

# Through Other Eyes -responsive agent-based webcam imaging-

Joris Slob and Sonja van Kerkhoff  
[jorisslob@gmail.com](mailto:jorisslob@gmail.com) or [sonjavank@hotmail.com](mailto:sonjavank@hotmail.com)  
[www.sonjavank.com/through.htm](http://www.sonjavank.com/through.htm)

Media Technology, University of Leiden, 2007

## ABSTRACT

We conducted three experiments with live webcam imaging in which a python programme generates altered images which are the result of multi-agent perception.

In the first work, agents change colours in the webcam image pixel by pixel towards purer RGB colour. Their vision of colour is more ordered than the human, and this ordering reveals the primaries within all colour in the live image.

The second experiment blacks out all of the image except small windows which agents open if they are active at an edge of change, which may be either tonal difference or movement. A passing movement is captured as an area dotted with active windows, which then dissolve. So for the viewer, areas of greater visibility reflect change in the recent history of that area. The agents don't distinguish between change in time or in space, but for humans looking at the output, there is a distinction.

In the final work, the agents are individualized by their experience when they first interact with the image, according to the pixel they land on. The agents can then meet: if they share a similar level and type of experience a specific shape and colour appears at the location of the agent-to-agent interaction. Those 'cultural loci' grow in the output for a given time or until newer 'cultural loci' are created. So here the processed image is detached from location and instead simulates the agents' collective memory.

Like a memory landscape, these 'loci' fade in time and intermingle with other 'loci' and can be overwhelmed by new input. But not always: like human collective memory, there is a random element.

## 1. INTRODUCTION

This paper illustrates three experiments in python [1] in which the goal is to manipulate live webcam feed in intelligent ways to show something about the world that is beyond normal human vision.

In all our experiments artificial agents serve to 'order' the live feed while the world, through the webcam image, introduces 'dis-order.' Human nature tends towards a sense of order and our experiments serve to show how 'other

eyes' visualize order in different ways. Human history is a collection of shared peaks of experience and our final work creates a visual representation of these types of interaction between agents: in a sense it is an ordering of the highlights (or dramas) of the collective.

## 2. THE VALUES OF WHITE



Figure 1. *The original image and a few seconds later.*

In the first experiment the agent begins with a randomly assigned desire for either red, green or blue at a random location. They then 'read' the colours of that location. If the value of the primary colour in that location corresponds and is intense enough, then that primary colour is shown at that location. The colour we then see is a 'purified' extraction from the colour in the real world, which then fades. The agent moves to an adjacent location and the process is repeated. So what we see is a continual struggle of the ants purifying a world that is continually bringing disorder (in terms of non-primary colours).

The most successful result in terms of reaction to a changing image while retaining some recognition of objects, was when about 3000 agents land randomly over the image and where the threshold for outputting colour was fairly high (see Figure 2).

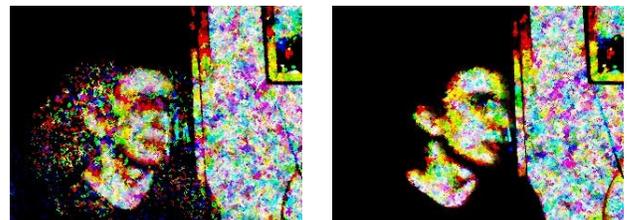


Figure 2. *He turns his head. Stills are about 20 seconds apart.*

The human eye is more sensitive to tonal change than to colour change and in fact we tend to see 'white' or 'black' as an absence of colour. White light is full of colour and the ants show this as the purest colour, as in Figure 1.[2]

### 2.1. Technical Description

The agents have a predetermined colour preference which is either pure red, green or blue. The threshold value determines whether pixels are purified or not. The threshold is a variable between 0 and 255 (where 255 results in the lowest possible output). In our experiments with these agents on our system, we found 3000 agents moving through a 320 x 240 pixel webcam stream, with a threshold of 100, gave the best results.

When the threshold is low, the output is a pretty abstract or a random colour mess, depending on your view. What this implies is that the selectively via the threshold directly correlates to our experience of order in the output.

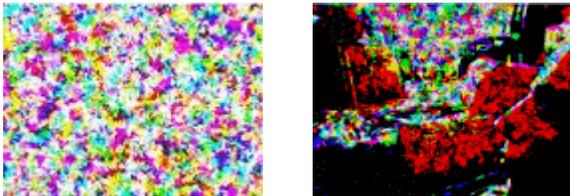
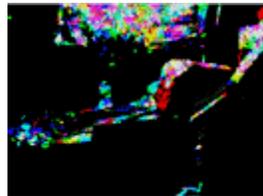


Figure 3. The same image: left: has a threshold of 5 while right has a threshold of 100. Below has a threshold of 150.

