An Interactive Artificial Ant Approach to Non-photorealistic Rendering

Yann Semet[1], Una-May O'Reilly[2], and Frédo Durand[2]

[1] Optimization and Machine Learning Group, INRIA Futurs, Orsay, France
yann.semet@tremplin-utc.net,
[2] CSAIL, Massachusetts Institute of Technology
unamay@csail.mit.edu, fredo@mit.edu
http://graphics.csail.mit.edu/~semet/antsNPR

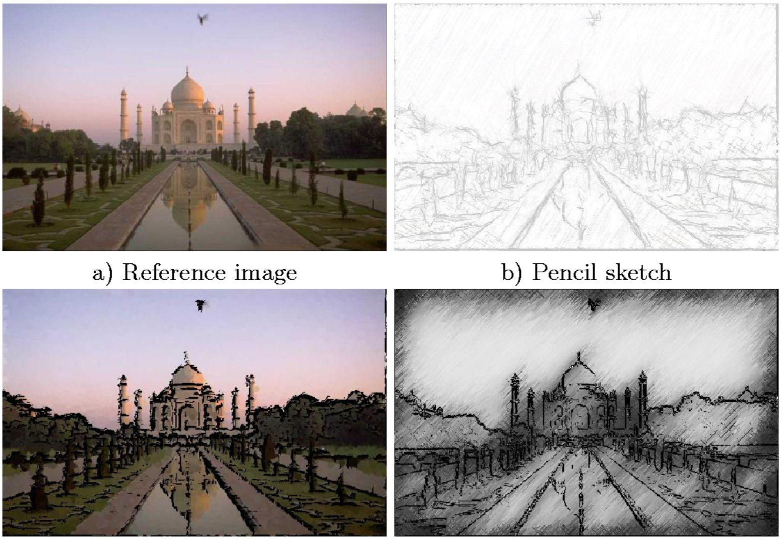
Abstract. We couple artificial ant and computer graphics techniques to create an approach to Non-Photorealistic Rendering (NPR). A user interactively takes turns with an artificial ant colony to transform a photograph into a stylized picture. In turn with the user specifying its control parameters, members of a colony of artificial ants scan the source image locally, following strong edges or wandering around flat zones, and draw marks on the canvas depending on their discoveries. Among a variety of obtained effects, two are painterly rendering and pencil sketching.

1 Introduction: Non-photorealistic Rendering and Ant Colonies

The goal of computer graphics has traditionally been to simulate the physics of light in order to produce photorealistic images. In contrast, the field of Non-Photorealistic Rendering (NPR), in recognition that realism is not always effective or superior, has the contradictory goal to develop traditional and novel pictorial rendering styles [1,2,3].

Our broad goal is to mediate creativity and artistry through decentralized activity and cooperation in a multiple-localized-agents metaphor. In this contribution we demonstrate how we have addressed one problem in the realm of NPR: creative transformation of a digital photograph into a natural and stylized picture. We have designed an artificial ant NPR system that is based on ants that navigate and sense the environment of a reference image. Ants deposit ink marks on an output picture according to where they are, what they sense and their short term memory which gives each its turn history for a few prior steps. The user interacts at a colony level to choose navigation and mark parameters. Then, the colony lives out its life on the image. These steps are repeated until the user is satisfied with the output picture. The approach launches the user and ant colony into an emergent, non-linear design process. The user interacts through the metaphor of an ant colony and, with acquired experience, it is possible to elicit a general artistic style. The system is able to provoke the user with specific and not entirely predictable applications of the directed style.

K. Deb et al. (Eds.): GECCO 2004, LNCS 3102, pp. 188–200, 2004.



a) Reference image b) Pencil sketch

c) Painterly d) pencil and smudge

Fig. 1. We have designed a distributed agent system where ‘ants’ navigate and sense the environment of a reference image. Ants deposit ink marks on an output picture according to where they are, what they sense and their short term memory which gives them turn history for a few prior steps. The user interacts at a colony level to choose navigation and mark parameters. Then, the colony lives on the image. These steps are repeated until the user is satisfied with the output picture.

