Robotic Canvas: Interactive Painting onto Robot Swarms

Merihan Alhafnawi¹, Sabine Hauert¹ and Paul O’Dowd¹

¹University of Bristol, UK
merihan.alhafnawi@bristol.ac.uk

Abstract

Robotic Canvas combines swarm robotics, human-robot interaction, and art. The project creates a robotic canvas using a swarm of decentralised robots that a human can interact with to paint. Beauty emerges from the robots interacting with one another, their environment, and a human performer. Robotic Canvas is a dynamic and interactive visual art medium, capable of displaying static images or video feeds, upon which humans can paint with their physical gestures or dedicated robots. Art making therefore becomes a collaborative and immersive performance between artists, digital content, and robotic agents. Results are demonstrated through a 200-robot performance, with each robot acting as a single pixel of the canvas. We further describe and characterise the various interaction modes, called “Painting Modes”, of the system. Results from this work form the basis for future research in expressive human-swarm interactions.

Introduction

This paper presents Robotic Canvas, an interactive swarm robotic system in the context of art. Robots act as pixels on a canvas, which a human can interact with to create paintings via 6 different modes of interaction, called “Painting Modes”. Robots act based on local interactions and interactions with the environment, preserving the key concept of decentralisation of swarm systems.

A swarm system operates via the emergence of group-level behaviours from many interacting agents (Bonabeau et al., 1999). Swarm robots are programmed to interact locally with other robots and their environment, leading to decentralised control (Trianni and Dorigo, 2005). As such, the behaviour of the swarm as a whole is emergent (i.e. group behaviour is not coded into individual robots). This has the potential to make swarm systems robust, scalable, and flexible (Şahin, 2005).

For swarm systems to be useful in the real world, humans will need to understand their state and influence, steer, or even control swarm systems. This is increasingly studied in the field of human-swarm interaction (Kolling et al., 2016). For instance, a human may want to communicate changes in previously set goals. However, robot swarms are challenging to command due to their typically large numbers (Bashyal and Venayagamoorthy, 2008) and their self-organised emergent properties (Kira and Potter, 2009). For instance, there are no centralised mechanisms (e.g. a communication channel, a hierarchy, or a single leading robot) to reliably and pervasively propagate a control signal. Therefore, it is important to develop interaction methods by which a human can influence the swarm as a whole, rather than controlling every robot to do so (Bashyal and Venayagamoorthy, 2008).

In this work, we influence the self-organising behaviour of a swarm without compromising the decentralised principles of its operation. As a proof of concept, robot swarms are used as a dynamic and interactive visual art medium, capable of displaying static images or video feeds, upon which humans can paint with their physical gestures and dedicated robots. Physical gestures utilise either the shadow cast by the human hand, or by manipulating dedicated robots. Art making therefore becomes a collaborative and immersive performance between artists, digital content, and robotic agents. Artistic interactions with swarm systems provide a compelling way by which human-swarm interactions can be studied. By engaging humans in an intriguing and enjoyable interaction experiences, such as painting with robots, human-swarm interaction studies can be carried out to explore interaction modalities and create compelling symbiotic performances.

Several modes of interaction were developed in Robotic Canvas to aid a human in creating paintings with the swarm of robots. The state of the robots is visible to the human by enabling swarm-level readout through the LEDs of the swarm robots. This visual feedback is important to create a closed-loop interaction by enabling the human to confirm that the swarm has understood the intended interaction. A human can influence the behaviour of the swarm as whole, or just a portion of it. Multi-human-swarm interaction is demonstrated, giving the ability for multiple humans to interact with the swarm simultaneously. A human (or multiple humans) can create performances with the robots, which may help understand and study the nature of human inter-
actions with robots in the future. We expect this research to serve as a basis for studying expressive swarms and novel forms of human-swarm interaction in future work.

This paper proceeds as follows: Section II situates this research against related works. Section III presents the Robot Pixels used to create Robotic Canvas and their characteristics. Section IV describes the modes of interaction developed. Section V shows the results in a form of a performance between a human and a swarm of 200 robots. Finally, section VI concludes the paper, while presenting the future work related to this research.

