PUBLIC NEWS NETWORK: DIGITAL SAMPLING TO CREATE A HYBRID MEDIA FEED

A Thesis

by

JACK ERIC STENNER

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

May 2003

Major Subject: Visualization Sciences

PUBLIC NEWS NETWORK: DIGITAL SAMPLING TO CREATE A HYBRID MEDIA FEED

A Thesis

by

JACK ERIC STENNER

Submitted to Texas A&M University
in partial fulfillment of the requirements
for the degree of

MASTER OF SCIENCE

Approved as to style and content by:

Carol LaFayette
(Chair of Committee)

Karen Hillier
(Member)

Donna Hajash
(Member)

Phillip J. Tabb
(Head of Department)

May 2003

Major Subject: Visualization Sciences

ABSTRACT

Public News Network: Digital Sampling to
Create a Hybrid Media Feed. (May 2003)

Jack Eric Stenner, B.E.D., Texas A&M University
Chair of Advisory Committee: Prof. Carol LaFayette

A software application called Public News Network (PNN) is created in this thesis, which functions to produce an aesthetic experience in the viewer. The application engenders this experience by presenting a three-dimensional virtual world that the viewer can navigate using the computer mouse and keyboard. As the viewer navigates the environment she sees irregularly shaped objects resting on an infinite ground plane, and hears an ethereal wind. As the viewer nears the objects, the sound transforms into the sound of television static and text is displayed which identifies this object as representative of an episode of the evening news. The viewer "touches" the episode and a "disembodied" transcript of the broadcast begins to scroll across the screen. With further interaction, video of the broadcast streams across the surfaces of the environment, distorted by the shapes upon which it flows. The viewer can further manipulate and repurpose the broadcast by searching for words contained within the transcript. The results of this search are reassembled into a new, re-contextualized display of video containing the search terms stripped from their original, pre-packaged context. It is this willful manipulation that completes the opportunity for true meaning to appear.

To my wife Stephanie, and daughter Sydney, who put up with me living in front of the computer all this time.

ACKNOWLEDGMENTS

I would like to thank Carol LaFayette, who understood my interest in digital technology and art, and gave me license to shape this project as an artwork. Her insight was critical in assisting me to coalesce many ideas into a whole. I would also like to thank Karen Hillier for her gracious support for me, as well as my family, over the last several years. Her understanding of my concerns enabled me to feel comfortable following a path that is a bit "different." Thanks also to Donna Hajash who "saved-the-day" by replacing one of my original committee members who was no longer able to serve. Additionally, I would like to thank Andruid Kerne, whose input regarding Java helped me immensely.

I would also like to thank several of my classmates. Amy Tucker was an inspiration to watch as she grew artistically while at the VizLab. A similar inspiration was Jeremy Sternberg, who always seemed to provide a fresh, if skewed, perspective on everyday life. Also, I would like to thank Michael Mistrot and Jay Faulkner, who helped push me over the "hump" on several technical issues that had me stumped.

TABLE OF CONTENTS

Page

ABSTRACT iii

DEDICATION iv

ACKNOWLEDGMENTS v

TABLE OF CONTENTS vi

LIST OF FIGURES viii

CHAPTER

I INTRODUCTION 1

Objective 1
Foundation 3

II PREVIOUS WORK 7

Artworks 7
Art Tools 13

III METHODOLOGY 19

Concept 19
Simulation 21
Sampling 23
Dynamism 25
Cognition 26
Digital Rights 28
Design 29
Scope 30
Client Application Goals 30
Server Application Goals 38
Design Development 40
Application Programming Interface 40
Hardware 45

CHAPTER Page

Construction 47
Client Application 47
Initialization 47
Main Window 50
Episode Information and Transcripts 51
Episode Video 52
Mouse Navigation 53
Video Search 54
Settings and Preferences 55
Server Application (future) 56
Video 57
Transcripts 58
Models 60

IV IMPLEMENTATION AND RESULTS 62

Client Application 62
Models 62
Textures 64
Camera 64
Environment 65
Video 67
Transcripts 68
Preferences 69
Evaluation 75

V CONCLUSION AND FUTURE WORK 78

Implications for Future Research 78

NOTES 81

BIBLIOGRAPHY 84

APPENDIX A 88

VITA 89

LIST OF FIGURES

FIGURE Page

1 A sample view of the PNN application. 2

2 A view of the Web Stalker interface, showing a search of the CNN website. 8

3 Shredder reformats existing web pages, such as this one at jodi.org. 9

4 Netomat presents data as a flow of information, responsive to the viewer’s mouse input. 10

5 Live Wire installation view. 11

6 Overall view of EARTH visualization. 12

7 Typical Max/MSP patch or window. 14

8 A typical Jitter patch or window. 15

9 A typical Nato.0+55+3d patch. 16

10 Sample Auto-Illustrator window. 18

11 A preliminary schematic view of the PNN process. 37

12 Formac Studio DV/TV video capture device. 46

13 BOCA Labs TextGrabber GP500 closed caption device. 46

14 The initial startup dialog. 48

15 The Online/Offline selection dialog. 48

16 A frame from the PNN startup animation. 49

17 The main application window showing neutral textures. 50

FIGURE Page

18 View showing episode description and transcript display. 52

19 A bird's eye view showing an episode textured with video................ 53

20 The search dialog. 54

21 Simulated view of video search results. 55

22 A schematic diagram of the server process. 57

23 A schematic view of the video search process. 59

24 A schematic view of one possible model production methodology..... 61

25 The PNN Settings "Information" tab.. 70

26 The PNN Settings "Help" tab.. 71

27 The PNN Settings "View Settings" tab. 72

28 The PNN Settings "Corporate Settings" tab. 74

CHAPTER I

INTRODUCTION

OBJECTIVE

The art form of music has been a bellwether of external influences on culture. Inexpensive computer programs give artists of limited means the power to create music only large studios could have produced previously. Programs such as Pro Tools [1], Digital Performer [2], Reason [3], and Max/MSP [4] have directly influenced the kind of music popular today. These artists often produce work that is characterized by the term "sampled." Digital bits and pieces of previously recorded material, environmental sounds and heavily manipulated sonic matter are arranged to produce a new layer of meaning. This means of production is a reflection of the way our culture is produced. We take bits of information, often devoid of context, and assemble them into what is personally relevant. Increasingly, as a method of working, sampling has begun to influence the visual arts. While concepts of juxtaposition and assemblage have long been a part of the visual artist's lexicon, only recently have they begun to harness the power of digital technology to interpret and "sample" the visual world. In the visual arts, collage and assemblage juxtapose imagery to create a new sum from the parts. Marcel Duchamp and the Dadaists created work utilizing the technology of the early 1900s, and combined static imagery culled from newspapers, magazines, and photographs, or constructed sculpture from "readymade" objects. Sergei Eisenstein

This thesis follows the style and format of *Leonardo.*

treated film in a similar manner by developing the technique of montage. The advent of affordable digital sampling has given the visual artist greater access and control over this method of working than ever before. The primary distinction between collage/assemblage and sampling is the measure of granularity of access. Digital technology allows the artist control over the image at the microscopic level. This control is not limited to static imagery, but also applies to time-based media such as film and video. To date most computer applications have been concerned with solving utilitarian problems. Typical art projects have been self-reflexive in their use of the Internet, and have ignored the fact that our culture is often shaped by other forms of media. This project will design an application called Public News Network (PNN) (Fig. 1) that can be considered an interactive work of art.

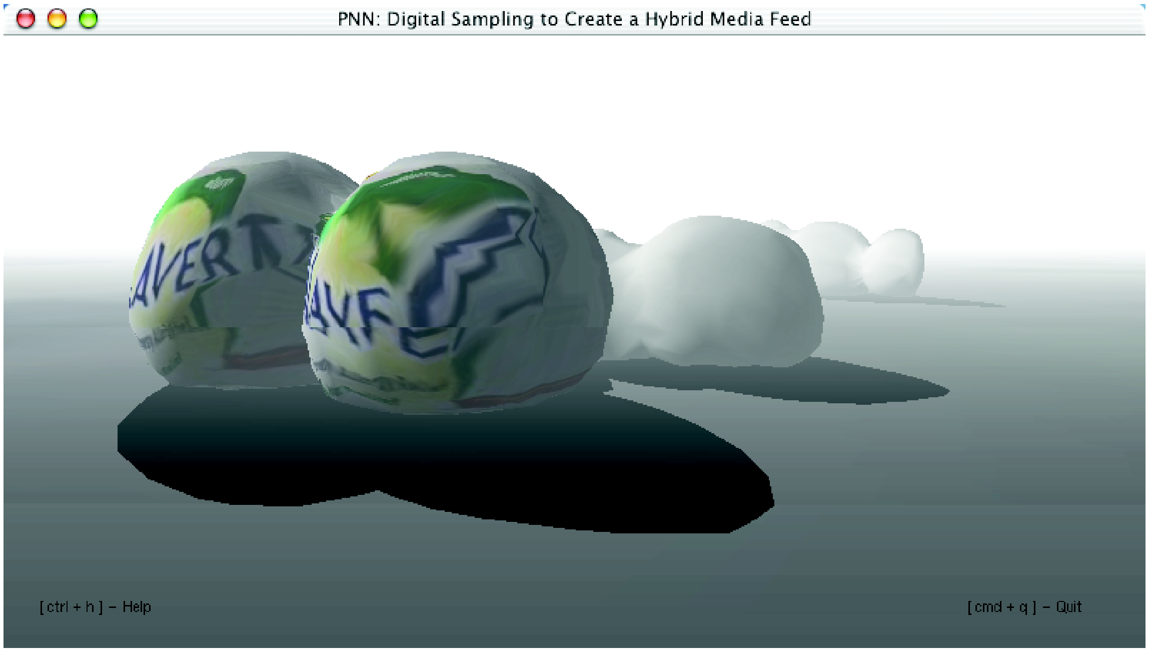


Fig. 1. A sample view of the PNN application.

It will be used to re-purpose existing cultural content that is not intrinsically tied to the Internet, and allow the user to manipulate and investigate the result. The application will serve as the future basis for multiple Internet enabled projects using similar methodology. The objective for this project is to create a framework for a computer application that utilizes sampling techniques to manipulate cultural data in order to produce an aesthetic experience.

FOUNDATION

When one hears the word "sampling" the immediate thing that comes to mind is "Hip Hop" music. Beginning in the early 1980s, with the advent of affordable electronic devices called "samplers," musicians could easily record and playback portions of previously recorded material. These segments, or audio samples, could readily be reassembled and manipulated to create a new song. The reality is that sampling has a long history that pre-dates Hip Hop. Throughout history songs, and musical structures have been passed from generation to generation. The essence of many forms of music we recognize today are defined by a historic set of chord progressions or rhythmic structures. Blues is often defined by a series of three chords. Even classical composers would build their compositions based on earlier work. In the late 1940s, a method of creating music from "found sound," manipulating tape loops to assemble, cut and shape sound was dubbed Musique Concrete. Composers such as Karlheinz Stockhausen, John Cage, and Steven Reich have since influenced generations of musicians.

The world of visual art also has a long history of copying from previous work. First there was Donatello's David, then Verrocchio's, then Michelangelo's, followed by Bernini's; all using the same cultural referent and similar structures. Picasso once said:

> *'What is an artist, basically? He's a collector who wants to establish a collection of his own, making the pictures himself that he likes by other people. That's how I begin, and then it turns into something else.'*
>
> and…
>
> *"To copy others is necessary, but to copy oneself is pathetic."*

Closely related to the idea of sampling is collage. Dada and Surrealism used collage to provoke subconscious associations which create or obscure meaning in the work. Rather than drawing exclusively from historical sources, cultural content was culled from magazines and newspapers of the day. Only upon reflection do the objects of these artists attain the artifact of nostalgia. At the time they were created, they were manipulating the news of the day. In the early 1980's, artists such as Sherrie Levine, Barbara Kruger, and Richard Prince utilized methods of appropriation. In 2001, Internet artist Michael Mandiberg created a website called AfterSherrieLevine.com [5] which re-appropriated the imagery that Levine had appropriated from Walker Evans in 1979. Today, artists use appropriation in their work to the extent that sampling other artists' work—much less cultural items such as television, newspapers, and magazines—is an accepted part of the artist's oeuvre. While copyright has a long-standing history in Western society, the advent of digital technology and the concomitant desire of artists to experiment with its promises presents new challenges. Fear of a lack of control over digital assets, and greed on behalf of modern corporations has begun to impinge on the artist's right to use culture as a basis for art making.

Another key component to PNN is the idea of "simulation." Throughout history the picture plane, or screen, has evolved from a distinctly two-dimensional display to the fully immersive environments of Virtual Reality (VR). Humanity seems to exhibit an innate desire to create and interact with environments that communicate a sense of the immersive. Oliver Grau, in *"Virtual Art: From Illusion to Immersion"*[6], traces the advent of simulation to Roman frescoes and large-scale murals created ca. 60 B.C. "Through the device of seeming to extend the wall surface beyond a single plane, the room appears larger than its actual size and draws the visitor's gaze into the painting, blurring distinctions between real space and image space." Fifteen hundred years later Renaissance art was defined by the invention of true perspective. The picture plane began to function not only to represent an idea but also to invite the viewer to enter a diegesis [7] with the content. In the 1830s the development of the screen evolved with the invention of photography. Suddenly the imagery presented was an "exact" reproduction of reality. The idea that the "camera does not lie" began to blur the distinction between reality and simulation. With the advent of cinema, or motion imagery, in the 1880s, and later with the addition of sound, the screen became a truly immersive environment. Especially in the case of theaters, the diegesis was heightened by the sheer enormity of the screen and the enclosed nature of the space. Though less immersive than cinema, television is an extension of this development of screen/space relationship. These technologies place the viewer in the role of "receiver" of information. Not until the development of the personal computer has the role of the viewer's relationship to the screen been challenged. The computer extends this idea of simulation by continuing the representation of screen space while adding a level of interactivity that transforms the viewer into more than a passive consumer. Through technological means artists can now engage the viewer more fully by creating a space of illusion that invites viewer participation. At this point in history, full Virtual Reality as expressed by CAVE (CAVE Automatic Virtual Environment), and other immersive environments are in their infancy and are not readily available to the masses. For this reason PNN is situated between cinema and full Virtual Reality in the realm of the computer display.

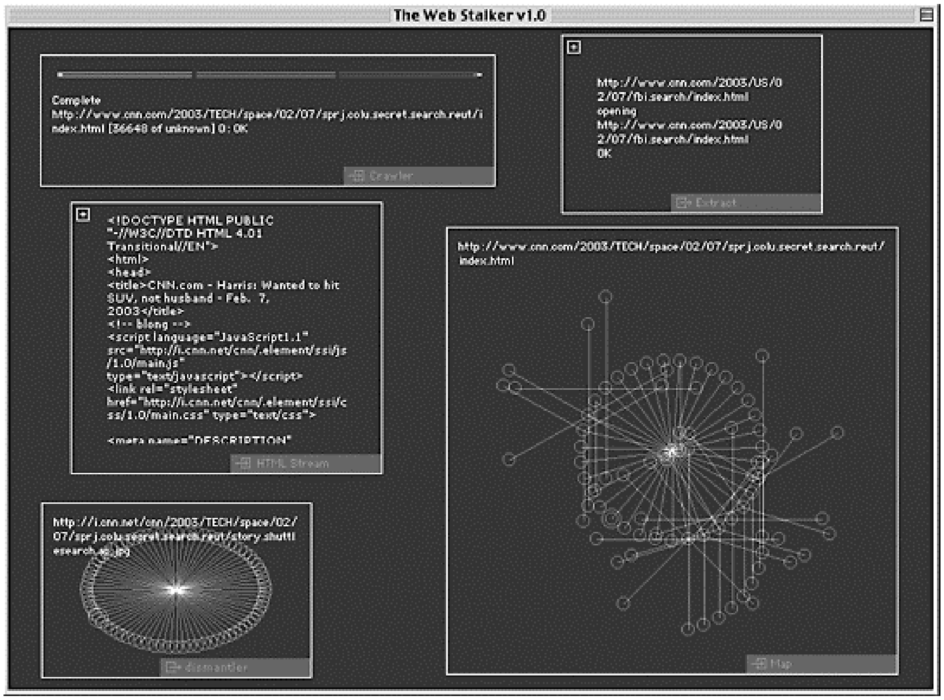
CHAPTER II

PREVIOUS WORK

Due to the relative youth of this new medium the precedence and context for the development of a project of this nature is limited. Generally work can be categorized into one of three types: artworks, art tools, and hybrids.

ARTWORKS

Internet art is usually created dependent on the intervening interface of a web browser such as Internet Explorer, or Netscape. Standard HTML (hypertext markup language) or DHTML (dynamic html), combined with scripting languages such as JavaScript or VBScript are integrated with proprietary plug-ins such as Macromedia Flash to create an aesthetic experience. No matter how it is accomplished the work is filtered through the window of the typical browser. In 1997 a collective from London called I/O/D (Simon Pope, Colin Green and Matthew Fuller) created a new web browser called Web Stalker [8] (Fig. 2). Web Stalker is considered the first "standalone" Internet art application. It does not rely on a web browser for display; rather, it is an alternative browser that presents information to the viewer, filtered through an interface designed by the artists, I/O/D.



```
Complete
http://www.cnn.com/2003/TECH/space/02/07/sprj.colu.secret.search.reut/i
ndex.html [36648 of unknown] 0 : OK
```

```
http://www.cnn.com/2003/US/0
2/07/fbi.search/index.html
opening
http://www.cnn.com/2003/US/0
2/07/fbi.search/index.html
OK
```

```
<!DOCTYPE HTML PUBLIC
"-//W3C//DTD HTML 4.01
Transitional//EN">
<html>
<head>
<title>CNN.com - Harris: Wanted to hit
SUV, not husband - Feb.  7,
2003</title>
<!-- blong -->
<script language="JavaScript1.1"
src="http://i.cnn.net/cnn/.element/ssi/js
/1.0/main.js"
type="text/javascript"></script>
<link rel="stylesheet"
href="http://i.cnn.net/cnn/.element/ssi/c
ss/1.0/main.css" type="text/css">

<meta name="DESCRIPTION"
```

```
http://www.cnn.com/2003/TECH/space/02/07/sprj.colu.secret.search.reut/
index.html
```

```
http://i.cnn.net/cnn/2003/TECH/space/02/
07/sprj.colu.secret.search.reut/story.shuttl
esearch.ap.jpg
```

Fig. 2. A view of the Web Stalker interface, showing a search of the CNN website.

Web Stalker presents an interface that allows the viewer to enter a search URL and view the information in one of several different modes. The application "crawls" the web site and creates a graph illustrating the linkages to other URLs it contains, as well as presenting the code behind particular links, their size, etc. While many consider Web Stalker a standalone application, it is actually based on the application Macromedia Director. The language used in its construction is called Lingo, and is in reality a scripting language, rather than a true programming language.

Another example of a digital sampling art project is Mark Napier's 1998 web browser, "Shredder" [9] (Fig. 3). Like Web Stalker, Shredder can be considered an alternative web browser - a new way of viewing existing content on the Internet.

Utilizing Shredder, the user can enter an Internet address and it pseudo-arbitrarily re-formats the content of the web page. Shredder appropriates the data of the web and creates an abstraction by processing standard html pages using JavaScript and dynamic html. It then displays the new formulation within a standard web browser, such as Internet Explorer or Netscape.