Our system is most precisely a distributed agent approach. Calling our agents ‘ants’ provides a convenient metaphor for its users to conceptualize agent navigation, image processing (as sensing) and ink deposition (as trail marking). Our system is capable of a range of styles. We have named three styles ‘pencil sketching’, ‘painterly rendering’ and ’sugar sculpture’ . In Figure 1a) we present a reference image which is a photograph and not pictorially rendered. This is followed by example pictures, Figure 1b) through Figure 1d), that the system rendered during different user-interaction sequences while using the reference image.

We proceed as follows: in Section 2 we motivate our use of distributed agents (i.e. artificial ants). In Section 3 we describe how our system is designed. In Section 4 we describe how two pictorial styles can be generated and show a variety of pictorial outcomes. In Section 5 we try to ’put a finger’ on what makes our system ‘work’ and compare it to related approaches. We conclude with our future vision for the system.

2 Our Rationale: Why Use Artificial Ant Colonies?

We considered three factors in our decision to address NPR using artificial ant colonies.

1. **Pictorial outcome:** One thread of NPR research involves transforming a digital photograph into a creative picture. With stroke approaches, randomization is used to achieve the inconsistent, uneven nature that strokes of a human artist exhibit [4,5,6]. Many systems draw strokes in random order or slightly randomize stroke length, the 'brush' texture or size. In addition, in general, ([7]'s importance map and individual stroke control such as [4] are exceptions) the output image is globally processed in a uniform manner each time a stroking style is selected and applied.
   With a distributed agent approach, the potential for non-global and semi-local treatment of the input image is possible. Ants are placed randomly and each ant's navigation is directed by the source image's varying local information. An ant's navigation history and memory capacity effects its ink deposition and navigation. Thus, conditions in different parts of the canvas have the opportunity to be treated individually. In addition, iterating colonies of ants with different behaviors for each colony allows another level on non-linearity and non-uniformity that one shot or multi-layers homogenous approaches can not achieve.
2. **Design initiative:** Our goal is for creative contributions to be made by both the user and the computational software. Ideally they are a design *team* with complementary roles and shared influence over the outcome. The ant approach has the user specify a process or behavior that will be executed, but not directly the outcome. This means of composing removes the onus on the user to be in complete control. Instead, the system is given more initiative in the design activity.
3. **User interaction:** Can a radical metaphor for mediating user interaction empower creativity? It is extremely eccentric for a user to work with an ant colony metaphor when attempting to create a pictorial artifact. However, perhaps this interface is so different it may inspire new creative process? There is an adage that creativity arises from constraints and in some respects the ant colony control metaphor is constraining. Additionally, creativity can also arise when existing approaches are exchanges for new resources and methods.

3 System Description

Our technique takes as input a digital photograph that is transformed into a non-photorealistic picture in the following way:

Step 1. The digital source image is preprocessed to create environmental maps.
Step 2. The user sets up parameters of a colony of artificial ants

Step 3. The ant colony navigates the source image by sensing it and the environmental maps. Some elements of navigation behavior are controlled by colony parameters. Each ant is able to deposit ink marks on an output image. Its decision to do so depends on what it senses and a short term memory of its turn angles. Ink marks emergently form the strokes of the picture. The colony is finished when each ant has exhausted its limit of navigation steps.

Step 4. Goto Step 2 unless user is satisfied with current output picture.

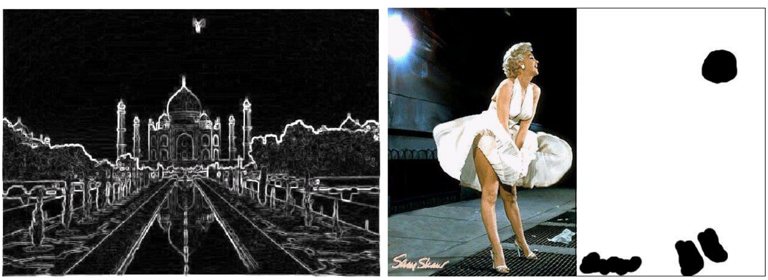
3.1 Preprocessing

Ants sense the digital input image by referencing its pixel values. In addition, they can sense the following pre-computed environmental maps:

Luminance Map. Assuming colour images, a *luminance* value is computed for every pixel through a weighted average of the Red (r), Green (g) and Blue(b) channels that reflects the global visual impact of the pixel as suggested in [8]: $L = 0.299r + 0.587g + 0.114b$.

