UNIVERSITY of CALIFORNIA
SANTA CRUZ

ANALYSIS AND APPROACHES TO IMAGE LOCAL ORIENTATION ESTIMATION

A thesis submitted in partial satisfaction of the
requirements for the degree of

MASTER OF SCIENCE

in

COMPUTER ENGINEERING

by

Xiaoguang Feng

March 2003

The Thesis of Xiaoguang Feng is approved:

Professor Peyman Milanfar, Chair

Professor Hai Tao

Professor Roberto Manduchi

Frank Talamantes
Vice Provost and Dean of Graduate Studies

Contents

List of Figures

Abstract

Analysis and Approaches To Image Local Orientation Estimation

by

Xiaoguang Feng

Image local orientation estimation plays an important role in many computer vision and image processing tasks such as edge detection, image segmentation, and texture analysis. The 2-D local orientation estimation is also directly related to optical flow estimation, which is the generalization of orientation in a 3-D space/time volume.

This thesis presents an image local orientation estimation method, which is based on a combination of two already well-known techniques: the principal component analysis (PCA) and the multiscale pyramid decomposition. The PCA analysis is applied to find the Maximum Likelihood (ML) estimate of the local orientation. The multiscale decomposition helps in improving the estimation robustness and accuracy. The underlying theories of the proposed method, such as directional statistics, the principle component analysis and the multiscale decomposition are introduced and discussed.

The proposed technique is shown to enjoy excellent robustness against noise. It is also shown to achieve an automatic balance between estimation stability and accuracy. We present both simulated and real image examples to demonstrate the proposed technique.

Acknowledgements

First of all, I would like to express my gratitude to my supervisor, Dr. Peyman Milanfar, for his continuous guidance and help throughout this research. I have learned a lot from him.

I also wish to thank the members of my thesis reading committee, Dr. Hai Tao and Dr. Roberto Manduchi, for their support and advices. I also need to thank Dr. Simon Clippingdale, for his kindly sharing his PhD thesis, Multiresolution Image Modelling and Estimation, which is very helpful for my work.

I want to thank my classmates in the Multi-Dimensional Signal Processing Lab, Dirk Robinson, Sina Farsiu, Amyn Poonawalla, Morteza Shahram and my office-mates YuanWei Jin and Rui Li, with whom I made many valuable discussions and learned much. Also thank all my friends in UCSC.

Above all, the love and understanding of my family and Claire support me everyday during the two years study life in US, making the completion of my graduate work possible. This work is for you.

Santa Cruz, CA March 20, 2003 Xiaoguang Feng

Chapter 1

Introduction

In this chapter we will briefly introduce the image local orientation estimation problem. We will discuss several previous approaches in orientation estimation. And finally, we will introduce the organization of this thesis.

1.1 Image Local Orientation

In some image processing or computer vision problems, what we are interested is not the gray level of the image, but some properties of the image feature. Image local orientation is an important image feature in many computer vision and image processing tasks such as edge detection [4], image segmentation, and texture analysis [25].

For example, in the fingerprint recognition problem, what we are interested in is the orientation of the fingerprint lines, instead of the brightness value of the fingerprint image.

Figure 1.1: A fingerprint image

The 2-D local orientation estimation is also directly related to optical flow estimation, which can be seen as the generalization of orientation in a 3-D space/time volume.

As mentioned in [22], the image orientation estimation problem can be generalized into n-dimension space. For example, in the RGB color space, a specific color corresponds to a 3-D vector. Then the orientation estimation of the *color vector* can be used to match or recognize object in color images.

1.2 Previous Works

Several techniques for orientation estimation have been proposed in the past. Here in this section we will give some brief introductions of these works.

Most established local orientation estimation techniques are based on the analysis of the local gradient field of the image. But the local gradients are very sensitive to noise, making the estimate of local orientation directly from these rather unreliable. How to deal with the noise effect is the major problem that all the gradient-based methods have to face. Perona [24] extended the idea of anisotropic diffusion in images to orientation maps. It is claimed that by carrying such diffusions forward for sufficiently long, accurate and effective estimates can be obtained. Bigun et al. [2] posed the orientation estimation problem as the least-squares fitting of an axis in the Fourier transform domain. The implementation is actually carried out in the image spatial domain without doing a Fourier transformation. The solution of the least-squares fitting problem is shown to be an eigenvalue solution of a matrix derived from the gradient field of the given image. The implementation method employed in [2] is directly related to the Principal Component Analysis(PCA) method we propose in this thesis. But the PCA method is numerically much more stable and efficient. Another set of techniques is based on steerable filters [11], but they are often limited in precision and generalization.

Since the noise sensitivity of the gradient operator is the major problem in gradient-based orientation estimation methods, there are several papers that discuss about optimizing the local gradient operators. Lyvers [17] examined the accuracy of different local differential operators in both noiseless case and in the presence of additive Gaussian noise. Zhou et al. [32] estimated the orientation by Gaussian gradient filter. The problem of this kind of methods is the scale of the analysis window. In order to reduce the noise sensitivity, the width of the Gaussian needs to be big enough, but this will cause the loss of the estimation localization. Costa et al. [6] introduced a new operator which is a combination of the gradient-like operator and the *valley-ness* operator. The combined operator can be used to estimate orientation in both the brightness sloped regions and brightness crests-valleys regions of the image. Also the operator can be optimized to get better bias reduction and noise robustness. The problem of this operator is it needs a prior knowledge of the texture model.

In contrast to most orientation estimation approaches which are based on gradient field, Mester [21] introduced an approach which directly estimates a small central part of the autocovariance function (*acf*) of the gray value image and derived the sought orientation from the direction of the main curvature in the *acf* origin [21]. It shows that the matrix of the eigenvalue problem in [2] can be directly determined from the autocovariance function, without computing the gradients. But this method needs much more computation than the classical simple gradient-based methods.

In order to solve the noise sensitivity problem of the gradient operator, there is often a need for some local or even global process to 'smooth' the estimate. However, by smoothing one gives up localization. A mechanism with both noise robustness and feature localization is needed. Wilson et al. [31] developed a multiscale orientation estimation approach, which is closely related to the one we propose here. But our method of combining the PCA and multiscale is novel, more efficient, and will give more robust results.

1.3 Thesis Organization

In this thesis we will introduce a multiscale principal component analysis method for image local orientation estimation problem. We will start with the introduction and discussion of some underlying theories. In Chapter 2, we will first talk about directional statistics. We will show that for zero mean Normal distribution random vectors, the conditional distribution of the vector orientation, given the norm of vectors equal to one, is the Bingham distribution. And for Bingham distribution, the optimal (Maximum Likelihood) estimation can be obtained by the Singular Value Decomposition (SVD) of the data vector matrix, which is the method we will use in the proposed estimation scheme.

In Chapter 3, we will discuss the SVD and its application in oriented energy analysis. We will develop the results of the random eigenvalue theory to get the distribution of an orientation dominance measure (the normalized difference of the singular values), which will be used in our estimation method.

We will present the multiscale estimation scheme in Chapter 4. We start with the introduction of the basic Gaussian and Laplacian pyramid. For our task of orientation estimation, we introduce a multiscale scheme which yields the minimum variance estimation. A Kalman-filter based optimal (Maximum Likelihood) multiscale scheme will be discussed and simplified for practical usage. In order to solve the blocking effect of the multiscale decomposition, we use an overlapped multiscale model and will give some analysis of its statistical properties.

In Chapter 5 we will describe the implementation of the proposed estimation method and provide some experimental results, including the comparisons of some different estimation methods, on both simulated and real image examples.

Finally, the Chapter 6 is the conclusion of the thesis and a further applications of the proposed multiscale principal component analysis method will be briefly introduced.

Chapter 2

Directional Statistics

In this chapter we will discuss the field of Directional Statistics, which can be seen as the underlying theory of the orientation estimation problem. We will introduce some common distributions and show that for the Bingham distribution, the Singular Value Decomposition(SVD) can be used to find the Maximum Likelihood(ML) estimation of the orientation, which is the method we propose in this thesis.

2.1 Introduction

Directional statistics [19] is mainly concerned with performing statistical analysis on the direction of unit random vectors. The standard statistical analysis method for linear random variables are not appropriate here. Special directional methods which take account of the periodic nature of the directional data are needed. For more details about the difference between directional statistics and conventional linear statistics, we refer the reader to [18] and [30].

Because the concerned vectors in the directional statistics are mainly unit vectors, the statistical behavior of these vectors can be studied by the distribution of the vector angle θ. In the next sections, we will introduce some commonly used distributions. We will start with the von-Mises distribution, which plays a key role in the directional statistics, analogous to the normal distribution in the conventional statistics. Actually, as we will show, the von-Mises distribution can be obtained by conditioning bivariate normal distribution [19].

In some cases, such as our problem of image orientation estimation, the orientation of an *undirected* line is of interest. In other words, the unit vectors $\mathbf{x}$ and $-\mathbf{x}$ are indistinguishable. Such observations can be described as *axes* rather than *directions* [19]. The probability density function($p.d.f$) of axes is antipodally symmetric:

$$f(\mathbf{x}) = f(-\mathbf{x}).$$

One important axes distribution is the Bingham distribution, which, in 2-D case, is the 2-wrapped von-Mises distribution and also can be obtained by conditioning zero-mean normal distributions on $\|\mathbf{x}\| = 1$.

What is the relationship between Directional Statistics and the image orientation estimation problem? As described in the first chapter, most image orientation estimation methods are based on the analysis of image gradient. For a given image $f(x, y)$, the gradient of the image is defined as:

$$g(x, y) \equiv (g_x, g_y)^T = (\frac{\partial f(x, y)}{\partial x}, \frac{\partial f(x, y)}{\partial y})^T$$

For discrete digital image, the gradient can be approximated by the difference of neighboring pixels. There are several differential operators for the approximation of gradient. In our work, we use the gradient() function in Matlab, which computes the gradient by:

$$\tilde{g}(x,y) \equiv (\tilde{g}_x, \tilde{g}_y)^T \approx (\frac{f(x+1,y)-f(x-1,y)}{2}, \frac{f(x,y+1)-f(x,y-1)}{2})^T \quad (2.1)$$

(For the effect of different gradient operators on the accuracy of orientation estimation, we refer the reader to [17].)

In this work, the image of interest is modeled as being composed of a noiseless feature image and additive random noise:

$$f(x,y) = f_0(x,y) + n(x,y)$$

Then the gradient vector $\tilde{g}$, which is approximated by (2.1), can also be seen as the sum of the gradient of the noiseless image and the gradient of the additive noise:

$$\tilde{g}(x,y) = \tilde{g}_0(x,y) + \tilde{g}_n(x,y). \quad (2.2)$$

From the determined vector $\tilde{g}_0(x,y)$, we can directly get the image orientation(by rotating the orientation of $\tilde{g}_0(x,y)$ by $\pi/2$ and normalizing the vector to unit length). Then the object of the orientation estimation problem is estimating the underlying *ideal* gradient vector $\tilde{g}_0(x,y)$ from the noise-corrupted gradient vector $\tilde{g}(x,y)$, which is a random vector with some distribution. If we restrict the additive noise as Gaussian noise, then the gradient vector $\tilde{g}(x,y)$ will also have Gaussian distribution. As we will show in next sections, Gaussian distributed vector field has close relationship with von-Mises distribution and Bingham distribution, for which the optimal (Maximum Likelihood) estimation of the orientation is known.

2.2 Von-Mises Distribution

The von-Mises distribution $M(\mu, \kappa)$ [19] has *p.d.f* as:

$$f(\theta; \mu, \kappa) = \frac{1}{2\pi I_0(\kappa)} e^{\kappa \cos(\theta - \mu)} \tag{2.3}$$

$$I_0(\kappa) = \frac{1}{2\pi} \int_0^{2\pi} e^{\kappa \cos(\theta)} d\theta \tag{2.4}$$

The parameter μ is the mean of the angles and the parameter κ is called *concentration parameter* [19]. The larger the value of κ, the greater the distribution clustering around the angle μ, just as shown in Fig. 2.1

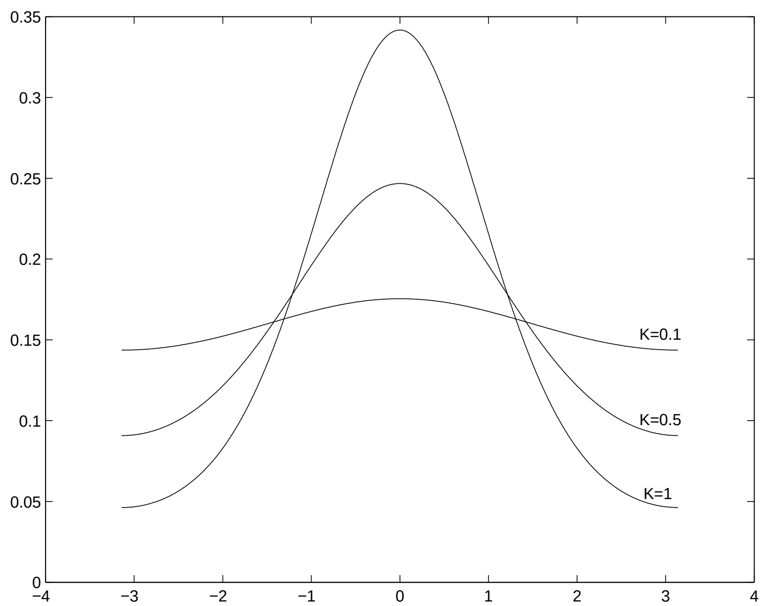


Figure 2.1: The *p.d.f* of Von-Mises distribution $M(0, \kappa)$

From the shape of the von-Mises distribution we can see, the von-Mises distribution is very similar to the normal distribution. Actually, for large κ, the von-Mises distribution $M(\mu, \kappa)$ can be approximated by wrapped normal distribution $N(\mu, 1/\kappa^2)$ [19]. (When $\kappa = 0$, $M(\mu, \kappa)$ is the uniform distribution.)