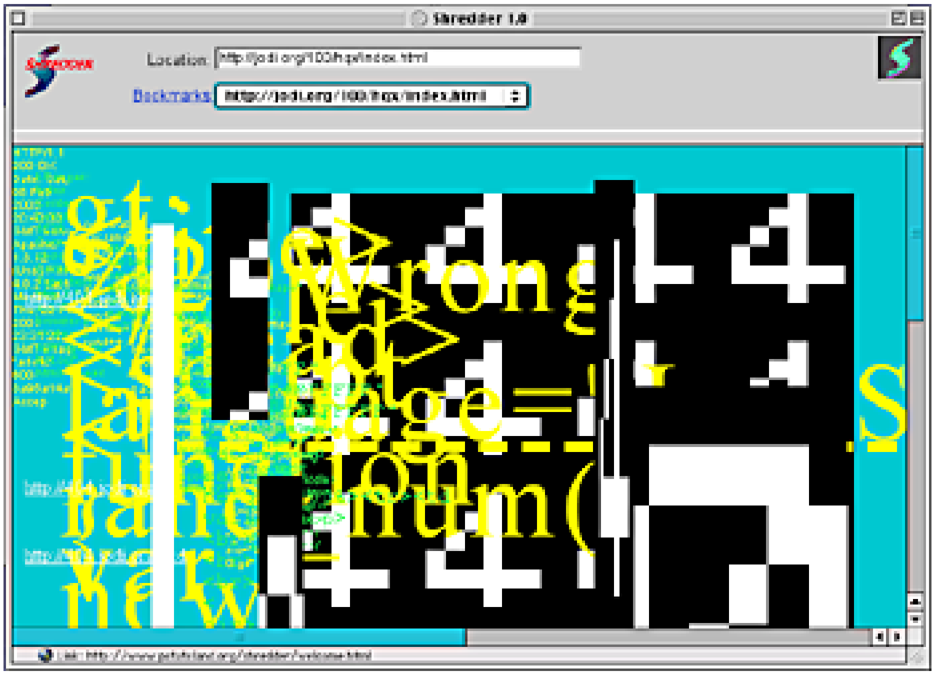
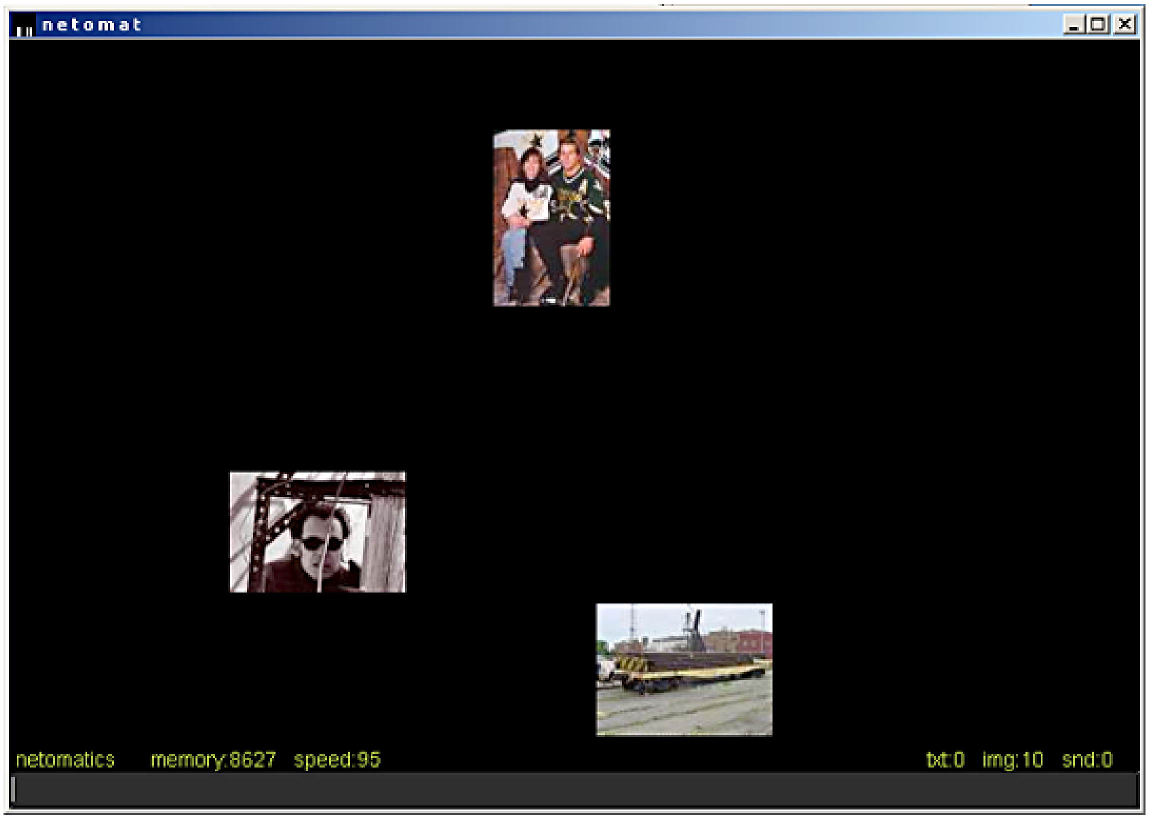

Shredder 1.0
Location: http://jodi.org/100/hqx/index.html
Bookmarks: http://jodi.org/100/hqx/index.html

Fig. 3. Shredder reformats existing web pages, such as this one at jodi.org.

While Shredder appears to be a web browser it is, in fact, reliant on the underlying browser for display and content. Shredder is essentially a web page that presents information using the interface afforded by the viewer's browser.

There have been numerous versions of the idea of an "alternative browser," but in 1999 Maciej Wisniewski advanced the idea by creating "Netomat" [10] (Fig. 4). This alternative browser is significant because it is a freestanding application that does not rely on a standard web browser for content and display. The user downloads and installs a Java application that accepts search strings and retrieves information from the Internet. Netomat presents the information as a “flow” of data across the user’s computer monitor. As the user moves her mouse the images and text traverse the axes of the display. Data is stored in a file format called a “netomatic,” which is a XML file containing URLs, keywords and presentation information.



netomatics memory:8627 speed:95 txt:0 img:10 snd:0

Fig. 4. Netomat presents data as a flow of information, responsive to the viewer’s mouse input.

Netomat enables the client to store, retrieve and share custom “netomatic” files. As is common among many sampling art projects, Netomat tends to be self-reflexive. Most of these projects’ sources are gathered exclusively from the Internet and therefore they have limited themselves to the manipulation, criticism, and presentation of the Internet itself.

In the early 1990s an artist in residence at Xerox PARC, Natalie Jeremijenko, created “Live Wire” [11] (Fig. 5). Live Wire is a physical manifestation of unseen data traffic as it flows through an Ethernet network. It consists of an 8 foot wire suspended from the ceiling, to which is attached a motorized interface to the local network.

Fig. 5. Live Wire installation view.

Live Wire continuously monitors the network traffic, and as the information flow increases, the wire jiggles faster in response. Live Wire is an attempt to provide a physical manifestation of what is culturally thought of as immaterial information. The artist illustrates the fact that the bits and bytes of the network can be represented in the physical world. Of interest to this researcher is the notion of dynamic information. Live Wire is always changing based on the data represented at a singular point it time.

More recently, in 2001, John Klima produced “EARTH” [12] (Fig. 6). EARTH is a visualization of Internet derived information mapped onto the surface of a simulated three-dimensional model of the Earth.

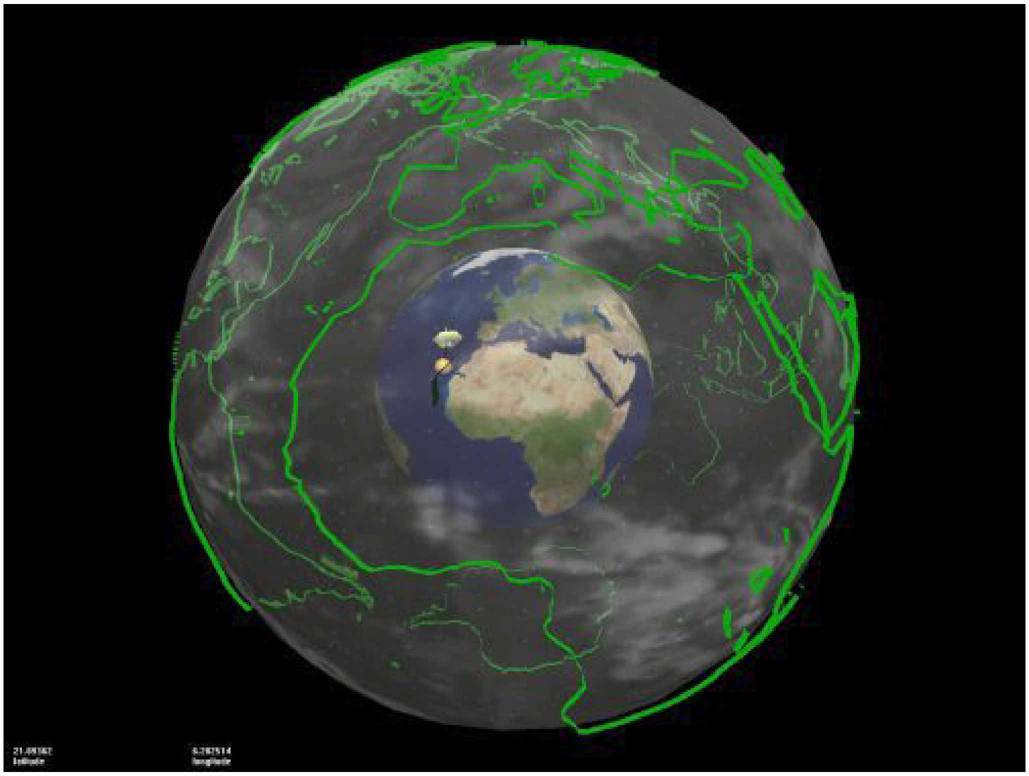
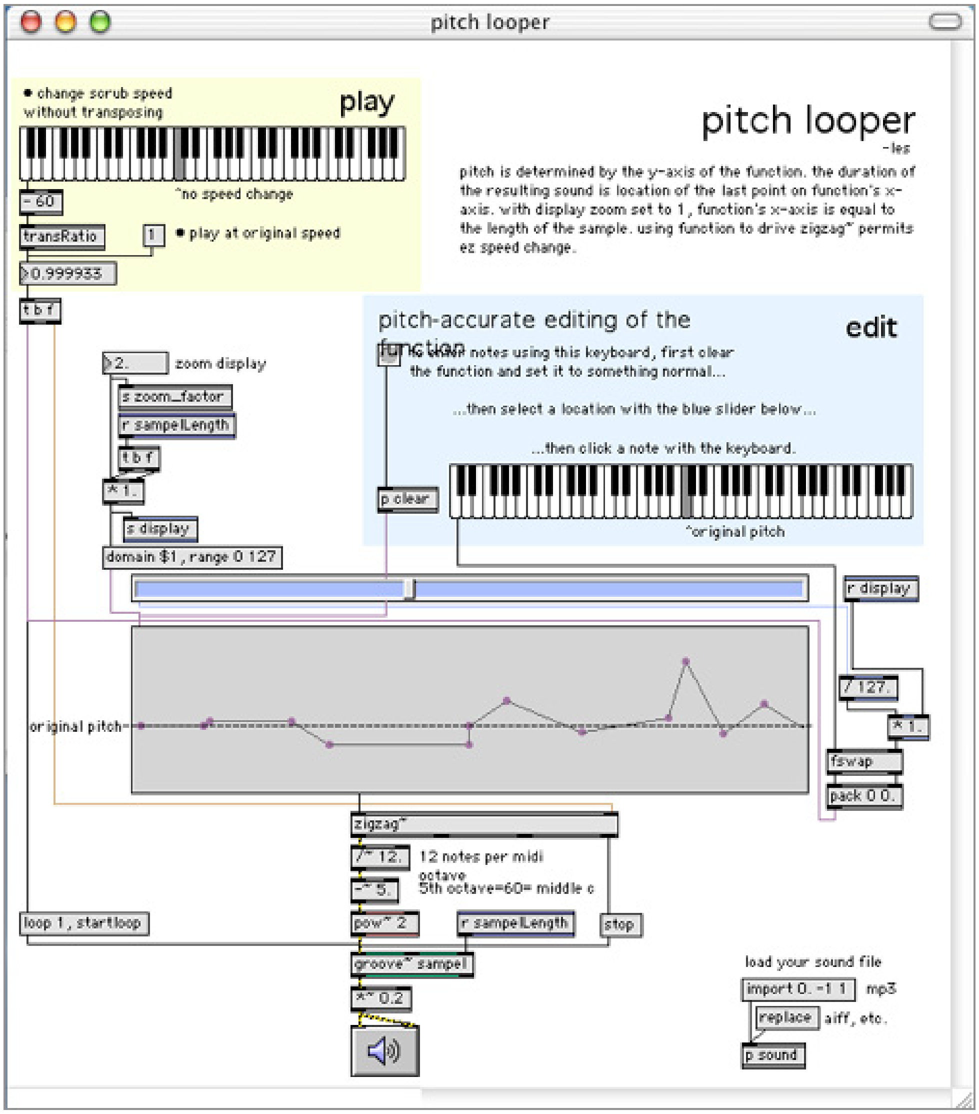


Fig. 6. Overall view of EARTH visualization.

By culling data from the United States Geological Survey, Military, and Weather information, the viewer's location is presented within the context of an image of the world. The globe is surrounded with layers of information that are organized from a macro to micro point of view and can be navigated with the mouse and keyboard. EARTH is interesting because, like Live Wire, it is an attempt to provide a comprehensive view of information that cannot be conceived of independently. EARTH provides a simulation of reality that gives form to data.

ART TOOLS

Art production tools have been commonplace from the inception of the color monitor. To date most applications are written with commerce in mind and generally provide some utilitarian or entertainment value. Typical graphics programs can be thought of as "tools" to produce other forms of graphics. Recently several art production tools have begun to blur the distinction between application as art and application as tool. Cycling '74's Max/MSP [13] (Fig. 7) is a graphical programming environment that is commonly used to produce electro acoustic music. Max/MSP works with the metaphor of chained objects to allow artists to serially link programmatic functions using patch chords. Many of these objects have a direct relation to C/C++ counterparts.



● change scrub speed without transposing

^no speed change

● play at original speed

pitch looper

-les

pitch is determined by the y-axis of the function. the duration of the resulting sound is location of the last point on function's x-axis. with display zoom set to 1, function's x-axis is equal to the length of the sample. using function to drive zigzag~ permits ez speed change.

pitch-accurate editing of the function

to enter notes using this keyboard, first clear the function and set it to something normal...

...then select a location with the blue slider below...

...then click a note with the keyboard.

^original pitch

zoom display

12 notes per midi octave

5th octave=60= middle c

load your sound file

mp3

aiff, etc.

Fig. 7. Typical Max/MSP patch or window.

The Max portion of Max/MSP is concerned with MIDI data and its manipulation, while MSP works with audio data such as sound files. The standard Max package also comes with a limited set of graphics capabilities. Because the graphics and audio are integrated in one package, it is easy to create graphics utilizing audio data, and vice versa. Another strength of Max/MSP is its integration with hardware. Most any parameter in Max/MSP can be controlled via external devices connected to the host computer's serial or USB port. This makes Max/MSP a favorite application for installation artists and others who need an interface between the analog and digital worlds. In 2002, Cycling '74 introduced Jitter (Fig. 8). Jitter is an extension to Max that emphasizes graphic production. It allows control over video and has some three-dimensional graphics support.

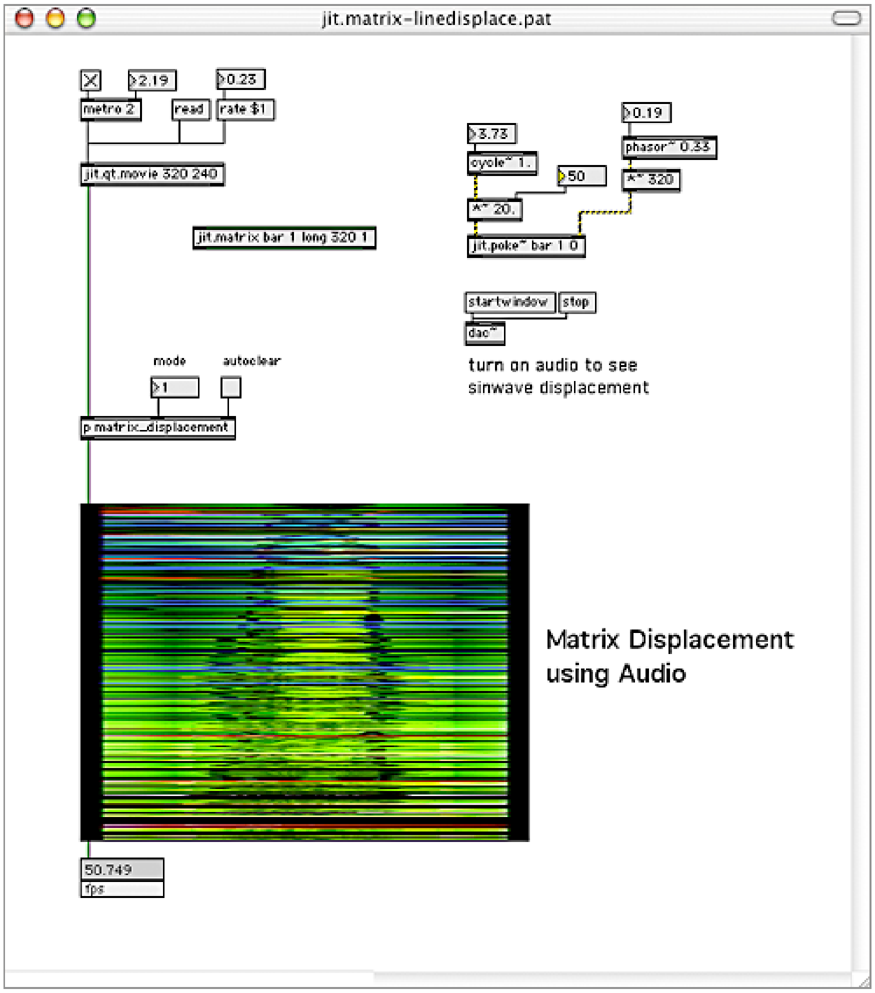


Fig. 8. A typical Jitter patch or window.

The interesting thing about Max/MSP is the fact that it has brought the sampling paradigm into the realm of computer graphics by mixing the tools of music sampling with those of imagery manipulation. While Jitter was introduced in 2002, the genesis of Jitter is another extension to Max written by Netochka Nezvanova in 2000.

Nato.0+55+3d [14] (Fig. 9) provides sample level control over video as well as static imagery. This node based programming environment allows the programmer to create applications that cull content from the Internet, as well as, to process audio and video input.