See [www.sonjavank.com/through.htm](http://www.sonjavank.com/through.htm) to view larger images.



### 2.2. Discussion

The resulting images look similar to the Pointillist and later Impressionist paintings in which tiny dots of pure color were applied to the canvas.

Not only do the changing images look similar, the process is conceptually similar.

Agents, like the painters, order colour to obtain greater purity. The resulting space seems flattened and mutable.

The main visual affect works much like a filter, but a filter created by numerous randomly appearing agents.

## 3. EDGES OF CHANGE

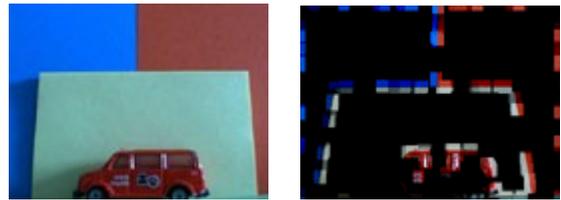


Figure 4. The original image and a few seconds later.

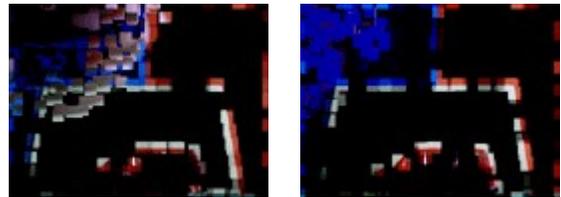


Figure 5. A hand moves over the top left and the same image a few seconds later.

Next we aimed to represent change as an instantaneous image. We did this by blacking out the image, and then allowing agents to open windows to it at the locations where they are stimulated by contrasting colour values, which may be due to either change or contrast in the image. For example in the last image of Figure 5, the area of blue windows indicates that there has been recent movement in the top left. The red-white windows show areas of high contrast in a static part of the image.

Normal vision sees solid objects in different locations from which we deduce movement, but this system aims to make the movement itself (and not the object) visible. The blue in the example above is the blue background, not the object that was displaced. So here each agents' 'eye' shows where change has occurred, where there is strong contrast in the image. If the image is static you end up seeing lines such as in figure 6 below.

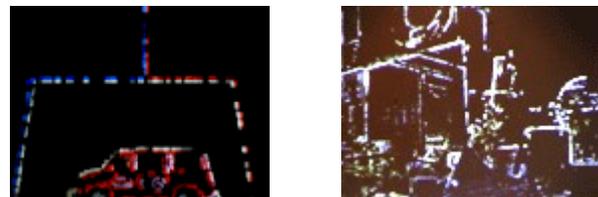


Figure 6. Right: Each agent at an edge creates 7 x 7 pixel 'window'. Left: Each agent creates a window of 3 x 3 pixels.

### 3.1. Technical Description

Our optimum version used a population of 5000 agents. Each agent has a memory (which stores one colour value) and a vision (the current colour at that location in the webcam image). They

appear at a random location with their memory set to zero (black). Before each move the agent compares the value of its current memory to that of the streamed colour at that location. If the difference between the two values is great enough, a window appears showing the webcam image, which then fades via a determined decay factor. The pixel size of the window was a variable we experimented with. While the larger window showed the goals of our experiment more clearly, we decided that a small value was a more interesting aesthetic experience. Here contours in a static image show up as fine lines while areas of change show as dots of colour.

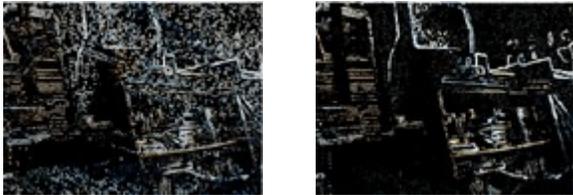


Figure 7. Left: A hand movement close to the camera, which then fades, while the contours remain (right).

### 3.2. Discussion

The eyes of the agent see delta (change) in ways humans can't. The areas showing a recent change slowly fade to black, resulting in a delicate aesthetic experience, reminiscent of Mark Tobey's white writing paintings. Paintings in which he aimed to visualise movement.

## 4. SHARED EXPERIENCE

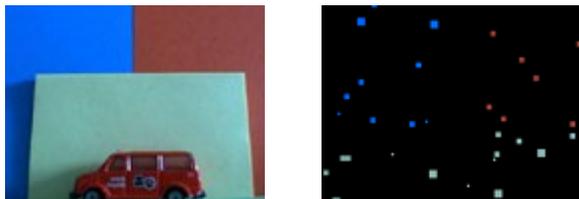


Figure 8. The original image and the first seconds of Shared Experience.