Related Works

Art has been used in the past as a context to explore human-swarm interaction. SwarmArt by Boyd et al. (2004) explored human-swarm interaction through art. The agents in the system used local interactions, and their position with respect to the centre of the task environment, to calculate their direction of motion and orientation to respond to the human’s influence. However, prepossessing of the interaction caused by human motion was done first, before being introduced to the swarm, which causes a risk for a single point of failure for the interaction process. In contrast, Robotic Canvas seeks to avoid single points of failure by a closer adherence to the principles of swarm robotics and decentralisation.

There has been research done that looked at how the self-organisation and emergence of swarms can potentially create abstract paintings. Greenfield and Machado (2014) summarised well some of the research that has been done in this field. For example, Aupetit et al. (2003) created digital paintings using the ant colony optimization algorithm. While this research and Robotic Canvas share the theme of painting, most of the swarm art is done in simulation, not allowing humans to interact with the swarm system directly. Robotic Canvas makes an emphasis on demonstrating human-swarm interaction.

Moving from simulated agents to actual robots, Pixelbots (Ackerman, 2014) is a project which aimed to create animations with many robots, each acting as a pixel on the image (Alonso-Mora et al., 2012). The authors also expanded their research to include human interaction with the robots, through arm gestures tracked by a Microsoft Kinect RGB-D sensor (Alonso-Mora et al., 2015). However, the control method used in Pixelbots is a centralised control, where a camera tracks the robots and a central computer sends commands to the robots separately. Robotic Canvas uses dedicated robots and gestures in order to facilitate human-swarm interaction within the mechanisms of decentralisation employed in the swarm.

Another application of art with swarm robotics is creating visuals in the sky via drones. An article that showed many research examples in this area was written by Waibel et al. (2017). Intel’s famous light shows are an example where up to 1218 drones were used to create animations in the sky (Intel, 2018). During the light show, a single computer controls the drones (Intel, 2019). Therefore, the system is still centralised, where each robot is told what to do.

Glowbots by Jacobsson et al. (2008) implemented a decentralised human-swarm interaction system in the context of art. LEDs on the robots were used to show patterns and react to human interaction, which involved physically shaking them. Depending on the direction of individual robot shaking, the robot’s pattern either got influenced by the patterns displayed by its neighbours, or its pattern influenced its neighbours’ patterns. Glowbots, similar to Robotic Canvas, tackled the human-swarm interaction challenge in a decentralised manner, where the human actions influenced the swarm. Robotic Canvas builds on this work, combining decentralised control, artistic expression, and novel interaction modalities on real robot swarms. However, Robotic Canvas explores different methods of interaction using dedicated robots and gestures.

Robot Pixels

Robotic Canvas is made of 2 main components: a swarm of 200+ stationary Kilobots and a Digital Light Projector (DLP). In this research, we refer to Kilobots as “Robot Pixels”, since every Kilobot acts as a pixel on the Robotic Canvas. Future work will look into adding movement to the Robot Pixels to create 2D sculptures. The setup of Robotic Canvas is shown in figure 1.

Figure 1: Robotic Canvas setup. a) Projector creating an image over the Robot Pixels, changing the ambient light around them. b) Robot Pixels interacting with one another through infra-red communication. c) User interacting with the Robot Pixels via hand gestures and observing their LED output. d) Toolbar containing dedicated robots that a user can use to interact with the Robot Pixels.

Kilobots are used as the swarming robots (Robot Pixels) in Robotic Canvas. Kilobots are small simple robots (33mm diameter) that use infra-red communication for local interactions and ambient light sensing for environmental interac-
tions (Rubenstein et al., 2012). In Robotic Canvas, the user can interact with the Robot Pixels either via dedicated robots collectively found on the “Toolbar”, exploiting the local interactions of the robots, or by changing the ambient light, affecting the Robot Pixels’ environment. The user can use their hand or an object as a shadow over the Robot Pixels to create a drop in ambient light, or use the projector to change the colour of light. There are a total of 6 different interaction modes the user can use to interact with Robotic Canvas, the “Painting Modes”. “Painting Modes” in Robotic Canvas are described in details in the next section.

