Stylisation of Recognisable Characters

Sarah Noonan
B.A.I. Engineering
Final Year Project – April 2014
Supervisor: Dr. John Dingliana

School of Computer Science and Statistics
O’Reilly Institute, Trinity College, Dublin 2, Ireland
Declaration

I, the undersigned, declare that this work has not previously been submitted as an exercise for a degree at this, or any other University, and that unless otherwise stated, is my own work.

__________________________________________  __________________
Name                                           Date
Permission to Lend

I, the undersigned, agree that Trinity College Library may lend or copy this report upon request.

__________________________________________  ___________________
Name                                                     Date
Acknowledgements

I would like to thank my supervisor Dr. John Dingliana, for the opportunity to work in this area, and for all the guidance and advice that he has provided to me throughout the year.

I would also like to thank my parents for their continuous support over the course of my degree, and my roommate Clementine for the encouragement and motivation she has given me in the last year, as well as acting as my model for the pictures in this report.
Abstract

For many years, the main goal of computer graphics has been to create a scene that is visually indistinguishable from reality. However this is not always the desired result. Depending on the context, graphics that have been rendered to a specific style can be more effective than if they were realistic.

However, stylisation can be a problem when it comes to rendering faces. The brain’s reaction to a human face is unique. All faces consist of the same major features, yet our minds can perceive tiny details that enable us to distinguish between even the most similar of faces. Stylisation can obscure fine details, which with most objects is acceptable, but with faces can greatly affect the recognisability.

The aim of this project was to investigate the effects of stylisation on faces, and to develop a method to attempt to minimise any reduction in recognisability. The result is a program that can take in an image, and return a stylised version of that image, optimised so that any face in the image remains recognisable. To allow for the subjectivity of the perception of quality for stylisation, the program allows the user to modify the parameters of the style, creating a customised result.
# Table of Contents

Declaration.......................................................................................................................... ii
Permission to Lend................................................................................................................... iii
Acknowledgements............................................................................................................... iv
Abstract................................................................................................................................. v
Table of Contents...................................................................................................................... vi
Table of Figures....................................................................................................................... viii

Chapter 1: Introduction............................................................................................................. 11
  1.1 Introduction...................................................................................................................... 11
  1.2 Objectives....................................................................................................................... 11
  1.3 Possible Uses.................................................................................................................. 11
  1.4 Chapter Guide................................................................................................................ 12

Chapter 2: Background............................................................................................................. 13
  2.1 Non-Photorealistic Rendering......................................................................................... 13
  2.2 Stylisation...................................................................................................................... 14
  2.3 Stoke-based Painterly Rendering.................................................................................... 16
  2.4 Faces............................................................................................................................. 17

Chapter 3: Design..................................................................................................................... 18
  3.1 Techniques..................................................................................................................... 18
  3.2 Defining the Ideal Stylised Face...................................................................................... 23

Chapter 4: Implementation....................................................................................................... 26
  4.1 Tools............................................................................................................................. 26
  4.2 Creation of Style............................................................................................................ 27
  4.3 Optimisation.................................................................................................................. 30
  4.4 User Interface............................................................................................................... 31

Chapter 5: Analysis................................................................................................................ 33
# Table of Figures

Figure 2.1.1: Photorealistic rendering in use in the game “Assassin’s Creed IV: Black Flag” [AC414].............................................................................................................13

Figure 2.1.2: Non-photorealistic rendering used for a technical application [CCE13].....................................................................................................................14

Figure 2.2.1: Cel-Shading in use in “Legend of Zelda: Windwaker” [LZW03]....................................................................................................................14

Figure 2.2.2: Stylisation in use in “The Walking Dead” [TWD12].................14

Figure 2.2.3: A basic pencil sketch stylisation produced by the mobile app “Artist’s Sketch” [QTS12].....................................................................................15

Figure 2.2.4: The paint brush options in MS Paint [MSP14]..............................15

Figure 2.3.1: Examples of textures that can be applied to quads to simulate brush strokes [SHB86].........................................................................................16

Figure 2.3.2: A painterly rendering of an image using an automatic system [LPI97]...........................................................................................................17

Figure 2.4.1: Characters from the television programme “The Simpsons” [SIM14] .................................................................................................................17

Figure 3.1.1: Painterly image created using layering [LSF10].......................18

Figure 3.1.2: An example of a simple painterly filter......................................19

Figure 3.1.3: The weight that is attributed to a point when using Gaussian Smoothing (equation) [SSC01].................................................................19

Figure 3.1.4: The regions of which the average and standard deviations are calculated (for Kuwahara filter).................................................................20

