Entertaining with Science, Educating with Dance

JENNIFER BURG, Wake Forest University and
KAROLA LÜTTRINGHAUS,[1] Alban Elvěd Dance Company

Two dance performances were produced as a collaboration between a modern dance company and university scientists. The performances were thematically based on concepts from mathematics and computer science and used digital imagery, poetry, and real-time computation directed by a MIDI device. The first production played on the Fibonacci sequence and the Golden Ratio and involved real-time fractal computation in response to the dancers' movements. The second production introduced, at the layperson's level, theoretical concepts of computer science such as computability, language expressiveness, and Turing machines. The two collaborations wove computer science not only into the production but also into the content of the performance.

Categories and Subject Descriptors: J.5 **[Computer Applications]:** Arts and Humanities; K.3 [**Computing Milieux**]: Computers and Education

General Terms: Performance
Additional Key Words and Phrases**:** Digital media, multimedia, parallel computation, real-time computation, Turing machine

1. INTRODUCTION

At first glance, dance and science appear to be two very different worlds, the first inhabited by artists and athletes, the second by those in the physical, mathematical, and natural sciences**.** But, on stage and in classroom, dance and science share more than you might expect. Consider first the relationship between dance and mathematics, a pure science. The rhythms and patterns of dance are mathematical in nature, and thus mathematics can be part of how dance is taught. This works from the opposite perspective, also. Students can learn concepts of geometry and mathematical sequences by acting them out through dance. Science in the form of computer technology also has a close working relationship with dance, to the point where "dance technology" has become a recognized field of study. It is not uncommon for modern dance companies to use computer technology in their works. Computers are used in design and choreography, and modern dance productions can include digital imagery, sound, video, 3-D stereo projections, computer-driven interactivity, and virtual environments.

In this article we explore another, less common, relationship between science and dance – that is, making science a *thematic element* of a dance performance. We describe two productions that resulted from a collaboration between the alban elvěd dance company in Winston-Salem, NC and computer scientists at Wake Forest University. The goals of the dancers and scientists were complementary. The dancers were curious to explore themes of mathematics and technology and find new artistic perspectives by working with scientists. The scientists were challenged by the prospect of performing real-time computation along with live dance and by the opportunity to present their digital media work to the public. Both artists and scientists saw the collaboration as an opportunity to share their passions with new audiences. The hope was that there might be some who would attend the performances for the sake of the math/science theme, and thus be introduced to modern dance; and others would attend for the sake of the dance, and, in turn, would learn a little of the fascinations of mathematics and science.

The productions described in this article were a continuation of alban elvěd's "Free Space" series, an experiment that teamed dancers and scientists to

---

[1] Authors' addresses: J. Burg, Dept. of Computer Science, Wake Forest University, P.O. Box 7311, Winston-Salem, NC 27106; email: burg@wfu.edu. K. Lüttringhaus, alban elvěd dance company, Winston-Salem, NC; Web: http://www.albanelved.com. email: albanelved@triad.rr.com.

see just how well they could talk to each other, and what they might create if they tried. The first production was a collaboration between alban elvĕd and scientists from Duke University, where the technological part of the performance was centered primarily on a variety of cameras and projections and real-time computation. One dance piece used Argus, an array of cameras situated at multiple viewpoints, which created, by means of parallel computation, a 3D reconstruction of a dance. The work called "Fibonacci and Phi" moved the collaboration to Wake Forest University and extended it to a one-and-a-half hour performance involving parallel computation for real-time navigation through fractal digital poetry, digital imagery, stereo projections, and audience interaction via hand-held computers. The second alban elvĕd/Wake Forest production, entitled "Une Journée Abstraite," introduced fundamental concepts of computer science by means of digital poetry, digital image projections, and a "Turing Machine Dance." Parts of "Fibonacci and Phi" and "Une Journée Abstraite," from a collaboration of the present authors, are described below.

2. RELATED WORK

2.1 Teaching Dance with Mathematics, Mathematics with Dance

Science and dance have more to do with each other than you might imagine at first thought. When you realize that mathematics is a science, the relationship between science and dance becomes more obvious. Dance movements and choreography can be described in mathematical terms, e.g., symmetry, correlations, patterns, geometrical shapes, tessalations, and even chaos theory, are sometimes used in teaching dance. Looking from the opposite perspective, dance can be used in teaching mathematics, especially at the elementary or middle-school level [Math Dance 2001; Schaffer and Stern 2001]. When students act out mathematical concepts with steps, movements, and gestures, the concepts become real to them. Weaving dance into mathematics instruction is viewed as a way of integrating the arts into the mainstream curriculum and, at the same time, making mathematics more interesting and extending the attention span of students.

