 will be released. This version will address scaling issues, particularly, of the hand and upper limbs. I have already cleaned the geometry interface between *Jack* and SASS, in response to users' feedback. This results in a much more accurate scaling, and allows for better global appearance.

  This version should also allow Viewpoint Datalabs body scaling. The contour body will not be supported any longer, even though it will be included in the release package.

- Rule System

  The design and implementation of a new rule system for SASS was partially completed. One of the major drawbacks of SASS is the impossibility to include and/or modify rules. This rule system works on top of a object-oriented database, and should allow for user defined rules, such as stature constraints, limb length constraints, etc. Work needs still to be done in rule system integrity and incorporating the system into SASS.

5 Human Body Scaling and Shape Control: Pei-Hwa Ho

This section summaries the current techniques used in scaling human bodies, the underlying assumptions, and limitations. It hopes to address the concerns of end-users and to give them a better understanding of the fundamentals in human body scaling which is not clearly visible when using *Jack*.

5.1 The Reference Frame Problem

Before any geometry can be scaled a reference frame must be established, a process we called normalization. One of the methods is the bounding box approach where a geometry is put into a box with the origin at the center and the three axes lie parallel to the three sides of the box, the box is then scaled down to a unit box. Scaling is done by simply stretching the dimensions of the box to the desired values with the object in it. This is what we call a linear scaling. It works well with symmetric objects like cylinders and ellipsoids but causes distortions when objects are not symmetric.

There are two type of distortions introduced by linear scaling, one is shape distortion, e.g., a rectangle can be scaled into a diamond shape because of a slightly rotated frame, and the other one is joint axis distortion caused by the misalignment of the line connecting joint centers with the frame's axes.

We have developed different types of normalization to accommodate different types of segment geometry. These normalization techniques were reported in Quarterly Report #52.

5.2 Scaling

Many types of scaling can be applied to an object but within the context of human body modeling they can be categorized as follows:

- Constant scaling, where the same proportion is applied to all three dimensions of an object, e.g. scale a unit cube to a cube of size two by two by two. Only one number is needed to specify the intended scaling.
- Linear scaling, where each dimension of the object is subject to the same scale factor but different scale factors are applied to different dimensions, e.g. a unit cube is scaled to a block of size two by three by four. Three numbers are needed for this type of scaling.
- Conic scaling, where each segment is treated like a cone with different sized elliptical cross sections at the two ends. To scale this to a different sized cone two set of numbers are needed to specify the size of the desired ellipses at the two ends and linear interpolation is used for any point in between.
- Non-Uniform scaling, where scaling is not constant throughout each dimension. We choose to use sinusoidal basis functions for their smoothness, along with positional constraints to direct the amount of scaling to each point on the object. The gives us a powerful tool to control the shapes of objects and to relate the change to the underlying physiological characteristics of each object.

Of the four scalings mentioned above all except constant scaling are used in the human model construction process. Constant scaling is considered a special case of linear scaling. All scalings change the shape of the geometry being scaled, but to various degrees.

5.3 Segment Normalization

Depending on its skeleton, a segment can be in one of three categories for normalization considerations:

- Longitudinal: the skeleton lies along the long axis of the segment. The arms, legs, neck, fingers and toes belong to this category.
- Surround: the skeleton encloses the segment. The head belongs to this category.
- Irregular: the upper and lower torso, the foot, and palm of the hand belong to this category.

For the longitudinal segments the reference frame should be such that the skeleton lies along the long axis (we choose to be z) since all shape changes are centered around the long bone(s). This can be approximated by using the line that connects the two joint centers at the ends of the segment as the z-axis with the origin assigned to the proximal end, though joint centers do not always lie on the long axis (e.g. the hip joint) and dedicated sites can be established for normalization purposes. Landmarks on the segments, when available, can also be used to locate the proper long axis. The x and y axes can be set up so that they point to the front and side of the segment. The long axis may or may not lie in the center of the geometry, depending on the shape of the segment, and thus no symmetry is assumed. All scaling can be done in this reference frame which is a more natural way of modeling growth (or shrinkage) of the segments. For the surround and irregular segments the growth tends to be uniform or planar and in these situations a reference plane instead of an axis needs to be established and dedicated sites can be used to established such a plane.

5.4 Segment Scaling

We describe the types of scalings we developed so far. They are intended to serve the wide variety of segments and their shape transition characteristics.

Realistic scaling should be designed with the following criteria:

- Scaling should not be done uniformly across all segment types all the time, rather, it should consider the structure of each individual segment and how mass is distributed in that segment under specific circumstances.
- Scaling should be done under the constraint of measurable physiological attributes to measure its effect.
- The overall shape and appearance of the model should remain consistent and natural.

To meet those criteria mechanisms must be provided to handle:

- Determination of the physiological attributes of the model.
- Correctly normalizing model geometries.
- Scaling of individual segments under physiological constraints.
- Integration of segments into a realistic model that satisfies their respective physiological constraints.

Given a segment with its specification, we want to be able to scale it to meet a different specification, provided the specification is computable. Computable in this context means that the specification can be meaningfully computed in the given geometry.

As was mentioned earlier, linear scaling assumes that a segment is symmetric about the x and y axes and that it treats every geometry the same without considering its specific shape characteristics. We developed non-uniform segment scaling to remedy those drawbacks.

5.5 Non-Uniform Scaling

Before meaningful scaling can be applied a segment must be normalized first to establish a reference frame. Since we are interested in generating figures of various sizes based on a limited selection of geometries the scaling is focused on the transformation of a geometry with known characteristics into another one with a different set of characteristics.

Non-uniform scaling focuses on mass distribution patterns within each segment in order to generate a visually convincing model. Our research focus is not on growth simulation or human morphology but understanding how segments undergo shape changes is necessary to the design of reasonable scaling schemes.

The human body structure is supported by the skeletal and muscular systems, covered by fat and skin layers. The shape of the body is determined mostly by the amount of muscle and fat contained within each segment. Factors that affect the skeleton, the amount of muscle or fat in turn affect the size and shape of the body.

Shape changes in a segment can be separated into two components, changes in skeleton and changes in soft tissues. Skeleton changes are uniform and linear, the overall shape stays the same even though its length, diameter or density may change dramatically. Soft tissue, like muscle and fat layers, undergo non-uniform changes depending on the factors causing the change (exercise, malnutrition, puberty, etc.).

We want to design non-uniform scaling mechanisms that, when combined with linear scaling, can control the shape changes of a segment that are both natural and compatible with neighboring segments. The non-uniform scaling schemes we developed are functions of locations in a segment. We assume that all segments are aligned so that z is the long axis, x points to the front and y to the side in a coordinate frame. The origin of the segment frame is where the proximal site is. The bottom portion of a segment is near the proximal end, the top is near the distal end, and the middle is between the two. The front section of a segment is the positive portion of the x axis, the rear the negative x axis. The right section is the positive y axis; the left, the negative y.

The major consideration for designing non-uniform scaling schemes is the way mass, is distributed in human body segments. We want to introduce three types of non-uniform scaling-axial, planar, and skin scaling-to give us the ability to model the range of possible shape changes. Axial scaling is a function whose value depends on the distance along the long axis of the segment; planar scaling is a function of the distance along a coordinate plane (xy, yz, or zx). Besides the scaling functions, we can also limit scaling to certain regions of a segment (top, middle, bottom, front, rear, left, or right). For limbs the distribution is obviously axial: soft tissues are attached to the central supporting skeleton. For other segments, like the upper torso, the distribution function could focus on the front or back plane of the segment. There are also instances where the distribution has both an axial and a planar effect. Skin scaling is designed to mimic the effect of increase or decrease in skinfold thickness of a segment and is a special case of axial scaling.

For axial and planar scaling we will use sinusoidal basis functions to guide the scaling process, the peak of the function will be near the specified location (top, middle, etc.) of the segment. This takes advantage of the sinusoidal basis functions for their continuity, smoothness, and zero derivatives at the peaks and valleys to ensure smoothness in the resulting geometries. The height of the profile will be determined by the scale factors. The scaling will be regional, thus it will be a function defined along an axis or a plane. The basis functions can be combined to represent various shapes with just a few terms, a wonderful quality appreciated by those who use the Fourier series.

5.6 Scaling Profile Specification

With the scaling functions defined in the previous section we can now specify how a segment is to be scaled, assuming that a proper coordinate frame is already established. The specification, a scaling profile, can be divided into the following parts:

- **Type**. Axial, Planar, or Skin.
- **Axis**. The x, y, or z axis. In planar scaling the axis represents the norm of the plane, thus z axis means the xy plane.
- **Mode**. This specifies which portion of the segment is to be affected along the long axis.
- **Region**. This specifies to which region (front, rear, left, right, or all) to apply the scaling function. Regions are defined relative to the long axis.
- **Starting Position**. This specifies the starting position of the sinusoidal profile, normalized between zero and one.
- **Ending Position**. This specifies the normalized ending position.
- **Weight**. How much of the segment's non-uniform scaling is to be done in this manner. Between zero and one.