Gradient Maps. The norm and orientation of the gradient are computed on the luminance map at every pixel using 3x3 Sobel filters.[9] A gradient map is shown in Figure 2.

Importance Map. A user can specify regions of the image that she deems particularly significant and that, as such, need particular attention. An example of an importance map is shown in Figure 2. Ants typically use this information to deposit finer brush strokes around important regions such as the eye of a portrait.



a) b)

Fig. 2. a) A gradient map of the Taj Mahal image of Figure 1a). b) Close-up of a photograph containing a face and the corresponding importance map (important areas are in black). Marilyn's face and shoes as well as the artist's signature are preserved.

3.2 Setting Up a Colony

A colony is a collection of ants (typically around 500). Parameter values set for the colony apply to every ant (except the one that specifes colony size).

Table 1 shows all colony parameters. All but the last five listed are influential in navigation while the final five dictate how the ant draws its mark. In the course of describing ant navigation algorithms, the navigation parameters will individually be described. The five ink marking parameters set the mark's shape, length, angle, thickness and means of being colored.

The parameters of the colony are accessible and modifiable through our Graphical User Interface (Fig. 3). A keystroke quickly runs a colony with a given set of parameters. The elementary navigation behaviors of edge drawing, filling and hatching, and smudging have been combined into composite behaviors which are also executed with a keystroke. The process of getting to a final result is strongly time dependent so ready access to composite behaviors is an interface feature.

Table 1. User specified ant colony parameters.

Parameter Name and/or symbol	Type	Typical value
colony size	integer	$1 \ldots 1000$
marking behavior: a.k.a. navigation algorithm	edge drawing filling and hatching smudging	
step-size	pixels	$1 \ldots 200$
gradient start threshold (T_0)	[0.0...256.0]	80.0
gradient continues threshold (T_1)	[0.0...256.0]	20.0
gap threshold (T_{gap})	[0.0...256.0]	
crosshatch threshold (T_{hatch})	[0.0...256.0]	
memory size	integer	3
jump-radius	pixels	5
max-steps	integer	200
max-jumps	integer	5
sharp turn threshold	degrees	45
mark shape	circle, line, cross	
mark length	pixels	$1 \ldots 200$
mark angle (α)	degrees	
mark thickness	pixels	$1 \ldots 20$
color: underlying pixel in source image is used	copy RGB grey scale from luminance sketch effect	

3.3 Marking Behavior Is Directed by Ant Navigation Algorithms

An Ant Relies Solely on Local Information. Each ant is an agent situated on the source picture. It senses certain local, low level features of this picture (i.e. its environment) and acts solely upon this information. An ant is not subject to any centralized control that would direct it to coordinate with others to fulfill image-wide, high level intentions of the user. No global or high level features of the picture, such as strokes or shadows, are apparent to an ant. Each ant has an encapsulated individual state description in terms of current position, velocity and short term memory, see Table 2. Position is the x and y coordinates of the pixel the ant is on. Because an ant moves, it is characterized at each time step by a velocity vector that has two elements: angular orientation (normalized at the ant's level and expressed with x and y components), and step-size (i.e. magnitude) which does not vary over time and is set as a colony parameter. To avoid overly sharp turns, an ant stores its successive turning angles in memory and only makes a mark when a memory check confirms this will not occur. An ant also records the number of steps it has taken since its birth. This limits the maximum length of the strokes.

