Entertaining With Science, Educating With Dance

JENNIFER BURG and KAROLA LÜTTRINGHAUS[1]

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.

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 rationalists and intellectuals. But dance and science share a stage or classroom more than you might expect based on their apparent differences. 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 now 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 paper, we explore another relationship between science and dance that is less common – that is, making science a *thematic element* of a dance performance. We describe two productions that resulted from a collaboration between the alban elvĕd dance company in Winston-Salem, NC, and computer scientists from 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 motivated by the opportunity to present their digital media work to the public. Both the artists and the scientists saw the collaboration as an opportunity to share their passions with new audiences. The hope was that, on the one hand, there might be some who would attend the performances for the sake of the math/science theme, and thus they would be introduced to modern dance; and there might be others who would attend for the sake of the dance, and they in turn would learn a little of the fascinations of mathematics and science.

The productions described in this paper were a continuation of alban elvĕd's "Free Space" series, an experiment that teamed dancers and scientists to see just how well they could talk to each other, and what they might create if they tried. The first production in the series was a collaboration between alban elvĕd and scientists from Duke University, where the technological part of the performance centered primarily on a variety of cameras and projections and

[1]Authors' addresses: Jennifer Burg is a faculty member in the Department of Computer Science, Wake Forest University, P.O. Box 7311, Winston-Salem, NC 27106. Email: burg@wfu.edu.
Karola Lüttringhaus is choreographer and artistic director of the alban elvĕd dance company, Winston-Salem, NC. Web: http://www.albanelved.com. Email: albanelved@triad.rr.com.

real-time computation. One dance piece used "Argus," an array of cameras situated at multiple viewpoints and creating, by means of parallel computation, a 3-D reconstruction of the dance. "Fibonacci and Phi" moved the collaboration to Wake Forest University and extended it to encompass a 1 and ½ hour performance involving parallel computation for real-time navigation through a fractal, digital poetry, digital imagery, stereo projections, and audience interaction through handheld 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." The parts of "Fibonacci and Phi" and "Une Journée Abstraite" that resulted from a collaboration of this papers' 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, science's relationship with dance becomes more obvious. Dance movements and choreography can be described in mathematical terms – symmetry, correlations, patterns, geometrical shapes, tessalations, and even chaos theory – and mathematics is sometimes used in the teaching of dance. Looking from the opposite perspective, we find that dance can be used in the teaching of 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, Pre-Production, and Performance

Scientific analysis has been applied to choreography by Rudolf Laban, a central European architect whose spatial thinking drew his interest toward dance theory. Laban developed a notational system that records in detail a dancer's movements based on spatial, anatomical, and dynamic principles [Laban 1966]. This notation, first known as *Kinetography Laban*, was formalized as *Labanotation* by Anne Hutchinson Guest [Guest 1984] and is still in use in teaching and choreography.

Computer technology is well-rooted in modern dance production and performance, to the point where "dance technology" is now a recognized term defined in on-line encyclopedias and featured in descriptions of dance troupes and academic programs. (See, for example, Ohio State's Dance Technology program or 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, pre-production, and performance. For choreography, modeling systems allow the artist to design dance steps and movements by means of 3-D computer models (e.g, Poser and Life Forms/DanceForms [Calvert et al., 1993]). Manipulating the 3-D models opens the choreographer to new possibilities for dance phrases, freeing her from her own 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.

With groundbreaking works such as “Biped,” choreographer Merce Cunningham has received wide acclaim 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 integrate computer animations 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 set inside a digital alarm clock. Also of note is “Ghostcatching,” a 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, which has been 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 dancers’ movements and sends them to a computer, where a motion capture program records them, to be recreated in the form of computer-animated figures. These figures can then become characters in a virtual dance or appear along side human dancers in a live performance. Early innovators in motion capture for dance also include The Troika Ranch “digital dance theatre company.” ([Meader et al., 2004] gives a good summary of Cunningham’s and Troika Ranch’s work with motion capture.)

