Identity SA 1.6

An artistic software that produces a deformed audiovisual reflection based on a visually interactive swarm

Tatsuo Unemi
Soka University
1-236 Tangi-machi
Hachioji, 192-9577 Japan
unemi@iss.soka.ac.jp

Yoshiaki Matsui
Soka University
1-236 Tangi-machi
Hachioji, 192-9577 Japan
e08m5221@soka.ac.jp

Daniel Bisig
ICST, Zurich Univ. of the Arts
Baslerstrasse 30
CH-8048 Zurich, Switzerland
daniel.bisig@zhdk.ch

ABSTRACT

Identity SA is an interactive and generative installation that combines a swarm-based simulation with real time camera based interaction. The agents' distributions are transformed into painterly images by employing a variety of different visualization techniques and styles. Camera based interaction is based on a simple motion detection algorithm that affects the agents' movements as well as their coloring. Identity SA acts as a visual and acoustic mirror, which distorts the continuity of the visitor's physical existence into ephemeral patterns and flowing motions. The new version 1.6 adds the capability to select musical tracks depending on the agents' movements. This system has been employed both as an experimental installation in an exhibition and as interactive video projection for a contemporary dance performance.

Categories and Subject Descriptors

J.5 [**Arts and Humanities**]: Miscellaneous

General Terms

Interactive Swarm-based Art

1. INTRODUCTION

Identity SA is a computer based interactive artwork that creates abstract reflections of the visitor's appearance. It acts as a visual and acoustic mirror, which distorts the continuity of the visitor's physical existence into ephemeral patterns and flowing motions. It reminds the visitor of the fragility of his/her own self-awareness and existence. The installation mimics an abstract painting whose moving brush strokes create a portrait of the visitor. The integrity of this portrait can only be preserved by the sustained activity of the visitor. Whenever he/she stops moving, the portrait becomes increasingly vague and fuzzy and eventually disappears entirely in a swirl of colors. This continuous struggle versus one's own disintegration metaphorically reflects our permanent exposure to the treat of losing ourselves through mental and physical decay. According to Buddhist teachings, sickness and aging together with birth and death form the group of four inescapable sufferings[3]. The suffix in the title for this project stands for sickness and aging.

In our former works [6, 2, 8], we have examined combinations of swarm simulation and camera-based visual feedback in order to realize pieces of generative art. Several other artists have chosen similar approaches in their works[4]. These works employ tens or hundreds of agents moving in a virtual 3D space to play music and/or to draw images. In our newest work presented here, we increase the number of agents to thousands but reduce the dimensionality of the agent world to 2D. As the agents move through this flat canvas space, animated patterns are created based on their distributions rather than their movements.

Throughout the next sections, we explain the technical implementation of the software. This includes descriptions of the swarm simulation, drawing methods, camera-based interaction, sound synthesis, and choreographed parameter transitions. Subsequently, we present the results of two public presentations of the software: as an experimental installation in an exhibition and as interactive video stage effect for a contemporary dance performance. We conclude this paper by outlining possible directions for future extension.

2. SWARM SIMULATION

In Identity SA, visual and acoustic output is created via a swarm simulation. A swarm consists of a collection of mobile agents each of which behaves according to simple rules. These rules take the interaction between neighboring agents and the local environment into account. Swarm simulations are mainly used as a computational model for group movement behaviors observed in nature such as in a school of fish and a flock of birds. In addition, these simulations are successfully applied as an optimization technique for distributed systems[1]. Similar to other models of complex systems, swarm simulations can be employed to create surprising results despite the fact that they are based on entirely deterministic algorithms. For these reasons, swarm simulations can be of interest for generative art.

Idendity SA implements agent behaviors that are based on the classical BOID's algorithm[5]. Accordingly, each agent is controlled by a set of forces that causes the following behaviors: (1) avoidance of collisions with close-by agents and

Advances in Computer Entertainment Technology 2008, Yokohama, Japan.

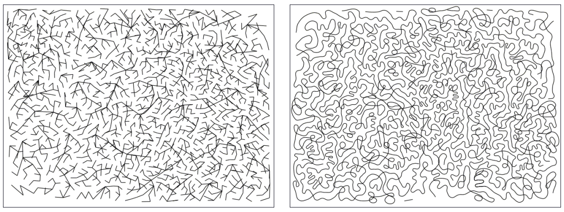

Figure 1: Examples of drawing line segments that connect the positions of agents.

obstacles, (2) velocity alignment with neighboring agents, and (3) movements towards the perceived center of neighboring agents. The repelling forces for collision avoidance are proportional to the inverse of the square of the distance between the agent and the obstacle. Based on the sum of all the forces that affect an agent, a goal angle and a goal speed are calculated. The agent tries to meet these goal values by modifying its current orientation and velocity within the allowed limitations of its steering angle and acceleration. Similar to the implementation in our previous project[7], the agents are divided into two species. This approach leads to a richer variety in swarm behaviors as compared to a classical single species BOID's swarm.