Any segment can be scaled by first defining the desired scaling profile and then entering a scale factor, without having to consider its compatibility with its neighbors. This is useful when we want to change the appearance of a geometry without considering anthropometric constraints. The user bears the responsibility of the integrity of the resulting segment and model.

5.7 Multi-Segment Scaling

There are times that a scaling profile is applicable to not one, but two or more segments like the seventeen-segment torso. Under such circumstances we want to be able to scale this chain of segments together. This is quite similar to the single segment scaling except that the stack of segments are scaled as a whole.

5.8 Effects of Segment Scaling

The purpose of using non-uniform scaling for segment scaling is to transform a geometry to fit a different set of specifications. All specifications are expressed in terms of computable anthropometric parameters. Some of the parameters affected by the scaling can be computed without actually performing the scaling, e.g. the length of the segment, while others can only be known after the scaling is done, e.g. the volume of the segment when non-uniform scaling is dictated. Since we want to create human models based on those numerical specifications, the parameters that we choose will affect the computational framework used in building such models. When all parameters are predictable without performing any scaling the model creation process need only scale each segment once, after deciding how each segment is to be scaled. When some of the parameters cannot be accurately predicted the model creation process may need to perform multiple rounds of scalings to converge onto the desired values. The effects of segment scaling on those parameters is discussed in this section.

The parameters that we consider are: length, thickness, width, circumference and volume. The current definition of thickness, width, and length of a segment are obtained from the dimensions of the bounding box enclosing the segment. Thickness and width are half of the respective dimensions of the bounding box's due to the symmetry assumption.

The relationship between circumference, thickness, and width is more complicated. For example, the circumference of an ellipse defined by:

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1,$$

where a and b are thickness and width (or the long and short axis) respectively and $a > b$, can be found by finding the solution to the elliptic integral:

$$a \int_0^{2\pi} \sqrt{1 - \frac{a^2 - b^2}{a^2} \sin^2 \theta} d\theta.$$

Elliptic integrals do not have analytic solutions and look-up tables are often used. Human body segments do not usually have elliptical cross sections but the non-linear relationship still holds. Parameters that are determined by more than one-dimensional factors are all non-linear due to the irregular shape of the body segment.

With linear scaling, length, thickness and width are affected linearly by the scaling factors applied. The changes in volume can be computed by the product of the scaling factors. Circumference, from the above discussion, cannot be accurately predicted before scaling is applied, and has to be recomputed after scaling. With non-uniform scaling, all parameters affected cannot be accurately predicted before scaling.

For those non-predictable parameters we can precompute them with incremental scale factors, e.g. 0.9, 1.0, 1.1, 1.2, etc., then interpolate the scale factors to obtain a good starting point in actual scaling, and thus make numerical convergence faster. Consistencies between segments can be maintained by requiring compatible scalings to neighboring segments without knowing the exact effect on the parameters.

The next Quarterly Report will discuss our approaches to the overall smoothness of the models built with non-uniform scaling techniques.

6 Pipeline Rendering and System Issues: Paul Diefenbach

My work over this period has focused on extending the concepts of Pipeline Rendering, a method described in Quarterly Report #50. I presented a technical sketch at SIGGRAPH '94 on this work and have received inquiries from Silicon Graphics. Extensions on this work include adding facilities for rendering correct transparency.

I also worked on advancing the geometric translation tools available for *Jack*. This includes working with Arthur Pro to add trimmed parametric surfaces in the IGES translator and making modifications to the IGDS translator.

A dual-Pentium system was received on loan from Intergraph and the feasibility issues involved in porting *Jack* to this platform were investigated. Such a port would include creation of an OpenGL version using Tk/Tcl as the user interface.

7 Planning Human Reaching Motions: Xinmin Zhao

We have developed and implemented a motion planning algorithm for human reaching motions. It is based on the randomized algorithm described in [1]. The algorithm currently controls 9 degrees of freedom: 1 at the elbow joint, 3 at the shoulder joint, 3 at the waist joint, and 2 at the foot (X and Z translations).

Input to the algorithm is the goal position of the palm center. Output from the algorithm is a sequence of joint angles that move the palm to the goal position. Performance: Tasks such as that shown in the Figure above take about tens of seconds to a few minutes to compute on an SGI Indy workstation.

References

[1] Barraquand, J., and Latombe, J.-C. Robot motion planning: A distributed representation approach. *International Journal of Robotics Research 10*, 6 (December 1991), 628–649.

8 Implementaion of a Forward Dynamics Algorithm: Evangelos Kokkevis

We have been working on the implementation of a forward dynamics algorithm for articulated figures. The goal was to find a fast method to simulate dynamically correct motion of arbitrary rigid body linkages.

We chose to implement Armstrong and Green's [1] algorithm because of its computational efficiency. The recursive algorithm used has O(n) performance, n being the number of links in the

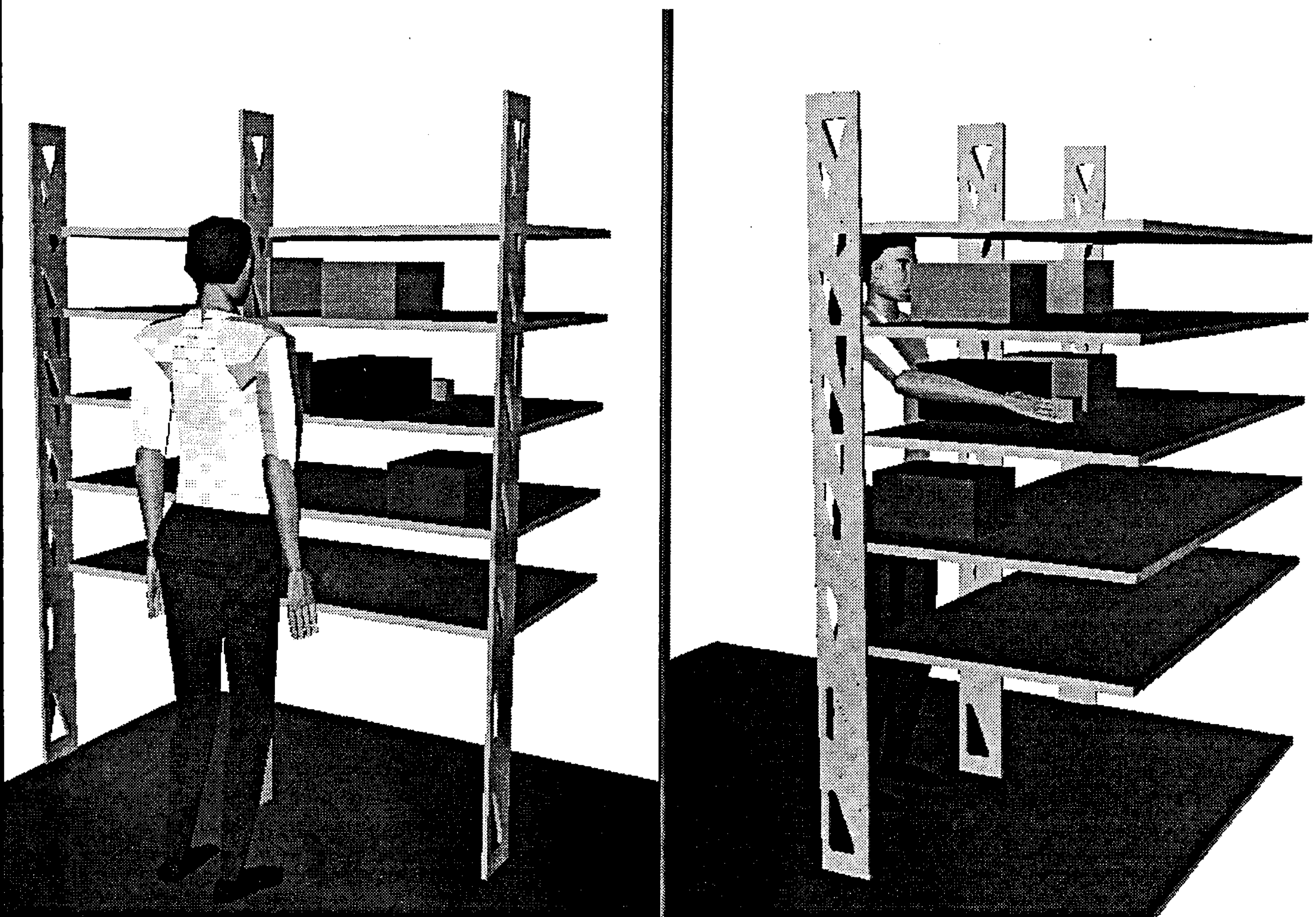

Figure 1: *Jack* reaches for a spare part in the equipment rack

system simulated. Non-recursive methods generally have $0(n^2)$ or higher complexity.

