# Estimating Bulk Geometrical Properties of Cellular Structures

by

Shannon M. Puddister

A thesis

presented to the University of Waterloo

in fulfilment of the

thesis requirement for the degree of

Master of Applied Science

in

Systems Design Engineering

Waterloo, Ontario, Canada, 2003

The University of Waterloo requires the signatures of all persons using or photocopying this thesis. Please sign below, and give address and date.

# Acknowledgements

I would like to thank my family for their support, and more importantly, for putting up with my continued absence from the “real world”.

I would also like to thank my supervisors, David A. Clausi and G. Wayne Brodland, without whom none of this work would have been possible.

# Contents

# List of Tables

# List of Figures

# Chapter 1

# Introduction

## 1.1 Motivation

Automatically extracting meaningful bulk shape and orientation statistics from a stationary image can provide valuable information. For example, cell shape has been shown to be an important indicator of stresses in an epithelium [6]. Another example is the estimation of ice floe size in synthetic aperture radar (SAR) sea ice images, in which floe size provides important information for assisting ship navigation in ice-infested waters [13]. Each object in imagery from these two examples has its own geometrical properties (length, width, orientation). Taken together, they can be used to describe a general characterization of all objects in the image. Extracting these metrics, however, is a difficult task for a computer algorithm. The object of this research is generate automated methods to produce such estimates.

## 1.2 Thesis Summary

In this thesis, many of the issues involved with applying an automated bulk statistic estimation algorithm to the power spectrum of an image will be discussed. The following outlined will be used:

- Chapter 2 - Background

  A description of the important biological issues that motivated much of this work are discussed. The problem being considered is then discussed followed by a more formal problem definition.

- Chapter 3 - Image Data

  Several different types of images will be analyzed. Each type has unique noise processes and structural issues associated with it. In this chapter, we discuss those processes and introduce the data sets that will be inspected.

- Chapter 4 - Methods

  The bulk statistics of several image sets will be estimated by four algorithms. In Chapter 4, each of the algorithms are discussed in detail.

- Chapter 5 - Testing

  Chapter 5 presents the results from applying the four algorithms to each of the data sets. The results will be compared and inherent strength and weaknesses of each algorithm will be discussed.

- Chapter 6 - Conclusion

The thesis is concluded by elaborating on the results, stating some of the limitations facing this work and providing direction for future work in estimating bulk statistics.

# Chapter 2

# Background

## 2.1 Biological Considerations

The development of an explanation for embryonic development is far from complete. An exhaustive mathematical description will require drawing from a variety of disciplines such as biology, mechanics and image processing as we further our understanding.

What we do know is that as embryos develop, they undergo precise and self-driven changes of shape [5]. Acting as both muscle and skeleton, the cytoskeleton and other structures are commonly recognized to drive these movements [14]. An understanding of the exact causes and details of these movements is still in its formative stages.

## 2.2 Embryonic Modelling

Mathematically modelling the various stages of embryo development requires a more complete understanding of the biomechanical forces that act at the cellular level. Combining the principles and methods of biology and engineering, modelling epithelial reshaping in embryos requires knowledge of how that epithelium reacts in the presence of external forces.

What is known is that there is a strong relationship between the stresses that are present in a epithelium and the actual shape of the cells themselves [6]. Not only do the shape of cells affect the forces in an epithelium, but the forces also affect cell shape [11]. What has been shown to be important in characterizing cell shape is the aspect ratio and density of the cells, and the orientation in which the cells are stretched [8].

Cells are mechanically complex. Finite-element software has be created in an attempt to determine the forces that drive important morphogenetic processes [7]. By examining embryo development in this manner, morphogenetic movements can be broken down, analyzed and modelled at the level of individual cells. By making the mathematics more manageable, this method offers a scalable solution for considering entire cell patches. That is, individual models can be reassembled and expressed as a system of equations, the solutions to which gives spatial information about each cell. Equipped with this knowledge, information about the internal and external forces that drive the displacements can be inferred. As the embryo develops over time, displacement information can be detected at each time step. Measuring the displacements and updating force models as the embryo develops will lead to a description of morphogenetic processes which can then be tested against observable phenomena.

Approaching the modelling problem of morphogenetic processes using finite-element methods (FEM) has met considerable success [14]. Software capable of accurately predicting shape changes produced by driving forces in the neural plate of an axolotl (*Ambystoma mexicanum*) embryo during the process of neurulation, the precursory morphogenetic process to the creation of the brain and spinal cord, has been developed. The axolotl is a particularly useful animal in which to study this process because they are easy to obtain and are able to withstand experimental surgery [1]. Finite element software was able to demonstrate that only one of the existing neurulation theories could produce the observed changes in shape, thus demonstrating the model's applicability to biological problems [5].

Combining existing FEM models with accurate descriptions of cell statistics will help to further our understanding of the relationship between cell shape and the forces that act upon them.

## 2.3 Problem Description

Techniques have been developed that return numerical descriptives of the global structure present in an image based on a number of assumed and known properties of the data. As was discussed in the previous section, the statistics that are useful in characterizing a cellular texture are the aspect ratio (ratio of the major and minor axes), the orientation of the major axis and the cell density in the epithelium. In computing these descriptives, spatial-frequency domain techniques will be used.

An explanation of the statistics used to characterize a cellular texture and the spatial-frequency domain is presented in the subsections that follow.

### 2.3.1 Bulk Statistics and the Composite Cell

The problem being addressed in this thesis is in characterizing the statistical properties of an image by using the image's spatial-frequency representation as a search space. The complete set of statistics needed to robustly describe an epithelium are the orientation ($\alpha$), the aspect ratio ($\kappa$) and the density ($\rho$).

In developing computer simulations of mitosis, Brodland and Veldhuis [8] described overall cell shape and orientation in an image by creating the concept of a *composite cell.* This elliptical composite cell, conceptually the average cell in a patch, was 'constructed by summing separately the centroidal moments and products of inertia of the cells in the region and then dividing by the number of cells in the region'. In their work, the geometric properties of the composite cell were used in their computations of spatial deformation.

Numerically, the composite cell can be described by its orientation and major and minor axis lengths, since aspect ratio and density are both functions of the composite cell's dimensions. When the term *bulk statistics* is used, it will refer to three metrics: the composite cell orientation ($\theta$), major axis length ($L_{major}$) and minor axis length ($L_{minor}$). It is important to note that $\kappa$ and $\rho$ can be determined if $\theta$, $L_{major}$ and Ł$_{minor}$ are known. The term *composite cell* will denote that cell characterized by a certain set of bulk statistics.

A composite cell is shown in Figure 2.1. Each algorithm uses a different approach

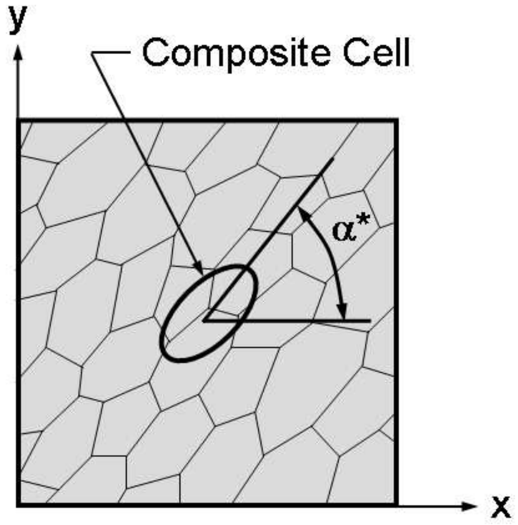


Figure 2.1: Parameters that describe the fabric of the epithelium. A composite cell is defined using the moments and products of inertia of the individual cells in the rectangular patch. The angle, measured counter-clockwise from the horizontal axis to the long axis of this composite cell, is called $\alpha^*$ and the ratio of its maximum to minimum axis lengths gives the shape parameter $\kappa$. [8]

to evaluate the values that describe the composite cell. The orientation is the angle $(-\pi/2 < \alpha \leq \pi/2)$ measured counter-clockwise from the horizontal axis to the major axis of the composite cell [8].

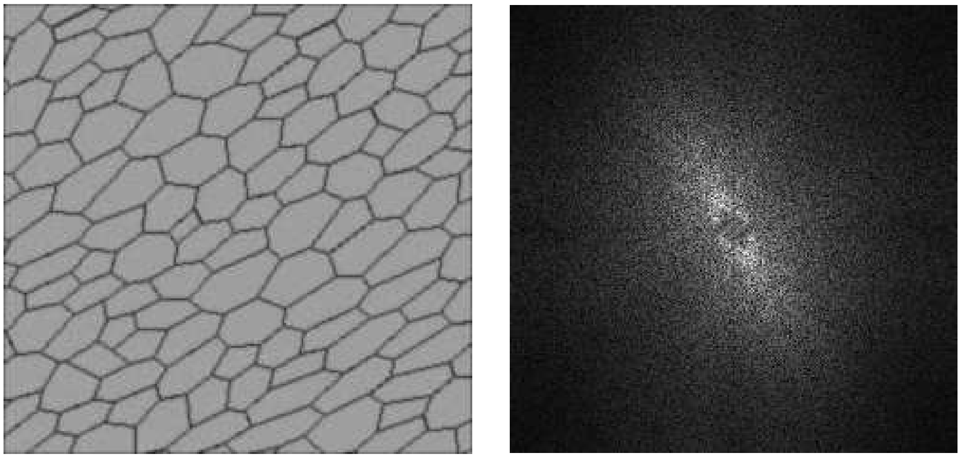
### 2.3.2 Spatial-Frequency Representation

Filtering in the spatial-frequency domain is a powerful tool that is used routinely in image processing applications [22]. Any practical signal can be expressed as a sum of weighted sinusoids. The Fourier transform uses this property to take a signal and transform it into a representation of the sinusoid coefficients. In the spatial-frequency representation of an image (a two-dimensional signal), well-defined periodic structure found in an image produces peaks in the transformed space. These peaks are used in estimating the bulk geometric characteristics from a global perspective.

The Fourier Transform (FT), used to obtain the spatial-frequency representation of an image, can provide the information from which to estimate the geometrical parameters of interest. After every transformation, the origin is shifted to the centre of the image. Similar to the spatial domain, the spatial-frequency domain can be described by coordinates. The u- and v-axes are used in place of the x- and y-axes, to describe the horizontal and vertical axes, respectively. The power spectrum can determined by taking the magnitude of the Fourier transform [10]. All power spectrums considered and displayed in this work have had their DC component set to zero. This is done to compensate for a lack of calibration in the images and to improve visualization. Shown in Figure 2.2 is a sample image and its corresponding spatial-frequency representation.

If the resolution and magnification of a spatial image is known, objects can be measured based upon their size in pixels. In the power spectrum, individual pixel locations can be described by an orientation, taken counter-clockwise from the u-axis, and a frequency in cycles per image (*cpi*), measured outward from the image centre.

Note the similarity in shape, rotated by 90 degrees, between the power spectrum distribution and what one might characterize as the composite cell in Figure 2.1, as described in the previous subsection. This is an important property and one that forms the basis of an assumption that will be stated later – that elliptical objects



(a) (b)

Figure 2.2: Synthetic mesh representing cells in an epithelium (a) and its power spectrum (b). A square root operator has been applied to the power spectrum to enhance visibility.

in the spatial domain produce elliptically shaped power spectrums. Distances from the origin in the spatial-frequency domain are represented as cycles per image (cpi) while metrics associated with length in the spatial domain are presented in pixels. With known image dimensions, cycles per image can be converted into pixels through an inverse relationship.

## 2.4 Problem Definition

The work in this thesis can be most succinctly described as using spatial-frequency domain techniques to measure the orientation and major and minor axes lengths of the composite cell that best describes the elements in an image. An sample image (Figure 2.3(a)) and its power spectrum (Figure 2.3(b)) are used to explain this

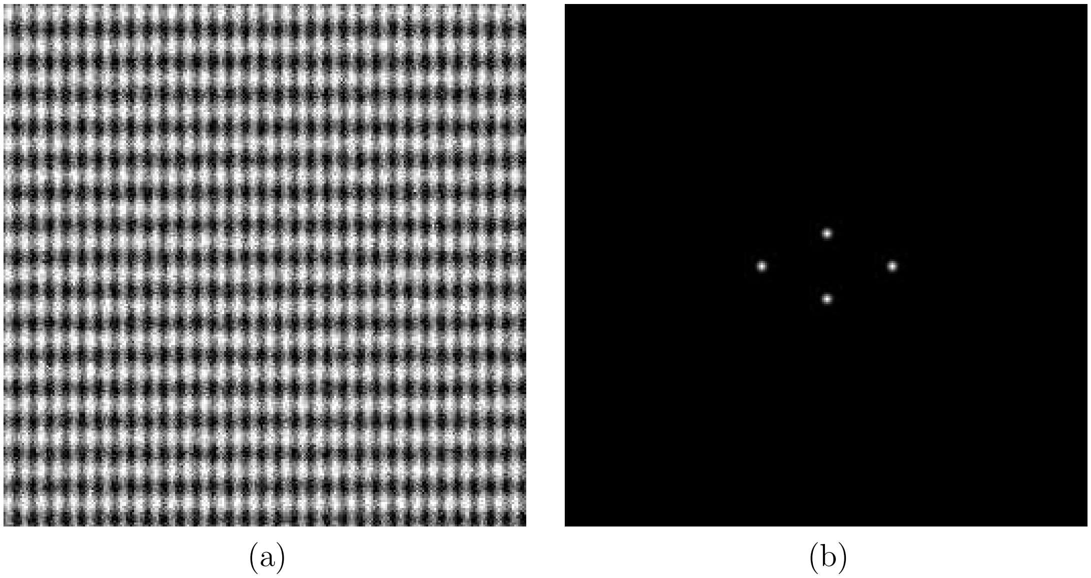


Figure 2.3: Two overlapping sinusoids are shown (a), the first with a frequency of 32cpi oriented at 0°, and the second of 16cpi at 90°. These values are clear in the image's power spectrum (b).

definition.

In the spatial image, two perpendicular repeating patterns are clearly visible. The first is a sinusoid oriented at 0° with a frequency of 32cpi (8ppc). The second signal is also a sinusoid, but this time oriented at 90° with a frequency of 16cpi (16ppc). In the power spectrum, these patterns show up as strong peaks at their respective orientation and frequency coordinates.

Each of the algorithms presented in Chapter 4 will attempt to use the information available in the spatial-frequency to measure the three statistics needed to describe the composite cell: the orientation, the major axis length and the minor axis length.

# Chapter 3

# Image Data

In determining what type of information would be analyzed, decisions regarding the scope of the problem were made with biological considerations in mind. In the type of data being analyzed, camera limitations (e.g. resolution and lighting) and environmental issues (e.g. pigmentation variation and cell division) limit the capacity for automatic geometrical interpretation.

Spatial domain techniques (e.g. edge detection [9], snake contours [2], watershed algorithm [10]) are extremely sensitive to noise. Poorly defined cell boundaries and the presence of additive noise render such computer algorithms insufficient for the problem presented. As such, estimates of the bulk geometric characteristics are made on a global basis using the spatial-frequency representation of the image under consideration. Each algorithm will make use of a different approach to characterize the composite cell described in Chapter 2.

## 3.1 Undesirable Image Properties

The epithelial cell images captured for this project are corrupted by a number of processes: the presence of noise, poorly defined boundaries and discolouration. These distortions restrict the ability for automated algorithms to extract the necessary bulk statistics. A description of these distortions is presented next.

### 3.1.1 Presence of Noise

The presence of noise in the images being tested presents a serious challenge to the effectiveness of the algorithms. Noise is considered to derive primarily from three sources. The first, camera noise, is assumed to be sufficiently small so as to not impact analysis results. The second is high frequency noise which results from the natural pigmentation and textural variation in images. A third noise process, discolouration, resulting in low frequency noise is considered in its own subsection.

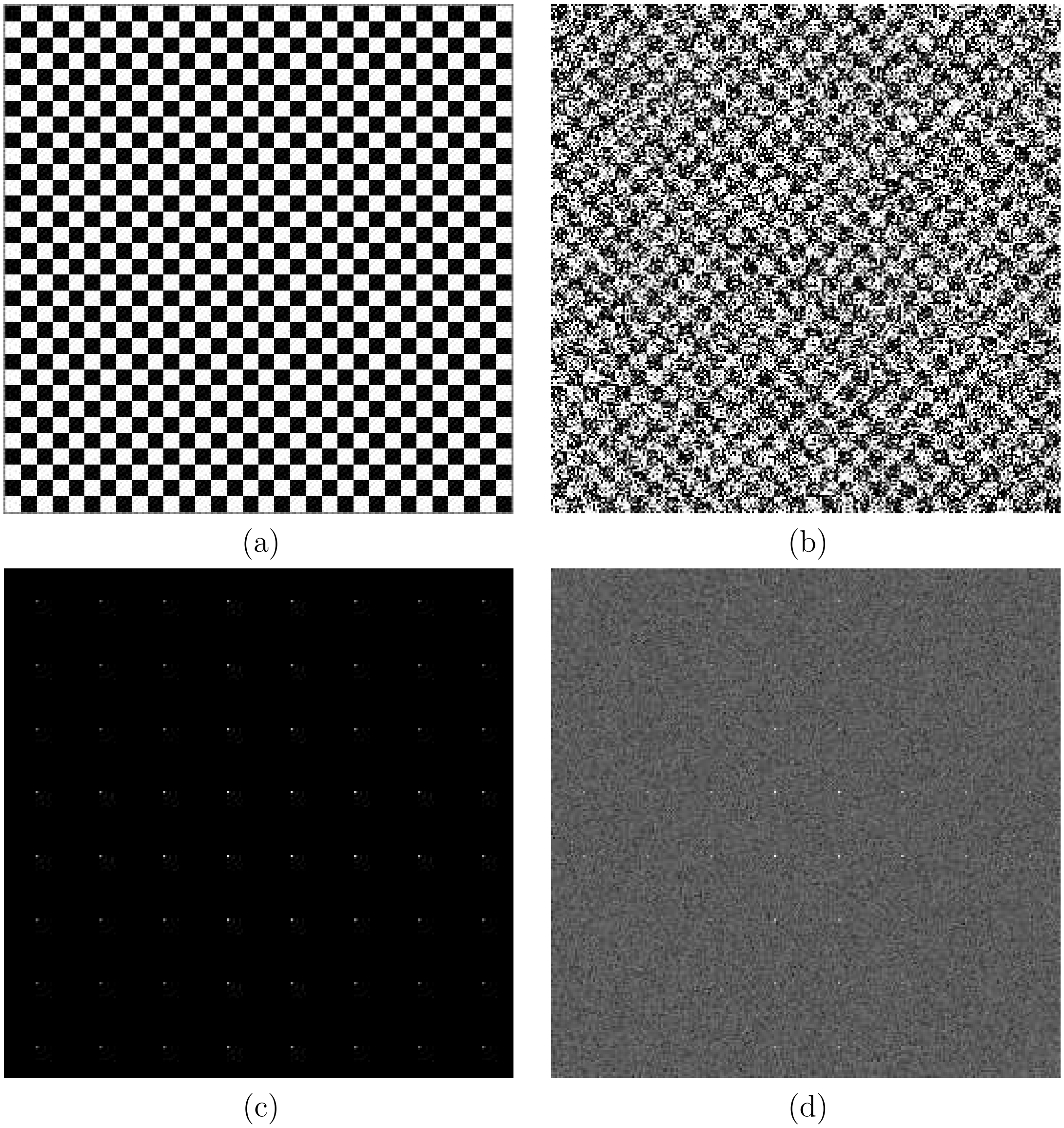
Each of the algorithms developed to estimate the orientation of cellular textures attempt to characterize the overall shape of a power spectrum. The power spectra are symmetric. However, in textures that do not completely cover the image region being analyzed, edge effects and zero padding may create artifacts along the u- and v-axes originating from the image centre. To minimize this, images were chosen or created to be approximately square and images containing extra non-textural information had those regions cropped.

In a well-structured image, high-frequency noise is not likely to significantly affect orientation estimates. The same cannot be said for low-frequency noise. Both noise types, however, can notably affect measurements of the composite cells axis lengths. The effect of noise on a well-structured image is shown in Figure 3.1. A checkerboard image is shown in its original state (foreground values set to one and background values to zero) and with Gaussian noise ($\mu = 0$, $\sigma = 1$) injected, along with their respective power spectra. Negative values resulting from the zero-mean noise are dealt with by rescaling the noisy image.

### 3.1.2 Poorly Defined Boundaries

Poorly defined boundaries between the cellular elements in an image can affect measurements of both the orientation and size. For example, an indistinguishable boundary between neighbouring cellular elements effectively gives the appearance of a larger cell. If only a few poor boundaries exist, there may be sufficient global information to accurately interpret the bulk statistics. However, too many poor boundaries can result in a meaningless characterization of the image. Figure 3.2 illustrates the problem that poorly defined boundaries present. The outline of a checkerboard is shown, as well as the same outline with some of its edges removed. The resulting power spectra are shown to illustrate the effect of poor boundaries on the search space. A small amount of noise was injected into the checkerboard outlines to improve visibility of the power spectrum.

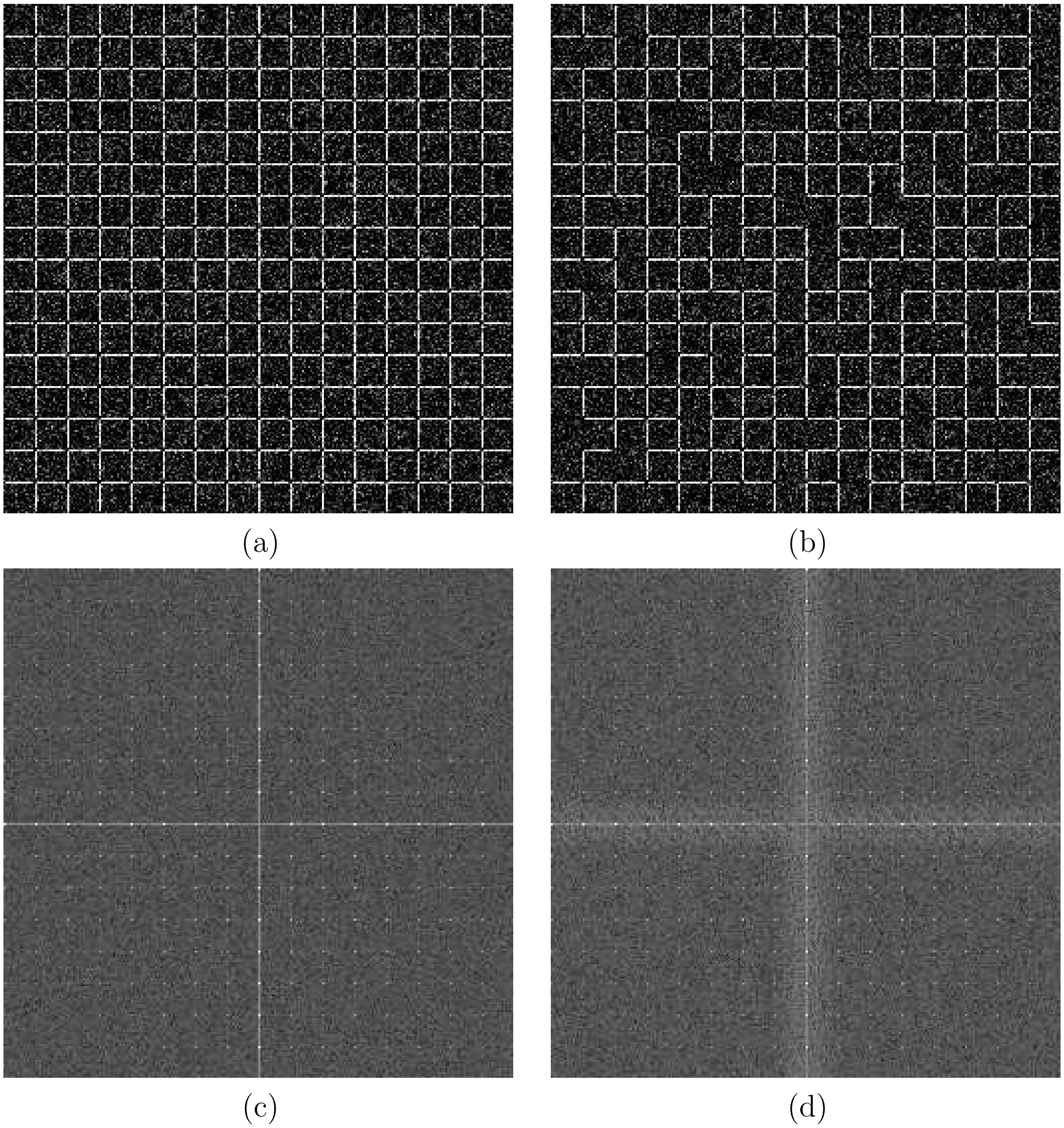
Notice that the resulting energy signal is not as strong in the image with some boundaries removed. This is an important example in understanding the relationship between spatial and spatial-frequency domains, since it is the boundaries of cells, and not their colour or textural qualities, that are being determined.



