Development of Portable Painting Format for CNC Painting

Jerome J. Leary
University of Brighton
Faculty of Science and Engineering
Brighton BN2 4GJ
United Kingdom
jj.leary@brighton.ac.uk

This paper describes the development of a Portable Painting Format for CNC machine-produced paintings. The file format was based on G-codes, used time-based vector graphics, and included additional data for paint and brush properties. The file format can be used either to manufacture a real painting on a CNC machine or used to create a digital visualisation of the painting as an animation. A browser application called CNCPaint was developed to generate the file format for Pop-Art style paintings.

1. INTRODUCTION

The work described in this paper concerns the development of a vector-based file format to enable a standard Computer Numerical Control (CNC) vertical machining centre to produce paintings using traditional brushes and paints. The initial aim of the project was to build a painting ‘aid’ to support artists with disabilities who wished to paint with the help of a machine.

An additional requirement was the need for a real-time vector-based computer simulation of the same painting, showing the individual brush strokes being applied, and to show the paint layers building up as the painting progressed. The simulation needed to be panned and zoomed without being limited by screen resolution.

The real and virtual applications required a vector-based file format and it seemed that a common standard could be developed to fulfil both roles based on an industry-standard language for CNC milling called G-codes (Smid 2008) which allowed extensions to add metadata. The additional data could include parameters specific to painting, such as brush size and type, paint viscosity, colour and transparency. The width of the brush strokes in the CNC machine was to be controlled by the z-dimension and this enabled the start and finish of each brush stroke to taper as the brush tip varied the pressure on the canvas. The rate of flow of paint from the brush would depend on parameters such as the pressure applied, the viscosity of the paint, the acceleration and speed of the brush, temperature, and the time elapsed. Paint behaviour could also be predicted from knowledge of viscous fluid properties.

The requirement for a common portable painting format (PPF) led to an interdisciplinary project within the university involving staff from areas of software design, engineering and art.

2. SYSTEM DESCRIPTION

2.1 CNC machines

Computer numerical control (CNC) machine tools have revolutionised traditional manufacturing processes and are now widely used in production. They have evolved into reliable workhorses of manufacturing industry and can operate automatically, accurately and continuously for long periods of time, usually only requiring manual intervention for setting up and removing jobs. The costs of CNC machines have also reduced and some self-build kits are now available. Their flexibility to be programmed for different tasks has led to other uses, for example: prototyping, mould-making, sculpting and even robotic development.
In a typical CNC process, a component is designed using a computer-aided design (CAD) package and then produced using computer-aided manufacturing (CAM) software. The CAM software produces a computer file in the text-based language of G- and M-codes which provide specific instructions to the CNC machine, such as:

G00 – Rapid move
G01 – Controlled feed move in a straight line
G02 – Arc or circle interpolation
G04 – Dwell for a predetermined time period
G21 – Program in millimetres
G57 – Switch coordinate systems
G90 – Absolute positioning
M05 – Spindle stop
M06 – Tool change
M09 – Coolant off
M30 – Program end code

An example of a G-code program is shown below. A brush is selected in Tool holder 12 and is moved in a straight line from position x=0, y=0 to x=100, y=100 mm at a feed rate of 1000 mm/minute. The ‘%’ characters denote the beginning and end.

```
%01234
G21 G40 G80 G90
G57 M9
M06 T12 S0 M3
G43 H12
G0 X0 Y0 Z10
G1 X0 Y0 Z0 F1000
G1 X100 Y100
G0 Z10
M30
%
```

For this project a Kryle VMC700M vertical 3-axis CNC machining centre was used with a working volume of 700 x 450 x 300 mm. This particular CNC machine has twelve tool holders which can be fitted with drill chucks, and the machine can hold up to a maximum of twelve brushes. The maximum useful canvas size was therefore about 650 x 400mm.

A mechanical rig was built to hold the canvas in place and fix the position of the paint pots relative to the canvas. These positions were used by the CNCPaint software in generation of the PPF file.

### 2.2 CNCPaint

To build the CNCPaint web-application, three methodologies were considered: HTML5/Javascript, Microsoft Silverlight, and Adobe Flash.

HTML5 introduced the `<canvas>` graphics element which allows for dynamic, scriptable rendering of shapes and bitmap images (and is due to be released as a formal W3C standard in 2014). HTML5 doesn’t require a plug-in to be installed, unlike Silverlight and Flash and is already supported by many new platforms, such as mobile smartphones and tablet PCs. However, at the time of writing, some features essential to this project had not yet been implemented in HTML5 such as the FileSystem API for reading and writing local files.