Figure 3.1.5: The output value given by the Kuwahara filter (equation).................................................................20
Figure 3.1.6: Test image after Gaussian smoothing filter and Kuwahara filter
Figure 3.1.7: Original image, and with reduced saturation
Figure 3.1.8: Partial derivatives for Sobel [KDO13]
Figure 3.1.9: Original image, and with edges detected using Sobel operators
Figure 3.1.10: Examples of Haar feature masks [VJR01]
Figure 3.1.11: The Haar feature masks used as part of a cascade to detect a face
Figure 3.2.1: Self Portrait, Vincent Van Gogh
Figure 3.2.2: Original image, successfully stylised and two failed attempts [DDS01]
Figure 4.2.1: Original image, and with Kuwahara filter applied
Figure 4.2.2: A close up of the top of the arch
Figure 4.2.3: Original image, and reduced saturation
Figure 4.2.4: Original image, and with texturising strokes
Figure 4.2.5: Feature enhancing layers
Figure 4.2.6: The results of layering
Figure 5.1.1: Custom style applied to image
Figure 5.1.2: Custom style applied to image of face
Figure 5.1.3: Face detection locates the stylised face
Figure 5.1.4: Style applied to different image
Figure 5.2.1: Face in image located using face detection
Figure 5.2.2: Image stylised leaving face untouched
Figure 5.2.3: Image with Kuwahara filter applied and without
Figure 5.2.4: Saturation reduced as in original style, saturation not reduced, modified saturation reduction

Figure 5.2.5: Medium layer of original style, modified medium layer

Figure 5.2.6: The second fine layer in the original style, and the modified second fine layer

Figure 5.2.7: Original image and the image stylised using the modified parameters

Figure 5.2.8: The stylised face located using edge detection

Figure 5.2.9: Modified style applied to different images

Figure 5.2.10: The original style, and the style with the modified parameters

Figure 5.2.11: Original image, image stylised with modified parameters in face area

Figure 5.3.1: The resulting user interface

Figure 5.4.1: Example 1

Figure 5.4.2: Example 2

Figure 5.4.3: Example 3

Figure 5.4.4: Example 4

Figure 5.5.1: Face in profile not picked up by the face detector

Figure 5.5.2: Original, and resulting image (close up face)

Figure 5.5.3: Original, and resulting image (far away faces)

Figure 5.5.4: Original image, stylisation using default parameter values, stylised with edge detection thresholds increased

Figure 5.6.1: Early stages of working with stylisation in 3D environment: luminance quantisation applied using shaders
Chapter 1: Introduction

1.1 Introduction

In this project, we investigate the use of non-photorealistic rendering, with the intention of developing a program that allows an end user to input an image, and receive a stylised version of that image. The challenge will be to attempt to ensure that any faces in the image maintain their recognisability.

1.2 Objectives

The aim of the project is to investigate the effects of stylisation on faces in comparison to other objects, and how stylising a face affects its recognisability.

In order to reach this main objective, a number of steps must first be achieved:

- creation of custom style,
- application of the style to a face, and investigation into the effects of the style on the face’s recognisability,
- use of different methods to modify the style in an attempt to optimise the recognisability of the face.

1.3 Possible Uses

Existing commercial implementations of stylisation tend to use a manual artist-driven approach. An example of this is the film “A Scanner Darkly”. The film was shot as normal, before a team of artists was employed to go through the video, frame by frame, and manually draw in the stylisation. It was estimated that over 500 hours could be spent animating one minute of film [ASD06]. An automatic system with the ability to stylise a
scene, while retaining the recognisability of characters, could save the game and film industries money and labour-hours.

1.4 Chapter Guide

Here the general content of each chapter will be outlined.

Chapter 2: Background
This chapter will discuss the fundamental ideas on which this project is based, and the methods that are currently in use in the industry.

Chapter 3: Design
This chapter will contain the different techniques that are used during the development of the project.

Chapter 4: Implementation
This chapter will outline in detail the work that has been done on the project.

Chapter 5: Analysis
This chapter will contain the results of the project, and a discussion on these results. It will also contain any limitations of the project, as well as the possibility for future work.

Chapter 6: Summary
Here there will be a review of the outcomes of the project, and a conclusion will be drawn on whether the objectives have been successfully met.
Chapter 2: Background

This chapter will give a brief overview of the research that was done before the project began. It will discuss the ideas on which the project is based, and the current standards in the industry.

2.1 Non-Photorealistic Rendering

The majority of work that has been done in the computer graphics industry to date has been on developing photorealistic rendering (Fig. 2.1.1), i.e. making the graphics appear as realistic and true to life as possible, as if they were a photograph or video.

Figure 2.1.1: Photorealistic rendering in use in the game “Assassin’s Creed IV: Black Flag” [AC414].