2.2 Science and Technology for Choreography, Preproduction, and Performance

Rudolf Laban, a central European architect with an interest in dance theory, applied scientific analysis to choreography. Laban developed a notational system that provides a detailed record of a dancer's movements on the basis of spatial, anatomical, and dynamic principles [Laban 1966]. His notation, first known as *Kinetography Laban*, was formalized as *Labanotation* by Anne Hutchinson Guest [Guest 1984], and is still used to teach choreography.

Computer technology is well-rooted in the production and performance of modern dance, to the point where "dance technology" is now a recognized term, defined in online encyclopedias and featured in descriptions of dance troupes and academic programs (see, e.g., Ohio State's Dance Technology program and other programs that are part of ADaPT, the Association for Dance and Performance Telematics). Technology can be used at all stages of a dance production: choreography, rehearsal, preproduction, and performance. Modeling systems allow the artist to design dance steps and movements by means of 3D computer models (e.g., Poser and Life Forms/DanceForms [Calvert et al. 1993]). Manipulating the 3D models opens the choreographer to new possibilities for dance phrases, freeing her from personal dance clichés or preference for movements that are natural to her own body. Once the choreography is completed, the dancers can use the computer-generated images in rehearsal. The computer models can also be part of the performance. In some cases, the film or video is itself the artistic end-product or the computer images can become virtual dancers projected on-stage to interact with human dancers.

Choreographer Merce Cunningham received wide acclaim for groundbreaking works such as “Biped” and for his interdisciplinary collaborations and use of computer technology. “Biped” began as a virtual dance installation called “Hand-Drawn Spaces,” a 1997 collaboration with Paul Kaiser and Shelley Eshkar that was translated to a full-length dance. Cunningham was among the first to use modeling software such as Life Forms/Dance Forms and to integrate computer animation with dance. Other innovators have created dances that exist entirely within the computer rather than being performed on-stage. For example, “Hyperalarm” by Michael Cole [St. Petersburg 2003] is a “4-minute computer animated modern dance music video” that imagines a dream sequence inside a digital alarm clock. “Ghostcatching” is a noteworthy virtual dance installation that merges dance, drawing, and computer technology, created by choreographer Bill T. Jones and digital artists Shelly Eshkar and Paul Kaiser [Ghostcatching].

“Biped,” “Hyperalarm,” and “Ghostcatching” all use motion-capture, one of the highlights of dance technology since the 1990s. Motion-capture works by means of reflectors attached to a dancer’s body. A camera senses the dancer’s movements and sends them to a computer where a motion-capture program records them and then recreates them in the form of computer-animated figures. These figures can then become characters in a virtual dance or appear alongside human dancers in a live performance. Early innovators in motion-capture also include The Troika Ranch “digital dance theatre company.” (Meader et al. [2004] give a good summary of the work of Cunningham and Troika Ranch on motion-capture.)

2.3 Machine Choreography, Interactivity, and Nondeterminism In Dance

Going a step beyond using computer programs for computer models and virtual dancers, the artist can allow a computer program to actually design the choreography. Programs for computer-generated choreography do exist and have been used, but they are probably more interesting to computer scientists than to choreographers, who are naturally reluctant to abandon artistic control to a machine. Handing choreography over to a machine or to a random selection of movements from a given repertoire may lead to unexpected results, but has questionable artistic and emotional value. In his essay, “Frequently Pondered Questions,” Paul Kaiser observes that Merce Cunningham’s predilection for “chance” combinations of elements in “Hand-drawn Spaces” works because only 71 phrases were in the repertoire of choices, all of them created such that logical transitions existed between any two phrases. A repertoire of 71,000 phrases would not lend itself as well to chance combinations. Kaiser posits that a better system for computer-generated choreography would involve “weighted probability,” that is:

> Complex contingency, which comes only by building networks of IF>THEN relationships. For chance to be powerful, its effects must ripple down through many possible branches. Which means that it’s not pure chance, but rather weighted probability that deserves our attention. In reality, isn’t it rarely the case that multiple outcomes are equally likely? [Kaiser 2002]

Rather than relinquishing the choreography entirely to a computer program, a more intriguing possibility for artists is to allow some amount of non-determinism in the emergence of a total dance production. This is a feature that Merce Cunningham experimented with in various forms. His method, confirmed by his collaborators, is to reveal just a “phrase or two” about his artistic intentions, and then free the musicians, visual decor designers, etc., to create their own components for the production – the parts not put together until the last moment [Kaiser 2000]. In the same “limited nondeterminism” vein, Trisha Brown choreographed a performance called “how long does the subject linger on the edge of the volume" with human dancers; motion sensors; infra-red cameras; and geometric computer animation, lights, and music that respond in real-time to the dancer's movements based on a computer algorithm. The non-deterministic result is that images – mostly lines and abstract shapes – are not the same in every performance of the dance. This work was done in collaboration with Marc Downie, Shelley Eshkar, Paul Kaiser, and Arizona State computer engineers as part of their Motion-e project [how long… 2005].