Microsoft Silverlight also uses a similar `<canvas>` element as HTML5, and also enables programmable rendering of shapes and bitmap images on the canvas. It uses the .NET framework via a plugin. However, the Silverlight FileSystem API is more mature and permits reading and writing.
of files on the client computer from within the client browser. Other useful properties of Silverlight are that the downloaded code is pre-compiled and also protected from the user. But mainly due to its FileSystem API, Silverlight was chosen for the project.

The development system consisted of Microsoft Visual Studio 2012 IDE, the C# compiler and Silverlight version 5. The active server pages ASPX and XAP files were hosted on an ASP.NET server. The resulting CNCPaint web-application works in most modern browsers with the Silverlight plug-in is installed (PC: Internet Explorer, Firefox, Chrome; OSX: Safari, Firefox, Chrome), but not on mobile or tablet devices. Since beginning this project, Microsoft and Adobe have ceased development of their own proprietary frameworks (Silverlight and Flash) and are now supporting HTML5, and a FileSystem API is being developed. It is therefore probable that further development of this project would use HTML5.

The short example code shown above is the minimum code needed to control the motion of the CNC machine to paint a single brush stroke. Additional subroutines were developed to perform the following functions:

- Set the position of the canvas
- Set the length and size of the brushes
- Dip the brush in the paint
- Clean the brush
- Change brush
- Control the painting pressure
- Control the painting speed and direction
- Control brush stroke length
- Paint repeated stokes
- Estimate the paint left on the brush
- Estimate the levels of paint in the pots

An input configuration file defines the default geometry (the location of the pots and canvas) the brush parameters and the paint colours, for example:

```xml
<config>
  <pot id="1" x="500" y="200" z="0" hole="25" height="100" level="50%"/>
  <paint id="1" type="oil" colour="rgba(255,0,0,0.4)" />
  <brush id="1" tool="12" length="10"
        dila="5" type="round"/>
  <canvas id="1" x="0" y="0" z="0"
           length="200" width="100" height="30" />
</config>
```

One pot is reserved for brush cleaner, and when the painting of each colour is completed the brush is dipped into the cleaner and cleaned by moving back and forwards vigorously. At the end of the painting the brush can be held submersed to prevent the brush drying out.

2.3 CNCPaint - Pop Art

In the initial experiments in this project, paintings were to be produced in a limited range of colours to produce works in the pop art style of Andy Warhol. Warhol developed a signature style, slowly eliminating the handmade from the artistic process. He developed the technique of projecting photographic images to a silk screen, and tracing drawings from slide projections. The process enabled him to produce repetitive patterns within the same piece. At the height of his fame he had several assistants who manufactured his silk-screen multiples, following his directions to make different versions and variations (Colacello 1990).

As Warhol attempted to remove the handmade from the art production process, it is conceivable that, if CNC technology had been available in the early 1960’s, Warhol may have considered the use of such mechanical tools. So, for this project, rather than stepping on the hallowed ground of old-school artists, where ‘handmade’ is the essence of the art, the author of this paper feels comfortable in the knowledge that for the pop-art style, at least, Warhol himself may also have used CNC mechanical production techniques had they been available.

3. IMAGE PROCESSING

3.1 Image Preparation

Once the software CNCPaint is launched in the browser a digital image (either JPG or PNG) is read from the local file system. The image is then automatically smoothed, reduced to a few monochrome shades, and de-speckled using algorithms for image processing. For this project, four monochrome shades were used. The intensity levels of the original corresponding to each monochrome shade could be interactively altered by the user. The lightest shade is mapped to the colour of the first layer of paint on the canvas and the darkest to colour of the top layer of paint.

3.2 Creation of Contours

The next stage of image processing involves selection of paint layers and the calculation of contours which represent the distance of a brush from an edge. This distance is used in the painting process to control the reduction of pressure on the brush, so that when near to an edge, only the tip of the brush is touching, and when the brush is further away from an edge, or perhaps where there is an area to be filled, it presses down more firmly, to apply more paint to the canvas.
The distance from each point to the nearest edge is calculated and the results are stored in a 3D contour map. Each contour represents an increment in the distance to the nearest edge. The distance between contours is set at 1 pixel corresponding to a surrounding square of size 3 x 3 pixels. As a result, the further the contours are away from an edge detail, the smoother the contour lines. Figure 1 shows the stages of contour generation when the image is reduced to four shades.

3.3 Calculation of Brush Path

The calculation of brush path occurs in two stages which are done automatically: block fill, and edge tracing.

(i) Block Fill
This involves calculating the brush path for the large areas that need painting or filling in. The brush moves down onto the surface at random contour positions (on a contour greater than n, where is n is the radius of the brush in pixels) and then moves off at a random angle for a random distance. The brush then lifts off the surface and moves somewhere nearby when any of the following conditions occur:
- The brush touches the innermost contour
- The brush touches the edge of the painting
- A preset distance has been covered
- The brush needs to refill with paint

When the brush moves to a new position, the software checks that:
- The new point is close to the previous position, unless the brush has been refilled
- The new point has not already been painted

An intermediate result is shown where the contours and individual brush strokes can be seen. The final result of block filling is shown.

