Expanding the Expressive Pallette

K. Mason and M. S. T. Carpendale
Department of Computer Science,
The University of Calgary,
Calgary, Alberta, Canada

Abstract

Current methods in non-photorealistic graphics can place a heavy amount of focus on the algorithm, as opposed to the artist. In this paper, we analyse these trends, and present a conceptual framework for putting control back in the hands of the artist. Combining ideas from non-photorealistic graphics and artificial intelligence, we present new methods of supporting alternative artistic styles. Details of our implementation of this model are described, as well as methods for interaction. Finally, we create a simulation under this framework, and show preliminary results.

1 Introduction

Current techniques in non-photorealistic computer graphics are largely algorithm-focused. Generally they consist of either rendering techniques, which create an image from an artist-generated model, or post-production techniques, which take an image, and manipulate it to create some effect. While this provides support for a variety of styles, it takes a great deal of control out of the hands of the artist, and puts it into the hands of an algorithm designed by a computer scientist. Frequently the artist is left with only some indirect ability to tweak rendering or filtering parameters. The very effects that are intended to make the image more 'expressive', are produced by a deterministic system.

It is true that artists can and do use computers as tools and that algorithms are an important part of these tools. If the goal is to create tools that make it easier to produce images that follow a particular style, like impressionism[8] or cartoons[3], algorithms are becoming increasingly successful. If the goal is to generate illustrations quickly and automatically[7], then an algorithmic approach is appropriate. However, if the goal is to provide tools for artists, it must be remembered that the method should not involve the replacement of the artist with an algorithm. To accomplish this goal, it is necessary to take an approach that focuses on artists, instead of on algorithms.

In this paper we present an interactive paradigm that endeavours to place the creative decisions in the control of the artist. Section 2 describes the related work, and categorises it according to which parts of the process can involve artist interaction. Section 3 outlines our approach and Section 4 explains how interactive control is provided. Section 5 illustrates our initial results, and is followed by a brief discussion of future directions in Section 6.

2 Analysing Interactivity

Over the last few years, research in computer graphics has come to understand just how important it is to provide support for a broad range of artistic styles. Work has been done in trying to simulate traditional media, such as sketching (see [5, 26, 25, 28, 21]), watercolour (see [4, 8]) and sculpting [16, 17, 18]. Many alternate rendering styles have been developed (see [23, 12, 13, 15, 2, 11]) as well as alternate post production systems (see [19, 8]). A few techniques have looked at non-standard geometries (see [20, 14]) and there are some promising new techniques that involve the use of evolutionary algorithms as tools for artists (see [27, 10, 1]).

As a first step towards our goal of developing interactive graphics techniques that support and encourage an artist's personal expression and control of his or her own work, we consider the potential for interaction in existing techniques. For our purposes, we have simplified the activities in the computer graphics production pipeline into four principle processes which, for the purposes of this discussion, we define as follows:

- Set-Up: during the set-up phase the system is initialised,
- Modelling: during the modelling phase the artist creates the principle forms and geometry or alternatively, the artist creates the structures from which the forms and geometry will be mathematically derived.
- Rendering: the rendering phase of the pipeline involves the conversion of a three-dimensional model into a two-dimensional image.
- Post-production: during this phase a rendered or previously existing image (possibly a photograph) is adjusted to create different visual impressions.

The characteristic importance of each of these phases of the pipeline to a technique are depicted in the central line of images in Figure 2 to Figure 5. For explanation of the textures used see the legend in Figure 1. The top line in these images shows the area where artistic involvement is possible. These are the moments during the process where the artist is allowed some input, control or choice in the process. The bottom line gives an indication of those parts of the pipeline in which non-photorealistic effects are added. The left-hand end represents the start of the process and the right-hand represents the finish.

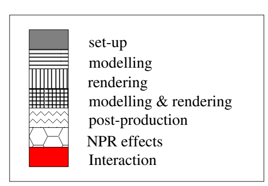

Figure 1: This legend applies to Figures 2 through 7.

Non-Photorealistic Rendering: There is an initial set-up period in which the artist does not take part, as described above. When this is complete the modelling phase is entered. Here the artist can interact with primitives of differing characteristics to create a virtual model, to modify an existing virtual model or to create a data-structure that will define the model. This phase is quite interactive and the artist exerts considerable control.

When the model meets with the artist's satisfaction, he or she can make a few choices as to the rendering algorithm, by setting various parameters. At present, there are a growing number of options, as more and more algorithms are developed. However, once these choices are made, the process becomes a 'black box'. The algorithm does its 'magic', and the artist waits to see if he or she will approve of the result. Figure 2 contains a depiction of this process. Examples of research in this category include [12, 2, 15].

If, on completion, the image does not prove satisfactory the artist has two choices. First, she or he can go back to the modelling phase and make adjustments they hope will improve the result. These adjustments could involve the model itself, or the rendering parameters. Secondly, the artist can revert to pixel by pixel touch-up.

