Stellenbosch University Faculty of Engineering

Image Processing Tools

May 2004

Riaan van den Dool

Fourier-Mellin Transform

1 INTRODUCTION

The purpose of this document is to examine the theory of the Fourier-Mellin Transform as an Image Processing Tool and to look at some applications as well as relevance to the thesis proposed by the author.

2 THEORY

Overview

The Fourier-Mellin transform is a useful mathematical tool for image recognition because its resulting spectrum is invariant in rotation, translation and scale. The Fourier Transform itself (FT) is translation invariant and its conversion to log-polar coordinates converts the scale and rotation differences to vertical and horizontal offsets that can be measured. A second FFT, called the Mellin transform (MT) gives a transform-space image that is invariant to translation, rotation and scale.

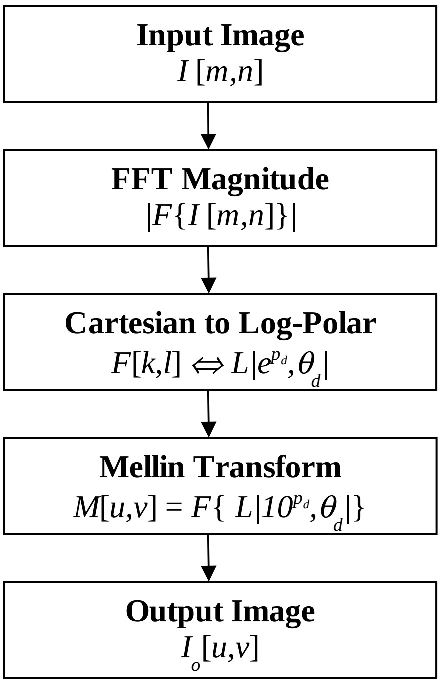

Figure 1: Block diagram of the Fourier-Mellin Transform

The Fourier Transform

The Discrete Fourier Transform (DFT) is given by the following expression:

$$\left|F\{I[m,n]\}\right| = \left|F[k,l]\right| = \left|\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} I[m,n]\, e^{-j\left(2\pi/M\right)(km)}\, e^{-j\left(2\pi/N\right)(ln)}\right| \qquad (1)$$

This is often computed using the Fast Fourier Transform algorithm to speed up the time needed for the calculation.

Cartesian to Log-Polar conversion

The FFT is projected onto the log-polar plane by the coordinate transform shown below.

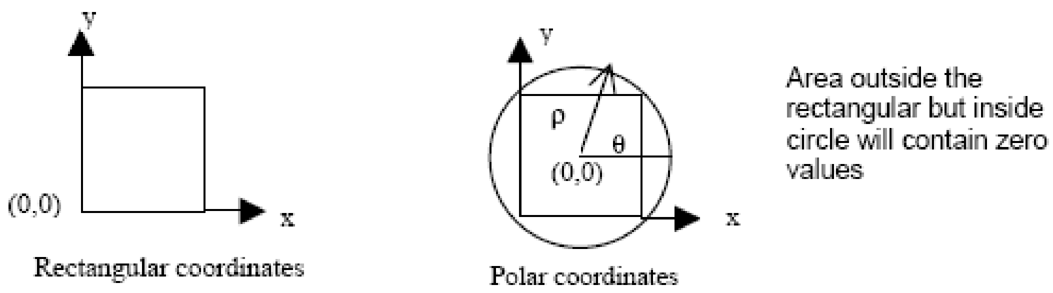

Fig 1. Transformation from rectangular to polar coordinates

For the conversion from Cartesian coordinates to Log-Polar coordinates the following equation is true:

$$r = \sqrt{x^2 + y^2} \qquad (2)$$

The origin (m_o, n_o) should be at the center of the image matrix to ensure the maximum number of pixels is included. If the image consists of a square N x N matrix then the coordinates of the origin are:

$$\begin{aligned} m_0 &= N/2 \;\; ; \;\; n_0 = N/2 && \text{for } N \text{ odd} \\ m_0 &= (N-1)/2 \;\; ; \;\; n_0 = (N-1)/2 && \text{for } N \text{ even} \end{aligned} \qquad (3)$$

The maximum sampling radius for the conversion can now be calculated as:

$$\rho_{max} = min(m_o, n_o) \quad \text{…inscribed circle}$$

$$\rho_{max} = sqrt(m_o^2 + n_o^2) \quad \text{…circumscribed circle} \qquad (4)$$

If an inscribed circle is chosen as the conversion boundary, some pixels that lie outside the circle will be ignored. If a circumscribed circle is chosen, all pixels will be taken in account, but some invalid pixels will also be included (pixels falling inside the circle but outside of the image matrix).

Since the pixels in Cartesian coordinates cannot be mapped one-to-one onto pixels in the Log-Polar coordinate space, an average of the surrounding pixels needs to be calculated. The standard methods to do this includes nearest neighbor, bilinear and bicubic resampling.