Real-time interactivity allows for a certain amount of nondeterminism in dance if it is based on the dancers' movements or audience reactions that may not be exactly the same from one performance to the next. Motion sensors are one way to achieve this interactivity; another way is by means of MIDI devices. The Troika Ranch dance company has invested creativity in this technology. Their MidiDancer device is a wireless movement-sensing system that measures the flexion of a dancer's joints and communicates the information to a computer via MIDI messages. There is also a graphic authoring tool that orchestrates the real-time performances of dancers and musicians via MIDI communication.

The alban elvěd dance company has been working with this technology since 2001, when they created the work "MiDi" using a movement-to-MIDI converter designed by Robert Andrew Turner. The device allowed the dancers to control the sounds for the live, on-stage performance by moving through a light sensor – laser switch grid. Individual sound components were composed by Turner and programmed into a sampler such that they could be combined in many variations to produce melodic and rhythmic entities. (See the description of alban elvěd's MIDI device in Section 3.2.)

Palindrome is another dance company at the forefront of real-time interactivity, which is based on two types of devices: electrodes and "frame-grabbing" software. Electrodes are small electrically-conductive pads that are attached to the dancer's skin, allowing electrical signals from within the body to be detected. The signals were EMG for skeletal muscles and ECG for the heart muscle. In an early work called "Heartbeat Duet," electrodes were used to sense the heartbeats of two dancers, from which a musical score was composed by turning the heartbeats into separate musical notes in counterpoint rhythm. Palindrome's "frame-grabbing" software, called EyeCon, captures video images from a live performance and sends them to a computer [Weschler et al. 2004]. The dancer's movements shown on the computer screen interact with lines and fields superimposed by the choreographer, trigger computer events like sound, music, lighting, and animation.

In addition to "nondeterminism," another computer science term, called "emergent behavior," can be applied to modern dance. In computer science, emergent behavior is a sequence of events that is not consciously "programmed in" but arises from a computer program or simulation. That is, the sequence of events that emerges is not explicitly spelled out anywhere in the program. Instead, the behavior is implied by rules that are applied to the circumstances in which the program is run. It can't be predicted exactly what a behavior will be for every situation by just reading the program superficially, since the rules can be complex and interact with each other. In this vein, the e-Merge project is an art/science research collaboration that relates the concept of emergent behavior to "Nature's self-organizing processes." Viewing emergent behavior as a close relative of chaos and complexity, the goal is to create computer systems that aid the design of dance performances, mimicking natural self-organizing systems like plant growth or bird migrations [e-Merge].

2.4 Thematic Integration of Science and Dance

Use of technology in dance preproduction and production is now fairly common. Thematic use of science and mathematics is less common, but a few examples can be cited. Two groups, already mentioned, attempt scientific themes. E-Merge's goal is to represent processes that could be described as natural, organic, mathematical, or physical – all in the realm of scientific analysis. Palindrome's "Heartbeat Duet," in its use of electrodes to monitor the human heart, makes art of science – an interest of Palindrome's choreographer Robert Wechsler. We were surprised to find a quotation from Wechsler expressing goals remarkably similar to our own:

> Palindrome has an unusual focus: to make dance works, interactive performance pieces, installations, and workshops which correlate concepts and phenomena from science and technology with art. Sometimes, as in the case of the dance DNA…(1981), the connection is literal. The dance is a scale model of the DNA molecule. Other pieces, such as Möbius Band (1995) and TRIO (1989) combine symmetries in time and space and function like puzzles which the audience solve as they watch. It is their artistic concept to bring science and technology into the sphere of art [Weschler 1997, footnote 5].

Another example that integrates a computer science theme into artistic production is worth mentioning because it is the closest parallel we found to our Turing machine dance. In a play called "The Difference Engine," New York playwright Samantha Hunt uses Charles Babbage's invention of one of the first mechanical computers as the premise for reflections on mathematics, numbers, and structure in the human perception of reality. A 2003 production by The Theatre of the Two-Headed Calf included live video feed with an abstract representation of Babbage's difference engine as well as a "number dance" composed of "expressive gestures representing each operative word in Babbage's text" [The Difference Engine 2003].

Fishwick's work in aesthetic computing is similar in spirit to our Turing machine dance [Fishwick et al. 2005]. Aesthetic computing conceives of computation, algorithms, and coding in terms of visual models with an artistry that gives alternative ways to understand computation. This is a reversal of the usual associations between computers and art. Rather than use computers to create art, the idea is to recognize and make use of the aesthetics of computation itself. Although we were unaware of Fishwick's work when we conceived of the "Turing Machine Dance," it might be considered an example of aesthetic computing.