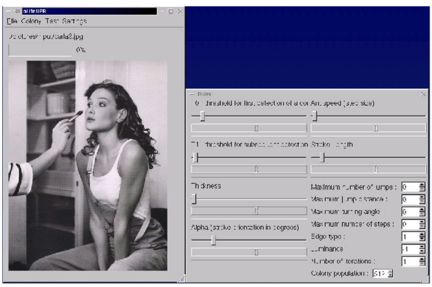


Fig. 3. Graphical User Interface

Table 2. An ant's local state information

Variable	Type	Representation
position (p_t)	(x, y)	floating point
orientation (θ_t)	(x, y)	floating point
velocity (v_t)	θ_t stepsize	 integer
$memory[t]$	vector	floating points
steps counter	scalar	integer
jumps counter	scalar	integer

Values are time dependent except for counters. Stepsize is a global parameter but is shown here because it is an element of velocity. Using floating point representation for position and orientation is necessary to achieve smooth strokes.

The computational implementation of some of this local, non-feature-based information is crucial to our system's visual quality. In particular, storing position and the orientation value of velocity in floating point representation and rounding them to reference a pixel in the image yields smoother strokes.

Each ant is initially placed at a random position in the image. The system simulates one ant at the time, allowing it to move a maximum number of steps. Each ant, using its local information and environmental maps, executes a navigation algorithm for one of the three marking behaviors:

Edging Navigation. This is guided by the gradient of the original image. At a given time and position, ants base their trajectory calculation on the local gradient's norm and orientation. This is very similar to line integral convolution (LIC) [10] though LIC is a filtering approach with more rigourous normalization that is globally and uniformly applied across an image for the purposes of visualizing vector information. Both [5] and [6] also exploit gradient information (though for different purposes), the former to clip brush strokes and the latter for the placement of control points for anti-aliased cubic B-splines that model strokes.

Let us denote, G_{norm} to be the gradient map in terms of norm. Ants sense G_{norm} to follow edges, i.e. trails of high gradient points. The algorithm proceeds as follows for an ant originally situated on a pixel $p_0(x_0, y_0)$:

Edging Navigation:

1. If $G_{norm}(p) > T_0$, continue, else STOP
2. If jumps-counter < max-jumps :
   a) While *continuing conditions* hold do:
      i. Put a mark on the canvas.
      ii. Move from $p_0(x_0, y_0)$ to $p_1(x_1, y_1)$ with $p_1 = p_0 + \boldsymbol{v}_t$
      iii. Increment steps counter.
      iv. Update short term angular memory ($memory[t] = \theta_t - \theta_{t-1}$).
   b) If gradient track vanishes, jump to a local maximum, increment jumps-counter.

Continuing conditions: The ant keeps on following the edge only if

1. It has not exceeded the maximum allowed number of steps.
2. It has not exceeded the sharp turn threshold as calculated from $memory[t]$.
3. It is still within the boundaries of the image.
4. The gradient still exists: G_{norm} at the current pixel $p_0(x_0, y_0)$ is over T_1.

Local jumps: When a gradient track vanishes, an ant scans the surrounding pixels within a given jump radius (a colony parameter) to look for other high gradient points greater than T_1. The ant then moves to the highest of them. To avoid sharp turns and backturns, the exploration of the neighbourhood is restricted to a portion of circle situated in front of the ant.

Filling and Hatching Navigation. This behavior macroscopically maps color information from areas of the source image to the output picture. It is macroscopic in two ways: it relies on multiple pixel information and it draws marks that are larger than a single pixel. If the ant starts at a pixel with a low gradient (i.e. $G_{norm} < T_0$), it copies the color of the source image pixel and uses it to draw a mark. This mark extends beyond the current pixel and is either a cross or a line (both angled using value α) or small circle. The ant then navigates ahead and keeps on depositing marks on the output canvas as long as the underlying pixel has a color that is sufficiently close to that of the pixel the ant started from. This comparison uses colony parameter, T_{gap}. In this way, the ant generates a group of identical marks that, referring to the source image, generalize one pixel to a group of similar, neighboring pixels.