(a) (b)

(c) (d)

Figure 3.1: A checkerboard pattern (a) and the same pattern with noise injected (b) are shown. To demonstrate the effect of noise on the spatial-frequency representation of an image, the power spectrums are shown in (c) and (d), respectively, with a fourth root operator applied to improve visibility.



(a)

(b)

(c)

(d)

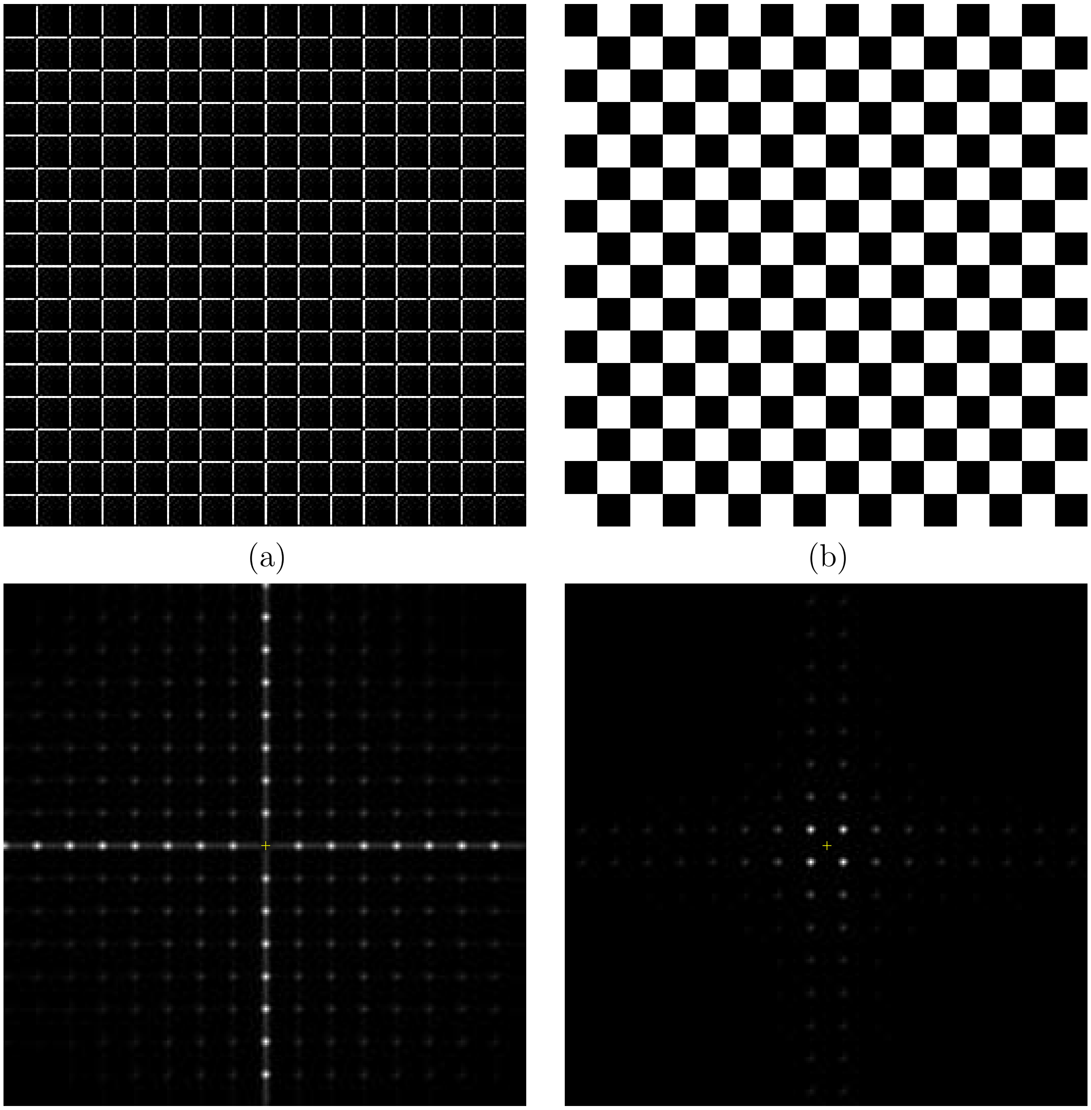
Figure 3.2: The edges of a checkerboard is shown (a), along with many of those edges removed (b). The power spectrums of the original and the image used to demonstrate poorly defined boundaries are shown in (c) and (d), respectively. A fourth root operator is applied to enhance power spectrum visibility.

### 3.1.3 Discolouration

Two different types of discolouration exist that can significantly impact the effectiveness of the algorithms. The first type – inconsistent illumination across an image – is considered to be low frequency noise. A homomorphic filter can be applied to an image to reduce the effects of the illumination [10], but some low frequency noise may remain.

The other type of discolouration results from the natural pigmentation variation in an epithelium. As cells undergo the process of mitosis, a visible darkening in their colour is apparent. While the same homomorphic filter can be applied to reduce such shading variation, some noise, probably non-linear, will remain.

To demonstrate the effect of natural cellular discolouration on an image's power spectrum, Figure 3.3 shows (a) a checkerboard outline, (b) a checkerboard and their respectively power spectra ((c) and (d)). To create the checkerboard outline, a simple edge detection operator was run on the checkerboard to create the outlines that are visible. The result is two images with the same edges, but different shading. As can be seen in the power spectra, the checkerboard outline is comprised primarily of horizontal and vertical signals, but the checkerboard has its strongest signals based at 45° and $-45°$. Although similar representations of the images, the bulk statistics as determined from looking at the power spectra are quite different.



(a)

(b)

(c)

(d)

Figure 3.3: The outline of a checkerboard are shown (a) beside a checkerboard (b). Their respective power spectra, used to show the effect of shading (discolouration), are shown in (c) and (d). The visibility of the power spectra was enhanced with a Gaussian smoothing operator. The yellow ticks are used to denote the image centre.

## 3.2 Data

The algorithms presented in Chapter 4 are designed to operate on a specific type of image. The type of images being used for testing are described in this section.

### 3.2.1 Image Characteristics

An image is said to be stationary if the statistics – mean, variance and spectrum – of any subset accurately describe the statistics of the entire image [20]. It is not an uncommon practice to assume stationarity when data is only approximately stationary. Approximate stationarity is important in this problem because it constrains the resulting sinusoidal coefficients from the Fourier transform to those frequencies and orientations from a single texture. As a result, all energy in a power spectrum derives either from the image being considered or from noise associated with that image. It is assumed that the data under investigation in this work are approximately stationary.

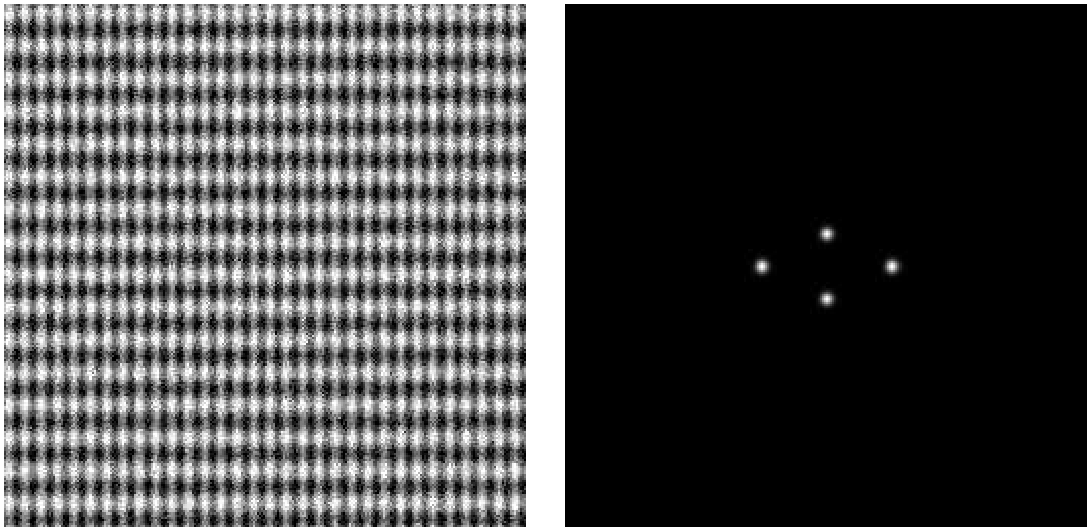
Algorithms are designed with the end goal of analyzing images that will be obtained from an epithelium undergoing deformation in the presence of external forces. Experience has shown that it is reasonable to assume that cells in the epithelium will align themselves along an approximately common orientation. Furthermore, since forces are expected to be consistently distributed across the patch, it is anticipated that aspect ratio metrics, while varying, are sufficiently similar that estimates of the composite cell aspect ratio can be made by analyzing the problem from a global perspective. That said, the following two primary assumptions are made for the image data being analyzed:

1. Images will be approximately stationary; and
2. Cells will be approximately elliptical in shape.

Subtle variations are to be expected in the individual cell characteristics, but it is assumed they will be approximately similar and that approximate stationarity will hold. The second assumption states that these images will contain elliptically shaped cells. Stationary images with such elliptically shaped elements in the spatial domain tend to have elliptically shaped power spectrums as well. This is an important relationship as it is the shape of the energy in the power spectrum that will be characterized. That is, from the shape of the energy in the power spectrum will be derived the bulk statistics that describe the composite cell.

### 3.2.2 Sinusoidal Imagery

The first image considered is a simple 2d sinusoidal image. This image is shown in Figure 3.4. It consists of two sinusoids, the first running along the y-axis (90°) at 16cpi, and the second along the x-axis (0°) at 32cpi. This image, although a fairly simple test case with a very high signal-to-noise ratio due to its simple and uninterrupted pattern, represents an important test case. With so little energy external to the two sinusoids, it is not unreasonable to assume such an example would present the simplest case for an algorithm. While some algorithms perform well on such simple images, it will later be shown that it is not necessarily true for even the most consistent algorithm developed.



(a) (b)

Figure 3.4: A 2D image (a) comprised of one sinusoid at 32cpi along the horizontal and another at 16cpi along the vertical. A small amount of Gaussian noise has been injected. The power spectrum is shown in (b), after being smoothed by a Gaussian filter and rescaled to improve visibility.

### 3.2.3 Brodatz Imagery

Images taken from the Brodatz collection [4] represent the second class of test images being considered. This collection has its own labelling convention – a letter 'D' followed by an image's number. The three images used will be Brick (D95), rotated Brick (also D95, herein denoted D95R) and Woven Aluminum Wire (D1). Images D95 and D95R have been slightly modified from their original state: half-sized bricks that were interspersed with the full-sized bricks were removed to create consistent brick sizes. These images and their respective power spectra are shown in Figure 3.5.

These images were selected to demonstrate specific principles. While the bulk statistics of the Brodatz imagery had to be calculated manually, they were tested because their complexity lies between that of the 2d sinusoidal image and the more complicated imagery of the following subsections. That is, these images are well-structured and have more variation associated with them than a 2d sinusoidal image but lack the individual cellular element variation that will be presented in the following subsection. The ability of an algorithm to discern the major and minor axis lengths of the composite cell is imperative, even in cases where there is no obvious dominant orientation in the texture.

### 3.2.4 Artificial Cellular Imagery

The second set of data was chosen to more closely resemble actual cellular data. Created by annealing Voronoi tesselations [11], these images are well suited to testing the algorithms. Not only are the cellular elements shaped similar to actual

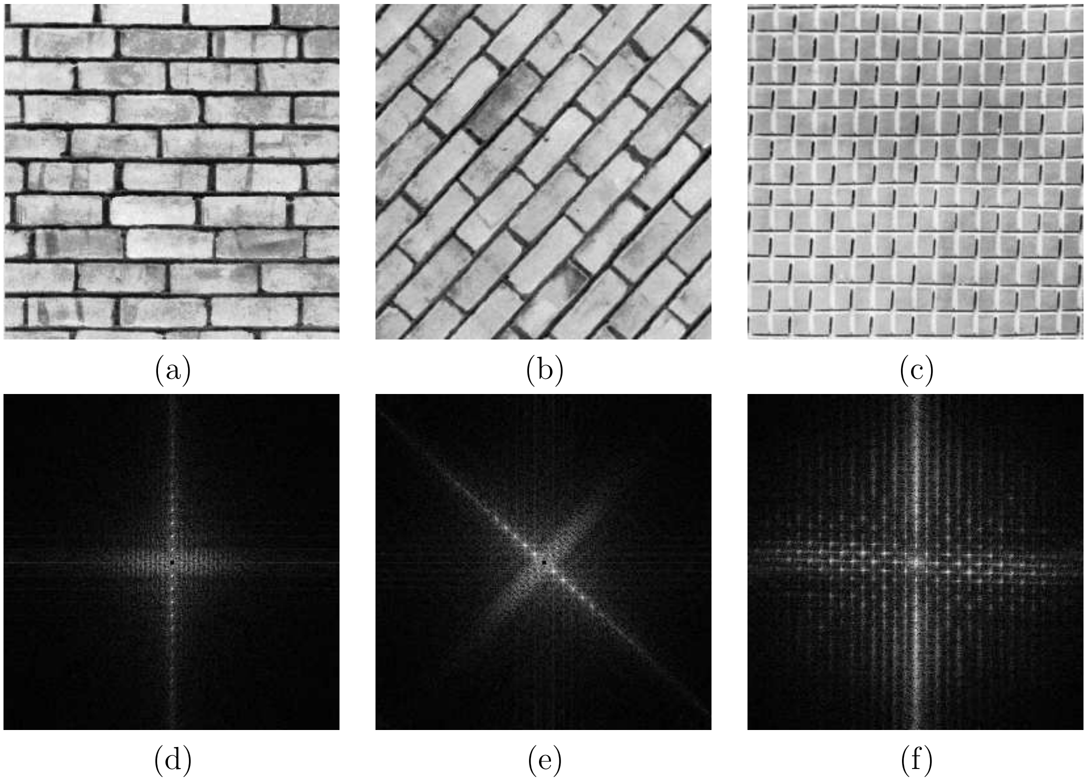


Figure 3.5: Three Brodatz images and their respective power spectra are shown: Brick (D95) in (a) and (d), Brick (D95R) rotated by 47.5 degrees in (b) and (e), and Woven Aluminum Wire (D1) in (c) and (f). A logarithmic operator has been applied to the power spectra to enhance visibility.

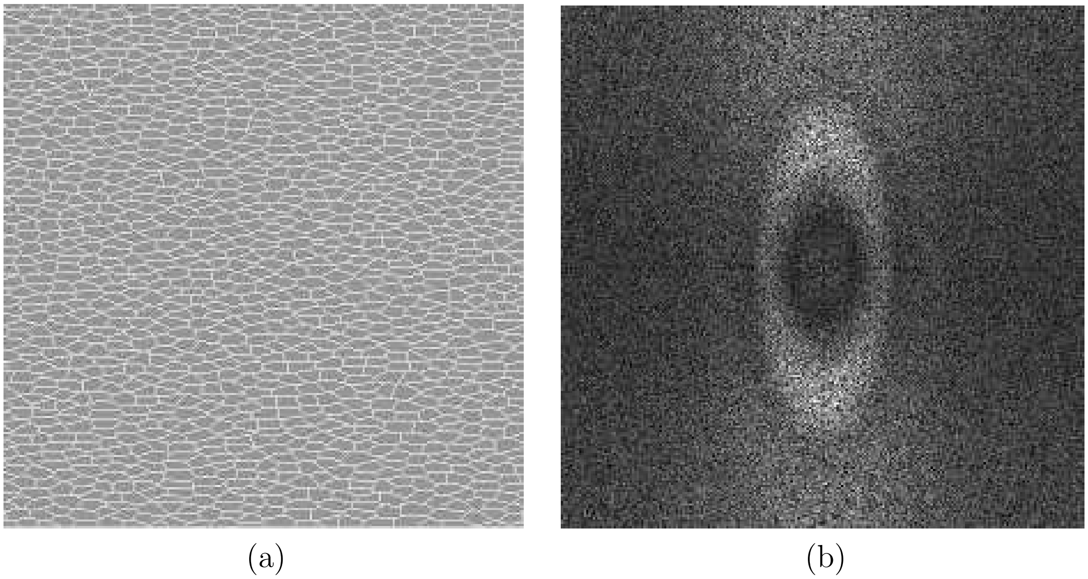


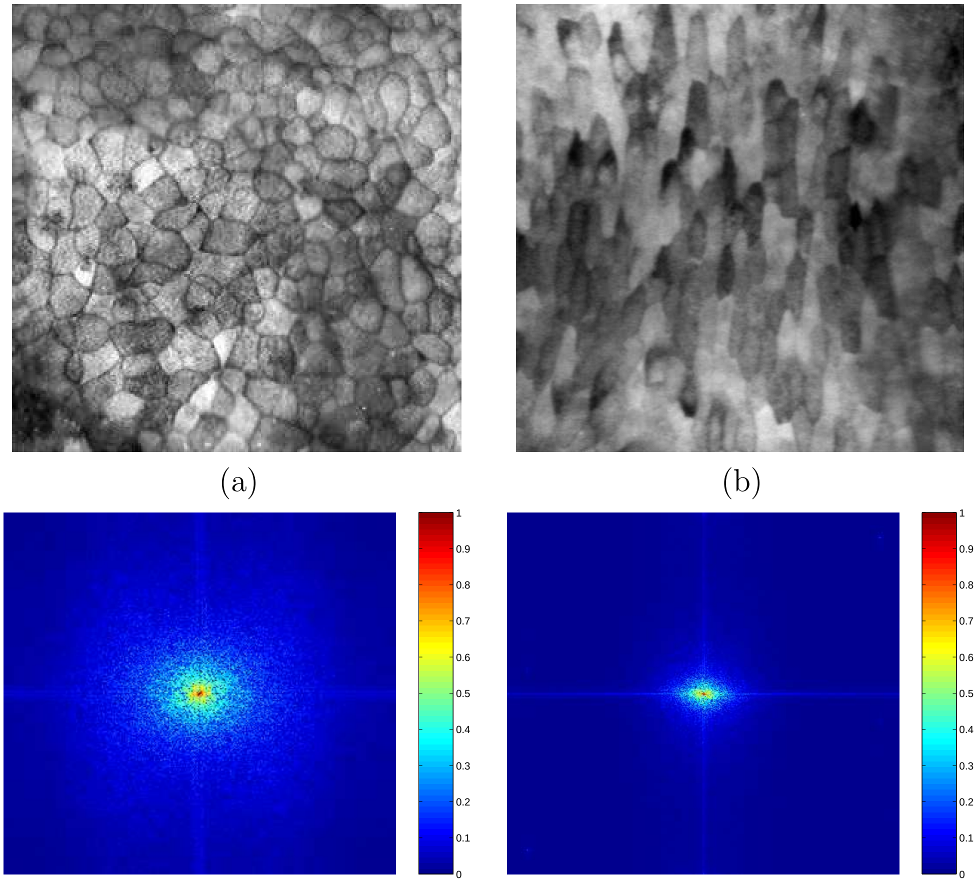
Figure 3.6: A synthetic mesh representing cells in an epithelium (a) and its power spectrum (b). Because their bulk statistics are known *a priori*, meshes play an integral role in the testing of the algorithms.

cells, but since they were artificially generated, their bulk statistics are known *a priori*. The variable nature of the individual cellular elements add complexity that was not present in the 2d sinusoidal and Brodatz imagery, and produces a more dispersed power spectrum. An example of this artificial cellular imagery, a synthetic mesh representing cells in an epithelium, is shown in Figure 3.6.

### 3.2.5 Natural Cellular Imagery

The final data set to be considered will be actual cellular data captured while an axolotl epithelium is undergoing natural deformation (a) and in the presence of external forces (b). The embryos from which these images are taken follow a natural progression which is readily understood and well-documented. Each stage has specific characteristics associated with it and are categorized according to their stage number in the progression [1]. The image undergoing natural deformation was captured approximately 20 hours (Stage 8) into development. This image was captured using a DAGE-MTI CCD-C72 camera and associated optics (Wild M420 microscope). The cells in the presence of external forces were captured approximately 60-70 hours (Stage 18) into development using a Sony XCD-X700 with a Flolite 6000II zooms lens.

Shown in Figure 3.7, this data represents the most challenging example used in testing the robustness of the algorithms. It also represents the data that this work is seeking to ultimately characterize. As can be seen from the images, a number of noise processes are present which make even visual inspection difficult. Two illumination processes are present in the natural data: the natural discolouration that exists in cells undergoing mitosis and the larger scale noise that is a result of illumination during image capture (lighting and reflection). In addition, there are at least three other noise processes occurring: one from the camera, another from the noise inherent in capturing digital imagery and a more significant third related to natural pigmentation variation within individual cells (unrelated to mitosis).



(a) (b)

(c) (d)

Figure 3.7: Two images of natural cellular data are shown: (a) an image with fairly distinct boundaries between cells and (b) an epithelial patch that has been stretched by external forces. Their respective power spectra are shown in (c) and (d). A logarithmic operator is applied to the power spectra to enhance visibility. Additionally, due to the concentration of energy near the image origin, the power spectra are displayed in colour to improve visibility.

# Chapter 4

# Methods

Four techniques were developed to extract the bulk statistics of a cellular texture: Area Moments, Correlation and Ellipse Fitting, Gabor Filters and Least Squares Ellipse Fitting. The first algorithm makes use of area moments to determine the orientation and then applies localized statistical analysis to determine the major and minor axis lengths. The Correlation and Ellipse Fitting method computes the orientation by comparing correlation measures of rotated power spectrums and then fits an ellipse to estimate the axis lengths. The third algorithm uses Gabor filters to obtain a response at a particular filter configuration and attempts to maximize that response by comparing values in the search space. The final method makes use of efficient mathematical algorithms to determine the ellipse that minimizes the sum of the squares of the distances between energy in the power spectrum and a best fit ellipse.

The algorithm that makes use of Gabor Filter and the Correlation and Ellipse Fitting method both employ exhaustive searches in their attempts to characterize the image under investigation. As these techniques were intended to be proof of concept, such a search is considered sufficient. However, experimentation was conducted to adapt these algorithms for use in an optimization framework. While testing is far from complete on the optimization algorithms, their potential is promising. As such, the optimization algorithm is described in detail in Section 4.5.

The Matlab code for all techniques described in this chapter can be found in Appendix B. The Optimization code is in Appendix C.

## 4.1 Area Moments

Area moments provide information about geometric characteristics of an object about a fixed point. Often used in engineering analysis, moments can be used to determine the resistance, or lack thereof, of a body to angular acceleration. By determining the centroidal moments ($I_{uu}$, $I_{vv}$) and product of inertia ($I_{uv}$), statistics associated with an objects centroid, orientation and aspect ratio can be measured [23]. These inertial measures, if applied to the power spectrum, are determined as follows:

$$I_{uu} = \sum_{v=1}^{rows} \sum_{u=1}^{cols} F(u, v) \cdot v^2 \tag{4.1a}$$

$$I_{vv} = \sum_{v=1}^{rows} \sum_{u=1}^{cols} F(u, v) \cdot u^2 \tag{4.1b}$$

$$I_{uv} = \sum_{v=1}^{rows} \sum_{u=1}^{cols} F(u, v) \cdot u \cdot v \tag{4.1c}$$

where $u$ is the column index, $v$ is the row index, *rows* is the number of rows, *cols* is the number of columns and $F$ is the image power spectrum [23].

After computing the inertial measures, the orientation of the composite cell major axis can be computed as:

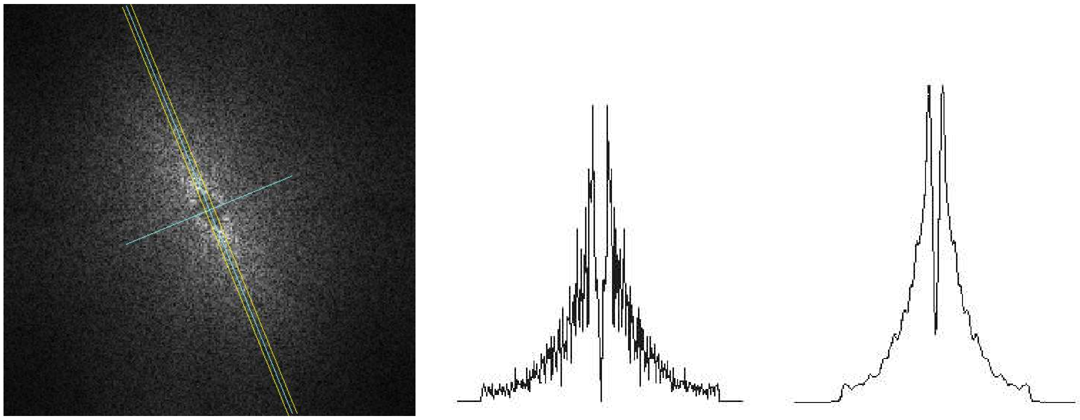
$$\theta = \arctan(\frac{2I_{uv}}{I_{uu} - I_{vv}}) \tag{4.2a}$$

provided $I_{uu} \neq I_{vv}$. In the case that $I_{uu} = I_{vv}$,

$$\theta = \begin{cases} \pi/4 & \text{if } I_{uv} > 0 \\ 0 & \text{if } I_{uv} = 0 \\ -\pi/4 & \text{if } I_{uv} < 0 \end{cases} \tag{4.2b}$$

In terms of physical objects, area moments can be used to compute the orientation and aspect ratio, provided the object is of uniform density. However, the spatial-frequency representation of each image being investigated is not uniform in density. As such, preliminary testing indicated that the natural variability and the noise in the images rendered a direct approach using area moments to compute the aspect ratio as infeasible.