However, this is not always the desired effect. Non-photorealistic rendering is used in a variety of different applications. These include stylisation for entertainment purposes such as games and film (see section 2.2), as well as for educational and technical applications [CCE13] (Fig. 2.1.2). Non-photorealistic rendering can be applied to both 3D and 2D scenes.
2.2 Stylisation

One of the major applications of non-photorealistic rendering is for stylisation. Generating graphics in a certain style can aid in creating a theme or atmosphere.

3D Stylisation

Stylisation for 3D applications can be seen in games. Many games aim to immerse the player in their world. Using stylisation can instantly create an atmosphere that draws the player away from the real world and into the fantasy (Figs. 2.2.1 & 2.2.2).
Image-Based Stylisation

Stylisation in a two-dimensional environment can involve replicating the effects of a certain artistic style, e.g. impressionism, sketch drawing, watercolour. A number of programs and mobile apps use this type of stylisation. A number of mobile apps exist that take in a photograph and output a stylised version of the image [QTS12] (Fig. 2.2.3). There are also programs which allow the user to create their own drawings in a painterly style [MSP14] (Fig. 2.2.4).

Figure 2.2.3: A basic pencil sketch stylisation produced by the mobile app “Artist’s Sketch” [QTS12].

Figure 2.2.4: The oil (left, red) and watercolour (right, blue) paint brush options, available in the Windows 8 version of MS Paint [MSP14]. This is an example of stroke-based painterly rendering (see section 2.3).
2.3 Stroke-Based Painterly Rendering

Stroke-based painterly rendering is a widely researched and developed field in image-based stylisation. In order to replicate a realistic artist’s stroke, it must take into account a number of different variables, such as the type of desired brush, medium, and canvas [GGN01]. These combine to produce a highly realistic result.

Brushes

The simulation of realistic brush strokes created by paint brushes is a major challenge, as there are so many different variables that alter the result. The stiffness of the bristles, the density, the pressure applied by the user, and the speed at which the brush is moved all contribute to how the bristles move and the pigment is applied. An attempt to simulate brush strokes can be made by using texture mapped quads, textured with the pattern that reflects the desired result [SHB86] (Fig. 2.3.1).

![Figure 2.3.1: Examples of textures that can be applied to quads to simulate brush strokes [SHB86].](image)

Automatic Systems

A different method must be used for the automatic stylisation of images that already exist, i.e. are not being drawn by the user in the program. The program must not only emulate the texture of the brush strokes, but also the size, position, and orientation, which would have been determined by the artist. One way of doing this is to use edge detection (see section 3.1). In this method, the edges in the image act as the stopping point of the strokes, and the contours in the image determine the orientation of the strokes [LPI97] (Fig. 2.3.2).
2.4 Faces

The importance of faces to the human mind is well documented. It has been shown that infants as young as a few weeks old prefer looking at faces to a range of other stimuli [FPD66]. This behaviour has also been observed in new-borns who had never seen a human face before [GSW75], suggesting that facial recognition is an evolved trait, rather than learned.

Stylisation of Faces

Humans rely on tiny differences to distinguish between different faces. In the process of stylisation, these details can be lost. Because of this, many games, movies, etc. where stylisation is used do not rely on a characters face to distinguish them. Instead, characters may always wear a certain outfit, or have an exaggerated hairstyle (Fig. 2.4.1).

Figure 2.4.1: The characters Homer Simpson and Krusty the Clown from the television programme "The Simpsons". The characters faces alone are indistinguishable, and are instead told apart by their clothes, hair, and makeup.
Chapter 3: Design

This chapter will outline the various techniques that were investigated for use in the project.

3.1 Techniques

This project involves developing a style. To create this style, a number of different techniques that currently exist to stylise an image were investigated.

Layering

Layering is a technique that can be used to attempt to recreate a painterly style using computer graphics [HPR98] [LSF10] (Fig. 3.1.1). The image is rendered separately in different layers, which are then combined to produce the final result. Each layer has an increasing level of detail, so that the first layer may just be blocks of colour, while the final layer may contain the fine details.

It was decided to use layering in the style to be created, in order to build up detail incrementally in a painterly style.

Figure 3.1.1: Painterly image created using layering [LSF10].
Circular Brush Strokes

An almost impressionist painterly style can be recreated roughly by selecting random points in an image and rendering large circular strokes of colour that match the colour of the point (Fig. 3.1.2). This was a technique that was considered for the coarse layer of the created style, however, it gave a messy look to the image when combined with the other layers.

![Figure 3.1.2: An example of this simple painterly filter applied to an image of front arch, Trinity College Dublin.](image)

Smoothing

Smoothing involves blurring the lines between colours in an image, removing fine detail. A number of different smoothing filters can be used.

