UNIVERSITY OF LINCOLN

SCHOOL OF MATHEMATICS AND PHYSICS

Classification of Biological Organisms from Images Using Advanced Mathematical Techniques

Timothy Smith

25796944

Supervised by Dr Danilo Roccatano

March 5, 2024

Scientific Report in the 3rd Year Module MTH3009 Mathematics Project

Abstract

Contents

1 Introduction

2 The Elliptic Fourier Transform

The Fourier transform is a way of finding constituent frequencies within a function on the real domain. The Fourier transform extends the concept of the Fourier series (which openerates on a bounded interval) to the real domain.

2.1 The Fourier Series

The Fourier series (1) lets you take a single repeating unit of a periodic function and use this to generate an infinite sum of sinusoidal functions that converge to it. This is useful because trigonmetric functions have mathematical properties that not all functions posess, allowing you to work with them nicely in more situations.

$$s(x) \sim A_0 + \sum_{n=1}^{\infty}\left(A_n \cos\left(\frac{2\pi n x}{P}\right) + B_n \sin\left(\frac{2\pi n x}{P}\right)\right) \tag{1}$$

The $\sim$ symbol indicates that the series doesn't strictly converge in all cases.

Firstly we have a set of numbers, A_0, A_n and B_n:

$$\begin{aligned} A_0 &= \frac{1}{P}\int_{-\frac{P}{2}}^{\frac{P}{2}} s(x)\,dx \\ A_n &= \frac{1}{P}\int_{-\frac{P}{2}}^{\frac{P}{2}} s(x)\cos\left(\frac{2\pi n x}{P}\right)dx \\ A_b &= \frac{1}{P}\int_{-\frac{P}{2}}^{\frac{P}{2}} s(x)\sin\left(\frac{2\pi n x}{P}\right)dx \end{aligned} \tag{2}$$

Here A_0 is a constant, and A_n and B_n are functions of n. In fact, since both the denominator of the fraction and the range of the integral for A_0 are the same, P, A_0 is simply the average around which the function oscillates.

2.2 The Fourier Transform

The transform function ($S(t)$) for frequency f is given by (3).

$$S(t) = \int_{-\infty}^{\infty} s(t)\cdot \mathrm{e}^{-i2\pi f t}\,dt \tag{3}$$

2.3 Fourier Analysis

2.4 Elliptical Fourier Analysis

The seminal paper on this topic is [1].

3 Processing the Images

In order to find shapes in our images for the purposes of analysis, we will need to detect edges. For this we will use the Canny Edge detection method, developed by John F. Canny.

3.1 Convolution Kernels

The most basic aspect of edge detection is the convolution kernel. This is an $M \times M : \{M = 2n+1, n \in \mathbb{N}\}$ matrix (an odd sided square matrix). Odd side lengths allow the kernel to be centered at each pixel.

If we let the function $f(x, y)$ represent the original image, $g(x, y)$ represent the convolved image and ω represent the kernel, then:

$$g(x,y) = \sum_{i=-n}^{n} \sum_{j=-n}^{n} \omega(i,j) f(x-i, y-j) \tag{4}$$

Notice that the coordinates of the kernel are not 1 to M as with a traditional matrix, but rather $-n$ to n.

The effect of this on an image will be to make each pixel of a convolved image a function of the surrounding pixels. The simplest convolution matrix is the identity convolution matrix, (5).

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{5}$$

More examples include edge detection kernels like (6) and (7).

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \tag{6}$$

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} \tag{7}$$

3.2 Canny Edge Detection

4 Conclusion

5 Acknowledgements

6 Appendix

Note: These are unlikely to be in the final report, they are simply there as an example appendix for now.

main.cpp

```cpp
// Credit:
// https://gnome.pages.gitlab.gnome.org/gtkmm-documentation/sec-helloworld.html
#include "window.h"
#include <gtkmm/application.h>

int main(int argc, char *argv[])
{
	auto app = Gtk::Application::create("com.thatchapthere.elliptic.fourier");

	return app->make_window_and_run<Window>(argc, argv);
}
```

elliptic_fourier.h

elliptic_fourier.cpp

drawing_area.h

drawing_area.cpp

7 Bibliography

References

[1] F. P. Kuhl and C. R. Giardina, "Elliptic fourier features of a closed contour," *Computer Graphics and Image Processing*, vol. 18, no. 3, pp. 236–258, 1982.