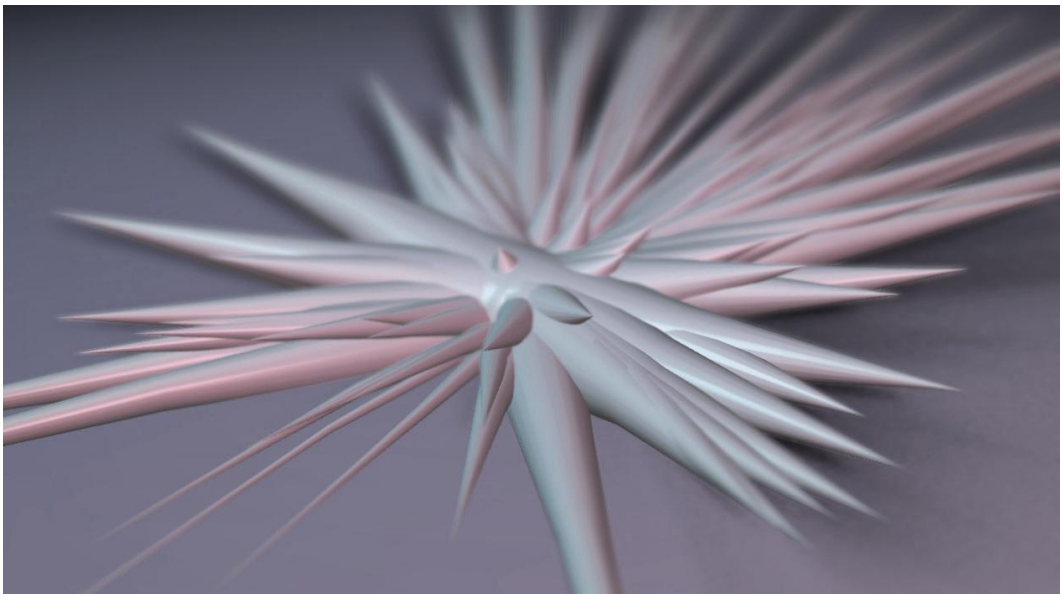
Chapter 1

Introduction



1.1 Overview

Media artists and engineers engage in a wide variety of techniques for making art. These techniques may include working with physical media or devices, apply existing tools, or developing their own tools. In the 1950s scientists such as Michaell Noll,

Ken Knowlton and Freider Nake developed the first digital images, and the computer presented itself as a new form for making art [Dietrich, 1986]. Since then the variety of forms of expression using digital techniques has expanded rapidly. Some digital artists, such as the Algorists, began making art by writing computer code expressing a particular aesthetic. In other areas, artist-engineers such as William Reeves, Craig Reynolds and Karl Sims developed code to explore naturalistic behaviors through simulation. Artists began working with databases, computer interfaces, simulations, and generative systems, resulting in many new forms.

As these new fields emerged, many artists and engineers began making their own tools, including the development of integrated frameworks for shared use in research and education. By the mid 1990s, graphics languages such as GINO, PHIGS, IRIS Inventor, OpenGL, and DirectX provided generic interfaces for expressing primitive graphic elements to machines [Dam, 1998]. Languages such as Java, Flash and Processing expanded the use of text-based languages to make them more adept at expressing visual ideas, and allowing media artists greater access to learning how to program. Commercial applications such as Maya, 3D Studio MAX, and Houdini, have allowed digital artists to develop sophisticated work flows for building complex digital worlds, while tools such as Max/MSP, VVVV, Quartz Composer, and Soundium have allowed artists to work with visual and auditory media in installations, exhibits, and live performances.

While there is a great deal of overlap in these forms of expression, over time tools for artists have specialized to serve the needs of each group. As Linda Candy observes,

any tool comes with certain *inherent constraints* which limit the range of expression [Candy, 2007]. These may be distinguished from the creative constraints the artist imposes in resolving a conceptual problem, since the inherent constraints of the tool limit creative freedoms beyond the control of the artist. Chapter 1 explores the specialized constraints of existing tools for media artists, focusing specifically on systems for representing shape, form and behavior, areas of particular interest to the author.

Although every tool has constraints, the question addressed in this work is: Whether the specific constraints and boundaries between current tools for digital artists are necessary or if they are a by-products of the various communities which have developed over time? Are our current technical constraints really inherent constraints of the digital media itself, or are they constraints resulting from the evolution of goals in creative communities? If the former case is true, it would indicate some real distinction in media between different forms of artistic expression. If the latter is true, then it should be possible to develop tools which combine the expressive capabilities of many different tools, reducing work and allowing for greater cross-over in techniques between artistic communities.