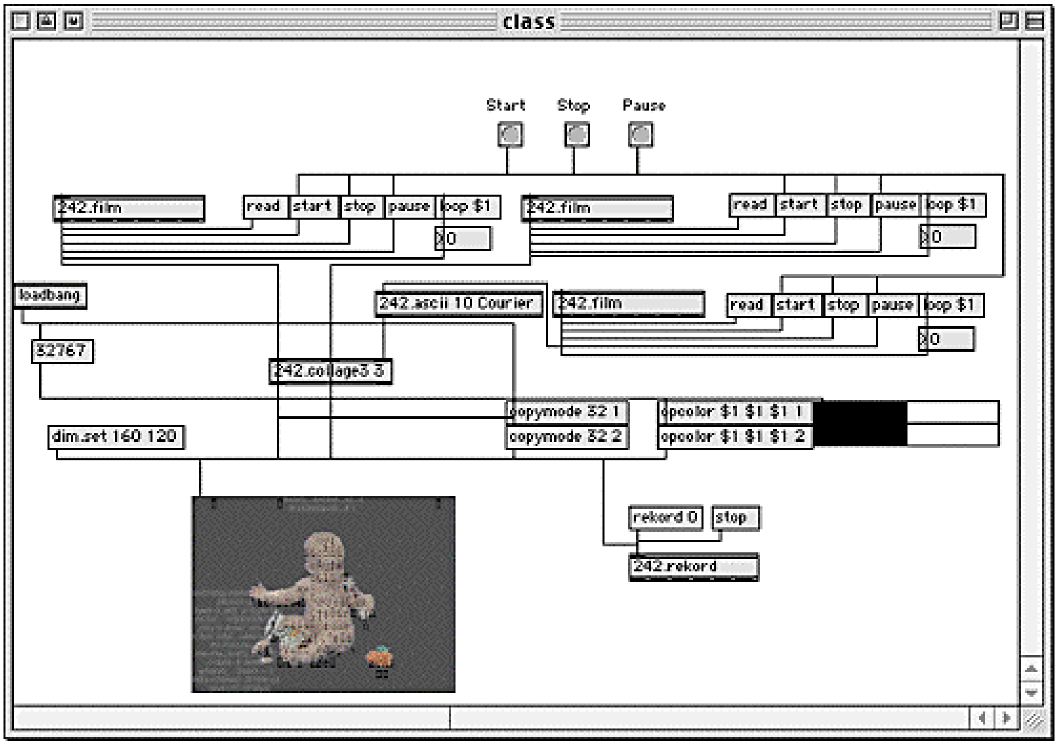
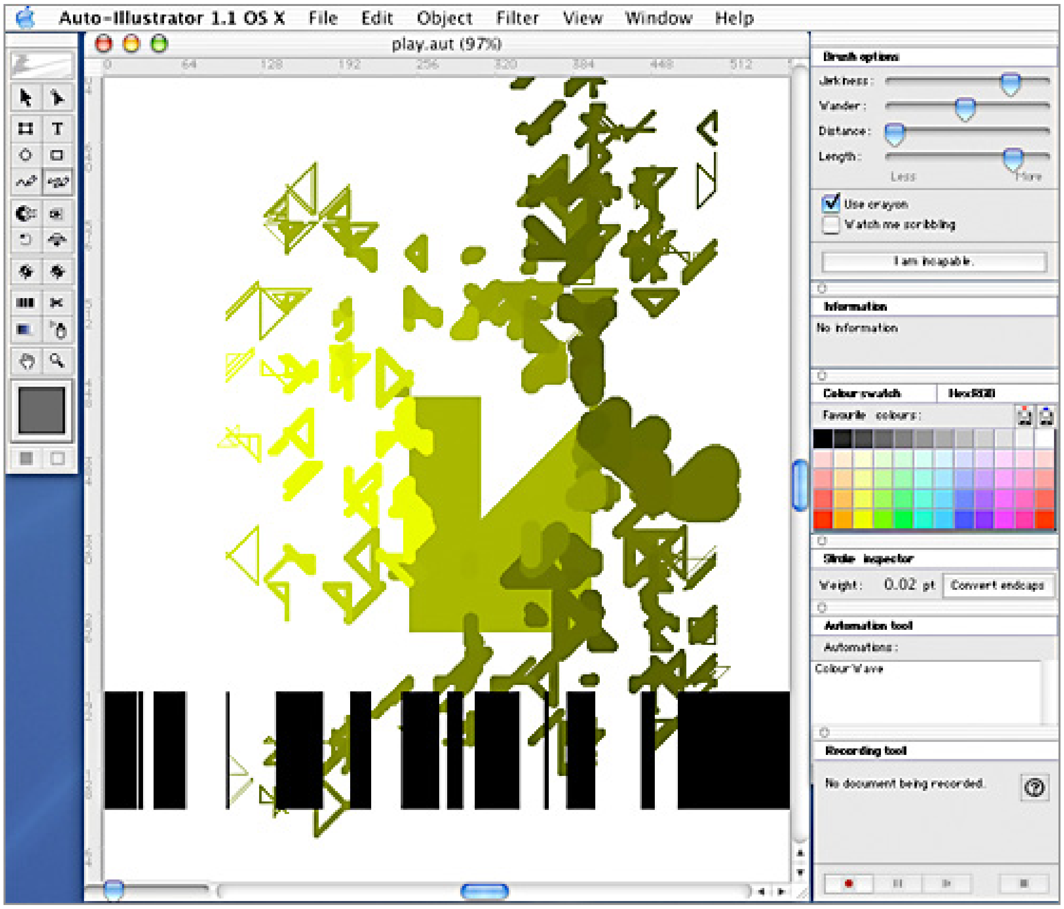


Fig. 9. A typical Nato.0+55+3d patch.

The package consists of over 150 additional externals (objects) for Max, many of which are optimized for real-time execution. Of interest to this researcher, with regard to Nato, is the blurring of distinction between software tool and art. With Nato, the creator considers the software to be a work of art. It embodies the idiosyncrasies of the author to the point that many are "put off" by the compromises involved. For example, the help documentation is written in a pseudo "hackeresque" style. A sort of "Eastern European" bad girl attitude is infused with the software and supporting website. In reality, Netochka Nezvanova is a pseudonym that translates loosely as "nameless nobody." NN, as she refers to herself, is famous for her ASCII laced, hacker-like emails to many of the web art and technology mailing lists. At the base level, though Nato is still essentially an art-making tool as opposed to an artwork.

In 2001, a program called Signwave Auto-Illustrator [15] (Fig. 10) shared the Transmediale Artistic Software award with Nato. Auto-Illustrator is a program similar to Adobe Illustrator, which produces vector graphics. The difference, though, is that Auto-Illustrator is more experimental in the way it responds to the user. Auto-Illustrator uses procedural, generative techniques to modify user input. With this software the output is truly the result of the collaboration of man and machine. Despite these advances, Auto-Illustrator and Nato are still more closely aligned with a graphic production environment than with an autonomous work of art.



Jerkiness:
Wander:
Distance:
Length:

☑ Use crayon
□ Watch me scribbling

No information

Favourite colours:

Weight: 0.02 pt

Automations:
Colour Wave

No document being recorded.

Fig. 10. Sample Auto-Illustrator window.

CHAPTER III

METHODOLOGY

In this research the author used a methodology taken from the world of architecture. This methodology is characterized by four different stages: Concept, Design, Design Development, and Construction. The Concept phase is an attempt to identify issues of relevance to the topic. These issues establish priorities that guide the Design phase. During the Design phase project goals are established and the general scope of development is determined. In Design Development the researcher identifies tools necessary to complete the project and solidifies the choices made. The Construction phase outlines the actual components necessary to implement the design. Detailed discussion of the implementation of the Construction components will occur in Chapter IV, Implementation and Results.

CONCEPT

A number of concepts informed the development of PNN. These concepts were drawn from investigations into the disciplines of art theory, cognitive science, and law. The creative process of this artist is not to first research and identify theories from which to develop the work. Instead this artist seeks to maintain awareness and continually develop his knowledge of contemporary cultural issues. This knowledge, then, allows the artist to recognize “synchronicities” during the idea generation process, and allows one to selectively strengthen or weaken ideas as the project proceeds. In this manner the theory provides a framework that informs rather than dominates the end result.

From art theory, research into the concepts of representation and its relationship to digital simulation situate the work within the current critical discourse of "the screen." The simulation capabilities of PNN are then discussed in relation to the theoretical discourse of the Situationist International and the dérive. Following the discussion of the dérive, notions of collage and assemblage and the semiotic creation of new meaning via sampling are explored. To this researcher, digital simulation combined with sampling presents new opportunities to experiment with the idea of the dérive. The final concept taken from Art Theory is that of dynamism. How can PNN take advantage of the dynamic nature of the Internet to enhance the viewer's experience?

From cognitive science the proposition that we assemble meaning dynamically, much like an object oriented programming language encouraged this researcher to find ways to strengthen this inherent tendency. Similarly research in cognitive science suggests that many people interpret stories from news broadcasts independent of the facts presented in the broadcast itself. This finding, too, suggests that artistic principles such as collage, juxtaposition and sampling might be methods to encourage or subvert meaning.

The concepts that inform PNN are completed with a discussion of digital rights and copyright. The instability of law with regard to this new medium has important ramifications for the design of a project such as PNN. In this section, a solution is offered that places responsibility on the ethics of the viewer.

Simulation

Simply put, a simulation is an attempt to make one thing look like another. A simulation, as opposed to a representation, though, is distinctly different in that simulation implies a closer relationship to the original than a mere representation. The major portion of art history was concerned with man's representation of his world. But as Oliver Grau traces Virtual Reality back to the Roman frescoes of 60 B.C., it is obvious that mankind has always been driven by a deeper desire to simulate reality.

Lev Manovich, in *"The Language of New Media"* [16] describes this desire in terms of the "screen." He identifies four types of screens and positions them within our history. First there was the classical screen. The classical screen is a rectangular view port that is essentially static. Traditional paintings and photographs are emblematic of classical screens. The next type of screen is the dynamic screen. The dynamic screen contains moving imagery of the past, such as cinema. The third type of screen is the real-time screen. This screen contains moving imagery of the present. Often cited as one example of this type of screen is that of the television. The final screen type is the interactive screen. This is the screen of the computer. According to Manovich the ultimate screen is not really a screen at all, but the dissolution of the screen, Virtual Reality. The result of this inexorable journey towards Virtual Reality is an existence free of our bodies. Many theorists surmise that our innate desire for virtual existence is a result of our desire to be free from the limitations imposed by our bodies. If our existence is not reliant on the frailty of the human body, we are essentially immortal.

These theories are interesting in that they explain the motivations of history and where those motivations might take us. From the viewpoint of the artist, and in particular, PNN, the desire to present this project in the form of an interactive virtual reality has to do with focus. An environment that is stripped of superfluous stimuli is more likely to focus attention on the visual attributes and concepts that are being presented. This researcher desires to engender an aesthetic experience that evaluates the visual environment encumbered with as little "noise" as possible. While PNN is not a "screenless" VR experience, the diegesis created by the computer screen and 3-D environment will hopefully create a disembodied experience that maximizes the opportunity for aesthetic response.

During the 1950's one group of cultural theoreticians known as The Situationist International (SI) described an experience where meaning was derived from surrounding environmental conditions. The SI was concerned with the disparity between ideology and the reality of meaning in society. In order to expose these disparities they would create "situations." A "situation" was the staging of an activity that would provide an opportunity for "détournment," or "turning around." Détournment is that moment when the individual, in a moment of clarity, "thinks for herself." In 1958 Guy Debord described the Situationist International concept of dérive [17]. Translated as "drifting," dérive is "a technique of rapid passage through varied ambiances. Dérives involve playful-constructive behavior and awareness of psychogeographical effects, and are thus quite different from the classic notions of journey or stroll." The typical dérive was a stroll through an urban environment with an open-minded attitude of discovery and investigation. The goal was to experience the spatial environment allowing chance to interject new meaning. In this manner preconceived notions could be "turned on their heads." It is this type of space this researcher wishes to create with PNN. PNN will create an architectural simulation that will function as a "situation." This situation will provide and opportunity for the viewer to experience "détournment."

Sampling

Sampling is in some respects a postmodern method which procures information from diverse sources and culls material from existing data. Rappers sample historic rock riffs and Techno artists sample obsolete synthesizer sounds to create new meaning from the old. In the visual arts, sampling is an extension of the idea of collage or montage. Art critic Gregory Ulmer [18] claims "collage is the single most revolutionary formal innovation in artistic representation to occur in our century." Collage in the visual arts had its genesis with the cubist abstractions of Picasso and Braque, and Sergei Eisenstein developed the filmic concept of montage. As a means of art production the function of collage in art is well described by the semiotic description of the "sign." According to semiotics, the "sign" is composed of "signifier" and "signified." The signifier is that which carries meaning, while the signified is the meaning that is carried. The important notion is that the signifier is not essentially related to the signified. The word "green" is not intrinsically green, much like "verde" is not intrinsically green; they are constructs of our culture and the meanings associated with words are therefore suspect. There is a rupture in our language between the representation of objects and the meaning we attribute to those objects. This arbitrary nature of language can be extended to the visual construction of meaning as well. Images removed from context often take on an altered meaning. This concept has had profound effect on the development of Western art throughout the late 20th Century. By re-combining and re-contextualizing imagery, new meaning and associations can be generated. With the development of digital sampling the scope with which artists can experiment with the signs of culture is greatly expanded. It is the desire of this researcher that PNN work with the idea of re-contextualized imagery through the process of sampling. PNN will work with these ideas by extracting imagery from a common source of cultural content in order to provide the viewer with an opportunity to create new meaning.

One means with which digital technology has empowered the artist to experiment with these ideas is via the rapid cataloguing and retrieval of data. Databases are a key component of many business-oriented applications, and databases drive much of the Internet. In fact, in some respects, the Internet itself can be thought of as a global database with random linkages forming a whole. With regard to sampling, the database provides the ability to categorize and quickly retrieve vast amounts of information. A database with non-linear access capitalizes on the strengths of computer technology, while conceptually reinforcing the tendency to sample information in society.

Dynamism

Especially with regard to the Internet, computer technology allows us to expand the idea of sampling to include dynamism. Instead of limiting the source of material to historic data we can take advantage of the "instant" connectivity of the Internet to provide dynamically updated content. In contrast to creating artworks as static displays, technology allows us to create artworks that respond to input, in which content changes with time. These characteristics might allow the artist to create work that more successfully engages the audience and maximize the viewer's ability to create new meaning.

In *"The Language of New Media"* [19], Lev Manovich discusses databases and navigable space as key forms of new media. In particular Manovich describes navigable space as unique to new media by virtue of it becoming its own media type:

> *"From one point of view, navigable space can legitimately be seen as a particular kind of an interface to a database, and thus something that does not deserve special focus. I would like, however, to think of it also as a cultural form in its own right, not only because of its prominence across the new media landscape and, as we will later see, its persistence in new media history, but also because, more than a database, it is a new form that may be unique to new media. Of course, both the organization of space and its use to represent or visualize something else have always been a fundamental part of human culture. Architecture and ancient mnemonics, city planning and diagramming, geometry and topology, are just some of the disciples and techniques that were developed to harness space's symbolic and economic capital. Spatial constructions in new media draw on all these existing traditions – but they are also fundamentally different in one key respect. For the first time, space becomes a media type. Just as other media types – audio video stills, and text – it can now be instantly transmitted, stored, and retrieved; compressed, reformatted, streamed, filtered, computed, programmed, and interacted with. In other words, all operations that are possible with media as a result of its conversion to computer data can also now apply to representations of 3-D space."*

These thoughts led to the idea that we might be able to transmit shapes via the Internet and in turn dynamically create navigable space on the viewer's computer. This researcher's experience as an architect has taught him that space is often more readily defined by the objects placed within it, than by boundaries such as walls and ceilings. What kind of space could be created that would link ideas of sampling, dynamism and navigable space?

Cognition

Often the way meaning is created is different than our expectations. Just as semiotics teaches us the opportunity for dichotomy between signed and signified in language, cognitive science helps us understand the disparity between imagery and our interpretations of that imagery. These investigations can assist the artist to evaluate the world with a fresh perspective. While researching topics for this thesis, this researcher's interest was piqued by several ideas from the world of cognitive science that seemed to coincide with ideas that inform the world of digital art. In an article, *"Object-Oriented Models of Cognitive Processing"* [20], researchers describe the relationship between our visual cognitive awareness and object oriented programming languages. The researchers describe how we process visual imagery in a somewhat haphazard fashion rather than a serial fashion. Like an object-oriented program, messages are passed from object to object and there are no strong links or threads of logic that tie separate objects together. Connections are made dynamically, "on-the-fly." In opposition to this are procedural programming languages that process information in a linear or serial manner. One bit of information is processed before being passed to the next function, which subsequently passes the information on to the next function. Instead, our minds are efficient enough to make simultaneous intelligent linkages to whichever portion of the brain is necessary. Tangent to this idea is the idea of "Story Elaboration." In an article entitled *"The Rest of the Story: Sociocultural Patterns of Story Elaboration,"* by Karen Cerulo [21], researchers describe how the nature of story telling is fundamentally changing in society. Historically our information was gathered in a somewhat linear fashion via newspapers and magazines. Today the majority of Americans procure information via the television news. The nature of these mediums differs especially in terms of the depth of coverage. Information presented via the evening news is aptly described as a "newsbyte." In addition to the size of information we use to create meaning, the researchers found that many individuals tend to ignore the facts of a story and create meaning by adding bits and pieces of other stories or previous knowledge. The idea of "Story Elaboration" suggested that television news might be a good source of material for this project. The synergy between the concept of viewers creating new stories from portions of broadcasts combined with the practice of digital sampling is aesthetically stimulating to this researcher. In response to this conceptual relationship, this researcher began to think of ways to reinforce those connections. The combination of object oriented cognition and story elaboration have interesting implications with regard to art, and in particular, sampling and meaning. Is there a way to capitalize on our cultural momentum and innate tendencies in order to produce an artwork that maximizes the ability to create new meaning?

Digital Rights

Like many things in life, a space that embodies the concepts of digital sampling and dynamic content must address a society that has allowed corporate interests to usurp individual expression. Copyright originated in England and was a means of censorship, granting monopoly to the publishing industry. The few who owned printing presses were given the power to seek and destroy illegal presses and books. In 1710 the law was rewritten with the goal of encouraging public learning. It was re-defined as an "anti-censorship" law and individual authors were given the right to claim copyright on their own work. In the United States copyright was integrated into our system with the goal of protecting individual copyright holders for a length of time that they could benefit from their contribution to society. The first U.S. Copyright Act was called, "An Act for the Encouragement of Learning." Not only was copyright intended to reward copyright holders, but also to integrate these contributions into the public domain as quickly as possible.

> *"Copyright law is an odd bird. It establishes a strange sort of property when compared with other property... Copyright is a balance between expression that the owners can control, and expression that is left open to the commons. There is a commons for intellectual property, and the Constitution is committed to feeding that commons. For the commons is a resource for other creators later on. And the commitment of the Constitution is that there be lots later on."*
>
> Lawrence Lessig [22]

Never was it intended that corporations would copyright individual's work for resale and then maintain that copyright for generations. It also was never intended that elements of free speech, nor everyday contributions to our societal dialog such as radio and television, should be so closely guarded that merely referencing them in a artwork would raise the specter of copyright violation. Questions concerning the interpretation and application of copyright law with respect to this project are an interesting component of the artwork itself. In the circumstance where the intention behind sampling a part of our cultural dialog is a commentary on the role it plays in our constitution, and in consideration of the fact that there is not a profit motive, this researcher believes that the issue must be addressed in the work, but not allow it to commandeer the aesthetic goals. Digital Rights Management (DRM) and existing copyright law are beginning to strangle the creativity, inventiveness, and freedom that once made the United States a leading innovator. This researcher believes that this is an issue of morality, where the individual must decide what is right, and the design of the application should address the issue from that standpoint. This project will find a way to address the issue of copyright with regard to sampling and dynamic content generation.

DESIGN

Once concepts of interest have been identified, the next step is to begin to design the project. Client application and server application goals must be established and preliminary thought must be given to the implementation of these goals in order to create a project that succeeds not only technically but also artistically. Because the server application is not the primary focus of this research, it will not be covered with as much detail as the client application.

Scope

During the development of this thesis, it became apparent that the scope of work required to address the conceptual objectives exceeded the time allotment available. It became apparent it was necessary to break the project into three major components: the client-side application, the server-side application, and a web site. This researcher decided to concentrate on the implementation of the client-side application for this project, because it is key in determining what function will be required of the complementary components. The client-side application is responsible for the presentation and delivery of information to the viewer. The server-side application is responsible for preparing, organizing, and delivering information. The website is responsible for providing an initial interface between the viewer and the artist. Notice that there is an overlap in the functionality of the client and server components regarding delivery of information. Some functions of the programs rely on the existence of the other program for completion. In these circumstances, and where the actual implementation of the complementary program is uncertain, the functionality of the client-side application will simulate the final result. As it turned out this was a fortunate decision due to the number of changes made throughout the development process.