(ii) Edge Tracing
This involves calculating the brush path following along the boundary. Only the first few contours are used. During this phase the brush does not refill but blends the wet paint which is already on the surface of the canvas.

3.4 Output to File

The browser then saves the completed G-code program to a text-file file on the client's computer in the portable painting format (PPF) described in the next section.

4. FILE FORMAT

The portable painting format (PPF) file contains G-codes, M-codes and meta-data. The structure of the PPF is a text-file consisting of the following sections in order:

(i) Title and description entered by the user.

(ii) Geometry information. This contains the 3D specifications for a target CNC machine to manufacture its own painting apparatus from a single board.
(iii) Paint information. This defines the oil paint colours and properties, and can include supplier part-number.
(iv) Brush information. This defines the dimensions and properties of the brushes and also part-numbers.
(v) Setup information. Instructions to the CNC operator how to configure the machine origin and axes.
(vi) Program. The following is repeated for each colour. This is where the main body of the file is contained using standard G-code commands which can be interpreted by the CNC machine motion controllers.
   (a) Brush selection.
   (b) Paint dipping
   (c) Paint lines
   (d) Brush cleaning
(vii) Shutdown. The CNC is put into a passive state and the brushes are put away.

The PPF file contains all the information needed to produce a painting on any 3-axis CNC machine. It has the following uses:
- The PPF file can be sent directly to a local CNC machine configured to produce a painting
- The PPF file can be transmitted over the internet to produce a painting remotely on another CNC machine which is configured for painting.
- The PPF file can be used to generate a computer animation of a painting being produced.
- The PPF file can be used to create a digital image of a completed painting.

5. EXPERIMENTS

During the experiments to produce pop-art style paintings on the CNC machine (Figure 2) several issues were identified which were addressed as follows:

Paint spread outside the target brush contact zone because a build-up of wet paint migrated or diffused into adjacent wet areas. Also discharging of paint from the brush sometimes affected neighbouring areas. Although the encroachment in most cases was relatively small, the visual effect on the painting could be quite noticeable. The following solutions were tested:
- Wipe the brush on the sides of the container to control the amount of paint.
- Applying a safety zone around the brush contact area so that the bristles do not approach too closely to a colour boundary.
- Allowing for the bending of bristles in the painting algorithm to predict the contact area.

![Figure 3: Brush motion](image)

When the brush descended vertically onto the canvas the bristles splay out slightly and separate from each other in a circle. This resulted in a larger than intended initial paint contact area. One enhancement to solve this was:
- To descend onto the canvas at an angle in the direction of the intended motion, and then gradually increase the pressure of the brush whilst moving. The bristles then all bend backwards together (Figure 3).

When the brush is initially moving on the canvas with a full load of paint, the rate of application of paint is greater at the start of the stroke than at the end, when the brush is empty of paint. This resulted in lines of paint of reducing density. Two approaches used were:
- Vary the speed of the brush so that it starts quickly and slows down during the stroke. This gives the paint less time to flow down the bristles and onto the canvas when the flow rate is faster.
- Touch the brush lightly at the start of a stroke and gradually increase the pressure; by lowering the brush proportionately as the distance travelled increases.

The horizontally supported canvas sagged slightly where the support is the least, in the middle, and the sag increased as the weight of paint built up and the canvas loosens. This resulted in the brush not making as good a contact in the centre as the painting progressed. Two solutions were:
- Addition of a support beneath the canvas so that the whole width is mechanically supported.
- Addition of software to compensate for the sag, by lowering the brush in the centre, as the cumulative weight of paint can be predicted.
6. CONCLUSIONS

The portable painting format (PPF) is based on an extension of the industry standard G-code language for CNC milling machines with extra information added for: geometry, brush size and type, paint viscosity, colour, transparency and canvas size. Brush width was controlled by motion in the z-dimension, which enabled the start and finish of each brush stroke to taper as the brush tip varied the area of its contact point. The calculation of the amount of paint remaining in the brush depended on the applied pressure, the viscosity of the paint and the time elapsed, and paint flow behaviour was modelled from knowledge of viscosity and changes in the acceleration of the brush.

A web-application called CNCPaint was developed which converted digital images into 'pop-art' PPF files, enabling any other standard CNC machining centre to be used for production of pop-art paintings. In the painting experiments described within, only one brush size was used but the method can cope with multiple brushes. A painterly effect was achieved by using random brush strokes in the block fill areas and visible brush marks around the detailed edges gave the painting an appearance of being hand-painted. The finished painting can be duplicated on any other standard CNC machine anywhere in the world.

7. REFERENCES


Microsoft Silverlight, Microsoft Corporation.