For us, the most important property of the von-Mises distribution is that it can arise from conditional normal distribution [19]. For a bivariate normal distribution vector $\mathbf{x} = r(\cos\theta, \sin\theta)^T$ with mean $\vec{\mu} = r(\cos\mu, \sin\mu)^T$ and covariance matrix $\kappa^{-1}\mathbf{I}_2$, the conditional distribution of θ given $r = 1$ is $M(\mu, \kappa)$.

For a set of independent and identically distributed ($i.i.d$) sample vectors, with angle $\theta_1, \ldots, \theta_n$, drawn from a von-Mises distribution $M(\mu, \kappa)$. The log-likelihood function is [19]:

$$
\begin{aligned}
l(\mu, \kappa; \theta_1, \ldots, \theta_n) &= n \log 2\pi + \kappa \sum_{i=1}^{n} \cos(\theta_i - \mu) - n \log I_0(\kappa) \\
&= n\{\log 2\pi + \kappa \bar{R} \cos(\bar{\theta} - \mu) - \log I_0(\kappa)\}.
\end{aligned}
$$

Here $\bar{\theta}$ is the mean of the sample angles and $\bar{R}$ is the average length of the sample vectors. For given κ, the log-likelihood function $l(\mu, \kappa; \theta_1, \ldots, \theta_n)$ is a function of μ, and it attain its maximum value when $\mu = \bar{\theta}$. Then the maximum likelihood estimate $\hat{\mu}$ of μ is the mean of the sample angles:

$$\hat{\mu} = \bar{\theta}.$$

We note that the von-Mises distribution is not directly applicable for the image orientation estimation problem, where the *axes*, instead of *direction*, is of interest.

We will next introduce Bingham distribution, which is an important axes distribution for our task, in the following section.

2.3 Bingham Distribution

Before we introduce the Bingham distribution, let us first go back to the relationship between bivariate normal distribution and von-Mises distribution. From last section we recall that the von-Mises distribution can be obtained by conditioning normal distribution $N(\mu, \kappa^{-1}\mathbf{I}_2)$ with $\|\mathbf{x}\| = 1$. We can extend this relationship to a more general case [19]. Let $\mathbf{x}$ have a normal distribution as:

$$\mathbf{x} \sim N(-\frac{1}{2}\kappa(A + c\mathbf{I}_2)^{-1}\mu, -\frac{1}{2}(A + c\mathbf{I}_2)^{-1}). \tag{2.5}$$

where A is a symmetric 2×2 matrix and c is such that $A + c\mathbf{I}_2$ is negative definite. Then the conditional density of $\mathbf{x}$ given $\|\mathbf{x}\| = 1$ is:

$$f(\mathbf{x}; \mu, \kappa, A) = \frac{1}{\alpha(\kappa, A)} \exp\{\kappa\mu^T\mathbf{x} + \mathbf{x}^T A\mathbf{x}\}, \tag{2.6}$$

which is called the Fisher-Bingham distribution [19].

If $A = 0$ then (2.5) reduces to white Gaussian distribution, and the Fisher-Bingham distribution (2.6) will become the von-Mises distribution. If $\kappa = 0$ then (2.5) reduces to a zero-mean normal distribution, and the Fisher-Bingham distribution (2.6) will become the Bingham distribution. If both $A = 0$ and $\kappa = 0$, then (2.5) reduces to a simple zero-mean white normal distribution, and (2.6) will turn into a uniform distribution.

The $p.d.f$ of Bingham distribution is [19]:

$$f(\pm\mathbf{x}; A) =_1 F_1(\frac{1}{2}, \frac{p}{2}, A)^{-1} \exp\{\mathbf{x}^T A\mathbf{x}\} \tag{2.7}$$

$${}_1F_1(\frac{1}{2}, \frac{p}{2}, A) = \int_{S^{p-1}} e^{\mathbf{x}^T A\mathbf{x}} d\mathbf{x}. \tag{2.8}$$

where p is the dimension of the vectors, S^{p-1} is the unit sphere in p dimensions. In the 2-D case, the Bingham distribution reduces to the *2-wrapped* von-Mises distribution, which means, given the $p.d.f$ of von-Mises distribution $f(\theta)$, the $p.d.f$ of Bingham distribution $f^*(\theta) = f(2\theta)$ [19]:

$$A = \begin{bmatrix} \cos 2\mu & \sin 2\mu \\ \sin 2\mu & -\cos 2\mu \end{bmatrix}$$

$$f(\theta; \mu, \kappa) = \frac{1}{2\pi I_0(\kappa)} e^{\kappa \cos 2(\theta-\mu)}.$$

Bingham distribution is important for our task of image orientation estimation because it can arise from conditional normal distribution. We will discuss the optimal (Maximum Likelihood) estimate for Bingham distribution in next section.

2.4 Maximum Likelihood Estimation

For a set of random vectors $\mathbf{x}_1, \ldots, \mathbf{x}_n$ from the Bingham distribution (2.7), the log-likelihood function is:

$$l(A; \mathbf{x}_1, \ldots, \mathbf{x}_n) = n\{\log tr(A\bar{T}) - \log_1 F_1(\frac{1}{2}, \frac{p}{2}, A)\}$$

$$\bar{T} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i \mathbf{x}_i^T.$$

From [19], the maximum of $tr(A\bar{T})$ can be obtained when $\mu = \arg(\mathbf{v}_1)$, where $\mathbf{v}_1$ is the first eigenvector of the matrix $\bar{T}$. Then the maximum likelihood estimate $\hat{\mu}$ of μ is:

$$\hat{\mu} = \arg(\mathbf{v}_1).$$

Note that this eigenvector of the scatter matrix $\bar{T}$ is equivalent to the singular vector of the data matrix $\mathbf{X}$, corresponding to its largest singular value:

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1^T \\ \mathbf{x}_2^T \\ \vdots \\ \mathbf{x}_n^T \end{bmatrix} = \begin{bmatrix} r_1 \cos\theta_1 & r_1 \sin\theta_1 \\ r_2 \cos\theta_2 & r_2 \sin\theta_2 \\ \vdots & \vdots \\ r_n \cos\theta_n & r_n \sin\theta_n \end{bmatrix}. \tag{2.9}$$

And the singular value decomposition(SVD) method is numerically much more stable and efficient. We will discuss more details about the SVD method in the next chapter.

How to apply this maximum likelihood estimate to the image orientation estimation problem? Remember in the first section of this chapter, we describe the gradient vectors of the concerned image as a set of random vectors with some distribution. If we assume the additive noise to be Gaussian noise (the most common assumption in signal/image processing), then the gradient vectors $\mathbf{g}$ will also have Gaussian distribution: $\mathbf{g} \sim N(\mu, C)$. As we discussed before, if the gradient vectors have white Gaussian distribution, say, $\mathbf{g} \sim N(\mu, \sigma^2\mathbf{I}_2)$, then the *direction* of the vectors given $\|\mathbf{g}\| = 1$ is von-Mises distribution. And the maximum likelihood estimate of the underlying dominant *direction* is given by the mean of the sample vectors (Note that this is not applicable for our task, where the *axes* but not *direction* is interested). If the gradient vectors have zero-mean Gaussian distribution, $\mathbf{g} \sim N(0, C)$, then the *axes* of the gradient vectors given $\|\mathbf{g}\| = 1$ is Bingham distribution. And the maximum likelihood estimate of the underlying dominant *axes* can be obtained from the singular value decomposition of the data matrix (2.9), which is the method we proposed in this thesis.

Note that for most case in the image orientation estimation problem, the gradients can be locally modeled as a deterministic gradient vector plus a noise vector, as shown in (2.2). Suppose the additive noise vectors have zero mean and the estimation window is big enough to contain multiple image feature lines, then the determine gradient vectors also have zero mean, and finally the noisy gradient vectors have zero mean. The assumption of zero mean is satisfied in most of our test images.

In this chapter, we introduced the underlying theory of the image orientation problem: directional statistics. As we know, in the literature dealing with image orientation, there is rarely an explicit reference to the field of directional statistics. We showed that for zero-mean normal distribution gradient vectors, which is true for most case in our tests, the *axes* of those vectors has Bingham distribution and the maximum likelihood estimate of the dominant orientation can be obtained by the singular value decomposition method. We will discuss more details of the SVD method in the next chapter.

Chapter 3

Principal Component Analysis

In this chapter we will introduce the Principal Component Analysis (PCA) method for image orientation estimation. As described in last chapter, the Singular Value Decomposition (SVD), which is the computational basis of PCA, can be used to obtain the maximum likelihood estimate of the dominant orientation for images with Gaussian additive noise. In this chapter we will discuss the SVD and the orientation estimation problem from another point of view. We will also introduce an orientation dominance measure R and discuss its statistical behavior.

3.1 Introduction

Principal Components Analysis is used to compute the dominant vectors representing a given data set and also provides an optimal basis for minimum mean-squared reconstruction of the given data. It is also sometimes referred to as the Karhunen-Loeve Transform [7]. The computational basis of PCA is the calculation of the Singular Value Decomposition of the data matrix, or equivalently the eigenvalue-decomposition of the data covariance matrix. Here we describe the method in terms of the SVD. For more details about SVD and PCA, we refer the reader to [7], [29] and [23].

3.2 Singular Value Decomposition

Any given $m \times n$ matrix A (without losing generality, assume $m > n$) can be decomposed as:

$$A = USV^T, \tag{3.1}$$

where S is an augmented diagonal $m \times n$ matrix with entries $s_1, \ldots, s_n$:

$$S = \begin{bmatrix} s_1 & & \\ & \ddots & \\ & & s_n \\ 0 & \cdots & 0 \\ & \vdots & \\ 0 & \cdots & 0 \end{bmatrix}_{m \times n}.$$

U and V are orthonormal matrices such that U is $m \times m$ and V is $n \times n$. The diagonal entries of S, $\{s_1, \ldots, s_n\}$, are called the singular values of matrix A, the columns of the matrix U are called the left singular vectors of A and the columns of the matrix V (or the rows of matrix V^T) are called the right singular vectors of A.

Since the last $m - n$ rows of the matrix S are all zeros, the corresponding last

$m-n$ left singular vectors $\mathbf{u}_i, i=n+1,\ldots,m$ are not used. Another more efficient way to write the decomposition is called *economical* SVD, in which the matrix U is an $m \times n$ orthonormal matrix that only contains the first n left singular vectors, S is an $n \times n$ diagonal square matrix. We will use the economical SVD in our method.

One important application of SVD in signal processing is the analysis of *oriented energy* [7]. The *oriented energy* of a matrix A, measured in the direction $\mathbf{q}$, is defined as:

$$E_q(A) = \sum_{k=1}^{n} (\mathbf{q}^T \mathbf{a}_k)^2,$$

where $\mathbf{a}_k, k=1,\ldots,n$ are the columns of A and $\|\mathbf{q}\| = 1$. Doing SVD of A as shown in (3.1), then each direction of the critical oriented energy is generated by a right singular vector with the critical energy equal to the corresponding singular value squared [7]. The left singular vectors represent each sample's *contribution* to the corresponding principal direction. This can be illustrated in 2-D in Figure 3.1.

In Figure 3.1, the sample points have maximal oriented energy at the angle $\pi/4$. The two vectors represent the singular vectors of the sample point matrix, which is a $n \times 2$ matrix, where n is the number of sample points and the two columns are the x and y values of the sample points. The lengths of the vectors represent the corresponding average energy in that direction: $e_i = s_i^2/n, i=1,2$, where s_i are the corresponding singular values.

How can we apply the oriented energy analysis to the image orientation estimation problem? As discussed in Chapter 2, for zero-mean Gaussian distributed

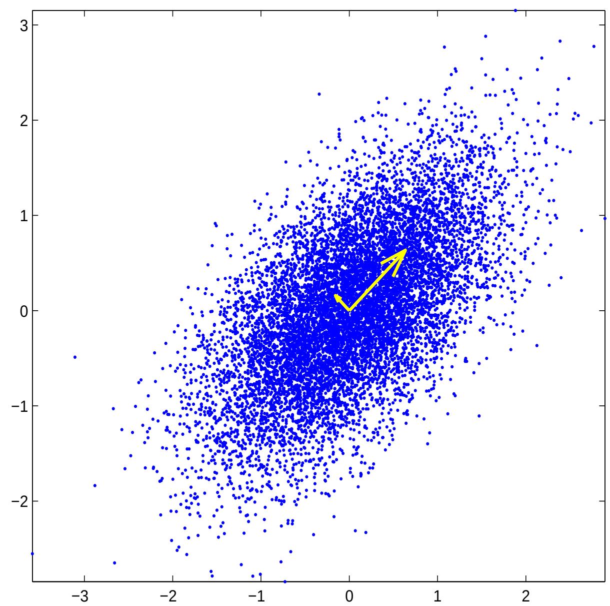


Figure 3.1: Oriented Energy and Singular Vectors

gradient vectors, the maximum likelihood estimate of the underlying orientation can be obtained by the SVD of the gradient matrix. Here we will discuss this problem as a least square estimation [10], without considering the distribution.

Let us assume that in the image of interest $f(x, y)$, the orientation field is piecewise constant (this assumption is only needed to get a single estimate at each location, we don't need this assumption to implement our approach). Under this assumption, the gradient vectors in an image block should *on average* be orthogonal to the dominant orientation of the image pattern. So orientation estimation can be formulated as the task of finding a unit vector $\mathbf{q}$, to maximize the average of angles θ_i between $\mathbf{q}$ and gradient vectors $\mathbf{g}_i$ ($i = 1, \ldots, n$, within a local window), as shown in Figure 3.2.

