SwarmArt: Interactive Art from Swarm Intelligence

Jeffrey E. Boyd
Dept. of Computer Science
University of Calgary
Calgary, Alberta, Canada
boyd@cpsc.ucalgary.ca

Gerald Hushlak
Department of Art
University of Calgary
Calgary, Alberta, Canada
hushlak@ucalgary.ca

Christian J. Jacob
Dept. of Computer Science
University of Calgary
Calgary, Alberta, Canada
jacob@cpsc.ucalgary.ca

ABSTRACT

This paper describes *SwarmArt*, a collaborative project between computer science and art, which resulted in two installations of interactive art that incorporates swarm intelligence. We describe the scientific context of the artwork, technical details of the video system, through which user interaction is controlled, and the realization of the swarm-based simulations, which were projected onto a large screen. Finally, we explain and illustrate our two recent SwarmArt installations in a Calgary gallery.

Categories and Subject Descriptors

J.5 [**Computer Applications: Arts and Humanities**]: Fine arts; H.5.1 [**Information Interfaces and Presentation**]: Multimedia Information Systems—*Animations, video*; H5.2 [**Information Interfaces and Presentation**]: User Interfaces—*Input devices and strategies*; I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Coherence and coordination, multiagent systems*; I.4.8 [**Image Processing and Computer Vision**]: Scene Analysis—*Motion, tracking*

General Terms

Algorithms, Design, Experimentation, Human Factors

Keywords

Swarm intelligence, swarm dynamics, emergence, swarm art, video information server, interaction through video, art installation

1. INTRODUCTION

SwarmArt is a collaboration between an artist and two computer scientists that combines elements of swarm-based systems with interaction through video systems.

The project originates from the idea that the highly dynamic properties of a swarm-based system can be utilized in the following ways:

- Using a dynamic, evolving system provides new tools to create artwork, such as virtual sculptures built by computer-simulated swarms of 'ants'.
- Consequently, the creative process is not completely under the control of the artist, as he/she has to interact with the evolving system, which, in turn, has its own dynamics and can only be influenced by the artist to a certain degree.
- The experiences gained from building interactive art installations that involve swarm-based, evolutionary systems feed back into the sciences of complexity, as we learn to what extent systems of massively parallel, interacting agents can be 'designed' through external influences. This has important consequences for understanding the evolution of such complex biological (swarm-based) systems as the blood clotting cascade, the immune system, or genetic regulatory networks [1].

Hence, science provides new tools for the arts, and the arts do not only raise awareness of emergent phenomena and their usability, but greatly enhance the building of a new mindset to better grasp the enormous potentials of self-organization and evolution in swarm intelligence systems.

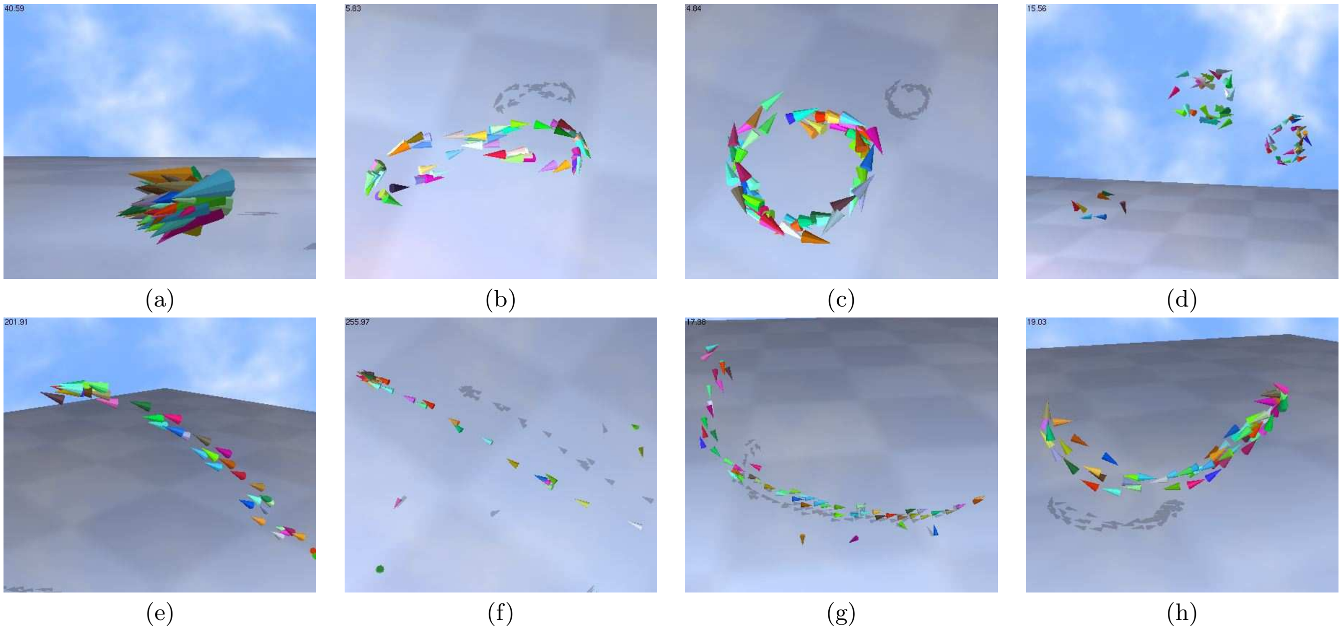
Interest in video processing for a variety of surveillance applications is resulting in a plethora of new technologies that extract descriptive content from video streams. We describe *video information servers* that process video, locate, and describe salient objects in a camera's field of view. These servers provide the descriptions to client applications in the form of XML documents, in addition to provided compressed images. The servers are dynamically configurable to perform a variety of tasks. In *SwarmArt* we use them as the basis for interaction with participants in our installations.

