

Digital Airbrush

Borut Batagelj, Jakob Marovt, Miha Troha, Dominik Mahnič

University of Ljubljana, Faculty of Computer and Information Science, Tržaška 25, 1000 Ljubljana, Slovenia
E-mail: borut.batagelj@fri.uni-lj.si

Abstract - *The basic idea is simple: graffiter paints with a specially modified can, which when pressed on the cap does not leave color traces, but emits infrared light. The computer application draws the appropriate graffiti on the canvas, which is positioned in front of the graffiter. Infrared camera provides detection of the light source for the application, which then correctly determines the color, size and density (the graffiter regulates these parameters through the speed buttons, which are located on the can) and in the end of this process, the current drawing track is projected on the canvas. The most important algorithms used in the implementation of the software solution are: an algorithm for locating the brightest area, the algorithm for the implementation of the drops and the algorithm for interpolation. Important component of simulation is also a Bluetooth connection to the Arduino BT platform, which is connected to three potentiometers. These are used by graffiter to manipulate painting parameters. The results of extensive tests have shown that the prototype has potential and that the final simulation perfectly follows the basic idea. The problem occurs only in the lack of quality of the equipment (infrared camera, computer power) for full implementation.*

Keywords – *digital airbrush, graffiti, airbrush, virtual art, computer vision, interactive entertainment*

1. INTRODUCTION

Digital airbrush project deals with the interesting, but somewhat neglected area of artistic creation. Digital graffiti drawing throughout the world presents a unique technical challenge, but not many artists, until now, fulfilled the ambition. The idea for the project was provided by Dominik Mahnič, a postgraduate student of the Academy of Fine Arts and Design, University of Ljubljana, since he dedicated much of his free time for making graffiti paintings. His idea proposal included the use of infrared locator followed by "spray" and the canvas (standing in front of the graffiter) on which the graffiti are drawn in real time. To detect the source, we used the infrared camera and a computer application that processes the image and sends it to the canvas. At the same time, according to the artist request, there was also the possibility of manipulating the color, thickness and density of ink. Three rotational buttons (potentiometers) inserted in the can were used for this purpose. The open-coded interface Arduino [1] is attended to read the information from analog potentiometers and sending them via a wireless Bluetooth connection to the computer application.

The paper is organized as follows. In section 2 we explain similar projects and give a description of the system components. The computer vision method for detection of the IR light source is described in section 3. In section 4 we describe the implementation of the artistic elements (style of the trace, droplets etc.). After testing, due to the technical disadvantages of the camera, we implement an interpolation which is described in more detail in section 5. More about the integration

of the Arduino platform is found in section 6. In conclusion, we describe the main problems and their solutions, which have contributed to the final result of the project.

2. PRESENTATION OF IDEAS AND INDIVIDUAL COMPONENTS

Graffiti as one of the urban art appears everywhere, but there are very few places, where they are allowed and accepted as something positive. The basic idea of this work is to make the graffiti drawing acceptable, because they can be removed at any time. The image is projected and not really drawn, as in the case of real graffiti.

All other components of graffiti drawing must remain unchanged and similar to reality. Similar ideas have persuaded many individuals [2, 3, 4]. The authors present the main idea but no one gives the detailed description. Mainly the common parts are the same: the camera, which locates the IR light source, computer for image processing and the projector, which projects the processed image to the surface. The significant differences are in how to select the color, size and density of the stamp. Some of the projects select these parameters in the surface, but mostly they don't include these options. Our implementation allows the selection of these parameters by means of potentiometers, which are located on the can itself. In this way the "digital can" better illustrates the usual situation.

Even programming languages for implementation are quite different. Some use Flash, C++ etc. We use a Processing [5], which is an open source programming language and an environment

for programming images, animations, and interactions. It is ideal for artists, designers and researchers for learning, prototyping and production. The electronic board Arduino [1] development environment is also based on the Processing programming language. During the development we have seen the need for interpolation; hence we used the already tested mathematical square interpolation technique [6].