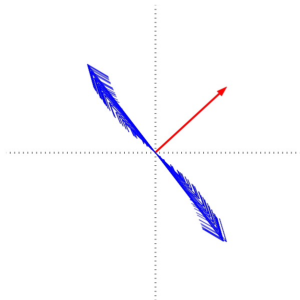


Figure 3.2: Gradient vectors and the dominant orientation.

Equivalently we can minimize:

$$\sum_{i=1}^{n}(\mathbf{q}^T\mathbf{g}_i)^2 = \mathbf{q}^T\sum_{i=1}^{n}(\mathbf{g}_i\mathbf{g}_i^T)\mathbf{q} = \mathbf{q}^T C\mathbf{q}$$

subject to $\|\mathbf{q}\| = 1$, where

$$C = \begin{bmatrix} \sum_{i=1}^{n} g_{i,x}^2 & \sum_{i=1}^{n} g_{i,x}g_{i,y} \\ \sum_{i=1}^{n} g_{i,x}g_{i,y} & \sum_{i=1}^{n} g_{i,y}^2 \end{bmatrix},$$

$g_{i,x}$ and $g_{i,y}$ are derivatives in $\mathbf{x}$ and $\mathbf{y}$ direction at pixel location indexed by i, respectively. This problem is a constrained optimization problem and can be solved using Lagrange multipliers:

$$f(\mathbf{q},\lambda) = \mathbf{q}^T C\mathbf{q} - \lambda(\mathbf{q}^T\mathbf{q} - 1)$$

The extremum can be found by taking the derivative of $f(\mathbf{q},\lambda)$ with respect to $\mathbf{q}$:

$$\begin{aligned} \frac{\partial f}{\partial \mathbf{q}} &= \frac{\partial}{\partial \mathbf{q}}(\mathbf{q}^T C\mathbf{q}) - \lambda\frac{\partial}{\partial \mathbf{q}}(\mathbf{q}^T\mathbf{q}) \\ &= 2C\mathbf{q} - 2\lambda\mathbf{q}. \end{aligned}$$

Set it equal to zero, we will get:

$$(C - \lambda I)\mathbf{q} = 0.$$

This is a standard eigenvalue problem. So the unit vector $\mathbf{q}$ minimizing $\mathbf{q}^T C\mathbf{q}$ is the eigenvector of C corresponding to the smallest eigenvalue, or equivalently, the singular vector of $\mathbf{X}$ corresponding to the smallest singular value; $\mathbf{X}$ is defined in (2.9). Note that the SVD method is numerically much more stable and efficient.

In order to get the *local* orientation estimate, we first divide the region of interest into small blocks. Within each block, rearrange the gradient vectors $\mathbf{g}_i = [g_{i,x}, g_{i,y}]^T, i = 1, \ldots, n$ into an $n \times 2$ matrix as (2.9). Then compute the economical SVD of the gradient matrix:

$$\begin{bmatrix} g_{1,x} & g_{1,y} \\ g_{2,x} & g_{2,y} \\ \vdots & \vdots \\ g_{n,x} & g_{n,y} \end{bmatrix} = \begin{bmatrix} u_{1,x} & u_{1,y} \\ u_{2,x} & u_{2,y} \\ \vdots & \vdots \\ u_{n,x} & u_{n,y} \end{bmatrix} \begin{bmatrix} s_1 & 0 \\ 0 & s_2 \end{bmatrix} \begin{bmatrix} v_{11} & v_{12} \\ v_{21} & v_{22} \end{bmatrix}$$

The image orientation can be obtained from the first right singular vector (note that the gradient vectors are orthogonal to the image orientation we seek, so after obtaining the principal direction of the gradient vectors, we need to rotate by $\pi/2$ to get the orientation we want). Recall that the singular values represent the square root of the energy in corresponding principal directions. So the difference between the singular value s_1 and s_2 can be used as a measure of accuracy or dominance of the estimate. However, since $s_1 - s_2$ is an energy dependent measure, the quantity

$$R = \frac{s_1 - s_2}{s_1 + s_2} \tag{3.2}$$

will be more suitable for this task [2]. Note that when the gradient vectors have only one *clean* dominant orientation, s_2 will equal to zero and then R will equal to 1. On the other hand, if there is no dominant orientation, $s_1 \approx s_2$ and then R will be close to zero. For other intermediate cases, R has values ranging from 0 to 1.

We will discuss in more details the statistical behavior of the dominance measure R, in next section.

3.3 Distribution of Random Singular Values

It is well known in the earlier works that the singular values can be seen as the measure of *concentration* around the principal axes [20]. In this section, we will try to discuss the statistical properties of the dominance measure R in a more systematic way for the orientation estimation problem.

First, note that R, defined as a normalized difference of the singular values in (3.2), is related to the condition number k of the gradient matrix X:

$$k = \frac{s_1}{s_2},$$

$$R = \frac{k-1}{k+1}.$$

From the theory of random matrices [9], we know the *p.d.f* of the condition number k of an $n \times 2$ zero mean *i.i.d* white Gaussian matrix is:

$$f_K(k) = (n-1)2^{n-1}\frac{k^2-1}{(k^2+1)^n}k^{n-2}$$

(here we discuss $n \times 2$ matrix because in our approach of image orientation estimation, the gradient matrix is $n \times 2$, but much more general results are available). From basic statistics we know, given the $p.d.f$ of a random variable x as $f_X(x)$, and a function $x = h(y)$, then the $p.d.f$ of y can be calculated as:

$$f_Y(y) = f_X(h(y))|h'(y)|$$

In our case, we know the $p.d.f$ of the condition number k, and we know the function relating k to R, we are trying to get the $p.d.f$ of R:

$$R = \frac{k-1}{k+1} \Longrightarrow k = \frac{1+R}{1-R},$$
$$\frac{\partial}{\partial R}\left(\frac{1+R}{1-R}\right) = \frac{2}{(1-R)^2},$$

then we have:

$$f_R(R) = (n-1)2^{n-1}\frac{(\frac{1+R}{1-R})^2 - 1}{((\frac{1+R}{1-R})^2+1)^n}\left(\frac{1+R}{1-R}\right)^{n-2}\frac{2}{(1-R)^2}.$$

Simplifying, we get the $p.d.f$ of R for an $n \times 2$ zero mean $i.i.d$ Gaussian matrix:

$$f_R(R) = 4(n-1)R\frac{(1-R^2)^{n-2}}{(1+R^2)^n} \tag{3.3}$$

This density is illustrated in Figure 3.3.

Note that this distribution is independent of the variance of the Gaussian matrix, but only depends on the size of the matrix, as shown in Figure 3.4.

This $p.d.f$ is only valid when the Gaussian matrix has zero mean. Monte Carlo simulation shows that by changing the mean value, the shape and the position of the $p.d.f$ will also change, as shown in Figure 3.5.

How to make use of this $p.d.f$ in the orientation estimation problem? We can use this $p.d.f$ to do a significance test [8] to distinguish between a pure noise image

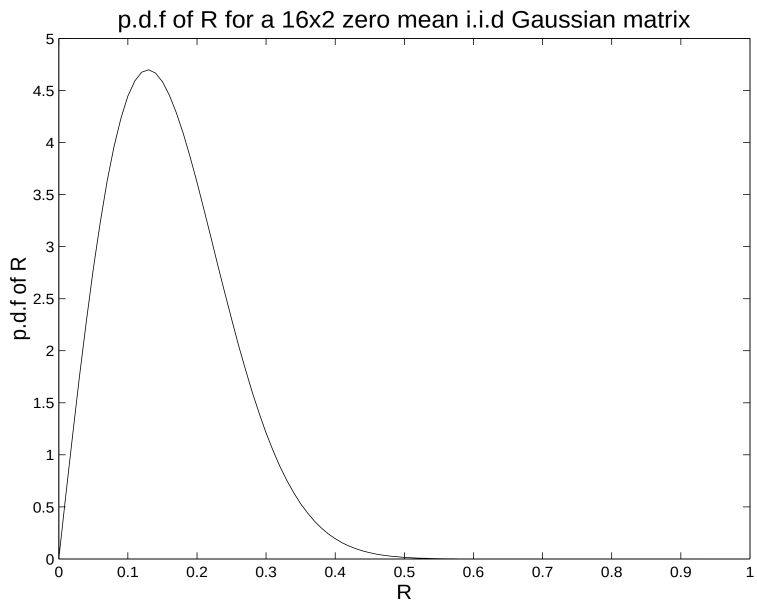


Figure 3.3: The $p.d.f$ of dominance measure R, $n = 16$.

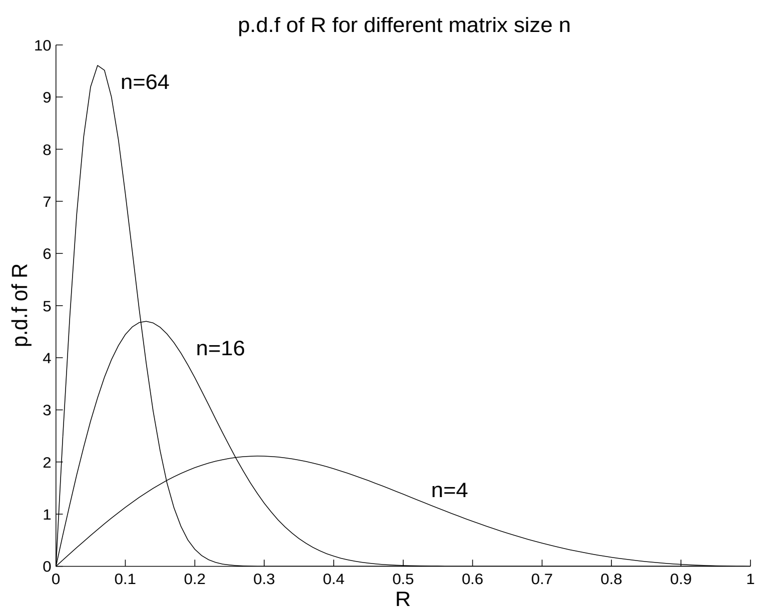


Figure 3.4: The $p.d.f$ of R for different n.

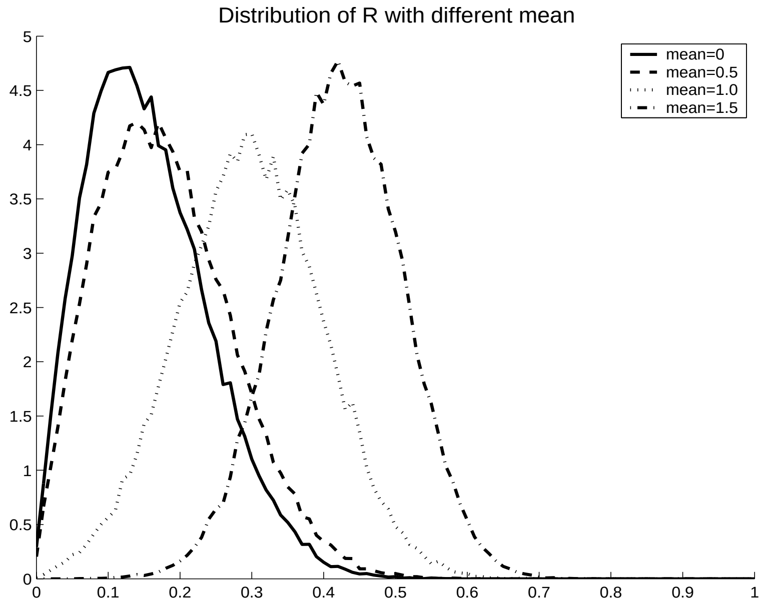


Figure 3.5: The $p.d.f$ of R for different mean (n=16).

and an image with orientation pattern. First we set a *significance level* threshold R^*. For any given image block, we perform the PCA-based estimation, if the R is less than R^*, it is very likely that the corresponding image block is only pure white noise and contains no orientation information. As shown in Figure 3.6.

In Figure 3.6, the test image (shown in left side) has oriented lines with additive noise. We do the estimation within 4×4 blocks. The estimated orientation

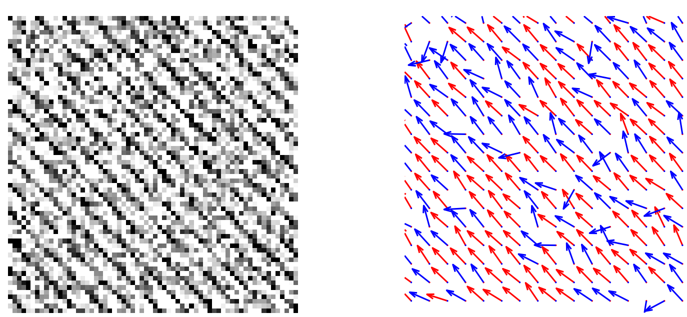


Figure 3.6: A significance test on the dominance measure R

map is shown in the right part of Figure 3.6. If we set the significance threshold as 0.3 (significance level of 93%), all the blue vectors will be eliminated. We can see that, by doing so, we can eliminate most error estimates, but the problem is we will also eliminate some correct estimates. That is the limitation of the significance test [8].

Note that the value of the dominance measure R is related to the size of the estimation blocks. So only by knowing the $p.d.f$ of R, as a function of the block size, we can determine the threshold appropriately.

3.4 Conclusion

In this chapter we discussed the Principal Component Analysis in the orientation estimation problem. We presented the method of using SVD to get the least-square estimate of the orientation. We also introduced an orientation dominance measure R and discussed the statistical properties of R, which have not been fully discussed in previous works. Our tests show that the SVD method will give quite accurate and stable estimate of the image orientation, as shown in Figure 3.7 and 3.8.