This paper is organized as follows. Section 2 gives a brief introduction to collectively intelligent systems and their importance to understand complex biological systems, their usefulness for complex system designs, in general, and for artwork, in particular. Interaction with swarm-based systems play a major role in our installations; section 3 describes the technical details of the video information system used in our exhibitions. Section 4 describes our two recent interactive installations from 2002 and 2003.

2. SWARM INTELLIGENCE

Studies of biological and physical systems that exhibit emergent properties such as synchronization and self-organization have attracted more and more attention in recent

MM'04, October 10-16, 2004, New York, New York, USA.
Copyright 2004 ACM 1-58113-893-8/04/0010 ...$5.00.



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 1: Examples of evolved swarm formations: (a) tight flocking, (b) figure-eight, (c) double, counter-spin ring, (d) groups of rings, (e-f) line formations, (g-h) lines with ring characteristics. Compare Figure 7 for the use of these formations in the SwarmArt display.

years [13, 12]. Amazingly enough, emergence phenomena are mani-fested in a wide variety of systems, all of which consist of a large number of interacting 'agents' that mainly communicate through local interaction rules. The synchronization mechanisms of fireflies flashing in a seemingly orchestrated fashion fall into this category. However, there is no central conductor firefly; synchrony is achieved through a different kind of self-organization process [21]. Phenomena of self-organization are also responsible for the basic building blocks of living organisms: the gene regulatory network of our body cells constitutes an intricate network of interaction among genes and regulatory and signaling proteins. The dynamics and resulting behaviors of social insects are another primary example of such 'collectively intelligent' systems. Ants have been studied intensively in this context, which has led to the more general notion of *swarm intelligence* [2]:

> "It is no longer possible to use traditional, centralized, hierarchical command and control techniques to deal with systems that have thousands or even millions of dynamically changing, communicating, heterogeneous entities. [...] the type of solution **swarm intelligence** offers is the only way of moving forward, [we] have to rethink the way [we] control complex distributed systems."

Following Hoffmeyer [7], a *swarm*, in the context of collective intelligent systems [23], can be defined as a set of (generally, mobile) agents which communicate with each other — either directly or indirectly — by acting on their local environment. The agents' interactions result in distributed collective problem solving strategies. Swarms of bees or ants, which exchange information through stigmergic pheromones, schools of fish, which determine their behavior (such as swimming direction and speed) based on a small set of locally close swarm mates, or flocks of birds, which communicate by sound and perceptive inputs from surrounding birds, are all examples of such decentralized, yet highly coordinated behavior patterns.

2.1 The Swarm Mindset

Obviously, being able to understand, build and harness the emergent properties of such systems would be highly beneficial. Designers of complex systems could utilize their adaptability and robustness. Such systems would construct themselves, through self-organization. However, system designers and programmers are facing an enormous challenge. How can we actually build highly distributed systems of which we have only limited understanding? Would we have to invent new ways of building, maintaining, and controlling such systems? It seems to be necessary to explore a completely new mindset for programming, system control, and design. But not only engineers, physicists, computer scientists or biologists can profit from the study of complex systems and simulation tools currently being built and investigated (e.g., [20, 14, 4, 11]). Artists can help tremendously in opening up new ways of approaching, presenting, and utilizing emergent system phenomena—in particular when interactive installations are involved, as is the case for our two SwarmArt exhibitions (Section 4).

2.2 Making Them Move

For our swarm simulations we use Breve, a simulation package designed for the modeling of decentralized systems and artificial life in both continuous time and continuous, 3-dimensional space [19]. The Breve swarm simulation we use is an implementation of Craig Reynolds' classic "boids" model [16]. Our simulation is typically composed of 50 individual agents that fly through a virtual 3D world. The agents are confined within a bounded volume of space, and steer away from the boundaries when they get too close. Agents can collide with one another and the ground plane, on which they may land.