The next problem solved was that of handling collisions between two articulated figures or between a figure and its surroundings (such as collisions with the ground or walls). The effect of a collision is simulated in two phases: The first phase is at the instance when the colliding objects first come in contact. The instantaneous change in their velocities is calculated by solving a linear system of momentum balance equations [2]. In the second phase, if the two objects do not separate after the impact, the contact force between them is computed. The contact force is such that the contact surfaces do not penetrate each other.

The elasticity coefficient allows us to simulate all the range of impacts, from perfectly inelastic to perfectly elastic. Moreover, the roughness properties of the colliding surfaces is modeled through the friction coefficient which can be set arbitrarily.

We are currently working on the incorporation of the above routines into *Jack* so that dynamic animations can be created interactively.

References

[1] William Armstrong and Mark Green. The dynamics of articulated rigid bodies for purposes of animation, The Visual Computer(1985),1:231-240.

[2] Matthew Moore and Jane Wilhelms. Collision Detection and Response for Computer Animation, Computer Graphics, Volume 22, Number 4, August 1988.

9 Sensor-Based Navigation: Barry D. Reich

This quarter I completed the "duck" sensor. A duck sensor is used to detect objects which, if the agent were to pass under them, would require the agent to duck. I also wrote a PaT-Net which uses the duck sensor to monitor the environment during a simulated terrain navigation and duck the agent when necessary. Unlike the terrain, depth, and hostile field-of-view sensors, the output of the duck sensor is not used to contribute to guiding the agent. Instead it is used to increase the realism of the navigation.

This quarter I also completed a PaT-Net-controlled simulation where birds fly to and land on a wire. Once there, each bird shuffles left and right in order to maintain an approximately equal distance to the birds on either side. Birds have been observed exhibiting this behavior.

I am currently working on planning project with Chris Geib and Mike Moore. We are using *Jack* to create an architecture where AI planning can be combined with sensor-based navigation. In the PaT-Net-controlled simulations, planning is used to generate intentions. The intentions are achieved by through the use of PaT-Nets which configure a set of simulated sensors to execute the desired actions. We are writing a paper on the planning project for the AAAI Spring Workshop.

10 Realistic Animation of Liquids: Nick Foster

10.1 Animation of Fluid Phenomena

The goal of this ongoing project is to animate effects such as splashing and wave motion for different liquids. Existing code written by the author for modeling two and three dimensional water motion was extended to include a rendering system for creating pictures of simulated fluid surfaces that are realistic to the human eye [1]. The model is based on the Navier-Stokes equations for incompressible flow, with emphasis on aesthetic realism rather than scientific accuracy.

10.2 High Velocity Collisions

At high velocities boundaries between colliding solids can behave like fluids. Visualizing the behavior of such an interface is difficult because the materials involved may go through many state changes depending on pressure and temperature. An interface was written as a front end for existing mechanical engineering software for calculating impacts between solid bodies in two dimensions. Its main role is to visually display information on the variables in a system, such as velocity, acceleration, pressure, and geometry. The next stage in this project is to extend the interface by developing techniques for visualizing collisions in three-dimensions.

References

[1] N. Foster and D. Metaxas. Visualization of Dynamic Fluid Simulations: Waves, Splashing, Vorticity, Boundaries, Buoyancy. Journal of Engineering Computations, accepted for publication.

11 Pursuing Lung Modeling: Jonathan Kaye

During this period, I have developed qualitative models of respiratory dynamics that I published (with Dimitri Metaxas, John R. Clarke, and Bonnie Webber) in the Proceedings of the First International Symposium for Medical Robotics and Computer-Assisted Surgery, in Pittsburgh. I linked my qualitative model with a graphical rendering of inhalation and exhalation, demonstrating the qualitative model and wound trajectory work in a video for the "Friends of the NLM," shown in Washington to the National Library of Medicine on September 26. For our virtual organ modeling work, we were originally going to use the qualitative dynamics model to drive a Finite Element Method model. However, the dynamics model by Douglas DeCarlo superseded my graphics demonstration, and now we will focus on using the qualitative models to drive non-mechanical aspects of ventilation.

For the next quarter, I am developing qualitative models of blood flow in a vessel and the effects of a compressive force on that vessel. I will continue to develop my models for ventilation by considering an approach that distinguishes forces in the rib cage, diaphragm, and abdomen (previously lumped together as 'chest wall' forces). Ultimately, these efforts will come together in demonstrating the pathological condition called a *tension pneumothorax.*

12 Free Form Deformation: Bond-Jay Ting

This past quarter, I have been working on modeling the respiratory system and creating a visualized simulation for normal breathing and the pneumothorax effect. There are three motions in human breathing that changes the size of the chest cavity: a rib cage motion, diaphragm contraction, and relaxing. The size change of a sealed chest cavity then creates the pressure change which forces the lungs to contract or expand. As simulation is concerned, a rib cage motion is a rigid body joint motion, and the diaphragm and lung motions involve deformable objects.

To simulate the rigid body joint motion, we first create the joints between ribs in the skeletal system. The diaphragm is also attached to the skeletal system.

For a deformable object, the simulation is more complicated. To assure the proper shape and size, we implement the inverse free form deformation. The first step is to create the fully expanded and fully contracted lungs and diaphragm. Then we use inverse free form deformation to compute the control nodes movement. The simulation is accomplished by linear interpolation of the control node movements.

In a pneumothorax case, the chest cavity is no longer sealed. The same motions by the rib cage and diaphragm create only limited pressure change. This creates non-fully expanded lungs. Due to the immaterial properties, there is no fixed shape for non-fully expanded lungs. Gravity will create different lung shapes for different postures. In this simulation, a standing posture is assumed.

To simulate it, we create several intermediate stages of the collapsed lung model based on the X-ray of a real patient. Using the above scheme we can accomplish the simulation task.

13 Dynamics and Lung Modeling: Douglas DeCarlo

I have been implementing a dynamic simulation of the human lung. A 2-D lung model has been constructed that uses finite-elements to simulate the elastic properties of the lung. This lung is embedded in an environment with a deforming diaphragm and chest wall. During inhalation, the chest wall and diaphragm increase the volume of the chest cavity. This in turn, increases the volume of the intra-pleural space (the region between the lung and the chest wall and diaphragm). Pressure changes produce forces that cause the lung to stretch elastically. Currently, pressure changes are instantaneous within a closed region. We will soon use Laplace flow equations to simulate the flow of air. Also, collision detection is used to prevent penetration of the lung and chest wall. In the current model, lung injuries such as a simple pneumothorax can be observed by exposing the intra-pleural space to external pressure. The resulting lung model collapses appropriately.

I soon hope to incorporate more realistic elastic properties of lung tissue, and plan on incorporating these with the model. Once the qualitative behaviors of a lung are accurately simulated, a

3-D model will be constructed. All of the methods uses in the model described above should extend to 3-D (although some, such as the collision detection, will increase in complexity).

14 Efficient Rendering: Jeff Nimeroff

During the last quarter I started work on efficient techniques for hierarchical radiosity. The hierarchical radiosity method is based on the research of the N-body problem in physics and is asymptotically more efficient than traditional or progressive approaches to radiosity. Hierarchical radiosity has recently been extended to handle glossy reflection (three-point transport) but is still limited to reasonably small static scenes. Current trends in geometric simplification and clustering attempt to alleviate the restrictions on small scenes but research is limited to creating radiosity simulations of dynamic environments.

15 Distributed Multilevel Radiosity Solution: Min-Zhi Shao

In last quarter, I investigated a distributed multilevel radiosity solution for complex environments. Radiosity is in many ways a potential method for rendering complex environments such as architectural models. Beside the rapid development of algorithms and computer hardware, however, the radiosity method is still practically limited to relatively simple environments such as single rooms with several thousands of polygons. Since even a modest building design may contain millions of polygons and thousands of light sources, a recent trend in radiosity research is to develop algorithms to meet the application demands.

There are two major obstacles as we view the problem: speed and memory. First, radiosity is a physically-based method which calculates the diffuse light interreflection between each pair of polygons in the environment. The visibility computation of the form-factor matrix is still a dominant factor in the radiosity solution process. Beside the recent development of faster numerical techniques such as the shooting method, the hierarchical method, the clustering method, as well as the wide adoption of specific graphics hardware, it still takes over a hundred hours to obtain a preliminary approximation of an environment with up to tens of thousand input polygons and one million output elements. Second, perhaps a even more significant problem is the memory limitation. One can hardly expect environment models with the above mentioned geometric complexities to fit in the main memory of an ordinary user's workstation. Even with virtual memory supported in many current systems, the data swap between the main memory and the hard disk can force the iteration process to become hopelessly slow.