Client Application Goals

The goals established for the client application will provide a framework against which decisions regarding the aesthetic and functional attributes of the program can be evaluated. These goals are general enough to provide flexibility in the event that portions of the project must change. This researcher identified seven major client application design goals:

1. The client application must be cross-platform.
2. The client application must be scalable.
3. The client application must be navigable.
4. The client application must display cultural data.
5. The client application must be able to dynamically retrieve cultural data.
6. The client application must provide a way of sampling cultural data.
7. The client application must maximize the opportunity for "new meaning."

A primary design goal is the creation of a cross-platform application. A cross-platform application is an application that will successfully run on multiple computer operating systems. Because the client-side application will be available for download from the Public News Network web site, there is very little control over the platform choices of potential viewers. If designed to accommodate multiple platforms PNN will have a much broader client base and will be accessible by more viewers.

In addition to cross-platform functionality, the application needs to be scalable. Because the application is symbiotic with other applications and multiple operating systems, it needs to be easily updated and modified to anticipate future changes. If PNN is designed with scalability in mind it will be easier to maintain its functionality as computer hardware and software changes and will be better able to take advantage of beneficial developments within the computer industry.

Another primary requirement is that PNN present a navigable space. Availability of resources and the desire for public accessibility suggested that a full Virtual Reality system would not be appropriate for this project, so this researcher began to think of alternative forms that provide a level of simulation. A Computer Aided Design architectural “fly through” was one option, and pre-rendered scenery output from a 3-D package such as Maya [23] was another. Either of these might have addressed the aesthetic issues and might have been able to produce a video that dealt with issues such as sampling and cultural reference. Either of these might successfully communicate a sense of simulation, but neither would allow the level of dynamism and real-time navigability this researcher envisioned. Another option was to use the techniques common to Machinima [24] to produce this project. Machinima is a “movement” whose adherents use open source video game software to produce movies and video shorts. These artists use the 3-D video game engines that drive games such as Quake and Unreal to produce personalized movies and shorts. Initially this appealed to this researcher because of the immersive real-time nature of video game technology, as well as the conceptual relationship one could draw between the idea of sampling culture, and Machinima’s sampling of existing computer software. Upon further investigation it appeared that Machinima was more suited to video output rather than interactive navigation. This researcher decided that dealing with the constraints inherent in adapting existing software would needlessly restrict the development of this project, especially in regard to network access. Machinima did suggest that video game technology should be further investigated. Eventually this researcher decided that, indeed, the form of the contemporary video game best met the conceptual requirement to provide an immersive, navigable experience.

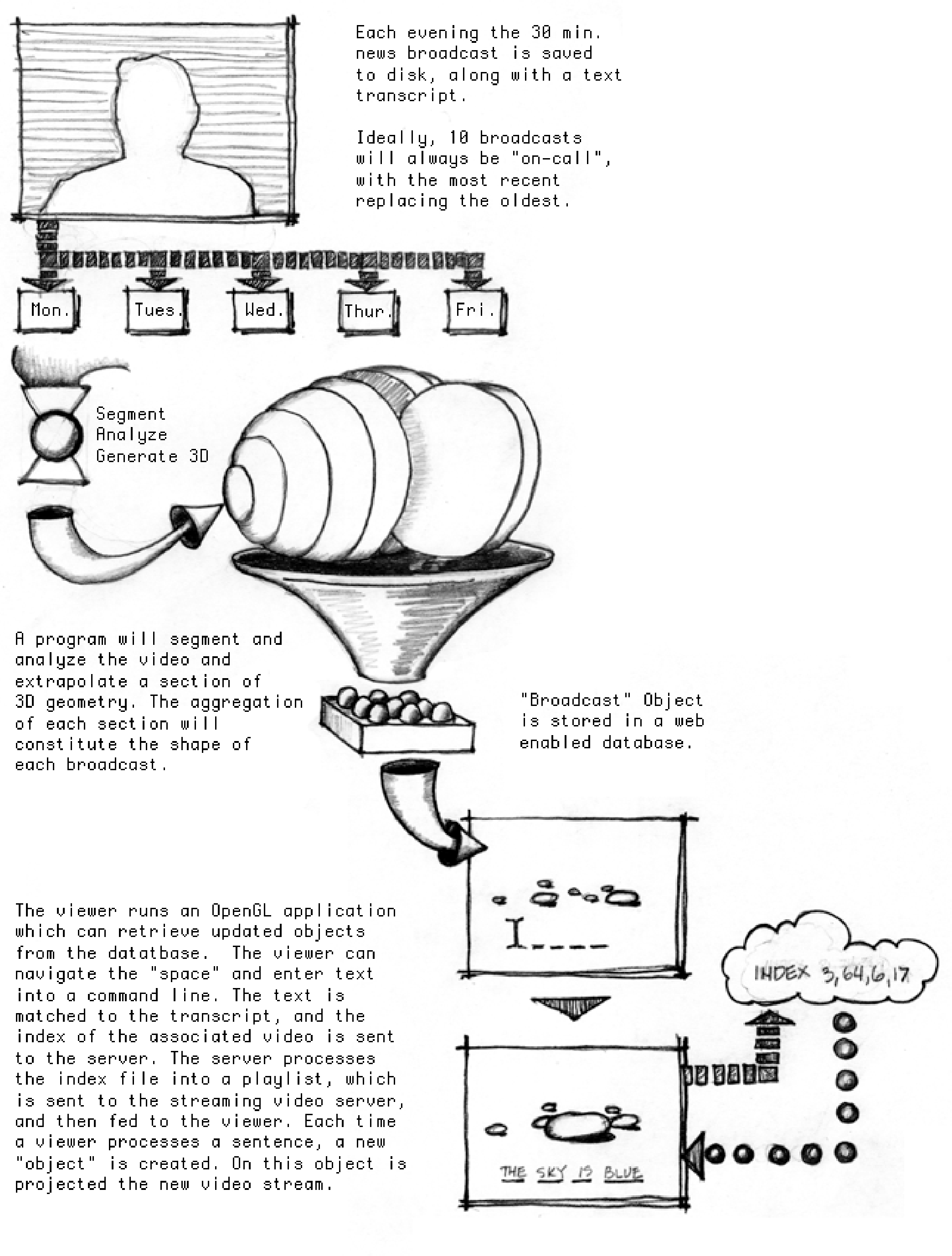
Once the decision was made to use video game methodologies the next primary design goal was to consider the integration of cultural data. What form should the data take, and how should it be presented? An issue that flows from the idea of sampling is context. When artifacts are removed from their context and placed in another, what happens? What new relationships are formed when imagery is re-contextualized? What happens when imagery is de-contextualized? In this context, re-contextualization focuses on the relationship between images and their juxtaposition. With respect to PNN this would equate to "What happens when a segment of Thursday's news is placed next to Monday's news?" De-contextualization focuses on the result of imagery removed from its original context. For PNN this equates to "What happens when the news is no longer packaged as a 30 minute program displayed on a box in the living room?" If PNN were to deal exclusively with re-contextualization it might simply juxtapose video segments in a manner that mimics the common presentation of a television screen. For example, the imagery might be presented full screen so the computer monitor becomes a substitute for the television. Alternately, the imagery might be presented composed within a mock television frame. This researcher decided that PNN should work with both re-contextualization and de-contextualization because both of those issues feed the concepts of object oriented cognition and story elaboration. This lead to the idea of creating a space which functions as a repository, or warehouse of cultural artifacts divorced from their original context. As mentioned previously this researcher decided that in order to create a sense of space, objects would be the defining elements, rather than walls or ceilings. It seemed appropriate to create a vista of news object contained within an apparently infinite space. By creating a minimal, stark space, attention will be focused equally on the formal aspects of the video image as well as the content generated by new segment associations. This will create a space maximized to encourage the generation of new meaning on behalf of the viewer.

In order to create this spatial vista the news objects will be 3-D models, each representing an episode of the evening news. In order to maintain a responsive user interface on the viewer's computer, PNN will display the 10 most recent episodes. While there were several options to present the cultural data (video broadcast) this researcher decided that texture mapping the surface of each 3-D model with video taken from that day's broadcast most effectively re-contextualized, and de-contextualized the data. The shape of each of the episode models will be generated by the server-side application via an analysis of the video luminance and color characteristics. In this manner there is a mapping of data from one form (2-D) to another (3-D). This action has artistic relevance because it is a mirror of the activity of analog to digital conversion, as well as the conversion of meaning from institutional to individual. This analysis and model generation will take place each night following the recording of the nightly broadcast. After downloading the application from the PNN web site and installing it on her computer, she will launch the PNN client-side application. Upon launch the application will download the latest episode models from the server. This communication with the server will assure that PNN is always dynamically updated to represent the current social climate as represented by the evening news. These objects, then, will define a space the viewer can navigate via the mouse and keyboard.

The next primary goal is to provide a facility to retrieve cultural data. Cultural data, in this case, video, will be cataloged in a database housed on the PNN server. Every evening at 5:30 PM Central Standard Time, the CBS Evening News with Dan Rather will be recorded. A broadcast from any of the networks could have been chosen but the CBS Evening News was chosen because of CBS's reputation as the iconic American network news organization. This researcher could have chosen to programmatically record news broadcasts from multiple sources but decided that constraining the selection to one network might reveal patterns that would go otherwise unnoticed. The video will be prepared by the server-side application on a nightly basis, so that it can be readily accessed and streamed from the PNN server, directly to the viewer's PNN client-application.

The PNN application must also satisfy the goal to allow the viewer to sample this digital information. This will be facilitated by providing an area for the viewer to enter search terms that will be matched against a transcript of each day's broadcast. When the viewer enters search terms, video segments containing the search terms will be stitched together to form a new video representing the search. This new video will then be mapped to a new episode model and will be presented in front of the viewer.

The final goal is to design the application is such a way that it maximizes the opportunity for the viewer to produce "new meaning." This will require a balance between the needs of multiple processes required to execute the end result. The processes require the coordination of a great deal of custom software and hardware. Fig. 11 shows an early design schematic of the flow of control required to execute this concept. Despite the complexity, this researcher believes that if the previous six design objectives are met this methodology will produce a computer application that satisfies this project's objective to create an aesthetic response in the viewer.



Each evening the 30 min. news broadcast is saved to disk, along with a text transcript.

Ideally, 10 broadcasts will always be "on-call", with the most recent replacing the oldest.

A program will segment and analyze the video and extrapolate a section of 3D geometry. The aggregation of each section will constitute the shape of each broadcast.

"Broadcast" Object is stored in a web enabled database.

The viewer runs an OpenGL application which can retrieve updated objects from the datatbase. The viewer can navigate the "space" and enter text into a command line. The text is matched to the transcript, and the index of the associated video is sent to the server. The server processes the index file into a playlist, which is sent to the streaming video server, and then fed to the viewer. Each time a viewer processes a sentence, a new "object" is created. On this object is projected the new video stream.

Fig. 11. A preliminary schematic view of the PNN process.

Server Application Goals

Despite the server applications status as a future development, specific goals must be established for its behavior in order to successfully design the client application. The more concrete these goals the less likely it will be that major client application changes will be required. At this preliminary stage it is anticipated that what is called the "server application" might actually be a series of communicating applications that perform the server tasks. This researcher identified six major server application design goals:

1. The server application is not required to be cross-platform.
2. The server application should be scalable.
3. The server application must coordinate the storage of cultural data.
4. The server application must retrieve broadcast transcripts.
5. The server application must capture video and correlate it with transcripts.
6. The server application must analyze video and prepare models.

The server application is not required to be cross-platform. The server application will run on the machine or machines that service the Public News Network. Because the platform choice can be controlled in this scenario the application does not need to be cross-platform.

Like the client application the server application should remain scalable so that it can be easily extended based on changes in technology. The server application should attempt to use API's in such a way that other API's can be substituted and provide the same functionality. This will protect the application in the event that a particular technology is discontinued.

One of the primary goals of the server will be to act as a repository of information such as video, models, and transcripts. The file sizes associated with video data are enormous and therefore the PNN server will require a great deal of storage space. Presently the server contains a 120 GB hard drive that will be dedicated to video storage.

Another primary design objective is the retrieval of broadcast transcripts. In order to allow the client applications the ability to retrieve video associated with search terms the server (or delegate) must handle the retrieval of news broadcast transcripts. This task must be accomplished on a nightly basis and must handle the storage of these transcripts on the PNN server.

A critical design objective is the capture and correlation of video with transcripts. The server will act as a database that will correlate the varied media and allow the client applications to search and retrieve cultural data. This video will be segmented into easily streamed sizes that correspond with the transcript time code. The PNN web server will handle these video clips as streaming media files.

The final primary design requirement is the analysis of captured video and the production of 3-D models. Each day's video capture will be used by the server application to generate 3-D models representing each day's broadcast. It is anticipated that this portion of the server application will likely be a separate application dedicated to this function.

DESIGN DEVELOPMENT

The pragmatism of actually developing a functioning application is a challenge considering the multitude of choices to be made. Application Programming Interfaces (API) are plentiful, each with its own set of advantages and disadvantages. This section will outline API chosen for this project as well as some of the decisions made along the way. This section will then discuss the hardware chosen to implement PNN.

Application Programming Interface

A computer can be programmed using any number of programming languages such as C, C++, Java, Paschal, Fortran, etc. Each of these languages has advantages and disadvantages. Hierarchically below the choice of programming language is the choice of API. An API is a set of functions or libraries that encapsulate certain behavior and simplify the programmer's task of retrieving that behavior from the application. To understand API one must understand "higher-level" and "lower-level" when it refers to programming languages. A lower-level programming language interacts with the computer at a "lower" level. In other words, it might interact with the computer at the bit switching, hardware level. The written code at this level might not be easily recognizable by humans. A higher-level programming language interacts with the computer at a "higher" level by abstracting the behavior of the computer into understandable chunks. Typically the higher the level the more closely the language reflects human language patterns. An API takes a set of desired functionality and encapsulates it into a higher-level form to ease the programmer's job. The API acts as an interface between the programmer and the hardware, or other software at a lower level. During the development of this project API choices were one of the most difficult and time consuming issues encountered. The API is the software battleground of corporations, each jockeying for control to establish a profitable computer standard.

Every platform operating system has its own preferred set of programming methodologies that limit an application to running on that one operating system. Corporations do this intentionally to lock programmers into their operating system and thus preserve their place in the market. A key design goal for this project is cross-platform compatibility. This choice immediately reduces the number of API choices, but actually complicates program design because there are multiple ways to accomplish the same goal. If you program for one platform most of these decisions are made for you.

This researcher identified two primary APIs that would be necessary to implement this project. An API will be required to handle the creation and display of the 3-D virtual space and a second API will be required to handle the multimedia requirements of video display. The two primary 3-D graphics APIs that exist today are OpenGL [25] and Direct3D [26]. Direct3D is a Microsoft proprietary API, so the choice for 3-D graphics was simple; this project will use OpenGL. The second fundamental API will handle multimedia integration such as audio, video and still image display. There are a number of APIs that handle still images and/or audio. The key requirement necessary for the implementation of PNN is streaming video. There are three major multimedia APIs that handle streaming video, DirectShow [27], RealMedia [28], and QuickTime

[29]. DirectShow is a Microsoft technology and therefore it is not cross-platform. At the time this project was initiated RealMedia was a proprietary protocol of Real Networks which cost several thousand dollars to serve. To this researcher, RealMedia seems be closely tied to the interface of their RealPlayer software. For a project such as this, access to the core video display routines, independent of a pre-packaged proprietary interface, is needed. Recently Real Networks released their source into the open source community under the name Helix and it appears there is support for all of the major platforms. For this reason Helix may be an API to reconsider in the future. QuickTime is a multimedia API by Apple Computer, Inc. QuickTime supports Windows and Macintosh platforms and also supports Linux with a shareware plug-in. Unfortunately for this project, the Linux plug-in will not provide the functions needed to stream video onto a 3-D surface. Despite the lack of Linux support, QuickTime is the most compatible API for this application, and will be used for this project.

Having decided the two primary APIs required for this project, this researcher began to program a rough skeleton project in August of 2001. After months of programming and achieving a partially working C/C++ version of the project it became obvious that cross-platform issues, as well as the feasibility of accomplishing the networking goals of the project, were going to be problematic. In order to program OpenGL in a cross-platform manner using C/C++, the GL Utility Toolkit (GLUT) [30] was to used to handle interface and event mechanism design. GLUT was created to allow the programmer to design an OpenGL program that will run on all platforms that support OpenGL without having to implement platform specific mechanisms. For this reason it seemed like the ideal solution for this project. Except for a few minor bugs in the Macintosh implementation it worked well. As the project developed this researcher began to face the limitations of GLUT. GLUT does not provide any graphical interface design capabilities. For example, it does not provide items commonly called “widgets.” Widgets are the buttons, menus, checkboxes, etc., common in computer interfaces of all platforms. This was not a major issue because there are other toolkits that can be utilized along with GLUT to compensate for this limitation. The largest problem with GLUT was the seeming instability of the event handling mechanism on the Macintosh. Some of these problems were the result of the immaturity of the Macintosh OSX platform during the initial stages of the development of this project. Most of these problems were solved and it appeared the project was on track, until this researcher began to investigate what would be necessary to implement the networking capabilities of PNN. Designing a program that could contact a server, retrieve files, and send information back to the server would require platform specific code for each of the platforms supported. This is not impossible, but for a program written by a lone individual it appeared this would take more time than this researcher was willing or able to invest.

In order to solve these problems this researcher began to investigate alternatives to the initial path. There were several APIs that promised cross-platform and networking capabilities, such as Simple DirectMedia Layer (SDL) [31], Open Scene Graph (OSG) [32], wxWindows [33], and Qt [34], but each of these had disadvantages that made them undesirable for this project. SDL somewhat weakly supports video, but the combination of video, video streaming, and OpenGL was questionable if not impossible. OSG doesn't support streaming video. OpenGL support was extremely weak in wxWindows, and Qt is a commercial product on all but Linux systems and is too expensive. Eventually this researcher decided to utilize Java [35] as the base programming language for PNN. Java is renowned for its cross-platform capabilities. Java has very strong Graphical User Interface (GUI) design capabilities and excels at cross-platform networking. Java has 3-D graphics capabilities but they are not nearly as fast or as flexible as OpenGL. Java also has multimedia capabilities but they are not as full featured as QuickTime. Fortunately there are APIs that allow the programmer to use both OpenGL and QuickTime with Java.

OpenGL for Java [36], also know as GL4Java, is an open source binding of Java to the supporting platform's OpenGL implementation. The beauty of this API is that it works with the native OpenGL C++ routines and is therefore relatively fast. Using this library with Java provides the best of both worlds: a strong GUI design capability combined with a well-developed 3-D API. The final API necessary to complete this puzzle is QuickTime for Java [37]. Like GL4Java, QuickTime for Java is a binding of Java to the supporting platform's QuickTime implementation. As long as QuickTime is installed on the host computer, QuickTime for Java allows the programmer to call QuickTime functions from the application.

In summary PNN will utilize two APIs in conjunction with the Java programming language: OpenGL for Java, and QuickTime for Java. These APIs will allow this researcher to take advantage of the inherent GUI and networking strengths of Java while allowing access to the host computer's multimedia and 3-D graphics architecture.

Hardware

PNN will be programmed on an Apple Macintosh G4 computer with dual 800 MHz processors. The application will also be tested on an Apple iBook 600 MHz and an Apple dual 500 MHz system. All of the Apple systems will be running the latest version of the OSX operating system (OS) with the dual 500 MHz machine running the server version of OSX. Windows OS compatibility will be tested on a Dell Inspiron 1.8 GHz machine running Windows XP as well as a "home-built" machine with dual Intel 733 MHz processors running Windows NT.

Five nights per week the evening news broadcast is recorded and stored on the server hard drive. The recording is accomplished using a Formac Studio DV/TV (Fig. 12) video capture device. This device connects to the Firewire (IEE-1394) port of the dual 800 G4, and at the scheduled broadcast time, records the news in Digital Video (DV) format. The captured video is then re-compressed into Motion Pictures Expert Group 4 (MPEG4) format. This format is ideal for streaming video due to its ability to provide small file sizes while maintaining image quality.