In the next chapter, we will introduce multiscale decomposition, which will further improve the stability of the estimate and get adaptive compromise between the resolution of the orientation field and its relative accuracy.

Figure 3.7: The fingerprint test image

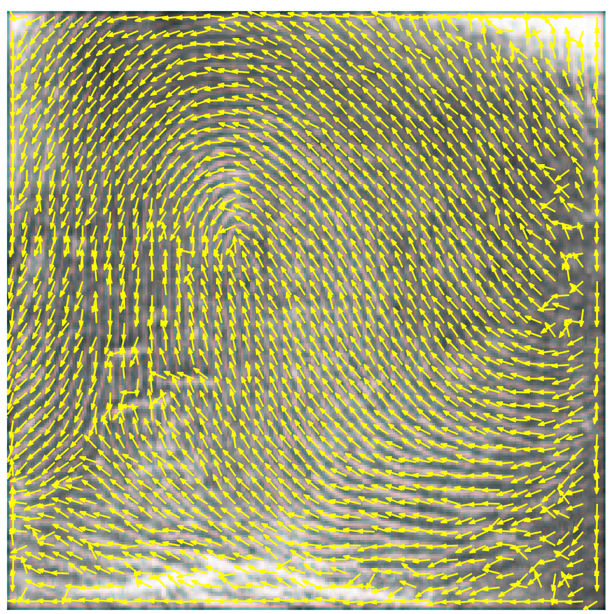


Figure 3.8: The fingerprint image with orientation map overlaid.

Chapter 4

Multiscale Estimation

In this chapter we discuss a multiscale approach to the PCA-based orientation estimation. We will start with the pyramid decomposition, and discuss how to choose propagation weights in the multiscale approach to obtain optimal estimates. We will also introduce an overlapped multiscale model, which can be used to eliminate the blocking effect of the multiscale method.

4.1 Introduction

Multiscale signal and image analysis have been investigated for some time with applications in data compression [3], edge detection [26], and segmentation [28], etc. The multiscale model describes images in terms of an evolution from coarse to fine scales, which is a more natural way than the common raster scan model and is also more similar to the human visual system. Another benefit of the multiscale approach is the significant computational efficiency. For more discussions about the multiscale image model, we refer the reader to [15].

Why use the multiscale approach in image orientation estimation? As we discussed in the first chapter, a major problem of the orientation estimation is the noise sensitivity of the gradient operator. In order to depress the noise effect, one solution is using larger estimate window, since more neighboring gradients will be used to get the estimate and the averaging process will depress or eliminate the noise effect. But this will cause the loss of the estimate resolution. A mechanism with both noise robustness and feature localization is needed. Multiscale model provides an efficient way to combine the information from coarse scales and fine scales. As we will describe in following sections, by choosing appropriate propagation weights, the multiscale approach can be used to obtain Minimum Mean Square Error ($MMSE$) estimate, which is based on a Kalman filter-like multiscale model, introduced in [5].

A problem of the multiscale approach is the blocking effect. Actually, blocking effect exits in any block-based method, but because multiscale approach combines information from coarse scales, with larger block, this makes the blocking effect more obvious. A simple way to eliminate the blocking effect is doing spatial smoothing on the resulting estimate. But this smoothing will cause a decrease in the resolution of the estimate. Another, better, method to solve this problem is using overlapped blocks in the multiscale model, as introduced in [14] and [27]. We will use the idea of overlapped blocks in our approach and discuss its properties.

According to our research of the previous literature in orientation estimation,

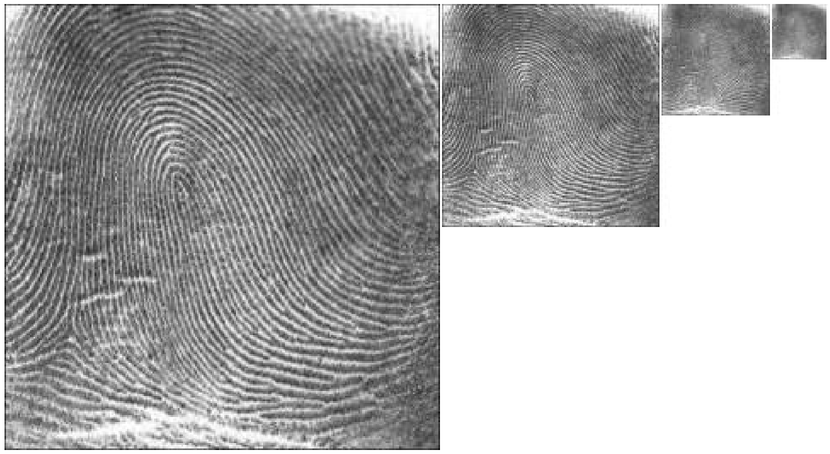


Figure 4.1: An example of image pyramid.

there is only one paper clearly talking about using multiscale estimate approach [31], by R Wilson et al., their method is closely related to the one we propose here. But our method of combining the PCA and multiscale is novel, more efficient, and will give more robust results. We will discuss more details about the difference of our method and Wilson's approach in the last section of this chapter.

4.2 Pyramid-based Estimation

Before introducing our multiscale estimation approach, we first need to define the image pyramid. Image pyramid is a collection of copies of an original image in which both sample density and resolution are decreased [1], as an example shown in Figure 4.1.

The bottom layer of the pyramid is the original image. Without losing generality, denote the first(bottom) layer as G_0, and increase the layer number index as the pyramid goes up to lower resolution. More specifically, low-pass-filter G_0 and subsample it by 2 to obtain the next pyramid layer G_1. Repeat the low-pass-filter/subsample steps to generate remaining pyramid layers [1]:

$$G_l(i,j) = \sum_m \sum_n w(m,n) G_{l-1}(2i+m, 2j+n)$$

where $0 < l < N$, N is the total number of layers in the pyramid, and $w(m,n)$ is the low pass filter weighting function. If we use Gaussian low pass filter, the image pyramid is often referred as a *Gaussian pyramid*. Another type of image pyramid, which is often referred to as *Laplacian pyramid*, is obtained by recursively subtracting each Gaussian pyramid layer (up-sampled to fit the size) from the next high resolution layer [15]. Laplacian pyramid is composed of bandpass images. Since our goal of using multiscale approach is to depress noise effect, Gaussian pyramid is more appropriate for our task.

There are three major steps in our multiscale estimation approach, as shown in Figure 4.2.

1. **Built up the image pyramid**. From the original image, repeat the low-pass-filter/subsample steps as described in this section to get the Gaussian image pyramid, then calculate gradients at each scale. Another option is to calculate the gradients of the original image first, then build up a gradient pyramid. These two options will give similar results, as we will show the test results in the next chapter.

2. **PCA-based estimation on each layer**. On each layer of the image pyramid(or gradient pyramid), use the PCA-based method introduced in Chapter 3

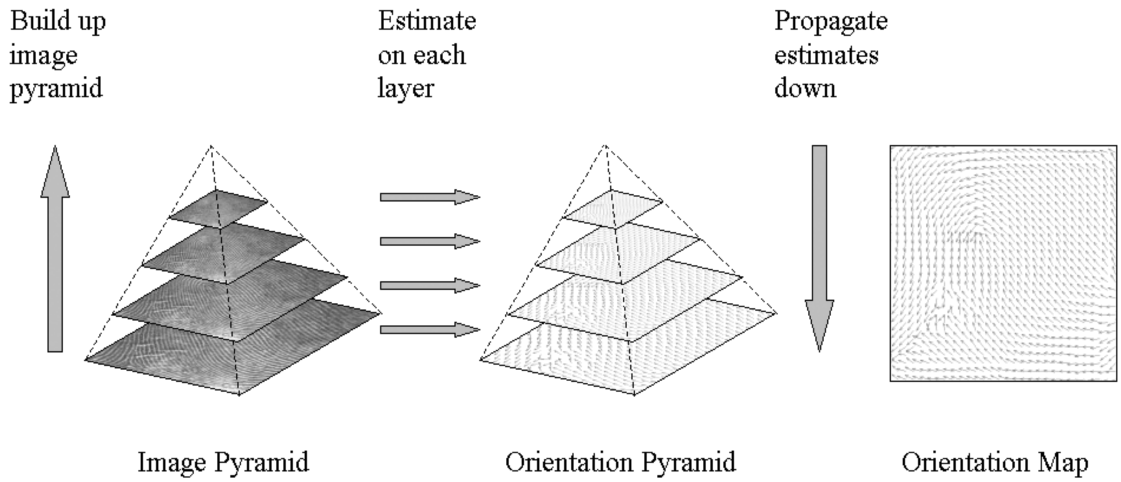


Figure 4.2: Multiscale estimation approach.

to estimate the local orientation. This will produce an orientation pyramid, with orientation maps in different resolutions. Notice in this step the interest image can be divided into non-overlapped blocks or overlapped blocks. We will discuss more details later in this chapter.

3. **Propagate down the estimation from low resolution to high resolution**. From the top layer of the orientation pyramid (low resolution), propagate the estimates down. Combine the estimates from different scales to get a final orientation map.

Among these three steps, we have discussed the first two. Now the problem is how to combine different scale estimations to get an optimum orientation map. We will discuss this issue in next two sections.

4.3 Minimum Variance Estimation

The most straightforward way to propagate the estimates from different scales is weighted linear combination.

$$\hat{\theta}_n = k[n]\theta_{\gamma n} + (1 - k[n])\theta_n. \tag{4.1}$$

where θ_n is the estimated orientation angle map on current layer, $\theta_{\gamma n}$ is the estimated orientation angle map on the parent layer(with lower resolution, up-sampled to fit the size), $k[n]$ is the propagation weight, and $\hat{\theta}_n$ is the resulting orientation estimate for the current layer. Note that, in order to simplify the description, in this section, we use orientation angles to model the orientation, while in other sections we use unit vectors to model it. As we will show in next section, by assuming the variance in x direction and in y direction are equal, these two models are equivalent.

If we choose the linear combination approach, a natural question is how to choose the propagation weight $k[n]$? One commonly used criterion is minimum variance of the resulting estimate. From (4.1), the variance of $\hat{\theta}_n$ can be calculated as:

$$\sigma_{\hat{n}}^2 = k^2[n]\sigma_{\gamma n}^2 + (1 - k[n])^2\sigma_n^2 + 2k[n](1 - k[n])C_{n,\gamma n},$$

where σ_n^2 is the variance of orientation in the current layer, $\sigma_{\gamma n}^2$ is the variance of orientation in the parent layer and $C_{n,\gamma n}$ is the covariance of the orientation between current layer and the parent layer. The differential of $\sigma_{\hat{n}}^2$ with respect to the propagation weight $k[n]$ is:

$$\frac{\partial \sigma_{\hat{n}}^2}{\partial k[n]} = 2k[n]\sigma_{\gamma n}^2 - 2(1 - k[n])\sigma_n^2 + 2(1 - 2k[n])C_{n,\gamma n}.$$

Setting the above equal to zero we will yield the propagation weight $k[n]$ to reach the extremum of the variance of $\hat{\theta}_n$ as:

$$k[n] = \frac{\sigma_{\gamma n}^2 - C_{n,\gamma n}}{\sigma_n^2 + \sigma_{\gamma n}^2 - 2C_{n,\gamma n}}$$

The second derivative of $\sigma_{\hat{n}}^2$ is:

$$\frac{\partial^2 \sigma_{\hat{n}}^2}{\partial k^2[n]} = 2\sigma_{\gamma n}^2 + 2\sigma_n^2 - 4C_{n,\gamma n}$$

From the observation of our simulations, the covariance of the orientation between two layers is very small compared with the variance of the orientation in each layer. Hence, the second derivative of $\sigma_{\hat{n}}^2$ can be simplified as:

$$\frac{\partial^2 \sigma_{\hat{n}}^2}{\partial k^2[n]} \approx 2\sigma_{\gamma n}^2 + 2\sigma_n^2 > 0,$$

which means the extremum is minimum. And we can also simplify the propagation weight as:

$$k[n] \approx \frac{\sigma_{\gamma n}^2}{\sigma_n^2 + \sigma_{\gamma n}^2}. \tag{4.2}$$

Note that here calculating the variance of orientation angles is a computationally expensive task. In order to further simplify the computation, recall that in Chapter 3, we introduced an *orientation dominance* measure R (3.2). As we described, the value of R can be viewed to be *inversely proportional* to the variance of orientation. Thus, we elect to define the propagation weight in terms of R, which can be directly obtained from the SVD step:

$$k[n] \approx \frac{R_n}{R_n + R_{\gamma n}}. \tag{4.3}$$

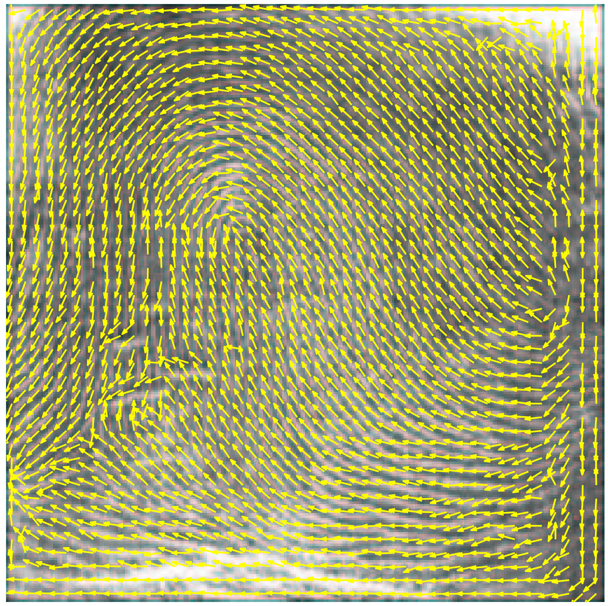


Figure 4.3: Multiscale estimation result of Figure 3.7.