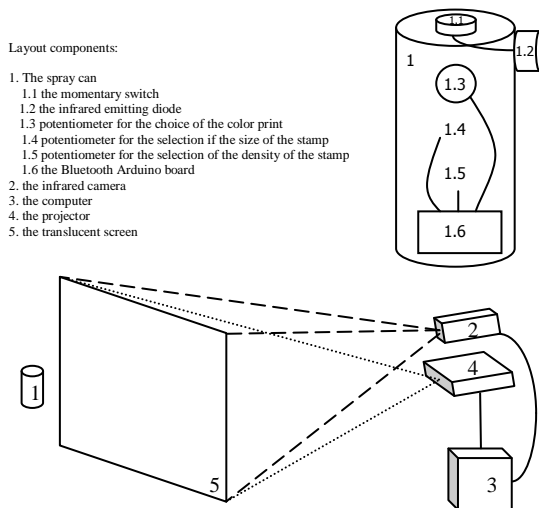


Fig. 1. The main components of the layout for digital graffiti painting

Description of the graffiti painting process (Fig. 1): Graffiter chooses the desired painting color (1.3), the radius (1.4) and density (1.5) of the ink stamp with potentiometers. He moves closer to the translucent screen (5) with the can (1), pressing the button (1.1) at the location on which he wishes to spray. Pressing the button triggers a source of infrared light (1.2). Infrared camera (2) captures the image and sends it to the computer (3). A computer vision algorithm detects the source of the infrared light (1.2) and determines its location. Through the Bluetooth connection Arduino (1.6) sends the value of the potentiometers to the computer (3). Computer (3) on the basis of location and value of the potentiometers produces an image and sends it to a projector (4). The image is then being projected on the translucent screen (5). Such a procedure allows the graffiter interactive drawing on the canvas.

In the next sections we describe each of the algorithms running on a computer (3). These algorithms process the images supplied by the camera (2) and use the values provided by the BT Arduino board (1.6). A drawing is then projected with the projector (4).

3. THE LIGHT DETECTION ALGORITHM

The general algorithm, which simply checks all the pixels in the video frames and finds the brightest ones, appears useless, since it is extremely sensitive

to interference and reflections from the environment. Very often the case is that the brightest points on the screen are the result of infrared light source being reflected from glass or other surface. This in practice means a print on the drawing, which is quite far from the location of the infrared light source. Such a mistake is very noticeable, and thus unacceptable. An excellent solution is to search for the brightest area and not just for the brightest point.

Experimentally, we have decided for the area of a square of 25×25 pixels. But we found the intuitive algorithm, which calculates the brightness of the surroundings of each pixel, too slow, since such implementation uses a lot of duplicated computation. In the region of 10×10 pixels for example two horizontally adjacent pixels calculate the brightness for the same rectangle of 10×9 pixels region. Our algorithm calculates the brightness more effectively. To calculate the brightness of the right pixel, we first subtract from the brightness of the left pixel the most left column and then add the brightness of the most right column (Fig. 2).

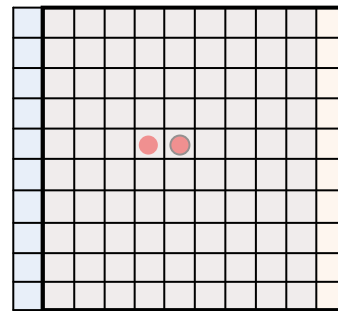


Fig. 2. The example of the 10×11 pixels region used in effective calculations of the brightness

A similar conclusion can be made in the vertical direction. Furthermore, we find that we can use the same approach for calculating the brightness of columns and rows. Taking into account the above facts, the sum of the brightness of any two adjacent pixels is made only once. Then it is stored for further use. Such an implementation of the algorithm is considerably faster and quite suitable for use in our project. To make the search even faster, we can take into consideration the fact that the painter is not very fast and in $1/30$ sec (this is the time between the two captured frames) the painter does not move for more than 5 cm, what is 200 pixels in our resolution. Thus, in the case that the IR source is turned on, we do not have to search the entire image, but only a sufficiently large surrounding area around the former location of the IR light. When the light is off, we have to search the whole picture, since the source can appear anywhere.

4. THE DROPS IMPLEMENTATION ALGORITHM

In the case of real graffiti drawing, in the places with high ink saturation, drops appear and are moving on the wall in the direction of the gravitation (Fig. 3). For the implementation of this phenomenon, we split the whole image in the regions of 20×20 pixels. For each region the data of the red, green and blue color component are saved. The color of newly created droplets is equal to the average of the colors that are currently in the specified location. Each field has also a mass. The weight increases with each new print in the particular area, but the increase also depends on the size of the radius and the density selected. The mass of the areas decreases over time, which is equivalent to drying in the real world. The quantitative value of the mass of an area is constantly compared with the randomly chosen numbers and when a mass is greater the new drop appears. At that time the weight of that area is reduced to 0. The droplet moves at random speed. With movement the drop is losing weight and when it is sufficiently reduced (this value is random) it stops.



Fig. 3. The graffiti drawing with the implementation of the drops

5. THE INTERPOLATION ALGORITHM

The camera captures images with a frequency of 30 Hz. Assuming that the hand drawing often makes rapid movements, we conclude that capturing images with the camera in such a way can be problematic. For example, rapid hand movement has velocity around 1.5 m/sec. This means that at such a movement a broken line with a 5 cm (150 cm / 30 frames) space between line parts can appear (Fig. 4).

We found the solution in interpolation. Algorithm draws an interpolation polynom through given points which allows us to properly determine the coordinates of points even between the given points. In our case we keep the history of the last two calculated points, which with the last calculated point forms the basis for calculating the coefficients for 2. degree interpolation polynomial (a linear

combination of the three Lagrange polynomials). Between the last two points we then, with the help of interpolation polynomial, obtain a predefined number of points. In order to evenly distribute the points between the target points, we get the component x of the desired points by drawing a line between two target points. Uniformly spaced perpendicular points in the intersection with interpolation polynomial mark the desired point. In some extreme cases, where use of the quadratic interpolation is not possible whether the x component of the individual points are the same or identical to components of y), we use a simpler linear interpolation algorithm, which uses 1. degree interpolation polynomial and requires only two target points (Fig. 5).