Light Sensing
The DLP is used to shine an image/video/GIF over the Robot Pixels. Sensing the light colour through their ambient light sensor, the Robot Pixels recreate the image/video/GIF using their LEDs. Ordinarily, a Kilobot only detects an undifferentiated ambient light level (brightness) encoded as an 8-bit value [0:1023]. However, digital signal processing allows us to isolate which segment of the DLP colour wheel is being used to colourise the projection. Therefore, a brightness level can be correlated to a segment of the colour wheel. The Kilobot’s ambient light sensor output is sampled to match the DLP colour wheel rotation, and a 16-bit sample waveform is constructed to filter out three mutually exclusive colour states, as shown in table 1.

Table 1: Projected colours, the waveform created by the Kilobot’s ambient light sensor when those colours are sensed, and the colours the Kilobot’s LED is programmed to show when those colours are detected.

<table>
<thead>
<tr>
<th>Projected Colour</th>
<th>Waveform Pattern</th>
<th>LED output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Red/Green/Blue</td>
<td>1100000000000000</td>
<td>Blue</td>
</tr>
<tr>
<td>Yellow/Cyan</td>
<td>1111111000000000</td>
<td>Yellow</td>
</tr>
<tr>
<td>Magenta</td>
<td>1100001100000000</td>
<td>Red</td>
</tr>
</tbody>
</table>

The ability to reliably detect these colour wheel segments is reproduced by the individual Kilobots as three distinct output LED colour states: blue, yellow, and red. To provide a reliable detection and reproduction, projected images should only use colours from those 3 groups. Figure 2 shows a projected image reproduced by Kilobots (Robot Pixels) on Robotic Canvas.

The error rate of the display (i.e. the rate at which the robots display the incorrect LED value) is almost negligible: 0.09%. This error rate was determined by tests carried out on 50 Kilobots, separated into 10 sets (each has 5 robots). Each of the 6 colours were projected over the Kilobots for 5 minutes. The experiment was repeated 4 times. The total number of erroneous recognitions for the whole experiment (including all colours) was 311 out of a total number of 360,000 updates (since every robot updates their colour every second) for all robots and all colours. The average number of errors from the 4 repetitions for every colour is shown in figure 3.

Figure 2: Image of the sunset, shown on the left, with Robot Pixels adopting the corresponding LED colour on the right.

Figure 3: Errors in colour detection. The y-axis represents the number of errors in colour detection (311 errors in total, at a rate of 0.09%). The x-axis represents each colour. The points on the graph represent the average number of errors (from the 4 5-minute experiments) produced in each set (each point corresponds to 1 set, 10 in total).

Some wave patterns are recognised much better than others. Magenta is a good colour to use during performances since it was 100% error-free. Between cyan and yellow, yellow is obviously the colour to choose since it yields much less error than cyan. Finally, between red, green and blue, blue performed best. Hence, optimal projection during a performance can be achieved by using magenta, yellow and blue.

It takes Kilobots no more than 1.2 seconds to change from one colour to the other (when the projector shines a different colour). 50 robots were used to test how much time it takes the robots to completely react to colour changes. The test was repeated 10 times for each colour change. The overhead projector was used to display one colour, and then a human operator changed the image displayed to another colour. The whole process was recorded. The videos were then analysed to know how many video frames it took the robots to change from one colour to the other, which helped determine the number of seconds, down to millisecond precision, the change took.

Magenta, yellow and blue, identified from the previous
experiment, were used in this experiment. Hence, there are 9 possible colour changes. Figure 4 shows the average number, and standard deviation in brackets, of seconds it took all 50 robots to change from one colour to the other during the 10 repetitions.

![Figure 4: Matrix showing the average number of seconds (and standard deviation in brackets) it took the robots to change colour. The gradient represents the colour change that took place. For example, the top-left square shows that it took the robots 1.19 seconds to change from blue to magenta.](image)

Most colour changes performed very close to each other, with one exception of blue to magenta, which was slightly higher than the rest, with a few milliseconds difference. However, if the maximum number of seconds is taken as standard to apply to videos and GIF, it should cover all other colour changes. Hence, 1.19 seconds, equivalent to approximately 36 frames, deems suitable.