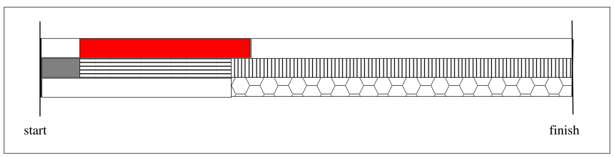

Figure 2: Non-Photorealistic Rendering: Interaction is supported through the modelling phase and at the onset of the rendering phase.

Non-Photorealistic Post Production The diagram describing our characterisation of this category is found in Figure 3. In this category, expressive effects are created by applying a process to an existing image, be it an image generated by another algorithm, a digital photograph, or any scanned-in image. The artistic choice in these methods is limited to the selection of a starting image, and the choice of parameters in the image filtering process. Examples of research that falls in this category include [8, 19, 22].

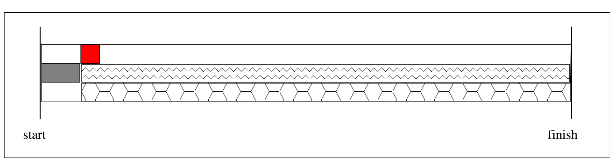

Figure 3: Non-Photorealistic Post Production: Interaction consists of adjusting parameters that the filters will use to create the final image.

A problem shared by the methods in both of the above cases is that the so-called 'expressive' effects are dealt with entirely within the algorithm. The end results may be thought of as being as much the personal expression of the algorithm's creator as of the artist who is using it. In fact, when looking at a piece of art created with one of these techniques, it is often easier to recognise the author of the algorithm, than to tell anything at all about the style or the expressive intent of the artist.

Algorithm Interaction The methods that fall into this category require comparatively more set-up than the previously described techniques. However, once the algorithm is established within an interaction paradigm, the artist can work directly with the algorithm (Figure 4). Research that can be characterised in this manner includes [24, 9].

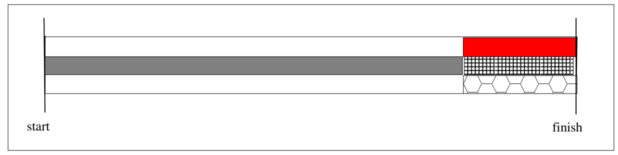

Figure 4: Algorithm Interaction: The set-up phase is vastly increased in importance, however the artist is able to interact directly with the algorithm.

The biggest advantage of these methods is what Snibbe and Levin refer to as *phenomenology* [24]. Simply put, this is the idea that the expression of the artist should be allowed to come through directly, without being shuffled through an algorithm that might, however subtly, change the meaning of that expression. Anytime an art piece must go through a phase where direct artistic control is not permitted, there is a risk of this. While the systems in this category endeavour to remove this risk, once the algorithm is chosen the interaction methods and visual responses are fixed. The definition of these visual responses is not within the scope of control of the artist who is using the system. Once again they are made during the design of the program.

Evolutionary Art A relatively new direction in expressive graphics uses ideas from evolutionary programming to generate new types of non-photorealistic graphics from one or more existing pieces. The evolutionary algorithm can contain the possibility of mutation and cross-over effects. These result in new and often surprising visual effects, which can be selected based on the taste of the artist, and can then in turn be further evolved. This approach holds much potential for developing new styles, as hybrids of existing styles are created. The artistic control comes during the choice of starting models, and in the evolutionary algorithm chosen (Figure 5). It is possible to use this process iteratively, making choices, awaiting results, and again making choices. However, the interaction is hit and miss, and the artistic expression is the result of patience and luck, at least with present techniques. For examples in the literature see [27, 10, 1].

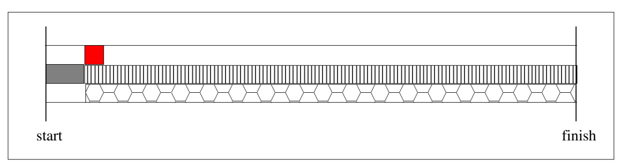

Figure 5: Evolutionary Art: These methods are applied iteratively. The artist has a moment of choice that affects outcome for the next loop. The rest is algorithmic.

What is Ideal Interaction?

The categorisation just presented illustrates the amount of control the artist has in these methods. At the same time, it also helps us to see where it may be possible to add control. We propose a conceptual model for an 'ideal' solution. While current hardware limitations make this model impossible in practice, a diagram of this 'ideal' model is represented in Figure 6 for purposes of discussion.

In this model, there is a small set-up phase, but for the remainder of process the interaction consists of the artist us-

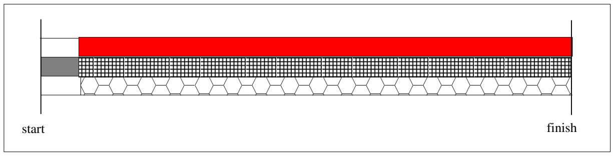

Figure 6: Ideal Interaction: The only time the artist cannot exert direct control is in the initialisation and set-up phase. Otherwise, the artist can always intercede as much or as little as desired.

ing the tools provided to create his or her work of art with immediate response, while still allowing algorithmic adjustments to be made. This immediacy would allow for the artist's expression to be direct, without fear of algorithmic interruption. This can be thought of as blending the realisation of non-photorealistic effects with the modelling process. Further, it would also allow algorithmic adjustment, but under the artist's direct control.