This question is addressed by examining current practices by media artists, and through the development of LUNA, a novel visual language for creative expression which integrates these disparate practices. The dimensions of technique investigated here, and present in varying degrees in currently existing tools, are not intended to be

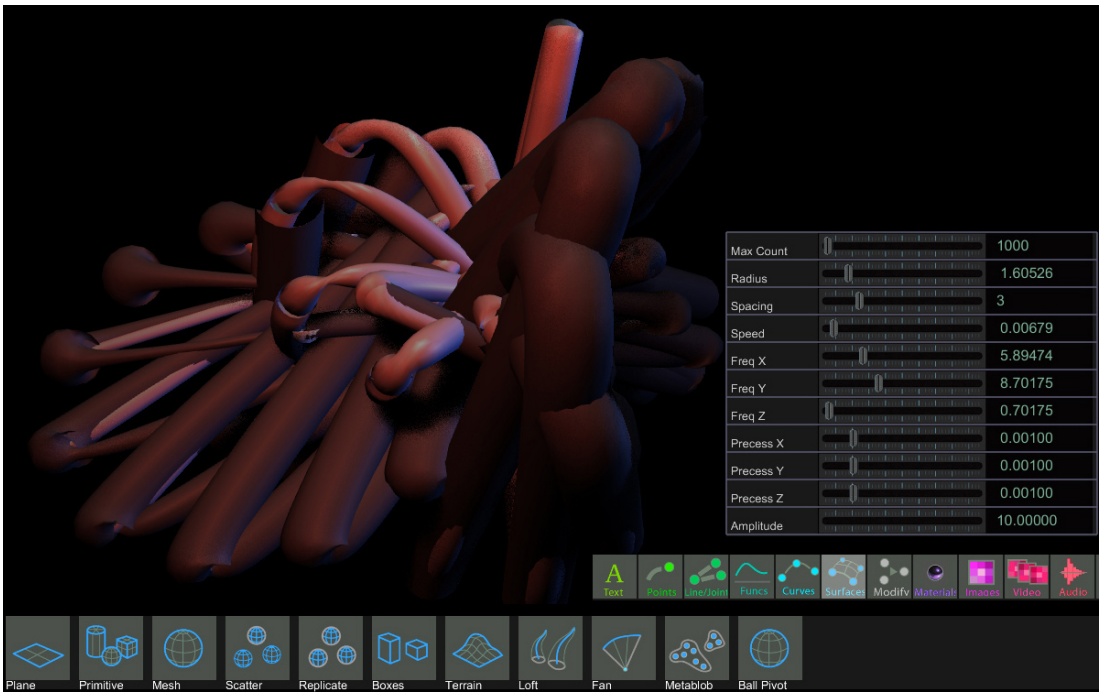
an exhaustive map of creative practice, but were selected as a representative set of some essential, recurrent issues for media artists.

The dimensions explored in this thesis include:

1. Programming and Language
2. Modality and Media
3. Live Performance and Computation
4. Motion, Dynamics and Autonomy
5. Structure and Surface
6. Image and Idea

The motivation for these particular dimensions are based on issues that touch several different groups in digital and media arts, and are explored in detail in Chapters 5 and 6. The dimension of Programming and Language addresses the need for low level rule-making in text-based languages in comparison to the desire to quickly mix ideas in visual languages for non-programmers. The dimension of Modality and Media addresses the desire of media artists to work with many different media, including three-dimensional form, images, video, audio, databases, and bringing these into computers from the real world through various devices. Live Performance and Computation addresses the distinction between real-time systems for live performance and offline computation for high fidelity structures and imagery. Motion, Dynamics and Autonomy addresses the desire of media artists to control the dynamic behavior, motion and growth of digital forms, while Structure and Surface addresses the need of certain artists to control and

manipulate the visual appearance, surface and style of these forms. Finally, Image and Idea addresses the relationship between images and words, i.e. the ability of machines to look at digital images as semantic objects.



1.2 Summary of Contributions

The primary contribution of this thesis is the development of LUNA, a visual data flow language for exploring the different dimensions in creative expression described above. LUNA was developed over a period of three years, with influences from several other frameworks. Conceptually, the visual language for LUNA is strongly influenced by the board game Scrabble. LUNA is also influenced by other animation languages such as ConMan, Stephen May's AL (Animation Language), Processing, and Soundium.

The novel aspects of LUNA are the contribution of a real-time language for procedural modeling, rendering by deferred shading and a natural, minimal graphical interface for expressing structure and behavior.

Specific contributions of this thesis are as follows:

1) LUNA allows for creative expression along the various dimensions described above, demonstrating that it is possible to develop tools which integrate these creative possibilities into a single framework. These dimensions, and how LUNA addresses them, are explored in Chapters 5 and 6.

