CHAPTER III
NOISE ANALYSIS AND FILTERING

3.1 Introduction

Fourier Reconstruction Methods have a speed advantage over statistical methods, but the images reconstructed by Fourier methods are characterized by a high degree of noise. To improve the quality of the reconstructed images, it becomes necessary to filter raysum images before the reconstruction process. In this chapter, we first model and analyze PET noise occurring in Fourier methods. Then, we study several conventional filtering schemes, including Wiener filter, square root filter and an adaptive filter.

In the statistical model, the raysum image is degraded by noise. The objective of image restoration is to reduce or eliminate the noise in the image. Simple image enhancement algorithms such as lowpass filtering methods can reduce the noise in some degree. To design a more effective restoration scheme, we need to exploit the characteristics of the image and noise and design filtering methods based the characteristics. The square root filter is an attempt to do so. To compare the effectiveness for different restoration schemes, we need a criterion and usually the minimum mean square error criterion is employed. But as we will see in this chapter, the minimum mean square (MS) filter is generally difficult to design and therefore we discuss the MS filter under some constraints. Wiener filter and the adaptive filter are two examples.

3.2 Model of PET Noise

An additive noise model is given by

$$G(i,j) = F(i,j) + N(i,j)$$

where G(i,j), F(i,j) and N(i,j) denote observed image, original image and random noise respectively. If N(i,j) is not a function of F(i,j), the noise is called additive signal-independent noise. If N(i,j) is a function of F(i,j), the noise is called additive signal-dependent noise. Note that that "signal-independence" is not related to statistical independence. For, if F is a constant c then N and F are always statistically independent. But the noise

$$N(i,j) = \alpha c(i,j)^{1/3} u(i,j),$$

where α is a constant and u(i,j) is a standard Gaussian random variable (N(i,j) is called film-grain noise in literature), is signal-dependent by the above definition. On the other hand, if F is a non-constant random variable, the fact that N is not a function of F does not imply that F and N are statistically independent. "signal-dependence" describes whether or not the level (or variance) of a noise is related to the level of a signal.

If an object PET image is modeled as a Poisson model, then the number of photons detected in detectors $\{G(i,j), i,j = 1,2,...,N_d\}$ constitute independent Poisson random variables with each mean $\lambda(i,j)$. We may write it in the form of

$$G(i,j) = \lambda(i,j) + N(i,j)$$

where $N(i,j) = G(i,j) - \lambda(i,j)$ denotes noise. Since N(i,j) has mean zero and variance $\lambda(i,j)$, we can write N(i,j) = $\sqrt{\lambda(i,j)}\epsilon(i,j)$ where $\epsilon(i,j)$ is a white noise with mean zero and unit variance. Here we see that N(i,j) is a function of $\lambda(i,j)$. Therefore, the PET raysum noise is a signal-dependent additive noise.

If an object PET image is modeled as a Cox process, then for given mean measure $\Lambda(i,j)$, then the number of photons detected in detectors $\{G(i,j), i,j = 1,2,...,N_d\}$ constitute independent Poisson random variables with each mean $\lambda(i,j)$, *i.e.*,

$$P(G(i_s,j_s) = g(i_s,j_s), s = 1,2,...,k | \Lambda(i_s,j_s) = \lambda(i_s,j_s), s = 1,2,...,k)$$

$$= \prod_{s=1}^{k} e^{-\lambda(i_s,j_s)} \frac{\lambda(i_s,j_s)^{g(i_s,j_s)}}{g(i,j)!}.$$

Since G(i,j) can also be written in the form of

$$G(i,j) = \Lambda(i,j) + N(i,j)$$

where $\Lambda(i,j)$ denotes the random mean measure of G(i,j) and $N(i,j) = G(i,j) - \Lambda(i,j)$ denotes noise, the PET raysum noise is also a signal-dependent additive noise.

From the discussion above, we conclude that the PET raysum noise is a signal-dependent additive noise for either Poisson model or Cox model. In the remaining of this chapter, we discuss several conventional noise filtering schemes for this additive noise model.

3.3 Wiener Filter

Wiener filtering is one of the first methods developed to reduce additive noise. It was first considered for image restoration in the 1960s by Helstrom *et al* [HELS67] and has influenced the development of many other image restoration methods. In principle, the minimum mean square error estimator of an unknown image is the conditional expectation given the observation. This will be formally stated in Theorem 3.3.1. The conditional expectation is generally difficult to obtain if we do not make any assumption about the unknown image and the observation. For a linear and shift-invariant (LSI) system and under the stationarity constraint, however, the minimum mean square error estimator can be obtained by using a Wiener filter, which can be explicitly expressed and easily implemented. This will be addressed in Theorem 3.3.2.