Fig. 12. Formac Studio DV/TV video capture device.

Simultaneously with the recording of the news video a BOCA Labs TextGrabber GP500 (Fig. 13) is used to capture the closed caption information recorded into line 21 of the NTSC video signal. This device is connected to the computer via the serial port and extracts the transcript as well as the time code and broadcast information of the associated video. This data is saved to the computer as a text file that is then uploaded to the PNN server for processing by the PNN client application.

Fig. 13. BOCA Labs TextGrabber GP500 closed caption device.

CONSTRUCTION

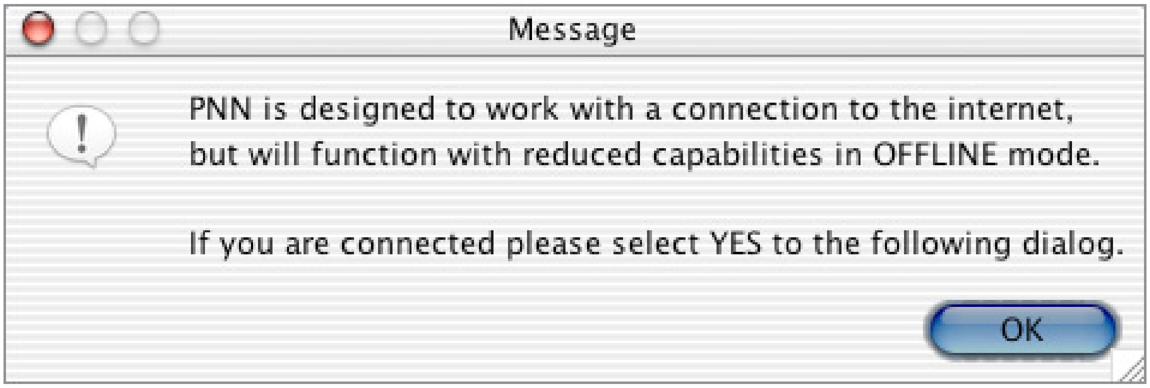
Once the design goals, APIs, and hardware have been identified the next step is to establish the resultant functionality of the application. An application as complex as PNN requires many modules each performing a specific task in order to accomplish the design objectives. The "Client Application" section provides a linear description of the client application from initialization to conclusion. It describes what is seen based on viewer input. The details of its implementation will be described in Chapter IV, Implementation and Results. The subsequent section, "Server Application (future)" outlines the processes that will be required of the future server application. This section illustrates the flow of control necessary to deliver the required content to the client application.

Client Application

The client application is the visual component of the PNN project. The client program is responsible for presenting the aesthetic to the viewer, and for the navigation of space. The following sections "walk" the reader through the PNN client application.

Initialization

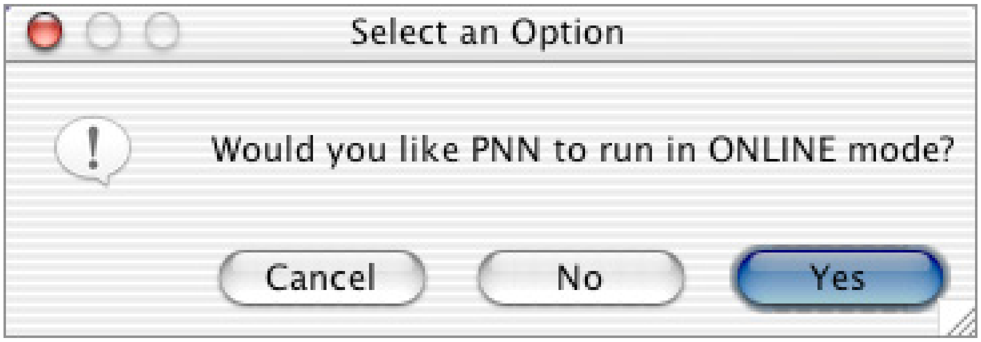
When the viewer first initiates the program, she will be presented with a dialog (Fig. 14) that explains the two modes within which PNN will operate, "online" and "offline." Acknowledgement of this dialog presents a second dialog (Fig. 15) giving the viewer a choice between the two modes. In "online" mode, PNN retrieves data such as models and movie files from the PNN server. As the program is initiated, each model and an associated news transcript is downloaded from the server and placed within application memory.



PNN is designed to work with a connection to the internet, but will function with reduced capabilities in OFFLINE mode.

If you are connected please select YES to the following dialog.

Fig. 14. The initial startup dialog.



Would you like PNN to run in ONLINE mode?

Fig. 15. The Online/Offline selection dialog.

In “online” mode, video sources are also retrieved from the PNN server. In this mode, the application is dynamic by virtue of the fact that it is a reflection of the latest information available in the PNN database. “Offline” mode is available for those situations where the viewer has no connection to the Internet, or otherwise wishes to work with data that is stored locally within the application. In this mode a default set of model files and transcripts, as well as abbreviated movies included with the application distribution, are the source of imagery.

Following the online/offline dialogs PNN presents a startup window that displays the PNN logo animation (Fig. 16). This animation, created using Macromedia Flash, cycles while the application is loading resources.

Fig. 16. A frame from the PNN startup animation.

At the completion of the resource-loading phase this startup window is closed and the main application window is displayed.

Main Window

The main application window (Fig. 17) presents a three-dimensional space defined by a horizon line and a view of one or more of the ten objects representing news episodes. Each day, as a new news episode is recorded, the oldest previous episode is replaced with the current recording.

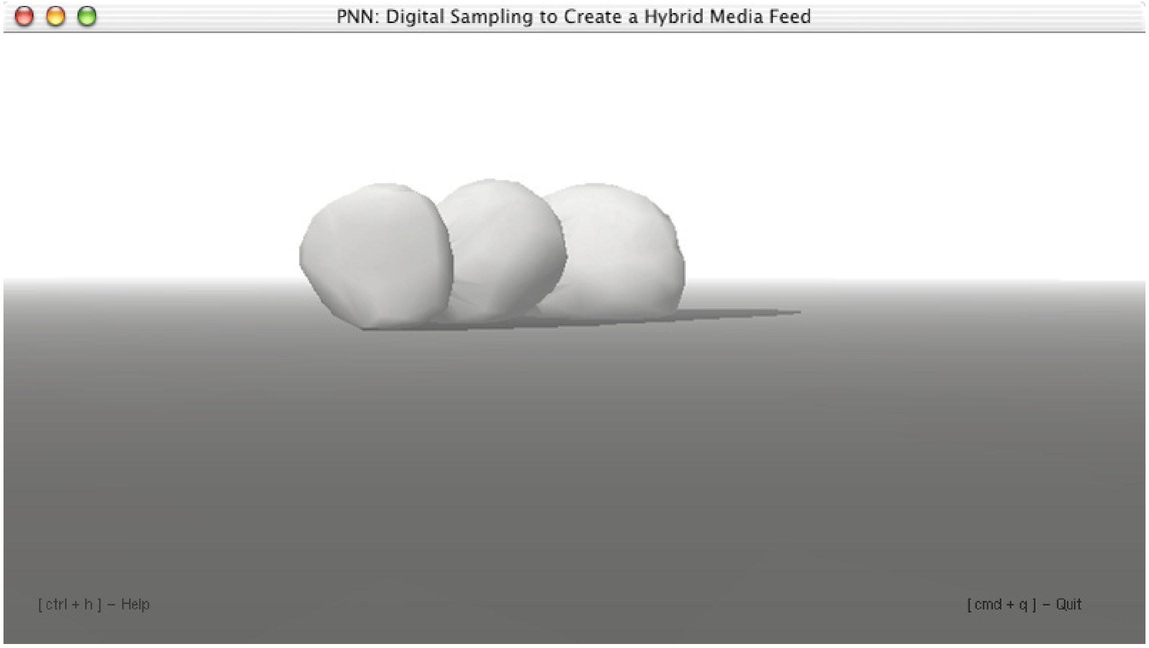


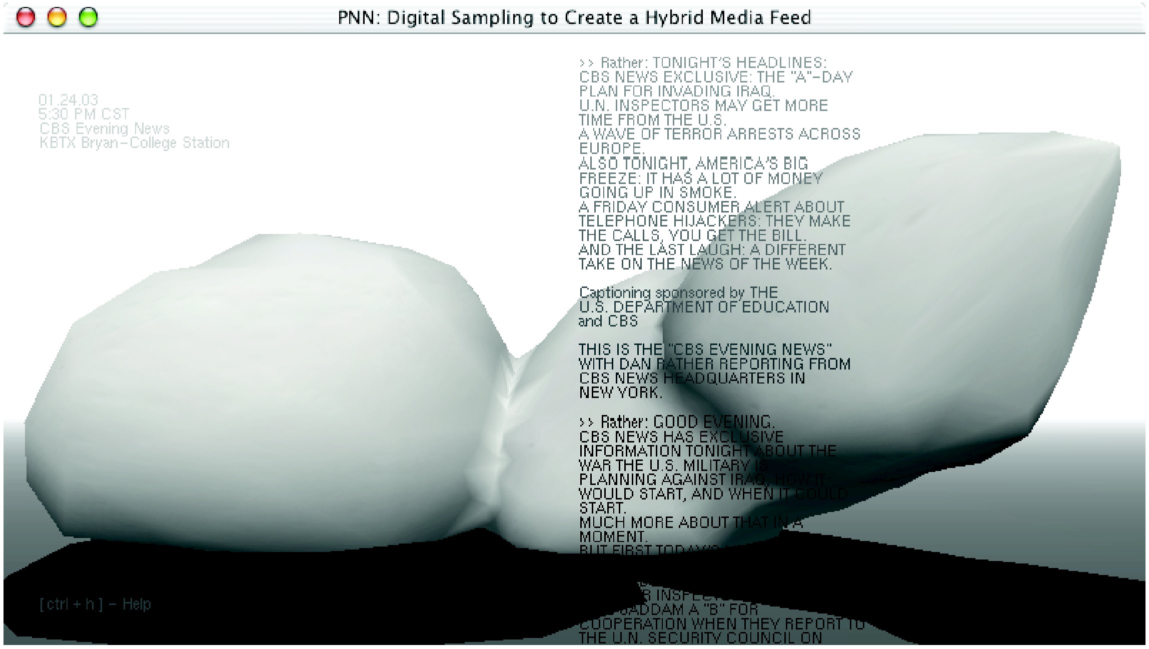
Fig. 17. The main application window showing neutral textures.

This limit of ten episodes keeps the total amount of data to be displayed by the client application within a reasonable value. When the initial scene loads all ten of the episode objects as well as the ground plane are surfaced with a neutral off-white texture. The episode objects are distributed randomly throughout the space and appear to rest on the surface of the ground plane. The only other element defining the space is the sound of wind blowing in the background.

At the completion of scene loading the viewer will notice that the mouse cursor has changed from the familiar arrow icon to a small tee or cross shape. As the viewer moves the cursor the viewpoint rotates left, right, up, and down. The cursor controls which direction the viewer looks within the space. The viewer can navigate the space by using the cursor to look in a particular direction while pressing either the up and down keys (or alternately, the “w” and “s” keys) to move forwards and backwards. These keys are the standard keys used for similar behavior in most video games. The first thing the viewer will notice as she navigates the space is that the background audio changes as she moves towards episode objects. As she approaches an episode, the audio cross dissolves into the sound of television static. Similarly, as she passes on the left or right of an episode object, the audio is shifted to the right or left stereo audio channel. This simple spatialization of the audio enhances the immersive qualities of the experience.

Episode Information and Transcripts

As the viewer approaches an episode object a short textual description of the object appears in the upper left hand portion of the screen (Fig. 18).



01.24.03
5:30 PM CST
CBS Evening News
KBTX Bryan–College Station

>> Rather: TONIGHT'S HEADLINES:
CBS NEWS EXCLUSIVE: THE "A"–DAY
PLAN FOR INVADING IRAQ.
U.N. INSPECTORS MAY GET MORE
TIME FROM THE U.S.
A WAVE OF TERROR ARRESTS ACROSS
EUROPE.
ALSO TONIGHT, AMERICA'S BIG
FREEZE: IT HAS A LOT OF MONEY
GOING UP IN SMOKE.
A FRIDAY CONSUMER ALERT ABOUT
TELEPHONE HIJACKERS: THEY MAKE
THE CALLS, YOU GET THE BILL.
AND THE LAST LAUGH: A DIFFERENT
TAKE ON THE NEWS OF THE WEEK.

Captioning sponsored by THE
U.S. DEPARTMENT OF EDUCATION
and CBS

THIS IS THE "CBS EVENING NEWS"
WITH DAN RATHER REPORTING FROM
CBS NEWS HEADQUARTERS IN
NEW YORK.

>> Rather: GOOD EVENING.
CBS NEWS HAS EXCLUSIVE
INFORMATION TONIGHT ABOUT THE
WAR THE U.S. MILITARY IS
PLANNING AGAINST IRAQ, HOW IT
WOULD START, AND WHEN IT COULD
START.
MUCH MORE ABOUT THAT IN A
MOMENT.
BUT FIRST TODAY'S [illegible]
[illegible]
[illegible] INSPE[illegible]
[illegible] SADDAM A "B" FOR
COOPERATION WHEN THEY REPORT TO
THE U.N. SECURITY COUNCIL ON

Fig. 18. View showing episode description and transcript display.

This description identifies the episode object by date and time of production, the name of the broadcast, and the local television network from which it was recorded. If the viewer single clicks on the episode object, the transcript of the days news broadcast will scroll across the screen. Single clicking again or moving away from the object disables the transcript display.

Episode Video

Most of the actions available within PNN are specific to episode objects located within the space. The viewer can enable the display of video by double clicking on any episode object (Fig. 19). This will cause the episode object as well as the ground plane to be textured with whatever video corresponds with the episode selected.

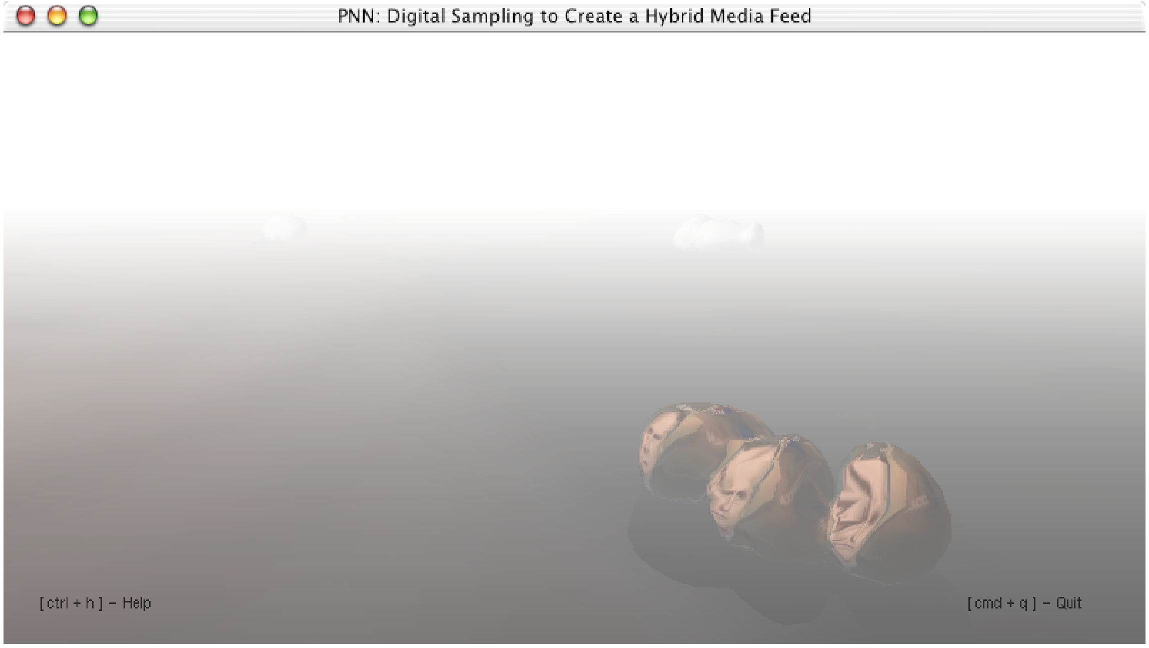


Fig. 19. A bird's eye view showing an episode textured with video.

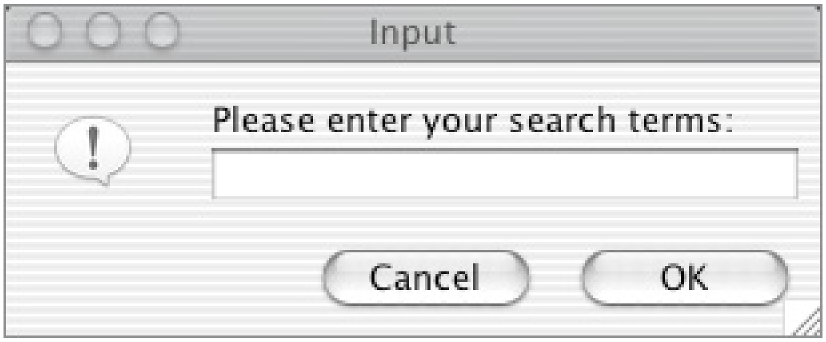
The viewer can disable this video texture display by either double clicking again on the same object, which will revert the scene to the original textured display, or by double clicking on another episode object. In the later case the video will change to the video corresponding to the newly selected episode object.

Mouse Navigation

In some circumstances it may be desirable to disable the ability of the mouse cursor to control the viewer's orientation, or the "mouse look" feature. This can be accomplished by single clicking anywhere in the scene except on an episode object. This will immediately freeze the orientation of the scene and allow the viewer to focus on other tasks. The viewer can again re-enable “mouse look” by double clicking anywhere in the scene except on an episode object. This mouse/keyboard navigation, and episode selection/de-selection, describes the fundamental navigation and manipulation of the PNN environment.

Video Search

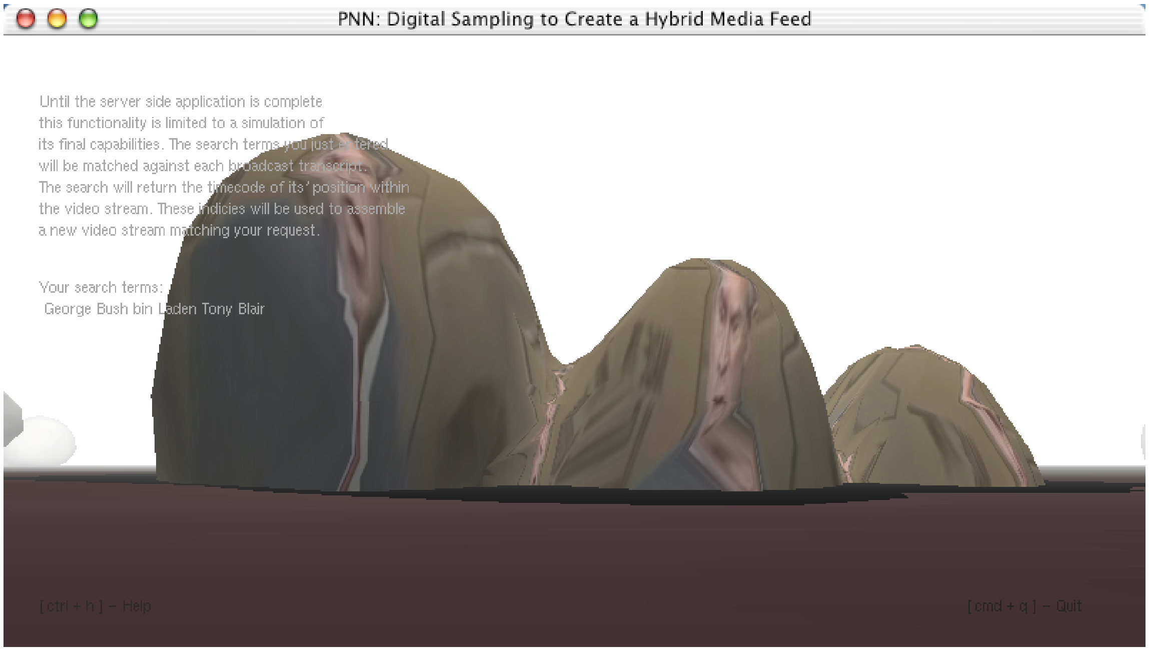
In addition to basic navigation and episode interaction PNN provides the ability to recombine and manipulate the evening news. The viewer can type the control key plus the spacebar to initiate the search dialog (Fig. 20). This will pop up a window that allows the viewer to enter a search string.