In our implementation, a swarm consists of more than 2,000 agents. The agents move in a two dimensional Euclidean space bounded by the rectangular borders of the screen. Due to recent improvements in computational power, it became feasible to simulate such a large swarm in real time on a single personal computer. The system is currently implemented on an Intel-based 2 GHz Core 2 Duo CPU running MacOS X 10.5.

To calculate forces among agents, neighborhood relationships need to be determined. If we used an exhaustive algorithm to check the distance between every possible pair of agents, the computational time complexity would be proportional to square of the number of agents. To reduce this complexity, we introduced a method that divides the 2D space into a number of sub-areas. Each of these sub-areas maintains information about the agents contained within. For this reason, we can restrict distance calculations to agent pairs that reside within neighboring sub-areas.

3. DRAWING METHODS

The agents' positions and movements are transformed into a *painting* by employing a variety of visualization techniques and styles. The agents themselves can be drawn in one of the following six different styles. (1) A triangle pointing towards the agent's movement direction, (2) a textured surface depicting a predefined pattern, (3) spray spots that are randomly distributed around the agent's position, (4) short line segments that are parallel to the agent's movement direction, (5) a hexagon drawn in such a way that it looks like a 3D cube, and (6) a glyph that is selected from a predefined set of Roman, Japanese and Chinese characters. Each of the shapes listed above is drawn in a single color that is selected based on the agent's orientation angle.

Neighborhood relationships can be displayed as connections between agents' positions in the following two different styles: (1) a set of lines that connect each agent to its nearest neighbor, and (2) a single curved line that forms a chain through all agents' positions. The curved line is a sequence of cubic Bezier curves that smoothly connects the positions of agents. The maximum segment length, the line width, and the roundness of the curved line are adjustable parameters. Figure 1 depicts examples of these two drawing styles. Each line segment is drawn in a color that is determined by the orientation of the agent at the start point of the line segment.

Finally, a number of different types of post-processing effects such as blur and afterimage are also applicable. The afterimage can be rotated and magnified by an amount that is proportional to the intensity of visitor movement. Combinations of these effects and different transparency settings allow the creation of a wide variety of aesthetically complex images.

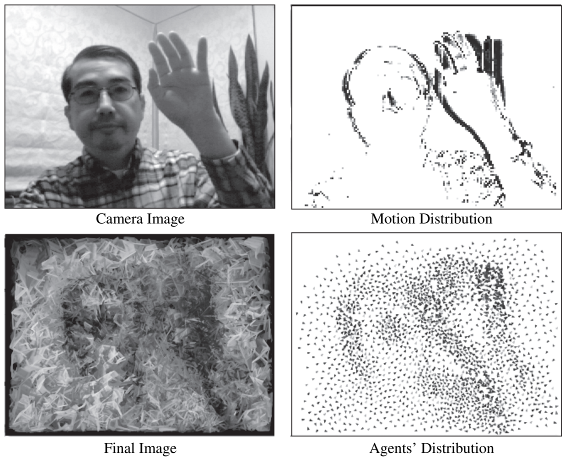

Figure 2: Camera-based interaction.

4. CAMERA-BASED INTERACTION

Camera-based interaction is based on a simple motion detection algorithm that affects the agents' movements as well as their coloring. Basically, motion is detected by calculating the color difference between the previous and current frame at each pixel position. To avoid errors in motion detection due to noise and light fluctuations, frames are accumulated into an frame memory buffer by calculating a weighted sum between this memory buffer and a newly acquired frame.

The next step involves the calculation of the absolute scalar values of the pixel differences between subsequent frames. The current implementation selects the maximum difference among the three RGB color components. This algorithm results in the creation of a single channel image that encodes the amount of detected motion in each pixel. Each agent responds to this tracking information by adding a force towards the center of the detected motion that lies within its local neighborhood.

In the absence of interaction, an agent's color is solely determined by its orientation as described in the previous section. When the tracking system detects motion, the agent's color is additionally affected by the corresponding pixel color in the camera frame. The ratio of orientation and interaction based color change is proportional to the amount of detected motion. Accordingly, the visitor's image is reflected when he/she is moving, but it disappears when he/she stops moving.

Figure 2 depicts different stages in video tracking and visual feedback generation. The motion distribution image (Figure 2, top right) is calculated from the camera image (top left). The agent distribution (bottom right) is affected by the motion distribution image. The final visual output (bottom left) is created by taking both the agent distribution and original camera image in account.