The relationship between the polar coordinates (ρ, θ) used to sample the input image and the polar coordinates of the log-polar image (r, θ) can be described by:

$$(\rho, \theta) = (e^{r}, \theta) \tag{5}$$

To map the input image pixels $image_{in}(x_i, y_i)$ onto the output image pixels $image_{out}(r_m, \theta_n)$ the coordinates x_i, y_i are computed using:

$$x_i = round(\rho_m * cos(\theta_n) + m_o)$$
$$y_j = round(\rho_m * sin(\theta_n) + n_o) \tag{6}$$

where $(\rho_m, \theta_n) = (e^{r_m}, \theta_n)$ according to (5). $Image_{in}$ is an i x j matrix and $image_{out}$ is a m x n matrix.

```
function output = logpolar(input)
    oRows = size(input, 1);
    oCols = size(input, 2);
    dTheta = 2*pi / oCols;                  % the step size for theta
    b = 10 ^ (log10(oRows) / oRows);        % base for the log-polar conversion
    for i = 1:oRows                         % rows
        for j = 1:oCols                     % columns
            r = b ^ i - 1;                  % the log-polar
            theta = j * dTheta;
            x = round(r * cos(theta) + size(input,2) / 2);
            y = round(r * sin(theta) + size(input,1) / 2);
            if (x>0) & (y>0) & (x<size(input,2)) & (y<size(input,1))
                output(i,j) = input(y,x);
            end
        end
    end
```

Fig 2. A Matlab implementation of the log-polar conversion

The images below show a normal and rotated version of a white box on a black background with its 2D FFT in the Cartesian and Log-Polar planes. Note the horizontal shift to the right of the second Log-Polar image compared to the first. This shift indicated the rotation between the two images.

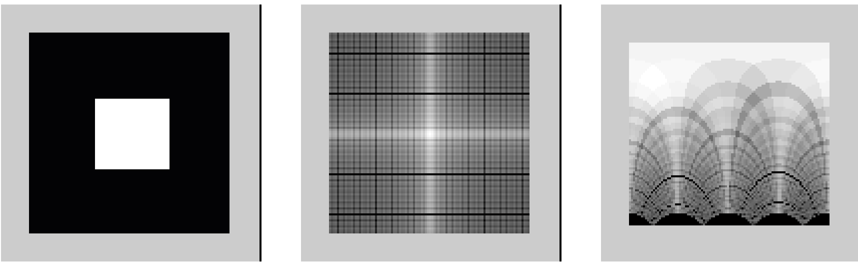

Fig 2. Original image, FFT in Cartesian coordinates, FFT in Log-polar coordinates

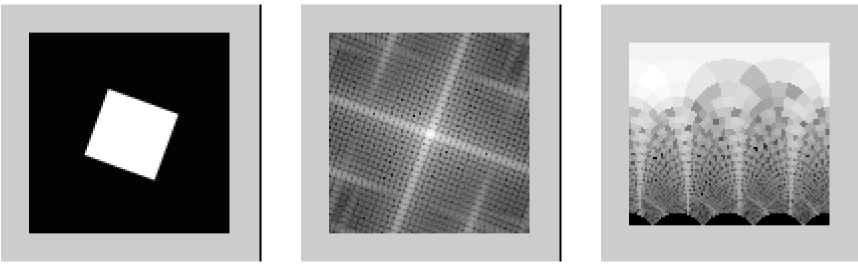

Fig 3. Rotated image, FFT in Cartesian coordinates, FFT in Log-polar coordinates

The Mellin Transform

The next step is to get a transform-space image that is a rotation and scale invariant representation of the original image. This is done using the Mellin transform which can be expressed as:

$$M(u,v) = \int_0^\infty \int_0^\infty f(x,y) x^{-ju-1} y^{-jv-1} dxdy \quad ; \quad \forall\, x,y \rangle 0 \tag{7}$$

Converting to polar coordinates using (2) we have:

$$M\{f(r)\} = \int_0^\infty f(r) r^{-ju-1} dr \tag{8}$$

and making $r = e^{\gamma}$, $dr = e^{\gamma} d\gamma$:

$$M\{f(e^{\gamma})\} = \int_{-\infty}^{\infty} f(e^{\gamma}) e^{-ju\gamma} d\gamma \qquad (9)$$

This is clearly a Fourier Transform. If we make a change in coordinates from Cartesian to a Log-Polar system, we can directly perform a DFT over the image to obtain the scale and rotation invariant representation.

3 APPLICATIONS

The Fourier-Mellin transform has been used for image recognition purposes [3] and the authors claim to be able to recognize a human face by comparing the obtained spectrum with a database of previously calculated spectra.

Another common use of the Fourier-Mellin transform is in detecting watermarks in images regardless of scale or rotation.

4 EXAMPLE

A good example where Fourier-Mellin is used to register two images can be found at http://www.mathworks.nl/matlabcentral/fileexchange/loadFile.do?objectId=3000. The Matlab source code can be downloaded and contains a useful implementation of the FM transform.

5 RELEVANCE TO THESIS

The FM transform might be used as a method to compare the extracted GCP chips. It could also be useful for removing the rotation and scale differences between the chips. In a test done by the author the rotation could be extracted from the log-polar representations of a test image.

6 REFERENCES

[1] Mouton, C.J. : “Processing of onboard images to assist automatic forward motion compensation for micro-satellites”, University of Stellenbosch, April 2003, p29 - 36

[2] Lin, Wu, Bloom, Cox, Miller, Lui : “Rotation, Scale and Translation Resilient Watermarking for Images”, IEEE transactions on Image Processing, Vol. 10, No. 5, May 2001, pp. 767-782

[3] R. Moller, H. Salguero, E. Salguero: “Image recognition using the Fourier-Mellin Transform”, LIPSE-SEPI-ESIME-IPN, Mexico