2.3. Machine Choreography, Interactivity, and Non-Determinism in Dance

Going a step beyond using computer programs for computer models and virtual dancers, the artist can allow a computer program actually to 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 completely 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. Paul Kaiser comments on this in his essay “Frequently Pondered Questions.” He first 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]

A more intriguing possibility for artists than relinquishing the choreography entirely to a computer program is to allow some amount of non-determinism in the emergence of a total dance production. This is a feature that Merce Cunningham has experimented with in various forms. Cunningham’s method of working, as testified to by his collaborators, is to reveal just a “phrase or two” about his artistic intentions, and then to set free the musicians, visual decor designers, etc. to create their own components for the production – the various parts not being put together until the last moment [Kaiser, 2000]. In this same vein of "limited non-determinism," Trisha Brown has choreographed a performance called "how long does the subject linger on the edge of the volume" involving human dancers; motion sensors; infra-red cameras; and geometric computer animations, 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 non-determinism in dance if it is based on dancers' movements or audience reaction 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. They also have a graphic authoring tool that orchestrates real-time performances using MIDI communication of dancers and musicians.

The alban elvĕd dance company has been working with this technology since 2001, when they created a work "MiDi" using a movement-to-MIDI converter designed by Robert Andrew Turner. The device allowed the dancers to control the sounds for the performance live on stage 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 elvĕd's MIDI device in Section 3.2.)

Another dance company at the forefront of real-time interactivity is Palindrome. Their interactivity 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 could be EMG for skeletal muscles or ECG for the heart muscle. An early work called "Heartbeat Duet" used electrodes to sense two dancers' heartbeats and composed a musical score by turning the heartbeats into separate musical notes in counter-point rhythm. Palindrome's "frame grabbing" software, called EyeCon, captures video images from a live performance and sends them to a computer [Weschler, Weiß, and Dowling, 2004]. The dancer's movements shown on the computer screen interact with lines and fields superimposed by the choreographer, triggering computer events such as sounds, music, lighting, or animations.

In addition to "non-determinism," another computer science term that can be applied to modern dance is "emergent behavior." In computer science, emergent behavior is a sequence of events that arises from a computer program or simulation that was not consciously "programmed in." That is, the sequence of events that emerges is not explicitly spelled out anywhere in the program. The behavior is instead implied by rules that are applied to the circumstances in which the program is run. Different behavior emerges from different situations, and you can't predict exactly what that behavior will be for every situation just by superficially reading the program, because 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 what they call "Nature's self-organizing processes." Viewing emergent behavior as a close relative of chaos and complexity, their goal is to create computer systems that aid in 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 pre-production and production has now become fairly common. Thematic use of science and mathematics is less common, but a few examples can be cited. Two groups already mentioned approach 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 – something that apparently has long been an interest of Palindrome 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 solves as they watch. It is their artistic concept to bring science and technology into the sphere of art [Footnote 5 in Weschler, 1997].

Another example that integrates a computer science theme into an 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 mathemtics, 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 inroads to our understanding. This is a reversal of usual associations between computers and art. Rather than to use computers to create art, the idea is to recognize and make use of the aesthetics of computation itself. Though we were unaware of Fishwick's work when we conceived of the Turing Machine Dance, the dance 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" and was performed December 4-7, 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 was set minimally, with only tables and some chairs and props such as an old radio. With the stage 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 in 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 occur using 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 Jennifer Burg,
spoken by Rhan Small
audio link

The biggest technical challenge we took on in “Fibonacci and Phi” was staging a “fractal duet” where the dancers would dance in front of a Mandelbrot fractal, giving the audience the illusion of moving through it 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 this in real-time and give the audience some sense of the beauty of the mathematics, the complexity of the computation, and the technology that makes real-time computation possible. We’ll first explain our technical implementation.