The computation of this strategy is simpler (i.e. geometrically linear) than the edging algorithm. It requires extending the ant's state information to include local variables that record the 'gap' between the first pixel's and current pixel's color and the first pixel's color and position.

The details of drawing the mark depend on whether it is a circle, line or cross. A circle can be immediately drawn in a radius around the current pixel. A line is drawn in direction α then $-\alpha$. Hatches are drawn when luminance is less that T_{hatch}. Navigation is repeated from the first pixel while drawing a second line normal to the first.

Filll and Hatch algorithm:

1. Retrieve the underlying colour (C_0) or luminance (L_0) at P_0.
2. While *continuing conditions* hold do:
   a) Draw a mark. For a line, mark it in direction α.
   b) Move ahead
   c) Compute the gap: $gap = L_{local} - L_0$ or $gap = C_{local} - C_0$.
   d) Retrieve the local gradient norm $G_{norm}(p)$.
   e) Increment the steps counter.
3. Return directly to P_0.
4. Repeat Step 2 with $\alpha = -\alpha$.
5. If $L_0 < T_{hatch}$ do:
   a) Return to P_0.
   b) Repeat Steps 2, 3 with $\alpha = \alpha + \frac{\pi}{2}$

Continuing conditions:

The ant keeps on following the edge only if the three following conditions hold:

1. $gap < T_{gap}$: This ensures that an ant marks only where underlying pixels in the source image have sufficient color similarity.
2. $G_{norm}(p) < T_0$: Filling stops when the ant meets a high gradient point.
3. The maximum allowed number of steps has not been exceeded.

Smudging. For traditional media, smudging consists in rubbing one's finger or a small piece of paper over a portion of the picture to blur it or make it more uniform. To simulate this, an ant's smudge navigation behavior copies that of edging except that deposition of the ink mark is different. The ant uses the source image to compute the average luminance or colour within a square area of given radius around the pixel it is on and places its mark (circle, square, line or cross) using that value. A variation of smudging that led to an effect we call 'sugar' (see Section 4.3) actually introduced another dimension of non-linearity. An ant averages values on the current version of the output picture (not the source image).

4 Stylistic Results

By selecting a colony's parameters' values interactively each time before its ants navigate, a user is able to direct the style of the resulting picture. The picture itself is a time series of directly overlaying layers each of which can overwrite some or all pixels of the previous. From our personal experience with the system, an effective user design plan to explore outcomes in the vein of traditional styles is to progressively refine the picture detail each turn. In general, step-size controls the 'sketchiness' of emergent strokes. Coarseness is achieved by setting a high velocity for navigation. The step-size component of velocity (a colony parameter) controls the distance (in pixels) an ant travels between sensing steps. When gradient information farther apart is connected this effectively leads to strokes that are only roughly approximate to the photograph's edges and that are composed of long linear subsegments. Small step-size values will follow the photograph's edges more closely and have shorter linear subsegments.

Using knowledge of a technique of painting that conveys coarseness via long, wide, light brush strokes and detail, conversely, with short, fine, heavy brush strokes, velocity (i.e. step-size) can be complementarily teamed with appropriate mark parameter choices. Choosing the step size and mark parameters with this relationship will yield a style derived from conventional painting techniques. The user is also, however, free to invert traditional techniques or invent her own and explore the consequences of giving the colony this nature of direction. No two interaction sessions ever are the same due to both the random factor in ant placement and the user's inclination to explore something new, even if minor, each session.

Despite the developing nature of the system, it has already been able to produce some interesting results. Two traditional and easily identifiable effects that can be achieved are painterly rendering and pencil sketching. In addition, we observed a variety of intriguing, surprising and promising effects that might eventually form an element of a style (albeit not necessarily a rigorous pictorial one), one of which we call 'sugar sculpture'.