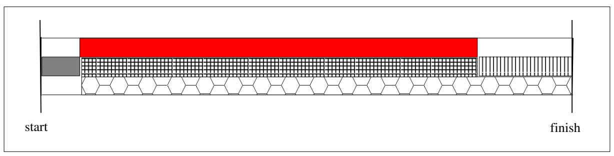

Figure 7: Our Interaction Model: In our conceptual model, the artist can interact through most of the process. More complex models can be abstracted to allow for real-time interaction, but will then require a non-interactive rendering phase.

Our Approach Figure 7 shows a representation of the model that is developed herein. In many ways it is a hybrid of non-photorealistic-rendering, as shown in Figure 2, and of algorithm interaction methods, as shown in Figure 4. By extracting the non-photorealistic effects from the rendering phase of the pipeline, and settling them squarely into the modelling phase, where an artist can interact directly with the system, it gains advantages from both models. It also comes much closer to the ideal form as shown in Figure 6.

3 Method

The conceptual solution to this problem is simple – one must find the right balance between direct artistic control and algorithmic assistance. Removing algorithmic support would render the process untenable – asking the artist to create their image on a pixel by pixel basis is equivalent to asking an artist to create an image by placing individual grains of sand. While there are some artists who do wish to work on this level, many prefer to use tools that assist (though not control) their work.

The practical issue of how this balance can be achieved is not as simple. The amount of control an artist might wish to retain is dependent upon on the individual artist. Some artists might wish to control nearly everything by hand. Others might prefer to instruct an algorithm, and have it do the majority of the work. In this latter case, the algorithm should still reflect the desires of the artist as much as possible, and the assumptions of the system architects as little as possible.

3.1 The Concept

Our system is a multi-agent environment, where each element in the image is represented by an agent. The artist instructs the agents on how to behave, and can then take as much, or as little control over them as he or she desires. At any point, the artist can interrupt the simulation, make adjustments, and then start it up again. The artist can intervene at every step, or make use of the algorithmic support provided. If the artist does choose to intervene, he or she would be able to do one of the following:

- **Manual Selection and Manipulation of Agents**: At the most basic level, the artist can take direct control of the agents. In the future, tools will be provided which allow for control over a number of entities at once. At present, however, it is only possible to select a single entity, and alter its placement in the scene. This allows an artist who wishes to work with their image on an element-by-element basis to get their virtual hands dirty.

- **Altering the State of Happiness**: The artist has control over how content the agents in the system are. Further details on this will be provided below. In essence, a content entity will stay where it is. A discontent entity will seek some way to make itself more content.

- **Alteration of Behavioural Rules**: If the artist finds that even with the above controls, things are not proceeding as desired, he or she can alter the rules that govern how the entities behave within the system.

It should be noted that this multi-agent simulation is really a complexity system. Much work has been done in other fields with similar systems. More information can be found at[6].

3.2 System Architecture

A broad view of the system architecture is modelled in Figure 8. It consists of an environment that can be populated with tribes and entities at the artist's discretion. These are defined as follows:

- **Environment:** The environment is the space in which the tribes and entities exist.

- **Tribes:** Tribes are groups of entities that possess like characteristics, and that share behaviours in common. Tribes are a convenient construct in that they allow the artist to control many entities in the same way. For example, suppose that an artist has created an entity to represent grass. He or she can then assign behavioural rules which are shared by all grass. For example, the grass agents may try to cluster together, or to find a surface to cling to.

- **Entities:** We use the term 'entity' to refer to an agent in the system. It is a construct created by the artist, be it a pixel, a brush stroke, a graftal, a surface, or any other object.

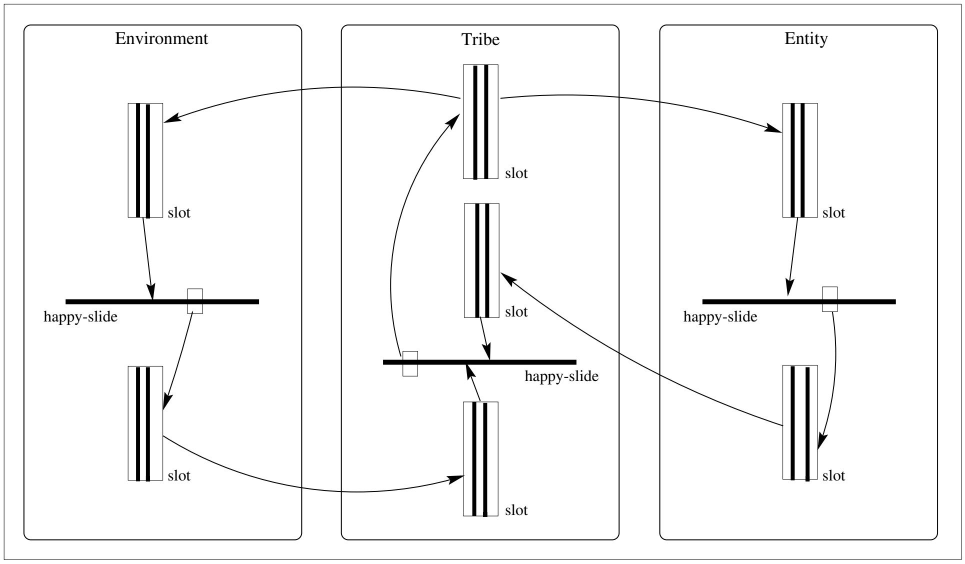

Figure 8: System Architecture, communication of happiness: The relationship between entities, tribes and the environment, with special focus on the connections by which changes in happiness are passed.