**Inter-robot Communication**

Local interactions are crucial among Robot Pixels since, through local communications and interaction with the environment, the Robot Pixels make a decision on which state to switch to (or stay on). In this sub-section, local interactions among the Robot Pixels are explained in terms of how the messages are exchanged and/or propagated throughout the swarm.

**Message Content** The message sent by Kilobots contains 12 bytes. 9 bytes are allocated for the payload, 1 byte is allocated for the message type, and finally, 2 bytes are allocated for CRC (cyclic redundancy check). 4 out of the 9 payload bytes are used by the Robot Pixels in Robotic Canvas. Each Robot Pixel was programmed to broadcast its ID, state, and LED colour. The ID code is broadcasted to all neighbours. The ID code is not used to identify any specific robot, rather, it is used to differentiate messages from neighbours. Each Kilobot has its own designated 16-bit ID, kilo_uid, that is given to it randomly while it is being calibrated. These IDs are used to track and update the neighbours. It is important to identify when a neighbour has last been seen, so that if the neighbour fails, is removed from the swarm or moved further away by a human, it would be removed from the list of neighbours when it has not been seen (heard from) again for one second. The reason why 1 second was chosen was so that the Robot Pixels can update their neighbourhood list as quickly as possible. This increases the adaptability of the system, by making the Robot Pixels adapt quickly when moved. The content of the messages exchanged by the Robot Pixels is shown in Figure 5.

![Figure 5: The content of the message broadcasted by robots.](image)

**Neighbourhood** The Kilobots send and receive messages continuously. Their communication rate can go up to 30 kb/s, and they broadcast roughly 3 messages/second (Rubenstein et al., 2012). This means that each Robot Pixel will receive many messages from its neighbours. To prevent overwriting high-priority messages (priority assignment will be described in the next section), each Robot Pixel keeps a volatile list of its closest 20 neighbours and their messages (among other attributes), and updates this list every second. The neighbourhood list is of size 20 because each Robot Pixel can reach a maximum of 20 neighbours in the setup used. The neighbourhood list is implemented in the form of an array of structs. Figure 6 provides a visual description of each Robot Pixel’s neighbourhood.

![Figure 6: The neighbourhood of each Robot Pixel in the form of an array of structs.](image)
The attributes stored for each neighbour are the ID of that neighbour, the distance from the robot to that neighbour, when the neighbour was last seen, the neighbour’s current state and the neighbour’s current LED colour (colour and state are important for some Painting Modes, as will be described in the next section).

**Message Propagation** One of the local interaction techniques used in Robotic Canvas is message propagation. This technique, used in 3 out of the 6 Painting Modes, was implemented to ensure that a message propagates through the canvas effectively and efficiently without having to communicate with every robot directly. The user can use a dedicated robot from the Toolbar to propagate a message to the entire swarm by placing it anywhere near Robotic Canvas. If this robot fails, it can be replaced with any other Kilobot. The dedicated robot constantly broadcasts a special state value, which has a high priority to make sure the human’s message is heard by all Robot Pixels. Priorities are discussed in the next section. When the human places the dedicated robot next to the Robot Pixels, the robots that are within communication range with the dedicated robot will store the dedicated robot as a neighbour and adopt its state (since it sends a high priority message). Then, their neighbours will adopt their state and propagate it to their neighbours, and so on. The propagated message reaches every Robot Pixel on the canvas without having to communicate with every Robot Pixel to do so. This makes the system flexible since the user can communicate with only a portion of the Robot Pixels, and the message will still reach all Robot Pixels in the swarm.

**Painting Modes**

Robotic Canvas is composed of a swarm of 200+ Robot Pixels. A human (or multiple) can interact with Robotic Canvas to produce paintings through 6 “Painting Modes”. There is no order of interaction modes, and not all modes have to be used. In some Painting Modes, the user can use dedicated robots from the Toolbar to communicate with the swarm of Robot Pixels. The Toolbar is shown in figure 7. The Painting Modes are explained in detail below. Results are shown in the next section as a performance that utilises the Painting Modes.