Please enter your search terms:

Fig. 20. The search dialog.

This search string will subsequently be matched against the transcripts of each of the ten episode broadcasts. When a word is found within a broadcast a segment of video encompassing the word is returned to the application and is pieced together with the results of the other search terms to produce a new video. This new video is mapped to a new episode object and displayed before the viewer (Fig. 21). In this manner PNN gives the viewer control over the news and enables the viewer to appropriate its content and meaning. Because this feature relies on the presence of the server application for the correlation of video and transcript information this feature is only simulated in this project.



Until the server side application is complete
this functionality is limited to a simulation of
its final capabilities. The search terms you just entered
will be matched against each broadcast transcript.
The search will return the timecode of its' position within
the video stream. These indicies will be used to assemble
a new video stream matching your request.

Your search terms:
George Bush bin Laden Tony Blair

Fig. 21. Simulated view of video search results.

Settings and Preferences

The final component of the client application is the ability to customize and save application settings. PNN will display the “PNN Settings” dialog, which includes several customization tabs by typing the control key plus the “h” key. This key combination will present the dialog with the focus on the “Help” tab. As the name suggests, the “Help” tab provides basic instructions to assist the viewer in the operation of the PNN client application. “PNN Settings” also includes a tab named “Performance Settings” which allows the viewer to tune the performance of PNN to her specific machine. Another tab called “View Settings” contains adjustments that tailor the appearance of PNN and enable some informational display, and yet another tab called “Corporate Settings” allows the viewer to tailor the application to match her individual beliefs concerning digital rights management. The final tab, “Information” displays the application name and credits. Many of the settings controlled within these tabs are integrated with a system of preferences that can be saved, and affect the characteristics of the application upon restart. In this manner PNN can be tailored to meet the expectations of the viewer without having to be customized at each launch. The function of these tabs will be discussed more fully in Chapter IV, Implementation and Results.

Server Application (future)

The PNN server application will consist of several sub-applications that perform three major tasks. The server application will collect, process and store news broadcast video; collect, process and store news broadcast transcripts; and produce 3-D objects based on an analysis of the news broadcast video.

Video

In order to collect, process and store news broadcast video, the server application will coordinate with the video capture software provided with the Formac DV/TV video capture device. Fig. 22 shows a schematic view of the overall process. On a nightly basis the server application will convert the DV format video captured by the capture device into a hinted streaming video in the MPEG4 format. This process will be accomplished using QuickTime for Java which includes the necessary conversion functions.

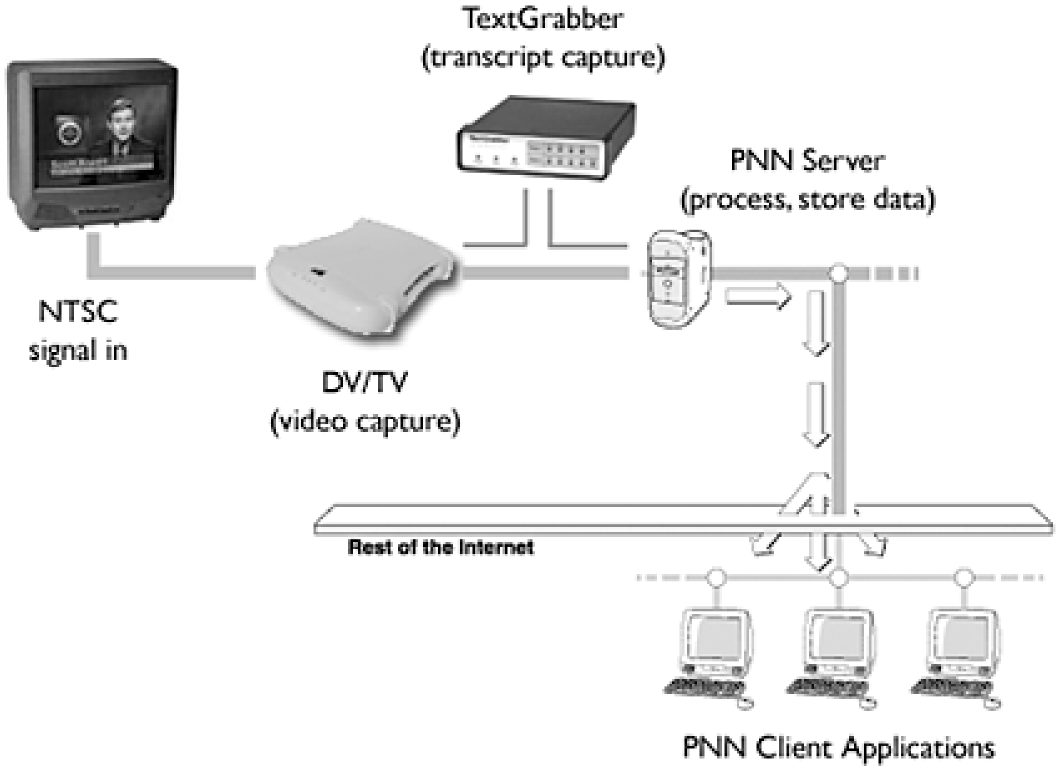
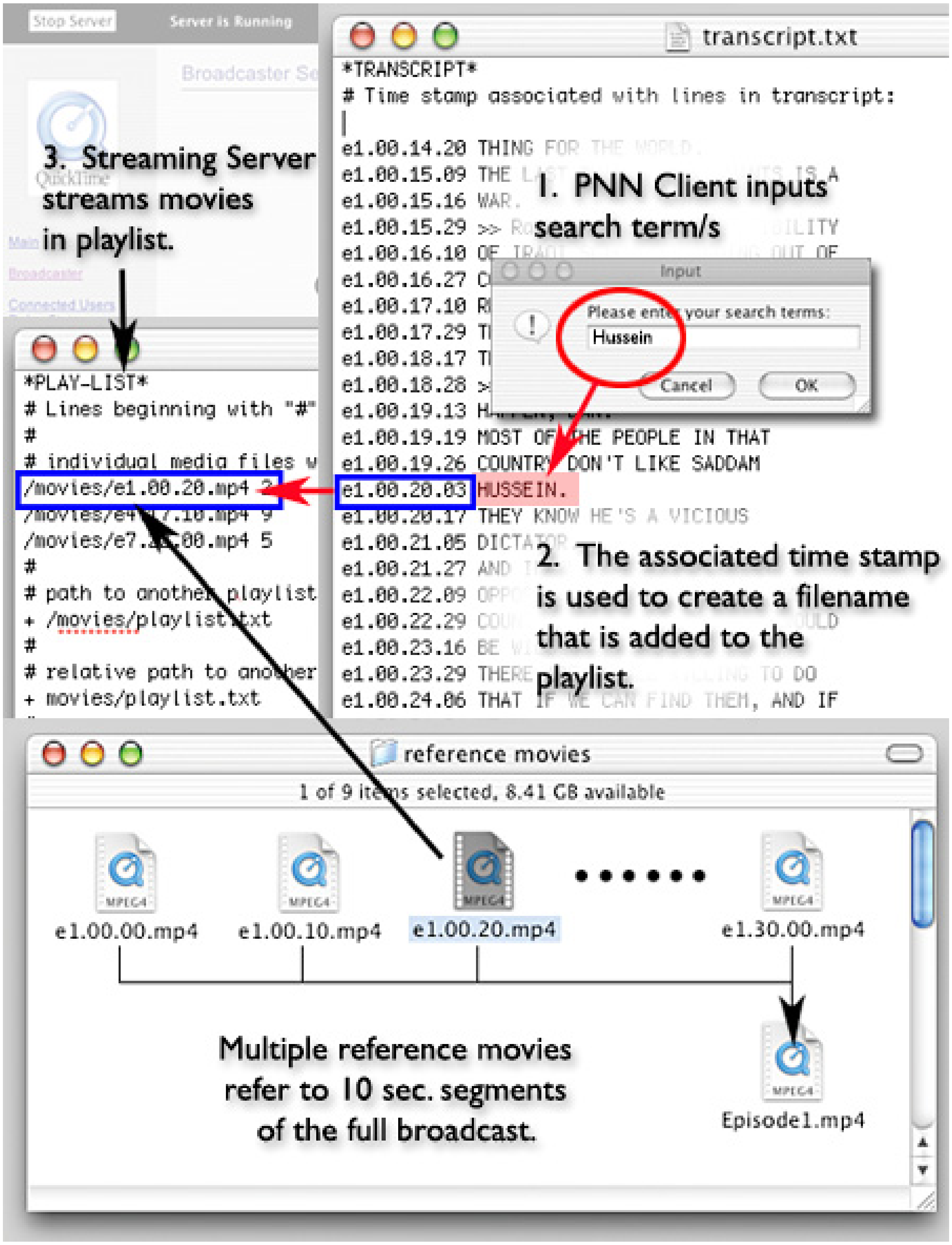


Fig. 22. A schematic diagram of the server process.

In addition to simple format conversion the application will prepare the movie for the PNN client application by creating multiple reference movies that correspond to ten-second intervals of movie time. A reference movie is a small QuickTime movie file that points to another movie file. A reference movie does not contain the actual movie data and therefore has a much smaller file size. It is these reference movies correlated to movie time code that will be used by the PNN client application to construct new movies based on search results. Following completion of the processing of the nightly movie the server application will move the movie and reference movies to the appropriate directory on the PNN web server and archive the oldest previous movie files.

Transcripts

The server application will also be responsible for handling the collection, processing and storing of news transcripts. Fig. 23 shows a schematic view of the process. Simultaneous with the recording of the news broadcast, the TextGrabber GP500 will be initialized to capture the closed caption information embedded in the news broadcast video stream. The closed caption information is available in all closed captioned broadcasts and is encoded into line 21 of the broadcast stream. The GP500 will be configured to not only capture the transcript itself but also to capture the time stamp information that correlates the text with the video time code.



3. Streaming Server streams movies in playlist.

```
*PLAY-LIST*
# Lines beginning with "#"
#
# individual media files w
/movies/e1.00.20.mp4 2
/movies/e4  7.10.mp4 9
/movies/e7.  00.mp4 5
#
# path to another playlist
+ /movies/playlist.txt
#
# relative path to another
+ movies/playlist.txt
```

```
*TRANSCRIPT*
# Time stamp associated with lines in transcript:
|
e1.00.14.20 THING FOR THE WORLD.
e1.00.15.09 THE L                    NTS IS A
e1.00.15.16 WAR.
e1.00.15.29 >> R                     ILITY
e1.00.16.10 OF IRAQ                  NG OUT OF
e1.00.16.27 C
e1.00.17.10 R
e1.00.17.29 T
e1.00.18.17 T
e1.00.18.28 >
e1.00.19.13 H
e1.00.19.19 MOST OF THE PEOPLE IN THAT
e1.00.19.26 COUNTRY DON'T LIKE SADDAM
e1.00.20.03 HUSSEIN.
e1.00.20.17 THEY KNOW HE'S A VICIOUS
e1.00.21.05 DICTATOR
e1.00.21.27 AND I
e1.00.22.09 OPP
e1.00.22.29 COU                      OULD
e1.00.23.16 BE WI
e1.00.23.29 THERE                    LING TO DO
e1.00.24.06 THAT IF WE CAN FIND THEM, AND IF
```

1. PNN Client inputs search term/s

Please enter your search terms: Hussein

2. The associated time stamp is used to create a filename that is added to the playlist.

1 of 9 items selected, 8.41 GB available

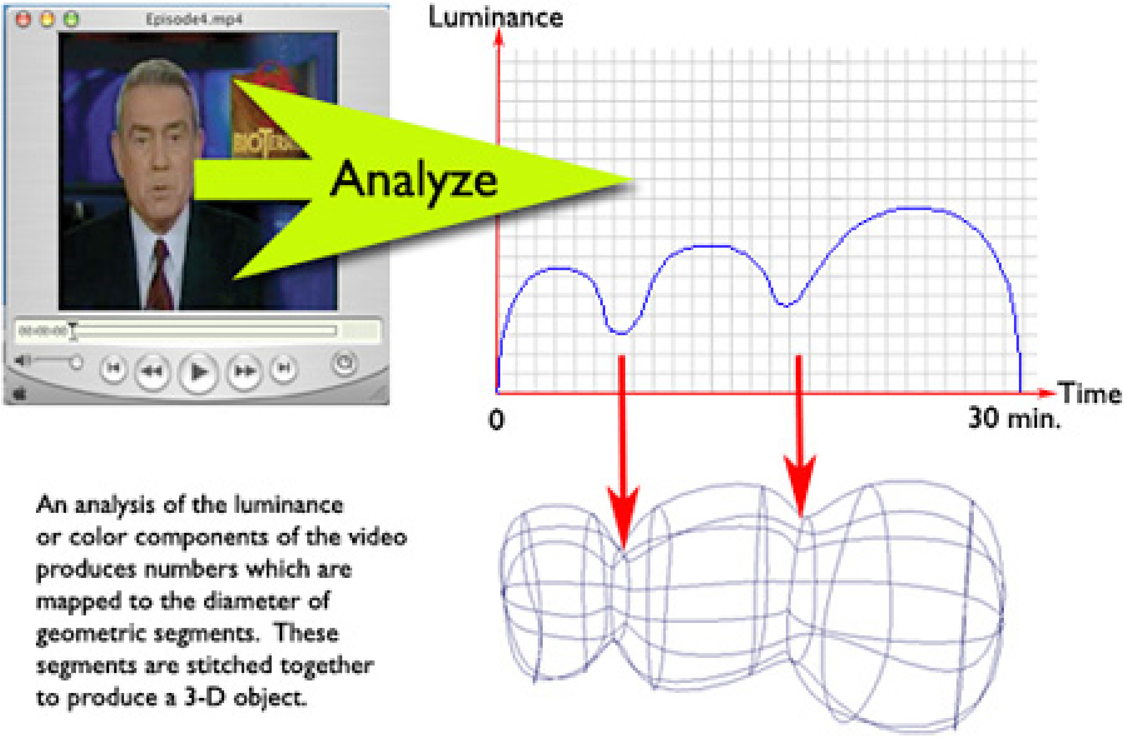
Multiple reference movies refer to 10 sec. segments of the full broadcast.

Fig. 23. A schematic view of the video search process.

The time stamp associated with each line of the transcript is the value returned when the PNN client application searches for terms within the transcript. These time stamp numbers provide the index number of the reference movies created during the video-processing phase, thus matching text to video. Once the transcript has been captured it will be moved to the appropriate directory on the PNN web server and the oldest previous transcript will be archived.

Models

The third primary function of the PNN server application will be the creation of 3-D episode objects. The intention is to map the video data in such a way that the 3-D model is a reflection of the video content. If the video can be described as exhibiting certain characteristics, for example, "cheerful," then the model generated from this video should look different than a model generated from a video associated with the term "dark." This is the least developed of the server-side concepts because there are multiple ways with which to accomplish the same goal, and without having begun to test these ideas, this researcher is not sure which method is the best. This phase should create geometry that can be efficiently displayed by OpenGL and that effectively represents what thirty-minute news broadcasts might look like as a 3-D object. One method that might accomplish this goal is to sample the video at regular intervals throughout its length and extract hue and luminance values (Fig. 24).



An analysis of the luminance or color components of the video produces numbers which are mapped to the diameter of geometric segments. These segments are stitched together to produce a 3-D object.

Fig. 24. A schematic view of one possible model production methodology.

These values could be used to create a cross sectional diameter that is used to create either a spline or polygon segment that can be stitched to adjacent segments, thus creating a whole. In this manner each episode object would be a direct reflection of the video that constitutes the broadcast it represents. Another method to accomplish this goal might be to use the same hue and luminance values to create a displacement map. This displacement map could be applied to a standard spherical object to create a new displaced shape representing the particular news episode examined. Either method would produce a shape that would subsequently be placed in the proper directory on the PNN web server. As with the previous steps, the oldest previous object would then be archived.

CHAPTER IV

IMPLEMENTATION AND RESULTS

The implementation of PNN requires the coordination of a number of functions and classes related to the presentation of the visual aesthetic, the processing of data and general usability. This chapter presents the implementation of the client application through a discussion of the functions and classes designed to achieve the project goals.

CLIENT APPLICATION

The focus of the work required to implement the client application involves the coding of a graphics pipeline that is flexible and efficient. Piecing together multiple APIs such that each works well with the other was a major challenge. The following sections detail the primary components necessary to implement the PNN client application.

Models

When this project was begun this researcher had very little experience utilizing OpenGL. In classes taught at the Visualization Lab at Texas A&M University the primary focus involved testing ideas utilizing basic GLUT or OpenGL primitives. This project would require a more sophisticated model format than the standard primitives could afford. In order to produce PNN the first task undertaken by this researcher was to identify a 3-D model format that would integrate with the goals of the PNN client application. After several weeks of testing different formats this researcher decided that the Alias ".obj" format appeared to be the most flexible format available. This format can be expressed as an ASCII text file and can therefore be easily manipulated and customized to meet the specific needs of PNN. During the initial phases of programming, while utilizing the C and C++ languages, this researcher used Mark Kilgard's glm.c and glm.h ".obj" library. This code is a well developed implementation of an ".obj" reader and writer. As mentioned previously, during later stages of the development of this project this researcher decided to change from C/C++ as the language of choice to Java. In order to work with the Java language this researcher rewrote and adapted the glm.c functions to work with the new language. Because the model generation capabilities of the PNN server application are not implemented during this phase of the project, Alias|Wavefront Maya was used to produce ".obj" models that approximate the expected future model visualization. These models are exported from Maya with normals and texture coordinates so they are ready for display within PNN. Upon launch PNN creates an instance of the class "Episode" for each of the ten models imported. The Episode class is used to describe an instance of the daily news including the model, the transcript, the movie, and the texture identification. The Episode class keeps track of any information specific to an episode such as whether it is presently selected, its position, etc. Each episode instance contains its own draw() function that is responsible for drawing the episode model at the appropriate location in space. Each episode draw() function is stored in an OpenGL display list in order to speed the display of information on screen.

Textures

For static image texturing PNN uses the ".png" texture loading utilities built into GL4Java. These texture utilities allow images to be loaded from the local disk or from a network uniform resource location (URL). This texture format works well for this purpose because the file size is small and it can contain an embedded alpha channel. An embedded alpha channel allows transparency effects to be utilized when displaying images or geometry. In anticipation of future developments a texture loader that utilizes QuickTime's ability to work with multiple image formats, such as ".jpg," ".tiff," ".gif," etc. was developed. This will allow PNN to texture models with most any common image format.

Camera

After implementing basic model import and texturing capabilities the next step was to integrate a system to navigate the space. This researcher investigated many different camera setups discussed in online tutorials and books and decided to base the camera setup in PNN on a C++ tutorial by Ben Humphrey [38]. This camera setup works well because it is time based and therefore maintains the same performance on computers of different speeds. It is also easily integrated with a bounding box collision system that keeps the camera from passing through objects in space without slowing the computer to a crawl as collisions are tested. This camera setup creates a navigation system very similar to the navigation system familiar to players of the video game Quake, or Unreal. PNN utilizes the camera from the first person perspective. This essentially places the viewer at the position of the camera as she navigates the virtual world. As described previously, the viewer uses the mouse to “look” at various objects within the world and controls the direction of her motion by “looking” in a particular direction and pressing the appropriate keys on the keyboard.