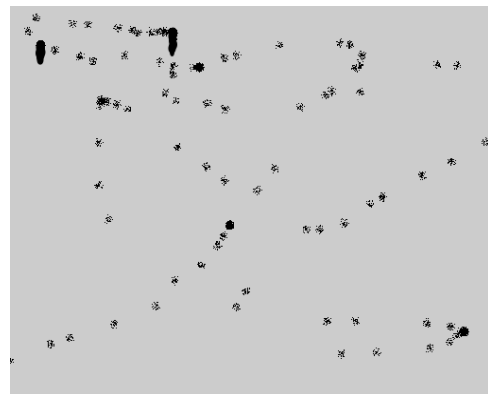


Fig. 4. The drawing without interpolation

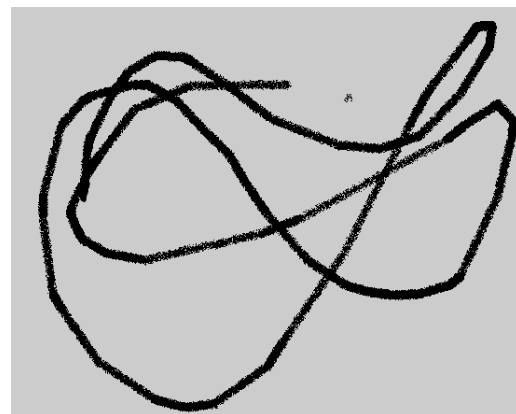


Fig. 5. The drawing with interpolation

6. THE CONNECTION OF THE ARDUINO BT INTERFACE

Arduino is an open-source electronics prototyping platform specifically adjusted for easy use and interaction with external components. The microcontroller on the board is programmed using the Arduino programming language (based on Wiring) and the Arduino development environment (based on Processing) [1, 5]. Thus, the connection

interface via a Bluetooth connection is fast, intuitive and easy to make. Three potentiometers were connected to the chip. Each potentiometer is installed to grounding output, power output (5V), and to one of the seven available analog inputs of the chip. A short program code is loaded on the chip, which initializes connection (115200 bit/sec) of the chip with a computer via the serial interface and a start up values of the three variables (one for each potentiometer).

Our software solution written in the Processing environment can connect to the Arduino chip via the serial interface and with the same frequency as the Arduino sends the information it also reads them. Data supplied by the Arduino occasionally contain anomalies in terms of completely divergent values (the error is probably due to the analog data). We solved this problem by keeping the history of the downloaded data and setting bounds of the possible values. The values obtained by potentiometers are between 0 and 1000. The potentiometer can be rotated to exactly one turn. The possible values of potentiometers variables for each parameter is adjusted appropriately (e.g. number of possible colors are 32, which means that each color takes 31 values).

7. CONCLUSION

We fully implemented the initial idea of the project Digital Airbrush. In the beginning, our perceptions on the implementation of the project were idealistic and, although we know that during the development of the project difficulties can occur, the magnitude and complexity of the project were underestimated. The first serious problems aroused with the problem of detecting infrared light source; we solved these problems with the algorithm for finding the brightest regions. With this algorithm it is possible to draw points or ink prints with the highest possible frequency. Fast drawing often results in broken lines which we can avoid by interpolation.

We also found some problems during the final testing and setup of the equipment. Because of the poor resolution of the initial camera, we used a modified webcam with a better resolution of 1380×960. Another problem was the focus of the camera and projector, which must be the same. For

manual focus we would need a better camera, so we were obliged to a few compromises, but the project was not compromised. A better camera would also allow the installation of camera on top of the projector in order to ensure the perfect synchronization of the two video devices, since otherwise experimenting with the setup of the system is needed.

The experience was very instructive and interesting in terms of the use of computer skills in artistic purposes. The idea itself and its implementation, with some additional tests and better equipment, could represent an interesting artistic concept, which in reality can serve as a "sandbox" for graffiter and better acceptance of their expression of art in public. With the use of our digital airbrush the artist already made some interesting drawings (Fig. 6).



Fig. 6. Dominik Mahnič: "The woman"

REFERENCES

- [1] Arduino, <http://www.arduino.cc/>
- [2] Wii Spray, <http://www.wiispray.com/>
- [3] YrWall - You Paint Digital Graffiti, <http://www.yrwall.com/digital-graffiti.aspx>
- [4] MobiSpray: Graffiti aus der Handy - Spruhdose, <http://mobispray.com/>
- [5] Processing, <http://www.processing.org/>
- [6] Interpolation - Wikipedia, the free encyclopedia, <http://en.wikipedia.org/wiki/Interpolation>