**Gaussian Smoothing Filter**

A Gaussian smoothing filter is a variation of a local averaging filter, which takes the average of an input pixel and its surrounding pixels as the result. Gaussian smoothing differs from local averaging as the pixel values are weighted (Fig. 3.1.3) so that the pixels closer to the input pixel are more important.

\[
g(x, y) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{d^2}{2\sigma^2}}
\]

*Figure 3.1.3: The weight that is attributed to point (x, y) when using Gaussian smoothing, where d is the distance from the centre point (the input pixel) and σ such that σ^2 is the width of the Gaussian function. [SSC01]*
A Gaussian filter is effective at reducing detail, however it also blurs edges (Fig. 3.16), and so is not optimal for this use.

**Kuwahara Filter**

The Kuwahara filter [KHE76] smooths an image by finding the average and standard deviation of each of four regions of pixels (Fig. 3.1.4) surrounding the input pixel, and takes the output as the average of the region that has the lowest standard deviation (Fig.3.1.5).

\[ \Phi(x, y) = m_i, \text{where } \sigma_i = \min(\sigma_{1,2,3,4}) \]

Figure 3.1.4: The four regions of which the average and standard deviations are calculated. Some pixels belong to more than one region.

Figure 3.1.5: The output value given by the Kuwahara filter, where m is the mean, and i is one of the four regions shown in fig.3.1.4.

The Kuwahara filter is an edge preserving smooth, so it reduces fine detail, but major features are maintained (Fig. 3.1.6). It was chosen to create the coarse layer for the created style.

Fig. 3.1.6: Test image (left) after Gaussian smoothing filter (centre), and Kuwahara filter (right).
**Saturation**

Even with the smoothing filters applied, the coarse layer can attract too much attention away from the finer details that will be added later, reducing the effect of the stylisation. To reduce emphasis on the background, the saturation of the colours can be reduced (Fig. 3.1.7).

![Figure 3.1.7: Original image (left), with reduced saturation (right).](image)

**Edge Detection**

Detecting the edges in an image can be useful for emphasising edges and details, and so is used to create the finer layers of the created style.

**Sobel Edge Detection**

Sobel (Fig. 3.1.8) is a differential edge detector [CED14] that can be used to detect edges, as well as the orientation of the edges (Fig. 3.1.9).

\[
g_x = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad g_y = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}
\]

![Figure 3.1.8: Horizontal and vertical partial derivatives for Sobel [KDO13].](image)
Face Detection

Face detection enables the location of any faces in an image to be located, if present. This can be done using a cascade of Haar classifiers [VJR01].

Haar feature masks [Fig. 3.1.10] are used to detect patterns of light and dark in a greyscale image.

Cascades of classifiers are created by comparing a number of feature masks with images of positive and negative samples; in this case, images containing faces and images containing no faces, respectfully. In this way the cascade of classifiers is trained with the feature masks that correspond to images of faces, and so is able to compare these to new images in order to detect faces [Fig. 3.1.11].
Cascades that are trained to recognise full frontal faces are available as part of OpenCV libraries, so do not need to be created from scratch.

### 3.2 Defining the Ideal Stylised Face

Before the program can be built, the ideal goal must be defined; in this case, the ideal stylised face. The quality of stylisation can be subjective, so an attempt must be made to define the ideal result quantitatively, in order to determine the success or failure of the program.

The ideal stylised face:

- must be recognisable as a face,
- must be recognisable as that specific face,
- the stylisation of the face must be in line with the rest of the image,
- the stylisation of the rest of the image must not be sacrificed.

The reasons these four requirements were decided upon are as follows:

**Recognisable as Face**

It has been argued that before a face can be analysed by the brain as such, it must first be classified as a face rather than any other object [ETA81]. To illustrate this, people suffering from the disease prosopagnosia, which prevents the sufferer from recognising faces, are still able to distinguish faces from other objects [JHR84]. This is why it was decided to split the recognisability as a face and the recognisability as a specific face into two separate requirements (Fig. 3.2.2 bottom right).

**Recognisable as Specific Face**

As discussed in section 2.2, stylisation is commonly used in games and film. In these applications, as part of the narrative, it is important for characters to be recognisable and distinguishable from each other.
**In line with rest of image**

If the stylisation is altered for the face, it is essential that it still fits in with the style of the image (Fig. 3.2.1) in order for the image to fit together cohesively.

![Self Portrait, Vincent Van Gogh. Note how although the size of the strokes have been altered in the face area, they retain the style of the strokes in the rest of the image.](image)

**Stylisation not sacrificed**

Although it is inevitable that the stylisation must be modified if it renders faces unrecognisable, it is important that the overall stylisation of the image is not sacrificed (Fig. 3.2.2, bottom left).
Figure 3.2.2: The original image (top left) successfully stylised (top right) and two failed attempts. The bottom left image was determined a failure, as the level of detail in the background of the image sacrificed the stylisation. The bottom right image was determined a failure as there is not enough detail for the face to be recognisable as such [DSS01].
Chapter 4: Implementation

This chapter will outline in detail the work that has been done on the project.