**Light Painting** This is the default mode for all Robot Pixels. The Robot Pixels sense the colour of the light projected on them (via a projector), and display the corresponding colour on their LED. The user projects an image of their choice (keeping in mind the constraints of colour sensing, described in section IV) over the Robot Pixels, which they then adopt. Another way to interact with the Robot Pixels via this mode is to project a video or a GIF file over them. This way, the Robot Pixels show a video rather than a static image. If the projector fails, the user can still use any of the following interaction modes to interact with the system.

**Copy and Paste** This mode allows users to select pixels, copy them, and paste them to another part of the canvas. When the user maintains a shadow over the Robot Pixels they wish to copy for 5 seconds, whether using their hand or an object, the pixels blink red and blue, indicating they are selected. Then, as the user moves their hand/object towards the part of the canvas they want to paste the pixels to (which can be anywhere on the canvas), the pixels move with them (the pixels move as the shadow moves). This provides feedback to the user that assures them that the system is aware of the shadow’s presence, and the copying is taking place. Then, when the user is satisfied with the new place to paste the pixels, they keep the shadow for 5 seconds over the new place they want to paste to, triggering the pixels in that place (i.e. under the shadow) to blink cyan and yellow. The pixels then paste permanently. The blinking of LED lights while copying and pasting provides a feedback mechanism to inform the user that their interaction took place.

**Undo All** Undo All is a mode that allows the user to undo all the changes they have made to the painting. The user can...
do so by using the Undo robot, which is a dedicated robot from the Toolbar, shown in figure 7C. When the Undo robot is placed near the canvas, it broadcasts a message that propagates throughout the whole swarm, telling the Robot Pixels to go back to their default mode, the Light Painting mode. When a Robot Pixel receives the undo message, it turns its LED red for one second, which causes the red colour to propagate throughout the swarm. This provides feedback to the user, as they see the colour red propagating, to know that the resetting of the canvas took place.

**Save All** The last mode in the system is the Save All mode, where the user can save and retrieve a painting they created. There are 5 dedicated robots, called the Memory robots, which the user can use to propagate a signal to the Robot Pixels telling them to save or retrieve paintings. These Memory robots are part of the Toolbar, shown in figure 7A. Each of the Memory robots changes their state every 10 seconds between save (showing purple LED) and retrieve (showing white LED). When their LED colour is purple, the user can use them to broadcast a message that tells the Robot Pixels to save the current painting. When Robot Pixels receive the save message, every robot will store its current pixel value (LED colour), which means Robotic Canvas will remember the painting saved. After 10 seconds, the Memory robots’ LED colour changes to white, meaning they have changed their state to retrieve. The user can then use them to broadcast a message to the Robot Pixels telling them to retrieve the pixel value saved. The user can overwrite the painting saved by the Robot Pixels when the Memory robots turn back to save mode. Since there are 5 Memory robots, the user can save 5 different paintings. This is because each of the Memory robots broadcasts a special number. Each Robot Pixel has an array of size 5 that stores pixel values corresponding to each Memory robot. When the user uses one of the Memory robots to propagate a save message, the Memory robot broadcasts a number that the Robot Pixels associate to a corresponding array element. When the Robot Pixels receive and propagate the save message, they turn their LED colour to purple for one second, similar to how they turn red in the Undo All mode. If they are propagating a retrieve message, they turn white. This again is done for user feedback, so the user knows that the save or the retrieve command took place.

**Coordinating Painting Modes**

Each of the Painting Modes act as a state in the system. A single Robot Pixel receives many messages, each of which may contain different states. This means that the Robot Pixel will have to make a decision based on possibly conflicting states. Therefore, a subsumption architecture (Brooks, 1986) was created. Each state has a priority in the system. Each Robot Pixel loops through its 20 neighbours and adopts the same state of the neighbour that has the highest priority state. Each state has its own unique number, which the Robot Pixels communicate to one another. Figure 8 shows the subsumption architecture diagram, which explains which states subsume other states, as well as what number is associated to which state. The end result is the value of LED displayed by the Robot Pixel. Save and retrieve states are represented by 5 different numbers. This is because there are 5 different Memory robots, each broadcasting 5 different numbers to be able to save five different paintings. The highest priorities were given to the states which involve message propagation to ensure they are able to flood the swarm. This is because it is desired that the message successfully reaches all Robot Pixels, so that the entire swarm is updated correctly.