In order to extract the frequency values required to characterize the aspect ratio and density, a statistical method is applied following the computation of the orientation. Figure 4.1 is used to help describe the process. All points within a narrow three pixel band of the composite cell major axis were projected onto a line. The major axis is the light blue coloured line and the narrow band is shown by the



(a) (b) (c)

Figure 4.1: A Statistical techniques for estimating cell lengths using Area Moments. Shown is the narrow band superimposed on the power spectrum in (a), the projection (b) and the projection after smoothing (c).

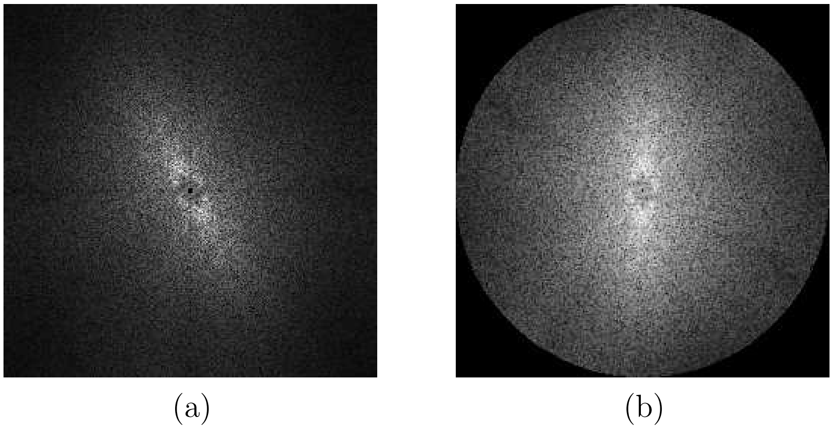
yellow lines in Figure 4.1(a). This narrow band was chosen to minimize the effects of high frequency noise while maximizing the contribution of local pixels. A plot of the resulting projection is shown in Figure 4.1(b). A Gaussian smoothing filter ($\mu = 0$, $\sigma = 1$) is then applied to the projection (shown in Figure 4.1(c)) and the coordinate associated with the maximum value, measured outward from the distribution centre, was taken to be the frequency most representative of the composite cell major axis length. A similar metric was obtained along the perpendicular axis to obtain the length of the composite cell minor axis.

Along a given projection, this algorithm is only capable of producing whole-valued measures of the major and minor axes. Results obtained by this algorithm not presented as a whole number are a result of computing lengths that have been rotated. Should more precise estimates be required, an interpolation of the points along the smoothed projections could be performed.

## 4.2 Correlation and Ellipse Fitting

Correlation between two variables is a measure of the extent to which a change in one variable will affect the other [25]. The second technique to be presented applies concepts central to correlation in order to compute a similarity metric. This metric is then used to determine the orientation which best describes the nature of an image's power spectrum. Once the orientation is known, the algorithm maximizes the energy per pixel of an ellipse that is fitted to the power spectrum. In addition to an orientation, this ellipse is completely described by its major and minor axis lengths. Both the orientation and the axes length determination algorithms utilize exhaustive searches.

The orientation algorithm operates by attempting to align the major and minor axes of the composite cell (and therefore, the power spectrum) with the u- and v-axes. That is, the image is rotated by every integer value between $-45^\circ \leq \theta < 45^\circ$. After each rotation, another image is created by "flipping" the rotated power spectrum about the vertical axis, bisecting the origin. A third image is created by flipping about the horizontal axis. A measure of similarity is obtained by summing the dot products of the original image with the horizontally flipped image and the dot product of the original image with the vertically flipped image. For a symmetric image, the orientation that best aligns with the u- and v-axes would produce the largest correlation measure. An example of a power spectrum that has been rotated to align with the u- and v-axes is shown in Figure 4.2. Only those pixels in an circle with a diameter equal to the length and width of the image are used in computing orientation estimates. This algorithm is repeated at one decimal place of resolution



(a) (b)

Figure 4.2: The orientation estimation portion of the Correlation and Ellipse Fitting algorithm. The original image (a) is rotated by a certain amount (b). A measure of correlation can be obtained as to how closely the top (and left) half of image lines up with the bottom (and right) half. The angle producing the highest similarity measure corresponds to the orientation.

about the integer-valued orientation that produces the largest correlation to obtain a more precise solution.

Three methods were considered for characterizing the major and minor axis lengths. The first two, Power Spectrum Rescaling (PSR) and Chi-Squared Curve Fitting (CSCF), did not work effectively. PSR, in a manner similar to the orientation algorithm, was based on a correlation process. Two identical power spectrums were used, but with the second spectrum offset by 90° from the original. By adjusting the level of rescaling, the height and width of the rotated power spectrum would change while retaining its volume. The idea was that the largest correlation measure would result when the spectra had been rescaled so as to appear superimposed immediately on top of one another. However, the presence of noise, image distortion during rescaling and the nature variation of some power spectrums rendered

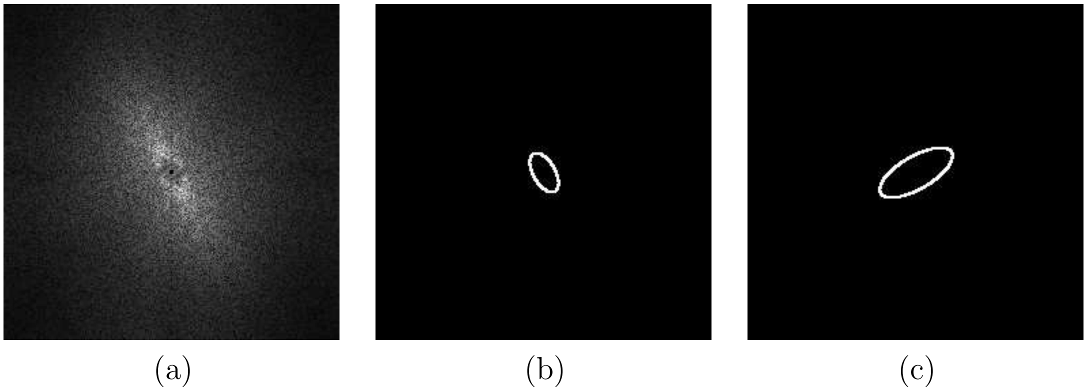


Figure 4.3: Correlation and Ellipse Fitting axis length estimation algorithm. The dot product is taken between ellipses with varying major axis lengths and aspect ratios and a power spectrum (a). The ellipse producing the highest energy per pixel best describes the power spectrum. In this example, the response from the ellipse in (b) would have a higher energy per pixel than that in (c). Although (c) might capture more energy, its energy per pixel would be lower than that from (b).

such a technique incapable of accurately gauging the geometric statistics.

The CSCF attempted to fit a chi-square curve to those points along the major and minor axes. In doing so, it was thought that the spatial extent of the power spectrum might also provide valuable information about the variation associated with the estimates. Two contributing noise processes – low frequency noise resulting from discolouration and high frequency noise resulting from variations in the size and orientation of the individual cells in an epithelial patch – resulted in this method not producing meaningful results.

The third method and that which is used in this algorithm attempts to find an ellipse that best characterizes the power spectrum. This method functions by performing an exhaustive search of major axis lengths and aspect ratios once the orientation has been estimated.

A binary image is created in which an ellipse about the image centre is created for every value of major axis length and aspect ratio. All pixels in the ellipse image on which the ellipse is found is assigned a value of 1, while all other pixels are assigned a 0. The dot product of this ellipse image with the original power spectrum is computed, which is then divided by the number of pixels comprising the ellipse. This creates a measure referred to as the *energy per pixel*. The major axis length and aspect ratio generating the largest energy per pixel are taken as descriptives of the composite cell. An example is shown in Figure 4.3.

## 4.3 Gabor Filters

Gabor filters are a class of multi-channel filters capable of decomposing an image into a collection of frequency and orientation texture features using multiple resolution techniques [16]. That is, a 2-d Gabor filter can be designed to be sensitive to specific frequencies and orientations. As a result, spatial images that do not contain a specific frequency and orientation produce a negligible response to the filter. Measuring the image response to a filter establishes a basis for determining the suitability of a frequency and orientation metric in describing the image being considered.

Gabor filters have received a great deal of attention because their multi-channel nature mimics characteristics of the human visual system. In experiments with human subjects, Rao and Lohse were able to identify several low-level features used in texture perception [24]. Two of these features – periodicity and directionality – correspond to the frequency and orientation, respectively, in the spatial-frequency domain [12].

Clausi and Jernigan [15] extended the notion of localized filters by comparing different techniques used in generating texture features for the purpose of optimal texture separability. As modulated Gaussian filters, the spatial-frequency domain representation of the complex Gabor filter is given by:

$$H(u,v) = \exp\left[-2\pi^2\{(u-F)^2\sigma_x^2 + v^2\sigma_y^2\}\right] \tag{4.3}$$

where $F$ is the spatial-frequency, $u$ and $v$ are the spatial-frequency coordinates, and $\sigma_x$ and $\sigma_y$ are the standard deviations of the Gaussian component of the filter in frequency and orientation, respectively, as determined by [19]:

$$\sigma_x = \frac{\sqrt{\ln 2}(2^{B_F}+1)}{\sqrt{2}\pi F(2^{B_F}-1)} \tag{4.4}$$

$$\sigma_y = \frac{\sqrt{\ln 2}}{\sqrt{2}\pi F \tan(\frac{B_\theta}{2})} \tag{4.5}$$

It was argued [15] on psychovisual grounds that the frequency bandwidth $B_F$ should be set to one octave, and the angular bandwidth, $B_\theta$, to 30 degrees for accurate coverage and to achieve resolution similar to the human visual system. Two examples of spatial-frequency Gabor filters are presented in Figure 4.4. Figure 4.4(a) has a frequency of 16cpi and is oriented at $-45°$ (as measured counter-clockwise from the horizontal axis), and Figure 4.4(b) is tuned to a frequency of 64cpi and oriented at 30°.

The magnitude response serves as the basis for this algorithm, which searches

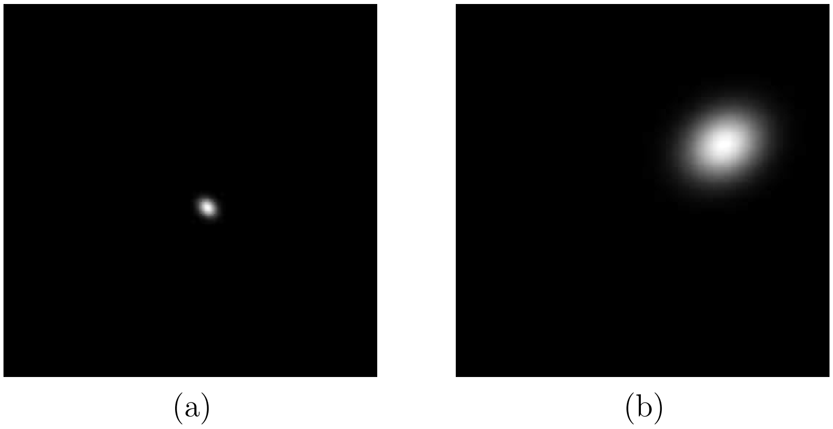


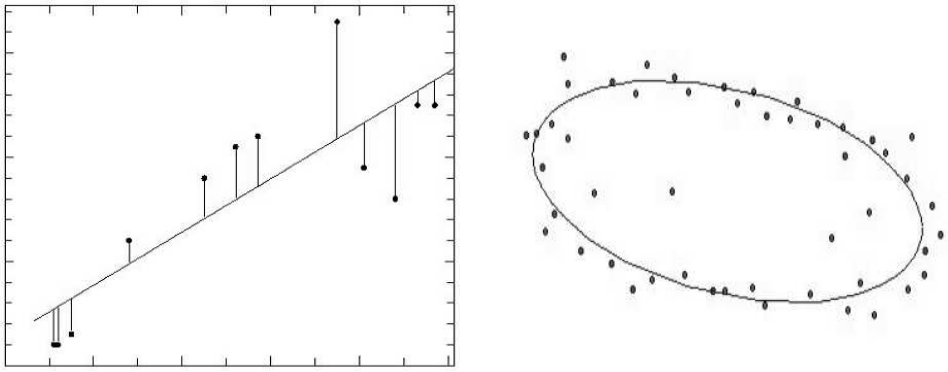
Figure 4.4: Spatial-frequency Gabor filters are shown. They are configured to (a) 16cpi oriented at −45° and (b) 64cpi oriented at 30°

for the filter configuration that produces the strongest magnitude response using a brute force search. In this framework, filter responses are tested at every other whole value of frequency in the range $min_{Freq} \leq F \leq max_{Freq}$, and every third whole valued orientation in the range $-90^\circ < \theta \leq 90^\circ$, where $min_{Freq}$ and $max_{Freq}$ are dependent upon the image size and chosen to ensure sufficient coverage of the filter in the search space. The frequency and orientation of the Gabor filter that produces the strongest magnitude response will be associated with the orientation and length of one of the axes from the bulk geometrical parameters. The algorithm then refines the search down to the nearest decimal value in both orientation and length.

Once one length and its orientation are known, it is then a matter of estimating the second axis length. A one dimensional exhaustive search is then performed along a ray perpendicular to the orientation described in the previous paragraph to identify the frequency along that ray producing the strongest response. This will correspond to the second axis length of the bulk statistics. The larger of the two length measures is returned as the major axis length and the angle associated with that axis is assigned to the orientation.

## 4.4 Least Squares Ellipse Fitting

The objective of curve fitting is to find a mathematical equation that best describes a set of points. Generally, those points represent a solid relationship (i.e. signal) in the presence of non-ideal results (i.e. noise). Linear least-squares is a method which is often employed to find the coefficients to linear equations that are a straight line "best fit" to a data set. Shown in Figure 4.5(a) is an example of linear least squares. Figure 4.5(b) shows an example of least squares ellipse fitting.



(a) (b)

Figure 4.5: Examples of least squares: linear (a) and ellipse fitting (b).

To accommodate for non-linear relationships between data sets, coordinate transformations are often employed. For this technique, a specialized least squares approach was used to calculate the coefficients of the equation for an ellipse to a set of data points.

Numerous methods have been developed for fitting ellipses to data points, each with varying levels of success. Many of these techniques [3][26][27] attempt to fit data points to a general conic and rely on an additional constraint to force the solution into an ellipse. Fitzgibbon et al. [17] present a direct least squares based ellipse specific method, and contrast this method to earlier ones. In this method, the general conic is represented by the second order polynomial:

$$G(\mathbf{r},\mathbf{x}) = \mathbf{r}\cdot\mathbf{x} = as^2 + bst + ct^2 + ds + et + f = 0, \tag{4.6}$$

where

$$\mathbf{r} = \begin{bmatrix} a & b & c & d & e & f \end{bmatrix}^T \tag{4.7}$$

$$\mathbf{x} = \begin{bmatrix} s^2 & st & t^2 & s & t & 1 \end{bmatrix}^T. \tag{4.8}$$

To ensure an ellipse specific conic, the following constraint is applied:

$$b^2 - 4ac < 0. \tag{4.9}$$

Fitting a general conic to the data was then a matter of minimizing the sum of the squared algebraic distances of the points to the conic. For a set of data points $(s_i, t_i)$, $i = 1 \ldots N$, this is represented by the coefficients $\mathbf{r}$ as determined by:

$$\min_{\mathbf{r}} \sum_{i=1}^{N} G(s_i, t_i)^2 = \min_{\mathbf{r}} \sum_{i=1}^{N} (G_{\mathbf{r}}(\mathbf{x}_i))^2 = \min_{\mathbf{r}} \sum_{i=1}^{N} (\mathbf{x}_i \cdot \mathbf{r})^2 \tag{4.10}$$

The minimization in Eq. 4.10 can then be solved using the rank-deficient generalized eigenvalue system:

$$\mathbf{D}^T\mathbf{D}\mathbf{r} = S\mathbf{r} = \lambda C\mathbf{r} \tag{4.11}$$

where $\mathbf{D} = \begin{bmatrix} x_1 & x_2 & \ldots & x_n \end{bmatrix}^T$ is called the *design matrix*, $\mathbf{S} = \mathbf{D}^T\mathbf{D}$ is called the *scatter matrix* and $\mathbf{C}$ is the matrix that expresses the constraint in Eq. 4.9, written as:

$$\min_r \|\mathbf{D}\mathbf{r}\|^2 \text{ subject to } \mathbf{r}^T\mathbf{C}\mathbf{r} = 1 \tag{4.12}$$

To estimate the composite cell metrics, the image power spectrum was rescaled between 0 and 1, and vectors of column ($\mathbf{x}$) and row ($\mathbf{y}$) coordinates were created by creating a coordinate at each pixel for every 0.1 value of intensity. Such a bin value was chosen to create enough points to sufficiently characterize the power spectrum without disregarding too many pixels with small, but not insignificant, intensities. After building the design and scatter matrices, the eigensystem in Eq. 4.11 can be solved and the eigenvector corresponding to $\mathbf{r}$ computed. The code used for this algorithm was that presented in Fitzgibbon et al. [17] and subsequently incorporated by the author.

## 4.5 Optimization

The techniques described in the previous sections each have their own inherent subtleties that impact their computational complexity. In an attempt to reduce the algorithm time, optimization techniques were explored. The area moments and least squares techniques were purely statistical in nature and therefore could not benefit from search-based optimization methods. However, for those algorithms that utilized Gabor filters and ellipse fitting, their objective value based decision algorithms lend well to non-derivative optimization methods.

The optimization framework applied to these algorithms was based on the method proposed by Hooke and Jeeves [18], which first appeared in 1961 as a method of finding the best solution through the 'sequential examination of trial solutions involving comparison at each trial solution'. In this method, a 'step' is taken from the frequency and orientation producing the current largest objective value, the acceptability of that step is measured, and either the new parameters are selected or a revision in the search step length is made.

Torczon was able to show that this optimization method, which is often referred to as a pattern search, converges when a sufficient number of directions are considered in finding a suitable search direction, unnecessarily short steps are avoided, and unsuitable steps are restricted [21][28]. These requirements were satisfied by checking along basis directions of the rational lattice, establishing a lower limit on step size (based on sampling accuracy), and structuring the objective function in such a way as to avoid steps that are too long.

The algorithm presented herein was implemented as a two-state system, and the automated optimization search continued until a sufficient decrease had been observed. An initial point was chosen centred in the search space so as to maximize coverage. The basis for the search were unit vectors of the search parameters of interest.

The pattern search begins in *State 1* and progresses as follows:

### Optimization Algorithm

#### *State 1*

An exploratory step is taken. This is done by doubling the step length, evaluating the response in each of the four basis directions, selecting the best value and comparing it to the current best point. If the result is improved, the step length is maintained at double the original and the best point is returned as the current point. If the result is not lower, the step length is returned to its original value and similar exploratory steps are taken. If a better value is found, that point replaces the previous point and the next iteration begins in State 1. Otherwise, State 2 is entered.

#### *State 2*

The purpose of State 2 is to check for sufficient decrease. Upon entering State 2, the step length is checked, and if sufficient decrease is not satisfied, the step length is halved and the state machine enters State 1. If sufficient decrease is met, the algorithm stops and the current location is returned as optimal.

# Chapter 5

# Testing

Each of the algorithms presented in Chapter 4 takes a unique approach to determining the orientation and major and minor axis lengths that best characterize the individual cellular elements in an image. How the energy is distributed in each of the power spectra on which they operate can significantly impact the effectiveness of the techniques. By applying each algorithm to a collection of images, each with different bulk statistics, the strengths and weaknesses of each method will be explored.

The testing of the algorithms is broken down into four stages based on the images shown in Section 3.2: a 2d sinusoidal image, three Brodatz images, eight images of artificial cellular data and two images of natural cellular data.

A 2d sinusoidal image is the first data tested. The purpose of this image is to introduce the algorithms with a simple and relatively noiseless image. The bulk statistics are also known *a priori* from the sinusoid parameters. Brodatz imagery is the second data set tested and is used to quantify the effectiveness of the algorithms on well-structured, natural-looking imagery. The images chosen from the Brodatz collection were chosen because of their rectangular shaped cellular elements and consistent pattern. The actual bulk statistical values of these images were measured manually.

Artificial cellular data – synthetic meshes representing cells in an epithelium – offer a variety of advantages in testing. By virtue of the manner in which they are created, synthetic meshes have known bulk statistics. Images were made with varying bulk statistics to test the ability of the algorithms to detect small changes. As a result, this section represents the most comprehensive test set of the four.

Natural cellular imagery, the final data set to be tested, presents a major challenge to automated bulk statistic measuring algorithms due to their irregularly shaped individual cellular elements and poorly defined boundaries. This makes determination of the bulk statistics far more difficult relative to the other three image types.

## 5.1 Test Results

The results of applying the automated bulk statistic measuring algorithms to a variety of images is shown in this section. In the Brodatz imagery, the measured bulk statistics are shown superimposed on the power spectra of images considered. In the data sets that have smaller individual elements, a composite cell is shown superimposed on a magnified section of the image. Acronyms of the algorithms to be used throughout this thesis include: Area Moments (AM), Correlation and Ellipse Fitting (CEF), Gabor Filters (GF) and Least Squares Ellipse Fitting (LSEF).

Results are presented in units of degrees for orientation measured counter-clockwise from the horizontal, cycles per image for major and minor axis length in the power spectrum, aspect ratio and pixels for major and minor axis length in the spatial image. *Cycles per image* is the original metric determined and *pixels* (actual pixel length) is computed based on the orientation and the image size. The orientation refers to the angle, in degrees, of the major axis of the composite cell in a spatial image. In each table that displays the results, the algorithm that best approximates the actual values will be italicized.

## 5.2 Sinusoidal Image

A 2d sinusoidal image presents an opportunity to introduce the algorithms on an image with simple statistics. While the images do not conform to the assumption of being elliptically shaped, they are certainly stationary. The results from applying the algorithms to this image are shown in Table 5.1.

A graphical representation of the results are shown in Figure 5.1. Here, a

| Method | $\theta$ | $L_{major}$ (cpi) | $L_{minor}$ (cpi) | $\kappa$ | $L_{major}$ (pixels) | $L_{minor}$ (pixels) |
|---|---|---|---|---|---|---|
| *AM* | *90.0* | *16.0* | *32.0* | *2.0* | *16.0* | *8.0* |
| CEF | 0.0 | 1.0 | 16.0 | 16.0 | 256.0 | 16.0 |
| *GF* | *90.0* | *16.0* | *32.0* | *2.0* | *16.0* | *8.0* |
| *LSEF* | *90.0* | *16.0* | *32.0* | *2.0* | *16.0* | *8.0* |
| Actual | 90 | 16 | 32 | 2.0 | 16.0 | 8.0 |

Table 5.1: 2d sinusoidal image test results. Results are shown for a 2d sinusoidal image with one sinusoid of 32cpi at 0° and a second at 16cpi at 90°. Actual values to manually measured.