The local environment and each agent's behavioral parameters determine the direction and speed of their flights. An agent's instantaneous acceleration vector $\vec{A}$ at each time step of the simulation is calculated as follows:

$$\vec{A} = A_{max} \frac{\vec{V}}{|\vec{V}|}$$

$$\vec{V} = c_1\vec{V}_1(d) + c_2\vec{V}_2 + c_3\vec{V}_3 + c_4\vec{V}_4 + c_5\vec{V}_5,$$

where

c_1 to c_5 are the weightings for the corresponding 'urge' vectors ($\vec{V}_1$ to $\vec{V}_5$) described below,

d is the preferred distance an agent tries to maintain with its neighbors (crowding radius),

$\vec{V}_1$ is a vector that points away from neighbors that are within d,

$\vec{V}_2$ is a vector toward the center of the world,

$\vec{V}_3$ is the average of the agent's neighbors' velocity vectors,

$\vec{V}_4$ is a vector toward the center of gravity of the agent's neighbors

$\vec{V}_5$ is a random unit-length vector,

$\vec{V}$ is the weighted summation of behavioral urges,

$\vec{A}$ is an agent's instantaneous acceleration vector at each time step of the swarm simulation with $|\vec{A}| \leq A_{max}$, and

A_{max} is the maximum acceleration of an agent.

In order to choreograph interesting dynamics for the swarm art installation, we have utilized an evolutionary breeding system, which helped us to tune the parameters d, A_{max}, and the weights c_1 through c_5 (more details are described in [14]). Not all parameters affect the behaviors at the same scale; for example, the crowding radius (d) has a more immediate influence on the overall dynamics than the agents' maximum acceleration (A_{max}). Figure 1 shows four examples of evolved swarm behaviors, which were subsequently built into the SwarmArt installation. In order to switch between different swarm formations and to adjust some of the control parameters we used a video interface as described in the following section.

3. VIDEO INTERACTION

Interaction is an essential part of SwarmArt. We implemented a visual interface that uses video cameras and common computer vision algorithms embedded into a multimedia *video information server.* Although originally designed for video surveillance tasks, without modification, these servers provide an excellent mechanism for visual interaction with art installations. The remainder of this section describes the servers and their use in SwarmArt.

3.1 Video Information Servers

Machine-vision based video surveillance systems perform several common functions required for low-level processing of video data. These algorithms include acquisition of images from a camera, calibration of a camera [6], segmentation of an image into foreground and background pixels [22, 8], location of objects that are connected components of foreground pixels, computation of object descriptions, and object tracking within a camera's field of view [5]. In addition to these low-level functions, surveillance systems may also perform object tracking in world coordinates, behavior recognition from object trajectories, and storage and retrieval of trajectories for long-term analysis, among others.

Recent advances in the speed and size of digital signal processing hardware make it possible to embed many of the low-level functions listed above into a video camera. For example, see the work of Wolf et al. [24]. Cameras with such embedded functionality have the option to provide not only raw pixels, but metadata that describe the locations and motion of objects viewed by the camera. The availability of metadata simplifies the implementation of surveillance systems that perform the high-level functions listed. While cameras with embedded metadata production can transmit data over a variety of media, use of computer networks is convenient because it uses established infrastructure as well as exploits the abundance of network devices that are already available. In order to distinguish metadata cameras from network cameras that provide only compressed video, we refer to these cameras as *video information servers.*

Sayles et al. [17, 3] give details on the operation of the video information servers we use for SwarmArt. The servers provide, via a network, a description of camera functions, a raw video stream, and a stream of documents that describe detected objects and their trajectories in image coordinates. The documents use an XML-based language with tags for camera interaction.

Our servers are *dynamically configurable.* Clients configure the server by sending XML documents that direct the server to *load* modules that provide desired functionality. For video information servers, we have modules to perform tasks such as user access control, video capture, video compression, object detection (using various segmentation techniques), and object tracking. Note that by changing the module responsible for generating XML tags for object description, the server could produce MPEG-7 documents.

The modules operate in a pipeline architecture where the output of a module can provide input to one or more other modules in a pipeline. A graphical editor allows a user to specify the topology of the pipeline and configure each module. When the user is satisfied with the configuration, the editor creates an XML document to load and configure the desired modules. Figure 2 shows the graphical configuration editor in use. The server is implemented in the Python programming language, and makes use of the Intel Integrated Performance Primitives Library [10] to optimize execution speed.

At the time of writing, we have a video information server running continuously, observing traffic at the University of Calgary. A Java applet at **http://toaster.-cpsc.ucalgary-.ca/** shows the data available from this server.

3.2 Interaction via Video Information Servers

Video information servers shift the development effort in a video surveillance system from video and image processing algorithms to manipulating data retrieved in XML documents. Given the broad acceptance of XML and the tools

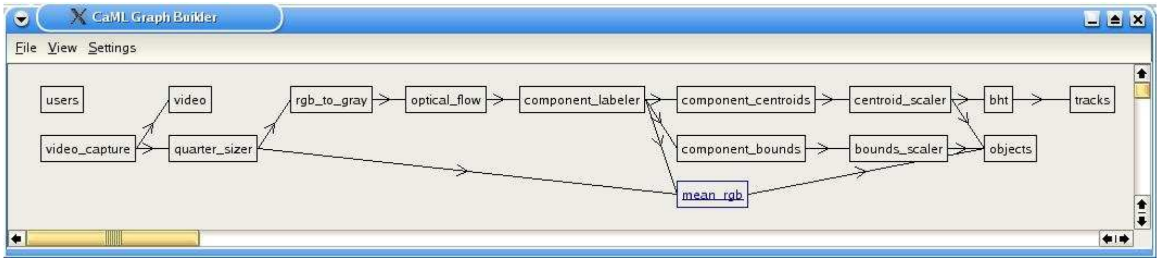


Figure 2: Screen capture of the graphical configuration editor. The editor creates XML documents that can be used to configure a server for a desired task. Note that the editor can send the documents to the server as they are edited, allowing a user to tune server performance interactively.