2) The graphical interface for LUNA enables non-programmers to rapidly develop complex structures and behaviors in dataflow diagrams. The graphical interface is also distinguished from other commercial packages such as Maya and Houdini in that its method of interaction is based on media rather than work flow. Top-level tool bars in LUNA express structures, while second-level tool bars express behavior. This arrangement, and its benefits, are described in Chapters 3 and 5.

3) While high level behaviors can be combined in the interface, LUNA allows programmers to develop new low level behaviors by authoring new nodes in C++ (programming language). This method of node development relies on the LUNA API to provide a common, shared framework for expressing geometric, image, and object structures while allowing the author to generate new behaviors as desired. This is described in detail in Chapter 5.

4) The procedural language of LUNA allows artists to build and express complex organic structures. This language is based on manipulating and generating multiple objects with standardized, yet expandable, memory structures for representing discrete geometry. (This non-standard representation does not use C++ class variables to store data, but instead uses variable length buffers with named semantics). The language of LUNA is described in Chapter 4.

5) LUNA is shown to be measurably faster than Houdini, a commercial application for procedural modeling, for a procedural reference model developed for testing. The reference model, a woven sphere, cannot be easily created by other modeling techniques, such as physical simulation or hand manipulation of polygonal models, and is presented as a novel object for testing procedural frameworks. Performance results for LUNA are found in Chapter 5.

6) LUNA incorporates a deferred shading engine. Commonly found in game engines, and unique to frameworks for media artists, deferred shading allows for real-time shadows, depth-of-field, multiple light sources, and other advanced rendering techniques, discussed in Chapter 5. The deferred shading engine is capable of running custom shaders on multiple monitors with any number of GPUs and displays running from a single system.[1]

[1] Mutliple display rendering was developed in MINT/VFX based on NSF IGERT work by the author on cluster rendering for the Allosphere, an immersive 30 foot diameter near-spherical collaborative environment for media artists and scientists at the University of California Santa Barbara

7) Profiling tools, which allow artists to tune the computational resources in LUNA, are also incorporated into the framework and provide real-time feedback for live performances. These profiling tools are described in Chapter 5.

8) LUNA allows media artists to explore, mix, and combine a range of different behavioral systems. Some of the current components in LUNA include particle systems, fluid simulation, spiroids (oscillators), spring-systems and others. These can be combined in the interface with high level operators which merge these objects to create more complex behaviors. The behavioral aspects of LUNA are described in Chapter 6.

9) LUNA can express a range of different media structures. These currently include points, trees, curves, surfaces, images and materials, and may be extended in the future to include volumes, audio, video, databases, and networking. Structures in LUNA may be procedurally generated, such as trees, or may be loaded from static data, such as meshes. Currently available structures in LUNA are described in Chapter 6.

10) A theory for digital semantics is presented in Chapter 6 which considers the difference between the digital model and its expression. Based in part on the work of Jorg Shirra, this theory motivates new directions in tools for media arts that consider images and words (or semantics).

11) Experiments by the author, in Chaper 6, which combine hand-sketched drawings with digitally generated compositions show that rule-less images expressed by drawing and photography cannot be completely replaced by digital models, and are thus important elements for visual synthesis into tools for media artists.

Overall, LUNA is presented as a novel language and an open system for media artists. The development of LUNA focuses first on providing a flexible modeling language, which is then populated with a number of specific behaviors or *nodes*. The current nodes were developed according to the author's own interests in sculptural form, geometry and behavior, and to demonstrate certain capabilities across different media (such as interactive shader manipulation). In the future, it is hoped that the capabilities and available nodes in LUNA will be further expanded by the artistic community.

1.3 Criteria for Evaluation

Evaluation of this work is based on a number of different metrics. Overall, the final test of any creative tool is its future adoption by the artistic community. Ideally, to meet the goals set out by LUNA it would be used by artists in several communities to demonstrate the ability to bridge different practices. Prior to adoption, LUNA is evaluated through a number of collaborative projects it has been applied to, by quantitative performance metrics, through the visual results it achieves, and by criteria established by a 2005 NSF Workshop on Creative Support Tools [Shneiderman et al., 2005].