5. SOUND AND MUSIC

Three types of sound effects are currently implemented: synthesized sound, modified sampled sound, and remixed music. All these sound effects are affected by the movement of agents and visitors. Only a subset of all agents is involved in the generation of sound. The probability that an agent creates a sound is proportional to the square of its angular velocity in order to make it easy for the visitor to recognize the reaction. Agents that are allowed to generate sounds are organized in a queue of fixed length. In the current implementation, the queue can hold 12 agents. If a new agent is selected for sound generation and the queue is already full, the new agent replaces the agent that has been in the queue for the longest time. The duration of a generated sound depends on how strongly an agent is attracted by visitor movement. Strongly attracted agents tend to produce a short sounds whereas agents that don't respond to interaction create long sounds. Finally, white noise is added to the audible output. The volume of the noise is proportional to the amount of motion captured by the camera. Stereo panning of the noise is controlled by the position of the center of gravity of the detected motion.

The following part of this section describes the generation of sampled sound and remixed music. The creation of synthesized sound as already been described elsewhere[9].

Modulated Sampled Sound.

This method employs pre-recorded sounds that are imported from audio files stored in an assignable folder. The audio data can be stored in any format that is supported by the Audio Toolbox of Macintosh OS X. Each agent plays back its own allocated audio file. As in the case of synthesized sound, audio panning is controlled by an agent's horizontal position and playback speed depends on it's vertical position. According to this playback speed, an amplitude envelope is generated and this envelope is applied in the same way as for the synthesized sound. A sound track is created by repeating the sampled audio data as many times as is needed to reach the required playback duration.

Remixed Music.

This method remixes pre-recorded sounds according to a uniform rhythm and loop interval. Usually, these tracks correspond to recordings of individual musical instruments such as a drum set, bass, piano and so on. In the current implementation, the screen space is evenly divided into consecutive vertical regions and each track is assigned to one of these regions. Tracks are selected for playback depending on the agent's horizontal position. For this method, the agents' vertical position is not taken into account. Unlike the modulated sampled sound version, the playback of the selected tracks is synchronized to a global clock. In case the selected tracks differ in length, shorter tracks are iterated in order to match the length of the longest track. Accordingly, all tracks restart at exactly the same time.

Figure 3: Photographs taken from the exhibition room in the campus festival in October 2007.

6. CHOREOGRAPHED TRANSITION OF PARAMETERS

The swarm simulation, as well as its visualization and sound generation can be controlled by scripting parameter changes in the AppleScript programming language. Scripting offers an easy approach to create choreographed transitions in the swarm's visual and acoustic appearance. Both parameter changes and their timing can be freely chosen and therefore allow to create a wide variety of choreographed scenarios. A large number of parameters can be modified via scripting, such as parameters that control agent shapes, agent colors, background color, blurring, afterimage, interaction between species, and flocking behaviors. A particularly interesting trasition is achieved by changing the parameter that controls the interaction among the two agent species. Depending on this parameter setting, agents from different species tend to mix uniformly or separate into distinct clusters. As with version 1.5 of the software, a script the randomizes parameter changes is executed by default. This script offers a user the opportunity to enjoy different visual and acoustic transitions whenever they restart the program.

7. EXPERIMENTAL EXHIBITION

We have organized experimental exhibitions of this software on three occasions: the campus festival of the Soka University in Hachioji in October 2007, the Generative Art Conference in Milano in December 2007, and the open campus event of the Soka University in March 2008. Figure 3 shows a photograph of the exhibition space at the campus festival. Our software was running on a hardware setup that consisted of a 3 GHz quad-core MacPro computer, a 32 inch LCD display, an iSight camera, and a pair of active speakers. This computational power was sufficient to simulate 2,000 agents at a rate of 25 frames per second and a resolution of 1024 by 768 pixels. Our observations of the visitor's reactions have been very encouraging. New visitors quickly understood the interaction possibilities of the system when they perceived how their reflected image emerged in the visual output. Some children and young persons then started to move their body more actively or even began to dance in front of the screen. Typical visitor comments were: "it is interesting" and "I never tire of this". On the other

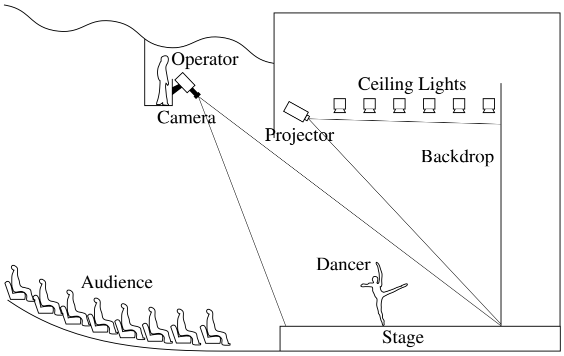

Figure 4: Installation for stage effect.

hand, one visitor wasn't particularly happy about the acoustic feedback and said: "I'm getting sick from hearing such a sound". This statement was made when the system was producing acoustic output according to the synthesized sound method only. The current version of the software was presented on campus event in March. On this occasion, the software remixed hip-hop music. The remixed music mode proved to be somewhat less effective in helping users to recognize musical changes in response to their interaction. On the other hand, this mode successfully attracted our visitors.