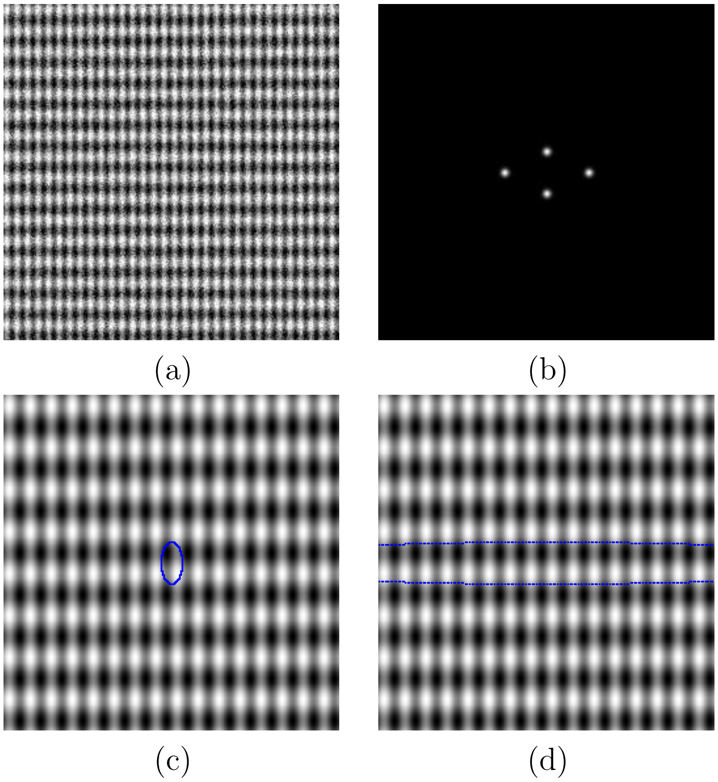


Figure 5.1: Graphical results for the 2d sinusoidal image. The original image is shown in (a), along with the power spectrum in (b). The composite cell is shown superimposed on the noiseless sinusoid for each of the algorithms: AM, GF and LSEF (c), and CEF (d). The image has been magnified by a factor of 2.

composite cell is superimposed on the actual image. Since the cellular elements are quite small, the composite cell is shown superimposed on a magnified portion of the image. To demonstrate what is happening in the spatial-frequency domain, the ellipses that best fit the power spectrum are shown super-imposed on the spectrum in Figure 5.2.

As can be seen, three of the algorithms (AM, GF and LSEF) were able to correctly estimate the bulk statistics. The CEF method was able to correctly

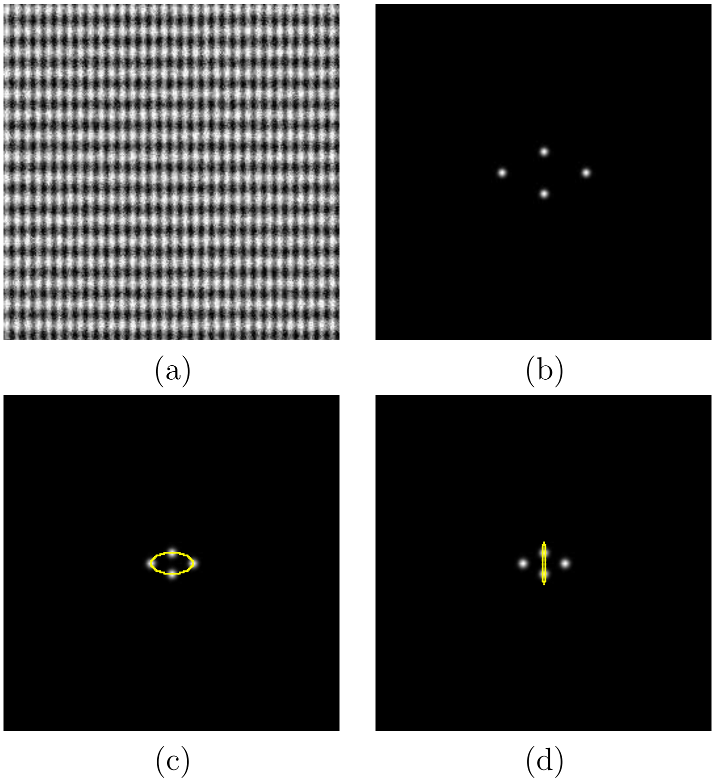


Figure 5.2: Spatial-frequency graphical results for the 2D sinusoidal image. The original image is shown in (a), along with the power spectrum in (b). The composite cell is shown superimposed on the power spectrum of the image for each of the algorithms: AM, GF and LSEF (c), and CEF (d). The power spectra in (c) and (d) are shown magnified by a power of 2.

identify the minor axis length, but assumed this to be the major axis.

The AM method uses the statistics of the whole image to obtain an axis of rotation, treating the power spectrum like a physical object. With only four points in the power spectrum, the area moments was able to easily identify the axis of rotation, which corresponds to the orientation.

In estimating the orientation, the CEF method attempts to maximize a correlation measure through a series of rotations. The algorithm, however, attempts to maximize the per pixel energy. This resulted in the best-ellipse going through only two of the points–those closest together and on the same axis. Expanding the ellipse to the other two points, along the u-axis, in the power spectrum actually resulted in a lower energy per pixel. For that reason, this method identified the orientation of the minor axis and its length, but not that of the major axis.

In terms of its ability to measure the orientation, the GF algorithm is actually well-suited for images with clearly defined and approximately perpendicular patterns. The method seeks out the highest value and temporarily assigns the orientation estimate to that value. It then looks along the perpendicular for the ”next highest” value, which, if responsive to a higher frequency, is assigned to the major axis length and the permanent orientation metric. The LSEF technique is also well-suited to operating on images with strong signals in the power spectrum in the presence of a small amount of high frequency noise.

## 5.3 Brodatz Images

The Brodatz images chosen for this part of testing are important in demonstrating the approach that each algorithm takes in estimating the bulk statistics. The Brodatz images are well-structured and, as a result, their power spectra have energy concentrated at a relatively small number of points, although they are still significantly more dispersed than the simple two-signal sinusoidal image. It must also be observed that while these Brodatz images satisfy the first assumption listed in Subsection 3.2.1 – that the images are approximately stationary – they do not satisfy the assumption of having individual cellular elements that are approximately elliptical.

The results of applying the algorithms to the Brodatz images are shown in the tables that follow: Brick (D95) (Table 5.2), Rotated Brick (D95) (Table 5.3) and Woven Aluminum Wire (D1) (Table 5.4). Shown in Figure 5.3 are the graphical results associated with the Brodatz Brick (D95) image.

The actual statistics associated with the composite cell of each image are shown in their respective tables. Since no such information exists for the actual bulk

| Method | $\theta$ | $L_{major}$ | $L_{minor}$ | $\kappa$ | $L_{major}$ | $L_{minor}$ |
|---|---|---|---|---|---|---|
| | | (cpi) | (cpi) | | (pixels) | (pixels) |
| AM | -2.3 | 7.0 | 11.0 | 1.6 | 36.6 | 23.3 |
| *CEF* | *0.0* | *2.0* | *10.0* | *5.0* | *128.0* | *25.6* |
| *GF* | *0.0* | *7.2* | *10.0* | *1.4* | *35.6* | *25.6* |
| LSEF | 0.4 | 9.9 | 24.1 | 2.1 | 25.6 | 10.6 |
| Actual | 0 | 4 | 10 | 2.4 | 64.0 | 25.6 |

Table 5.2: Brodatz Brick (D95) test results. Actual values to manually measured.

statistics of the images presented, the values shown as *actual* were, in fact, measured manually. The best-fit ellipses are also shown super-imposed on the image in Figure 5.3 and on power spectrum in Figure 5.4.

As can be seen, the orientation was measured with reasonable accuracy by each of the algorithms. With the exception of the LSEF algorithm, each algorithm was able to produce a reasonable estimate of the major axis length. However, obtaining a consistent measure of the minor axis length proved challenging. The AM and GF algorithms estimated a minor length that was approximately twice that of the actual, indicating that these algorithms were perhaps identifying an aliased version of the desired signal.

The CEF algorithm, which attempts to maximize the energy per pixel of an ellipse fit to the power spectrum, identified 2cpi as the minor axis length. While this is not correct, it is easily explained. While not as strong a signal as at 4cpi, there is a signal at 2cpi associated with the large bricks. Since the ellipse that intersects that signal would be smaller, it would also require less energy at 2cpi to generate the same, or greater, energy per pixel that an 4cpi.

The LSEF method, which attempts to characterize the bulk statistics of an image based on the overall shape of its power spectrum, was unable to identify either of the axis lengths. By minimizing the sum of the squares of the distances between the ellipse and the energy contributions at all pixels, this method is ill-suited to identifying specific signals in a spectrum with additional energy located relatively far from those signals this work is looking to identify.

The results from running the algorithms on the rotated Brodatz Brick image

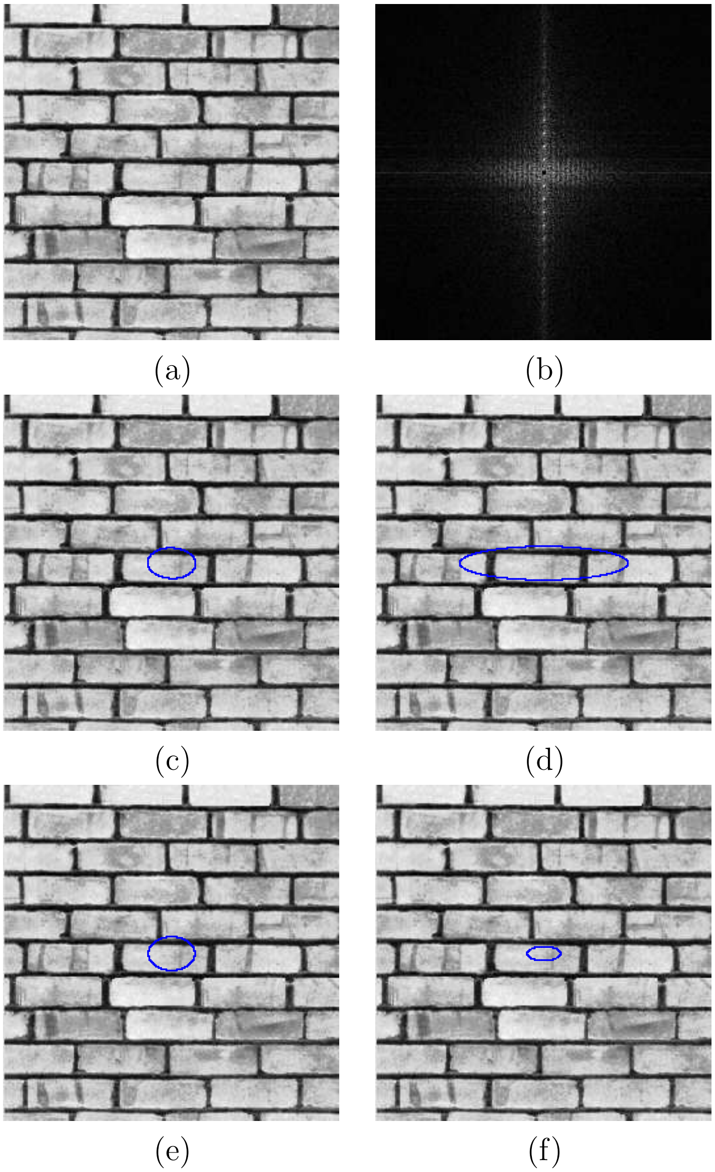
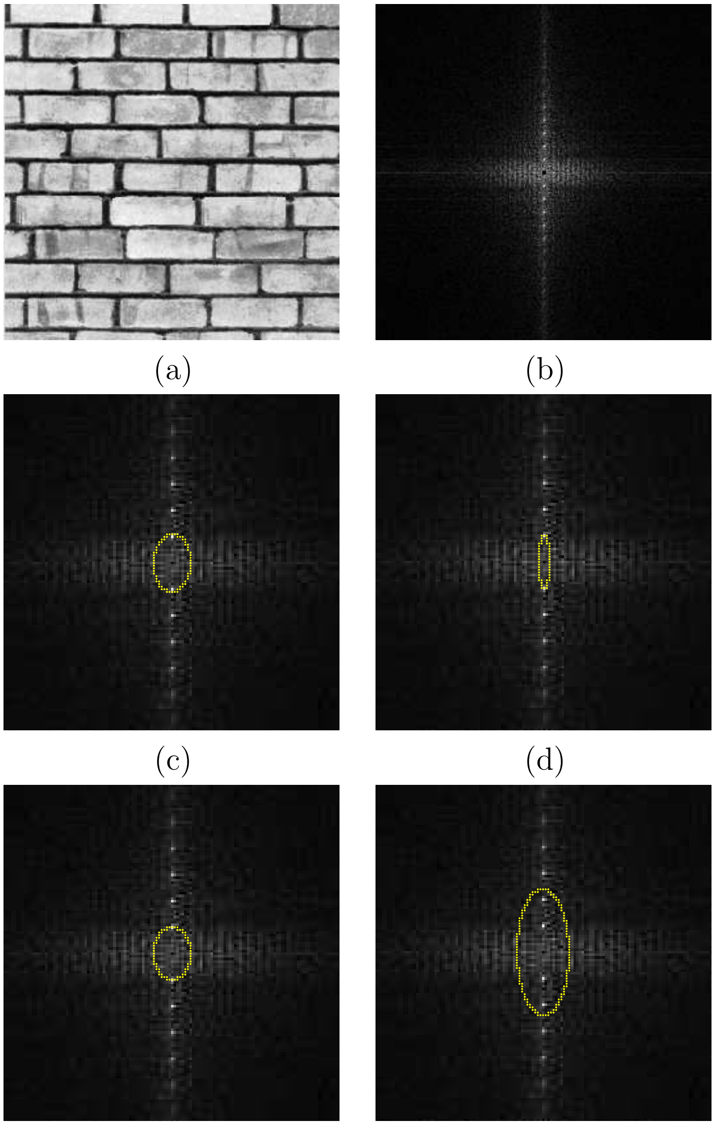


Figure 5.3: Graphical results for the Brodatz brick (D95) image. The image (a), the power spectrum (b) and the composite cell is shown for each of the algorithms (AM (c), CEF (d), GF (e) and LSEF (f)).



(a) (b)

(c) (d)

(e) (f)

Figure 5.4: Spatial-frequency graphical results for the Brodatz Brick (D95) image. The original image is shown in (a), along with the power spectrum in (b). The composite cell is shown super-imposed on the power spectrum for each of the algorithms (AM (c), CEF (d), GF (e) and LSEF (f)). The power spectra in (c)-(f) are shown magnified by a power of 2.

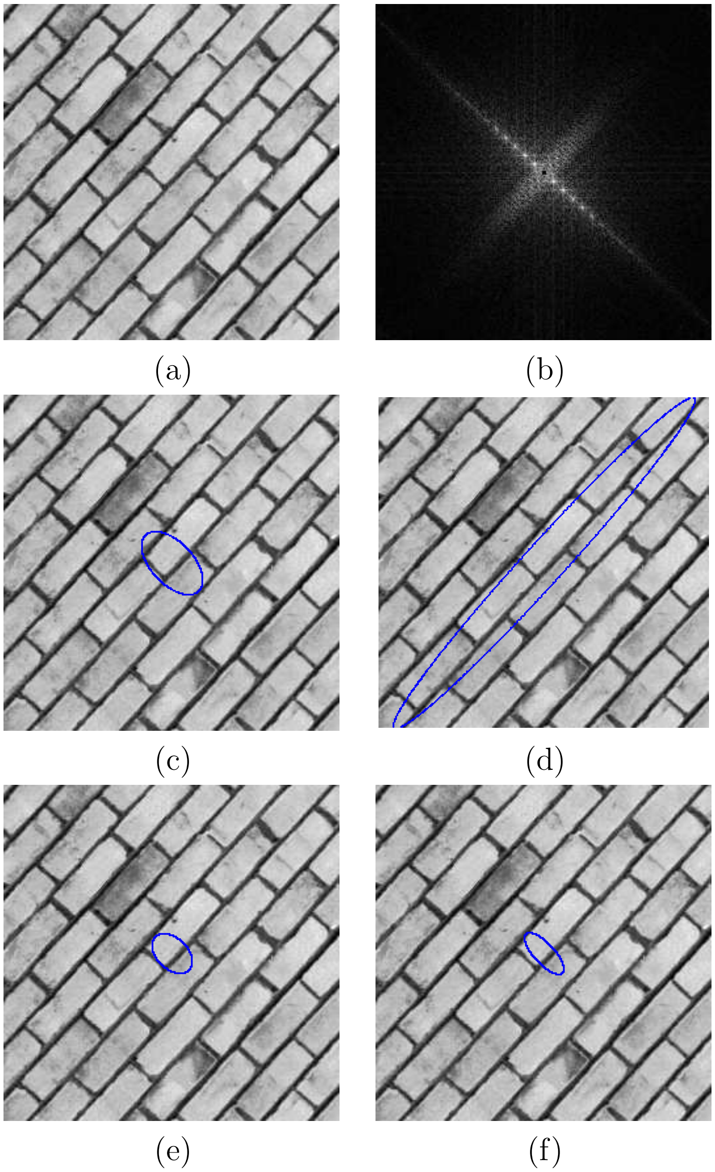
| Method | $\theta$ | $L_{major}$ | $L_{minor}$ | $\kappa$ | $L_{major}$ | $L_{minor}$ |
|---|---|---|---|---|---|---|
| | | (cpi) | (cpi) | | (pixels) | (pixels) |
| AM | -46.5 | 6.0 | 11.0 | 1.8 | 58.8 | 32.1 |
| *CEF* | *47.7* | *1.0* | *11.0* | *2.5* | *346.1* | *31.5* |
| GF | -44.1 | 9.9 | 15.2 | 1.5 | 36.0 | 23.5 |
| LSEF | -47.1 | 8.5 | 22.7 | 2.7 | 41.1 | 15.4 |
| Actual | 47.5 | 5.4 | 13.6 | 2.7 | 64.0 | 25.6 |

Table 5.3: Rotated Brodatz Brick (D95R) test results.

(D95R), which is rotated by 47.5°, are shown in Table 5.3. Figure 5.5 depicts the graphical results associated with these results. The best-fit ellipses are also shown super-imposed on the power spectrum in Figure 5.6.

As can be seen in the results shown in Table 5.3, only the CEF algorithm was able to correctly identify the major axis orientation. However, each of the other algorithms identified the axis perpendicular to the orientation (i.e. the minor axis orientation). In this example, none of the algorithms correctly identified the minor axis length, but the CEF algorithm was close. Although the AM and GF techniques appears to have correctly identified the minor axis length in Table 5.3, since those techniques calculated the orientation as offset by 90° from the actual orientation, this metric corresponds with the major axis length. As such, it is incorrect.

In this test case, there is a strong signal associated with the minor axis in the power spectrum. However, the increase in the number of periods along the minor axis (due to the rotation away from the shorter horizontal and vertical axes) increases the strength of those signal relatives to the major axis. This explains why the algorithms fared less well than in the unrotated Brick (D95) image in determining both the orientation metrics and the axis lengths.



(a) (b)

(c) (d)

(e) (f)

Figure 5.5: Graphical results for the rotated Brodatz Brick (D95R) image. The image (a), the power spectrum (b) and the composite cell is shown for each of the algorithms (AM (c), CEF (d), GF (e) and LSEF (f)).

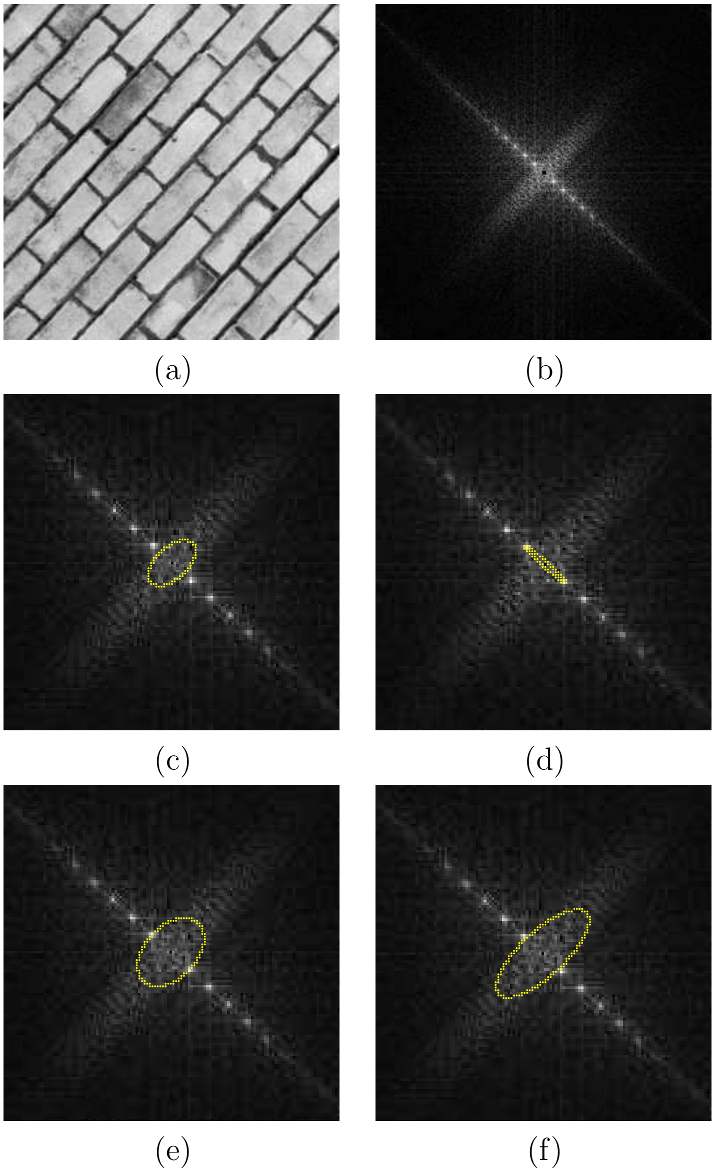


Figure 5.6: Spatial-frequency graphical results for the rotated Brodatz Brick (D95R) image. The original image is shown in (a), along with the power spectrum in (b). The composite cell is shown super-imposed on the power spectrum for each of the algorithms (AM (c), CEF (d), GF (e) and LSEF (f)). The power spectra in (c)-(f) are shown magnified by a power of 2.

| Method | $\theta$ | $L_{major}$ | $L_{minor}$ | $\kappa$ | $L_{major}$ | $L_{minor}$ |
|---|---|---|---|---|---|---|
| | | (cpi) | (cpi) | | (pixels) | (pixels) |
| AM | 2.8 | 16.0 | 26.0 | 1.6 | 16.0 | 9.9 |
| *CEF* | *0.0* | *9.0* | *13.0* | *1.4* | *28.4* | *19.7* |
| GF | 89.9 | 42.6 | 50.0 | 1.2 | 6.0 | 5.1 |
| LSEF | -4.8 | 41.5 | 53.8 | 1.3 | 6.2 | 4.8 |
| Actual | 0 | 8 | 13.3 | 1.7 | 32.0 | 19.2 |

Table 5.4: Brodatz Woven Aluminum Wire (D1) test results.

The results of running the algorithms on the third Brodatz image (D1), are shown in Table 5.4. The orientation was estimated reasonably well by the AM, CEF and LSEF algorithms. The CEF algorithm was the only algorithm to approximate the actual length statistics. The graphical results associated with the Brodatz Woven Aluminum Wire (D1) image are shown in Figure 5.7. The best-fit ellipses are also shown super-imposed on the power spectrum in Figure 5.8.

This texture was unique in that it presented a second pattern of 16cpi in the direction perpendicular to the dominant orientation. This can be seen as the white divide in the rectangular shapes of the spatial image. The AM method was able to identify this, but was unsuccessful in identifying a strong signal along the relatively noisy, dominant v-axis of the power spectrum. Neither the GF nor the LSEF methods produced any meaningful measure of cellular element length. The Woven Aluminum Wire image has, as can be observed in its power spectrum (Figure 5.7(b)), a strong component along the diagonals. With the LSEF attempting to minimize the sum of the square of the distance between a *best ellipse* and the energy in the power spectrum, the dispersed power spectrum and the isolated energy away from the statistics the algorithms were designed to characterize resulted in answers

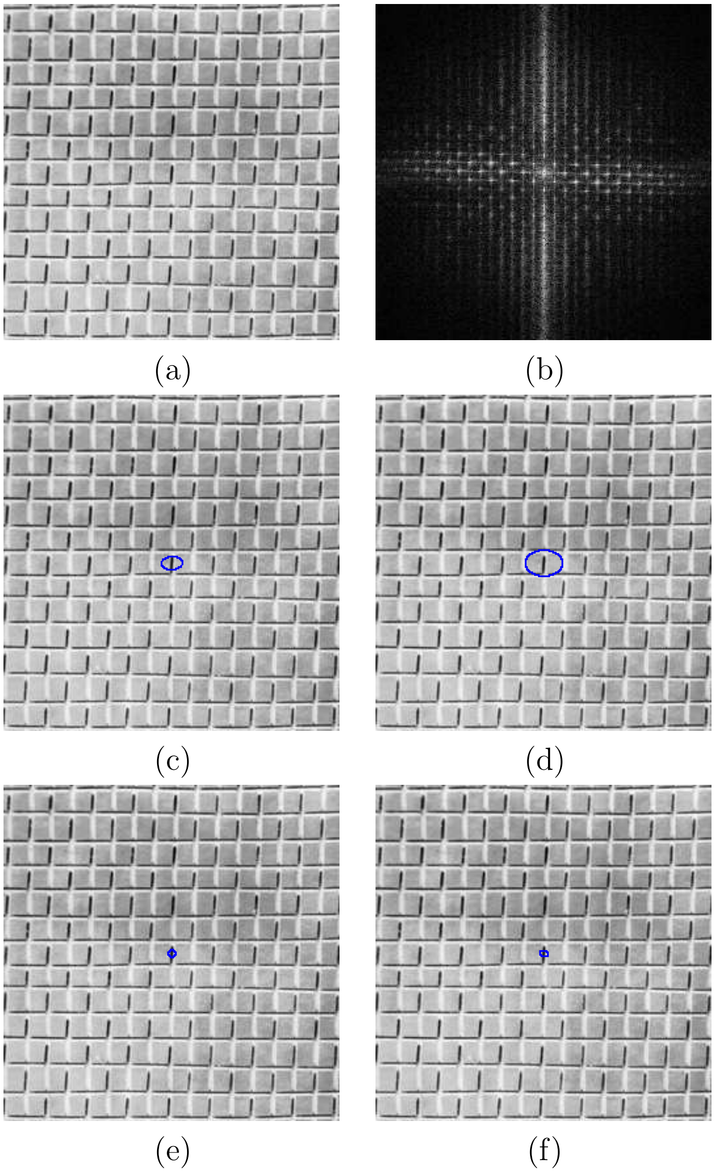


Figure 5.7: Graphical results for the Brodatz Woven Aluminum Wire (D1) image. The image (a), the power spectrum (b) and the composite cell is shown for each of the algorithms (AM (c), CEF (d), GF (e) and LSEF (f)).