Our tests show that this propagation weight can produce very robust estimation, and since it is a byproduct of the SVD step, it is efficient and will not require additional computations. A multiscale estimation result is shown in Figure 4.3. Here we use the same estimation window size as the single scale method in Figure 3.8, but the multiscale method will produce a more smooth orientation map. From the estimated orientation angle maps of these two methods in Figure 4.4, it is easy to see that the multiscale method has less *noisy* estimates than the single scale method. More test results will be shown in next chapter.

Note that the calculation of gradient on the image boundary is not accurate. So the orientation estimation on the boundary is not reliable. In the multiscale approach, since we make use of the estimates from low resolution layers, after being propagated onto the resulting orientation map, the boundary effect is increased, which

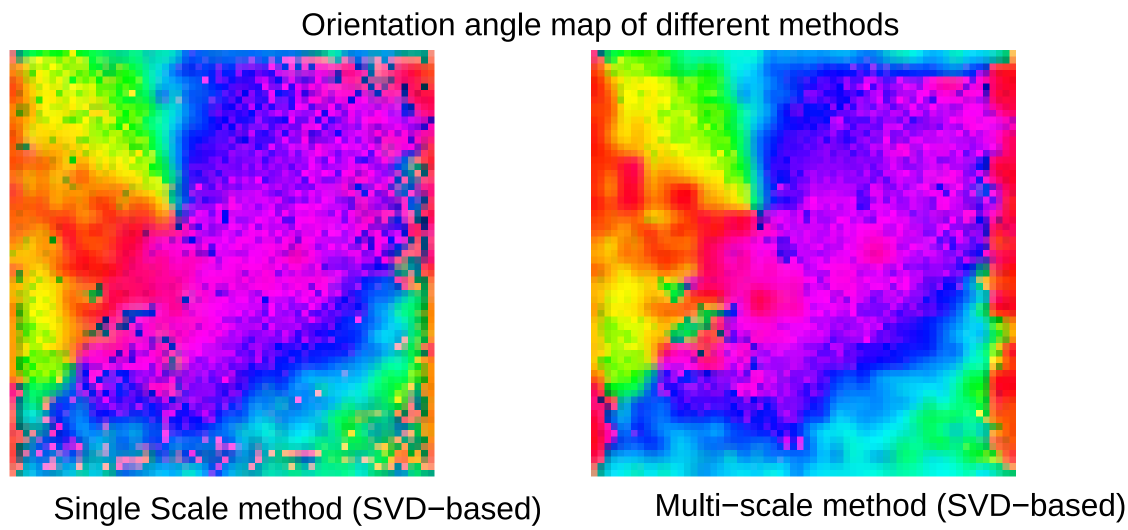


Figure 4.4: Orientation angle maps of single-scale and multiscale methods.

can been seen from Figure 4.3 and 4.4. We can do some post-processing to eliminate the boundary effect, or extend the image in various ways like replication, or mirroring.

In next section, we introduce a Kalman filter-like multiscale scheme. It is shown that, this multiscale scheme can also lead to the propagation weight in (4.3).

4.4 Multiscale Kalman Filter

As introduced in [5], multiscale estimation can be modeled within the framework of Kalman filtering across scales:

$$s[n] = A[n]s[\gamma n] + B[n]w[n]$$

$$x[n] = H[n]s[n] + v[n]$$

where $s[n]$ is the signal in the current layer; $s[\gamma n]$ is the signal in the parent layer (coarser resolution); $x[n]$ is the measurement in the current layer; $w[n]$ and $v[n]$ are the innovation signal and measurement noise in current layer, respectively. The covariance matrix of $v[n]$ is denoted as $C[n]$.

From the theory of Kalman Filter [16], the MMSE estimate $\hat{s}[n|n]$ of the signal $s[n]$, given all the low resolution layers, can be obtained recursively as:

$$\hat{s}[n|n] = \hat{s}[n|\gamma n] + K[n](x[n] - H[n]\hat{s}[n|\gamma n]),$$

where

$$\hat{s}[n|\gamma n] = A\hat{s}[\gamma n|\gamma n],$$

$$K[n] = M[n|\gamma n]H^T[n](C[n] + H[n]M[n|\gamma n]H^T[n])^{-1},$$

$$M[n|\gamma n] = E\{(s[n] - \hat{s}[n|\gamma n])(s[n] - \hat{s}[n|\gamma n])^T\}. \tag{4.4}$$

In the task of image orientation estimation, the signal $s[n]$ is the locally dominant orientation vector (or equivalently, the locally dominant gradient vector). The signal in the current layer can be modeled as the corresponding signal from parent layer (up-sampled to fit the size) plus an innovation vector. And the measurement is simply the signal vector plus a noise vector. So we consider the simplest form of the Kalman filter equations:

$$s[n] = s[\gamma n] + w[n]$$

$$x[n] = s[n] + v[n]$$

The MMSE estimate can be simplified in the form of recursive linear combination:

$$\hat{s}[n] = \hat{s}[\gamma n] + K[n](x[n] - \hat{s}[\gamma n])$$

where $K[n]$ is the Kalman gain matrix:

$$K[n] = M[n|\gamma n](C[n] + M[n|\gamma n])^{-1} \tag{4.5}$$

$M[n|\gamma n]$ is the covariance matrix of the estimation error, as in (4.4).

The optimal propagation weight $K[n]$ is given by (4.5) in terms of the covariance of estimation error and the covariance of noise. But these quantities are unknown or hard to determine. In our approach, we assume the innovation vector $w[n]$ and the measurement noise $v[n]$ are both zero-mean white Gaussian. Then both $M[n|\gamma n]$ and $C[n]$ will be diagonal matrices. We also assume the variances in x direction and y direction are same. Then the covariance matrix of the vectors can be simplified as variance of the orientation angles. The Kalman gain matrix $K[n]$ can be simplified as a scalar quantity $k[n]$ times the identity matrix, where $k[n]$ can be approximated as:

$$k[n] \approx \frac{\sigma_{\gamma n}^2}{\sigma_{\gamma n}^2 + \sigma_n^2}$$

where $\sigma_{\gamma n}^2$ and σ_n^2 are variances of orientation angles in the parent layer and current layer, respectively. Note that we obtained the same propagation weight in the last section (4.2), which is used to obtain the minimum variance estimation. As we did before, we can simplify the calculation by the *dominance measure* R, which can be obtained directly from the SVD estimate step. Our experiments show that this weight will give more robust estimates than Wilson's approach. We have:

$$\hat{s}[n] = \hat{s}[\gamma n] + \frac{R_n}{R_n + R_{\gamma n}}(x[n] - \hat{s}[\gamma n])$$

In these two sections, we discussed two multiscale estimation schemes, which start with similar objectives but lead to the same propagation weight (4.2). The propagation weight in (4.2) is directly related to the variance of the estimated orientation. In the areas with low SNR, $k[n]$ will give larger weight to the parent layer(coarser resolution), which is more reliable. While in the areas with high SNR, $k[n]$ will give larger weight to the current layer(finer resolution), which is more accurate. This property of $k[n]$ enable the multiscale approach to obtain an automatic balance between the estimate resolution and accuracy, which means in the final estimated orientation map, the algorithm will use larger estimate window in *noisy* area to ensure the estimate reliability and smaller estimate window in *clean* area to get more accurate estimate. We will show the experimental results in next chapter.

One problem of the multiscale approach is blocking artifacts. We will introduce an overlapped multiscale scheme in next section to solve this problem. It is shown that besides solving the blocking effect, overlapped multiscale method can also obtain more robust estimation results.

4.5 Overlapped Multiscale Model

The blocking effect exists in every block-based methods. The larger the block size, the more obvious the blocking effect. If the block size is a constant, the blocking effect can be eliminated by choosing appropriate size of the averaging kernel to smooth the resulting image. But as we discussed, in our case the resulting orientation map does

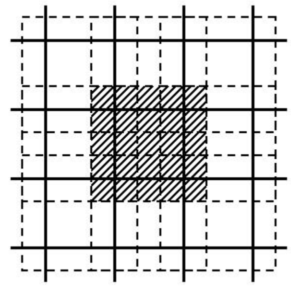


Figure 4.5: Overlapped blocks.

not have constant block size. In the areas with low SNR, the algorithm will tend to use larger blocks; while in high SNR areas, the algorithm will tend to use smaller blocks. The different block size makes the simple smoothing processing not appropriate. An efficient method to eliminate the blocking effect is using overlapped blocks, instead of separate non-overlapped blocks, as introduced in [14] and [27]. Recall that in the second step of our multiscale scheme, we need to first divide the current layer image into local blocks and do the PCA estimation within each block. In order to eliminate the blocking effect, we can divide the image into overlapped blocks, as shown in Figure 4.5.

Here the solid blocks are non-overlap blocks, dashed blocks are overlapped blocks and the shaded block is one sample of the overlapped blocks.

We do this overlap block division in each layer of the pyramid and use the estimation scheme we described before. We use the same block size and overlap size on every layer, but actually after being propagated to the finest resolution, the low resolution layer will have larger block size and also larger overlap size. That means, by using the multiscale approach, the overlap size is automatically adaptive to the block

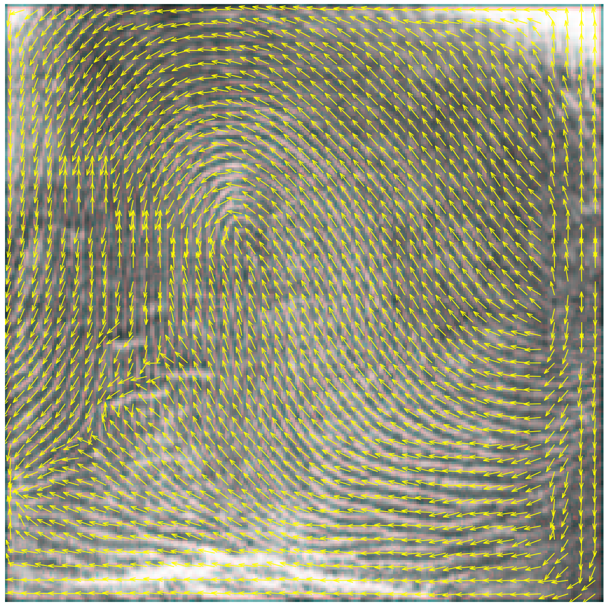


Figure 4.6: Overlapped multiscale estimation result.

size, which is desired to eliminate the blocking effect and also keep detail information. The estimation results are shown in Figure 4.6 and 4.7. It is shown that, comparing with non-overlap multiscale method, the overlapped multiscale estimation scheme will get more smooth results. Our further tests also show that the overlapped multiscale method is more robust to noise than the non-overlap method, as we will show in next chapter.

4.6 Conclusion

In this chapter we discussed multiscale estimation approaches. Compared with single scale estimation, the multiscale method is more robust to noise. By choosing the propagation weight as (4.2), our multiscale method will yield an automatic

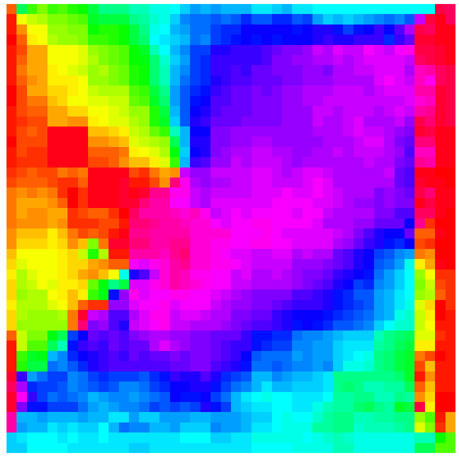


Figure 4.7: Overlapped multiscale estimation angle map.

balance between the estimate resolution and accuracy. We also used an overlap multiscale method to eliminate the blocking effect.

Wilson et al. [31] used similar multiscale approach to do orientation estimation. They also used the linear combination to propagate the estimates. They pointed out, as we described in this chapter, the optimum propagation weight is a function of the variance of estimation error and the variance of additive noise. Since these quantities are unknown or hard to determined, they used the rate of the average energy of gradient vectors as the propagation weight:

$$k[n] = \frac{\epsilon_n^2}{\epsilon_n^2 + \epsilon_{\gamma n}^2},$$

where ϵ_n^2 is the average value of the magnitude square of the gradient estimate [13]. Note that the vector energy has nothing to do with the vector orientation. So we think this quantity is not fit for the orientation estimation problem. In our approach, we use PCA-based estimation method in each scale, and use the by-product of the PCA step as propagation weight, which is directly related to the variance of orientation. The combination of the PCA and multiscale approach makes our method more robust and efficient than Wilson's method, as we will shown in next chapter.

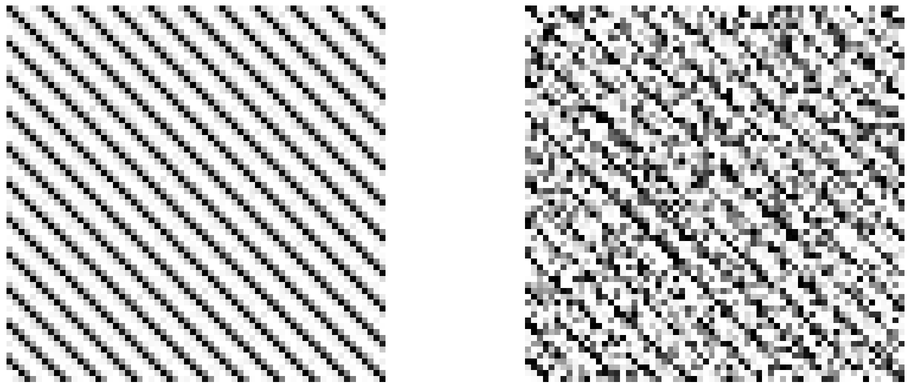
More comparisons between the single scale method and multiscale method, overlapped or non-overlapped, will be shown in the next chapter.