4.1 A Painterly Rendering Style

We obtained painterly effects by following Haeberli's [11] and Hertzmann's strategy [12] of producing a picture with multiple brush strokes of different sizes. See Figure 4 for an illustration of the process. Initially, each colony's navigation behavior is filling or hatching. The step-size component of each colony's velocity stays constant (it equals one in Figure 4 to provide maximum detail). By directing each ant of early colonies to deposit ink marks that are long and thick, (typically starting at 10 pixels wide and 15 pixels long), the canvas is first densely covered with long and thick brush strokes. The rendering is refined by gradually reducing the dimensions of the ink marks (down to 2 pixels wide and 4 pixels long in Figure 4) over successive colonies. Some emergent strokes may overwrite and overlap with earlier ones in a non-linear way as this reduction takes place. After filling and hatching, the smudging navigation behavior is used. The ants deposit tiny cross marks of average color and this results in fuzzy looking boundaries that roughly simulate the mixing of ink pigments.

4.2 Pencil Sketching

The key idea behind this effect relates to how an artist pencil sketches a portrait. The artist first draws the key edges very roughly using very long, linear strokes but with a very slight pressure. He then gradually refines his contours, making

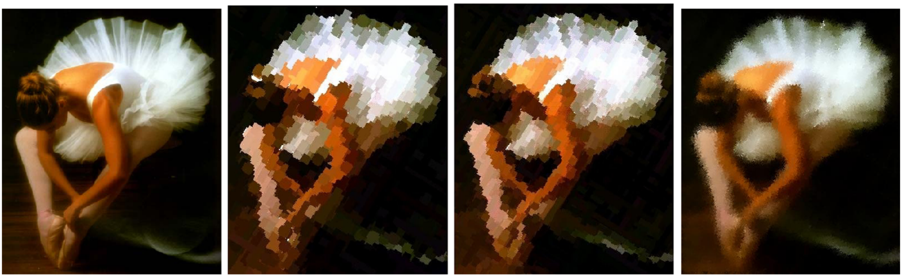


Fig. 4. An interaction sequence in the *painterly rendering* style. The source image is at left. The middle two pictures are intermediate results between which multiple brush dimensions were successively decreased 5 times over 35 keystroke (i.e. colony) sequences of filling and hatching navigation. Afterwards, 10 colonies navigated while smudging with a 2X2 radius brush. The final result of the interaction is far right. The step-size was one and the colony size 200 throughout the session.

them both smoother, stronger and shorter. To achieve this technique in term of pencil pressure, the mark's luminance was set to be inversely proportional to the mark's length during edging navigation (with a minor luminance correction). Then, by using a series of decreasing step sizes that decrease the successive colonies' velocity, emergent strokes will more accurately follow the image's edges and be both darker and shorter. This results in a style like gradual sketching. This particular effect is illustrated in Figure 5 and Figure 6 where the images' contrast has been enhanced for display.

4.3 Sugar Sculpture Effect

In open exploration of the system's creative range, we obtained an interesting effect which we call the 'sugar sculpture': smudging is usually performed by averaging neighbouring values on the source image ; by averaging values on the target canvas instead and by replacing cross marks by circle marks, we obtained the hard to describe "sugar sculpture" look and feel as seen in Figure 6 which uses a digital photograph of the Eiffel Tower as its source image.

5 How Does It Manage to Work?

What is the source of our system's capabilities? Additionally, how does it compare to other NPR systems? Producing a stroke-based image is a major source of the system's compelling nature. Strokes are associated with traditional artistic style and media. Digital brush strokes convey media in addition to scene aspects. They contribute to a result's interpretation as an artistic rather than realistic rendering.