3.2.1 Organizational Hierarchy

The relationship between the environment, the tribes and the entities is hierarchical and one-to-many. The environment may contain many tribes. An individual tribe may contain many entities. Anything which has an effect on a higher level in the system will affect all of its subordinates. By contrast, each node in the hierarchy views itself as a semi-autonomous agent. When it is told to proceed from higher up in the hierarchy, it will attempt to make itself more happy. In the case of a tribe, this is accomplished by giving a selection of entities leave to try to make themselves happy. When an entity is instructed to make itself happier, it will look at the rule set that has been declared for it by the artist, and move to fulfill the conditions that as described in the rule set.

3.2.2 Happiness Distrubtion Algorithm

Happiness is a reflection of the entities' general satisfaction with the current state of the system. If an entity is happy, it will remain static. If it is unhappy, it will try to seek ways to improve its happiness. For the purposes of this implementation, happiness is measured on a scale of zero to one hundred, zero being extremely unhappy, and one hundred being perfectly content.

The overall happiness of the environment is represented as the sum of the happiness over all of its tribes. Similarly, the happiness of a tribe is represented as the sum of happiness across all of its entities. The happiness of an individual entity is determined by its ability to fulfill the conditions that have been specified for it by the artist. The closer it comes to fulfilling these conditions, the happier it will be. It should be noted that not every condition must necessarily accord the same happiness. Filling one condition may be defined to give twice as much happiness as filling another. The determination of which conditions are more or less important in terms of happiness is left in the hands of the artist.

Figure 8 shows the propagation chains that are followed for changes in happiness, both those that originate with the artist, or those that come about because an entity is able to fulfill a condition that is specified to increase its own happiness.

This latter case is the more simple, because propagating happiness levels upward (from entity to tribe, or from tribe to the general environment) is a matter of adding or subtracting the change from the higher level's happiness. Propagating a happiness change downward is more complex. A decision must be made as to how to distribute that happiness over the next level down. We have chosen a rule-set for this based on what seemed to be most useful and intuitive. We recognise that in doing this, we are making a decision that not all artists might agree with. In the future, it would be wise to make it possible for the artist to specify these rules to suit his or her own expressive style. The current rules are:

If the happiness has increased,
then propagate the change downward
in a uniform matter.

That is to say that if there are five constructs at the next level down, then each of them will gain one fifth of the indicated change in happiness.

If the happiness has decreased,
then propogate the change downward
porportionally to current happines.

In this manner each construct in the next level down takes an amount of change relative to its proportion of the current greater happiness. This prevents an entity which has zero happiness from moving into the negative realms.

4 Interaction

In order to illustrate how a person interacts with this system, we will use a simple model. This model contains only one tribe, which has been called "Fish". Each "Fish" is a simple graftal. It has a set of rules for expression, and a set of rules for increasing its own happiness. The expression rules are here kept extremely simple, so as not to interfere with an understanding of the happiness rules.

In general, members of the tribe "Fish" are happier when they are near the centre of projection, and they are happier when they are not alone. These rules do not necessarily model the behaviour of schooling fish. These are simply the properties that the artist has chosen to give value to for this simulation. That is the material point – it is not the algorithm that decides what makes a member of tribe "Fish" happy. It is the artist.

"Fish" Happiness Rules:

- Choose a random number between zero and ninety nine. This represents the threshold of happiness that is tolerable in the current iteration.
- If that threshold is greater than the current happiness of an entity, then the entity is discontent and will try to find a way to increase its happiness. With probability of 1 in 2, it will move closer to the centre of the projection, and increase its happiness in proportion to the distance moved. Then, with probability of 1 in 10, it will summon a new tribe member to a random spot on the grid.
- If the random number was less than the current happiness of an entity, then that entity is happy enough for the current iteration.

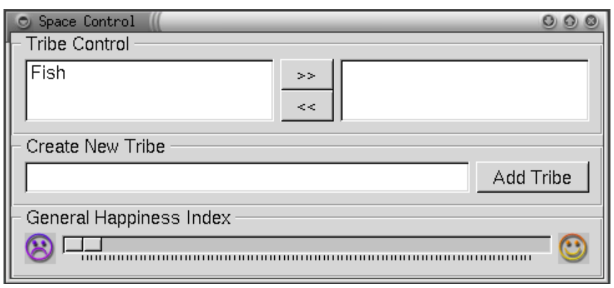

Figure 9: Environment Control: This control box allows the user to control the system at the global level. Global happiness can be added, and tribes can be added or removed from the environment.