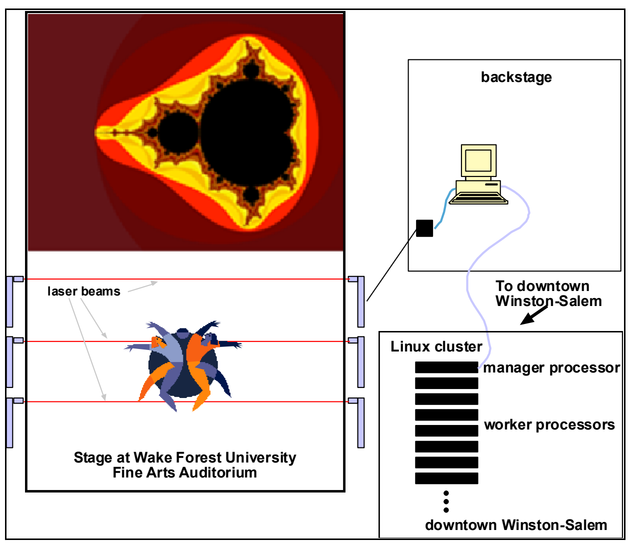


Figure 1. Stage setup for fractal duet in “Fibonacci and Phi”

The scene was designed so that the dancers performed in front of a Mandelbrot fractal which 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 a 1 or 0 depending on the light present. The 0’s and 1’s are forwarded to a MIDI converter. Signals sent from individual lights can be combined in various ways as programmed within the MIDI converter, and the messages can be interpreted as triggers for sounds, 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'll 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 that 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 pictured Figure 1 and the dance is pictured 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, with each step 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 was

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

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

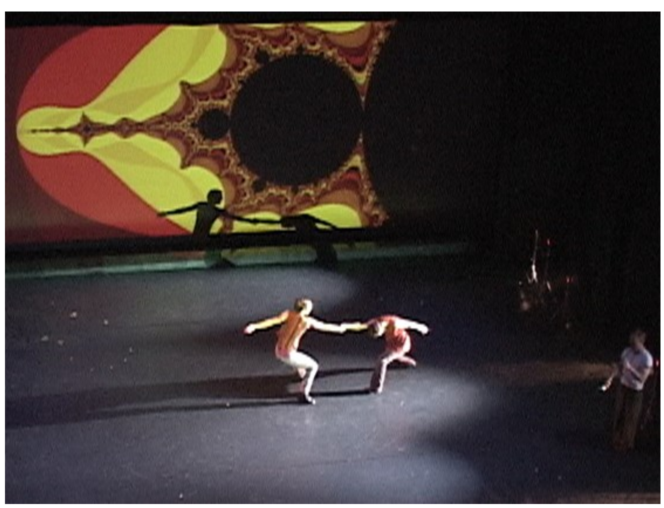


Figure 2. Fractal duet in "Fibonacci and Phi"
(picture courtesy of Ching-Wan Yip, Wake Forest University)

Our parallel version of the fractal computation was 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. Thus, 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.