We propose to use a local network of loosely coupled workstations to partition not only the computation load but also memory load so as to accelerate the iteration process. The adoption of a distributed approach is motivated by the following observations.

- As noticed by previous researchers, direct light interreflection in a typical complex environment, particularly architectural models, are locally dense but globally sparse. On one hand, the environment is highly complex in terms of the extremely large number of polygons. On the other hand, the environment is highly occluded which implies that often only a small cluster of polygons are visible to each other in any particular location. To accelerate the solution, it is thus natural to partition the environment according to the layout design and distribute the computation and memory loads to a network of workstations.

- The demand and supply is always an endless loop. The definition of the term complex depends on not only the number of polygons but also when you refer to it. Comparing to the ordinary architectural models in practice, even the most complex radiosity rendered environments to date can hardly claim that they are moderate complex. Besides, there are other details such as textures and lighting design which could further complicate the solution. Since the radiosity illumination model seems to be an ideal tool for future architectural design, it is valuable to research more efficient and practical algorithms.

- It is unrealistic to expect ordinary users to have the access to very expensive mainframe computers equipped with high computing power and massive main memory as a few research laboratories do. The complicated and heavy design tasks such as architectural models, however, usually require team work which is often performed and communicated by a local network of low-cost graphics workstations equipped with independent CPUs and memory. Therefore, a distributed approach seems to be a feasible alternative to the conventional centralized solution.

Based on the framework of a distributed approach, our goal of this research is to: 1) calculate the radiosity flow between connected partition cells; 2) minimize the data transmission between connected workstation nodes as the iteration proceeds and; 3) maximize convergence speed of the radiosity solution for the entire environment. Our basic ideas can be listed as following:

- Partition the environment into cells based on the design layout and distribute the computation and memory loads accordingly to a network of workstations. Establish a data transmission link between each pair of connected cells.

- For each cell in the environment, add virtual surfaces in portal areas. The cell environment is thus composed of two kinds of surfaces: real and virtual. The real surfaces are assumed to be ideal diffuse as usual. The virtual surfaces which record the radiosity flow coming from connected neighboring cells, however, are no longer ideal diffuse but directional. The directional radiosity flow depends on the radiosity distribution of the neighboring cell from where it comes and is calculated based on a pinhole model. By treating the virtual surfaces as light sources with zero reflectance and directional energy form-factors, a conventional radiosity equation strictly confined to the local geometry can be formed and solved for the cell environment.

- For each cell in the environment, after an approximation is obtained by solving the local radiosity equation, propagate the radiosity flow to its connected neighboring cells via virtual surfaces and at the same time wait for the radiosity flow coming from the neighboring cells. Once the data transmission is completed, the iteration process then proceeds until a converged solution is reached for the entire environment. A naive implementation of this cell-based iteration can proceed as follows: solving the radiosity equation to the finest subdivision for each cell environment, propagating the radiosity flow, solving the radiosity equation to the finest subdivision again with the updated directional radiosity in virtual surfaces from the neighboring cells, propagating the refined radiosity flow, and so on. Notice that the local cell radiosity solution is only an approximation since the radiosity flow from the neighboring cells can only be approximated, the solution to the finest level can be a big waste especially in the beginning stages of the iteration. Thus a better convergence should be expected if the iteration is performed in a coarse-to-fine fashion. In our approach, to further accelerate the convergence, we shall go one step further and adopt a multi-level iterative method in which the iteration is performed in an intertwined coarse-to-fine and fine-to-coarse pattern.

- The data transmission of radiosity flow can be a bottleneck in the iteration process particularly when there is some cell which is connected to many neighboring cells. In our algorithm, we propose to represent the four-dimensional radiosity flow map by the spline-wavelet function which exploits the spatial as well as directional coherence of the radiosity distribution in the virtual surfaces. As a result, the data transmission is greatly compressed.

A A Review of Human, Robotic, and Simulated Grasping Literature: Brett J. Douville

A Review of Human, Robotic, and Simulated Grasping Literature

Brett J. Douville

September 12, 1994

Abstract

This report consists of a description of an approach to graphic simulation of human prehension and an annotated bibliography of the literature surrounding human and robotic grasping. The approach suggested makes use of 16 types of grasps (consisting of both power and precision grips); grasps are selected by task and object knowledge which may be supplied by a database. Finally, the construction of an opposition space using task- and grip-appropriate heuristics drives the selection of particular sites for grasping. A hierarchical architecture to achieve these goals is proposed.

1 An Overview of Human Prehension

The most important ability of the human hand is the opposition of the thumb and the fingers, which allows us to grasp ([26]). Our most basic interactions with the world involve the manipulation of objects with our hands; even the infant quickly develops schemas for basic grasping ([27]). Furthermore, the most comfortable position for the hand, the rest position, is an intermediate position between the primary power and precision grasps ([26]). Any simulation of human abilities, therefore, whether generated by computer graphics or physical robots, must incorporate this very basic of human skills.

1.1 Opposition Spaces

Opposition spaces (see [11]) are a framework for describing stable prehensile grasps in which an opposition vector describes the relationship between two or more virtual fingers. Virtual fingers exist in a grasp where forces will need to be applied to effect a stable grasp (see, for example, [2]); virtual fingers are an abstract representation of forces which must be generated by the fingers and thumb of the human hand to grasp and manipulate an object. A mapping from virtual fingers to real fingers must then be generated to create an actual physical grasp from the one selected by virtual finger placement.

There are three types of oppositions:

1. PAD opposition: the opposition vector runs between hand surfaces in a direction parallel to the palm.

2. PALM opposition: the opposition vector runs between hand surfaces in a direction generally perpendicular to the palm.

3. SIDE opposition: the opposition vector runs between hand surfaces in a direction generally transverse to the palm.

1.2 Opposition Space Phases

The phases of opposition space usage reflect the psychological and physical events which occur during human grasping. MacKenzie and Iberall ([21]) present an overview of the theory of opposition spaces in human and robotic prehension, which was originally described in [11]. MacKenzie and Iberall separate the grasping task into several phases; their diagram of these is reproduced in Fig. 1.