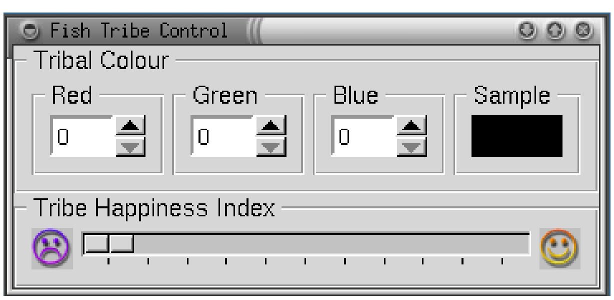

Figure 10: Tribe Control: The tribe control allows the artist to adjust the happiness settings for a tribe.

The controls for interacting with the simulation are shown in Figure 9 and Figure 10. In Figure 9, we see the general controls for the environment. Here the artist can add a new tribe, or adjust the global happiness. Figure 10 shows the tribal controls, which allow the artist to adjust the happiness level for one tribe only. As "Fish" is the only tribe in this simulation, the global happiness reduces to its happiness.

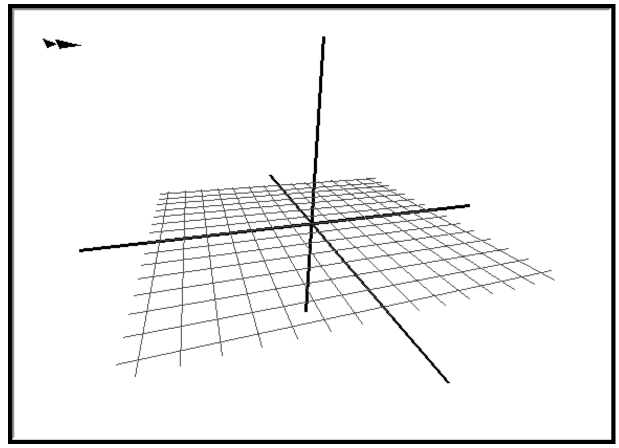

Figure 11: Initial Setup: At the beginning of the simulation, there is only one entity present in the environment.

The initial setup is shown in Figure 11. The artist has chosen to start with a single entity in the space. The initial happiness of that entity, its tribe, and the global space are all zero. For convenience of interpretation, the grid and axes have also been turned on.

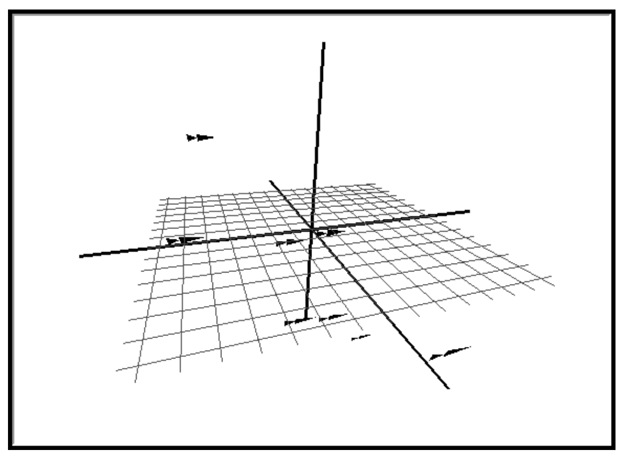

Figure 12: Seeking Happiness: The entity is given the opportunity to improve it's happiness, and seeks to do so, according to the rules specified by the artist. In this case, the entity summons other fish to school with, and moves toward an agreed meeting spot.

When the simulation is engaged, the entity follows the procedure described above to try to make itself happy. In Figure 12, the happiness level of the "Fish" tribe has reached about fifty percent. At each timestep, each of those entities tries to increase its own happiness level. As it succeeds, the happiness level increases and changes become less frequent. It becomes more and more likely that the threshold for acceptable happiness has been met in each iteration.

In Figure 13, the happiness level of the "Fish" tribe has reached one hundred percent, and the tribe becomes completely static. Any happiness threshold that is selected in

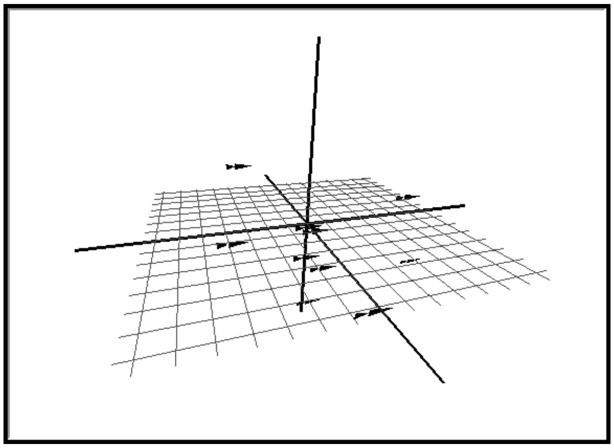

Figure 13: Happiness Achieved: At this point in the simulation, the happiness level of the fish reaches one hundred, and the fish become static.

the following iterations will be lower than the entities' happiness, so the entities will not again attempt to make themselves happier.