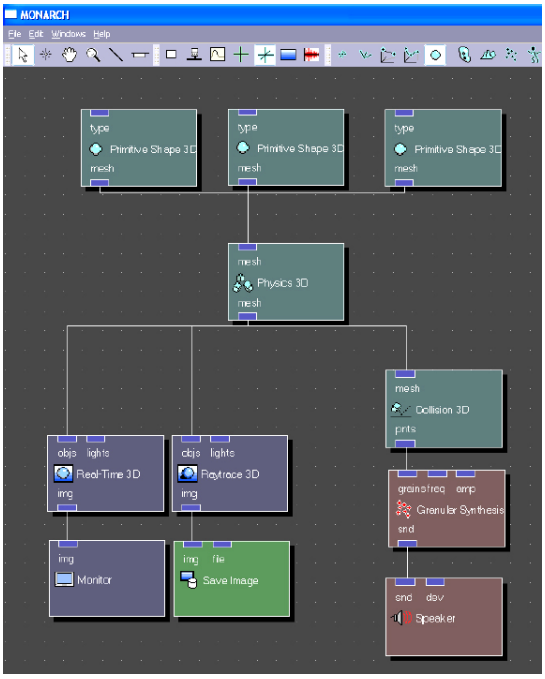
The primary evaluation of LUNA as a tool for different communities is based on the result of its application to a number of collaborations and inter-disciplinary projects described in Chapter 4. The creative dimensions discussed above are demonstrated in LUNA by showing the system is capable of expressing at least two points along each of the dimensions described. These results represent the abilities of the system

according to the particular ways that media artists frame their tools. For example, results in Chapter 6 show that artists can create structural objects such as trees with a realistic appearance in LUNA, but can also manipulate and distort this structure for other aesthetic ends, or to re-contextualize the tree to be applied to fractal structures in biological modeling. Thus, these dimensions represent not only a parametric change in the model, but conceptual shifts in how the artist conceives of using the tool. In this way, LUNA is demonstrated to meet several different aesthetics of interest to media artists.

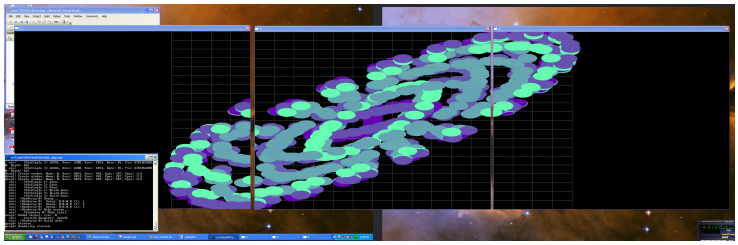
To evaluate LUNA as a tool for supporting creative expression, metrics are introduced from the 2005 NSF Workshop on Creative Support Tools. These metrics are based on the observation that creative tools in various disciplines ideally have 1) low thresholds, 2) high ceilings, and 3) wide walls. Low threshold means "that the interface should not be intimidating, and should give users immediate confidence that they can succeed," high ceiling means that "the tools are powerful and can create sophisticated, complete solutions," and wide walls means that "creativity support tools should support and suggest a wide range of explorations [Resnick et al., 2005]." These metrics, while not quantitative, provide a subjective criteria for evaluating the ability of LUNA to support creativity. The relative relationship between LUNA and other tools with regard to these metrics is explored in Chapter 6.

The computational performance of LUNA is evaluated using a novel procedural reference model, and compared to both Houdini and a baseline model in OpenGL.

Although no user study is developed for the LUNA interface, this reference model is also used to evaluate the ability to perform interface tasks in relation to Houdini.



(a)



(b)

Figure 1.1: Influences on the systems aspects of LUNA include a) Monarch, for interface aspects, b) MINT/VFX, for events and multi-screen rendering, and GameX for graphics architecture (not shown).

1.4 History and Development

The development of LUNA occurred in a series of stages. The early stages of LUNA were influenced by several other media systems frameworks. MINT, a collaboration among graduate students participating in an NSF IGERT project from 2005-2007 at the University of California Santa Barbara contributed ideas regarding the event system and multi-display aspects of LUNA, Figure 1.1a [Hoetzlein and Adderton, 2009]. A prototype interface for media artists, Monarch, was developed by R. Hoetzlein and Jorge Castellanos in 2006 to experiment with creative interactions, Figure 1.1b, although the system had no internal capabilities to simulate objects [Hoetzlein and Castellanos, 2006]. GameX, an earlier project used to co-found the Game Design Initiative at Cornell University in 2002, influenced the rendering system of LUNA [Hoetzlein and Schwartz, 2005].

The next key stage in LUNA, developed through 2009, was the creation of the visual dataflow language. Based on design sketches by the author since 1998, these sketches suggest a method by which different geometric structures may be combined to create new functional structures. Similar to procedural modeling in Houdini or Xfrog, a key difference is that these objects are envisioned as dynamic, complex, high-level systems that may be combined and connected while details are revealed only on demand. The board game Scrabble, in which combinations of single-letter tiles can create a wide range of expressive power, provided inspiration for the layout of the language itself. Each object assumes a basic behavior in which no other information