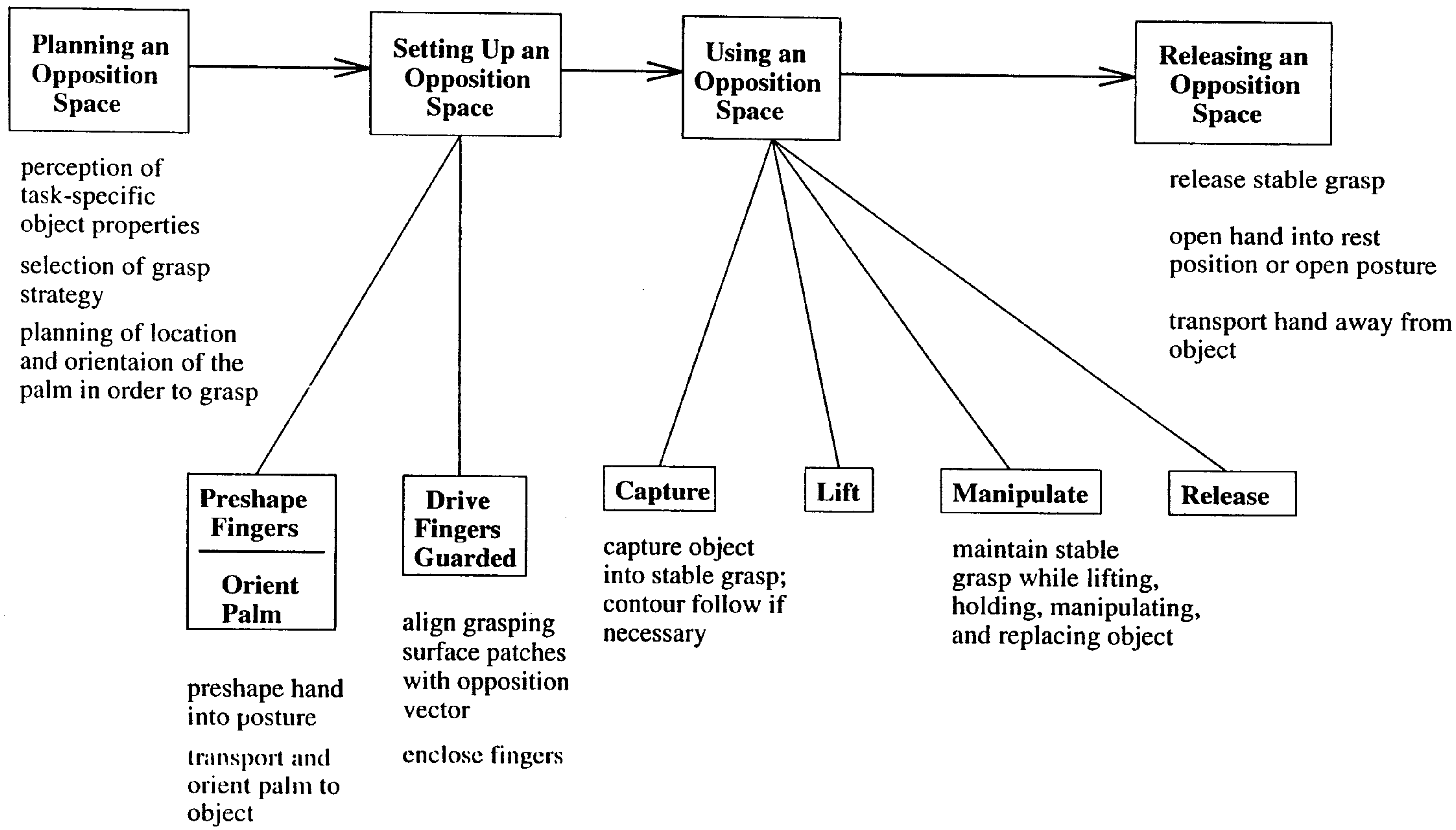

Figure 1: The Phases of Opposition Space Prehension

Planning an Opposition Space

In the first phase, an opposition space is planned from task-specific, intrinsic, and extrinsic object properties; from these, a grasp strategy is selected and a location and orientation of the palm for grasping is generated.

Perception of task-specific object properties drives the selection of a particular grasp. In general, this task knowledge is built up from past experience (for example, we know how to grab a hammer because we've hammered nails before), however, functionality can be discovered and an appropriate grasp selected on the basis of similar tasks in the past (for example, using a rolled-up newspaper when no fly swatter is available).

Other object properties affect the eventual grasp: for example, a particularly inaccessible surface would eliminate any grasps using that surface. Similarly, extremely smooth or sharp surfaces are also culled out as being unsuitable for grasping. Once an opposition space has been planned, setting it up can proceed.

Setting up an Opposition Space

Once the initial planning for an opposition space has been done, it remains for the central nervous system (CNS) to set up the opposition space before it can actually be used. Setting up consists of two subphases: the first, in which the hand is preshaped and the palm oriented, and the second, in which the fingers are enclosed about the object.

The hand is preshaped into a posture depending primarily on the object; the peak aperture to which the hand opens has been shown to be linear in object size [22], with an increase of .77 cm per object diameter increase of 1 cm from a baseline value. The adopted posture will flow from this peak aperture position once the hand reaches its destination.

The orientation of the palm is achieved by a "ballpark method" ([14], [2]) in which the hand is brought near to the object, but not in a precisely specified location.

Driving the fingers in a guarded manner occurs once the hand has reached its desired location and orientation. This consists of alignment of the hand's grasping surfaces (*i.e.* the pads of the fingers in pad opposition, or the palm of the hand and some finger pads in palm opposition), which will serve as virtual fingers, with the actual opposition vector which was selected in the planning phase. Once the fingers have been aligned, the fingers can be closed towards the grasp selected in the planning phase; perturbations can be perceived by tactile sensing, and adjustments for these can be made.

Using an Opposition Space

Although the use of an opposition space, and the release of such a space, is not fully covered by automatic grasping, a brief discussion of human usage of opposition spaces is offered for the sake of completion.

Use of opposition spaces is rougly divided into four categories:

1. Object capture: The object is captured into a stable grasp, depending on physics and tactile information. The physics governing stable grasping involves consideration of fingertip and joint forces, and is not entered into in detail here. For detailed coverage, see [21].

2. Object lifting: The object is removed from its support and the agent now supports the object.

3. Object manipulation: A stable grasp is maintained (though perhaps altered) in the performance of tasks and other manipulations of the object (for example, transport).

4. Object release: This differs from the release of an opposition space: in object release, the object is merely dropped.

Releasing an Opposition Space

The release of an opposition space is far different from the release of an object which can be encountered while using an opposition space. Releasing an opposition space consists of finding a supporting surface onto which the object can be placed, opening the hand to release the object (into an open or rest posture), and finally moving the hand away from the object.

2 An Overview of Robotic Prehension

The robotic prehension literature is grappling with several robotic grasping issues; these are primarily tactile sensing (see [7], [8], [24], [10], and others), knowledge representation (see [12], [30], [20], [13], and others), grasp recognition (*e.g.* [16] and [15]), and finally, grasp selection (*e.g.* [3] and [29]).

Although all those concerned with these topics have interesting points to make, the most important information for simulation can be drawn from the grasp selection and knowledge representation literature. In particular, the framework described by Bekey *et al* in [3] is the approach which appears to bring the most to the graphic simulation of human prehension.

Bekey *et al* describes an architecture in which four types of knowledge drive the selection of a particular grasp mode:

1. Knowledge about the robot / simulated hand

2. Knowledge of the target object geometry (in terms of constructive solid geometry)

3. Knowledge about the task

4. Knowledge about human grasping

For human prehension, knowledge of intrinsic object properties is also critical: for example, surface texture and temperature are properties which must be considered when selecting a grasp strategy. Thus, any useful simulation of human prehension must incorporate this information.

Bekey *et al* use a knowledge base to provide an ordered list of selected grasps when given task information and the geometric primitives which compose the object. Heuristics then provide additional information, such as where to grasp, where to place the fingers and approach orientation. This approach should be reasonably adaptable to simulated grasping, as the information above can be embedded in the Jack environment, the object-specific reasoner, and an executive controller driving automatic grasping.

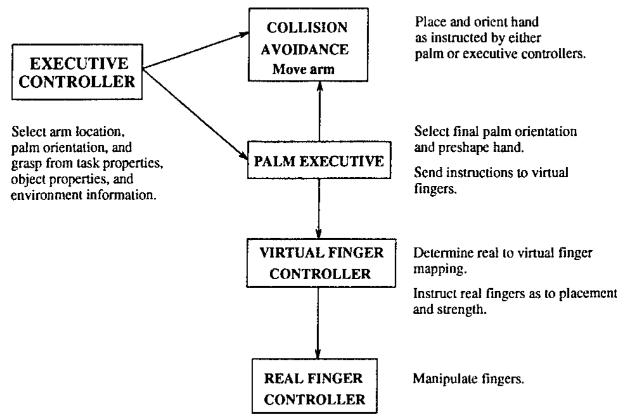

Figure 2: An Architecture for Automatic Grasping

3 An Overview of Simulated Grasping

Grasping simulation literature in computer graphics appears to be rather sparse; however, there was a paper by Rijpkema and Girard presented at Siggraph 1991 ([28]). The only other approach described in the literature addresses motion planning and not actual automated grasping ([19]).

Rijpkema and Girard's approach, though interesting, lacks a firm basis in the psychological elements of human grasping. They note the need for task-based information and a strong relationship with the environment, but don't ground their choices firmly in the cognitive literature. Rather than restricting the range of grasps to ten and not allowing adjustments to be made in them, in the interest of more realistic simulation, it seems more reasonable to have a large set of grasps which will be fine-tuned to fit the particular geometry of the target object.

4 An Architecture for Automatic Grasping

Figure 2 outlines an architecture for the simulation of automatic grasping. It is a distributed architecture with both feedforward and feedback correction: each level of the architecture passes information both forward to the next level, and backward to the previous level.

The executive controller will be responsible for selecting one of sixteen initial grasps (chosen from [6]) from task and object information as specified in [3]; using this information, the executive controller will select and set up an opposition space as outlined in Section 1.2, applying heuristics as necessary. (Later, this controller will consider the feasibility of the selected grasp based on strength data; in the preliminary implementation, the hand will be assumed to be strong enough to maintain the grasp stably.)

Once a ballpark estimate of the target location and orientation of the palm has been generated, the executive can pass this information along to the collision avoidance module and a controller (labelled the "palm executive" in Fig. 2) for the hand. While the collision avoidance module orients and places the hand, the palm executive will preshape the hand to an aperture (generated from a linear function of the diameter of the object), and plan specific orientation of the virtual fingers used in the grasp.

Once the hand has reached its target position and orientation, the palm executive will instruct the virtual fingers as to their placement. The virtual finger controller will then determine a real-to-virtual finger mapping (guided by object geometry, task information, and the selected grasp) and instruct the real fingers as to their positions. The fingers will then be closed about the object.

5 Miscellaneous

While doing a literature search, I came upon a number of items which were interesting, but did not directly influence the design of the architecture.

Brand and Hollister ([4]) present a number of interesting topics in terms of the mechanical manipulation of the human hand; in particular, they present data on the percentage of weight each muscle represents in the hand as well as tendon strength information.