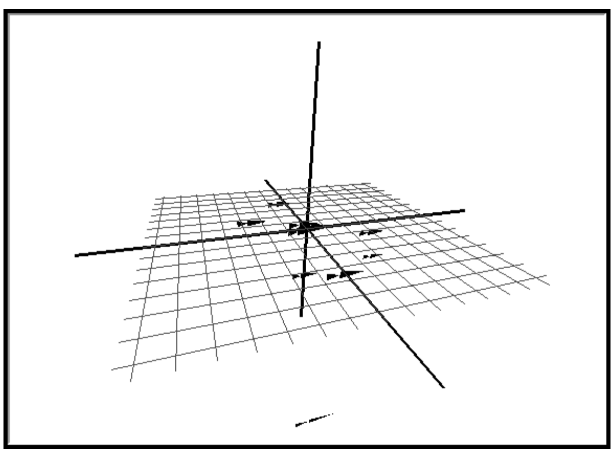

Figure 14: Happiness Re-Achieved: The artist, being unsatisfied with the result, as achieved in Figure 13, lowers the happiness for the entire fish tribe. The fish again seek to increase their happiness until they find a new balance.

Because the tribe has achieved algorithmic completion, however, does not mean that the artist is happy with the scene. In our case, the artist is not happy. There are not enough entities, and they are still too sparsely clustered. At this point, the artist can choose to change the happiness of the global space, the tribe, or of one or more selected entities. Our artist chooses to decrease the happiness of the tribe back down to fifty percent. The entities are discontent, and they again move to make themselves more happy. In Figure 14, they have again achieved one hundred percent happiness, and become static.

This time, the artist is satisfied, and turns off the simulation. The axes and grid are removed, and the fish are rotated into a better view. The final image is shown in Figure 15.

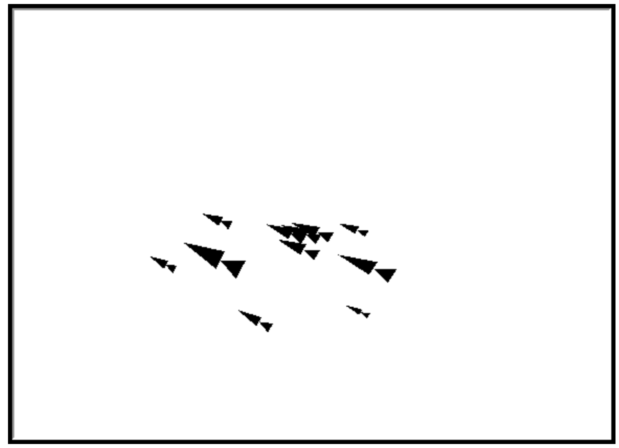

Figure 15: Final Spread: A view of the final cluster of fish, from a different angle. Note that if the artist was still not satisfied, he or she could simply lower the happiness of one or more of the fish to cause them to become active again.

5 Results

To show the potential of this way of looking at things, a more complex example has been constructed which involves multiple tribes, and interactions between the tribes. Following Hemert[27], we have tried to follow the style of some of the abstract work of Piet Mondriaan. In this model, the artist has created four tribes. A description of each follows.

Offsets

- **Expression Rules:** This tribe does not express itself visually in the image.
- **Happiness Rules:** If an entity in this tribe is given the chance to make itself happy, it will seek to find, and bond with a free member of tribe "Lines".

Widths

- **Expression Rules:** This tribe does not express itself visually in the image.
- **Happiness Rules:** If an entity in this tribe is given the chance to make itself happy, it will seek to find, and bond with a free member of tribe "Lines".

Lines

- **Expression Rules:** A line will express itself if and only if it has bonded with a width, and an with an offset. If it has done so, it will draw itself as a line of the claimed width. Its position is chosen by the application of its offset in the image space.
- **Happiness Rules:** If a line is unhappy, it will do one of the following: (i) Release a bond with member of tribe "width" or of tribe "offset" or (ii) Change its orientation from horizontal to vertical, or vice versa.

Fills

- **Expression Rules:** A fill expresses itself if it has found a bond with two unique horizontal and two unique vertical lines. If it has, it will express itself as a fill between these lines. By default, a fill will be white, unless it has chosen another colour in an attempt to make itself more happy.

- **Happiness Rules:** If a fill is unhappy it will do one of the following: (i) Change its colour, (ii) Try to bond with a horizontal or vertical line, or (iii) Surrender a previously claimed bond.

The first tribe controls the spacing between the lines, and the second controls the width of each line. The proportions of both are in approximate thirds. In both cases, the initial entities are carefully chosen by the artist such that there is only one entity with the largest proportion (one third). There are three times as many entities which hold the proportion one ninth, and three times as many as that which hold the proportion of one twenty-seventh. As with the rules and expression conditions above, all of this is the artist's choice. The third and forth tribes have a physical manifestation, which is described in detail above. Again, all the rules for expression and happiness are chosen by the artist. It is also possible specify initial values, though in this case, it has been chosen to have them selected randomly.