3. FIBONACCI AND PHI

3.1 Art and Theme

The first alban elvěd/Wake Forest collaboration was entitled "Fibonacci and Phi" performed December 4 to7, 2003 at Wake Forest University. This production explored two mathematical concepts that have long been linked with beauty in nature and art: the Fibonacci sequence and the Golden Ratio, Phi.

The dance piece was set in the span of one day and had five dancers. The stage set was minimal, with some tables and chairs and an old radio as props. As the stage was rather barren, we were able to focus on large projections such as the fractal. We chose to add aerial dance via ropes and bungee cords. An original musical score, composed by Mark Wienand and Sam Taylor with Jeff Schmitt and John Pratt, accompanied the dance. Wienand also played woodwinds live and danced throughout the entire piece. In keeping with the theme, the Fibonacci sequence and the Phi ratio were structural metaphors that guided the creation of the music. Intervallic patterning, sounds derived from nature, and rhythmic groupings on large and small scales all related back to these proportions in both obvious and subtle ways. The intervallic distances measured from middle C used Fibonacci proportions. Interestingly, the frequency of middle C itself is 100 * Phi. Lighting, designed by Jonathan Christman and Karola Lüttringhaus, also incorporated thematic elements. For example, a spiral pattern of light illuminated a piece called the “Fibonacci Spiral Dance.”

The performance was tied together by a poem expressing the mystery of mathematical beauty as it is manifested in the world around us. The poem, given below, was divided into parts and read by Los Angeles artist Rhan Small as a narrative backdrop to scenes spanning one day.

The poem “Phi”
written by Jennifer
Burg,
spoken by Rhan Small
audio link

The biggest technical challenge in “Fibonacci and Phi” was staging a “fractal duet” in which the dancers dance in front of a Mandelbrot fractal, giving the audience the illusion of moving through and into its infinitely self-replicating detail. Although we knew we could easily create this illusion with a canned video of the fractal, we wanted to do so in real-time and give the audience some sense of the beauty of the mathematics, the complexity of the computation, and of the technology that makes real-time computation possible. We’ll first explain our technical implementation.

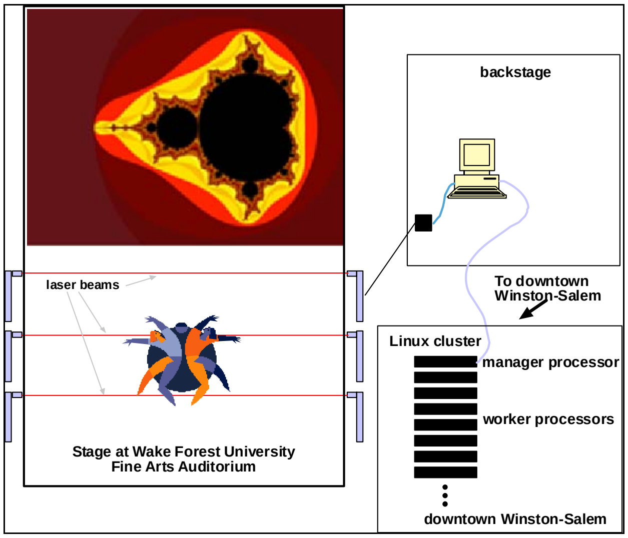


Fig. 1. Stage setup for fractal duet in “Fibonacci and Phi.”

The scene was designed so that the dancers performed in front of a Mandelbrot fractal that was projected onto a 40 × 25 foot-screen at the back of the stage. This piece made use of a movement-to-MIDI system designed by Robert Andrew Turner. The system consists of light sources and light-sensitive receivers matched up in pairs, the sources on one side of the stage and the receivers on the other. The receivers are wired to emit 1 or 0, depending on the light; the 0’s and 1’s are forwarded to a MIDI converter; the signals from individual lights can be combined in various ways, as programmed within the MIDI converter, and the messages can be interpreted as triggers for sound, light, or whatever the choreographer imagines for the dance.

For the “Turing Machine Dance,” we used six light/receiver pairs beamed across the stage. When the dancers moved through the light beams, a signal was sent to the MIDI converter and from there to a computer backstage, which we call the “client computer.” The client computer was also connected to a high-lumens projector (for the fractal display) and to a Linux cluster of parallel computers six miles away, in downtown Winston-Salem. Depending on which beam was broken, the signal sent from the client computer to the cluster indicated whether the fractal was to be recomputed *in*, *out*, *up and in*, *down and in*, *left and in*, or *right and in*, where *in* and *out* mean *on a smaller scale* and *on a larger scale*, respectively. Moving *in* gave the audience the illusion of moving deeper into the fractal; this setup is shown in Figure 1 and the dance in Figure 2.

Early experiments made it clear that sequential computation couldn't keep up with the messages sent by the dancers. For smooth animation, each step had to be small – only a few pixels closer in each dimension. For fast movement, since each step was so small, the fractal had to be recomputed and redisplayed several times per second.