available to manage, send and retrieve documents, this results in a major simplification of video system development. Thus, it is natural to employ a video information server as a tool for interaction in art. The server provides a tool that can handle complex visual tasks while giving the creator of artistic software a simple interface to the visual information. In both of our SwarmArt creations to date, the inclusion of a visual interface proved to be one of the simpler tasks, because the server is easy to use.

Dynamic configuration of the servers allows us to select various video processing algorithms as circumstances dictate. For example, we can change the way in which the server detects objects that it sees. In applications where objects are rigid, e.g. a car, or are articulated but far away, e.g., a walking human observed from a distance, the camera sees a clumped mass of moving pixels. In this case, an optical flow or background subtraction algorithm is sufficient for detecting moving objects. SwarmArt 2002 (Section 4.1) uses Lucas-Kanade optical flow [15] to track the motion of participants.

However, when a human is viewed at close range, the scale of motion tends to shift between motion of an entire body, e.g., when a person walks directly in front of the camera, and motion of individual limbs, e.g., when a standing person gesticulates. In these cases, optical flow motion detection alternately tracks a person or their limbs. This is not desirable for surveillance tasks because it leads to confusing trajectories that are difficult to relate to real-world objects. In SwarmArt 2002 we observed this behavior, but fortunately, this added variability to the exhibit and made it more interesting.

We can change the object detection algorithm to be sensitive to colors. For example, we can *tune* a color segmentation algorithm to recognize skin tones, or any other color we choose. By relying on color rather than motion, the server can track humans, or objects that they manipulate in close quarters, provided that the objects of interest have a distinctive color. In SwarmArt 2003 (Section 4.2) we tune the server to find objects corresponding to the colors of balls placed on a table. In doing so, participants can interact with the exhibit by moving balls placed freely on a table. Although the servers behaved differently in the two Swarm-Art exhibits, the same XML-documents-via-network interface applied in both cases. We merely reconfigured the modules that operate within the server.

Although video information servers significantly reduce the programming burden for any video processing system, the interface still requires some programming skill. Since SwarmArt was a collaboration between computer scientists and artists, the programming skill came from the computer scientists. Still, no video processing code beyond that which already existed in the servers was required. As XML continues to develop, we expect that tools will emerge that will allow non-expert programmers to use data from a video information server.

EyesWeb [9] provides a system with capabilities similar to our video information servers, i.e., a variety of video processing algorithms and a graphical environment for combining them, but with a focus on applications in performing arts. In contrast, our servers embed processing capabilities into a camera, interacting with multiple clients using XML over a network. The focus has been on surveillance applications, with applications to artistic endeavors coming as an unexpected, but fortunate side effect.

4. INSTALLATIONS

We have constructed two major installations based on the SwarmArt concept. Both of these were shown in the Nickle Arts Gallery on the University of Calgary campus. The following two sections describe these installations.

4.1 SwarmArt 2002

Figure 3 shows our first SwarmArt installation in schematic form. A swarm simulation runs on an Apple Power Mac G4 computer running OS X. The simulation displays the swarm activity as colored dots that move around the computer display. Each dot has a tail that fades with time so that both the current position of the swarm entities, and their motion over the last few seconds are visible in the display. An LCD projector projects the swarm onto a wall for participants to view.

A video information server aimed at a region in front of the projected swarm monitors the motion of the participants. The swarm simulation retrieves XML documents from the server, extracts the image coordinates of the moving objects (we assume the objects are the participants), then uses the coordinates as input into the simulation. Since the camera used by the server requires light to function properly, we have a conflict with the project which works best in a darkened area. Some careful control of the lighting allowed us to arrange the installation so that we had adequate light for the camera, with unduly affecting the projection.

Figure 4 shows images taken while testing SwarmArt 2002 in the laboratory before installation in the Nickle Arts Gallery,

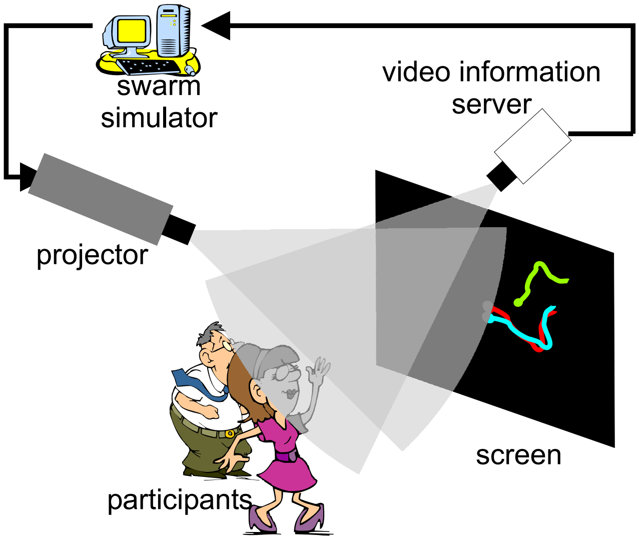


Figure 3: Schematic diagram of SwarmArt 2002.