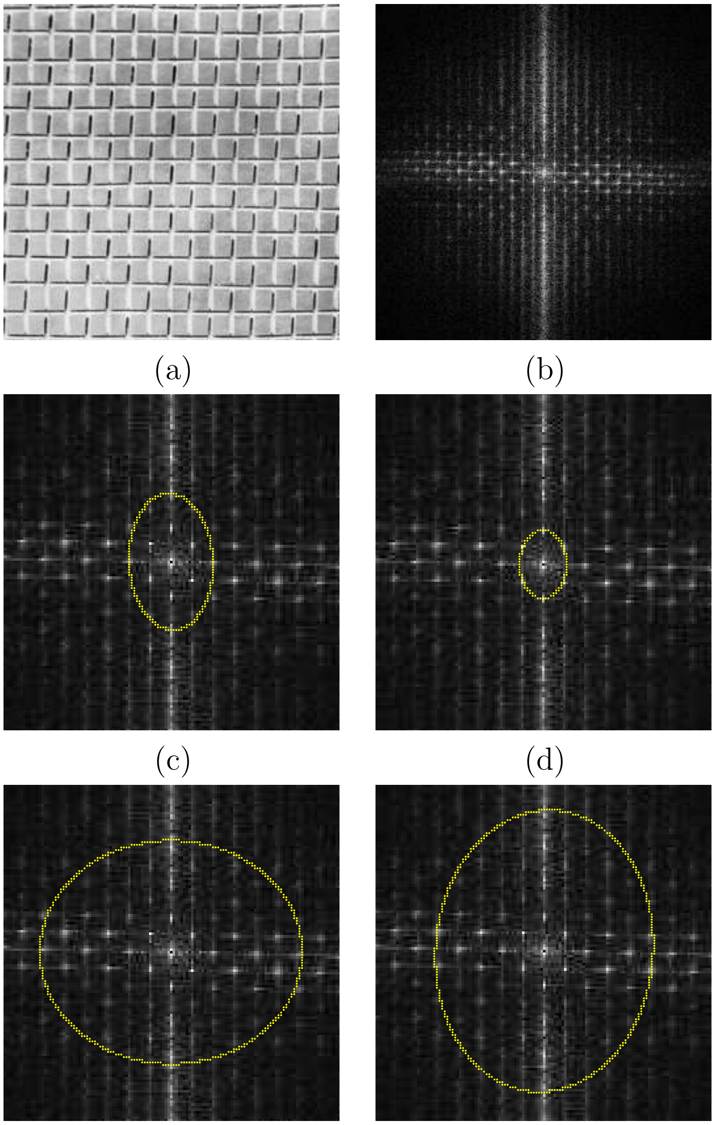


Figure 5.8: Spatial-frequency graphical results for the Brodatz Woven Aluminum Wire (D1). The original image is shown in (a), along with the power spectrum in (b). The composite cell is shown super-imposed on the power spectrum for each of the algorithms (AM (c), CEF (d), GF (e) and LSEF (f)). The power spectra in (c)-(f) are shown magnified by a power of 2.

uncharacteristic of the spatial image statistics.

The GF algorithm incorrectly identified both the orientation and the axis lengths. The reason that the axis lengths were identified was simply that the power spectrum contained a large number of individual strong signals that are spread out over much of the spectrum. At higher frequencies, the filters were capturing energy from more than one signal and each contributed to creating a larger response. The fault in determining the orientation was a result of the GF algorithm's method for identifying its composite cell axis lengths. The method identifies the centre frequency of the filter at which the maximum response is measured. This pixel has a characteristic length $L_1$ and orientation $\theta$. The algorithm then searches along the ray from originating at the image centre at an orientation of $\theta+90$, the strongest response at which is assigned to length $L_2$. The larger of these two values ($max(L_1, L_2)$) corresponds with the major axis of the ellipse best describing the shape of the energy in the power spectrum. In this case, a stronger response happened to correspond with the shorter power spectrum ellipse axis. As a result, the orientation is incorrect, but not completely uninformative.

In these samples, and in the much larger set of images used during testing and algorithm development, the orientation, or its perpendicular, could be consistently measured within a few degrees of the observable textural alignment in well-structured images by one or more of the algorithms. It is important to note that the Brodatz images considered in this section clearly do not satisfy the assumption of having elliptical cellular elements.

## 5.4 Artificial Cellular Data

Synthetic meshes representing cells in an epithelium present an opportunity to test the algorithms on cellular elements with well-defined boundaries in a relatively noiseless environment. The images also satisfy the assumptions in that the cellular elements are approximately elliptical and stationary.

The results of applying the algorithms to this artificial cellular data will be shown as part of two data sets. The first collection of images have cellular elements with fixed aspect ratios but varying orientation. The second set of images have a fixed orientation with varying aspect ratios. The bulk statistics of the images are known. However, the known values are not the major and minor axes lengths, but rather, the aspect ratio (the ratio of the two lengths). Due to the very small size and the irregular shape of the cells, manually measuring the lengths and orientations is not only extremely difficult, but would also have a significant variability associated with the accuracy. A single example will be shown in which the estimated values are shown superimposed on the actual images. All other images are shown in Appendix A. Only the composite cell shown superimposed on the spatial image is shown in the appendix.

Table 5.5 shows the results from the first data set, in which cellular elements have a consistent aspect ratio of 1.8, but varying orientations from 0° to 45°. Shown in Figure 5.9 are the graphical results of the image with $\theta$=20° and $\kappa$=1.8. The best-fit ellipses are also shown super-imposed on the power spectrum in Figure 5.10.

Whereas the algorithms operating on the Brodatz data set were capable of consistently estimating the dominant orientation (or its perpendicular), such regularity

Image One

| Method | $\theta$ | $L_{major}$ | $L_{minor}$ | $\kappa$ | $L_{major}$ | $L_{minor}$ |
|---|---|---|---|---|---|---|
| | | (cpi) | (cpi) | | (pixels) | (pixels) |
| AM | -0.5 | 32.0 | 65.0 | 2.0 | 7.4 | 3.6 |
| *CEF* | *0.0* | *31.0* | *57.0* | *1.8* | *7.6* | *4.1* |
| GF | -88.9 | 64.0 | 64.0 | 1.0 | 3.7 | 3.7 |
| LSEF | 1.4 | 70.9 | 85.7 | 1.2 | 3.3 | 2.8 |
| Actual | 0.1 | n/a | n/a | 1.8 | n/a | n/a |

Image Two

| Method | $\theta$ | $L_{major}$ | $L_{minor}$ | $\kappa$ | $L_{major}$ | $L_{minor}$ |
|---|---|---|---|---|---|---|
| | | (cpi) | (cpi) | | (pixels) | (pixels) |
| AM | 3.0 | 26.0 | 54.0 | 2.1 | 8.8 | 4.2 |
| *CEF* | *6.7* | *28.0* | *50.0* | *1.8* | *8.2* | *4.6* |
| GF | -81.0 | 55.0 | 62.0 | 1.1 | 4.2 | 3.7 |
| LSEF | -0.4 | 70.0 | 82.0 | 1.2 | 3.3 | 2.8 |
| Actual | 10.1 | n/a | n/a | 1.8 | n/a | n/a |

Image Three

| Method | $\theta$ | $L_{major}$ | $L_{minor}$ | $\kappa$ | $L_{major}$ | $L_{minor}$ |
|---|---|---|---|---|---|---|
| | | (cpi) | (cpi) | | (pixels) | (pixels) |
| AM | 8.7 | 28.0 | 38.0 | 1.4 | 8.2 | 6.0 |
| *CEF* | *16.9* | *25.0* | *45.0* | *1.8* | *9.4* | *5.2* |
| GF | -72.6 | 50.6 | 60.0 | 1.2 | 4.7 | 3.9 |
| LSEF | -4.3 | 68.8 | 79.8 | 1.2 | 3.3 | 2.8 |
| Actual | 20.3 | n/a | n/a | 1.8 | n/a | n/a |

Image Four

| Method | $\theta$ | $L_{major}$ | $L_{minor}$ | $\kappa$ | $L_{major}$ | $L_{minor}$ |
|---|---|---|---|---|---|---|
| | | (cpi) | (cpi) | | (pixels) | (pixels) |
| AM | -51.7 | 23.0 | 36.0 | 1.6 | 12.8 | 8.2 |
| *CEF* | *40.0* | *22.0* | *40.0* | *1.8* | *13.7* | *7.5* |
| GF | -49.3 | 43.6 | 62.0 | 1.4 | 7.0 | 4.9 |
| LSEF | -35.6 | 75.2 | 78.5 | 1.0 | 3.8 | 3.6 |
| Actual | 45.1 | n/a | n/a | 1.8 | n/a | n/a |

Table 5.5: Synthetic mesh data, fixed aspect ratio results. The aspect ratio is fixed at 1.8 while the orientation varies between 0° and 45°. Known values for this data are $\theta$ and $\kappa$. $L_{major}$ and Ł$_{minor}$ were not measured due to the uncertainty that would have been associated with manually measuring such a large number of small cells.

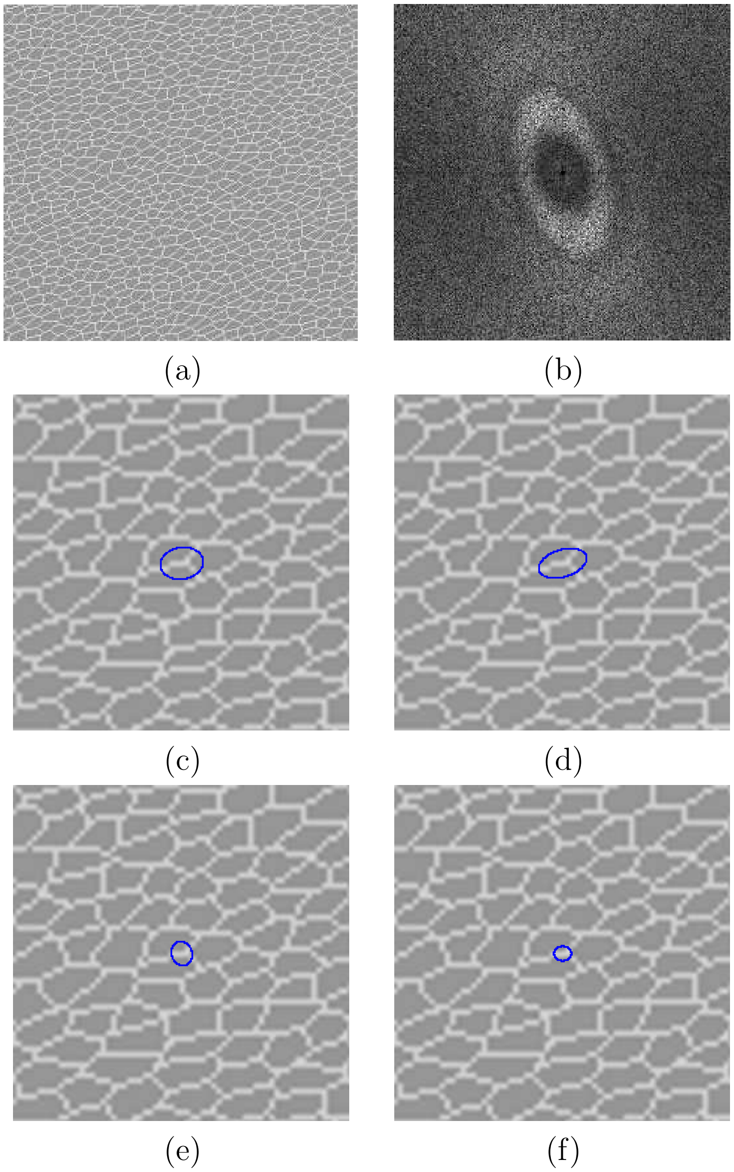


Figure 5.9: Graphical results for artificial cellular data with $\theta$=20° and $\kappa$=1.8 (Image #3 from Table 5.5). The original image is shown in (a), along with the power spectrum in (b). The composite cell is shown superimposed on the image (magnified by a factor of 4) for each of the algorithms: AM (c), CEF (d), GF (e) and LSEF (f).

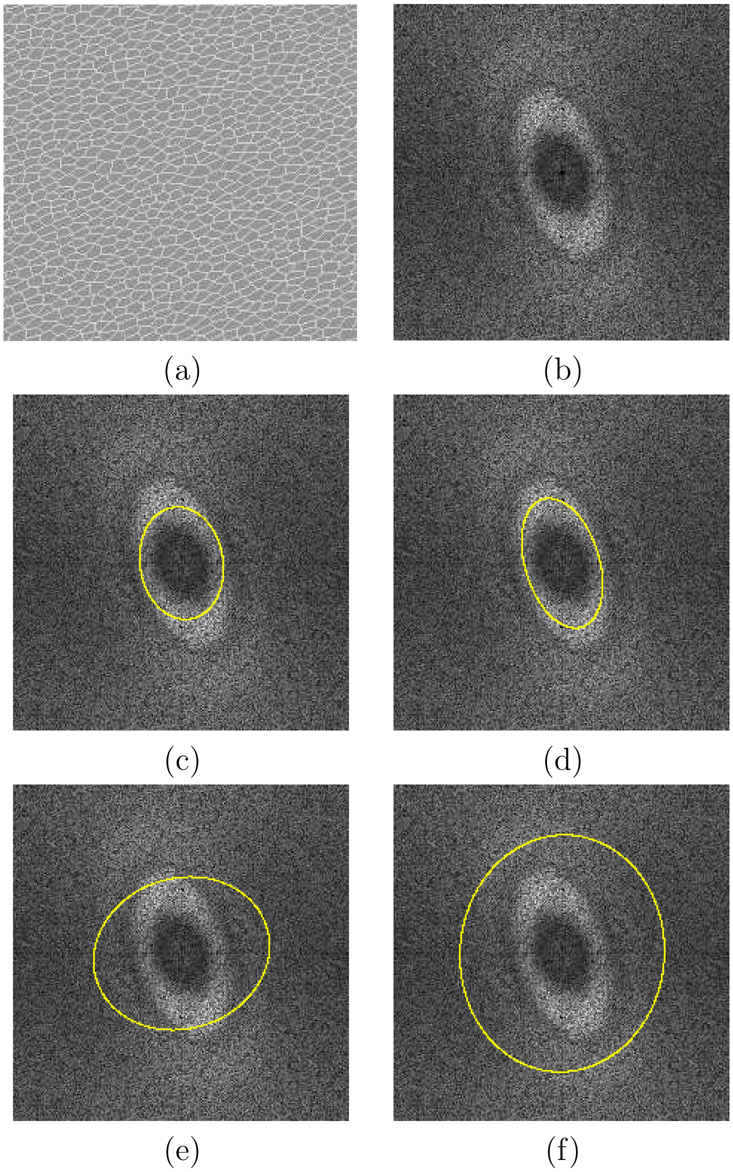


Figure 5.10: Spatial-frequency graphical results for the Brodatz Woven Aluminum Wire (D1). The original image (Image #3 from Table 5.5) is shown in (a), along with the power spectrum in (b). The composite cell is shown super-imposed on the power spectrum for each of the algorithms (AM (c), CEF (d), GF (e) and LSEF (f)). The power spectra in (c)-(f) are shown magnified by a power of 2.

did not exist with the more varied artificially generated cellular images. As is clear if the power spectrum in Figures 5.5(b) and 5.9(b) are compared, the power spectra of the synthetic cells were much less concentrated than those computed from the Brodatz images. This is an important observation because the appearance (distribution) of the power spectrum significantly impacts the effectiveness of an algorithm in estimating the bulk statistics.

As can be seen from the results of Table 5.5, the methods capable of measuring the orientation varied from case to case. However, it can be stated that the CEF algorithm was able to identify the orientation with relative consistency, and the GF algorithm routinely estimated a reasonable measure of the angle perpendicular to the actual orientation. The LSEF and AM methods, while closer to one of the axes' orientations than not, were inconsistent in their estimates. The problems associated with the LSEF algorithm were unique. The LSEF seeks to minimize the sum of the squares of the distances between points on the ellipse and those in the image. However, the information contained at the outer edges of the power spectrum contributed relatively more than those in the elliptical band we were seeking to characterize. As a result, the "best ellipse" fit to the data tended to be pulled outward toward the sides of the power spectrum, resulting in an almost circular characterization in every example. Due to this, the LSEF algorithm did not produce meaningful results.

The GF method, like the LSEF method, was drawn towards the outer areas of the images by the high frequency information in the power spectrum, and as a result, measures of axis lengths were consistently high. Similarly, the AM method was attempting to physically, and then statistically, describe a shape that was too dispersed to accurately distinguish.

The results from testing the second artificial data set is shown in Table 5.6. In this example, the orientations are fixed at 30° and aspect ratios vary from 1.0 (i.e. no dominant orientation) to 2.3.

With respect to performance, the results shown in Table 5.6 match closely with those in Table 5.5. The CEF method performed quite well, only deviating away from the known aspect ratio in a couple test cases. Once again, the GF algorithm was able to correctly identify the orientation, but consistently generated poor estimates related to axis length. The inability of the AM and LSEF methods to identify either the orientation or the aspect ratio suggest that these techniques are poorly suited to images with large variation in the individual cellular elements. Graphical results of the composite cell shown superimposed on the spatial image are shown in Appendix A.

Regarding the capacity of the algorithms to accurately estimate the bulk statistics, in which we only have the orientation ($\theta$) and aspect ratio ($\kappa$) on which to base algorithm performance, only the CEF method produced consistent results.

Image One

| Method | $\theta$ | $L_{major}$ | $L_{minor}$ | $\kappa$ | $L_{major}$ | $L_{minor}$ |
|---|---|---|---|---|---|---|
| | | (cpi) | (cpi) | | (pixels) | (pixels) |
| AM | 0.1 | 39.0 | 41.0 | 1.1 | 6.1 | 5.8 |
| *CEF* | *-25.6* | *41.3* | *41.5* | *1.0* | *6.4* | *6.4* |
| GF | 79.2 | 50.3 | 64.0 | 1.3 | 4.8 | 3.8 |
| LSEF | 4.1 | 80.7 | 81.9 | 1.0 | 3.0 | 2.7 |
| Actual | 68.7 | n/a | n/a | 1.0 | n/a | n/a |

Image Two

| Method | $\theta$ | $L_{major}$ | $L_{minor}$ | $\kappa$ | $L_{major}$ | $L_{minor}$ |
|---|---|---|---|---|---|---|
| | | (cpi) | (cpi) | | (pixels) | (pixels) |
| AM | -74.6 | 32.0 | 33.0 | 1.0 | 7.6 | 7.4 |
| *CEF* | *22.4* | *28.0* | *43.0* | *1.5* | *9.1* | *5.9* |
| GF | -65.3 | 47.1 | 62.0 | 1.3 | 5.5 | 4.2 |
| LSEF | -9.4 | 75.6 | 81.2 | 1.1 | 3.2 | 2.9 |
| Actual | 30.3 | n/a | n/a | 1.5 | n/a | n/a |

Image Three

| Method | $\theta$ | $L_{major}$ | $L_{minor}$ | $\kappa$ | $L_{major}$ | $L_{minor}$ |
|---|---|---|---|---|---|---|
| | | (cpi) | (cpi) | | (pixels) | (pixels) |
| AM | -74.0 | 25.0 | 27.0 | 1.1 | 9.3 | 8.6 |
| *CEF* | *23.5* | *21.0* | *45.0* | *2.1* | *11.6* | *5.4* |
| GF | -61.6 | 43.9 | 60.0 | 1.4 | 5.8 | 4.2 |
| LSEF | -9.8 | 67.0 | 77.8 | 1.2 | 3.4 | 2.9 |
| Actual | 30.0 | n/a | n/a | 1.9 | n/a | n/a |

Image Four

| Method | $\theta$ | $L_{major}$ | $L_{minor}$ | $\kappa$ | $L_{major}$ | $L_{minor}$ |
|---|---|---|---|---|---|---|
| | | (cpi) | (cpi) | | (pixels) | (pixels) |
| AM | -73.9 | 23.0 | 36.0 | 1.6 | 10.4 | 6.6 |
| *CEF* | *26.9* | *20.0* | *41.0* | *2.1* | *12.8* | *6.3* |
| GF | -63.3 | 44.6 | 62.0 | 1.4 | 5.7 | 4.1 |
| LSEF | -10.4 | 65.2 | 80.5 | 1.2 | 3.6 | 2.9 |
| Actual | 30.0 | n/a | n/a | 2.3 | n/a | n/a |

Table 5.6: Synthetic mesh data, fixed orientation results. The orientation is fixed at $30^{circ}$ while the aspect ratio varies between 1.0 and 2.3. $L_{major}$ and $Ł_{minor}$ were not measured due to the uncertainty that would have been associated with manually measuring such a large number of small cells.

| Method | $\theta$ | $L_{major}$<br>(cpi) | $L_{minor}$<br>(cpi) | $\kappa$ | $L_{major}$<br>(pixels) | $L_{minor}$<br>(pixels) |
|---|---|---|---|---|---|---|
| AM | 90.0 | 5.0 | 6.0 | 1.2 | 90.0 | 75.0 |
| CEF | 97.9 | 5.0 | 7.0 | 1.4 | 90.9 | 64.9 |
| GF | 85.0 | 6.0 | 6.0 | 1.0 | 75.3 | 75.3 |
| *LSEF* | *93.8* | *7.8* | *13.4* | *1.7* | *56.4* | *33.7* |
| Actual | 85 | 8 | 13 | 1.6 | 56.5 | 34.7 |

Table 5.7: Natural cellular data with elongated cells results.

## 5.5 Natural Cellular Imagery

The final test set belongs to actual images of epithelial cells. In the synthetic meshes considered in the previous section, the individual cellular elements were created by linking each cell's control points with a line. The result was a test image with ideal illumination and an absence of high frequency noise. This data set, however, does not possess such ideal characteristics.

The graphical results associated with the natural cellular image with elongated cells are shown in Figure 5.11. The best-fit ellipses are also shown super-imposed on the power spectrum in Figure 5.12.

The elongated cells depicted in Table 5.7 and in the Figures 5.11 and 5.12, despite having poorly defined boundaries, have an apparent orientation approximately along the y-axis. This corresponds to an elongation of the power spectrum distribution in the u-direction. Since the bulk statistics for this imagery are not known *a priori*, the actual results are only estimates measured manually. As can be seen from the results, there is a large variation in the estimates derived to characterize the image. The orientation values match approximately with the actual estimates.