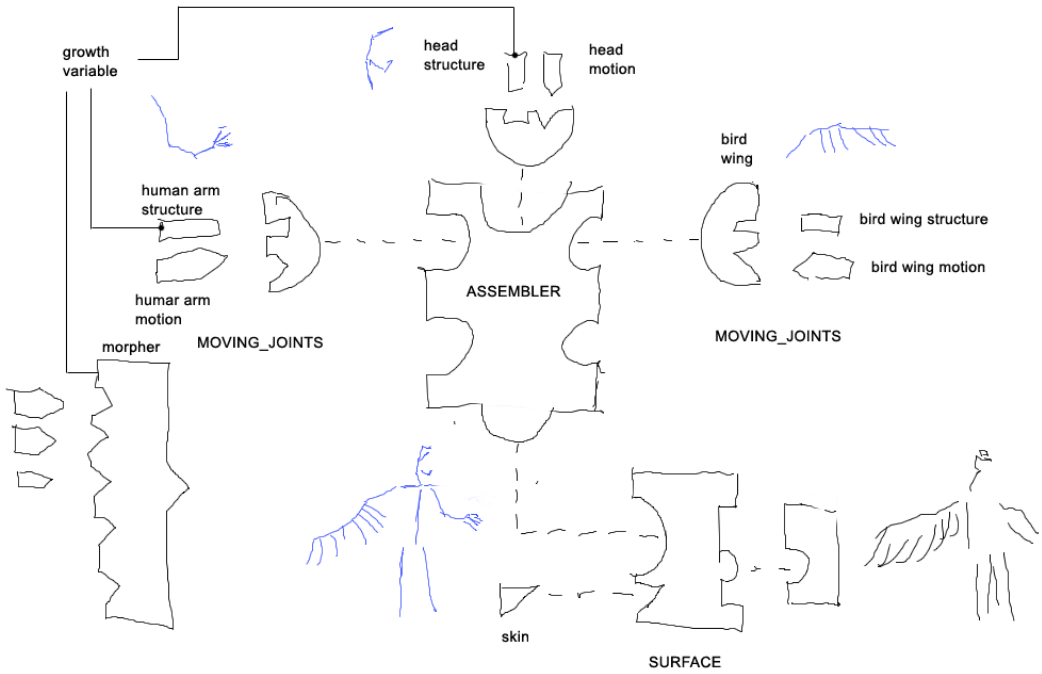
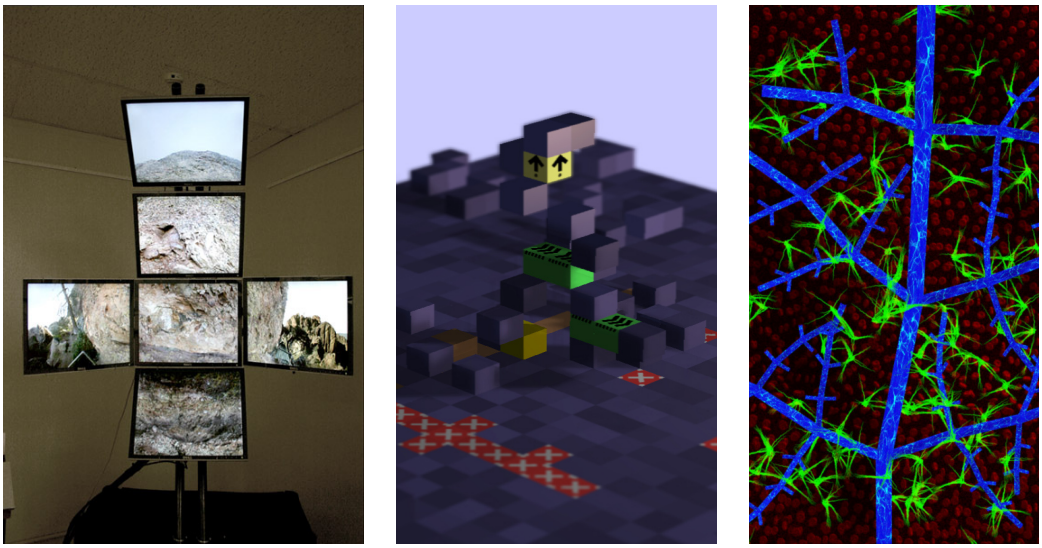


Figure 1.2: Sketches of the LUNA visual language.

is needed for construction, making the system usable by non-programmers. The visual dataflow language is described in detail in Chapters 3 and 4.