4.1 Tools

Here the main development tools that were used to create the project are listed.

Processing

Processing [PRO14] is an open source IDE (integrated development environment) that uses Java, and is commonly used for image processing development. It was originally intended to begin building the program using Processing and then move to Microsoft Visual Studios, using C++, but it was decided while working with Processing that it was a more suitable platform, as it has many available libraries and is relatively easy to debug, meaning that less time would be wasted and more time could be spent working on the core aspects of the project.

OpenCV

OpenCV [OCV14] is an open source computer vision library. It has over 2500 in-built algorithms, however the version used by this program has been created for use with Processing and is less extensive. This is acceptable as OpenCV is only used in this program for facial recognition, which is still available in the smaller version.

ControlP5

ControlP5 [CP514] is a GUI (graphical user interface) library for Processing. It has in-built functions to produce a variety of different controllers. In this project it is used to create the sliders for the user interface.
4.2 Creation of Style

In order to investigate the effects of stylisation on a face, a style must be chosen. For the purposes of this project, a custom method of stylisation was developed, to act as a case study.

This style was created in a number of steps:

- Smoothing
- Saturation
- Edge Detection – Texturising and Feature Enhancing
- Layering

**Smoothing**

As outlined in section 3.1, the smoothing kernel that was chosen for the style was the Kuwahara filter, which is an edge preserving smooth that blurs textures and fine details, but retains an objects shape and main features (Fig. 4.2.1).

*Figure 4.2.1: Original (left) and with Kuwahara filter applied (right).*
Saturation

The saturation of the colours in the smoothed image are reduced (Fig. 4.2.3). This fades the colours while retaining their hue, to reduce emphasis.

Edge Detection

Sobel edge detection is used to detect edges in the image, and the orientation of these edges. This is used in two different methods.

Texturising

To add texture to the image, a very basic stroke-based filter (see section 2.3) was applied to detected edges (Fig. 4.2.4). This adds texture around the edges and features of objects, giving a more painterly feel. A number of different combinations of stroke widths, lengths, and densities were experimented with on a variety of images. The chosen parameters were those that gave an optimal result for the most images. Of course, as this is the creation of a style, this is subjective.
Feature Enhancing

Features are enhanced in two different layers.

The first layer uses a higher edge detection threshold to find edges than the texturising stage. This means that only more pronounced edges are recognised. This layer consists of the edges as are, in the colour of the original image, giving definition (Fig. 4.2.5).

The second layer uses a higher edge detection threshold than the first layer. On this layer the edges are coloured black, emphasising the most pronounced edges in the image (Fig. 4.2.4).

Layering

To resulting image is assembled in layers (Fig. 4.2.5).
Coarse Layer

The coarse layer acts as a base for the final image. It contains the background colours and shapes, but only low levels of detail. In this example the coarse layer consists of the original image with the Kuwahara smoothing kernel applied, and the saturation levels of the colours reduced.

Medium Layer

The medium layer adds broader detail to the image, but not the fine details. In this example the medium layer consists of the texturised edges.

Fine Layers

The fine layers add definition and fine details. In this example the fine layers consist of the two feature enhancing layers, the colour and the black.

![Figure 4.2.6: The result of layering. Original image (top left), coarse layer (top centre), medium layer (top right), first fine layer (bottom left), second fine layer (bottom centre), resulting image (bottom right).]

4.3 Optimisation

To attempt to create the ideal stylised face, the style is modified in three different methods.
Leave face untouched

The first method tried is to stylise the background of the image, but not the face. In order to locate the face in the image, face detection is used, as explained in section 3.1. This area is left untouched while the parameters of stylisation are applied to the rest of the image.

Modify Parameters

The second method is to identify the parameters of the style that are the most responsible for altering the recognisability of the face, and to modify them in an attempt to minimise this. These modifications will be discussed in section 5.2.

Modify Parameters only for Face

The third method is to apply the modifications made in the second method, but only to the face area. As in the first method, the face in the image is located using face detection. This area is stylised according to the new, modified parameters, while the rest of the image is stylised as before.

4.4 User Interface

An effort has been made to define the ideal result of the program (see section 3.2). However, the quality of stylisation is a subjective matter, and so what may be defined as ideal may not be the desired result of the user. In an attempt to combat this, a graphical user interface (GUI) is developed, that allows the user to modify the parameters of stylisation.

The parameters that the GUI allows the user to modify are:

- whether or not the Kuwahara filter is applied,
- how much the saturation of the colours are reduced,
- the width and length of the strokes in the medium layer,
- the width of the lines in both the fine layers,
- the edge detection thresholds of the medium and fine layers.
The GUI also allows the user to view the layers individually as well as combined. It also allows the user to modify whether or not the style will be optimised to enhance the recognisability of any faces in the image input.
Chapter 5: Analysis

This chapter will contain the results of the work that was done in chapter 4. The final program will be discussed, with any limitations and possibilities for the future.