8. STAGE EFFECT FOR CONTEMPORARY DANCE

Identity SA was applied as a visual stage effect for a contemporary dance performance entitled "Vanishing Twin – unfinished works" choreographed by Jiří Kylián. The world premiere was held on February 7, 2008 at the Lucent Dance Theatre in the Hague and was played in several other theaters in the Netherlands until early March. Figure 4 depicts an illustration of the hardware setup that was used for this performance. It consisted of two 3 GHz oct-core MacPro computers, two DV-camcorders and two video projectors that were all mounted under the ceiling above the audience and the stage. The video images were projected frontally on the entire stage backdrops placed on the left and right side. The projection occasionally covered the dancers' bodies when they moved to the back of the stage. Each computer controlled 5,000 agents at a rate of 20 FPS. The agents were depicted as white Chinese characters on a black background. During the initial 24 minutes of the piece, the agents' movements was not affected by the dancers. After this phase, the agents became responsive to the dancers until the end of the piece. This effect was effective and it seems the audience appreciated the reactive and complex patterns that were generated by the agents and that resembled the movement of smoke or liquid. For this scenario, no sound output by Identity SA was employed.

9. CONCLUDING REMARKS

We are currently considering a variety of modifications that would extend the functionality of the software. For example it might be interesting to focus more on algorithmic composition as was the case in our previous work entitled "Flocking Orchestra". Another promising option involves the processing and modification of life audio that is captured via a microphone in order to create a distorted mirror effect also in the acoustic domain. Finally, it might be worthwhile to extend the flocking simulation and its visualization into the third dimension. The principal feasibility of the first two software modifications is out of question. But the third modification might be computationally to intensive for current generations of consumer-level computer hardware. In particular, 3D blurring and volume rotations are very costly to calculate. For these reasons, the implementation of these features might need to be postponed until suitably powerful hardware becomes available. To conclude, we hope that Identity SA creates an rewarding aesthetic experience that provokes visitors to reflect on their own identity and transitoriness. The software and sample scripts are downloadable from the project website http://www.intlab.soka.ac.jp/˜unemi/1/DT4/ .

10. ACKNOWLEDGEMENT

The authors would like to thank the students at Soka University and all of the visitors who supported and participated in the experimental exhibition on the campus. We highly appreciate the support of Prof. Dr. Rolf Pfeifer for this international collaborative project. Our most sincere thanks goes to the staff of the Netherlands Dance theatre, in particular Jiří Kylián, Dick Schuttel, Alexandra Scott who provided us with the exceptional opportunity to employ Identity SA as stage effect for a dance performance. We are very grateful to Keiko Taylor who initiated and mediated our collaboration in this dance project. This work is partially funded by Grants-in-aid for Scientific Research #17500152 from Ministry of Education, Culture, Sports, Science and Technology (MEXT) in Japan.

11. REFERENCES

[1] G. Beni. From swarm intelligence to swarm robotics. In E. Sahin and W. M. Spear, editors, *Swarm Robotics*, pages 1–9. Springer, 2005.

[2] D. Bisig and T. Unemi. Mediaflies – a video and audio remixing multi agent system. In *Proceedings of the nineth Generative Art Conference*, pages 63–74, Milan, Italy, 2006.

[3] D. Ikeda. *The Living Buddha – an Interpretive Biography*. Weatherhill, New York, 1976.

[4] C. Jacob, G. Hushlak, and J. Boyd. Swarmart: Interactive art from swarm intelligence. *Leonardo*, 40(3):248–255, 2007.

[5] C. W. Reynolds. Flocks, herds, and schools: A distributed behavioral model. *Computer Graphics*, 21(4):25–34, 1987.

[6] T. Unemi and D. Bisig. Playing music by conducting boid agents – a style of interaction in the life with a-life. In *Proceedings of A-Life IX*, pages 546–550, 2004.

[7] T. Unemi and D. Bisig. Music by interaction among two flocking species and human. In *Proceedings of the Third Iteration*, pages 171–179, Caulfield, Victoria, Australia, 2005.

[8] T. Unemi and D. Bisig. Flocking messengers. In *Proceedings of the nineth Generative Art Conference*, pages 272–280, Milan, Italy, 2006.

[9] T. Unemi and D. Bisig. Identity sa – an interactive swarm-based animation with a deformed reflection. In *Proceedings of the tenth Generative Art Conference*, pages 269–279, Milan, Italy, 2007.