The fractal computation program was based on the iterative equation $f(z) = z^2 + c$, where c and z are complex numbers. To compute a pixel's color, c initially represents the pixel's position; that is, the pixel's horizontal coordinate is mapped proportionately to a real-number between -1.5 and 1.5, and this becomes the real-number component of c. Similarly, the vertical coordinate is mapped to a real-number between -1.5 and 1.5; and this becomes the coefficient of the imaginary component of c; z is initially 0. $f(z) = z^2 + c$ is computed repeatedly for a maximum number of iterations or until the value converges. The result is a Mandelbrot fractal, like the one shown in Figure 1. This computation is obviously time-consuming. The worst-case complexity for computing one frame is $x * y * t$, where x is the horizontal resolution and y is the vertical resolution. For our application, this is

$$1024 * 768 * 1000 = 786{,}432{,}000 .$$

Each pixel computation requires four multiplications and three additions. Not all pixels require the maximum number of iterations – only the black pixels do. The frames with the most black are the most expensive to compute.

Fig. 2. Fractal duet in "Fibonacci and Phi."
(Courtesy of Ching-Wan Yip, Wake Forest University)

Our parallel version of the fractal computation is an MPI program running on a Linux cluster. The standard approach in MPI implementations of fractal computation works as follows: the master process divides the rows of a fractal frame among the worker processes; each worker computes the colors of the pixels in its rows; when a worker is done with its computation, it immediately sends XWindows calls to display the rows. This causes the rows to come streaming in separately, rather than having a complete frame displayed in one instant. For smooth animation, we needed to display fully-constructed frames, and thus we needed to funnel all the new pixel values for a frame through the master process. This created a bottleneck, necessitating fast interprocess communication and fast communication between the master process and the client computer. Hence, we established a gigabit ethernet link from the theatre on the Wake Forest campus to the Linux cluster six miles away in downtown Winston-Salem. Within the cluster, we used myrinet rather than ethernet connections, the highest-speed interprocess communication we had available. A final optimization was to use run-length encoding on the pixel data sent from the workers to the master process.

With the implementation described above, we were able to compute seven fractal frames per second. This was fast enough for smooth animation in response to the movements of the dancers. The final gesture of the dancers signaled a fast zoom into a black hole, which we were able to animate smoothly in real time.

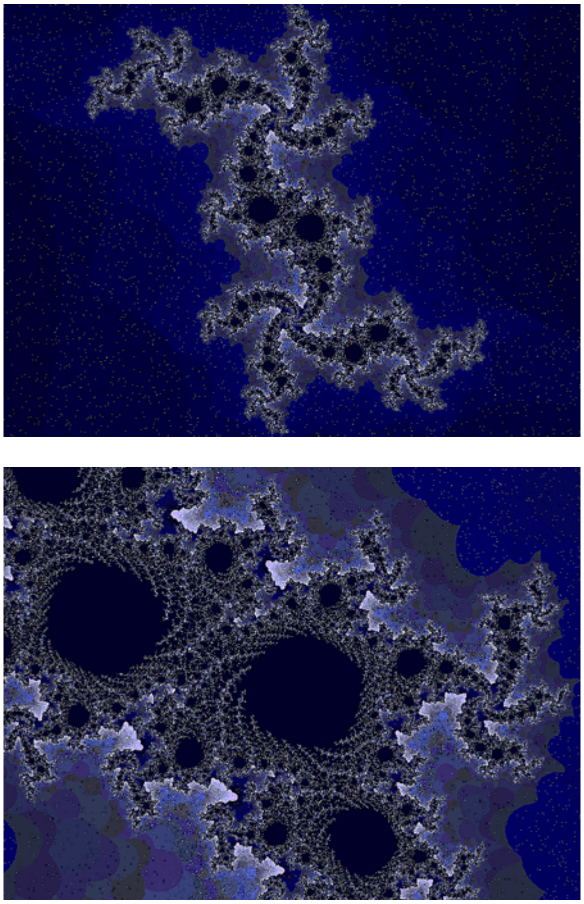


Fig. 3. Two views of "Night sky," a Julia fractal.

In the final scene of "Fibonacci and Phi," we implemented a different version of the fractal computation. The scene called for a "night sky," which we designed as a Julia fractal, suggested the shape of a swirling galaxy. We wanted to create a very slow descent into one of the black holes of the fractal. For this scene, we chose to create a video rather than use real-time fractal computation.