5.1 Results: Creation & Investigation

Creation

The results of the work carried out in section 4.2 is presented in this section (Fig. 5.1.1).

Investigation

With the style developed, the next objective was to apply it to an image of a face (Fig. 5.1.2), and investigate its effect on the recognisability.
The quality of the stylised face was analysed, to determine whether an ideal stylised face had been produced (as defined in section 3.2), with the results as follows.

**Recognisable as Face**

The shape of the face has remained unchanged, and the major features (eyes, nose, and mouth) are visible.

There is a positive result from face detection (Fig. 5.1.3).

![Figure 5.1.3: Face detection is able to locate the stylised face.](image)

**Recognisable as Specific Face**

The style was applied to a number of different images of faces (Fig. 5.1.4). Although all the major features of the faces are visible, any detail has been obscured by the thick black lines. Without this detail, it is difficult to distinguish one person’s face from another.

![Figure 5.1.4: Style applied to different images.](image)

**In line with rest of image**

The stylisation that was applied is uniform over the entire image, so there is no difference between the style of the face and the rest of the image.
Stylisation not Sacrificed

The stylisation has not been modified from how it was originally designed, so has not been sacrificed.

The requirements for an ideal stylised face were not met, as the face is not recognisable as a specific face.

5.2 Results: Optimisation

The next objective was to investigate different methods of modifying the stylisation, in an attempt to create the ideal stylised face (see section 4.4). The stylised faces created by these images are compared to the ideal, as was done in section 5.1.

Leave face untouched

First, the face was located in the image using face detection (Fig. 5.2.1), and this area of the image was left as is while the rest of the image was stylised as before (Fig. 5.2.2).

The face is analysed as before.

Recognisable as Face

The face is unchanged from the original image, and so is recognisable as a face.
Recognisable as Specific Face

Again, the face is unchanged and so is recognisable as that specific face.

In line with rest of image

The background of the image is stylised, while the face is a photograph. There is no cohesiveness between the two.

Stylisation not sacrificed

The stylisation in the background has not been changed from the original design, so has not been sacrificed.

The requirements for an ideal stylised face were not met, as the face area is not in line with the stylisation of the rest of the image.

Modify Parameters

The parameters of the style were investigated to determine which were most responsible for the reduced recognisability of the face, and what changes could be made to minimise this.

Smoothing

The Kuwahara filter removes fine detail (Fig. 5.2.3). However it is used in the coarse layer, where it is not required or desired to have much detail. For this reason it was decided to leave this parameter as is.

Figure 5.2.3: Image with Kuwahara filter applied (left) and without (right).
**Saturation**

Reducing the saturation of the coarse layer keeps the emphasis away from the background and towards the more detailed layers. However, altering the colour alters the skin tone of the face. To enhance recognisability by retaining some of the face’s skin tone, but also maintain the effect of reduced saturation, the saturation reduction is lowered (Fig. 5.2.4).

![Saturation comparison](image)

Figure 5.2.4: Saturation reduced as in original style (left), saturation not reduced (centre), modified saturation reduction (right).

**Edge Detection – Texturising**

Due to the amount of detail in the face, the edge detector picks up a lot of edges in a small area. This results in there being a high density of strokes around the face, creating a messy look and obscuring the features. By reducing the width and length of the strokes (Fig. 5.2.5), this can be minimised.

![Edge detection comparison](image)

Figure 5.2.5: Medium layer of original style (left), modified medium layer (right).

**Edge Detection – Feature Enhancing**

A major problem of the stylised face was the black lines on the face blocked all of the detail which is necessary for it to be recognisable. By reducing the thickness of these lines (Fig. 5.2.6), the features are still highlighted, but not obscured.
Figure 5.2.6: The second fine layer in the original style (left), the modified second fine layer (right).

**Result**

The modified layers are combined and the result (Fig. 5.2.7) is analysed as before.

Figure 5.2.7: Original image (left) and the image stylised using the modified parameters (right).

**Recognisable as Face**

The shape and features of the face are clearly visible. It is located correctly using face detection (Fig. 5.2.8).

**Recognisable as Specific Face**

The style was applied to a number of different images of faces (Fig. 5.2.9). The features of the face have maintained enough recognisability in order for the identity of the faces to be distinguished from one another.
Figure 5.2.8: The stylised face can be located using face detection.

**In line with rest of image**

The modified style has been applied to the whole image, so the face area is in line with the background.

**Stylisation not Sacrificed**

The alterations to the stylisation parameters has affected the style noticeably (Fig. 5.2.9).

The requirements for an ideal stylised face were not met, as the stylisation of the image has been sacrificed.