An, Berger, and Cooney ([1]) present several topics regarding the kinematics and mechanics of the wrist joint in particular, with some additional information about the hand. This will bear further research into strength-guided grasping.

Conclusion

The automation of human prehension, whether simulated by graphic agents or by real robots, presents a difficult task for the designer. The approach suggested here utilizes 16 particular grasps (consisting of both power and precision grips) which are selected by task and object knowledge. Finally, the construction of an opposition space using task- and grip-appropriate heuristics drives the selection of particular sites for grasping; a hierarchical architecture addressing these goals has been proposed.

References

[1] Kai-Nan An, Richard A. Berger, and William P. Cooney III, editors. *Biomechanics of the Wrist Joint.* Springer-Verlag, New York, 1991.

[2] M. A. Arbib, T. Iberall, and D. M. Lyons. *Coordinated control programs for movements of the hand*, pages 111–129. Springer-Verlag, Berlin, 1985.

[3] George A. Bekey, Huan Liu, Rajko Tomovic, and Walter J. Karplus. Knowledge-based control of grasping in robot hands using heuristics from human motor skills. *IEEE Transactions on Robotics and Automation*, 9(6):709–722, December 1993.

[4] Paul W. Brand and Anne Hollister. *Clinical Mechanics of the Hand.* Mosby, Philadelphia, 1993.

[5] Mark R. Cutkosky. *Robotic Grasping and Fine Manipulation.* Kluwer Academic Publishers, 1985.

[6] Mark R. Cutkosky and Robert D. Howe. Human grasp choice and robotic grasp analysis. In S. T. Venkataraman and T. Iberall, editors, *Dextrous Robot Hands*, pages 5–31. Springer-Verlag, New York, 1990.

[7] Ronald S. Fearing. Tactile sensing for shape interpretation. In S. T. Venkataraman and T. Iberall, editors, *Dextrous Robot Hands*, pages 209–238. Springer-Verlag, New York, 1990.

[8] Ronald S. Fearing and Joan M. Hollenbach. Basic solid mechanics for tactile sensing. In *Proceedings of the 1984 IEEE Conference on Robotics and Automation*, pages 266–272. IEEE, 1984.

[9] Dennis Holding. *Human Skills; Studies in Human Performance.* John Wiley & Sons, New York, 1989.

[10] Howe, Popp, Akella, Kao, and Cutkosky. Grasping, manipulation, and control with tactile sensing. In *Proceedings of the 1990 IEEE International Conference on Robotics and Automation*, pages 1258–1263. IEEE, 1990.

[11] T. Iberall, G. Bingham, and M. A. Arbib. Opposition space as a structuring concept for the analysis of skilled hand movements. In H. Heuer and C. Fromm, editors, *Generation and modulation of action patterns*, pages 158–173. Berlin: Springer-Verlag, 1986.

[12] T. Iberall, J. Jackson, L. Labbe, and R. Zampano. Knowledge-based prehension: Capturing human dexterity. In *Proceedings of the 1988 IEEE International Conference on Robotics and Automation*, pages 82–87. IEEE, 1988.

[13] Thea Iberall. The representation of objects for grasping. In *Proceedings of the Eighth Cognitive Society Conference*, pages 547–561. Cognitive Society, 1986.

[14] Thea Iberall. A ballpark approach to modelling human prehension. In *1987 IEEE International Conference on Neural Networks*, volume IV, pages 535–543. IEEE, 1987.

[15] Sing Bing Kang and Katsushi Ikeuchi. Grasp recognition using the contact web. In *Proceedings of the 1992 IEEE International Conference on Intelligent Robots and Systems*, pages 194–201. IEEE, 1992.

[16] Sing Bing Kang and Katsushi Ikeuchi. A grasp abstraction hierarchy for reccognition of grasping tasks from observation. In *Proceedings of the 1993 IEEE International Conference on Intelligent Robots and Systems*, pages 621–628. IEEE, 1993.

[17] Jung-Ha Kim. *Kinematics and Statics of Multifingered Grasping.* PhD thesis, University of Pennsylvania, 1990.

[18] Daniel Kleppner and Robert Kolenkow. *An Introduction to Mechanics.* McGraw-Hill, New York, 1973.

[19] Yoshihito Koga, Koichi Kondo, James Kuffner, and Jean-Claude Latombe. Planning motions with intentions. In *Computer Graphics (SIGGRAPH '94 Proceedings)*, pages 395–408, 1994.

[20] Huan Liu, Thea Iberall, and George Bekey. The multi-dimensional quality of task requirements for dextrous robot hand control. In *Proceedings of the 1989 IEEE International Conference on Robotics and Automation*, pages 452–457. IEEE, 1989.

[21] Christine L. MacKenzie and Thea Iberall. *The Grasping Hand*, volume 104 of *Advances in Psychology.* North-Holland, 1994.

[22] R. G. Marteniuk, C. L. MacKenzie, and J. L. Leavitt. The inadequacies of a straight physical account of motor control. In H. T. A. Whiting, O. G. Meijer, and P. C. W. van Wieringen, editors, *The Natural-Physical Approach to Movement Control*, pages 95–115. Ablex, Norwood, NJ, 1990.

[23] Matthew T. Mason and Jr. J. Kenneth Salisbury. *Robot Hands and the Mechanics of Manipulation.* MIT Press, Cambridge, Massachusetts, 1985.

[24] Ian D. McCammon and Steve C. Jacobsen. Tactile sensing and control for the utah/mit hand. In S. T. Venkataraman and T. Iberall, editors, *Dextrous Human Hands*, pages 239–266. Springer-Verlag, New York, 1990.

[25] P. Morasso and V. Tagliasco, editors. *Human Movement Understanding*, volume 33 of *Advances in Psychology.* North-Holland, 1986.

[26] John Napier. *Hands.* Princeton University Press, 1993. Revised by John Tuttle.

[27] Jean Piaget. *The Origins of Intelligence in Children.* London: Routledge & Kegan Paul, 1953.

[28] Hans Rijpkema and Michael Girard. Computer animation of knowledge-based human grasping. In Thomas W. Sederberg, editor, *Computer*

Graphics (SIGGRAPH '91 Proceedings), volume 25, pages 339–348, July 1991.

[29] Thomas Specter. Primitive-based control of the utah/mit dextrous hand. In *Proceedings of the 1991 IEEE International Conference on Robotics and Automation*, pages 866–877. IEEE, 1991.

[30] S. A. Stansfield. Knowledge-based robotic grasping. In *Proceedings of the 1990 IEEE International Conference on Robotics and Automation*, pages 1270–1275. IEEE, 1990.

[31] E. R. Tichauer. *The Biomechanical Basis of Ergonomics; Anatomy Applied to the Design of Work Situations.* John Wiley & Sons, 1978.

[32] Kimon P. Valavanis and George N. Saridis. *Intelligent Robotic Systems: Theory, Design and Applications.* Kluwer Academic Publishers, Boston, 1992.

B Sight and Sound: Generating Facial Expressions and Spoken Intonation From Context: Catherine Pelachaud and Scott Prevost

SIGHT AND SOUND: GENERATING FACIAL EXPRESSIONS AND SPOKEN INTONATION FROM CONTEXT

*Catherine Pelachaud and Scott Prevost**

Computer and Information Science, University of Pennsylvania

200 South 33rd Street, Philadelphia, PA 19104-6389, USA

Internet: pelachau@graphics.cis.upenn.edu, prevost@linc.cis.upenn.edu

Abstract

This paper presents an implemented system for automatically producing prosodically appropriate speech and corresponding facial expressions for animated, three-dimensional agents that respond to simple database queries. Unlike previous text-to-facial animation approaches, the system described here produces synthesized speech and facial animations entirely from scratch, starting with semantic representations of the message to be conveyed, which are based in turn on a discourse model and a small database of facts about the modeled world.

1 Introduction

As research on the simulation of autonomous virtual human agents progresses, two major issues in human-machine interaction must be addressed. First, proper intonation is necessary for conveying the information structure of utterances with respect to the underlying discourse structure, expressing important distinctions of contrast and focus ([19], [17], [18]). Second, realistic facial expressions and lip movements help in providing relevant information about discourse structure, turn-taking protocols and speaker attitudes ([7], [8], [14]). We propose that integrating models for generating proper intonation and facial expressions will improve the intelligibility and naturalness of utterances produced both by meaning-to-speech systems and by more elaborate systems involving virtual animated human agents (e.g. [3]).

The intonation generation model is based on Combinatory Categorial Grammar (CCG – cf. [19]), a formalism which easily integrates the notions of syntactic constituency, prosodic phrasing and information structure. Based on the CCG grammar, a simple discourse model and a small knowledge base represented in Prolog, the system produces spoken responses to database queries with appropriate intonation. Given the precise timings for phonemes and intonational phenomena in the speech wave, we produce precise specifications for generating the lip movements and facial expressions for a graphical model of a human head. Results from our current implementation demonstrate the system's ability to generate a variety of intonational possibilities and facial animations for a given sentence depending on the discourse context.