Using strokes to form the basis of a digital picture is widely practiced in computer graphics, for example, [13,14,15,16,17]. The basic concept of generating

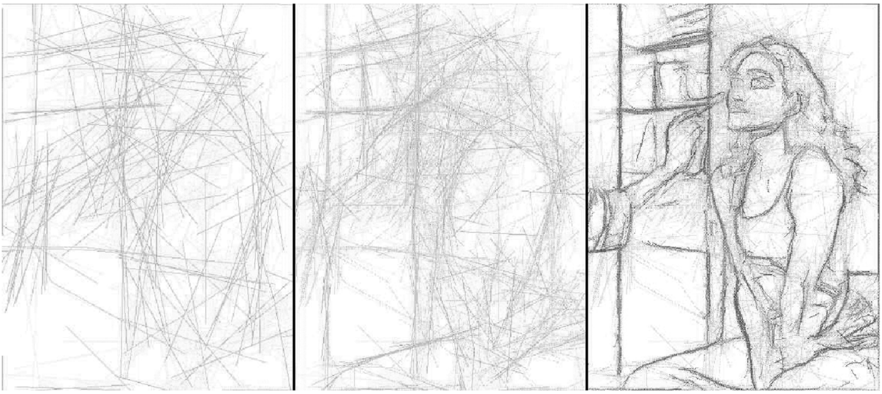


Fig. 5. A progression of gradual pencil sketching. Step-size is decreased in 7 non-uniform steps from 200 pixels to 7 pixels during the course of 35 colonies of 512 ants performing edge navigation. In between the final and previous to final sketch, the brush length is decreased from 100 to 7 in 5 steps over 20 colonies.

strokes is an approach we share with many graphics systems. In contrast to these examples, however, in our system a stroke is a one dimensional trajectory of ink marks determined by an ant's navigation behavior, its mark deposition behavior and its reference to memory of previous turn angles. This formulation creates strokes that are different in nature from the graphics systems. The use of gradient information creates strokes that are responsive to edge and varying aspects of content in the original image. Subtle, meaningful properties of the original image can be sensed and influence the outcome. This contrasts with the graphics systems where a stroke technique is uniformly applied and the only variation results from injected randomness. In addition, a stroke in our system

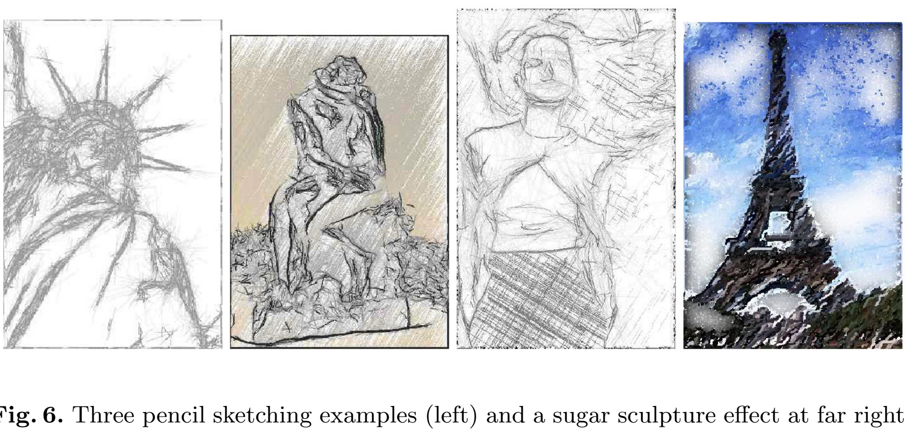


Fig. 6. Three pencil sketching examples (left) and a sugar sculpture effect at far right.

is not based on strictly local information, i.e. the information of a pixel and its immediate neighbors. It is created on *semi*-local information that is essentially gradient trails.

Another difference, this one obvious, is entirely external, i.e. in abstraction at the user's level: in the graphics systems, the user renders a picture by creating strokes. Stroke creation is an explicit task that user and system work on. By contrast, the user interface of our system does not provide access to a stroke. Ants give the user a unique control for influencing the rendering process. The user accesses ant-based navigation behavior, marking behavior and memory, instead. This radically different interface needs to be evaluated for its expressive capacity. Our limited, informal experience has shown it not to be taxing. However, it is more conducive to an explorative attitude rather than a functional one. It may better serve creative purposes rather than deliberate ones. The distributed and not wholly predictable nature of our ants balances creative exploration with purposeful intent.