**Modify Parameters only for Face**

The last optimisation method attempted was to apply the modifications of the previous section only to the face area (Fig. 5.2.10). The result is analysed as before.
Recognisable as Face

As the face area is identical to that of the previous method, it is also recognisable as a face.

Recognisable as Specific Face

Again, as the face area is identical to the previous method, it is still recognisable as the specific face, as shown before.

In line with rest of image

Although the parameters of stylisation in the face area and the rest of the image are different, the same methods are applied, so there is no noticeable transition.

Stylisation not Sacrificed

The background of the image remains in the original style, so the stylisation has not been sacrificed.

All four requirements for an ideal stylised face has been met, so it is chosen that this is the method that is used for optimising the style in the program.

5.3 Results: GUI

The GUI (Fig. 5.3.1) is modified to allow the user to optimise the style for the face area, as well modify the parameters for only that area.
The default parameter values are those that have been found through experimentation to produce the best results for the majority of images, and are the values used for the images in this report.

Figure 5.3.1: The resulting user interface. Shows original image (left), stylised result (right), and allows the user to modify the stylisation parameters for the result using the sliders.

**Slider Functions**

The far left column alters which layers are combined for the resulting image.

- “Background?” – Toggles on and off the background image; only causes visual change if next two sliders are in off position.
- “Kuwahara?” – Toggles on an off Kuwahara filtering on the coarse layer.
- “Saturation?” – Toggles on and off saturation reduction of the coarse layer.
- “Texture?” – Toggles on and off the medium layer.
- “Definition?” – Toggles on and off the first fine layer.
- “Black?” – Toggles on and off the second fine layer.

The second column from the left alters the edge detection thresholds used.

- “Texture Threshold” – Alters the edge detection threshold for the medium layer.
- “Definition Threshold” – Alters the edge detection threshold for the first fine layer.
- “Black Threshold” – Alters the edge detection threshold for the second fine layer.
The third column from the left alters the other parameters of the stylisation.

- “Saturation Threshold” – Alters the amount the saturation of the colours of the coarse layer are reduced.
- “Texture Length” – Alters the length of the strokes in the medium layer.
- “Texture Width” – Alters the width of the strokes in the medium layer.
- “Definition Size” – Alters the width of the lines in the first fine layer.
- “Black Size” – Alters the width of the lines in the second fine layer.

The fourth column from the left alters whether the stylisation is modified for the face area and which parameters.

- “Show Faces?” – Toggles on and off the face detector, which overlays a green box over any detected faces in the original image (Fig. 5.2.1).
- “Face Saturation?” – Toggles on and off the modified saturation reduction in the face area.
- “Face Texture?” – Toggles on and off the modified medium for the face area.
- “Face Black?” – Toggles on and off the modified fine layer for the face area.

The far right column alters the parameters of stylisation for the face area. These will only cause a visual change if the corresponding slider in the previous column is set to on.

- “Face Saturation Reduction” – Alters the amount the saturation of the colours of the coarse layer are reduced for the face area.
- “Face Texture Length” – Alters the length of the strokes in the medium layer for the face area.
- “Face Texture Width” – Alters the width of the strokes in the medium layer for the face area.
- “Face Black Size” – Alters the width of the lines in the second fine layer for the face area.
5.4 More Examples

Figure 5.4.1

Figure 5.4.2

Figure 5.4.3
5.5 Limitations

The program has a number of limitations.

**Subjectivity of Quality of Stylisation**

The success of stylisation depends greatly on whether the user finds the result aesthetically pleasing. This varies from person to person, and it is impossible to produce a result that is desirable for all.

The user interface was developed to minimise this limitation.

**Limited Database of Faces**

A large database of faces was not available, so it is unknown how the program will react to a range of varying skin tones, etc.

**Limitations of Face Detection**

The facial detection library used only works for faces that are fully facing the camera. Any other faces will not be picked up by the program, and so the parameters of stylisation will not be optimised (Fig. 5.5.1).
Figure 5.5.1: Faces in profile are not picked up by the face detector.

**Face Size**

The program works best on faces a moderate distance from the camera. Large and close-up faces do not have a satisfactory level of stylisation (Fig. 5.5.2), and far away or low quality faces become blurry and unrecognisable (Fig. 5.5.3).

Figure 5.5.2: Original (left) and resulting image (right). The image is not noticeably stylised.

Figure 5.5.3: Original (left) and resulting image (right). The already low level of detail in the faces left few recognisable features after stylisation had been applied.
**Irregular Objects**

Due to the edge detection used, natural objects which have a high number of irregular edges are not optimally stylised.

The user can minimise this problem by manually increasing the edge detection thresholds in the user interface (Fig. 5.5.4).