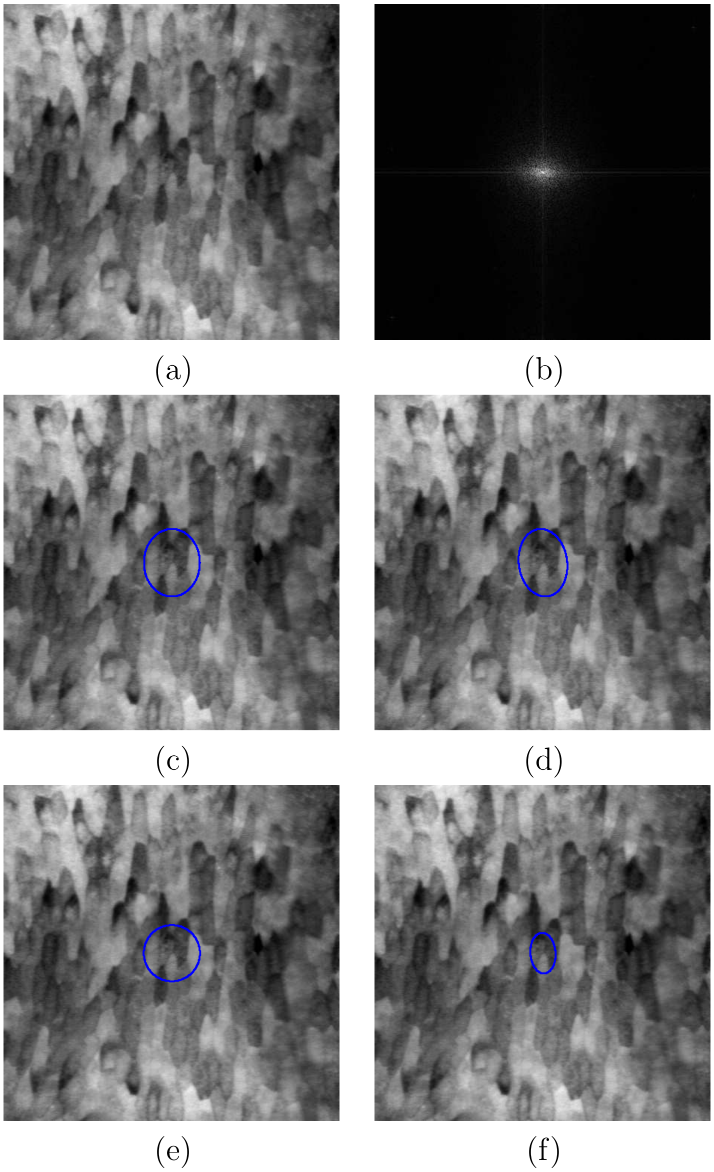
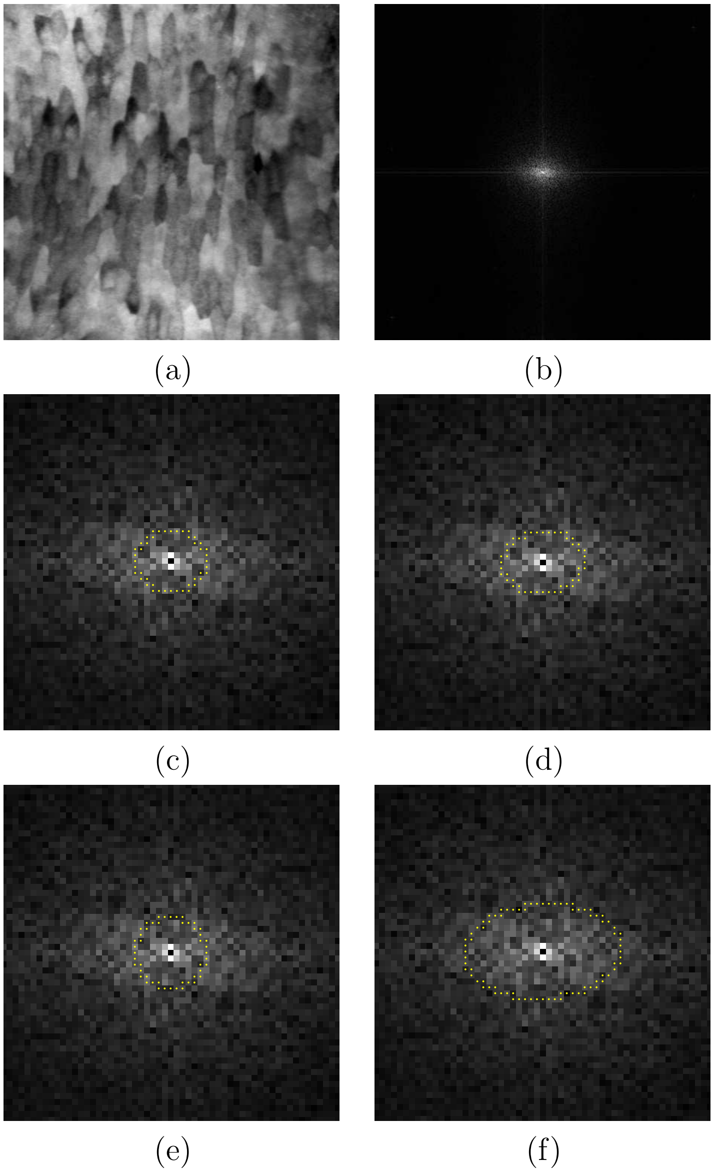


Figure 5.11: Graphical results for the natural cellular image with elongated cells. The image (a), the power spectrum (b) and the composite cell is shown for each of the algorithms (AM (c), CEF (d), GF (e) and LSEF (f)).



(a) (b)

(c) (d)

(e) (f)

Figure 5.12: Spatial-frequency graphical results for the natural cellular image with elongated cells. The original image is shown in (a), along with the power spectrum in (b). The composite cell is shown super-imposed on the power spectrum for each of the algorithms (AM (c), CEF (d), GF (e) and LSEF (f)). The power spectra in (c)-(f) are shown magnified by a power of 4.

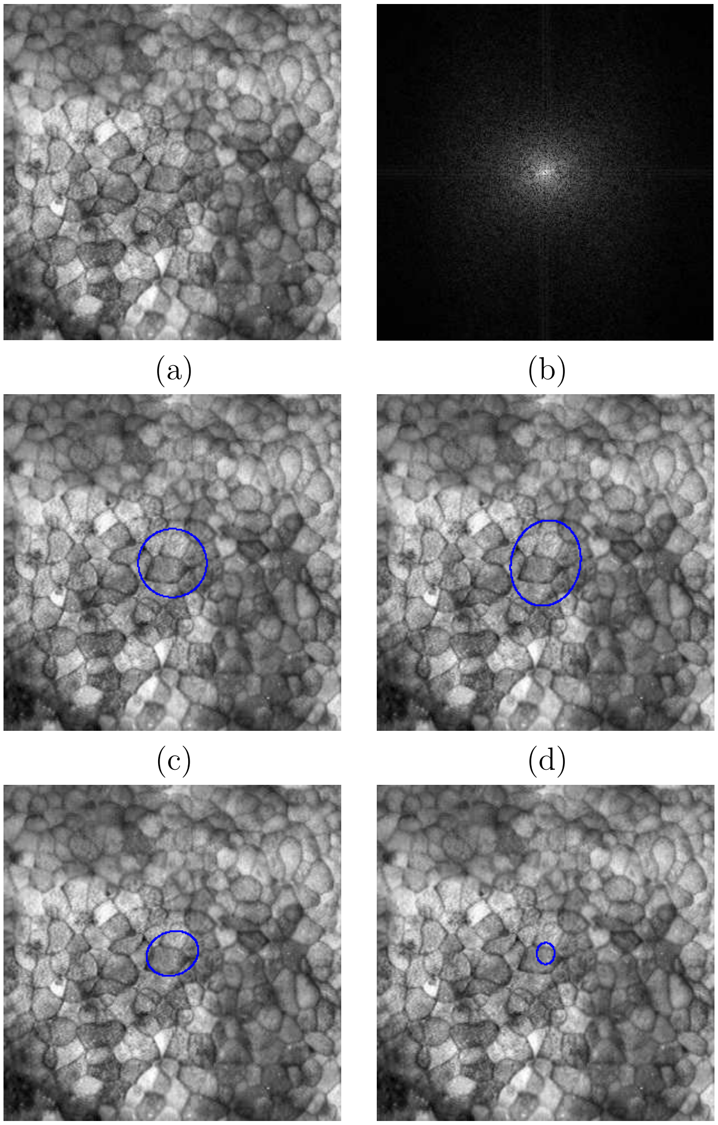
| Method | $\theta$ | $L_{major}$ (cpi) | $L_{minor}$ (cpi) | $\kappa$ | $L_{major}$ (pixels) | $L_{minor}$ (pixels) |
|---|---|---|---|---|---|---|
| AM | 9.3 | 5.0 | 5.0 | 1.0 | 61.4 | 61.4 |
| CEF | 77.0 | 4.0 | 5.0 | 1.3 | 77.7 | 62.2 |
| GF | 22.0 | 6.9 | 8.4 | 1.0 | 47.4 | 38.9 |
| *LSEF* | *88.4* | *15.7* | *19.0* | *1.2* | *19.3* | *16.0* |
| Actual | n/a | 17 | 17 | 1.0 | 17.8 | 17.8 |

Table 5.8: Natural cellular image with well-defined boundaries and no discernable stretching. In this case, there was no discernable stretching. As a result, an aspect ratio of approximately 1.0 was identified and the $L_{major}$ and $L_{minor}$ are recorded as being equal.

The length metrics for all of the algorithms except the LSEF over-estimated the actual cell's spatial metrics. The LSEF method was able to almost exactly measure the cell lengths. In the artificial cellular data, the spectra were very dispersed. This resulted in large amounts of high frequency noise that tended to pull the ellipse outward from what might be visually considered the best-fit ellipse. In this example, however, the vast majority of the information is contained very near the origin. Relatively, even with the squaring associated with the distance in the LSEF algorithm, the best-fit ellipse remained close to the concentrated energy and produced quite meaningful results.

The results from a natural image with well-defined boundaries are shown in Table 5.8. In this case, no strong orientation exists. The same accuracy in length associated with the LSEF algorithms was found in this example. The graphical results associated with well-defined boundaries are shown in Figure 5.13. The best-fit ellipses are also shown super-imposed on the power spectrum in Figure 5.14.

As can be seen from the power spectra in Figures 5.12 and 5.14, the hollow

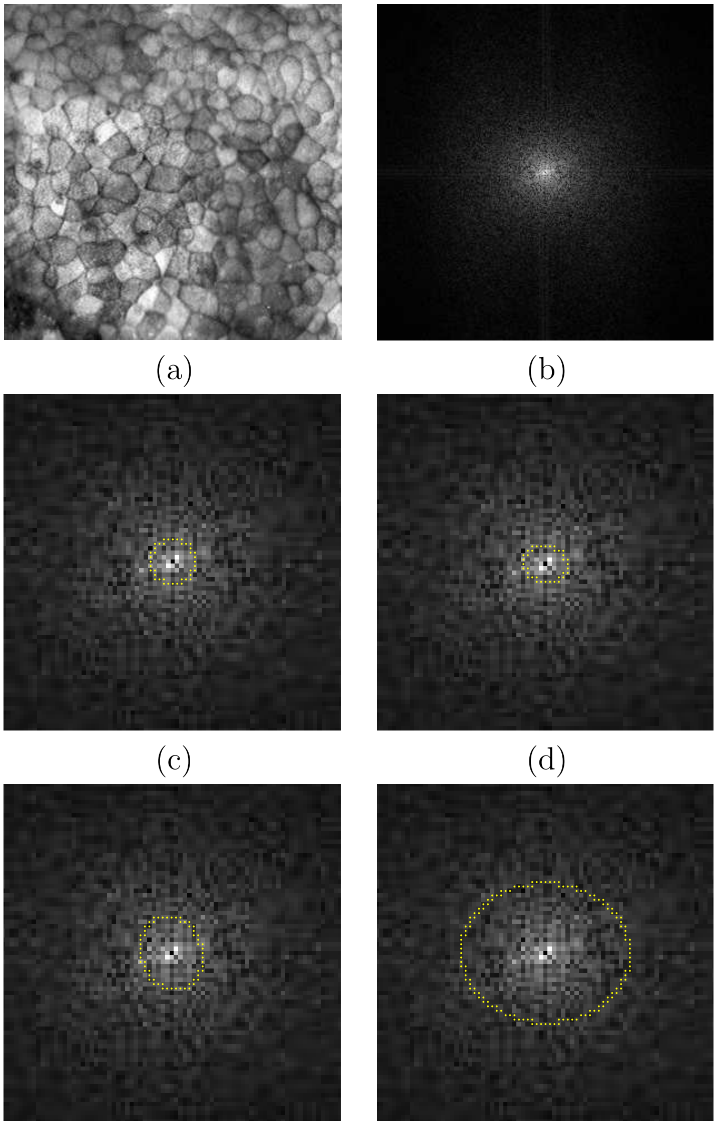


(a) (b)

(c) (d)

(e) (f)

Figure 5.13: Graphical results for the natural cellular image with well-defined boundaries. The image (a), the power spectrum (b) and the composite cell is shown for each of the algorithms (AM (c), CEF (d), GF (e) and LSEF (f)).



(a) (b)

(c) (d)

(e) (f)

Figure 5.14: Spatial-frequency graphical results for the natural cellular image with well-defined boundaries. The original image is shown in (a), along with the power spectrum in (b). The composite cell is shown super-imposed on the power spectrum for each of the algorithms (AM (c), CEF (d), GF (e) and LSEF (f)). The power spectra in (c)-(f) are shown magnified by a power of 3.

elliptical shapes in the power spectra that were present in the artificially generated data are not present in the natural imagery. This shape is better described as a Gaussian-like distribution centred on the origin. As a result, attempting to characterize the shape with an ellipse was unlikely to generate meaningful results. This explains why none of those algorithms (AM, CEF, GF) that attempted to directly estimate the major and minor axis lengths failed. The LSEF algorithm, which looks at a spectrum not in terms of orientations and lengths, but in terms of mathematically minimizing a metric associated with its relative position to all points in the spectrum, was able to generate meaningful results.

## 5.6 Comparison of Results

Of the algorithms implemented and discussed in this paper, several show promise for estimating the bulk geometric statistics in specific types of images. The AM algorithm performed well on the simple sinusoidal image, as well as some of the less-dispersed Brodatz data. While not producing exact results, it was able to discern the major axis in one example and a multiple of the original major axis (i.e. an aliased version of the signal) in several others. However, its performance on the artificial and natural cellular imagery indicates that it can not produce meaningful results on this type of data.

The CEF algorithm has proven quite successful, most notably on images with moderate to high shape variability in the individual cellular element statistics. On the simplest example, the sinusoidal image, the CEF algorithm actually incorrectly identified the minor axis as that of the major, and produced a minor axis value that was converging on zero. This result was also observed in the unrotated Brodatz Brick image. This estimate is a result of the energy per pixel maximization of the algorithm. For example, in the 2d sinusoidal image, despite the existence of a strong signal at 32cpi, the energy per pixel was still lower when that axis of the ellipse was close to the origin. This convergence to the origin could be restricted by applying a minimum axis length constraint, but there is no guarantee that such a condition would restrict the ellipse from converging to the constraint. The ellipse will simply always move to where its energy per pixel is maximized. The CEF algorithm was also the only algorithm of the four capable of routinely producing accurate statistical results for the noisier artificial cellular data. However, the CEF was unable to produce a meaningful description of the natural cellular data.

The algorithm based on the Gabor filter algorithm also performed well on the sinusoidal image. On both the Brodatz images and the artificial cellular data, it was unable to identify the major axis orientation, but in every case, estimated the angle perpendicular to the desired orientation. The reason for this lies in the nature of the algorithm, and is best explained by contrasting with another method. The CEF algorithm operates by attempting to fit an ellipse to a set of data points. Inherent in this method is the notion of bulk statistics and the composite cell. That is, in fitting an ellipse, all statistics are used at once to get an overall picture of the power spectrum. The Gabor filter algorithm, as implemented in this work however, does not make use of all the bulk statistics at the same time. Instead, it seeks out the individual maximum response and assigns the centre coordinates corresponding to that filter to two of the bulk statistics. As a result, and in particular due to the large extent of the spatial-frequency Gabor filter at higher frequencies, pixel locations at relatively large distances from the image centre tended to produce stronger responses in power spectrums in which the energy was not concentrated at only a few points. The algorithm often detected one of the desired signals, but in searching along the perpendicular orientation, drifted out further to obtain a stronger response. This led to a rotation in the major axis orientation.

The Least Squares Ellipse Fitting algorithm had varying results. It was able to correctly identify the statistics in the simple sinusoidal image, made successful approximations of the orientation in the Brodatz images, was unable to estimate any worthwhile characterization of spatial statistics in the artificial cellular data and was the only algorithm capable of correctly estimating the bulk statistics in the natural cellular images. By attempting to minimize the sum of the squares of the distances between the ellipse and pixels in the power spectrum, this method is very sensitive to energy related to high-frequency noise or caused by natural variability in the image being tested (i.e. the high-frequency components in the artificial cellular data). However, the nature of the algorithm in characterizing overall spectrum shape, rather than measuring localized areas of the spectrum, proved advantageous in testing the natural imagery.

## 5.7 Computation Time

As with any automated algorithm, the length of time required to produce a result can be a major factor in determining usefulness. In this section, we present the length of time required for each algorithm on images of varying sizes. Square images

| Image Size<br>(pixels) | AM | CEF | GF | LSEF |
|---|---|---|---|---|
| 16 | 0.1 | 8.1 | 1.0 | 0.1 |
| 32 | 0.3 | 19.3 | 5.1 | 0.2 |
| 64 | 1.3 | 71.8 | 42.6 | 0.9 |
| 128 | 5.4 | 532 | 615 | 3.5 |
| 256 | 21.8 | 5648 | 5191 | 14.0 |

Table 5.9: Computational requirements, in seconds, for algorithms based on image size. Computation times are reported in seconds, as measured on a Pentium IV, 1.6GHz computer running Windows 2000 and Matlab 6.1.0.450 Release 12.1 with 512MB RAM.

with lengths of 32, 64, 128 and 256 pixels will be used. The results are shown in Table 5.7. Computation times are reported in seconds, as measured on a Pentium IV, 1.6GHz computer running Windows 2000 and Matlab 6.1.0.450 Release 12.1 with 512MB RAM, are shown in Table 5.9.

# Chapter 6

# Conclusion

A unique approach to estimating the bulk geometric statistics of a stationary image was presented in this thesis. Four algorithms were developed to demonstrate the utility in quantifying spatial statistics using spatial-frequency domain techniques. This chapter contains a brief summary of the findings, a discussion of the limitations facing such an approach and an elaboration on the future potential of the algorithms developed.

## 6.1 Summary

Several of the algorithms developed in this thesis show promise in quantifying spatial statistics in the spatial-frequency domain. Each technique will be discussed separately.

The Area Moments technique proved useful when the images being considered were well-structured and devoid of noise. In future work, they could very well be used on imagery matching this description.

The most promising method developed in this work, due primarily to its consistency across data sets, was the Correlation and Ellipse Fitting algorithm. This method was able to produce meaningful estimates for every image type except the natural imagery. Further work conducted on this research should focus on advancing this method.

Gabor filters have proven enormously effective in feature extraction and texture segmentation applications. However, unless constrained to determining the orientation, their usefulness appears to be limited in the context of quantifying spatial statistics of highly variable cellular textures.

Of the four algorithms, the Least Square Ellipse Fitting technique proved to be ineffective on the Brodatz and artificial cellular data, but was the only algorithm to correctly identify the bulk statistics in the natural cellular imagery. This technique is highly sensitive to noise that is located at relatively distant pixels from the best-fit ellipse. As a result, the ellipse that would best characterize the power spectrum tended to be “pulled” away from its ideal location, resulting in almost circular composite cells in images with high frequency noise or variability in the individual cellular elements. However, in the natural cellular data, in which most of energy was concentrated around the image centre, the best-fit ellipse seemed quite effective in capturing the bulk statistics.

## 6.2 Limitations

The results of applying the algorithms to a selection of images with increasing complexity indicates that there is merit in using spatial-frequency domain techniques to solve the problem outlined in this thesis. However, the varying results between image test sets for each algorithm imply that the suitability of each algorithm is limited to a certain type of data. As such, no algorithm can be used with a certainty of producing accurate statistical estimates without knowing more about the nature of the cellular texture being considered.

The Correlation and Ellipse Fitting algorithm consistently produced strong results on the artificial cellular images. This effectiveness is related to the strong elliptical shape of this test set's spectra. It was believed that an algorithm that was effective in characterizing the artificial cellular images would also be effective in producing meaningful results when applied to the natural cellular imagery. However, the natural data set did not have the elliptically shaped power spectra that the CEF algorithm was designed for. If the CEF algorithm is going to produce meaningful results, the spatial image under consideration will need to undergo significant pre-processing. Despite well-defined boundaries and fairly consistent cell shape and size, the low frequency noise that is predominant in the power spectrum will need to be reduced.

In addition to the natural noise processes present in the images, other factors likely affected the performance of the algorithms. During the image capture process for the natural cell images, a non-linear noise component may have been introduced (i.e. reflectance and illumination). Improvements in the image capture process may help to minimize these imperfections.

In each algorithm, errors would have been introduced as a result of operating on a digital image. This is a necessary trade-off between computation time, and resolution and image size. In each computation that required decision-making based on a neighbourhood of pixels, bicubic interpolation was performed. However, any form of interpolation has associated imperfections.

## 6.3 Future Work

This thesis has demonstrated the utility in using the spatial-frequency domain to quantify spatial statistics. However, there are opportunities to expand upon the work in this thesis. General areas of improvement and technique-specific suggestions are outlined in this section. Of those algorithms presented here, two have areas where changes could significantly impact performance – LSEF and CEF.

While the LSEF algorithm was able to estimate the bulk statistics in the natural imagery, its failure to perform well on the artificial cellular data, especially given the strong elliptical shape of the spectra needs to be addressed. The fault in this algorithm lies in the technique central to the naming – least squares. If instead of summing the squares of the distances between pixels and the best ellipse, only the absolute differences in distance were summed, pixels farther from the ideal ellipse would have much less of an impact on determining the ellipse location. The current methods, which make use of efficient matrix properties associated with squares, would have to be abandoned and a more computationally expensive algorithm would likely result. Such a change may lead to an improvement in the performance when testing the artificial cellular data while preserving the performance when testing the natural data.

Another improvement that could be made to allow for a correct estimation of the bulk statistics in the artificial cellular data while still employing the computation efficient least square method would be to apply thresholding to the power spectra. Even tiny amounts of noise or variations, if a sizable distance from optimal ellipse, can significantly impact the estimate accuracy. By disregarding all pixel contributions below a certain intensity, which could be determined based on the variation in the image, the LSEF could be made considerably more effective. Also, if the algorithm were to be operating on a consistent type of calibrated image, it is not unreasonable to assume that a threshold could remain constant for the duration of testing, as similar images tend to have similar power spectrums.

The CEF and LSEF methods, as previously stated, varied as to which data types they were effective with, but show the most promise for future work.

Image properties can significantly affect the performance of any algorithm. Since we are looking exclusively at spatial-frequency domain techniques, it is important that information related to spatial statistics are optimally captured. In order to best characterize the bulk statistics, there must be sufficient periodicity in the cellular elements to create strong signals. In the case of the natural imagery, this means that the more cells exhibiting similar shape characteristics that can be captured in one image, the stronger the spatial-frequency domain search space will be. For future testing on natural cellular data, images should be captured with as great a spatial extent with as many periods as possible.

It is also worth noting that all of the algorithms, with the exception of the CEF method, were able to estimate the spatial statistics of the simple 2d sinusoidal image. Future direction for this work could lead to using different algorithms for different data types, but via an automated estimation program that would select the method most suitable for a certain type of image. If certain images required more complex and computationally expensive algorithms, those would be available, but if the spatial statistics could be estimated with a simple and quick method like the AM or LSEF algorithms, such a decision could be made automatically.

It was originally stated that the images being considered in this thesis are ill-suited to spatial domain techniques. This may be true, but to most effectively describe the utility in employing spatial-frequency domain techniques, a comparison should be drawn between the performance of spatial and spatial-frequency domain techniques. A similar comparison made, but with varying pre-processing techniques (i.e. smoothing, edge detection, homomorphic filtering, etc.) could provide insight into the effect of such pre-processing on each algorithm.

This study presented a novel approach to solving a complicated computer problem. While there is room for improvement, results were produced that indicate there is merit in employing spatial-frequency domain techniques to estimate spatial statistics.

# Appendix A

# Additional Results

This appendix contains the remainder of the images showing the graphical results from testing on the artificial cellular data.