December 2002. The participants see the swarm follow their motion around the exhibit as they walk in front of the projected display. As participants walk closer to the projection, the swarm moves down, and the swarm follows *moving* participants as they move left to right.

The motion of the swarm tends to lag behind the motion of participants for two reasons. First, the motion detection algorithm in the video information server waits until a moving object persists for several video frames before it reports the object in its XML documents. This has the desirable effect of eliminating spurious motion that could confuse the simulation, but it also means that motion must be sustained before the swarm will react to it. Second, the swarm itself does not react instantaneously to the presence of a moving object. The swarm only moves toward an object if it persists for three or four seconds.

Adults who entered the installation behaved as though they were self-conscious, and tended to move very little. When they did move, it was typically only very briefly, and did not sustain motion long enough for the swarm to react. Thus, the swarm did not notice adults much, and the adults lost interest in the swarm. The most delightful interaction with the installation came from children, who would run into the installation, and continue to move. They were rewarded when the swarm followed their motion. One author observed several children playing in the installation. Not only did the children interact with each other in their play, but they were also playing with the swarm. Children viewed SwarmArt 2002 as it was intended.

4.2 SwarmArt 2003

Figure 5 shows SwarmArt 2003 in schematic form. Again, a swarm simulation runs on an Apple Power Mac G5 computer running OS X. Entities in the swarm are portrayed with scanned images of 5.25-inch floppy disks (Fig. 7). The disks were previously removed from their jackets and painted with acrylics. Some of the disks have sections cut out. Mixed in with the images of disks are renderings of solid-colored spheres. Although the swarm is rendered in a two-dimensional image, the swarm simulation is three dimensional. An LCD video projector projects the swarm rendering onto a white wall. Figure 6 shows the actual installation in the Nickle Arts Gallery, December 2003.

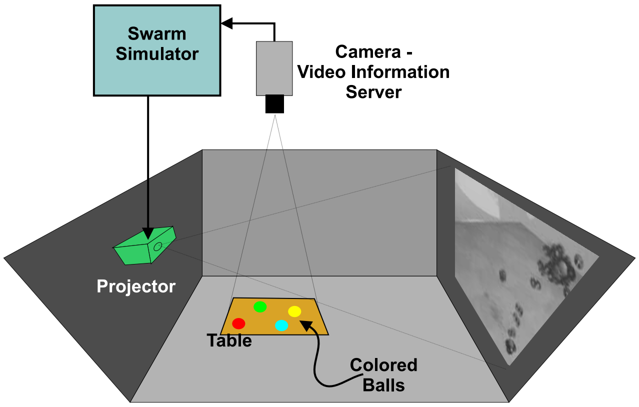


Figure 5: Schematic diagram of SwarmArt 2003.