A Julia fractal is computed like a Mandelbrot fractal, with the following difference: For each pixel computation, the pixel's horizontal and vertical coordinates are translated to real numbers to become the initial value of the real-number component and coefficient of the imaginary number component, respectively, of z,. c is constant for a fractal frame and chosen experimentally such that it creates “interesting” looking fractals. The fractal we designed is shown in Figure 3. Random points of light were added to the fractal to simulate stars, which “twinkled” as frames changed because their random positions changed. This fractal, animated in a video, was used as a backdrop to the final aerial dance. To create the video, the Julia fractal program wrote 2000 fractal frames to a file, each a little closer than the previous one. The frames were compiled and compressed into a Quicktime movie.

After the show’s run, the scientists did a formal analysis of the fractal computation to determine which speed-up factors had the most impact. For these experiments, the same 100 consecutive frames were computed each time. Zoom-in was repeated for 100 frames, each time as if a pixel area had been selected that was two pixels smaller in each dimension (moving in one pixel on left, right, top, and bottom). (Every fourth frame, the y direction was not changed in order to maintain the 4:3 ratio of the frame.) A summary of the analysis is given below; additional details are given in Burg and Miller, [2004].

From Table I, it can be seen that gigabit ethernet from client to cluster, myrinet on the cluster, and 16 processors allowed us, on the average, to compute one fractal frame about every 0.14 seconds, which is about 7 frames per second.

Table I. Average, Minimum, and Maximum Time to Compute a Frame for All 100 Frames

	4 *Processors*		8 *Processors*		16 *Processors*	
-- 100 Mb/sec from client to cluster -- ethernet on the cluster -- slower graphics card	avg	0.558267	avg	0.477561	avg	0.467278
	min	0.433044	min	0.383174	min	0.360396
	max	0.959640	max	0.639695	max	0.630410
-- 100 Mb/sec from client to cluster -- myrinet on the cluster -- faster graphics card	avg	0.551303	avg	0.466858	avg	0.468283
	min	0.431126	min	0.361114	min	0.359247
	max	0.779139	max	0.600989	max	0.599170
-- 1 gigabit/sec from client to cluster -- ethernet on the cluster -- faster graphics card	avg	0.303111	avg	0.178108	avg	0.128352
	min	0.102596	min	0.096622	min	0.098177
	max	0.580423	max	0.439282	max	0.325958
-- 1 gigabit/sec from client to cluster -- myrinet on the cluster -- faster graphics card	avg	0.303086	avg	0.171172	avg	0.137445
	min	0.100156	min	0.102299	min	0.106616
	max	0.624767	max	0.345077	max	0.325387

A factor not accounted for in Table II is run-length encoding (RLE), which we used to reduce the amount of data sent from the worker processes to the master on the cluster. The workers send the color of each pixel to the master, for a resolution of 1024 × 768 pixels per frame. Without run-length encoding, the amount of data sent from worker to master process would be a constant $1024*768*b = 786{,}432*b$ bytes per frame, where b is the color bit-depth. RLE consists of sending a two-byte integer d and a two-byte color code to indicate d consecutive bytes of the same color, as opposed to sending d two-byte color codes. Table III shows the average, minimum, and maximum number of bytes sent per frame over all 100 frames when RLE is used. Given the nature of the fractal images, which have long runs of the same color pixels, RLE reduces the data communication significantly – 9435 bytes sent on average, as opposed to 1024*768=786,432 bytes for the "average" frame.

Table II. Average, Minimum, and Maximum Bytes per Frame Communicated per Slave to the Master Over All 100 Frames

Average	*Minimum*	*Maximum*
9435	3188	20414

Table III shows the extent to which the transmission of pixel data saturated the network between the client computer and the cluster. For 16 processors, an average data rate of 92 Mbit/sec using a 100 Mbit network connection as opposed to 235 Mbit/sec using gigabit ethernet indicates that the gigabit ethernet speed between the client and the cluster makes a significant difference in the refresh rate of the animated fractal.

Table III. Network Usage from Client to Linux Cluster

100 *Mbit/second Network Connection from Client to Cluster*				1000 *Mbit/second (GigE) Network Connection from Client to Cluster*			
Ethernet connectivity in cluster		Myrinet connectivity in cluster		Ethernet connectivity in cluster		Myrinet connectivity in cluster	
8 procs	92 Mbit/sec	8 procs	94 Mbit/sec	8 procs	235 Mbit/sec	8 procs	236 Mbit/sec
16 procs	92 Mbit/sec	16 procs	94 Mbit/sec	16 procs	235 Mbit/sec	16 procs	304 Mbit/sec

The benefit of myrinet connectivity within the cluster is also visible in Table III, particularly in the difference between ethernet and myrinet for 16 processors. The data rate between the client and the cluster is 235 Mbit/sec when the cluster has an internal ethernet connection, whereas it is 304 Mbit/sec when the cluster has an internal myrinet connection. We found through further experiments that myrinet distributes the work load better among the processors, and that this helps to maximize network usage between the cluster and the client.