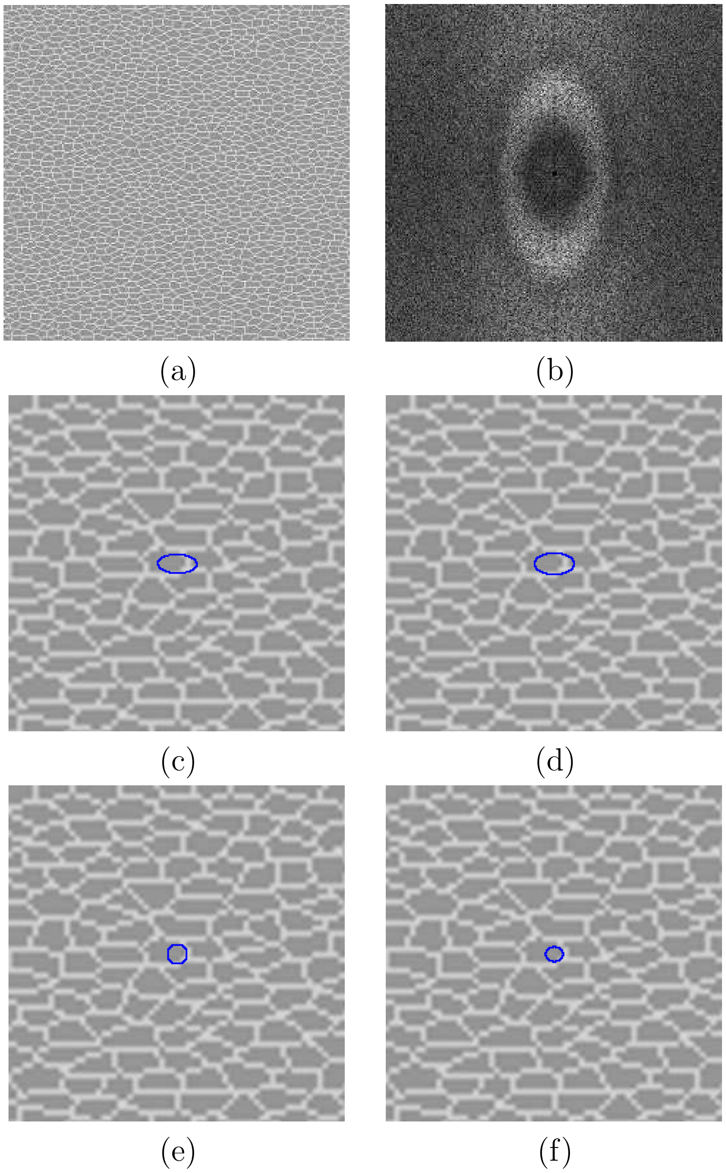

(a)
(b)
(c)
(d)
(e)
(f)

Figure A.1: Graphical results for artificial cellular data with $\theta$=0° and $\kappa$=1.8. The original image is shown in (a), along with the power spectrum in (b). The composite cell is shown superimposed on the image (magnified by a factor of 4) for each of the algorithms: AM (c), CEF (d), GF (e) and LSEF (f).

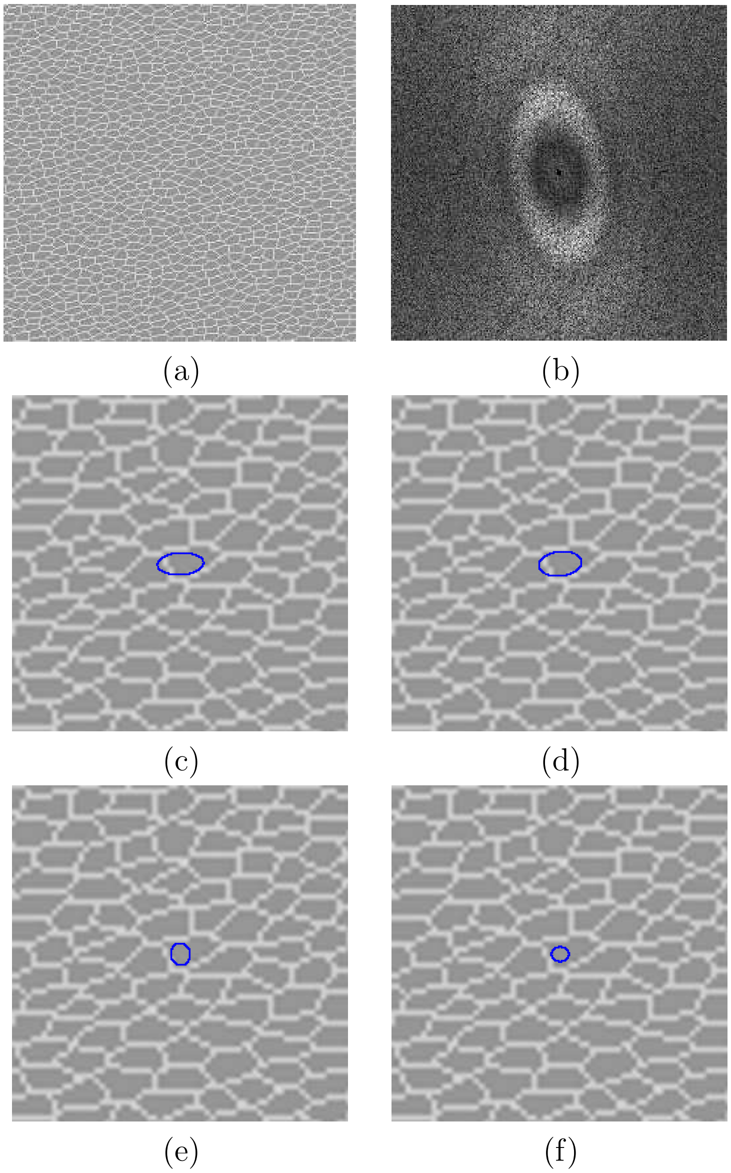


Figure A.2: Graphical results for artificial cellular data with $\theta$=10° and $\kappa$=1.8. The original image is shown in (a), along with the power spectrum in (b). The composite cell is shown superimposed on the image (magnified by a factor of 4) for each of the algorithms: AM (c), CEF (d), GF (e) and LSEF (f).

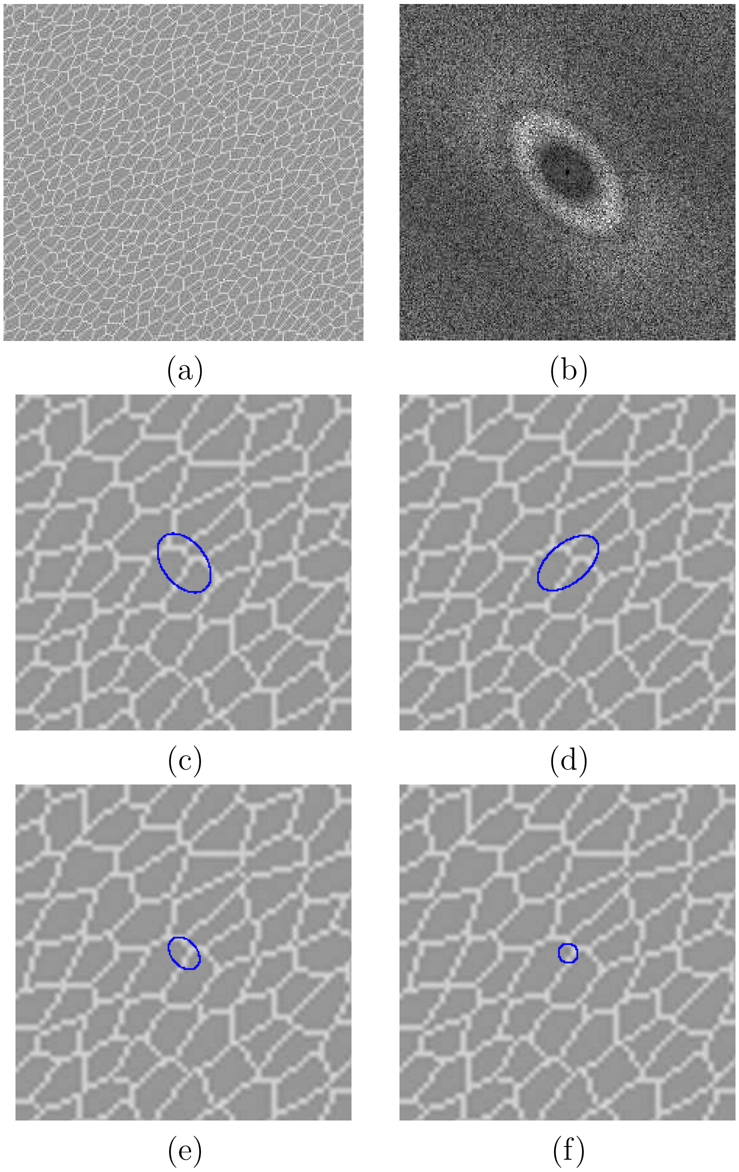
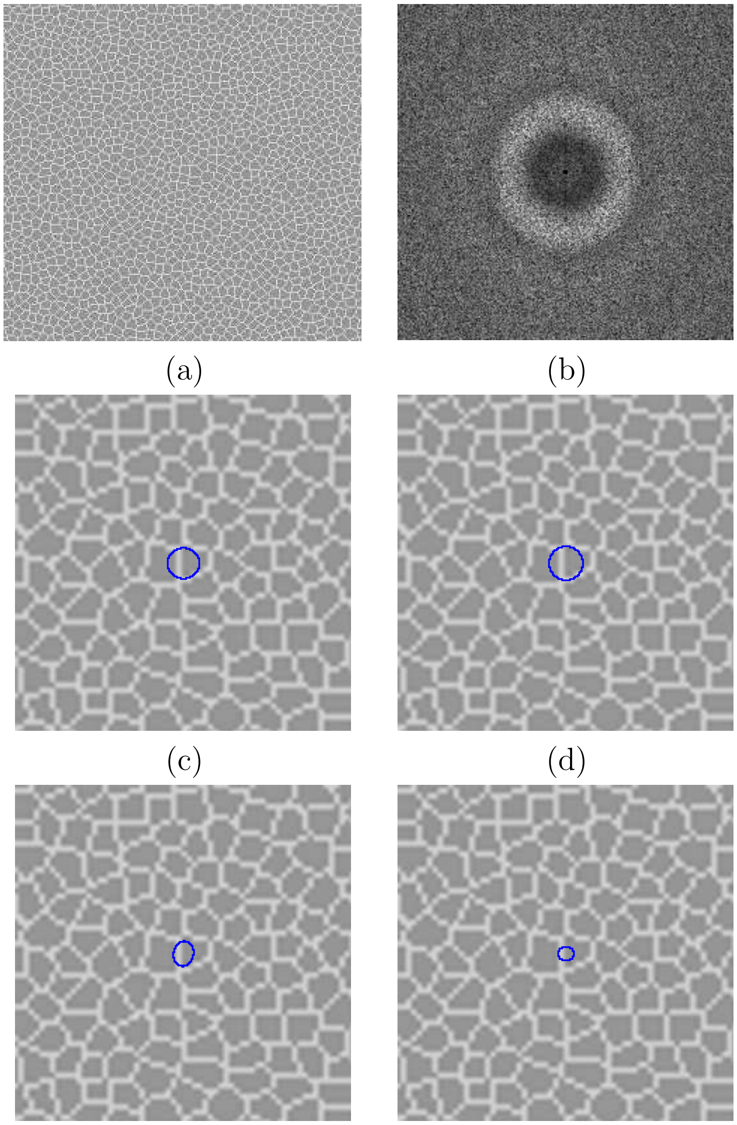


Figure A.3: Graphical results for artificial cellular data with $\theta$=45° and $\kappa$=1.8. The original image is shown in (a), along with the power spectrum in (b). The composite cell is shown superimposed on the image (magnified by a factor of 4) for each of the algorithms: AM (c), CEF (d), GF (e) and LSEF (f).



(a)

(b)

(c)

(d)

(e)

(f)

Figure A.4: Graphical results for artificial cellular data with $\theta$=30° and $\kappa$=1.0. The original image is shown in (a), along with the power spectrum in (b). The composite cell is shown superimposed on the image (magnified by a factor of 4) for each of the algorithms: AM (c), CEF (d), GF (e) and LSEF (f).

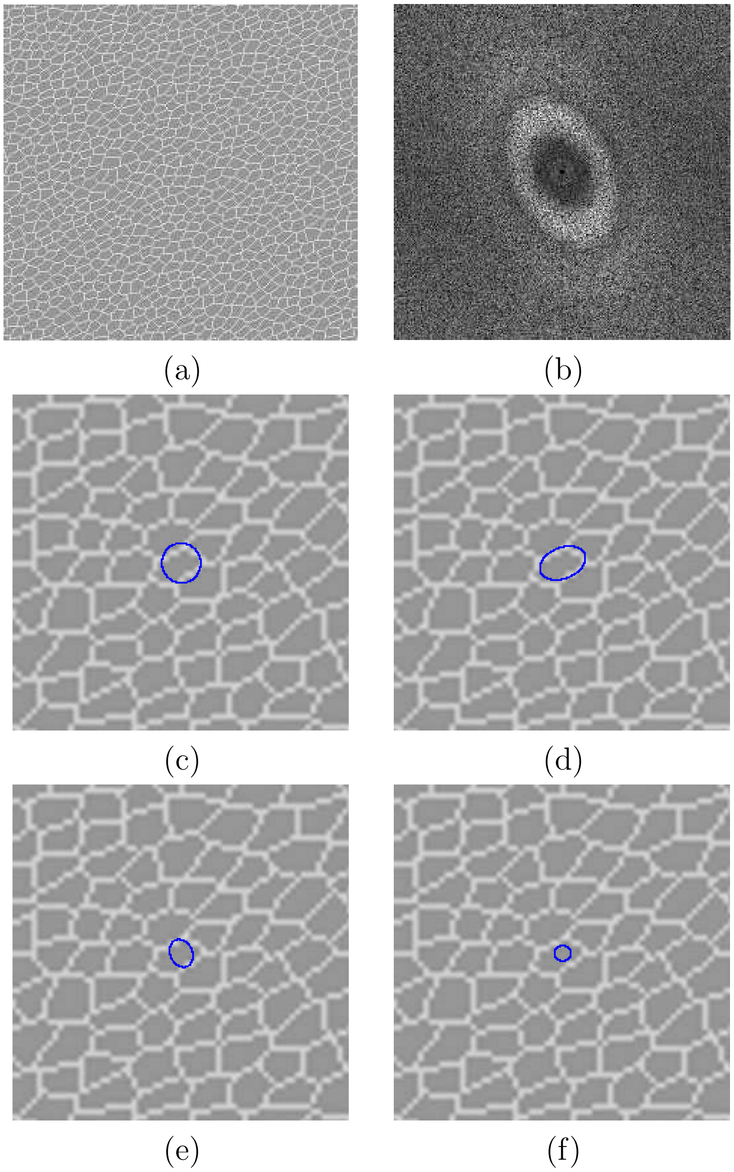
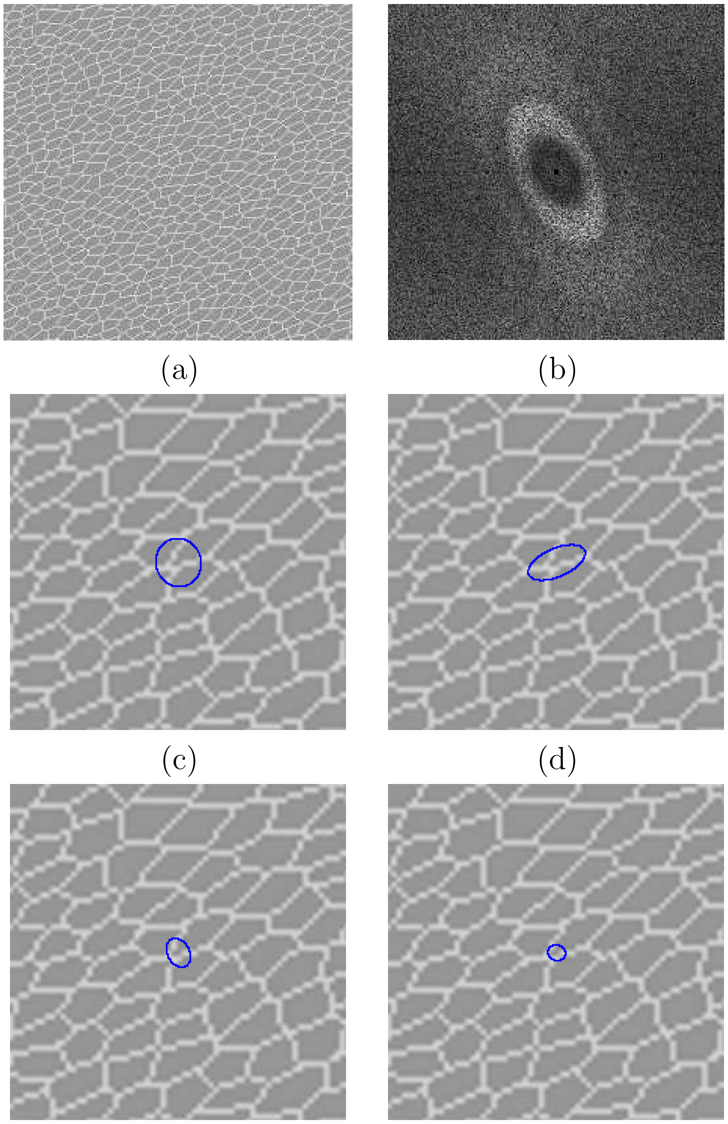


Figure A.5: Graphical results for artificial cellular data with $\theta$=30° and $\kappa$=1.5. The original image is shown in (a), along with the power spectrum in (b). The composite cell is shown superimposed on the image (magnified by a factor of 4) for each of the algorithms: AM (c), CEF (d), GF (e) and LSEF (f).


(a) (b)

(c) (d)

(e) (f)

Figure A.6: Graphical results for artificial cellular data with $\theta$=30° and $\kappa$=1.9. The original image is shown in (a), along with the power spectrum in (b). The composite cell is shown superimposed on the image (magnified by a factor of 4) for each of the algorithms: AM (c), CEF (d), GF (e) and LSEF (f).

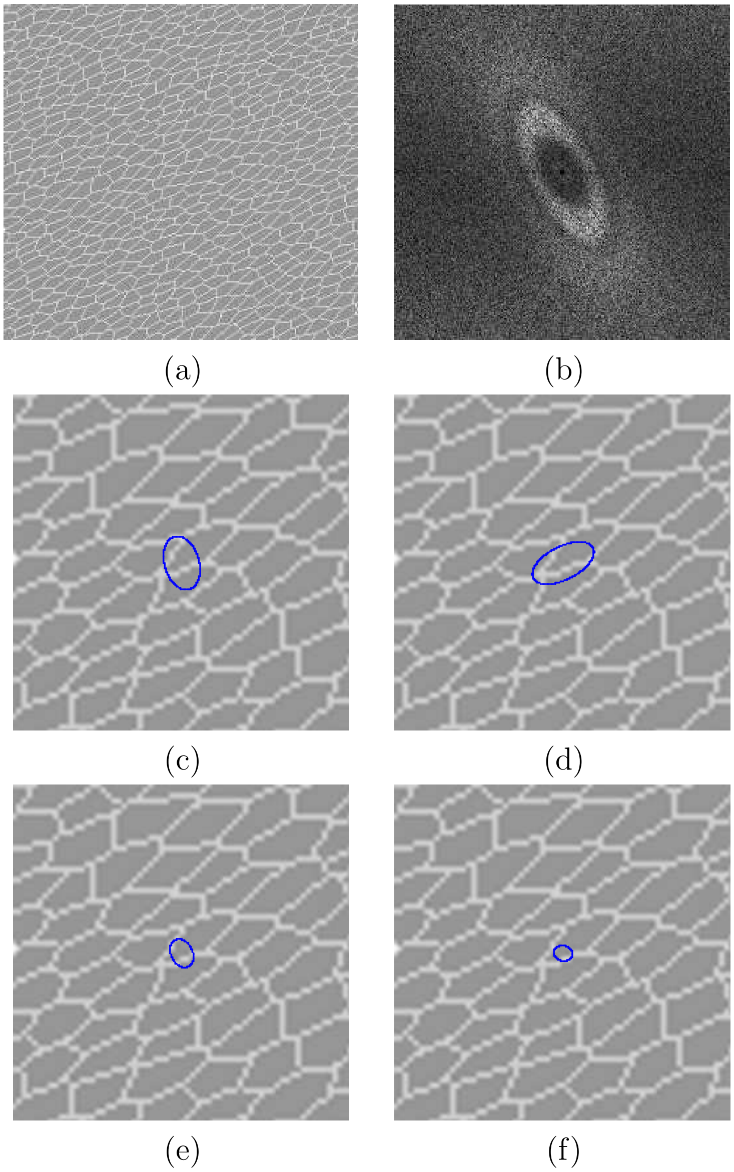


Figure A.7: Graphical results for artificial cellular data with $\theta$=30° and $\kappa$=2.3. The original image is shown in (a), along with the power spectrum in (b). The composite cell is shown superimposed on the image (magnified by a factor of 4) for each of the algorithms: AM (c), CEF (d), GF (e) and LSEF (f).

# Appendix B

# Matlab

This appendix contains the Matlab functions written to implement each of the algorithms.

## B.1 Area Moments

```
function results = am(im)
%
% am(im) takes a power spectrum ’im’ and returns the statistical
%   information about the bulk statistics of the shape of the
%   power spectrum it is analyzing.  This algorithm functions
%   by first using area moments to determine the orientation
%   that the power spectrum is rotated by, and then uses
%   projections and a smoothing operator to estimate the
%   distance from the image centre at which ‘most’ of the power
%   spectrum information is contained at.
%   results will be reported as follows:
%
%   results(1): orientation of the major axis in degrees
%       measured from the rightmost horizontal
%   results(2): length of the major axis in cpi
%   results(3): length of the minor axis in cpi
%
%   Assumption: Power spectrum ‘im’ must be square.
%

imLength = length(im);
centre = ceil((imLength+1)/2-1) + 1;

%%%%%%%%%%%%%%%%%%%%%%%%%%
% orientation estimation %
%%%%%%%%%%%%%%%%%%%%%%%%%%

% 2D matrix containing column and row pixel values
x=meshgrid(1:imLength,1:imLength);
y=meshgrid(1:imLength,1:imLength)’;

% Compute moment-based summations
x = x - centre;
y = y - centre;
a = sum(sum(double(im).*(x.*x)));
b = sum(sum(double(im).*(x.*y)))*2;
c = sum(sum(double(im).*(y.*y)));
determ = sqrt(b^2 +(a-c)^2);

if determ ~= 0
  sin2theta   = b/determ;
  cos2theta   = (a-c)/determ;
  theta       = atan2(sin2theta,cos2theta)/2;
else
  theta       = 0;
  sin2theta   = 0; %sin(0)
  cos2theta   = 1; %cos(0)
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% major and minor axis length estimation %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Requires orientation from above in radians

% Create coordinate matrix
x = zeros(imLength);
y = zeros(imLength);
for i=1:imLength,
  for j=1:imLength,
    x(i,j) = j-centre;
    y(i,j) = i-centre;
  end;
end;

% Create 1X3 matrix of all (x-coord, y-coord, pixel intensity)
vec = zeros(imLength^2,3);
index = 0;
for i=1:imLength,
  for j=1:imLength,
    index = index + 1;
    vec( index, 1 ) = x(i,j);
    vec( index, 2 ) = y(i,j);
    vec( index, 3 ) = im(i,j);
  end
end

% Rotate coordinates by angle determined above
```

```
newVec = zeros(length(vec),3);
for i=1:length(vec),
  newVec(i,1) = vec(i,1)*cos(theta)-vec(i,2)*sin(theta);
  newVec(i,2) = vec(i,1)*sin(theta)+vec(i,2)*cos(theta);
  newVec(i,3) = vec(i,3);
end

% Ensure all coordinate values are integer valued.  This is
%   required since these will be matrix indices.
for i=1:length(vec),
  newVec(i,1) = round(newVec(i,1));
  newVec(i,2) = round(newVec(i,2));
end

% Create projection for major and minor axes
majorVec = zeros(length(vec),2);
minorVec = zeros(length(vec),2);
for i=1:length(vec),
  majorVec(i,1) = newVec(i,1);
  majorVec(i,2) = newVec(i,3);
  minorVec(i,1) = newVec(i,2);
  minorVec(i,2) = newVec(i,3);
end

% Compute minimum and maximum values of x and y coordinates
temp = max(newVec);
maxMajor = temp(1);
maxMinor = temp(2);
temp = min(newVec);
minMajor = temp(1);
minMinor = temp(2);
```

```
% Determine length necessary for the projection
majorLen = maxMajor - minMajor + 1;
minorLen = maxMinor - minMinor + 1;

% Create axis for projection
majorProj = zeros(majorLen,1);
minorProj = zeros(minorLen,1);

% Project points onto the line
for i=1:length(majorVec),
  if ( abs(newVec(i,2)) < 2 )
    majorProj(newVec(i,1)-minMajor+1) = \\
        majorProj(newVec(i,1)-minMajor+1) + newVec(i,3);
  end
  if ( abs(newVec(i,1)) < 2 )
    minorProj(newVec(i,2)-minMinor+1) = \\
        minorProj(newVec(i,2)-minMinor+1) + newVec(i,3);
  end
end

% Zero pixel associated with image centre
majorProj( ceil( (length(majorProj)+1)/2) ) = 0;
minorProj( ceil( (length(minorProj)+1)/2) ) = 0;

% Rescale projection so max value = 1
SmajorProj = majorProj / max(majorProj);
SminorProj = minorProj / max(minorProj);

% Create 1D Gaussian for smoothing
xg = -20:0.5:20;
sig = 1;
g = exp( -xg.^2 ./ (2*sig^2) );
```

```
% Convolve projections with Gaussian to smooth projection
yMajor = conv( SmajorProj, g);
yMinor = conv( SminorProj, g);

% Determine centre point of projection
yMajorStartPoint = ceil(length(yMajor)/2);
yMinorStartPoint = ceil(length(yMinor)/2);

% Create a new projection including only half the points
%   along the projection.  Points should correspond
%   with being one pixel out from the image centre
newMajor = zeros(length(yMajor)-yMajorStartPoint,1);
newMinor =zeros(length(yMinor)-yMinorStartPoint,1);
for i=1:length(newMajor),
  newMajor(i) = yMajor(yMajorStartPoint+i);
end
for i=1:length(newMinor),
  newMinor(i) = yMinor(yMinorStartPoint+i);
end

% Determine max values and associated indices along
%    projections
[val,majorVal]=max(newMajor);
[val,minorVal]=max(newMinor);

angle = theta*180/pi;
results = [-angle, majorVal, minorVal];
```