Previous work in the area of intonation generation includes studies by Terken ([21]), Houghton, Isard and Pearson (cf. [11]), Davis

*We are grateful to AT&T Bell Laboratories for allowing us access to the TTS speech synthesizer, and to Mark Beutnagel, Julia Hirschberg, and Richard Sproat for patient advice on its use. The usual disclaimers apply. The research was supported in part by NSF grant nos. IRI90-18513, IRI90-16592, IRI91-17110 and CISE IIP-CDA-88-22719, DARPA grant no. N00014-90-J-1863, and ARO grant no. DAAL03-89-C0031.

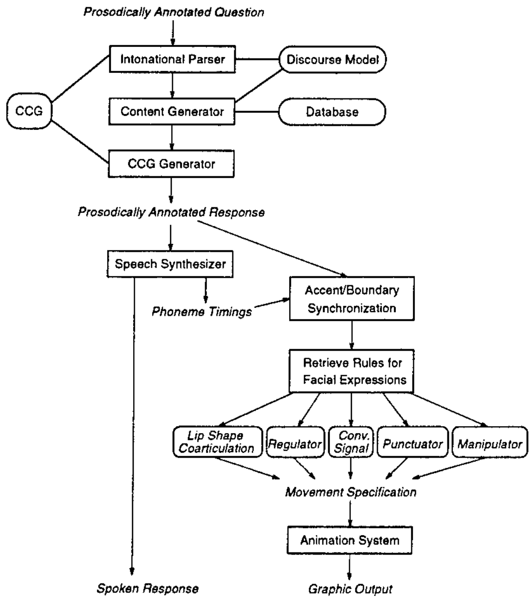

Figure 1: Architecture

and Hirschberg (cf. [6], [10]), and Zacharski *et al.* ([23]). Benoit *et al.* ([1]), Brooke ([2]), Cohen *et al.* ([4]), Hill *et al.* ([9]), Lewis *et al.* ([12]) and Terzopoulos *et al.* ([22]) have worked on synchronizing lip movements with speech, producing quite striking results. Takeuchi *et al.* ([20]) implemented a user-interface in which a 3D facial model responds to queries posed by a user. In this system, the generation of the facial expressions accompanying the answer depends on an analysis of the conversational situation and the selection of facial expressions from a database of facial displays.

The system described here expands the work of the aforementioned researchers by linking contextually appropriate intonation with the corresponding facial expressions, and generating the 3D facial animations automatically from semantic, information structural and discourse structural representations.

2 The Implementation

Using the CCG theory of prosody outlined in [19], [17] and [18], the implemented system undertakes the task of specifying contextually appropriate intonation and facial animation for spoken responses to database queries. The process, illustrated in figure 1, begins with a fully segmented and prosodically annotated representation of a spoken query as shown in example (1), which involves a simple database of facts about stereo components. We employ a simple bottom-up shift-reduce parser to identify the semantics of the question, dividing it into a topic or "theme" and a comment or "rheme", and marking "focused" items within themes and rhemes with the * operator, as shown in example (2).

(1) I know which components produce MUDDY bass,
but WHICH components produce CLEAN bass?
L+H* LH% H* LL$

(2) Proposition:
$s : \lambda x.component(x) \& produce(x, *clean(bass))$
Theme:
$s : \lambda x.component(x) \& produce(x, *clean(bass))/ (s : produce(x, *clean(bass)) \backslash np : x)$
Rheme:
$s : produce(x, *clean(bass)) \backslash np : x$

The content generation module has the task of determining the semantics and information structure of the response, marking focused items based on the contrastive stress algorithm described in [18]. For the question given in (1), the strategic generator produces the representation for the response shown in example (3), where the appropriate theme can be paraphrased as "what produces clean bass", the appropriate rheme as "amplifiers", and where the context includes alternative components and audio qualities:

(3) Proposition:
$s : produce(*amplifiers, *clean(bass))$
Theme:
$s : produce(x, *clean(bass)) \backslash np : x$
Rheme:
$np : *amplifiers$

From the output of the content generator, the CCG generation module (described in [17]) produces a string of words and Pierrehumbert-style markings representing the response, as shown in example (4).

(4) AMPLIFIERS produce CLEAN bass.
H* L L+H* LH$

The final aspect of speech generation involves translating such a string into a form usable by a suitable speech synthesizer. The current implementation uses the Bell Laboratories TTS system [13] as a post-processor to synthesize the speech wave and produce precise timing specifications for phonemes. The duration specifications are then automatically annotated with pitch accent peaks and intonational boundaries in preparation for processing by the facial expression rules (see also [3]).

The facial animation system starts from a functional group including lip shapes, conversational signals, punctuator signals, regulators and manipulators, offering algorithms which incorporate synchrony ([5]), create coarticulation effects, emotional signals, and eye and head movements ([15], [16]). The rules automatically generate the facial actions corresponding to the input utterance. Conversational signals, such as movements occurring on accents (e.g. the raising of an eyebrow), start and end with the accented word. For instance, on *amplifier*, the brow starts raising on 'a', remains raised until the end of the word, and ends raising on 'r'. On the other hand, the punctuator signals, such as smiling, coincide with pauses. Blinking is synchronized at the phoneme level, due to biological need, accents or pauses. On *amplifier*, for example, the eyes start closing on 'a', remain closed on 'm' and start opening on 'p'. Head nods and shakes appear on both accents and pauses. In addition, the movement of the head is affected by speaker turn-taking, moving away from the listener at the beginning of a speaking turn and toward the listener at the end of a speaking turn.

Two parameters characterize a facial action: its *presence* and its *type*. Such a decomposition permits one to simulate different behaviors, allowing one agent to punctuate each accent or pause by smiling, while allowing another agent to display hardly any expression at all. A set of such parameters is defined for all of the functional groups.

The computation of the lip shape is done in three passes. First, phonemes, which are characterized by their degree of deformability, are processed one segment at a time using the look-ahead model to search for the proximal deformable segments whose associated lip shapes influence the current segment. For example, in *amplifier* the 'l' receives the same lip shape as the following vowel 'i'—that is, the movement of the 'i' begins before the onset of its sound. Second, the spatial properties of muscle contractions are taken into account by adjusting the sequence of contracting muscles when antagonistic movements succeed one another (i.e. movements involving very different lip positions, such as pucker movements versus the extension of the lips). And finally, the temporal properties of muscle contractions are considered by determining whether a muscle has enough time to contract before (or relax after) the surrounding lip shape.

3 Examples

In the examples shown below, the speaker manifests different behaviors depending on whether s/he is asking a question, making a statement, accenting a word or pausing. When asking a question, the speaker raises the eyebrows and looks up slightly to mark the end of the question. When replying, or when turning over the floor to the other person, the speaker nods the head. To emphasize a particular word, s/he raises the eyebrows, nods the head and/or blinks. During the brief pauses at the end of statements and within statements, the speaker blinks and looks at the listener.

(5) I know which amplifier produces clean BASS,

but which amplifier produces clean TREBLE?
L+H* LH% H* LL$

The BRITISH amplifier produces clean TREBLE.
H* L L+H* LH$

(6) I know which British component produces MUDDY treble,

but which British component produces CLEAN treble?
L+H* LH% H* LL$

The British AMPLIFIER produces CLEAN treble.
H* L L+H* LH$

In utterance (5), the word *British* is accented and accompanied by a raised eyebrow indicating a conversational signal denoting contrast. In utterance (6), on the other hand, the word *amplifier* is accented and marked by the action of the eyebrows. The same argument differentiates the appearance of the movement on the word *treble* in (5) and the word *clean* in (6). Moreover, a punctuating blink marks the end of (6), starting on the pause after the word *treble*. In (5) a blink coincides with the accented word *treble* (as a conversational signal) and with the pause marking the end of the utterance (as a punctuator), resulting in two blinks emitted in succession at the end of the utterance. In both examples, the pause between the two intonational phrases '*the British amplifier*' and '*produces clean treble*', is accompanied by movement of the eyebrows and the turning of the speaker's head towards the listener.

4 Conclusions

The system described above produces quite sharp and natural-sounding distinctions of intonation contour as well as visually distinct facial animations for minimal pairs of queries and responses generated automatically from a discourse model and a simple knowledge base. The examples in the previous section (and others presented at the workshop) illustrate the system's capabilities and provide a sound basis for exploring the role of prosody and facial expressions in human-machine interactions. Future areas of research include evaluating results and exploring the relevance of our current system to large scale animation systems involving autonomous virtual human agents (cf. [3]).

5 References

[1] C. Benoit. Why synthesize talking faces? In *Proceedings of the ESCA Workshop on Speech Synthesis*, pages 253–256, Autrans, 1990. ESCA.