Chapter 5

Implementation and Results

In this chapter we will discuss some details of the implementation of our multiscale PCA-based orientation estimation approach and show some test results. We will start with the discussion of using image pyramid or gradient pyramid in the multiscale approach and compare the estimation results. Then we will introduce another propagation option in the third step of the multiscale approach and compare it with the linear combination propagation we used in last chapter. We will also introduce a preprocessing step which is needed to eliminate some *propagation errors* in the resulting orientation map.

In the second half of this chapter, we will give some test results using the proposed method. One important test is for noise robustness. As we claimed, the combination of PCA and multiscale scheme makes the proposed method to be very robust to additive noise. We will show this property in our tests. The test image is an image with a single known orientation pattern, as shown in Figure 5.1(a). We apply



(a) Clean test image (b) Noisy test image

Figure 5.1: The test image with single orientation.

additive Gaussian white noise on the test image, as shown in Figure 5.1(b), and perform orientation estimation on the noisy test image. Since the sought orientation is known, we can calculate the estimation error. Increasing the variance of the additive noise, doing Monte Carlo simulation (10000 times simulation), then we plot the change of the average estimation error with respect to the increase of the noise variance. The more robust the estimation method, the slower the average estimation error will increase. We will use this test to compare the noise robustness of the proposed method with some other approaches.

Another important property of the proposed method is the automatic adaptability of the estimation resolution and the estimation reliability, which means in the image area with low SNR, the algorithm will tend to use large estimate blocks to increase the reliability; while in the image area with high SNR, the algorithm will use smaller estimate blocks to get more localized estimates. We will show this property visually by an estimated orientation angle map.

Finally, we will show some estimate results on real and simulative images.

5.1 Image pyramid vs. Gradient pyramid

As we discussed in last chapter, in the first step of the multiscale approach, we can build up an image pyramid, then calculate the gradients on each layer of the image pyramid. Alternatively we can first calculate the gradients on the original image, then apply the low-pass-filter/subsample step on the gradient field to build up a gradient pyramid. From our first intuition, since both the gradient calculator we used and the low-pass-filter/subsample step in building the image/gradient pyramid are linear operators, these two options should yield the same results. But our test results show that they are not exactly. In order to see the difference, let us look more carefully at the process of building the image pyramid and the gradient pyramid. Two layers from the image/gradient pyramid are shown in Figure 5.2. Here we only discuss the gradient in the x-direction, the discussion for the gradient in y-direction is the same.

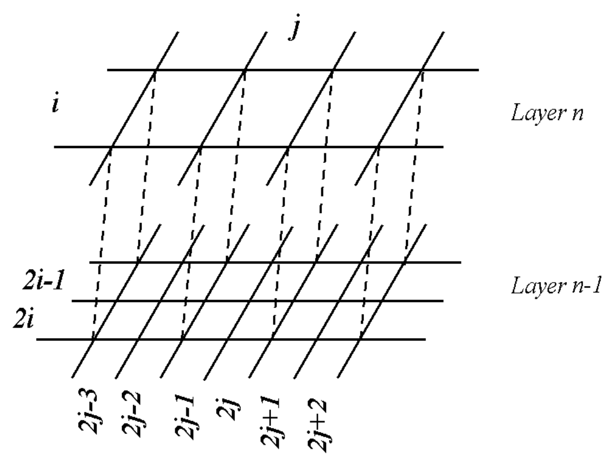


Figure 5.2: Two layers of the image/gradient pyramid

Let us first look at the image pyramid method. Denote the image pixel at the position (i,j) of layer n as $I_{n,(i,j)}$ and the gradient (in x-direction) at the position (i,j) of layer n as $g_{n,(i,j)}$. Using the MATLAB gradient function, $g_{n,(i,j)}$ is calculated as:

$$g_{n,(i,j)} = \frac{I_{n,(i,j+1)} - I_{n,(i,j-1)}}{2}.$$

In our approach, the image pyramid is built up with a 2×2 average kernel. Then each image pixel in the low resolution layer is composed by the average of its four *children* pixels in the high resolution layer:

$$I_{n,(i,j)} = \frac{\sum_{k=2i-1}^{2i} \sum_{l=2j-1}^{2j} I_{n-1,(k,l)}}{4}.$$

Then $g_{n,(i,j)}$ can be written as:

$$\begin{aligned} g_{n,(i,j)} &= \frac{\frac{1}{4}\sum_{k=2i-1}^{2i} \sum_{l=2j+1}^{2j+2} I_{n-1,(k,l)} - \frac{1}{4}\sum_{k=2i-1}^{2i} \sum_{l=2j-3}^{2j-2} I_{n-1,(k,l)}}{2} \\ &= \frac{1}{4} \sum_{k=2i-1}^{2i} \sum_{l=2j+1}^{2j+2} \frac{I_{n-1,(k,l)} - I_{n-1,(k,l-4)}}{2}. \end{aligned}$$

Notice here:

$$\begin{aligned} \frac{I_{n-1,(k,l)} - I_{n-1,(k,l-4)}}{2} &= \frac{I_{n-1,(k,l)} - I_{n-1,(k,l-2)}}{2} + \frac{I_{n-1,(k,l-2)} - I_{n-1,(k,l-4)}}{2} \\ &= g_{n-1,(k,l-1)} + g_{n-1,(k,l-3)}. \end{aligned}$$

Then we can simplify $g_{n,(i,j)}$ as:

$$g_{n,(i,j)} = \frac{1}{4} \sum_{k=2i-1}^{2i} \sum_{l=2j-2}^{2j+1} g_{n-1,(k,l)}.$$

This can be shown in Figure 5.3. The gradient on the low resolution layer, marked with shade, is composed from the sum of the gradients in the shaded area of the high resolution layer, divided by four.

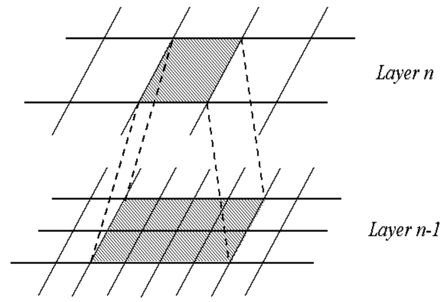


Figure 5.3: Estimation propagation for the image pyramid method.

This is for the case of image pyramid. For the gradient pyramid method, the low resolution gradient is simply the average of the four gradients in the high resolution parent layer:

$$g_{n,(i,j)} = \frac{1}{4} \sum_{k=2j-1}^{2j} \sum_{l=2i-1}^{2i} g_{n-1,(k,l)},$$

as shown in Figure 5.4.

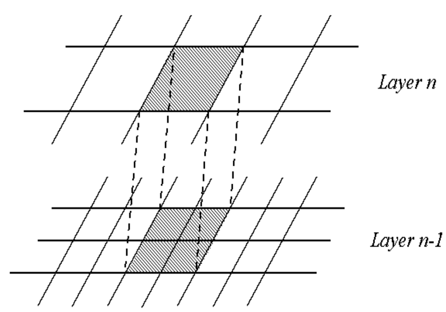


Figure 5.4: Estimation propagation for the gradient pyramid method.

From Figure 5.3 and 5.4 we know, if we use the MATLAB gradient function, in the image pyramid method, the low resolution gradients are composed with more neighboring high resolution gradients than in the gradient pyramid method. Since the average process can depress the noise effect, we have reason to expect, in our case, the image pyramid method should have better noise robustness than the gradient pyramid method. Our test results (using the robustness test and test images which are introduced in the beginning of this chapter; using error bars in the curve to represent the variance of the quantity) prove this prediction, as shown in Figure 5.5. Please note that this is only applicable for the MATLAB gradient operator. If we choose to use some other gradient operator, the image pyramid method is NOT always better than the gradient pyramid method.

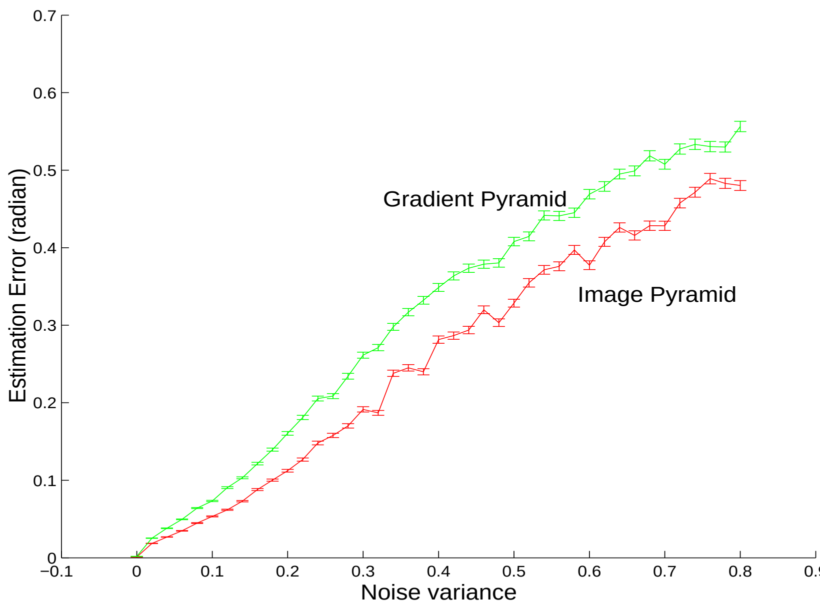


Figure 5.5: Image pyramid method VS. gradient pyramid method.

From Figure 5.5 we can see, with the increase of noise variance, the image pyramid method works a little more robustly than the gradient pyramid. The problem of the image pyramid method is the efficiency. It needs to calculate the gradients on each layer of the pyramid, while in the gradient pyramid method, we only need to calculate the gradients once.

5.2 Propagation: selection vs. weighted average

As we described in the last chapter, in the third step of the multiscale approach, we use the rate of the *dominance measure* R, (4.3), as the combination weight, to propagate the estimates from low resolution layers to high resolution layers with a linear combination. The problem of linear combination is if one of the estimates is error estimate, this error will be propagated down to the final result. Here we introduce another way to propagate the estimates, also using the *dominance measure* R. Instead of using R as combination weight, we can use it as a selection criterion. When propagate the low resolution estimates $\theta_{\gamma n}$ to high resolution estimates θ_n, within each block, we compare the dominance measure R, and choose the estimate with larger R:

$$\hat{\theta}_n = \begin{cases} \theta_{\gamma n} & \text{if } R_{\gamma n} > R_n \\ \theta_n & \text{if } R_{\gamma n} < R_n \end{cases}$$

where $R_{\gamma n}$ is the dominance measure in the low resolution layer,R_n is the dominance measure in the high resolution later and $\hat{\theta}_n$ is the resulting estimation. We compare the noise robustness of these two propagation options in Figure 5.6, using the test image in Figure 5.1.

From Figure 5.6 we can see, the selection propagation method is not as robust as the linear combination method. But since this method will prevent the error estimates from propagating down and also will need less computation than the combination method. Still, it is a viable alternate for the propagation of the estimates.

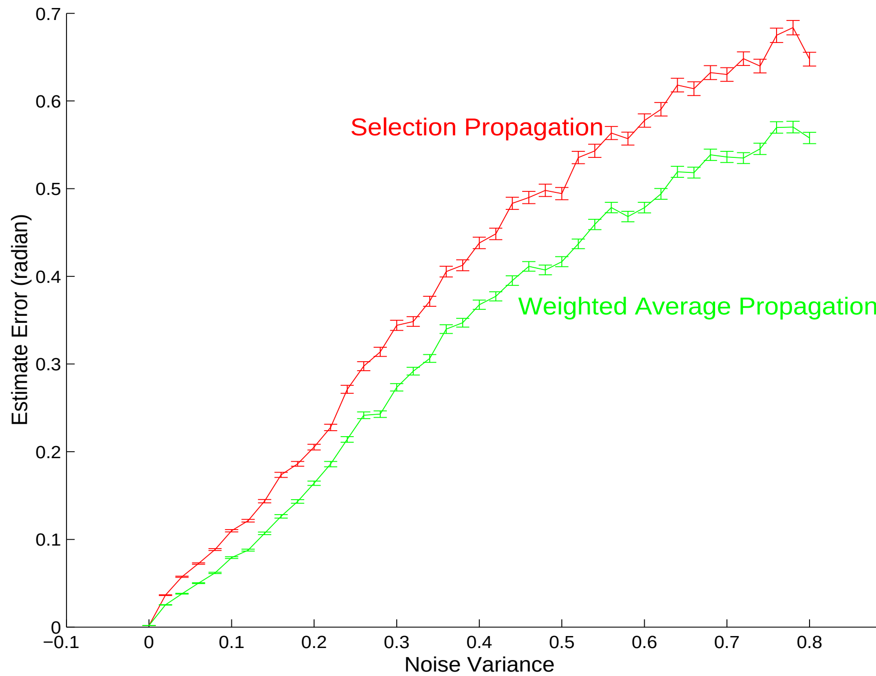


Figure 5.6: Selection propagation method VS. weighted average propagation method.

5.3 Eliminating propagation errors

As we discussed in previous chapters, the multiscale approach will yield robust and accurate orientation estimates. But in some of our tests, we find that when using the linear combination propagation, there are some unexpected errors. The resulting orientation estimation is about orthogonal to the correct direction, as shown in Figure 5.7.