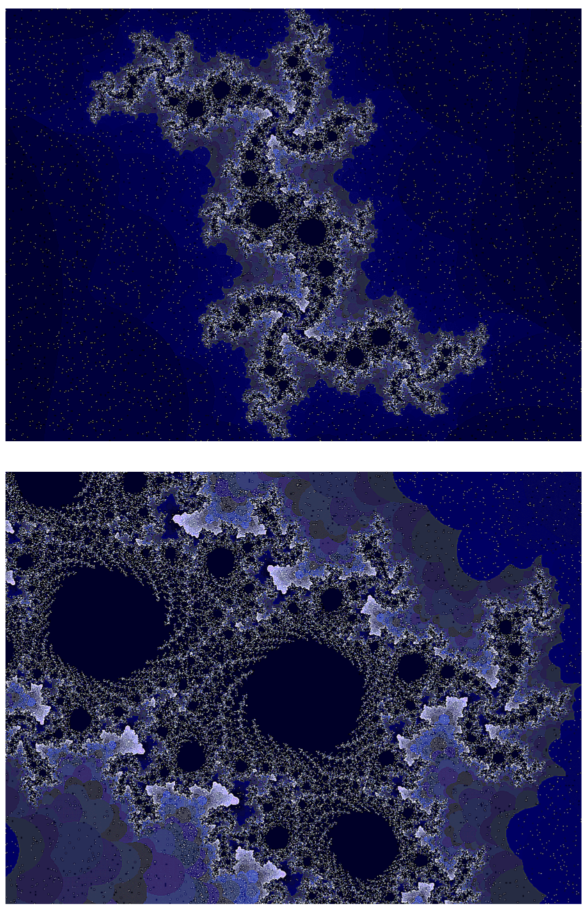


Figure 3. Two views of "Night sky" Julia fractal

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 that 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 chosen 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 of z, respectively. 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 one a little closer than the previous. 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 1 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.

	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

Table 1. Average, minimum, and maximum time to compute a frame for all 100 frames

A factor not accounted for in Table 2 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 3 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.

average	minimum	maximum
9435	3188	20414

Table 2. Average, minimum, and maximum number of bytes per frame communicated per slave to the master over all 100 frames

Table 3 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 gigabit ethernet speed between the client and the cluster makes a significant difference in the refresh rate of the animated fractal.

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

Table 3. Network usage from client to Linux cluster

The benefit of myrinet connectivity within the cluster is also visible in Table 3, 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 this helps to maximize network usage between the cluster and the client.

One conclusion drawn from the analysis was that it would be 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 would be smooth enough animation 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 uses [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, and the dancers also 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 (Figure 4). The dancers also had dialog, speaking in English, French, Italian, German, and Hebrew.

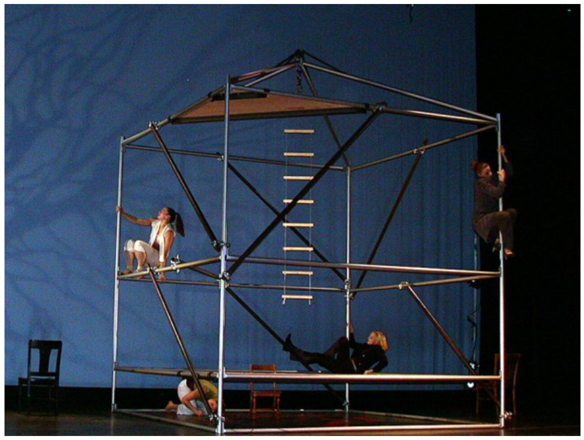


Figure 4. Structure from "Une Journée Abstraite"

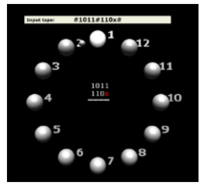
In the discussion below, we focus only on the scenes resulting from the authors' collaboration.

The scene opened with a dark stage and a projection of the computer, which spoke to the audience in its electronic voice. The computer's words appeared on the computer screen as it spoke. 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 reflected images of itself like a mirror within a mirror. Figure 5 links to the projection used in the opening scene.

Click image see projection.

Figure 5. Opening scene of "Une Journée Abstraite"

A highlight of the dance/science collaboration was the Turing Machine Dance, which represented the fundamentals of abstract machine computation by a combination of choreography and digital projections. The dancers enacted a Turing machine program that computes the sum of two binary numbers. At the beginning of the dance, the rules of the program flashed across the screen. During the dance, the digital projection showed 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 Figure 6. The program had twelve states, represented by twelve chairs that descended to the stage dangling from ropes as the dance began. The dancers moved from chair to chair with movements corresponding to inputs, outputs, and state changes.



Click image to see projection.