Between 2007 and 2010, LUNA was used to develop a number of creative and experimental collaborations. These include *Presence*, with Dennis Adderton and Jeff Elings (2008), an interactive panoramic high resolution display of natural scenes on a custom six-screen display exhibited at the University of California Davidson Library, Figure 1.3a, and *Blocks*, a massive virtual world of dynamic cubes developed with Mark Zifchock, Abram Connelly and Marty White (started in 2003), Figure 1.3b. LUNA was also used in a scientific collaboration with Mock (Panuakdet) Suwannatat and Tobias Höllerer, based on astrocyte imaging results by Gabe Luna, Geoffrey Lewis, and Steve

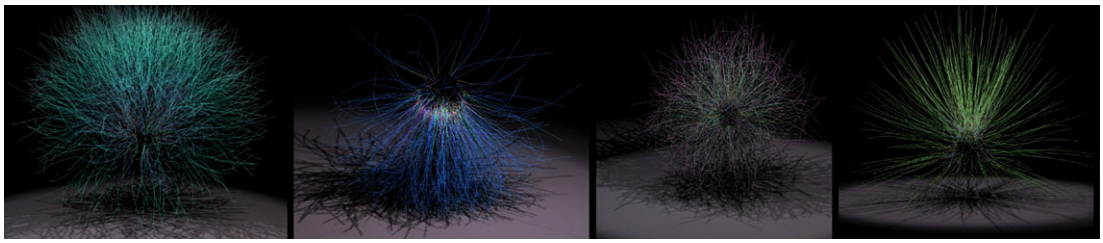


(a) (b) (c)

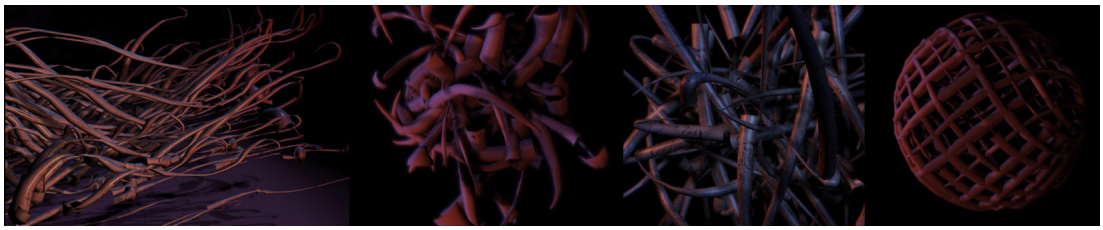
Figure 1.3: Collaborative projects created with LUNA include a) *Presence*, with Dennis Adderton, b) *Blocks*, with Mark Zifchock, and c) *Synthetic Rendering* with Mock Suwannata and the Neuroscience Research Institute (c) 2010. See text for project descriptions.

Fisher (Neuroscience Research Institute, Univ. of California Santa Barbara), and B.S. Manjunath (Dept. of Electrical and Computer Engineering). This was a project to explore *synthetic rendering*, the use of digital modeling to reproduce and render microscopic structures of retinal astrocyte images, showing the systems capabilities in simulating complex models (Figure 1.3c).

The final stage of LUNA (in the development of this dissertation) has been to improve the interface and expressiveness of the system to handle complex structural models. The tree object was added in 2010, as well as a pipeline for material and surface appearances using Cg shaders. Profiling tools were introduced to enable artists to interactively evaluate the performance of the system. The graphical user interface



(a)



(b)

Figure 1.4: Procedural modeling of various organic forms in LUNA based on different behaviors and structures.

was extended to include elements for nested 2D and 3D views, tool bars, scroll bars, and interactive sliders. The visual results of Figure 1.4, show that LUNA is capable of a range of behaviors as well as being able to model complex systems, Figures 1.5.

The most recent additions to LUNA include nodes that synthesize images from image sets (collections of images), using hand-sketched drawings in combination with generative composition. This has resulted in a number of aesthetic experiments and observations that reveal potentially novel art forms, see Figure 1.6,. Artists seeking to work with the hand drawn image may find, in the future, a number of different ways in which digital and traditional media may be combined in LUNA. Details of these techniques are found in Chapter 6.