[2] N.M. Brooke. Computer graphics synthesis of talking faces. In *Proceedings of the ESCA Workshop on Speech Synthesis*, Autrans, 1990. ESCA.

[3] J. Cassell, C. Pelachaud, N. Badler, M. Steedman, B. Achorn, T. Becket, B. Douville, S. Prevost, and M. Stone. Animated conversation: Rule based generation of facial expression, gesture and spoken intonation for multiple conversational agents. In *SIGGRAPH'94*, 1994. To appear.

[4] Michael M. Cohen and Dominic W. Massaro. Modeling coarticulation in synthetic visual speech. In D. Thalmann N. Magnenat-Thalmann, editor, *Computer Animation '93*. Springer-Verlag, 1993.

[5] W.S. Condon and W.D. Osgton. Speech and body motion synchrony of the speaker-hearer. In D.H. Horton and J.J. Jenkins, editors, *The perception of Language*, pages 150–184. Academic Press, 1971.

[6] J. Davis and J. Hirschberg. Assigning intonational features in synthesized spoken discourse. In *Proceedings of the 26th Annual Meeting of the Association for Computational Linguistics*, pages 187–193, Buffalo, 1988.

[7] S. Duncan. Some signals and rules for taking speaking turns in conversations. In Weitz, editor, *Nonverbal Communication*. Oxford University Press, 1974.

[8] P. Ekman. About brows: emotional and conversational signals. In M. von Cranach, K. Foppa, W. Lepenies, and D. Ploog, editors, *Human ethology: claims and limits of a new discipline: contributions to the Colloquium*, pages 169–248. Cambridge University Press, Cambridge, England; New-York, 1979.

[9] D.R. Hill, A. Pearce, and B. Wyvill. Animating speech: an automated approach using speech synthesised by rules. *The Visual Computer*, 3:277–289, 1988.

[10] J. Hirschberg. Accent and discourse context: Assigning pitch accent in synthetic speech. In *Proceedings of AAAI: 1990*, pages 952–957, 1990.

[11] G. Houghton and M. Pearson. The production of spoken dialogue. In M. Zock and G. Sabah, editors, *Advances in Natural Language Generation: An Interdisciplinary Perspective, Vol. 1*. Pinter Publishers, London, 1988.

[12] J.P. Lewis and F.I. Parke. Automated lip-synch and speech synthesis for character animation. *CHI + GI*, pages 143–147, 1987.

[13] M. Liberman and A. L. Buchsbaum. Structure and usage of current Bell Labs text to speech programs. Technical Memorandum TM 11225-850731-11, AT&T Bell Laboratories, 1985.

[14] D.W. Massaro. *Speech Perception by Ear and Eye: A Paradigm for Psychological Inquiry*. Cambridge University Press, 1989.

[15] C. Pelachaud, N.I. Badler, and M. Steedman. Linguistic issues in facial animation. In N. Magnenat-Thalmann and D. Thalmann, editors, *Computer Animation '91*, pages 15–30. Springer-Verlag, 1991.

[16] C. Pelachaud, M.L. Viaud, and H. Yahia. Rule-structured facial animation system. In *IJCAI 93*, 1993.

[17] S. Prevost and M. Steedman. Generating contextually appropriate intonation. In *Proceedings of the 6th Conference of the European Chapter of the Association for Computational Linguistics*, pages 332–340, Utrecht, 1993.

[18] S. Prevost and M. Steedman. Using context to specify intonation in speech synthesis. In *Proceedings of the 3rd European Conference of Speech Communication and Technology (EUROSPEECH)*, pages 2103–2106, Berlin, 1993.

[19] M. Steedman. Structure and intonation. *Language*, pages 260–296, 1991.

[20] A. Takeuchi and K. Nagao. Communicative facial displays as a new conversational modality. In *ACM/IFIP INTERCHI '93*, Amsterdam, 1993.

[21] J. Terken. The distribution of accents in instructions as a function of discourse structure. *Language and Structure*, 27:269–289, 1984.

[22] D. Terzopoulos and K. Waters. Techniques for realistic facial modelling and animation. In N. Magnenat-Thalmann and D. Thalmann, editors, *Computer Animation '91*, pages 45–58. Springer-Verlag, 1991.

[23] R. Zacharski, A.I.C. Monaghan, D.R. Ladd, and J. Delin. BRIDGE: Basic research on intonation in dialogue generation. Technical report, HCRC: University of Edinburgh, 1993. Unpublished manuscript.

C Automatically Generating Conversational Behaviors in Animated Agents: Justine Cassell, Catherine Pelachaud, Norman Badler, and Mark Steedman

1 Automatically Generating Conversational Behaviors in Animated Agents: Justine Cassell, Catherine Pelachaud, Norman Badler, Mark Steedman

In the creation of synthetic computer characters, the creators shouldn't have to create or control every move of their lifelike human agents: for example, during the progress of a search or planning system, responding to knowledge base queries, or portraying autonomous agents during real-time virtual environment simulations. For these *automated* characters we must *generate* behavior on the basis of rules abstracted from the study of human behavior.

The behavior that we concentrate on in this project is conversation (that is, an interactive dialogue between two agents). Conversation includes spoken language (words and contextually appropriate intonation marking topic and focus), but it also includes facial movements (lip shapes, expressions, gaze, head movement), and hand gestures (points, beats, and movements representing the topic of accompanying speech). Without all of these verbal and non-verbal behaviors, one cannot have realistic, lifelike, autonomous agents. To this end, our system *automatically* animates conversations between multiple human-like agents with appropriate and synchronized speech, intonation, facial expressions, and hand gestures.

The system is composed of a *Dialogue Generation* program which allows gesture and conversational intonation to be generated along with speech. The output of the dialogue generation program is speech annotated with descriptions of appropriate intonation and gesture, which are then sent on to an intonation synthesis module, facial expression specification module, and gesture and facial synthesis modules. The *Intonation Synthesis* model generates actual intonational tunes as a function of the information structure of the discourse. The *Facial Expression Specification* module generates head and eye movements as a function of dialogic categories such as `planning what to say, feedback to speaker's contribution`. The *Gesture and Facial Movement Synthesis* module has two parts;

1 - A *Synchronization module*: Interaction between agents and synchronization of gaze, hand and head movements to the dialogue for each agent are accomplished using Parallel Transition Networks (PaT-Nets), which allow coordination rules to be encoded as simultaneously executing finite state automata.

2- A *Movement Specification module*: It selects and generates nods, gaze direction, handshapes, wrist and arm motion.

The conversation below is an example of the discourse output from the dialogue generation program. Following it is a description of some of the nonverbal and intonational behaviors generated with the speech.

The dialogue is unnaturally repetitive and explicit in its goals because the dialogue generation program that produced it has none of the conversational inferences that allow humans to follow leaps of reasoning.

Gilbert:	Do you have a blank check?
George:	Yes, I have a blank check.
Gilbert:	Do you have an account for the check?
George:	Yes, I have an account for the check.
Gilbert:	Does the account contain at least fifty dollars?
George:	Yes, the account contains eighty dollars.
Gilbert:	Get the check made out to you for fifty dollars and then I can withdraw fifty dollars for you.
George:	All right, let's get the check made out to me for fifty dollars.

When Gilbert asks a question, his voice rises. When George replies to a question, his voice falls. When Gilbert asks George whether he has a blank check, he stresses the word "check". When he asks George whether he has an account for the check, he stresses the word "account".

Every time Gilbert replies affirmatively ("yes"), or turns the floor over to Gilbert, he nods his head, and raises his eyebrows. George and Gilbert look at each other when Gilbert asks a question, but at the end of each question, Gilbert looks up slightly. During the brief pause at the end of affirmative statements the speaker blinks.

In saying the word "check", Gilbert sketches the outlines of a check in the air between him and his listener. In saying "account", Gilbert forms a kind of box in front of him with his hands: a metaphorical representation of a bank account in which one keeps money. When he says the phrase "withdraw fifty dollars," Gilbert draws his hand towards his chest.

Although the two agents do not visually *perceive* each other's gestures, speech, etc., their actions are nevertheless determined by the evolving conversation. The sequence, and hence their motions, is not pre-determined. Thus if new information becomes available, then all the communication acts will adjust – and be animated – accordingly. It is this expressive flexibility and response to novel situations that make these automated characters lifelike.

References

[1] J. Cassell, C. Pelachaud, N.I. Badler, M. Steedman, B. Achorn, T. Becket, B. Douville, S. Prevost, C. Seah, and M. Stone. Animated conversation: Rule based generation of facial expression, gesture and spoken intonation for multiple conversational agents. In *SIGGRAPH '94*, 1994.

The research reported on here was carried out jointly with Brett Achorn, Brett Douville, Scott Prevost, and Matthew Stone.