![Subsumption architecture diagram showing priority of states](image)

**Results**

The system was tested by creating a performance that uses the Painting Modes. A human and a swarm of 200 Robot Pixels worked together to produce several paintings of the sky throughout different times of the day, as shown in figure 9. The performance is a story that the user paints with the robots about a sun that sets on the ocean, creating night with stars and clouds, and finally the sun rises again. A similar performance captured on video can be viewed on YouTube using this link: [https://www.youtube.com/watch?v=w7qhVT2V5nY](https://www.youtube.com/watch?v=w7qhVT2V5nY).

Figures 9a and 9b show a GIF of a sunset over an ocean being projected over the Robot Pixels (Light Painting mode). Figure 9c shows the user erasing pixels with the shadow of an object she is holding to create a night sky, shown in Figure 9d (Pixel Eraser mode). Multiple users can create shadows on different Robot Pixels at once, erasing pixels simultaneously. Figure 9e shows the user as she is painting Robot Pixels in the upper right corner using the magenta brush to create a star (Palette Colouring mode). Figures 9f through 9k show the user creating another star by copying the star she had already created in figure 9e and pasting it.
Figure 9: Results: performance made with 200 Robot Pixels, showing Painting Modes in action. a),b) GIF of the sun setting on the ocean appears on the Robot Pixels (Light Painting mode). c),d) Shadow is casted to create a night sky (Pixel Eraser mode). e) Magenta brush is used to create stars (Palette Colouring mode). f)-k) User creates another star (Copy and Paste mode). l),m) User creates 2 additional stars simultaneously. n) Clouds are painted using the cyan brush (Palette Colouring mode). o)-q) Undo robot in the lower right corner is used to propagate an undo signal (Undo All mode) triggering the Robot Pixels to go back to Light Painting mode. r) GIF of the sun rising appears on the Robot Pixels (Light Painting mode).
elsewhere (Copy and Paste mode). Figures 9l and 9m show 2 hands simultaneously copying and pasting to create 2 new stars, demonstrating multi-human-swarm interaction. Figure 9n shows clouds added using the cyan brush. Figures 9o through 9q show the undo all message from the Undo robot, placed in the lower right corner, as it propagates throughout the swarm, reaching all Robot Pixels (Undo All mode). As the message propagates, a wave of red LED colour propagates throughout the swarm. After the reset takes place, the Robot Pixels go back to showing a GIF being projected over them of a sunrise, shown in figure 9r (Light Painting mode).

**Conclusion**

This paper presented a novel system that combines art, swarm robotics and human-robot interaction: Robotic Canvas. A swarm of robots was used as a visual medium to create paintings in coordination with one or more human users. This paper outlines the technological mechanisms experimentally developed to enable such human-swarm interactions. Six modes of interaction, the Painting Modes, were developed through which a human can communicate with the robot swarm. Using a creative context, such as art (through painting), by which human users can engage with robot swarms provides an opportunity to create human-robot interaction studies in the world of swarm robotics.

The robots in the system used only local interactions and interactions with the environment (through ambient light sensing) to determine the nature of the human interaction that took place. Hence, the key feature of the decentralisation of swarm systems was preserved. Moreover, the system supported multi-human interaction; more than one human can interact with the system simultaneously. The system also gave the user visual feedback after every interaction, so the user can confirm their interaction took place, and understand the state of the system. To test the Painting Modes, a performance was created with a swarm of 200 robots (Kilobots), showing a sun setting on an ocean, creating a starry night with clouds, and finally the sun rising again.

This research is the first step towards studying expressive swarm systems that reflect their state to users to create better human-swarm interaction experiences. Future work will look at creating more performances to be taken for public engagement. This will provide good grounds to understand the usability of the system. Another step is to add movement to the swarm agents. This transforms the visual medium into a sculpting material, where the user can influence the direction of motion of the robots to create 2D sculptures.

**References**