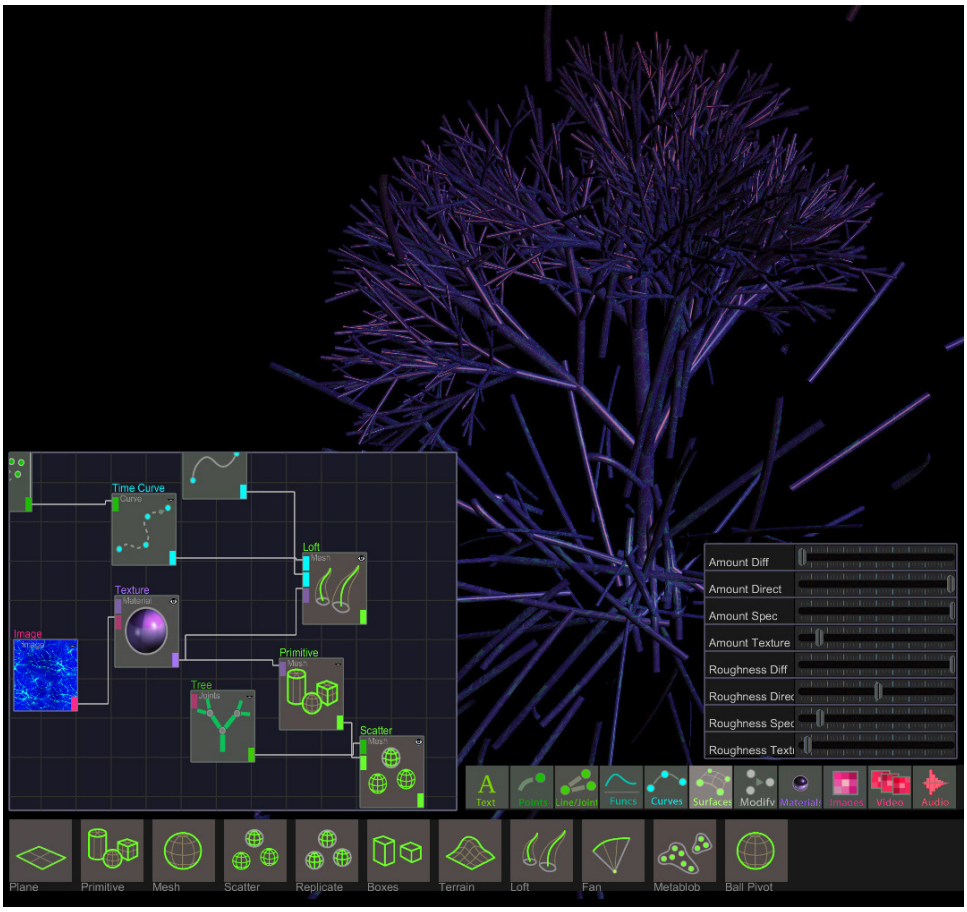


Figure 1.5: Tree modelled in LUNA. Different aesthetics are achieved by modifying the graph, which expresses behavior, geometry, and appearance.

1.5 Chapter Outline

The notion that media artists work best by using a handful of different tools suited to specialized tasks for creative work is challenged by the development of a novel visual data flow language, LUNA, which meets the needs of several creative techniques simultaneously. While LUNA achieves depth only in certain areas, such as procedural modeling, it shows that tools that integrate many different modes of working are

possible while resolving many of the issues that arise from developing cross-disciplinary tools. The dimensions along which LUNA explores these creative boundaries include 1) Programming and Language, 2) Modality and Media, 3) Live Performance and Computation, 4) Motion, Complexity and Autonomy, 5) Structure and Surface, and 6) Image and Idea.

LUNA is presented as an experimental system for digital and media artists to easily and rapidly explore visual possibilities through creative bricolage, to engage in a range of techniques without the need to learn new programming languages.

The remainder of the thesis is organized as follows:

- Chapter 2. Tool Survey. Covers existing tools and comparisons of major features of interest to media artists.

- Chapter 3. Interface Design. Addresses the design of the graphical interface for LUNA, with interface comparisons to other systems.

- Chapter 4. Procedural Modeling. Develops the procedural language for LUNA itself, and establishes the storage and memory structures used, along with performance comparisons to Houdini.

- Chatper 5. Creative Workflows for Media Artists. Establishes six dimensions of creative exploration of interest to media artists, and outlines the evaluation criteria to be used. Discusses the first three dimensions, which relate to the language of LUNA.

- Chapter 6. Structure in Dynamic Media. Discusses the last three dimensions of creativity which deal with the content aspects of LUNA, specifically dynamics, structure, and image.

- Chapter 7. Conclusions. Covers limitations of, and potential future directions for LUNA.

LUNA demonstrates that it is possible to develop a system that combines the conceptual needs of various techniques without sacrificing the primary goals of each. Although specific tools may each have inherent constraints, the division of creative techniques in media arts into separate tools is more likely to be a function of the separation and evolution of goals in different creative communities than a result of any inherent limitations of digital media. LUNA demonstrates this by presenting an alternative tool for media artists designed to explicitly resolve several of these techniques into a single tool, enabling artists to work together across creative boundaries.