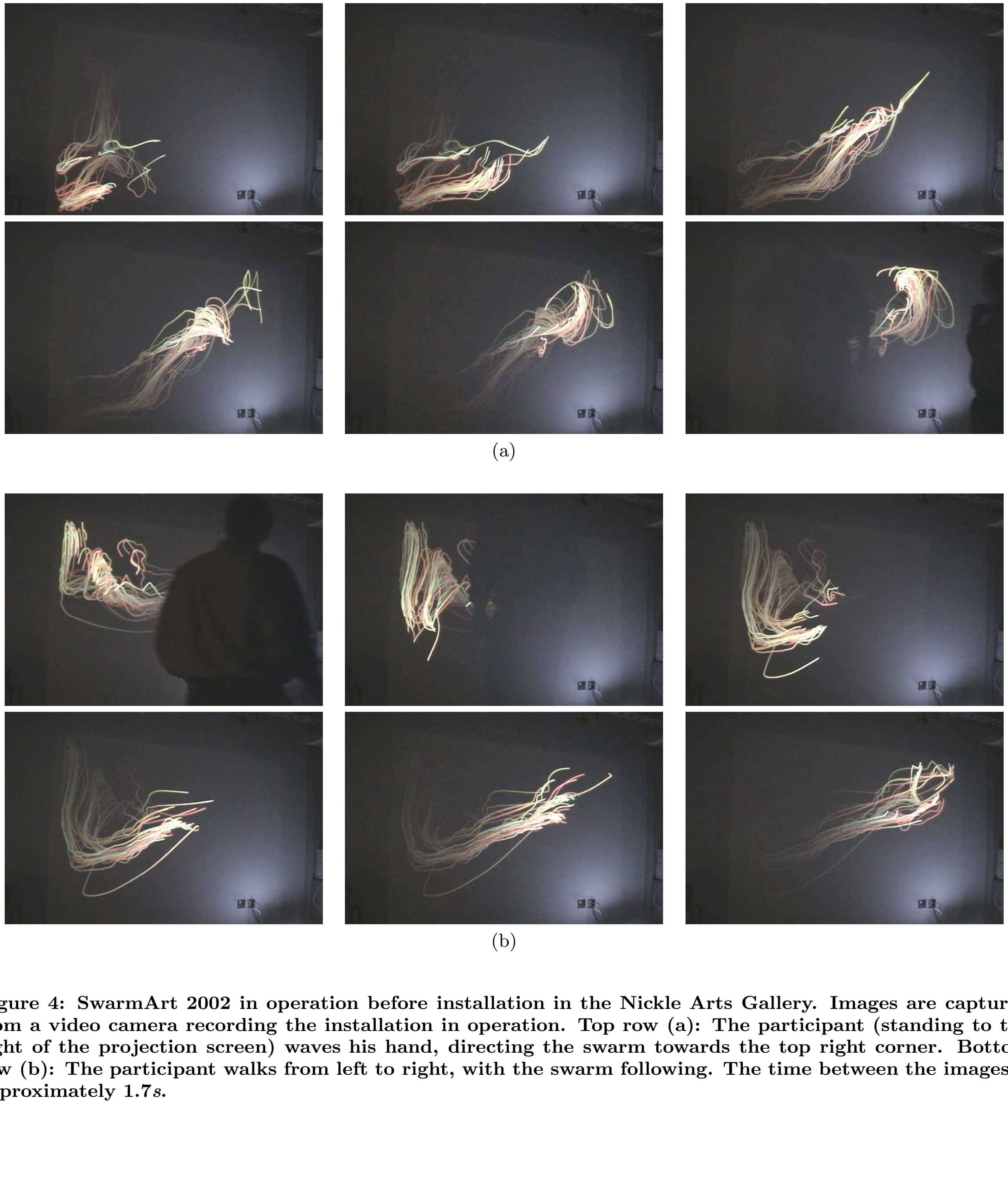
A video information server enables interaction with SwarmArt 2003. However, unlike 2002, SwarmArt 2003 does not react to motion. Instead, participants interact by altering the positions of four colored balls (about the size of boccie balls) placed on a table near the center of the installation (the table was fabricated from solid cedar beams fastened together by large woodworking clamps, Figure 6(b)). We place the camera and lighting for the table in the ceiling (Figure 6(c)), and configure the video server to track the colors of the balls on the table. The positions of three of the balls set parameters that control the simulation and display— e.g., swarm behavior (as described in Section 2.2), viewing angle, and zoom. The fourth ball does not affect the swarm in any way. Figure 7 shows photographs of some of the images projected by SwarmArt 2003. Periodically, the simulation ignores the positions of the balls and zooms in on the swarm to highlight the intricate scanned, acrylic images on the disks (Figure 7(d)).

SwarmArt 2003 proved more interesting to the staid behavior of adult participants. Tracking colors instead of motion meant that a participant could move a ball, then stand back and watch the reaction of the swarm.

5. CONCLUSIONS

Generating form and relationships using our artistic tools of swarm intelligence creates a symbiotic nurturing partnership between science and art. The real art is about identifying the potential of the swarms and anticipating the moment at which the apparent disorders become aesthetic curiosities and catalysts for thought. A flock of birds, a school of fish, a herd of cattle, a pod of whales, all establish relationships similar to swarm dynamics. However, it is only on the rare occasion that these related constituents position themselves in space, forming a whole that is interesting enough to name as sculpture or transcend time. Through a better understanding of swarm processes we attempt to create and share these playful sculptural moments in our museum installations.

We also found that the video interface that we used to control the dynamics of swarm systems helps us to develop new



(a)

(b)

Figure 4: SwarmArt 2002 in operation before installation in the Nickle Arts Gallery. Images are captured from a video camera recording the installation in operation. Top row (a): The participant (standing to the right of the projection screen) waves his hand, directing the swarm towards the top right corner. Bottom row (b): The participant walks from left to right, with the swarm following. The time between the images is approximately 1.7*s*.