The reason for these errors is that we limit the orientation to the range of $[-\frac{\pi}{2}, \frac{\pi}{2})$. As we discussed in Chapter 2, in the image orientation estimation problem, the orientation of an *undirected* line is of interest, which means the unit vectors $\mathbf{x}$ and $\mathbf{-x}$ are indistinguishable. The propagation errors may occur in the image area where

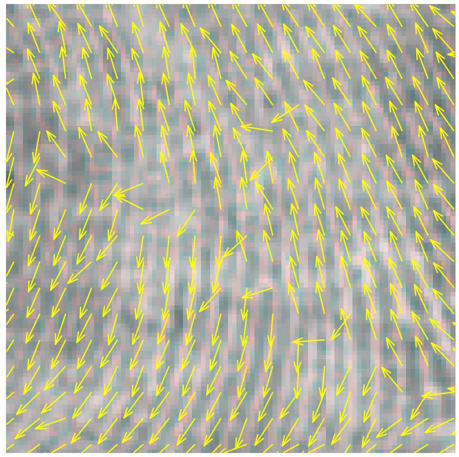


Figure 5.7: Estimation propagation errors.

the pattern orientation is close to $\frac{\pi}{2}$ (or -$\frac{\pi}{2}$), as in shown Figure 5.7. The reason is, on one pyramid layer, the estimate may be close to $\frac{\pi}{2}$, while on the other layer, the estimate may be close to -$\frac{\pi}{2}$. Then after we combine these two estimates by linear combination, the result will be roughly orthogonal to both of these two estimates, as shown in Figure 5.8.

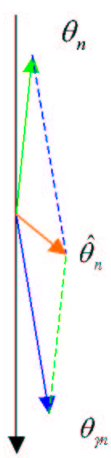


Figure 5.8: Reason for the propagation error.

The solution of this kind of propagation errors is to pre-process the high res-

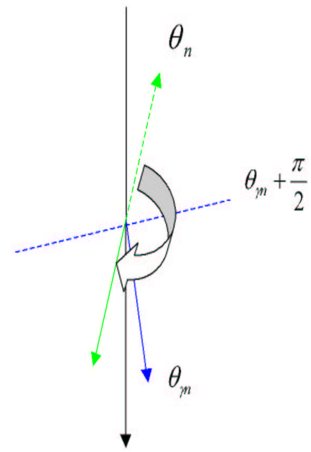


Figure 5.9: Pre-process the orientation(1)

olution estimate, with respect to the low resolution estimate, before the propagation. If $\theta_n > \theta_{\gamma n} + \frac{\pi}{2}$, then $\theta_n = \theta_n - \pi$, as shown in Figure 5.9. If $\theta_n < \theta_{\gamma n} - \frac{\pi}{2}$, then $\theta_n = \theta_n + \pi$, as shown in Figure 5.10.

Note that this processing only affects the *error* propagation. After this processing, the error estimates in Figure 5.7 will be corrected, as shown in Figure 5.11.

5.4 SVD vs. Mean

Starting with this section, we will show some more test results. First we will compare the SVD-based method with a simple averaging method. As we discussed in Chapter 2, for Von-Mises distribution, the maximum likelihood estimate of the dominant direction is the mean of the sample unit vectors. Although this does not fit the image orientation model we have considered (where the *orientation*, but not *direction*, is of interest), it does show that the averaging method may be a simple way to estimate the orientation. We compare the averaging method with our SVD-based

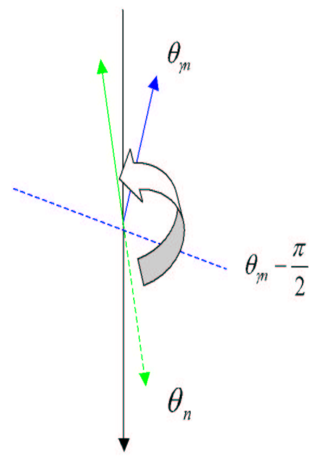


Figure 5.10: Pre-process the orientation(2)

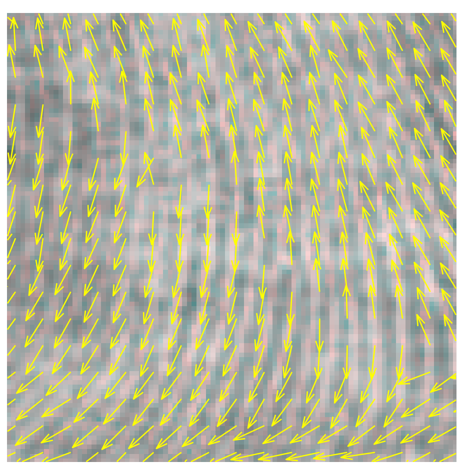


Figure 5.11: Propagation errors are corrected.

method, within a single block of the test image in Figure 5.1. The result, Figure 5.12, shows that the SVD-based method works much more robustly than the averaging method.

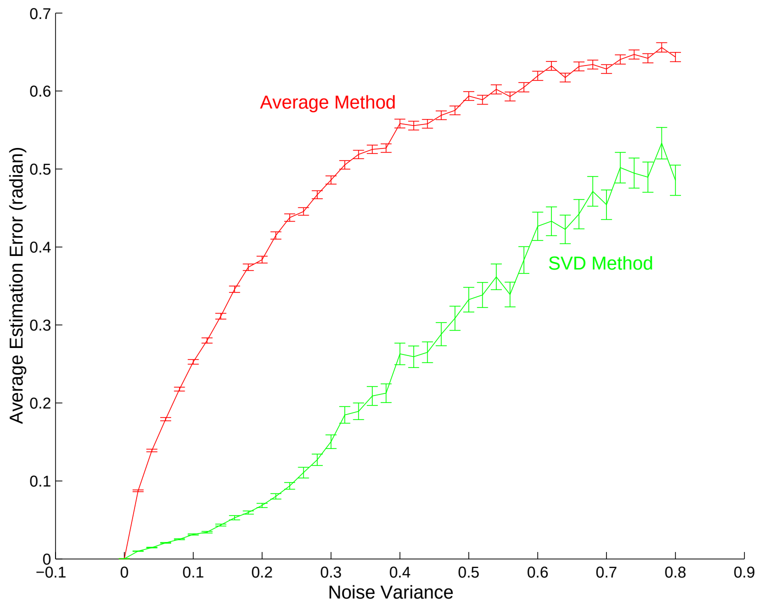


Figure 5.12: Single-scale average method VS Single-scale PCA method.

5.5 Multiscale vs. Single-scale

As we showed in the last chapter, the multiscale estimation approach works much better than the single scale approach in the presence of additive noise. Here we will show the robustness test results of the proposed multiscale PCA method, compared with the single scale PCA method. We also compare with Wilson's method [31]. The results are shown in Figure 5.13, using the test image in Figure 5.1. The block size is 8. For overlapped methods, the overlap size is 2.

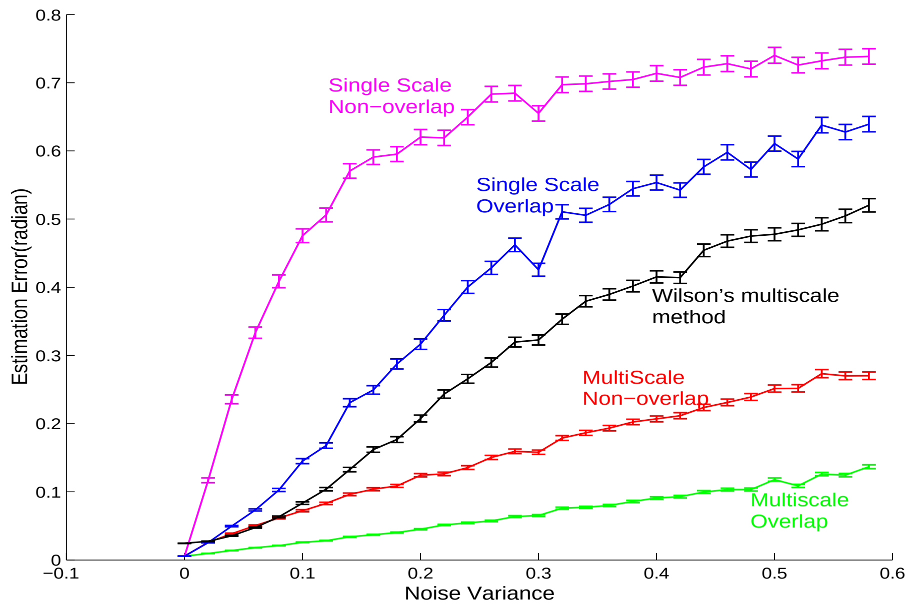


Figure 5.13: Multiscale PCA method VS Single-scale PCA method.

From Figure 5.13 we can see that, with increasing noise variance, the multiscale methods are much more robust than the single scale methods. Wilson's method works better than the single scale PCA methods, but worse than the multiscale PCA methods. We can also see that in both single-scale and multiscale methods, the overlapped block scheme is more robust than the non-overlapped scheme.

Notice that with increasing noise variance, the average estimation error will converge to some value, instead of going to infinity, as we expect from our first intuition. The reason for this convergence is, from the directional statistics (Chapter 2 in this thesis), for a pure Gaussian white noise vector field, the distribution of the orientation angle is a uniform distribution in $(-\frac{\pi}{2}, \frac{\pi}{2}]$, with the mean equal to 0. In our test, the sought orientation is $\frac{\pi}{4}$, so with the increase of the additive noise, the average estimation error will converge to $\frac{\pi}{4} - 0 = \frac{\pi}{4}$, as shown in Figure 5.13.

5.6 Adaptability Between Resolution and Reliability

One important property of the proposed multiscale PCA estimation method is the automatic adaptability of the estimation resolution and accuracy. We will show this property with the test image in Figure 5.14.

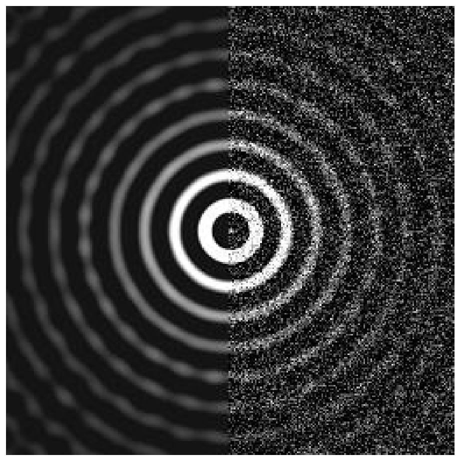


Figure 5.14: Test image for the adaptability.

The test image has diverse orientations with different contrast. Gaussian white noise is applied in the right half. From the estimated orientation map we can see (Figure 5.15 and Figure 5.16), the single scale method is quite messy in the noisy half, while the multiscale method works well on both sides.

Figure 5.17 shows the estimated orientation vector angles using these two methods, we can see that the multiscale method uses small blocks in the clean half of the image to get high resolution estimates, and uses large blocks in the noisy half to yield more reliable, but coarser, estimates.

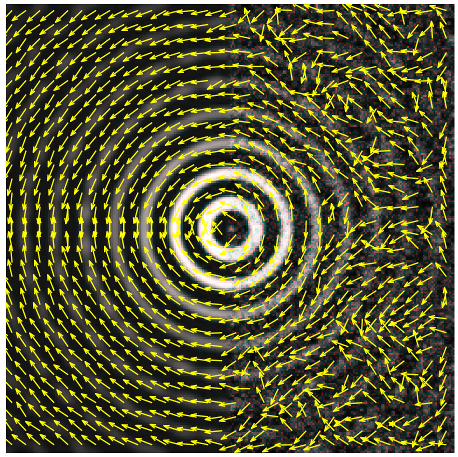


Figure 5.15: Single scale estimate result of Figure 5.14

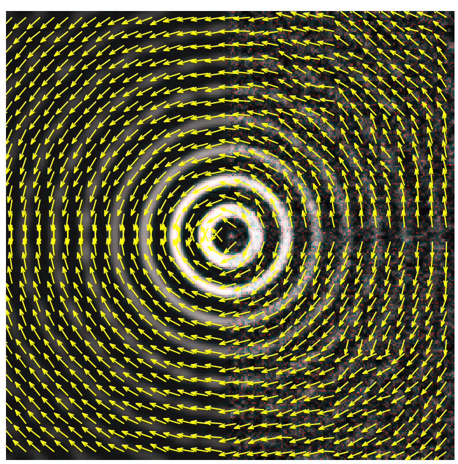


Figure 5.16: Multiscale estimate result of Figure 5.14

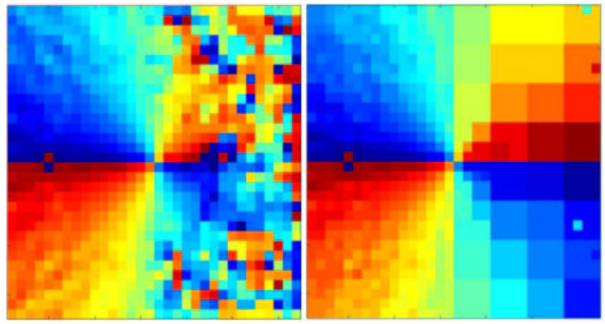


Figure 5.17: Orientation angle map of single-scale and multiscale methods.

5.7 Other examples

As an example of the performance of the proposed method on a real image, the orientation estimation result for a real fingerprint image (Figure 5.18) is shown in Figure 5.19 and Figure 5.20.