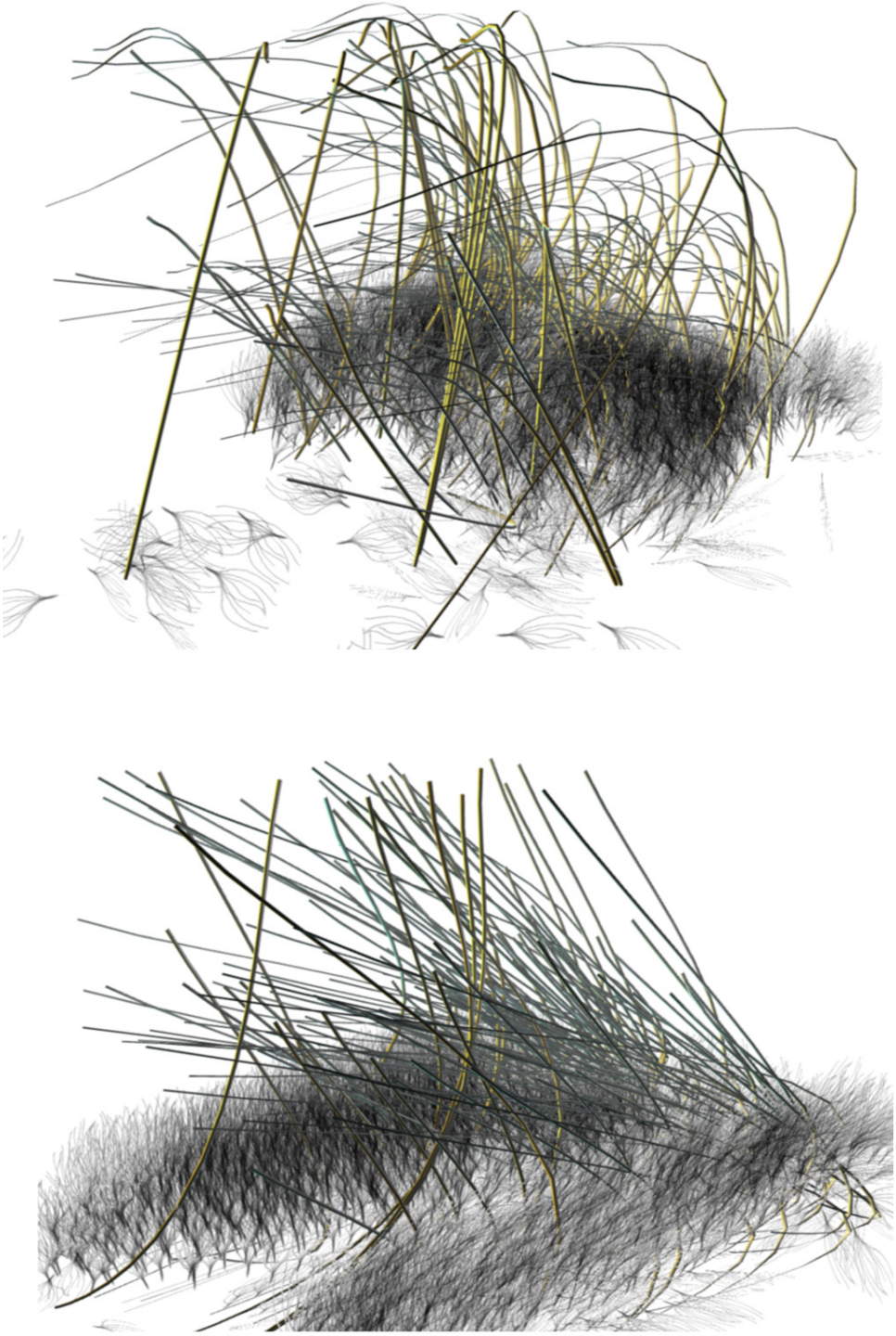


Figure 1.6: Hand-sketched images combined with generative modeling. The technique is described in detail in Chapter 6.

Bibliography

[Candy, 2007] Candy, L. (2007). Constraints and Creativity in the Digital Arts. *Leonardo*, 40(4):366–367.

[Dam, 1998] Dam, A. v. (1998). Some Personal Recollections on Graphics Standards. In *The History of Computer Graphics Standards Development.*

[Dietrich, 1986] Dietrich, F. (1986). Visual Intelligence: The First Decade of Computer Art (1965-1975). *Leonardo*, 19(2):159–169.

[Hoetzlein and Adderton, 2009] Hoetzlein, R. C. and Adderton, D. (2009). MINT/VFX - A High-Performance Computing Framework for Interactive Multimedia.

[Hoetzlein and Castellanos, 2006] Hoetzlein, R. C. and Castellanos, J. (2006). Monarch: Interface for a graphical programming language for digital artists.

[Hoetzlein and Schwartz, 2005] Hoetzlein, R. C. and Schwartz, D. I. (2005). GameX: a platform for incremental instruction in computer graphics and game design. In *SIGGRAPH '05: ACM SIGGRAPH 2005 Educators program*, page 36, New York, NY, USA. ACM.

[Resnick et al., 2005] Resnick, M., Myers, B., Nakakoji, K., Schneiderman, B., Pausch, R., Selker, T., and Eisenberg, M. (2005). Design Principles for Tools to Support Creative Thinking. *NSF Workshop on Creative Support Tools. June 13-14.*

[Shneiderman et al., 2005] Shneiderman, B., Fischer, G., Czerwinski, M., Myers, B., and Resnick, M. (2005). Creative Support Tools. *NSF Workshop on Creative Support Tools. June 13-14.*