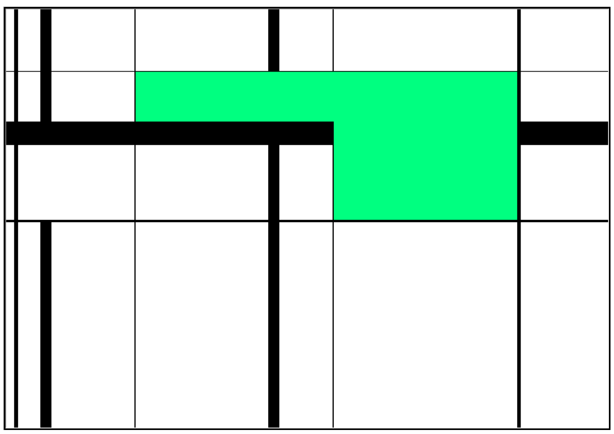

Figure 16: Simulation without Intervention: An example of what this algorithm might produce without artist intervention.

Figure 16 shows the result of a simulation in which the entities determine their own happiness without intervention. Without artist intervention, the system degenerates to algorithmic determinism, and the result may not be an expression of the artist, only the rules that the artist specified.

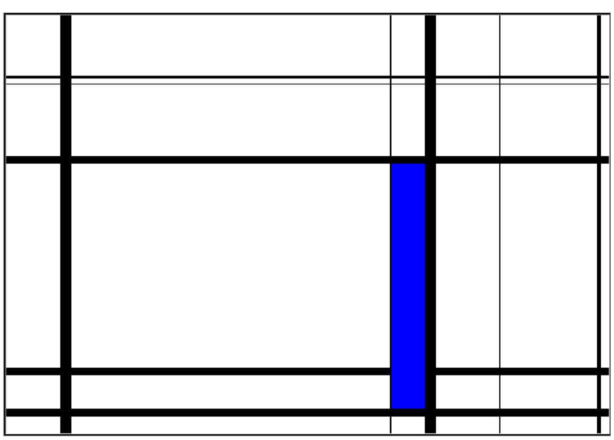

Figure 17: Mondriaan Example with Artist Intervention: This example was generated with minimal artist interaction.

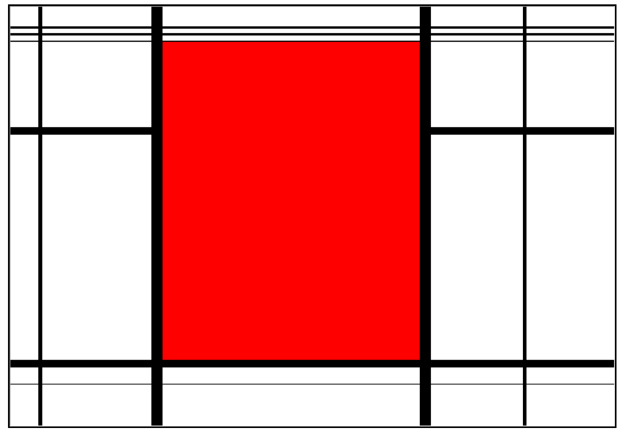

Figure 18: Mondriaan Example with Artist Intervention: This example was produced with a greater level of artist interaction.

Figure 18 and Figure 17 show examples of the system running with artist intervention. At any time, the artist may alter the state of happiness for the entire space, for a tribe, or for an individual entity. This will result in more change, but only in the aspects of the artwork that the artist is currently displeased with.

6 Future Work

We are currently considering a number of directions for this research. Tools will be developed to allow for the simultaneous manipulation of groups of entities. The use of evolutionary algorithms in entity creation could prove a valuable tool, as could the development a visual language to increase ease of interaction. More customisation will be allowed in the distribution of happiness. Expliting the inherent temporal coherence of the system described herein for the purposes of animation is under consideration.

As an example, consider the creation of tools for the manipulation of multiple entities. The artist could be given attractors and repulsors as tools. These could be inserted into the scene, and when switched on would have an effect on multiple entities simultaneously. Further, tools can be developed such that a manipulation of one agent (be it a rotation, translation or other change) is interpreted by a group of selected agents, or perhaps an entire tribe of agents, instead of by a single agent alone.

7 Conclusions

Artists are expert illusionists. They have been working for some millennia now to communicate their impressions and expressions. As computer scientists, we need to provide support for artistic control. This paper offers a perspective on the expansion of the expressive palette in computer graphics that places the focus on the artist, instead of on the algorithm.

By combining techniques from non-photorealistic rendering and artificial intelligence, this paper has proposed a novel interaction method that places the artist in the centre. This method has been explored in theory, and in practice, and methods for interacting with it have been described. Finally, preliminary resultwe show, illustrate the potential of this framework.

References

[1] Peter Bentley. *Evolutionary Design by Computers*. Morgan Kaufmann Publishers, Inc., 1999.