Environment

With the navigation system complete, the core 3-D elements of PNN were in place. At this point the project shifted to improving the aesthetic qualities of the presentation. In keeping with the stated goal of creating a space with a sense of stark beauty this researcher decided to create a space that is essentially white. The background color combined with the off-white texture color creates a monochromatic “light” palette. In order to emphasize the shape of each episode object the OpenGL lighting was placed to create an evenly lit space that emphasizes the shape of the modeled surfaces. In addition to the surface lighting PNN implements real time shadow generation. By producing shadows that update with the camera orientation the episodes appear more realistically located within the space. This researcher based PNN’s shadows on one of Nate Robbins’ shadow tutorials [39]. While this tutorial implements “soft shadows” in order to improve the general response and speed of the application it was decided to run the application with only a single hard-edged shadow. In the future PNN can be recompiled with multiple shadows enabled and soft shadows will work as designed. This will allow PNN to adapt whenever hardware speeds can handle the extra processing requirements. To produce soft shadows the shadow is drawn into the stencil buffer from the point of view of the light source in multiple passes. Each sample is offset and when all of the samples are recombined it produces a single shadow with a soft edge. While these shadows look more realistic than hard-edged shadows the extra processor time it takes to generate each sample negatively impacts an application that utilizes a lot of camera movement like PNN. From an aesthetic point of view these improvements began to create the sense of space this researcher desired. At this point the primary aesthetic weakness was the lack of depth-of-field within the scene. This researcher investigated several methods to provide this level of simulation. Actually implementing depth-of-field calculations in the production of frames for this application were obviously something that could not be done while simultaneously maintaining the level of interactivity and response required. This researcher decided that implementing OpenGL fog would suffice in this regard. OpenGL fog allows the programmer to select from a set of fog equations to determine the relationship of fog density to the distance from objects within the scene. The further an object is from the viewer the more the object is blended with the fog color. Fog doesn't appear to negatively impact scene drawing performance and is therefore an ideal way to create the sense of atmosphere desired for PNN.

Video

Once the navigable space was established both functionally and aesthetically this researcher's attention was next focused on the video display portion of this project. While display of video on 3-D geometry has been done before, to this researcher's knowledge there was no precedent for display of video textures on 3-D objects in a Java application. The most difficult element of this task was the close integration of three distinct APIs, QuickTime, Java, and OpenGL. This researcher looked at sample code that played video textures on OpenGL objects using QuickTime and wrote a class for PNN that accomplished the same function. This class accepts a path to a movie file and creates an off-screen graphic buffer to which it writes each frame of the movie. The path to the movie can be a path to a file on disk, a file on the network (ie. a URL), or a link to a streaming video server using the Real Time Streaming Protocol (RTSP). As the movie is initialized a "MovieDrawingComplete()" procedure is registered with QuickTime, which guarantees that full frames of video information are sequentially written to the off-screen buffer before continuing to the next frame. As each frame is drawn into this buffer OpenGL "grabs" the video pixels in the buffer and substitutes the pixels for the pixels that are presently texture mapped to the episode object using "glTexSubImage2D()." This system works well because there is no synchronization between the OpenGL texturing and the rate of movie play. If the movie rate lags, OpenGL simply redraws the same frame and continues, while the movie draws as fast as possible. The most difficult part of this process was related to the speed of movie display. In order to prepare video frames for OpenGL's texture routines individual frames of the movie have to be converted to a pixel map and copied into an integer or byte array. Several of the routines included with QuickTime for Java, which one would think should be optimized for the purpose of copying QuickTime data, were in fact many times slower than the Java native “ArrayCopy()” routines. Once this was established this researcher was able to attain an acceptable average frame rate of 22 frames per second.

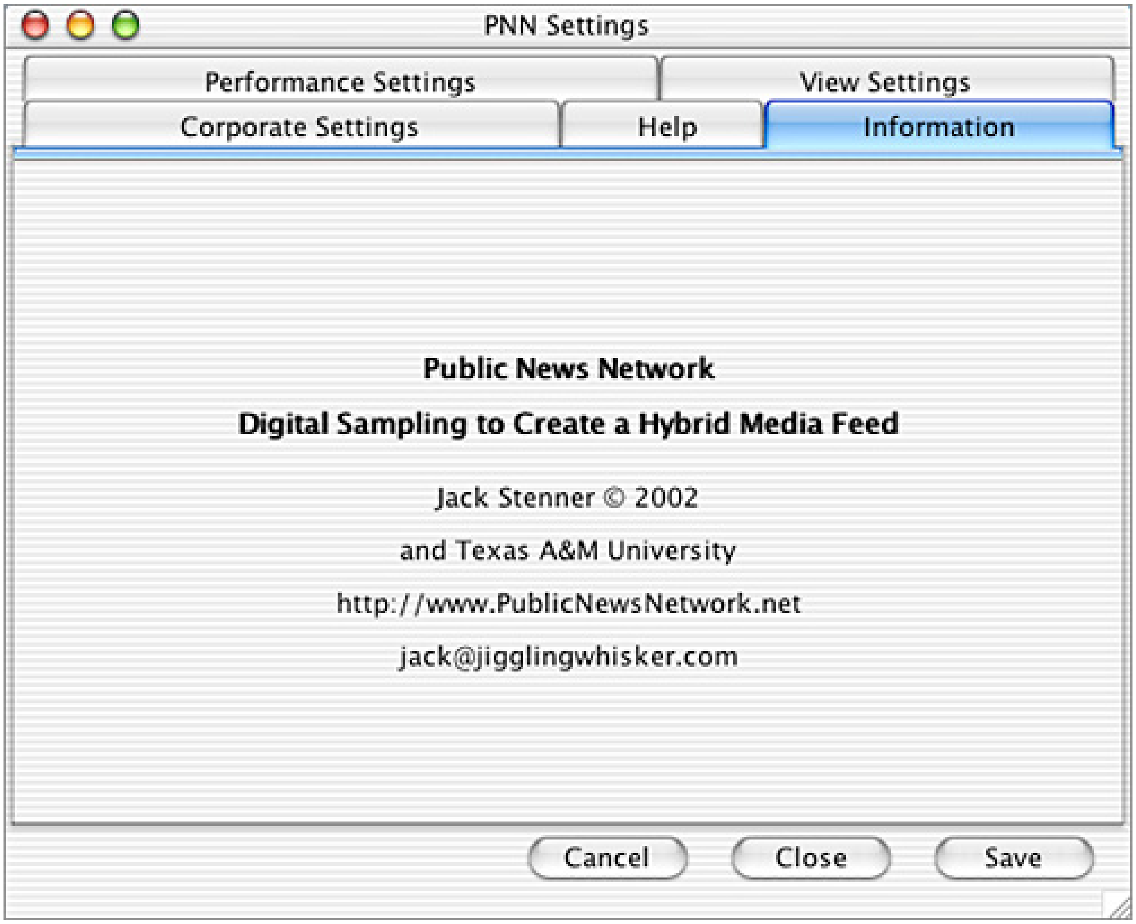
Transcripts

Having completed the 3-D visualization and the movie import and display routines the next step was to deal with the handling of transcript data. Java makes handling this type of information much easier that it would be using C or C++. The transcripts are text files saved on the PNN server and can be easily loaded using a URL encoded into the application. A class was created that contains structures to hold date and time information as well as header information and the body of the transcript itself. In order to facilitate the future implementation of video searching the transcript is read into a Java “HashMap” structure. A HashMap is a key-value pair structure. Whatever value is read into the value column can be made to return the associated key. This will benefit PNN, because the key associated with each line is a time stamp. The HashMap class includes the ability to search for a particular word, and when the word is found, the key (in this case, the time stamp) is returned. This functionality which is optimized for speed will make it relatively easy to implement text based video search when the server application is completed.

Preferences

The final element to be implemented in PNN was the help and preferences system. Because of the nature of this project this researcher wanted any graphical interface elements, other than the 3-D world itself, to have minimal impact on the aesthetic of the presentation. For this reason application settings and help information are accessible by a simple keyboard shortcut that is displayed in the lower left-hand corner of the main window. When the viewer types the control key plus “h,” a dialog window that consists of five tabs appears on screen.

The “Information” tab (Fig. 25) is a basic tab that simply provides the application name, the author and contact information.



Public News Network

Digital Sampling to Create a Hybrid Media Feed

Jack Stenner © 2002

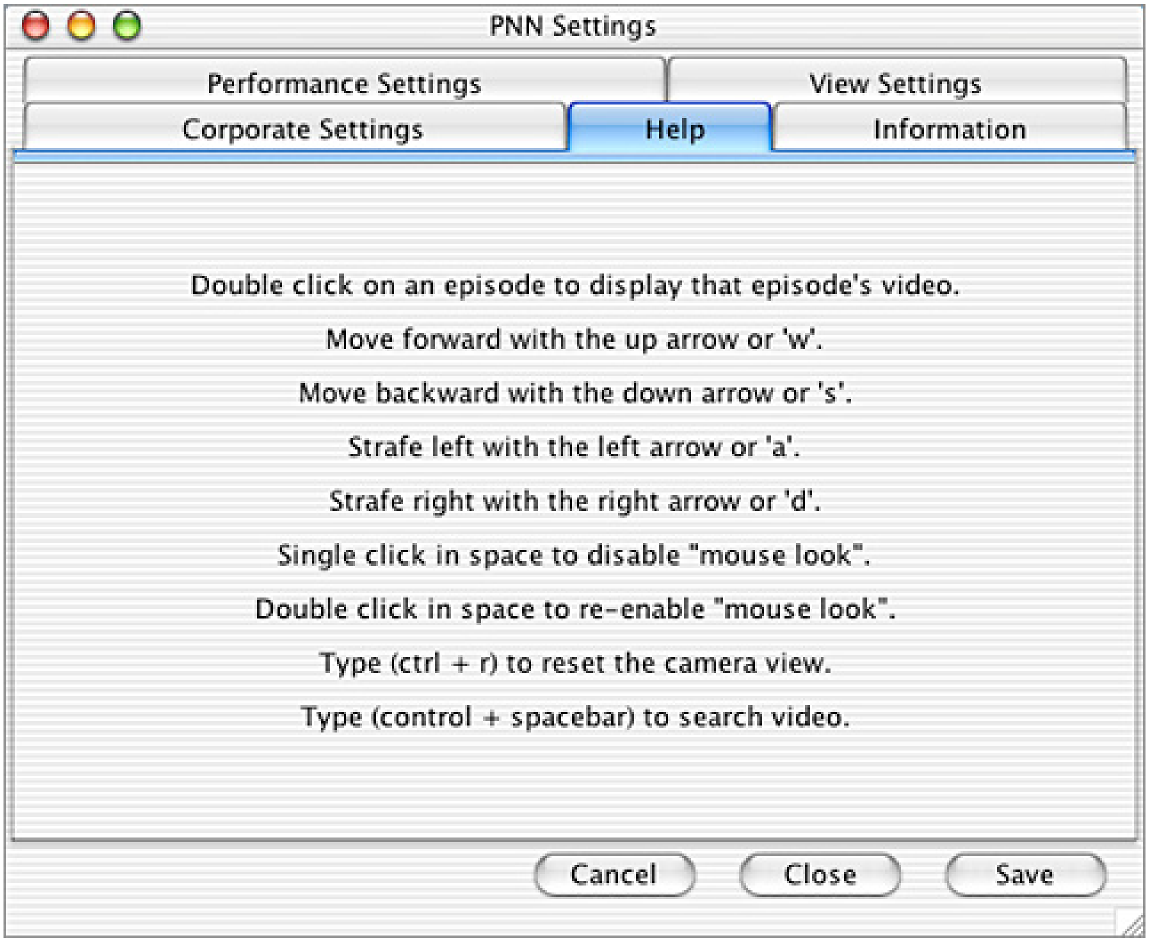
and Texas A&M University

http://www.PublicNewsNetwork.net

jack@jigglingwhisker.com

Fig. 25. The PNN Settings "Information" tab.

The primary tab that is displayed when the viewer initializes the PNN Settings dialog is the "Help" tab (Fig. 26). The Help tab outlines the basic operating instructions for PNN and provides the keyboard shortcuts that correspond to each option.



Double click on an episode to display that episode's video.

Move forward with the up arrow or 'w'.

Move backward with the down arrow or 's'.

Strafe left with the left arrow or 'a'.

Strafe right with the right arrow or 'd'.

Single click in space to disable "mouse look".

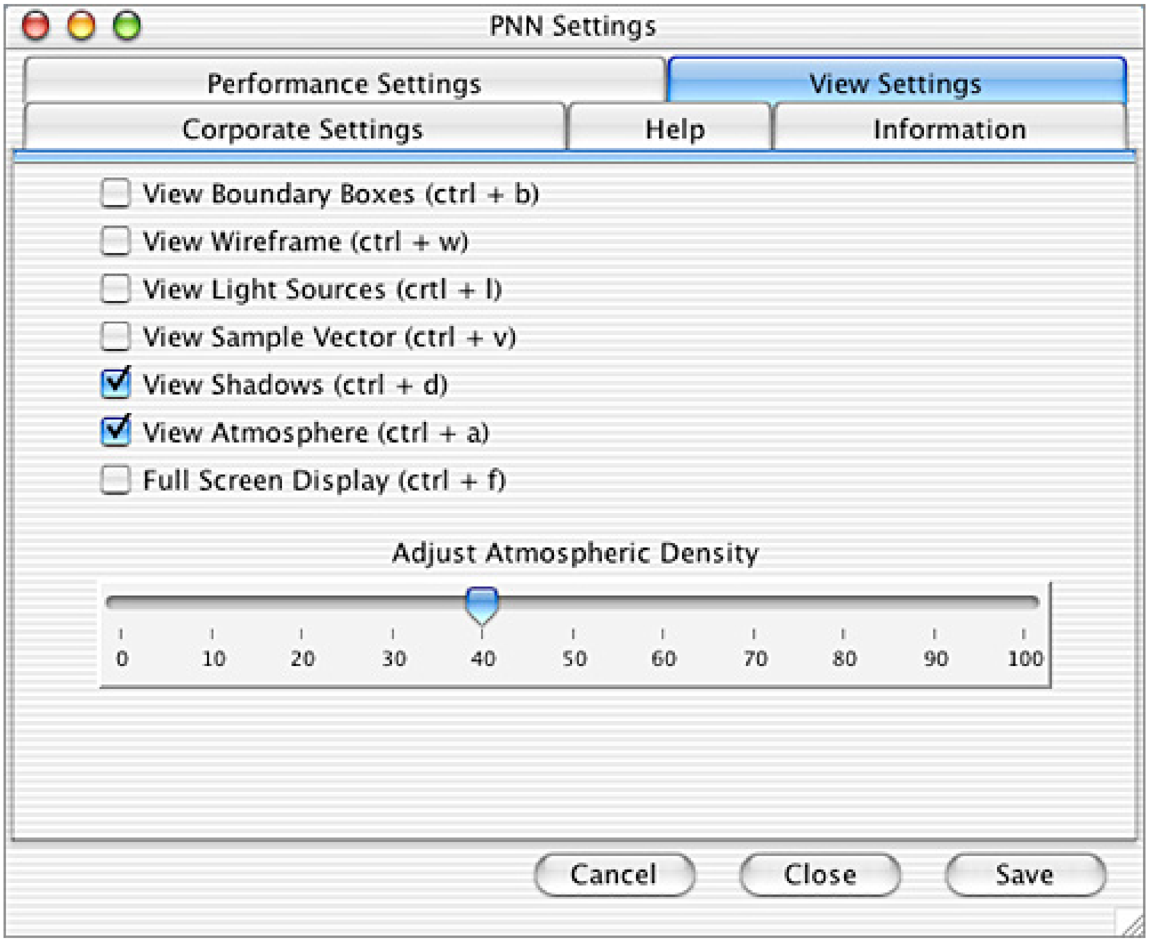
Double click in space to re-enable "mouse look".

Type (ctrl + r) to reset the camera view.

Type (control + spacebar) to search video.

Fig. 26. The PNN Settings "Help" tab.

The "View Settings" tab (Fig. 27) contains settings that control the appearance of the graphics within the 3-D window. Some of these settings are informational in that they are not settings that one would want to display by default, but they provide information about how the scene is constructed. For example, the "View Boundary Boxes" setting displays a representation of the boundaries around each object that are used to implement the collision detection routines. The "View Wireframe" setting does exactly as the title implies, and allows the viewer to see the structure of the 3-D geometry of each episode. When the "View Light Sources" setting is enabled, if the viewer orients her view towards the "sky," she will be able to view the lights that illuminate the scene.



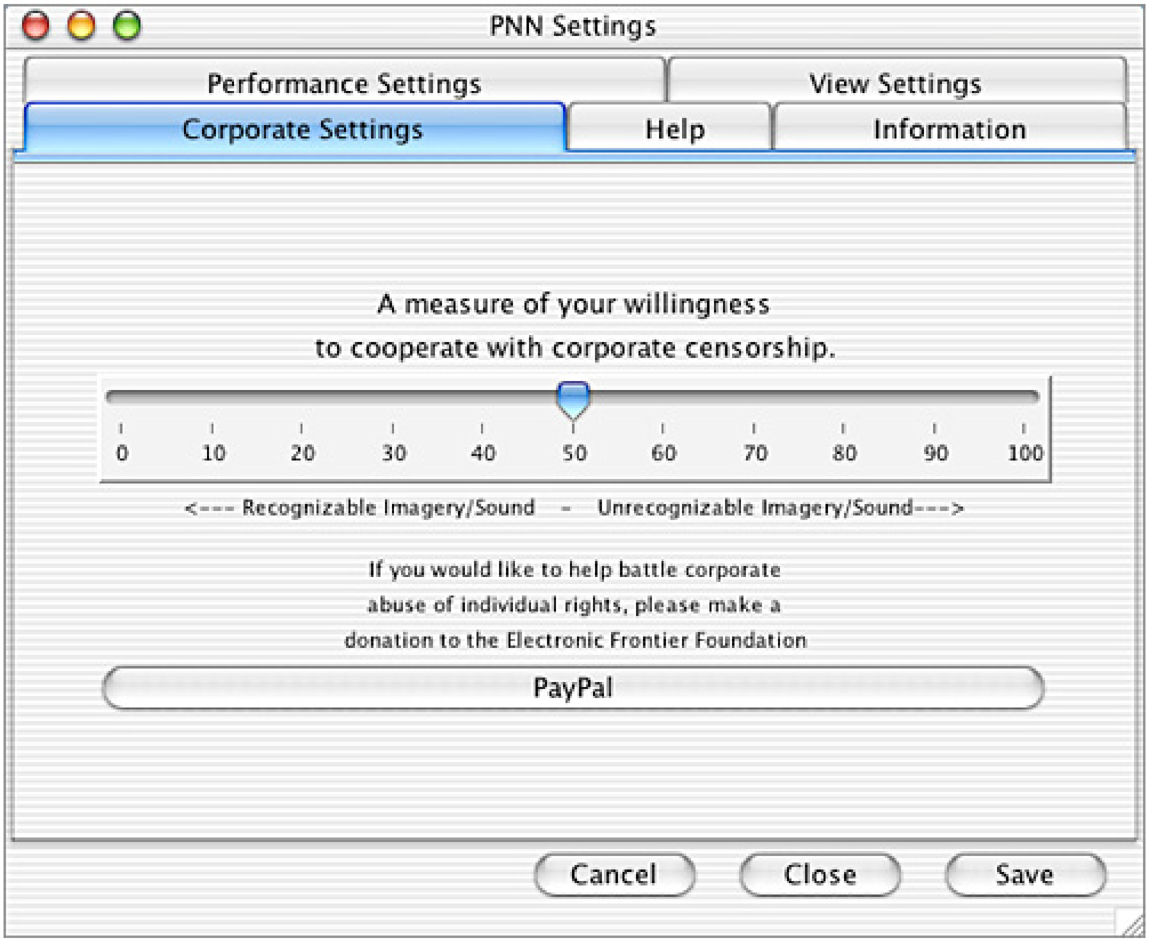
□ View Boundary Boxes (ctrl + b)
□ View Wireframe (ctrl + w)
□ View Light Sources (crtl + l)
□ View Sample Vector (ctrl + v)
☑ View Shadows (ctrl + d)
☑ View Atmosphere (ctrl + a)
□ Full Screen Display (ctrl + f)

Adjust Atmospheric Density

Fig. 27. The PNN Settings "View Settings" tab.