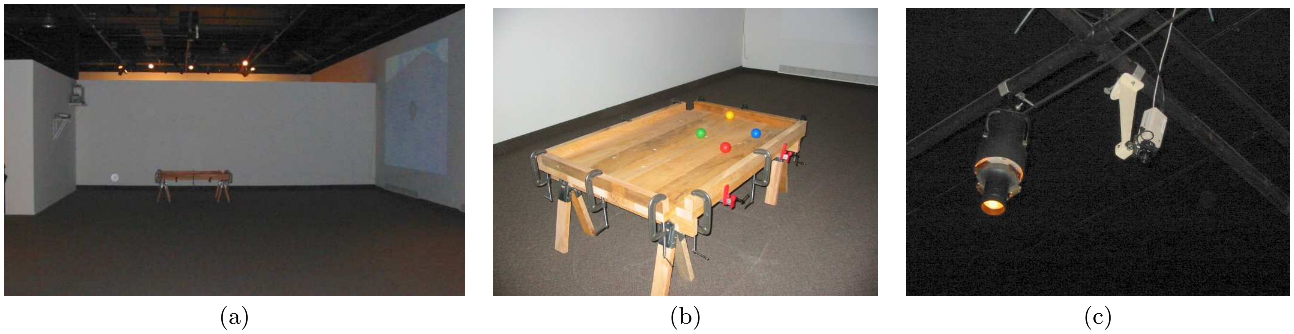


Figure 6: SwarmArt 2003: (a) installation in the Nickle Arts Gallery, (b) the table with balls that is the point of interaction, and (c) the camera and lighting installed in the ceiling above the table.

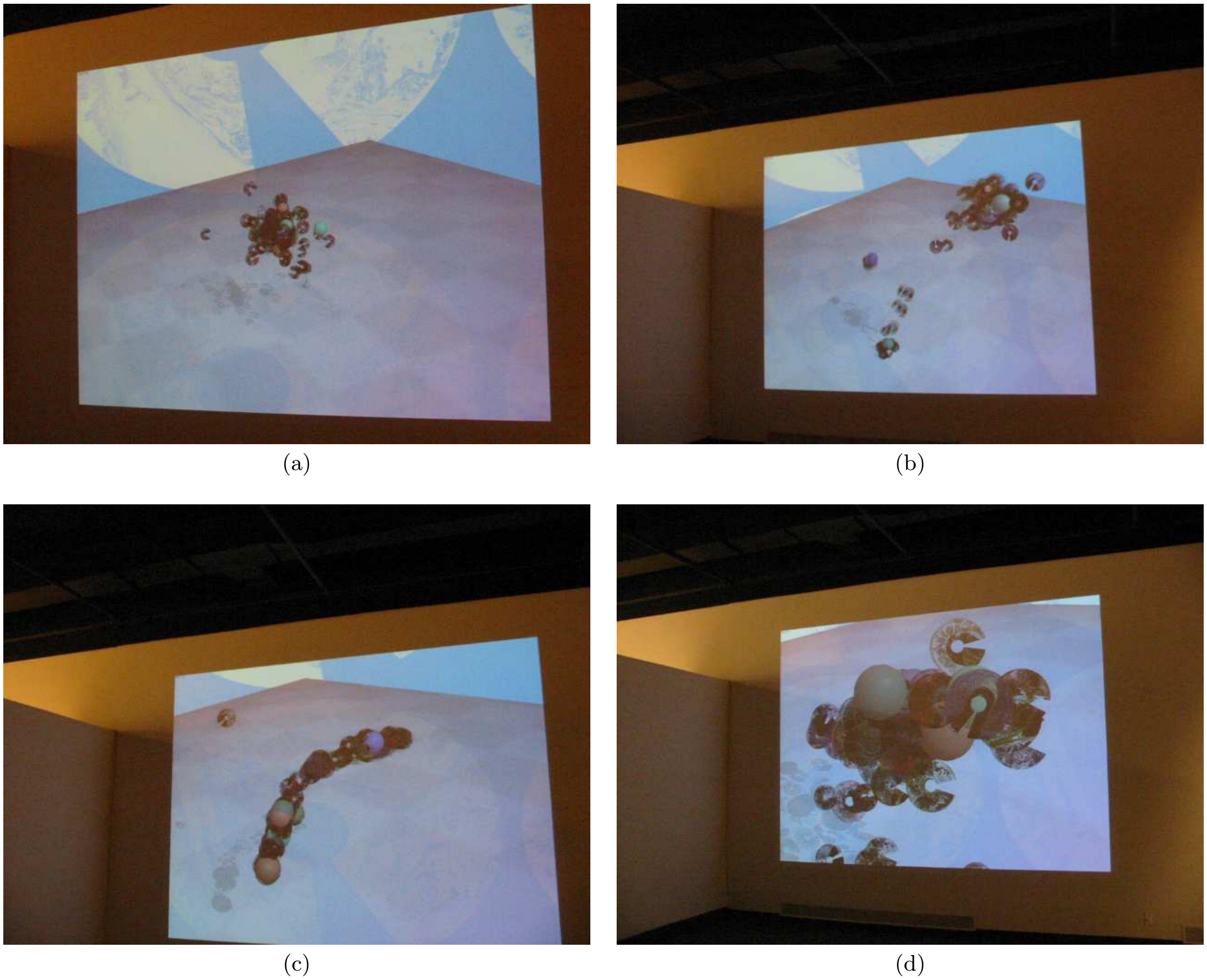


Figure 7: Photographs of the dynamic swarm system projected in SwarmArt 2003. The behaviors of the swarms are partly influenced by user interaction through the movement of colored balls on a table (Fig. 6(b)): (a) Tight flocking; (b) breaking up a tight cluster; (c) transformation from a cluster into a line formation; (d) time-triggered zoom onto a sculpture-forming swarm cluster.

interaction devices for large-screen 2- and 3-dimensional displays, which we are currently testing in a virtual immersive CAVE environment, where we build 3D, agent-based models of biomolecular systems for applications in the life sciences [4, 11, 18]. In future installations, we will also incorporate audio input, as we found that many times it is easier to detect a specific sound (such as hand clapping) rather than a particular gesture through video.