Figure 5.18: Another fingerprint test image

Finally, as an application of the image orientation estimation, we use it to

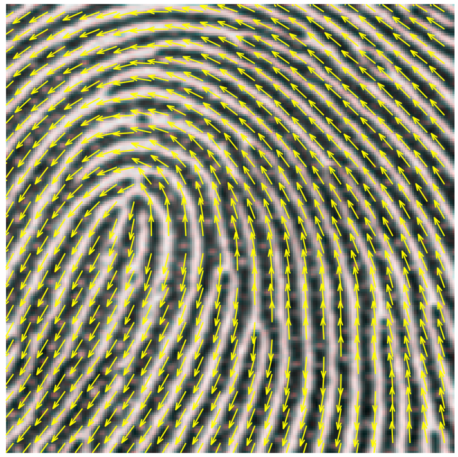


Figure 5.19: Orientation map of the fingerprint image

do image segmentation. The test image, Figure 5.21, has textures with different orientation in the middle. After applying additive noise on the test image, it is hard to visually tell the difference. But from the estimated orientation map (using multiscale method, block size 8, overlap 4), it is easy to segment the test image into different texture patches, as shown in Figure 5.22.

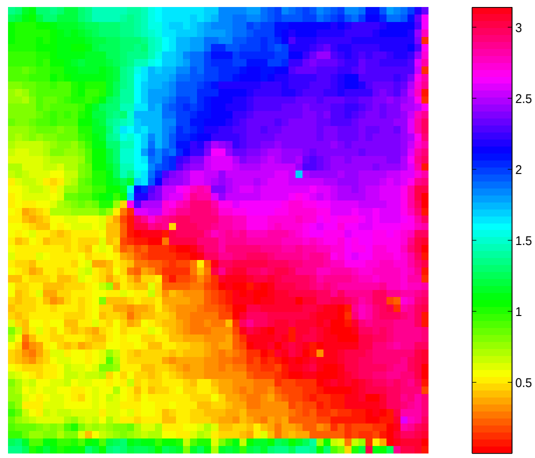


Figure 5.20: Orientation angle map of the fingerprint image

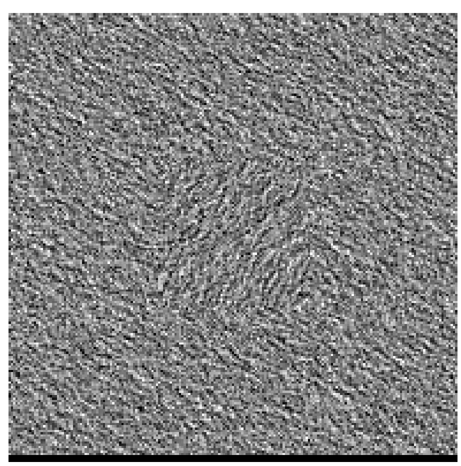


Figure 5.21: Test image with different textures.

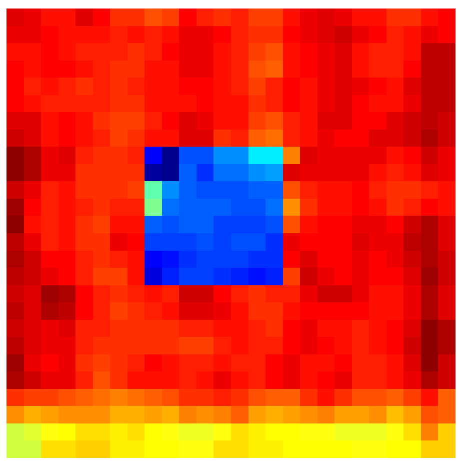


Figure 5.22: Orientation angle map of the texture image

Chapter 6

Conclusions and Future Directions

This chapter is the conclusion of the thesis. We will briefly introduce a further application of the proposed multiscale PCA-based estimation method and followed with a summary of the thesis contribution.

6.1 Further Applications

As we discussed in previous chapters, the proposed multiscale PCA estimation method will yield robust and accurate estimation of the image orientation. Actually, this multiscale PCA-based method can also be developed for other applications, e.g. corner detector.

The PCA method is used in the Harris corner detector [12]. For each pixel in the image, first calculate the gradients in its neighboring area, then calculate the

SVD of the gradients matrix (2.9). If the second singular value s_2 is larger than a threshold, this pixel is likely to be a corner [12]. It is easy to see that the Harris corner detection method is very similar to the PCA-based orientation estimation method. The difference is, for orientation estimation, the normalized difference of the singular values, R in (3.2), is used as the orientation dominance measure; while in the Harris corner detector, the second singular value s_2 is used as an indicator of corners. Actually, only the value of s_2 is not enough to detect the corners correctly. Since in the image area with sharp edge, both s_1 and s_2 may be larger than the threshold, but $s_1 >> s_2$. So in our simulation we use

$$s_2(1 - \frac{s_1 - s_2}{s_1 + s_2}) = s_2(1 - R)$$

as the corner indicator, which requires the value of s_2 to be large, and s_1 need to be close to s_2.

The problem of the Harris corner detector is, if we want to get accurate corner location, we need to limit the size of the calculation window. By doing that, some smooth corners can not be detected. We show this with the test image of Figure 6.1.

If we directly use the Harris corner detector on the image, only those sharp corners can be detected, as shown in Figure 6.2. If we build up an image pyramid, do corner detector on each layer of the pyramid, some larger scale corners can be detected in the low resolution layers. As shown in Figure 6.3, the corner detection on the third layer of the image pyramid will detect the smooth corner.

Combine the detections from different resolution layers, we will get a corner

Figure 6.1: Corner detection test image

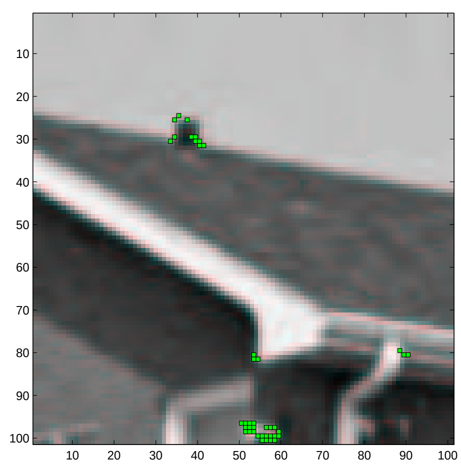


Figure 6.2: Corner detection on high resolution image layer

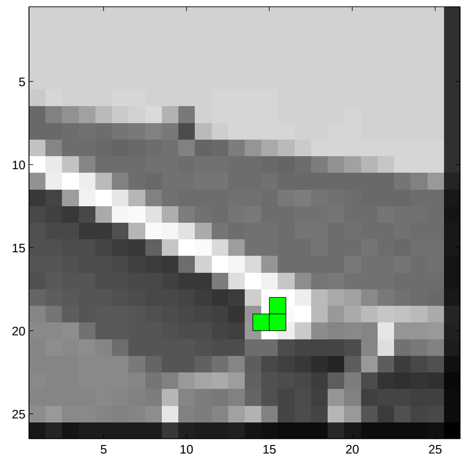


Figure 6.3: Corner detection on low resolution image layer

detection result with different scale corners.

6.2 Conclusions

In this thesis we presented a new approach to the image local orientation estimation problem. The proposed orientation estimation method works well in terms of both robustness and accuracy.

In Chapter 2 we discussed the underlying theory of the orientation estimation problem: directional statistics. As far as we know, in the literature dealing with image orientation, there is rarely an explicit reference to the field of directional statistics. From the theory of the directional statistics, for a Bingham distribute orientation vector field, which can be obtained from a zero-mean normally distributed gradient field, the optimum (maximum likelihood) estimation of the dominance orientation is the singular vector of the gradient matrix (2.9).

We discussed more details about the SVD-based estimation method in Chapter 3. It is well known that SVD can be used to analyze the *oriented energy* of the given data matrix [7]. We apply this to the image orientation estimation problem. We proposed a dominance measure R, (3.2), and discussed the statistical behavior of R. By obtaining the distribution function of R for a zero-mean *i.i.d* Gaussian matrix, we can do a significance test to eliminate the error estimates from the resulting orientation map.

In Chapter 4 we introduced a multiscale scheme to further improve the robustness and adaptability of the PCA-base estimation method. We discussed the multiscale scheme from two start points, minimum variance estimation and multiscale vector Kalman filter, and reach a same estimation propagation method. By using the dominance measure R, introduced in Chapter 3, our multiscale method can yield robust estimates and the automatic adaptability between the estimation resolution and the estimation reliability. We also proposed an overlap multiscale model to eliminate the blocking effect and get even more robust estimates.

In Chapter 5 we discussed several problems and details in the implementation of the proposed method. We discussed the different options in the method and compare the estimated results. From the results we can see, the multiscale method is much more robust than single scale method. The automatic adaptability between the estimation resolution and the estimation reliability is also shown visually in the resulting orientation map.

We also presented another application of the multiscale PCA method in image corner detection, which can detect corners at different scales but still needs further work in the implementation details.

Bibliography

[1] E.H. Adelson, C.H. Anderson, J.R. Bergen, P.J. Burt, , and J.M. Ogden, *Pyramid methods in image processing*, RCA Engineer **29(6)** (1984), 33–41.

[2] J. Bigun, G. H. Granlund, and J. Wiklund, *Multidimensional orientation estimation with applications to texture analysis and optical flow*, IEEE Transactions on Pattern Analysis and Machine Intelligence **13(8)** (1991), 775–790.

[3] P. J. Burt and E. H. Adelson, *The laplacian pyramid as a compact image code*, IEEE Trans. Comput. **COM-31** (1983), 532–540.

[4] J. Canny, *A computational approach to edge detection*, IEEE PAMI **8(6)** (1986), 255–274.

[5] K. C. Chou, A. S. Willsky, and R. Nikoukhah, *Multiscale systems, kalman filters and riccati equations*, IEEE Transactions on Automatic Control **39** (1994).

[6] J. P. Da Costa, F. Le Pouliquen, C. Germain, and P. Baylou, *New operators for optimized orientation estimation*, International Conference on Image Processing (2001).

[7] Ed. F. Deprettere, SVD *and signal processing: Algorithms, applications and architectures*, North Holland, 1988.

[8] A. W. Drake, *Fundamentals of applied probability theory*, McGraw-Hill, 1976.

[9] A. Edelman, *Eigenvalues and condition numbers of random matrices*, SIAM Journal on Matrix Analysis and Applications (1988), 543–560.

[10] X. G. Feng and P. Milanfar, *Multiscale principal components analysis for image local orientation estimation*, The 36th Asilomar Conference on Signals, Systems and Computers (2002).

[11] L. Haglund and D. J. Fleet, *Stable estimation of image orientation*, Proceedings of the First IEEE International Conference on Image Processing **III** (1994), 68–72.

[12] C. G. Harris and M. Stephens, *A combined corner and edge detector*, In 4th Alvey Vision Conference (1988), 147–151.

[13] Tao-I. Hsu, J. L. Kuo, and R. Wilson, *A multiresolution texture gradient method for unsupervised segmentation*, Pattern Recognition (2000), 1819–1833.

[14] W. W. Irving, P. W. Fieguth, and A. S. Willsky, *An overlapping tree approach to multiscale stochastic modeling and estimation*, IEEE Transactions of Image Processing **6** (1997), 1517–1529.

[15] B. Jahne, *Digital image processing*, Springer, 1997.

[16] S. M. Kay, *Fundamentals of statistical signal processing* I, Prentice Hall PTR, 1993.

[17] E. P. Lyvers and O. R. Mitchell, *Precision edge contrast and orientation estimation*, IEEE Transactions on Pattern Analysis and Machine Intelligence **10(6)** (1988), 927–937.

[18] K. V. Mardia, *Statistics of directional data*, Academic Press, 1972.

[19] K. V. Mardia and P. E. Jupp, *Directional statistics*, Wiley, 2000.

[20] K. V. Mardia, J. T. Kent, and J. M. Bibby, *Multivariate analysis*, Academic Press, 1979.

[21] R. Mester, *Orientation estimation: Conventional techniques and a new non-differential approach*, European Signal Processing Conference (2000).

[22] R. Mester and M. Muhlich, *Improving motion and orientation estimation using an equilibrated total least squares approach*, Proceedings IEEE International Conference on Image Processing (2001), 640–643.

[23] M. Moonen and B. De Moor, SVD *and signal processing* III*: Algorithms, applications and architectures*, Elsevier, 1995.

[24] P. Perona, *Orientation diffusions*, IEEE Transactions on Image Processing **7(3)** (1998), 457–467.

[25] A. R. Rao and G. L. Lohse, *Identifying high level features of texture perception*, CVGIP: Graphical Models and Image Processing **55(3)** (1993), 218–233.

[26] A. Rosenfeld and M. Thurston, *Edge and curve detection for visual scene analysis*, IEEE Trans. Comput. **5** (1971), 562–569.

[27] B. Simon and B. Macq, *New overlapped block reconstruction for tree-structured decomposition of images*, Proceedings ICIP-94 **3** (1994), 586–590.

[28] M. Spann and R. G. Wilson, *A quad-tree approach to image segmentation which combines statistical and spatial information*, Patt. Recogn. **18** (1985), 257–269.

[29] R. Vaccaro, SVD *and signal processing* II*: Algorithms, applications and architectures*, Elsevier, 1991.

[30] G. S. Watson, *Statistics on spheres*, Wiley Interscience, 1983.

[31] R. Wilson, S. Clippingdale, and A. H. Bhalerao, *Robust estimation of local orientations in images using a multiresolution approach*, SPIE Visual Communications and Image Processing **1360** (1990), 1393–1403.

[32] X. Zhou, J. P. Baird, and J. F. Arnold, *Fringe-orientation estimation by use of a gaussian gradient filter and neighboring-direction averaging*, Applied Optics **38(5)** (1999), 795–804.