The overall process of interaction between the user and the system during a 'design' session has novel consequences: the multiple, overlying layers result in unanticipated stroke interaction. One pixel may be overwritten multiple times - both by multiple ants in the same colony or by different colonies. In contrast, when a user controls strokes, it is less likely that many non-linear interactions can be simply explored. Or, in a system such as [6] which is automatic with access to only high level parameters, despite an iterative element to the algorithm's processing of increasing detail, an iteration is quite strictly defined: the user controls only brush size.

Our pencil sketch style results in drawings that are more pleasing than traditional computer vision edge detection. This is probably due to two factors. First, we successively use ants with different sets of parameters to build the drawing from coarse to fine, which results in a nice multi-scale creation of strokes that emphasize both the high-level appearance of the picture and the fine details. Second, the creation of strokes as trajectories and the use of a memory for the ants results in longer strokes that are also better shaped.

6 Conclusion and Future Work

This work demonstrates that an artificial ant system that capitalizes on NPR techniques from computer graphics can be teamed up with a user to accomplish visual representation in a novel way. We intend to incorporate interactive evolutionary computation to search for innovative navigation behaviors and stigmergic communication among ants plus conduct formal user evaluation studies.

Acknowledgements. We thank Martin C. Martin, Pierre Collet, Xavier Décoret and Stéphane Grabli.

References

1. Gooch, Gooch: Non-Photorealistic Rendering. AK-Peters (2001)
2. Strothotte, T., Schlechtweg, S.: Non-Photorealistic Computer Graphics. Modeling, Rendering, and Animation. Morgan Kaufmann, San Francisco (2002)
3. Durand, F.: An invitation to discuss computer depiction. In: Proc. of the ACM/Eurographics Symposium on Non-Photorealistic Animation and Rendering (NPAR). (2002)
4. Haeberli, P.: Paint By Numbers: Abstract Image Representation. In: SIGGRAPH 90 Conference Proceedings. (1990)
5. Litwinowicz, P.: Processing images and video for an impressionist effect. Proceedings of SIGGRAPH 97 (August 1997) 407–414
6. Hertzmann, A.: Painterly rendering with curved brush strokes of multiple sizes. Proceedings of SIGGRAPH 98 (July 1998) 453–460
7. Durand, F., Ostromoukhov, V., Miller, M., Duranleau, F., Dorsey, J.: Decoupling strokes and high-level attributes for interactive traditional drawing. Eurographics Workshop on Rendering (2001)
8. Foley, J., Van Dam, A., Feiner, S., Hughes, J.: Computer Graphics : Principles and Practice. 2nd edn. Addison Wesley (1997)
9. Jain, R., Kasturi, R., Schunck., B.: Machine Vision. New York, NY, McGraw-Hill. (1995)
10. Cabral, B., Leedom, L.C.: Imaging vector fields using line integral convolution. In: Proc. SIGGRAPH. (1993)
11. Haeberli, P.: Paint by numbers: Abstract image representations. Proce. SIGGRAPH (1990)
12. Hertzmann, A.: Painterly rendering with curved brush strokes of multiple sizes. Proc. SIGGRAPH (1998)
13. Salisbury, M.P., Anderson, S.E., Barzel, R., Salesin, D.H.: Interactive pen-and-ink illustration. Proceedings of SIGGRAPH 94 (July 1994) 101–108 ISBN 0-89791-667-0. Held in Orlando, Florida.
14. Sousa, M.C., Buchanan, J.W.: Observational models of graphite pencil materials. Computer Graphics Forum **19** (2000) 27–49 ISSN 1067-7055.
15. Durand, F., Ostromoukhov, V., Miller, M., Duranleau, F., Dorsey, J.: Decoupling strokes and high-level attributes for interactive traditional drawing. In: Eurographics Workshop on Rendering. (2001)
16. Ostromoukhov, V.: Digital facial engraving. Proc. SIGGRAPH (1999)
17. Curtis, C., Anderson, S., Seims, J., Fleischer, K., Salesin, D.: Computer-generated watercolor. Proc. SIGGRAPH (1997)