![Figure 5.5.4: Original image (left), stylisation using default parameter values (centre), stylised with edge detection thresholds increased (right).](image)

### 5.6 Future Work

There is a huge amount of scope for the future work that could be done to build upon the work done in this project.

**Apply Ideas to Other Styles**

For the purpose of this project, a custom style was developed from scratch, to aid in the understanding of stylisation. This style was used as a case study to investigate the effects of stylisation on faces. In the future, the techniques learned in this project could also be used to optimise other styles, in order to minimise their effects on a faces recognisability.

**User Interface Improvement**

The user interface was originally designed to assist with the investigation portion of the project, to enable the parameters to be quickly and easily modified to investigate how these changes affected the image. In order to make the program into an application for an end user, the user interface would have to be modified. Changes that could be made include incorporating an image loader, creating a tutorial, and including a glossary of terms to explain the parameters and the functions of the sliders, similar to that in section 5.3.
Optimise Performance

If the speed of the program was improved upon, the program could run in real-time. This would enable the program process videos as well as images.

3D Objects

Originally, it was intended to apply the methods in this project to both 2D and 3D environments (Fig. 5.6.1), however it was decided due to time constraints to concentrate on an image-based approach. In the future, the methods used could be altered to develop a program that would be able to stylise a 3D environment, as originally planned.

Figure 5.6.1: Early stages of working with stylisation in 3D environment: luminance quantisation applied using shaders.
Chapter 6: Summary

6.1 Review

In this project a custom method of stylisation was created, the effects of the style on the recognisability of a face was investigated, and the stylisation was optimised to minimise any reduced recognisability. The result is a program that takes in an input image from a user and outputs a stylised version of that image, optimised to enhance the recognisability of any faces in the image. The program also allows the user to modify the style to appeal to their own aesthetics.

6.2 Conclusion

It has become clear through the course of this project that when it comes to stylisation, faces cannot be treated the same as any other object. They are a unique entity which must be treated as such, to ensure that they remain recognisable.

The entertainment industries are growing, and with that, the demand for a reliable method for the stylisation of recognisable characters is high. The field is continuously developing, and not only requires a thorough knowledge in the fields of non-photorealistic rendering and stylisation, but also in how the human brain works, and what it is that makes a face unique. It is a fascinated area with endless possibilities for the future.
Appendix: CD Contents

The CD contains this report in PDF form, the program, and the Processing libraries used.

- Report.pdf

- Project
  - fyp_stylisation_faces
    - fyp_stylisation_faces - code
    - data
      - image1-11 - images used in program

- Processing libraries
  - ControlP5 - library used for gui
  - opencv - opencv library
  - opencv_processing - opencv optimised for Processing
Bibliography

http://assassinscreed.ubi.com/en-gb/home/
Last checked 30 March 2014

[ASD06] Interview with Richard Linklater, director of “A Scanner Darkly”.
http://www.awn.com/animationworld/scanner-darkly-animated-illusion
Last checked 31 March 2014


Last checked 1 April 2014

[CP514] Control P5 Processing Library.
http://www.sojamo.de/libraries/controlP5/
Last checked 30 March 2014

[DSS02] D. DeCarlo, A. Santella. Stylisation and Abstraction of Photographs. (Siggraph ’02)
https://www.cs.rutgers.edu/~decarlo/pubs/sg02.pdf
Last checked 30 March 2014


ISBN 1-56881-133-0


[HPR98] A. Hertzmann. Painterly Rendering with Curved Brush Strokes of Multiple Sizes. (Siggraph '98)
http://www.mrl.nyu.edu/publications/painterly98/hertzmann-siggraph98.pdf
Last checked 30 March 2014

[JHR84] M.A. Jeeves. The historical roots and recurring issues of neurobiological studies of face perception.

[KDO13] K. Dawson-Howe. A Practical Introduction to Computer Vision with OpenCV2 (due to be published next year, advance copy given to class to aid with lectures).


[LPI97] P. Litwinowicz. Processing Images and Video for an Impressionist Effect. (Siggraph '97)

Last checked 30 March 2014
http://www.zelda.com/windwaker/
Last checked 30 March 2014

Last checked 30 March 2014

[OCV14] OpenCV.
http://opencv.org/
Last checked 30 March 2014

[QTS12] Quinqi Tong. Artist’s Sketch.
http://appplot.com/apps/detail/529049274/artists-sketch-free
Last checked 30 March 2014

[PRO14] Processing.
http://www.processing.org/
Last checked 30 March 2014

http://www.thesimpsons.com/
Last checked 30 March 2014

Last checked 30 March 2014
ISBN 978-0130307965

https://www.telltalegames.com/walkingdead/
Last checked 30 March 2014

https://www.cs.cmu.edu/~efros/courses/LBMV07/Papers/viola-cvpr-01.pdf
Last checked 30 March 2014