[2] David J. Bremer and John F. Hughes. Rapid approximate silhouette rendering of implicit surfaces. In *Implicit Surfaces Proceedings*, 1998.

[3] Wagner Toledo Correa, Robert J. Jensen, Craigh E. Thayer, and Adam Finkelstein. Texture mapping for cel animation. In *Proceedings of the 25th annual conference on Computer graphics*. ACM, 1998.

[4] Cassidy J. Curtis, Sean E. Anderson, Joshua E. Seims, Kurt. W. Fleisher, and David H. Salesin. Computer-generated watercolor. In *Proceedings of the 24th annual conference on Computer graphics*. SIGGRAPH, ACM, 1997.

[5] Debra Dooley and Michael F. Cohen. Automatic illustration of 3d geometric models: lines. In *Proceedings of the 1990 symposium on Interactive 3D graphics*. ACM, March 1990.

[6] J. Fall and A. Fall. Seles: A spatially explicit landscape event simulator. In *Proceedings of the GIS and Environmental Modeling Conference, Santa Fe, New Mexico, Jan, 1996*, 1996.

[7] Amy Gooch, Bruce Gooch, Peter Shirley, and Elaine Cohen. A non-photorealistic lighting model for automatic technical illustraion. In *Proceedings of the 25th annual conference on Computer graphics*. ACM, 1998.

[8] Aaron Herzman. Painterly rendering with curved brush strokes of multiple sizes. In *Proceedings of the 25th annual conference on Computer graphics*. ACM, 1998.

[9] Takeo Igarashi, Satoshi Matsuoka, and Hidehiko Tanaka. Teddy: A sketching interface for 3d freeform design. In *Proceedings of the 26th annual conference on Computer graphics*. SIGGRAPH, ACM, 1999.

[10] Cristian Jacob. *Illustrating Evolutionary Computation with Mathematica*. Morgan Kaufmann Publishers, 2001.

[11] Matthew Kaplan, Bruce Gooch, and Elaine Cohen. Interactive artistic rendering. In *Non-PhotoRealistic Rendering and Animation 2000*. ACM, 2000.

[12] Michael A. Kowalski, Lee Markosian, J.D. Northrup, Ludomir Bourdev, Ronen Barzel, Loring S. Holden, and John F. Hughes. Art-based rendering of fur, grass, and trees. In *Proceedings of the 26th annual conference on Computer graphics*. SIGGRAPH, ACM, 1999.

[13] Lee Markosian. *Art-Based Modeling and Rendering for Computer Graphics*. PhD thesis, Brown University, May 2000.

[14] D. Martin, S. Garcia, and J.C. Torres. Observer dependent deformations in illustration. In *Non-PhotoRealistic Rendering and Animation 2000*. ACM, 2000.

[15] Barbara J. Meier. Painterly rendering for animation. In *Proceedings of the 23rd annual conference on Computer graphics*. ACM, 1996.

[16] Shinji Mizuno, Minoru Okada, and Jun ichiro Toriwaki. Virtual sculpting and virtual woodcut printing. *The Visual Computer*, 14, 1998.

[17] Robert Noble. *Intuitive Sculpting of Flexible Objects for Coherent Animation*. PhD thesis, Robert Gordon University, 1998.

[18] Robert Noble and Gordon J. Clapworthy. Improving interactivity within a virtual sculpting environment. In *Information Visualization '98*. Robert Gordon University, IEEE Press, 1998.

[19] Victor Ostromoukhov and Roger D. Hersch. Multi-color and artistic dithering. In *Proceedings of the 26th annual conference on Computer graphics*. ACM, 1999.

[20] Paul Rademacher. View-dependent geometry. In *Proceedings of the 26th annual conference on Computer graphics*. ACM, 1999.

[21] Ramesh Raskar and Michael Cohen. Image precision silhouette edges. In *Symposium on Interactive 3D Graphics*, April 1999.

[22] Michio Shiraishi and Yasushi Yamaguchi. An algorithm for automatic painterly rendering based on local source image approximation. In *Non-PhotoRealistic Rendering and Animation 2000*. NPAR, ACM, 2000.

[23] Alvy Ray Smith. Plants, fractals, and formal languages. In *Proceedings of the 11th annual conference on Computer graphics*. ACM, 1984.

[24] Scott Sona Snibbe and Golan Levin. Interactive dynamic abstraction. In *Non-PhotoRealistic Rendering and Animation 2000*. ACM, 2000.

[25] Mario Sousa and John W. Buchanan. Computer-generated graphite pencil rendering of 3d polygonal models. In *Computer Graphics Forum - Proceedings of Eurographics '99*, September 1999.

[26] Mario Sousa and John W. Buchanan. Observational model of blenders and erasers in computer-generated pencil rendering. In *Graphics Interface*, 1999.

[27] J.I. van Hemert and A.E. Eiben. Mondriaan art by evolution. In *Proceedings on the Eleventh Belgium-Netherlands Conference on Artificial Intelligence*, 1999.

[28] Lance Williams. Ink jets, level sets and silhouettes. In *Workshop on Image-Based Modeling and Rendering*, March 1998.