The websites **www.cpsc.ucalgary.ca/∼jacob/ESD** and **www.swarmart.com** provide further information on our recent and future work related to Swarm Art and Evolutionary Swarm Design.

6. ACKNOWLEDGEMENTS

This work is supported by grants from NSERC and the Canada Council for the Arts. We thank our students Euan Forrester and Paul Nuytten for helping us implement prototypes of swarm paintings and the first version of SwarmArt.

7. ADDITIONAL AUTHORS

Additional authors: P. Nuytten and M. Sayles, Dept. of Computer Science, University of Calgary, Calgary, Alberta, Canada, email: {nuytten,sayles}@cpsc.ucalgary.ca.

8. REFERENCES

[1] M. J. Behe. *Darwins Black Box: The Biochemical Challenge to Evolution*. Touchstone, Simon & Schuster, New York, 1998.

[2] E. Bonabeau, M. Dorigo, and G. Theraulaz. *Swarm Intelligence: From Natural to Artificial Systems*. Santa Fe Insitute Studies in the Sciences of Complexity. Oxford University Press, New York, 1999.

[3] J. E. Boyd, M. Sayles, L. Olsen, and P. Tarjan. Internet broadcast of hockey: a scale prototype. In *SPIE Internet Imaging V*, Santa Jose, CA, January 2004. accepted.

[4] I. Burleigh, G. Suen, and C. Jacob. Dna in action! a 3d swarm-based model of a gene regulatory system. In *First Australian Conference on Artificial Life*, Canberra, Australia, 2003.

[5] I. J. Cox and S. L. Hingorani. An efficient implementation of reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(2):138–150, 1996.

[6] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge University Press, Cambridge, UK, 2000.

[7] J. Hoffmeyer. The swarming body. In I. Rauch and G. F. Carr, editors, *Semiotics Around the World. Proceedings of the Fifth Congress of the International Association for Semiotic Studies*, pages 937–940, Berkeley, 1994. Mouton de Gruyter.

[8] T. Horprasert, D. Harwood, and L. Davis. A statistical approach for real time robust background subtraction. In *IEEE Frame Rate Workshop*, 1999.

[9] D.-U. o. G. InfoMus Lab. The eyesweb project. www.eyesweb.org, 2004.

[10] Intel Corporation. *Intel Integrated Performance Primitives for Intel Architecture: Reference Manual*, 2003.

[11] C. Jacob and I. Burleigh. Biomolecular swarms: An agent-based model of the lactose operon. *Natural Computing*, 2004. (in print).

[12] S. Johnson. *Emergence: The Connected Lives of Ants, Brains, Cities, and Software*. Scribner, New York, 2001.

[13] S. Kauffman. *At Home in the Universe: The Search for Laws of Self-Organization and Complexity*. Oxford University Press, Oxford, 1995.

[14] H. Kwong and C. Jacob. Evolutionary exploration of dynamic swarm behaviour. In *Congress on Evolutionary Computation*, Canberra, Australia, 2003. IEEE Press.

[15] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *International Joint Conference on Artificial Intelligence*, pages 674–679, 1981.

[16] C. Reynolds. Flocks, herds, and schools: A distributed behavioral model. *Computer Graphics*, 21(4), July 1991.

[17] M. Sayles, X. Wu, and J. E. Boyd. Caml: Camera markup language for network interaction. In *SPIE Internet Imaging IV*, volume 5018, pages 248–256, Santa Clara, CA, January 2003.

[18] C. W. Sensen. Using cave technology for functional genomics studies. *Diabetes Technology & Therapeutics*, 4:867–871, 2002.

[19] L. Spector and J. Klein. Evolutionary dynamics discovered via visualization in the breve simulation environment. In *Artificial Life VIII*, Reading, MA, 2002. Addison-Wesley.

[20] L. Spector, J. Klein, C. Perry, and M. Feinstein. Emergence of collective behavior in evolving populations of flying agents. In *Genetic and Evolutionary Computation Conference (GECCO-2003)*, pages 61–73, Chicago, IL, 2003. Springer-Verlag.

[21] S. Strogatz. *Sync The Emerging Science of Spontaneous Order*. Theia Books, New York, 2003.

[22] E. Sudderth, E. Hunter, K. Kreutz-Delgado, P. H. Kelly, and R. Jain. Adaptive video segmentation: theory and real-time implementation. In *1998 Image Understanding Workshop*, volume 1, pages 177–181, November 1998.

[23] F. T. Vertosick. *The Genius Within: Discovering the Intelligence of Every Living Thing*. Harcourt, New York, 2002.

[24] W. Wolf, B. Ozer, and T. Lv. Smart cameras as embedded systems. *Computer*, 35(9):48–53, September 2002.