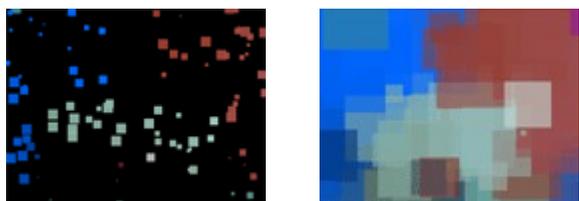


Figure 9. 5 seconds later, and 20 seconds later.

Thousands of agents appear at random locations on the webcam and they 'experience' that colour (the RGB values) of that location. Then they read a neighbouring position. If there is another agent there, they 'read' that agent's own colour experience and if this matches their own in intensity, they have a

'shared experience' and create a 'cultural locus.' in the form of a colour.

Our aim is to represent the agents' shared experiences. So the output colours are the result of two agents sharing an experience. Successive 'experience events' between other agents affect the duration of these loci. We see what was shared, not what was perceived, so in a sense the imagery here is of collective memory.

Each cultural locus appears as a square – or in some of our experiments as a circle – of the corresponding colour and starts as a barely visible, one pixel dot. This colour, the 'shared experience' then grows and gradually fades until replaced by a new event. Change in the webcam stream also affects these experiences, by increasing the probability of a shared experience.

### 4.1. Technical Description

Each agent has a placeholder for colour: a maximum experience memory. They appear with this set to zero (black). If the agent detects a colour change that has a higher intensity than it has in its memory, then it will remember the new experience. Then it plans to move and it checks if there is another agent in that location. If there is, it checks to see if that agent's experience memory is within the difference limits and, if it is, the agents' interaction results in the appearance of a shape (in our examples as circles or squares) which we call a 'locus.' Once the maximum number of loci is reached, the oldest locus is removed. The memory experience of that agent is set to zero and then it moves. At the end of the move the agent samples the colour from the webcam image again, and this becomes its new maximum experience memory.

Each experience 'locus' has an alpha value and growth variable. The alpha value is set to 255 and the growth variable ensures that the output image is always filled with colour. Our optimized version had a population of 1440 agents, with a maximum of 255 events possible.

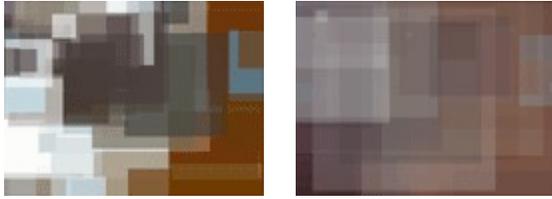
### 4.2. Discussion

This process mimics some features of shared human memory because agents reinforce each other's experiences. Memories that are not reinforced fade but with a certain randomness.

This simulation of a society of agents comes about because the agents do not simply respond to the webcam image as in the first experiment, or with an abstraction from the image (delta) as in the second experiment, but with each other. If there is no new input the output continues to change because of continuing interactions. The affect is dreamlike.



Figure 10. The image and three screenshots with no new input.



## 5. CONCLUSIONS

We started working in three different programmes, PureData [3], Jitter [4] and Python. Although neither of us had done anything with imaging in python before, it was easy to change our approach and our programmes as we experimented.

One limitation we found with python was that resizing the window any larger than 320 x 240 pixels slowed everything down. We will tackle this in our future development.

The first experiment works like a customizable filter. The second functions more like an aesthetic experience and invites participation because anything moving within the image leaves visual traces. The third appears to behave more intelligently as a whole. It is responsive to changes in webcam imaging but in ways that are not lineal.

## 6. REFERENCES

- [1] Python is an open-source object-oriented programming language. See: [www.python.org](http://www.python.org). We made use of the Python Imaging Library (PIL) at <http://www.pythonware.com/library/>. See our code and instructions for working with python for this project here: [joris.zestsoftware.nl](http://joris.zestsoftware.nl) or [sonjavank.com/through.htm](http://sonjavank.com/through.htm)
- [2] See: [www.sonjavank.com/through.htm](http://www.sonjavank.com/through.htm) for more visuals with links on this page to: [www.youtube.com/sonjavank](http://www.youtube.com/sonjavank) for movies that relate to this project.
- [3] PureData is an open-source real-time graphical programming environment for audio, video, and graphical processing. see: <http://www.puredata.org/>. GEM is the Graphics Environment for Multimedia library needed here.
- [4] Jitter. See: <http://www.cycling74.com/>