Figure 6. Turing machine projection from "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 The human element is juxtaposed with a poem spoken by a human silhouette. The poem harkens back to a poem spoken earlier in the piece by one of the dancers.

Click image to see projection.

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

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

Figure 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?

One question we grappled with was how explicit we should 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 is "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 love the power and beauty of computation. The artists are intrigued by having it under their control. Together, they want 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 before-hand 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 to speak to the audience in person about the nature of its computation and limits of its expressiveness.

In the end, the artist's and scientist's conclusions are not exactly the same, but one thing we both agree on is that we are reluctant now to separate ourselves so clearly as artist on the one hand 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."

REFERENCES

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

ALBAN ELVĔD DANCE COMPANY. http://www.albanelved.com/.

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

BURG, JENNIFER AND TIM MILLER. Cluster Computation in Step with Real-Time Dance: An Experiment in Art/Science Collaboration." *Proceedings of the 17th International Conference on Parallel and Distributed Computing Systems*, Sept. 15-17 2004.

CALVERT, TOM, ET AL. Desktop Animation of Multiple Human Figures. *IEEE Computer Graphics and Applications*, Vol. 13, No. 3 (May 1993), 18-26.

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

DOWLING, P., R. WECHSLER, AND F. WEISS. EyeCon: A Motion Sensing Tool for Creating Interactive Dance, Music and Video Projections. *Proceedings of the Society for the Study of Artificial Intelligence and the Simulation of Behaviour*, University of Leeds, England, March, 2004.

FISHWICK, PAUL A., STEPHAN DIEHL, JANE PROPHET, AND JONAS LÖWGREN. Perspectives in Aesthetic Computing, *Leonardo*, Vol. 38, No. 2, 133-141.

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

GUEST, ANN HUTCHINSON. *Dance Notation: the Process of Recording Movement on Paper*. New York: Dance Horizons, 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, PAUL. 2000. Essay: On Biped. http://openended.kaiserworks.com/artworks/pdf/Biped.pdf.
KAISER, PAUL. "Frequently Pondered Questions." In Envisioning Dance on Film and Video. Judy Mitoma, ed. Routledge Press, 2002.
LABAN, RUDOLF. *Choreutics*. L. Ullmann, ed. London: MacDonald and Evans, 1966.
MATH DANCE WITH DR. SHAFFER AND MR. STERN. 2001. http://www.mathdance.org/.
MEADOR, W. SCOTT, ET AL. Mixing Dance Realities: Collaborative Development of Live-Motion Capture in a Performing Arts Environment. *ACM Computers in Entertainment*, Vol. 2, No. 2 (April 2004).
THE MERCE CUNNINGHAM DANCE COMPANY. http://www.merce.org/.
NEW LIFE FORM EVOLVING. *Update Newsletter*, Vol. 3, No. 3 (April 1991). http://www.css.sfu.ca/update/vol3/3.3-Calvert.html. Published by The Centre For Systems Science Simon Fraser University Burnaby, BC Canada.
PALINDROME DANCE COMPANY. http://www.palindrome.de/.
SCHAFFER, KARL, AND ERIK STERN. 2001. Mathematics and Dance Bibliography. Linked from http://www.mathdance.org.
ST. PETERSBURG INTERNATIONAL DANCE FILM FESTIVAL. Program 3. 2003. http://www.kinodance.com/russia/films_program3_2003.html.
TROIKA RANCH DANCE COMPANY. http://www.troikaranch.org.
WECHSLER, R. Computers and Art: a Dancer's Perspective. *IEEE Technology and Society Magazine*, Vol 16 (Fall 1997), pp. 7-14.
WECHSLER, R., FRIEDER WEIß, and PETER DOWLING. EyeCon – A Motion Sensing Tool for Creating Interactive Dance, Music, and Video Projections. *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-TME FRACTAL ZOOMER AND MORPHER. 2005. http://xaos.sourceforge.net/english.php.