One conclusion we drew from the analysis is that it was possible to take "Fibonacci and Phi" on the road. With an 8-processor cluster, using ethernet in the cluster (which we could manage in portable form), we would be able to display, on average, over 5 fractal frames per second. This animation is smooth enough for the choreography as designed. However, if we wanted to try more challenging choreography or more complex fractal changes from frame to frame, we would need even greater speed than achieved with 16 processors and myrinet. It would be interesting to see if the "real-time fractal zooming" algorithm made public by Hubicka, March, and Kovacs could be adapted for our use [XAOS 2005]. This algorithm speeds up fractal calculation by not recomputing pixels that do not change from one frame to the next; we have not seen any parallel versions of the algorithm.

4. "UNE JOURNÉE ABSTRAITE"

The challenges in our second collaboration, "Une Journée Abstraite," were more conceptual than technical. We wanted to weave abstract concepts of computer science – in particular, machine computation and language expressiveness – into the theme of the performance. To do this, we made the computer a character on-stage, framing the performance with the computer's thoughts. The computer was projected as a digital image above the dancers; the dancers had a laptop computer on stage.

The set was built around a 15-foot-tall steel structure on which the four dancers walked, climbed, sat, and hung (Fig. 4); they also spoke in English, French, Italian, German, and Hebrew.

Fig. 4. Structure from “Une Journée Abstraite.”

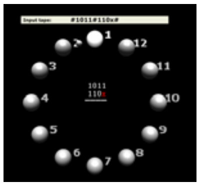
In the discussion below, we focus on only those scenes on which the authors collaborated.

The scene opens on a dark stage with a projection of the computer, which speaks to the audience in its electronic voice. As it speaks, the computer’s words appear on the computer screen. The computer’s opening monologue introduces concepts of computability, language expressiveness, and with irony, Gödel’s incompleteness theorem. At the end of the monologue, the computer reflects images of itself like a mirror within a mirror. Fig. 5 links to the projection in the opening scene.

Click image see projection.

Fig. 5. Opening scene of “Une Journée Abstraite.”

The “Turing Machine Dance” was the highlight of the dance/science collaboration. It represents the fundamentals of abstract machine computation with a combination of choreography and digital projections. The dancers enact a Turing machine program that computes the sum of two binary numbers. At the beginning of the dance, the rules of the program flash across the screen. During the dance, the digital projection shows machine states in the form of 3-dimensional spheres and a token moving from one state to the next based on an input tape, also displayed, as shown in Fig. 6. The program has twelve states represented by twelve chairs that dangle from ropes as the dance begins and then descend to the stage. The dancers move from chair to chair with movements corresponding to inputs, outputs, and state changes.



Click image to see projection.

Fig. 6. Turing machine projection from the “Turing Machine Dance.”

The last scene of “Une Journée Abstraite” brought the dance full circle, returning to the themes of expressibility and computability. A projection showed a move-the-tiles puzzle that unscrambled itself to reveal, line-by-line, the verses of the poem shown in **[author: shown in what?]** The human element is juxtaposed with a poem spoken by a human silhouette. The poem harkens back to one spoken earlier in the piece by one of the dancers.

Click image to see projection.

Fig. 7. Second and third poems from the conclusion of "Une Journée Abstraite."

The computer returns to have the final say, but its last words are in deference to human expressiveness.

Fig. 8. Computer clicks shut at conclusion of "Une Journée Abstraite."

5. CONCLUSIONS AND FUTURE WORK

What conclusions did we bring away from our two collaborative experiences, and what might we change if we try this again?

A question we grappled with was how explicit should we be in "telling" or "teaching" the audience the concepts we were presenting.

Without our telling them, the audience would never have known the difference between the real-time computation of the "Fractal Duet" and the canned video of the "Night Sky" in "Fibonacci and Phi." The obvious question then becomes "Why go to the trouble of real-time computation?" The answer is simply that this is the dance that the artists and scientists wanted to do together, both figuratively and literally. The scientists loved the power and beauty of computation; the artists were intrigued by being able to control it. Together, they wanted to share this experience with the audience through dance.

So how then do we make the audience aware of science behind the scenes? Do we simply tell them? Do we weave an explanation into the performance in subtle ways? Wechsler describes the problem this way:

> To simply explain the set-up beforehand is risky. The danger is that the performance, what is ostensibly a piece of art, becomes a lecture and a demonstration. One needs to find non-pedagogic ways to help the audience along. One solution is simply to have a piece build-up slowly, step-by-step, starting with the simplest kinds of interactions first. In this way the piece can "explain itself" as it goes along. Another possibility is to affect explanations using other media –

> projections, sound tracks, program notes, audience involvement (for example, posing questions to the public). And personally, I have no particular objection to an occasional verbal explanation, though this may come during or after a piece rather than before, giving the audience at least the chance to respond "innocently" to a work.