The "View Sample Vector" setting displays a line from the shadow casting light source to the origin of world space for each of the samples used to calculate shadows. As previously discussed PNN is only using one sample to create shadows at this time, due to speed issues. In the future, with faster processors, an increase in the number of samples is feasible. By default, when PNN is started, “View Shadows” and “View Atmosphere” are enabled. As the names suggest, “View Shadows” enables the display of shadows and “View Atmosphere” enables the display of the atmosphere or fog. The final checkbox setting is “Full Screen Display.” This setting toggles the display between full screen and a windowed environment. At the bottom of the PNN Settings window is a slider control that allows the viewer to adjust the density of the atmosphere or fog. In testing the PNN application on different systems this researcher realized that, depending on the video graphics card installed on the computer, the density of the fog varied greatly. On some machines, despite a moderate default fog density, the objects in the scene would initially be obscured. This slider control allows the viewer to tailor the density to the peculiarities of her system.

The “Corporate Settings” tab (Fig. 28) is where the viewer can control the application’s response to issues of digital rights management. Despite this researcher’s opinions regarding this issue, the researcher also believes that the issue is something that has to be decided on an individual basis. It is the responsibility of the individual to decide whether their use of cultural material violates the rights of those who produced it. This decision should not be put upon the software producer and certainly should not be left to the corporate interests whose goal is to profit by restricting access to these resources. This researcher’s solution to the problem is to provide a slider control that allows the viewer to tune the presentation to match his or her own personal beliefs. One extreme of the scale obscures the imagery and sound to the point it is unrecognizable.



A measure of your willingness
to cooperate with corporate censorship.

If you would like to help battle corporate
abuse of individual rights, please make a
donation to the Electronic Frontier Foundation

Fig. 28. The PNN Settings "Corporate Settings" tab.

Copyright law supports the idea that if there is no recognizable relationship to the original object, then a violation of copyright has not occurred. This is the safest route to avoid the issue. The opposite extreme rejects the idea that anyone owns the content being viewed, that it is a part of the culture-at-large, and therefore the imagery and sound can remain recognizable. In addition to the censorship slider the researcher has included a button that encourages the viewer to become involved with this issue and contribute funds to the Electronic Frontier Foundation. The distortion of the imagery controlled by the censorship slider will rely on the future implementation of the modeling functions of the PNN Server application, and therefore neither of the functions within the “Corporate Settings” tab is enabled. They are present to indicate future functionality that will be implemented as the rest of the program is developed.

The majority of options available within the PNN Settings dialog are tied to the PNNParameters class. This class maintains a properties file that is stored within the application bundle of an OSX application or in the “PNN\Resources” folder on Windows. Every time a setting is changed and the “Save” button is pressed the properties file is updated to reflect the state of the application. If the application is quit, upon re-launch, the PNN application will read the properties file and launch using the settings saved within the properties file. For example, if the viewer adjusts the atmospheric density in the “View Settings” tab, and subsequently presses the “Save” button, atmospheric density will be adjusted. Once the viewer quits the application and restarts it later the previous atmospheric density adjustment will be used in the current display.

EVALUATION

A work of art is notoriously difficult to evaluate objectively. PNN is no different in this regard. The response of the viewer is dependent on many factors outside the artist’s control, and “Applications as artwork” is not a large category against which one may make comparisons. Despite these limitations this researcher has sought to provide an environment that maximizes the potential for the creation of an aesthetic experience on behalf of the viewer and believes that this effort was successful. The function of the program can be evaluated more objectively. In addition to the equipment at the researcher's disposal several other graduate students at Texas A&M have evaluated PNN. Michael Mistrot [40] reports that on an Apple G4 700 MHz with a GeForce2MX graphics card the application works fine, though it slows considerably when trying to stream video onto the objects while in "online" mode. This is to be expected considering that the PNN web server is presently running on a standard cable modem with an outbound bandwidth limitation of 12Kb/second. Until the server can be provided greater bandwidth it is recommended that PNN be run in local mode only. Jeremy Sternberg [41] reported similar behavior using an Apple G4 Ti portable. This researcher plans to address this problem in the next few months. The Windows version has received little testing. This researcher is aware of several problems with the Windows version, probably related to the freeing of resources when the application switches from one movie file to the next. The application runs fine until the next movie is requested. The majority of the time this causes the application to quit. This will require more research to identify why QuickTime is not properly disposing of previous movies on this platform. One other problem is related to the display of shadows. On some computers the shadows display as white when fog is enabled. When fog is disabled the shadows return to their proper black color. This appears to be a hardware specific problem that must be addressed in the coding of the application. This will be addressed at a future date. Other than these problems and the inherent limitations imposed by the absence of the PNN server application PNN functions as the researcher had hoped.

CHAPTER V

CONCLUSION AND FUTURE WORK

PNN uses computer technology in the form of an application that takes advantage of digital systems' ability to manipulate information quickly and efficiently. PNN creates an immersive navigable space and presents that space in an interpretive aesthetic manner. The PNN space connects the viewer to an intrinsic part of our cultural identity and allows the viewer to re-contextualize the content. Internet technology, which can encourage dialogs independent of profit motives, enables this connection. This fundamental freedom of expression is the basis around which PNN exists as an artwork. PNN uses principles of digital sampling to enable the viewer to take control of media and create new meaning from old.

IMPLICATIONS FOR FUTURE RESEARCH

During the process of developing this application this researcher has been inspired by the many opportunities for development fostered by the technologies and methods employed. The most immediate direction, of course, is the development of the PNN server application. The digital mapping of video data to a representative three-dimensional form is an extremely exciting avenue of research. It will be interesting to see the results of this investigation and the expected differentiation based on content. The implementation of the server application will also allow this researcher to complete the text based video search capabilities of the project. Not only will this feature add to the efficacy of the PNN client application, but it also might provide an avenue for future research into the relationships of language, image and form. Patterns of language might be made apparent by this study and studies of term frequency and color-mood relationships might lead to new forms of expression. This researcher also envisions developing the application in a direction that would enable viewers to save episode objects representing their favorite searches and sharing these findings via a web based communal network. This activity would strengthen the form-giving component of PNN and possibly create a stronger relationship between the viewer and the search results. Another possibility is the adaptation of PNN to the traditional gallery installation environment. It would be possible to connect the PNN virtual camera to analog devices such as proximity or motion detectors, or joysticks, to create an environment that more fully immerses the senses. Similarly, PNN should be adaptable to the various virtual reality systems that are becoming available. An exciting aspect of this prospect is PNN’s capability to tie the local space of the installation to the global space of the Internet. One could interface PNN with a rapid prototyping system capable of producing physical objects from the models generated by search results. The concept of a gallery space as a machine for creating content induces a myriad of conceptual/physical developmental possibilities.

Eventually the exploration of topics other than the evening news is expected. An investigation into the “sculptural” characteristics of different types of television programming might be one such direction. An analysis of the “form” of local television programming culled from geographically disparate parts of the country or world might prove interesting. One option being considered is the possibility of releasing PNN, or some portion thereof, as an open source project. As an open-source project PNN might be more rapidly developed than it can be with a single author. Additionally the programming talents of a group of authors with diverse experience might lead to unexpected and exciting directions. Fundamentally this researcher’s interest lies in the giving of tangible form to abstract ideas, and the framework provided by PNN provides the foundation for this work as a communal experience, rather than as an independent experiment.

NOTES

1. {Digidesign #37}

2. Digital Performer, Mark of the Unicorn, Inc., Cambridge, MA.

3. Reason, Propellerhead Software, Stockholm, Sweden.

4. Max/Msp, Cycling'74, San Francisco, CA.

5. Michael Mandiberg, *AfterSherrieLevine.Com* [online] (Michael Mandiberg, 2001 [accessed December 2001]); http://www.aftersherrielevine.com.

6. Oliver Grau, "Historic Spaces of Illusion," in *Virtual Art: From Illusion to Immersion* (Cambridge, MA: MIT Press, 2003).

7. The diegesis can be thought of as the imaginary time and space of cinema.

8. Colin Green, Matthew Fuller, Simon Pope, *Web Stalker* [online] (I/O/D, 1997 [accessed 10 February 2003]); http://www.backspace.org/iod/.

9. Mark Napier, *Shredder* [online] (Mark Napier, 1998 [accessed 24 April 2002]); http://www.potatoland.org/shredder.

10. Maciej Wisniewski, *Netomat* [online] (Maciej Wisniewski, 1999 [accessed 24 April 2002]); http://www.netomat.net/began.html.

11. Natalie Jeremijenko, "Live Wire," (Stanford, CA: 1997).

12. John Klima, *Earth* [Installation, online] (2001 [accessed 10 February 2003]); http://www.cityarts.com/earth/.

13. Max/Msp [4].

14. Nato.0+55+3d Modular, Netochka Nezvanova.

15. Auto-Illustrator Ver. 1.0, Signwave UK, London, England.

16. Lev Manovich, *The Language of New Media* (Cambridge, MA: MIT Press, 2001).

17. Guy Debord, "The Theory of the Derive," in *Situationist International Anthology*, ed. Ken Knabb (Berkeley, CA: Bureau of Public Secrets, 1981).

18. Gregory Ulmer, "The Object of Post-Criticism," in *The Anti-Aesthetic: Essays on Postmodern Culture*, ed. Hal Foster (Port Townsend, WA: Bay Press, 1983).

19. Lev Manovich, "Navigable Space," in *The Language of New Media* (Cambridge, MA: MIT Press, 2001).

20. George Mather, "Object-Oriented Models of Cognitive Processing," *Trends in Cognitive Sciences* **5**, no. 5 (2001).

21. Karen A. Cerulo, "The Rest of the Story: Sociocultural Patterns of Story Elaboration," *Poetics* **28**, no. 1 (2000).

22. Lawrence Lessig, *The Law of the Horse: What Cyberlaw Might Teach* (1999 [accessed February 2003]); http://cyber.law.harvard.edu/works/lessig/finalhls.pdf.

23. Maya, Alias|Wavefront, Inc., Toronto, Ontario, Canada.

24. For more information, see: http://www.machinima.org/

25. Open Graphics Library Ver. 1.2, Silicon Graphics, Inc., Mountain View, CA.

26. Direct3D, Microsoft Corporation, Inc., Redmond, WA.

27. DirectShow, Microsoft Corporation, Inc., Redmond, WA.

28. Realmedia, Real Networks, Inc., Seattle, WA.

29. QuickTime Ver. 6.0, Apple Computer, Inc., Cupertino, CA.

30. OpenGL Utility Toolkit Ver. 3, Silicon Graphics, Inc, Mountain View, CA.

31. Simple DirectMedia Layer, Sam Latinga, Mattias Engdegard, Martin Donlon,Max Watson, Stan Shebs, Darrel Walisser, et al.

32. Open Scene Graph, Robert Osfield, Don Burns, Graeme Harkness, Neil Salter, Brende Johansen, et al.

33. wxWindows, Julian Smart, Robert Roebling, Stefan Csomor, David Webster, Roberto Alsina, Vadim Zeitlin, et al.

34. Qt, Trolltech, Inc., Oslo, Norway.

35. Java, Sun Microsystems, Inc., Santa Clara, CA.

36. OpenGL for Java (GL4Java), Jausoft, Open Source under LGPL.

37. QuickTime for Java, Apple Computer, Inc., Cupertino, CA.

38. Ben Humphrey, *Time Based Movement* [online] (2001 [accessed January 2002]); http://www.gametutorials.com/Tutorials/opengl/OpenGL_Pg1.htm.

39. Nate Robbins, *Real-Time Soft Shadows* [online] (Silicon Graphics, Inc., [accessed January 2001]); http://www.sgi.com/software/opengl/examples/more_samples/.

40. Michael Mistrot, e-mail communication, 3 February 2003.

41. Jeremy Sternberg, e-mail communication, 26 February 2003.

BIBLIOGRAPHY

Auto-Illustrator Version 1.0. Signwave UK, London, England, 2000 <http://www.auto-illustrator.com>.

Cerulo, Karen A. "The Rest of the Story: Sociocultural Patterns of Story Elaboration." *Poetics* 28, No. 1, 21-45 (2000).

Debord, Guy. "The Theory of the Derive." In *Situationist International Anthology*, edited by Ken Knabb (Berkeley, CA: Bureau of Public Secrets, 1981).

Digital Performer. Mark of the Unicorn, Inc., Cambridge, MA, 2001 <http://www.motu.com>.

Direct3D. Microsoft Corporation, Inc., Redmond, WA, 2002 <http://www.microsoft.com/directx/>.

DirectShow. Microsoft Corporation, Inc., Redmond, WA, 2002 <http://www.microsoft.com/directx/>.

Grau, Oliver. "Historic Spaces of Illusion." In *Virtual Art: From Illusion to Immersion* (Cambridge, MA: MIT Press, 2003) p.25.

Humphrey, Ben. *Time Based Movement* [online]. 2001 [accessed January 2002]. http://www.gametutorials.com/Tutorials/opengl/OpenGL_Pg1.htm.

Java. Sun Microsystems, Inc., Santa Clara, CA, 1995 <http://java.sun.com/>.

Jeremijenko, Natalie. *Live Wire*, Stanford, CA, 1997.

Klima, John. *Earth* [Installation, online]. 2001 [accessed 2003 10 February]. http://www.cityarts.com/earth/.

Lessig, Lawrence. *The Law of the Horse: What Cyberlaw Might Teach* 1999 [accessed February 2003]. http://cyber.law.harvard.edu/works/lessig/finalhls.pdf.

Mandiberg, Michael. *AfterSherrieLevine.Com* [online]. Michael Mandiberg, 2001 [accessed December 2001]. http://www.aftersherrielevine.com.

Manovich, Lev. *The Language of New Media*. (Cambridge, MA: MIT Press, 2001).

Manovich, Lev. "Navigable Space." In *The Language of New Media* (Cambridge, MA: MIT Press, 2001) p.251-252.

Mather, George. "Object-Oriented Models of Cognitive Processing." *Trends in Cognitive Sciences* **5**, No. 5, 182-184 (2001).

Matthew Fuller, Colin Green, Simon Pope. *Web Stalker* [online]. I/O/D, 1997 [accessed 10 February 2003]. http://www.backspace.org/iod/.

Max/Msp. Cycling'74, San Francisco, CA, 2001 <http://www.cycling74.com>.

Maya. Alias|Wavefront, Inc., Toronto, Ontario, Canada, 1998 <http://www.aliaswavefront.com/>.

Napier, Mark. *Shredder* [online]. Mark Napier, 1998 [accessed 24 April 2002]. http://www.potatoland.org/shredder.

Nato.0+55+3d Modular. Netochka Nezvanova, 2000 <http://www.eusocial.com>.

OpenGL for Java (GL4Java). Jausoft, Open Source under LGPL, 1999 <http://www.jausoft.com/gl4java/>.

OpenGL Utility Toolkit Version 3. Silicon Graphics, Inc, Mountain View, CA, 1994 <http://www.opengl.org/developers/documentation/glut.html>.

Open Graphics Library Version 1.2. Silicon Graphics, Inc., Mountain View, CA, 1992 <http://www.opengl.org>.

Open Scene Graph, Robert Osfield, Don Burns, Graeme Harkness, Neil Salter, Brende Johansen, et al., 2001 <http://www.openscenegraph.com/>.

Pro Tools. Avid Technology, Inc., Daly City, CA, 1996 <http://www.digidesign.com>.

Qt. Trolltech, Inc., Oslo, Norway , 1999 <http://www.trolltech.com/>.

QuickTime Version 6.0. Apple Computer, Inc., Cupertino, CA, 2002 <http://www.apple.com/quicktime/>.

QuickTime for Java. Apple Computer, Inc., Cupertino, CA, 2002 <http://www.apple.com/quicktime/>.

Reason by Propellerhead Software, Stockholm, Sweden, 2000 <http://www.propellerheads.se>.

Realmedia by Real Networks, Inc., Seattle, WA, 2001 <http://www.realnetworks.com>.

Robbins, Nate. *Real-Time Soft Shadows* [online]. Silicon Graphics, Inc., [accessed January 2001]. http://www.sgi.com/software/opengl/examples/more_samples/.

Simple DirectMedia Layer. Sam Latinga, Mattias Engdegard, Martin Donlon,Max Watson, Stan Shebs, Darrel Walisser, et al., 2001 <http://www.libsdl.org>.

Ulmer, Gregory. "The Object of Post-Criticism." In *The Anti-Aesthetic: Essays on Postmodern Culture*, edited by Hal Foster (Port Townsend, WA: Bay Press, 1983) p. 94.

Wisniewski, Maciej. *Netomat* [online]. Maciej Wisniewski, 1999 [accessed 24 April 2002]. http://www.netomat.net/began.html.

wxWindows. Julian Smart, Robert Roebling, Stefan Csomor, David Webster, Roberto Alsina, Vadim Zeitlin, et al., 1999 <http://www.wxwindows.org/>.

APPENDIX A

APPLICATION

Attached to this document are copies of the Windows and Macintosh OSX versions of the PNN client application.

1. Windows version 0.9a
2. Macintosh OSX version 0.91a

Please check http://www.publicnewsnetwork.net for more recent versions, and detailed installation instructions.

VITA

Jack Eric Stenner
11411 Sagecherry St.
Houston, TX 77089
jack@jigglingwhisker.com

Education

M.S. in visualization sciences	Texas A&M University, 5/03
Bachelor of environmental design	Texas A&M University, 5/85

Research Interests

Digital Mapping of Content into Form
Manipulation of Cultural Constructs
Critical Issues in Private vs. Public
Virtual Architecture/Virtual Reality

Employment

Texas A&M University, College Station, TX	Asst. Professor (GA), 2002
Frank Stella Mural, Moore's School of Music, Houston, TX	Artist, 1997
Purse Building Studios, Houston, TX	Exhibitions/Construction, 1993 – 2000
Independent Artist, Houston, TX	Artist, 1991 – present
Falick-Klein Partnership, Houston, TX	Architect, 1989 – 1991
Page Southerland Page, Houston, TX	Asst. Architect, 1987 – 1989
Pierce Goodwin Alexander, Houston, TX	Asst. Architect, 1985 – 1987

Honors

Third Place Award, "The Big Show," 1998, Claudia Gould, Artists Space, NY
Creative Artist Grant-Cultural Arts Council of Houston/Harris County, 1997
First Place Award, Visual Arts Alliance Annual Juried Exhibition, 1996
Award of Merit-Coconut Grove Art Festival, Dr. Christina Orr-Cahall, Director, Norton Gallery of Art, Juror, 1994
John E. O'Brien Scholarship in Environmental Design, 1985
Edward J. Romieniec Award for Outstanding Senior Designer, 1985

Activities

Juror, 1st Annual "Extremely Shorts," Aurora Picture Show, Houston, TX, 1999
Board Member, Buffalo Bayou Artpark, Houston, TX, 1998 - 2000
Board Member, Chair Artist Directors, Art League of Houston, Houston, TX, 1995-1997
Owner/Founder of Alternative Art Space, Purse Building Studios, Houston, TX, 1993 - 2000
Registered Architect, State of Texas, 1991 - present