## B.2 Correlation and Ellipse Fitting

### B.2.1 CEF Code

```
function results = cef(im)
% cef(im) takes a power spectrum ‘im’ and returns the
%   statistical information about the bulk statistics of the shape
%   of the power spectrum it is analyzing. This algorithm
%   functions by first using a correlation technique to determine
%   the orientation that the power spectrum is rotated by, and
%   then a brute force ellipse fitting algorithm to determine the
%   ellipse parameters resulting in the strongest energy per pixel.
%
%   results will be reported as follows based on spatial-frequency:
%
%   results(1): orientation of the major axis in degrees measured
%               from the rightmost horizontal
%   results(2): length of the major axis in cycles per image
%   results(3): length of the minor axis in cycles per image
%
%   Assumption: Power spectrum ‘im’ must be square.

% This section adds an row and column of zero padding if the power
%   spectrum dimensions are not odd.  This is done to compensate for
%   the manner in which the built-in Matlab functions ‘flipud’ and
%   ‘fliplr’ operate and ensures the image centre’s line up.
imLength = length(im);
if mod(imLength,2)==0
  temp_im = zeros(imLength+1);
  for i=1:imLength,
    for j=1:imLength,
      temp_im(i,j)=im(i,j);
    end
  end
  imLength=imLength+1;
  im=temp_im;
end

% This section estimates the power spectrum orientation
max_corr = 0;
best_angle = 0;
for angle = -45:1:45,
  % Create new image of image rotated by ‘angle’
  rot_img = double(imrotate(im,angle,’nearest’,’crop’));
  % Create new images by flipping image about the image
  %   centre (Matlab functions)
  flip_hor_img = fliplr(rot_img);
  flip_ver_img = flipud(rot_img);
  % Compute a correlation measure
  corr = sum(sum(rot_img.*flip_hor_img)) + \\
            sum(sum(rot_img.*flip_ver_img));
  % Compare correlation to current highest & replace if higher.
  if corr > max_corr,
    max_corr = corr;
    best_angle = angle;
  end
end

% Refine orientation search down to the nearest 0.1
%  Same code as previous section, but to a more precise angle
for factor = -0.5:0.1:0.5,
  angle = best_angle+factor;
  rot_img = double(imrotate(im,angle,’nearest’,’crop’));
  flip_hor_img = fliplr(rot_img);
  flip_ver_img = flipud(rot_img);
  corr = sum(sum(rot_img.*flip_hor_img)) + \\
            sum(sum(rot_img.*flip_ver_img));
  if corr > max_corr,
    max_corr = corr;
    best_angle = angle;
  end
end

% Need to multiply by -1 because orientation algorithm above
%   actually rotates power spectrum opposite to the orientation
%   when computing the correlation
best_angle = -best_angle;

% This section estimates the major and minor axis lengths by
%   employing an ellipse fitting brute force search
max_resp = 0; best_major = 0;
centre = ceil((imLength+1)/2-1)+1; % ellipse centre
thickness=1;  % thickness of ellipse to draw
theta = best_angle*pi/180; % convert best_angle to radians
% Create ellipses for varying major and minor axis lengths.  That
%   generating the highest energy per pixel corresponds to the
%   best ellipse.  a=major axis, b=minor axis.
for a=1:floor(0.65*imLength/2),
  for b=1:floor(0.65*imLength/2),
    % Create ellipse
    testImage = rescale( \\
       make_ellipse(centre,centre,theta,a,b,thickness,imLength) );
    % Compute energy per pixel
    resp = sum(sum( im .* testImage ) ) / sum(sum(testImage));
    % If higher energy per pixel, change stored values
    if resp>max_resp
      max_resp = resp;
      len_one = a;
      len_two = b;
    end
  end
end

% Refine Axis Lengths Measurement
%  Same code as previous section, but to a more precise value
len_one_temp = len_one;
len_two_temp = len_two;
for a_factor = -0.5:0.1:0.5,
  for b_factor = -0.5:0.1:0.5,
    a = len_two + a_factor;
    b = len_one + b_factor;
    testImage = rescale( \\
        make_ellipse(centre,centre,theta,a,b,thickness,imLength) );
    resp = sum(sum( im .* testImage ) ) / sum(sum(testImage));
    if resp>max_resp
      max_resp = resp;
      len_one_temp = a;
      len_two_temp = b;
    end
  end
end len_one = len_one_temp;
len_two = len_two_temp;

% The bulk statistics are analyzed.  The longest axis is chosen
%   as the ellipse major axis, and the orientation is assigned
%   the value associated with this axis.  The minor axis length
%   is the smaller of the two lengths.
if len_one > len_two
  theta = best_angle;
  major_len = len_one;
  minor_len = len_two;
else
  if (best_angle>=-90) & (best_angle<=0)
    theta = best_angle + 90;
  else
    theta = best_angle - 90;
  end
  major_len = len_two;
  minor_len = len_one;
end if theta>90
  theta = theta - 180;
elseif theta<-90
  theta = theta + 180;
end

% results are returned as the ellipse parameters in the SF domain.
results = [theta, major_len, minor_len];
```

### B.2.2 CEF Function Calls

```
function outimg=make_ellipse(x,y,theta,maj,min,thickness,N)
% make_ellipse(x,y,theta,a,b,thickness,N)
%
% This routine plots an ellipse with centre (x,y), axis lengths a,b
%   with major axis at an angle of theta (radians) from the
%   horizontal, in an NxN image.
%
%   Assumption: Power spectrum ‘im’ must be square.

if nargin<5,
  error(’Too few arguments to Plot_Ellipse.’);
end;
```

```
np = 1000;
ang = [0:np]*2*pi/np;
pts = [x;y]*ones(size(ang)) + \\
    [cos(theta) -sin(theta); sin(theta) cos(theta)] * \\
    [cos(ang)*min; sin(ang)*maj];
outimg = zeros(N);
for i=1:size(pts,2),
  if (pts(1,i) >= 1)&(pts(2,i) >= 1)&(pts(1,i) <= N)&(pts(2,i) <= N)
    outimg(round(pts(1,i)),round(pts(2,i))) = 1;
  end
end outimg = double(imdilate(outimg,ones(thickness)));
```

## B.3 Gabor Filters

### B.3.1 GF Code

```
function results = gf(im)
%
% gf(im) takes a power spectrum ‘im’ and returns the statistical
%   information about the bulk statistics of the shape
%   of the power spectrum it is analyzing.  This algorithm
%   functions by using gabor filters to determine the orientation
%   that the power spectrum is rotated by and the major axis
%   lengths that best describes the width of elements in a
%   cellular texture.  It them identifies the minor axis length
%   using a similar search.
%
%   results will be reported as follows, all as spatial frequency
%   statistics:
%
%   results(1): orientation of the major axis in degrees measured
%               from the rightmost horizontal
%   results(2): length of the major axis in cycles per image
%   results(3): length of the minor axis in cycles per image
%
%   Assumption: Power spectrum ‘im’ must be square.

imLength = length(im);
initialSearchSpacing = 3;

% Determine range of search space. min/max frequency are determined
%       based to ensure Gabor filter overlap with edges beyond two
%       standard deviations does not occur
minFrequency=ceil( sqrt(log(4))/(sqrt(log(4))-1));
maxFrequency=floor((imLength-2)/2*sqrt(log(4))/(sqrt(log(4))+1));

% This section performs a brute force search for strongest the
%       response.  This value will correspond with the
%       orientation of one of the axis lengths
max_resp = 0;
best_ang = 0;
len_one = 0;
for freq=minFrequency:2:maxFrequency,
  for ang=-90:2:90,
    % Determine response
    resp = mag_response( im, ang, freq );
    if resp > max_resp
      max_resp = resp;
      len_one = freq;
      best_ang = ang;
   end
  end
end
```

```
% Search refinement
for freq_factor=-1:0.1:1,
  for ang_factor=-1:0.1:1,
    freq = len_one + freq_factor;
    ang = best_ang + ang_factor;
    resp = mag_response( im, ang, freq );
    if resp > max_resp
      max_resp = resp;
      len_one_temp = freq;
      best_ang_temp = ang;
    end
  end
end len_one=len_one_temp best_ang=best_ang_temp

% This section searches along the ray perpendicular to the
%   orientation measured in the previous section.  The
%   strongest response is recorded as the second axis length.
max_resp = 0;
len_two = 0;
for freq=minFrequency:2:maxFrequency,
  resp = mag_response( im, best_ang+90, freq );
  if resp > max_resp
    max_resp = resp;
    len_two = freq;
  end
end
% Search refinement
for freq_factor=-1:0.1:1,
  freq = len_two + freq_factor;
  resp = mag_response( im, best_ang+90, freq );
  if resp > max_resp
    max_resp = resp;
    len_two_temp = freq;
  end
end len_two = len_two_temp;

% The bulk statistics are analyzed.  The longest axis is chosen
%   as the ellipse major axis, and the orientation is assigned
%   the value associated with this axis.  The minor axis length
%   is the smaller of the two lengths.
if len_one > len_two
  theta = best_ang;
  major_len = len_one;
  minor_len = len_two;
else
  if (best_ang>=-90) & (best_ang<=0)
    theta = best_ang + 90;
  else
    theta = best_ang - 90;
  end
  major_len = len_two;
  minor_len = len_one;
end if theta>90
  theta = theta - 180;
elseif theta<-90
  theta = theta + 180;
end

% results are returned as the ellipse parameters in the SF domain.
results = [theta, major_len, minor_len];
```

### B.3.2 GF Function Calls

```
function resp = mag_response( im, ang, freq )
%
```

```
% mag_response(im, ang, freq) takes a power spectrum ‘im’, an angle
%   ‘ang’ (in degrees) and a frequency ‘freq’ (in cycles per image)
%   and returns the resulting magnitude response of applying
%   a Gabor filter with parameters ‘ang’ and ‘freq’ to ‘im’.
%
%   Assumption: Power spectrum ‘im’ must be square.

imLength = length(im);
% Create Gabor Filter based on freq and ang
gS = complex_sf_gabor(freq,ang,imLength);
% Filter image in spat-freq domain
gimS = (gS.*im);
% Bring filtered image back to spat domain
gimS = ifft2(ifftshift(gimS));
% Determine magnitude response via real & imaginary components
m_gimS = sqrt( real(gimS).^2 + imag(gimS).^2 );
% Calculate average response based on image size
resp = sum(sum(m_gimS));

function [result]=complex_sf_Gabor(freq, ang, imLength)
% complex_sf_Gabor(freq, ang, imLength) generates a complex
%   spatial-frequency domain Gabor filter.  Frequency ‘freq’
%   must be in cpi and angle ‘ang’ in degrees.  Angular and
%   frequency bandwidths of the Gabor filter are chosen to be
%   similar to the human visual system.
%
%  For details in filter construction, see:
%  D.A. Clausi and M.E. Jernigan, "Designing Gabor filters for
%     optimal texture separability". Pattern Recognition,
%     33, 1835-1849, 2000.
%

% Two imLength-by-imLength basis matrices are created
```

```
[u,v]=meshgrid(-0.5:1/imLength:0.5-1/imLength, \\
               -0.5:1/imLength:0.5-1/imLength);

% Filter bandwidth chosen to match human visual
%   system sensitivity
Bf = 1;
Btheta = pi/6;
% angle ‘ang’ is converted to radians (from degrees)
ang = ang * pi / 180;

% Basis matrices are rotated by ‘ang’ radians
if (ang == 0)
  up = u;
  vp = v;
else
  up = xrotate(u,v,ang);
  vp = yrotate(u,v,ang);
end

% Frequency tuning is based on the length of the image.  As
%   such, must convert from cpi to ratio with image length.
if freq˜=0
  freq = freq/imLength;
else
  freq = 1;
end

% Gaussian standard deviation parameters are determined
sigmax = ( sqrt(log(2))*(2^Bf+1) )/( sqrt(2)*pi*freq*(2^Bf-1) );
sigmay = sqrt(log(2))/(freq*pi*sqrt(2)*tan(Btheta/2));

% Create modulated Gaussian
```

```
result = exp(-2*pi^2*(((up-freq).^2)*sigmax^2+(vp.^2)*sigmay^2));
```

## B.4 Least Square Ellipse Fitting

```
function results = LSEF(im)
%
% lsef(im) takes a power spectrum ‘im’ and returns
%   the statistical information about the bulk statistics of the
%   shape of the power spectrum it is analyzing.  This algorithm
%   functions by fitting an ellipse to the data using least
%   squares.
%
%   This is information is of the form:
%
%   results(1): orientation of the major axis in degrees
%       measured from the rightmost horizontal
%   results(2): length of the major axis in cpi
%   results(3): length of the minor axis in cpi
%
%   Assumption: Power spectrum ‘im’ must be square.

% Measure image dimensions
imLength = length(im);

% Pixel intensities are broken into individual points based on
%   ‘stepsize’.  In this case, it is set to 0.1.  Therefore, a
%   pixel location with an intensity of 0.72 would have seven
%   points associated with its location.
points = zeros(imLength);
stepSize = 0.10;
count = 0;
for i=1:imLength,
  for j=1:imLength,
    if im(i,j)>stepSize
      num = floor(im(i,j)/stepSize);
      points(i,j) = points(i,j) + num;
      count = count + num;
    end
  end
end

% Create and fill x and y vectors, and fill with binned points
x = zeros(count,1);
y = zeros(count,1);
count = 0;
for i=1:imLength,
  for j=1:imLength,
    for k=1:points(i,j)
      count = count + 1;
      x(count) = j;
      y(count) = i;
    end
  end
end

% Determine bulk statistics using ‘fitellipse’ least squares
%   algorithm of Fitzgibbon et al.,  The ‘fitellipse’ function
%   call is described and available as Matlab code in the
%   following paper:
%
% Fitzgibbon, M. Pilu , R.Fisher, "Direct least-square
%    fitting of Ellipses", IEEE PAMI, June 1999.
%
a = fitellipse(x,y);
```

```
ang = a(5)*180/pi;
maj = a(3);
min = a(4);
results = [ang, maj, min];
```

# Appendix C

# Optimization Matlab

This appendix contains the Matlab code associated with the optimization algorithms. This algorithm was used with the Gabor filter algorithm, but could be expanded to include the ellipse fitting segment of the Correlation and Ellipse Fitting algorithm. While the Gabor filter has three parameters that need to be computed, for simplicity, this algorithm is presented using only a two parameter optimization.

## C.1 Pattern Search

```
function [values]=pattern_search_optimization( im );
%
% pattern_search_optimization(im) takes a power spectrum ‘im’
%   with DC component set to zero and returns the orientation
%   and frequency pair that generate the largest response
%   from a Gabor filter.
%
%   [values] will be reported as follows:
%   values(1): orientation in degrees, measured counter-
%       clockwise from the u-axis.
%   values(2): frequency in cycles per image.
```

```
%
%   Assumption: Power spectrum ‘im’ must be square.
%

% Dimensions of image.  Used to determine upper and
%   lower bounds.  minFreq and maxFreq represent constraints
%   on the permissible search space.  Orientation is
%   automatically set to look between -90 and 90 degrees.
imLength=length(im); minFreq =
ceil(sqrt(log(4))/(sqrt(log(4))-1));
maxFreq=floor((imLength-2)/2*sqrt(log(4))/( sqrt(log(4))+1 ) );

% The algorithm searches along a rational lattice.  With this
%   in mind, a search basis much be created.  Unit vectors, one
%   associated with orientation and the other with frequency
%   are used.
basis = [0,1;0,-1;1,0;-1,0];

% stepData holds information relevant to the optimization
%   algorithm.  Since the optimization functions by moving
%   around in the search space, stepData(1) defines the
%   current movement length, and stepData(2) defines the
%   minimum allowable step length (i.e. the accuracy of
%   the final answer).
stepData = zeros(1,2);
stepData(1) = min([(imData(5)-imData(4))/4,32 ] );
stepData(2) = 0.5;

% pointData is the vector that holds information about the
% current best solution
% pointData = [best_response_value, best_angle, best_frequency]
pointData = [ 0, 0, (maxFreq-minFreq)/2];
```

```
% Initialize optimization state machine.  The optimization
%   algorithm has two states: exploring and reducing step
%   length. ‘stop’ is used as a flag to denote the algorithm
%   has achieved the desired accuracy in measurement.
temp_values=zeros(1,5);
state = 1;
stop = 0;

while (stop == 0)
  % take an exploratory step
  if state==1
    temp_values = exploratory_moves( \\
              IM, pointData, basis, stepData(1), minFreq, maxFreq);
    if temp_values(5)==1
      % values(5) is a flag to denote the new ’best-values’ have
      %   been obtained.  This conditional updates the pointData
      %   information.  If not improve was found in the exploratory
      %   steps, state 2 is entered.
      pointData(1) = temp_values(1);
      pointData(2) = temp_values(2);
      pointData(3) = temp_values(3);
      stepData(1) = temp_values(4); % current step length
    else
      state = 2;
    end
  else % check for sufficient decrease
    if stepData(1)>=stepData(2)
      stepData(1) = stepData(1)/2;
      state = 1;
    else
      stop=1;
    end
  end
end

values = [pointData(1), pointData(2)]
```

## C.2 Pattern Search Function Calls

```
function [value] = exploratory_moves(im, pointData, basis,
step_length, minFreq, maxFreq);

bestx = zeros(1,2); % current best orientation and frequency
bestf = pointData(1); % current highest response
% param contains values important to keeping track of the optimization
%           and passing information back and forth between this function
%           and the pattern search itself
%               param(1) = current highest response
%               param(2) = current best orientation
%               param(3) = current best frequency
%               param(4) = current step_length
%               param(5) = improvement flag (1=>higher response found)
param = zeros(1,5);
x = zeros(1,2); % temporary orientation & frequency register

% Start exploratory search by doubling the step length
%       i.e. taking bigger steps
step_length = 2*step_length;

% Search along each of the basis direction at a distance
%       step_length and check each response to see if it is
%       higher than that already computed
for i = 1:length(basis),
    search_dir = [basis(i,1),basis(i,2)];
    % x(1)=orientation, x(2)=frequency
    x(1) = pointData(2) + step_length*search_dir(1);
    x(2) = pointData(3) + step_length*search_dir(2);
    % Determine response.  See code in Appendix B.3.1
    f = mag_response( im, x(1), x(2) );
    % Check if there was an improvement in response.  If so
    %       update temporary 'best' registers
    if f>bestf
        bestx = x;
        bestf = f;
    end
end

% Check to see is there was an improvement between in
%    response.  If so, update param register and set
%    improvement falg (param(5)) to 1.
if bestf < pointData(1)
    param(1) = bestf;
    param(2) = bestx(1);
    param(3) = bestx(2);
    param(4) = step_length;
    param(5) = 1;
end

if value(5)==0
  step_length = step_length/2;
  for i = 1:length(basis),
    search_dir = [basis(i,1),basis(i,2)];
    % x(1) = Orientation, x(2) = Frequency
    x(1) = pointData(2) + step_length*search_dir(1);
    x(2) = pointData(3) + step_length*search_dir(2);
    % Determine response.  See code in Appendix B.3.1
    f = mag_response( IM, x(1), x(2) );
    % Check if there was an improvement in response.  If so
    %       update temporary 'best' registers
    if f>bestf
      bestx = x;
      bestf = f;
    end
  end
  % Check to see is there was an improvement between in
  %    response.  If so, update param register and set
  %    improvement flag (param(5)) to 1.
  if bestf>pointData(1)
    param(1) = bestf;
    param(2) = bestx(1);
    param(3) = bestx(2);
    param(4) = step_length;
    param(5) = 1;
  % Else pass back updated step length, set improvement
  %    flag (param(5)) to 0, and restore pointData values
  else
    param(1) = pointData(1);
    param(2) = pointData(2);
    param(3) = pointData(3)
    param(4) = step_length;
    param(5) = 0;
  end
end
```

# Bibliography

[1] J.B. Armstrong and G.M. Malacinski, editors. *Developmental Biology of the Axolotl*. Oxford University Press, 1989.

[2] A. Blake. *Active contours: the application of techniques from graphics, vision, control theory and statistics to visual tracking of shapes in motion*. Springer, 1998.

[3] F.L. Bookstein. Fitting conic sections to scattered data. *Computer Graphics and Image Processing*, 9:56–71, 1979.

[4] P. Brodatz. *Textures: A Photographic Album for Artists and Designers*. Dover, 1966.

[5] G.W. Brodland. The Differential Interfacial Tension Hypothesis (DITH): A comprehensive theory for the self-rearrangement of embryonic cells and tissues. *ASME Journal of Biomechanical Engineering*, 124:188–197, 2002.

[6] G.W. Brodland. A computer model for reshaping of cells in epithelia due to in-plane deformation and annealing. *Comput Methods Biomech Biomed Engin*, 6(2):89–98, 2003.

[7] G.W. Brodland and D.A. Clausi. Embryonic tissue morphogenesis modelled by FEM. *ASME Journal of Biomechanical Engineering*, 116:146–155, 1994.

[8] G.W. Brodland and J.H. Veldhuis. Computer simulations of mitosis and interdependencies between mitosis orientation, cell shape and epithelia reshaping. *Journal of Biomechanics*, 32:673–681, 2002.

[9] J. Canny. A computational approach to edge detection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986.

[10] K.R. Castleman. *Digital Image Processing.* Prentice–Hall, 1996.

[11] H.H. Chen and G.W. Brodland. Cell-level finite element studies of viscous cells in planar aggregates. *ASME Journal of Biomechanical Engineering*, 122:394–401, 2000.

[12] D.A. Clausi. *Texture Segmentation of SAR Sea Ice Imagery.* PhD thesis, University of Waterloo, Waterloo, ON, Canada, N2L 3G1, 1996. http://rousseau.uwaterloo.ca/ dclausi/Theses/daclausi_PhD.ps.

[13] D.A. Clausi. Comparison and fusion of co-occurrence, Gabor, and MRF texture features for classification of SAR sea ice imagery. *Atmosphere & Oceans*, 39(4):183–194, 2001.

[14] D.A. Clausi and G.W. Brodland. Mechanical evaluation of theories of neurulation using computer simulations. *Development*, 118:1013–1023, 1993.

[15] D.A. Clausi and M.E. Jernigan. Designing Gabor filters for optimal texture separability. *Pattern Recognition*, 33:1835–1849, 2000.

[16] J.G. Daugman. Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortex filters. *Journal of the Optical Society of America A*, 2(7):1160–1169, 1985.

[17] A. Fitzgibbon, M. Pilu, and R.B. Fisher. Direct least square fitting of ellipses. *Pattern Analysis and Machine Intelligence*, 21(5):476–480, 1999.

[18] R. Hooke and T.A. Jeeves. "Direct search" solution for numerical and statistical problems. *Journal of Assoc. Computing Machinery*, 8:212–229, 1961.

[19] A.K. Jain and F. Farrokhnia. Unsupervised texture segmentation using Gabor filters. *Pattern Recognition*, 24(12):1167–1186, 1991.

[20] A. Leon-Garcia. *Probability and Random Processes for Electrical Engineering*. Addison-Wesley, 2nd edition, 1994.

[21] R.M. Lewis, V. Torczon, and M.W. Trosset. Why pattern search works. Technical Report TR9857, Institute for Computer Applications in Science and Engineering, 1998. http://techreports.larc.nasa.gov/icase/1998/icase-1998-57.pdf.

[22] A.V. Oppenheim and A.S. Willsky. *Signals and Systems*. Prentice–Hall, 2nd edition, 1996.

[23] E.P. Popov. *Mechanics of Materials*. Prentice-Hall, New Jersey, 2nd edition, 1976.

[24] A.R. Rao and G.L. Lohse. Identifying high level features of texture perception. *CVGIP: Graphical Models and Image Processing*, 33(3):218–233, 1993.

[25] S.M. Ross. *Introduction to Probability Models*. Academic Press, 6th edition, 1997.

[26] P.D. Sampson. Fitting conic sections to very scattered data: An iterative refinement of the bookstein algorithm. *Computer Graphics and Image Processing*, 18:97–108, 1982.

[27] G. Taubin. Estimation of planar curves, surfaces and non planar space curves defined by implicit equations, with applications to edge and range image segmentation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 13(11):1115–1138, 1991.

[28] V. Torczon. On the convergence of pattern search algorithms. *SIAM J. Optimization*, 7(1):1–25, 1997.