We used a combination of the methods to introduce the audience to fractals, Fibonacci sequences, Phi, and parallel computation: program notes, poetry integrated into the performance, a post-performance chat with the audience, and a Saturday afternoon panel discussion. In "Une Journée Abstraite," the computer became a character that spoke to the audience in person about the nature of its computation and the limits of its expressiveness.

In the end, the conclusions of the artists and scientists were not exactly the same, but one thing both artists and scientists agreed on is that they were now reluctant to separate so clearly as artist on the one side and scientist on the other. By encoding the computation as a dance, the choreographer grew to understand precisely how the Turing machine managed to add two numbers using such a rudimentary model of computation. The scientist, for her part, wrote poetry, learned to listen and see more intently, and became even more fascinated with how mathematics allows us to weave our visions into sound, light, and motion. Both collaborators continue to share an interest in deciphering nature, science, the way things work, and what this means to how we live.

We have already begun brainstorming for our next production, to be entitled "Waves."

Bibliography

ADAPT. The Association for Dance and Performance Telematics. http://dance.asu.edu/adapt/.

ALBAN ELVĚD DANCE CO. http://www.albanelved.com/.

ATLAS, C. 2001. Merce Cunningham: A Lifetime of Dance. DVD. Winstar Home Entertainment.

BURG, J. AND MILLER, T. 2004. Cluster computation in step with real-time dance: An experiment in art/science collaboration. In *Proceedings of the 17th International Conference on Parallel and Distributed Computing Systems* (Sept. 15-17, 2004).

CALVERT, T. ET AL. Desktop animation of multiple humanf igures. *IEEE Computer Graphics and Applications 13*, 3 (May 1993), 18-26.

THE DIFFERENCE ENGINE. 2003. http://www.twoheadedcalf.org/www.difference/Difference_Engine.html/

DOWLING, P., WECHSLER, R., AND WEISS, F. 2004. EyeCon: A motion sensing tool for creating interactive dance, music and video projections. In *Proceedings of the Society for the Study of Artificial Intelligence and the Simulation of Behaviour* (University of Leeds, England, March 2004).

FISHWICK, P. A., DIEHL, S., PROPHET, J., AND LOWGREN, J. **2005.** Perspectives in aesthetic computing, *Leonardo 38*, 2, 133-141.

GHOSTCATCHING. http://www.openendedgroup.com/artworks/ghostcatching/ghostcatching.htm.

GUEST, A. H. 1984. *Dance Notation: the Process of Recording Movement on Paper*. Dance Horizons, New York, 1984.

How Long Does the Subject Linger at the Edge of the Volume….2005. http://openended.kaiserworks.com/artworks/howlong/howlong.htm.

E-MERGE. http://www.unsafesound.com/e-merge1.html.

KAISER, P. 2000. Essay: On biped. http://openended.kaiserworks.com/artworks/pdf/Biped.pdf.

KAISER, P. 2002. Frequently pondered questions. In *Envisioning Dance on Film and Video*. Judy Mitoma (ed.). Routledge Press, 2002.

LABAN, R. 1966. *Choreutics*. L. Ullmann (ed.). MacDonald and Evans, London, 1966.

Math Dance with Dr. Shaffer and Mr. Stern. 2001. http://www.mathdance.org/.

MEADOR, W. S. ET AL. 2004. Mixing dance realities: Collaborative development of live-motion capture in a performing arts environment. *ACM Computers in Entertainment 2*, 2 (April 2004).

THE MERCE CUNNINGHAM DANCE CO. http://www.merce.org/.

New Life Form Evolving. 1991. *Update Newsletter 3*, 3 (April 1991). http://www.css.sfu.ca/update/vol3/3.3-Calvert.html. The Centre For Systems Science, Simon Fraser University, Burnaby, B.C., Canada.

PALINDROME DANCE CO. http://www.palindrome.de/.

SCHAFFER, K. AND STERN, E. 2001. *Mathematics and Dance Bibliography*. Linked from http://www.mathdance.org.

ST. PETERSBURG INTERNATIONAL DANCE FILM FESTIVAL. 2003. Program 3. http://www.kinodance.com/russia/films_program3_2003.html.

TROIKA RANCH DANCE CO. http://www.troikaranch.org.

WECHSLER, R. 1997. Computers and art: A dancer's perspective. *IEEE Technology and Society Magazine 16* (Fall 1997), 7-14.

WECHSLER, R., FRIEDER, W., AND DOWLING, P. 2004. EyeCon – A motion sensing tool for creating interactive dance, music, and projections. In *Proceedings of the Society for the Study of Artificial Intelligence and the Simulation of Behavior and Cognition* (University of Leeds, England, March 2004).

XAOS: A Fast Interactive Real-Time Fractal Zoomer and Morpher. 2005. http://xaos.sourceforge.net/english.php.

Received; revised; accepted **November 2005**