Consider an additive noise model given by

$$G(i,j) = F(i,j) + N(i,j). \tag{3.3.1}$$

where F(i,j) is the original image, G(i,j) is the noisy observation and N(i,j) is a zero mean noise that can be signal-independent or signal-dependent. The estimator $\hat{F}(i,j)$ is said to be based on $G = \{G(m,n), \text{ all } (m,n)\}$ if $\hat{F}(i,j)$ is a functional of some entries of G. Then, the problem of the minimum mean square error estimation of F(i,j) based on $G = \{G(m,n) \text{ all } (m,n)\}$ can be solved by the following theorem, which is known as the fundamental theorem of estimation theory.

Theorem 3.3.1: Assume $E|F(i,j)|^2 < \infty$ and $E|G(i,j)|^2 < \infty$ for all i, j. Then, for any pixel (i,j), the conditional expectation $E(F(i,j)|G)$ minimizes the mean square error $E(F(i,j) - \hat{F}(i,j))^2$ for all estimators $\hat{F}(i,j)$ based the observation G.

Remark. This theorem implies that the conditional expectation $E(F|G)$ minimizes the mean square error

$$E(F-\hat{F})(F-\hat{F})^T = \sum_i \sum_j E(F(i,j) - \hat{F}(i,j))^2$$

for all estimators $\hat{F}$ based the observation G.

Proof. By completing square, we have

$$E(F(i,j) - \hat{F}(i,j))^2$$

$$= E(F(i,j) - E(F(i,j)|G))^2 + E(E(F(i,j)|G) - \hat{F}(i,j))^2$$

$$+2E(F(i,j) - E(F(i,j)|G))(E(F|G) - \hat{F}(i,j)).$$

Since both $E(F(i,j)|G)$ and $\hat{F}(i,j)$ are based on G, $Z(i,j) = E(F(i,j)|G) - \hat{F}(i,j)$ is also based on G. It follows that

$$E(F(i,j)Z(i,j)) = E[E(F(i,j)Z(i,j)|G)] = E[Z(i,j)E(F(i,j)|G)].$$

Therefore,

$$E[F(i,j) - E(F(i,j)|G)](E[F(i,j)|G] - \hat{F}(i,j)) = 0$$

and then

$$E(F(i,j)-\hat{F}(i,j))^2 = E[F(i,j)-E(F(i,j)|G)]^2 + E[E(F(i,j)|G)-\hat{F}(i,j)]^2.$$

Here we see that $E(F(i,j)-\hat{F}(i,j))^2$ is minimized if and only if

$$\hat{F}(i,j) = E(F(i,j)|G).$$

QED

Since the unknown F(i,j), which is to be estimated, is in the conditional expectation $E(F(i,j)|G)$, which minimizes the mean square error $E(F(i,j)-\hat{F}(i,j))^2$, the minimum mean square error is generally difficult to obtain if we do not know anything about F(i,j) or G. Wiener filter gives an explicit form of $E(F(i,j)|G)$ by imposing some assumptions on F(i,j) and G(i,j). Next, we derive Wiener filter.

An input-output system T associated with input X(i,j) and output Y(i,j) is defined by

$$Y(i,j) = T[X(i,j)].$$

T is said to be linear if

$$T[aX_1(i,j)+bX_2(i,j)] = aT[X_1(i,j)] + bT[X_2(i,j)].$$

T is said to be shift-invariant if

$$T[X(i-k,j-l)] = Y(i-k,j-l).$$

From signal processing theory, we know that a linear and shift-invariant (LSI) system with input X(i,j) and output Y(i,j) can be expressed as

$$Y(i,j) = h \star X(i,j) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} h(i-m,j-n)X(m,n)$$

where h(i,j) is referred to as an impulse response or a filter and $\star$ is referred to as the linear convolution operator.

As discussed in Chapter II, an image can be modeled as a two-dimensional random process. A two-dimensional random process X(i,j) is said to be stationary (in wide sense) if the mean of X(i,j) is a constant and if the covariance between any two pixels is only related the distance between them. Symbolically, X(i,j) is said to be stationary if:

(i) $EX(i,j) = m_X$ for all (i,j);

(ii) $EX(i,j)X^*(i+m,j+n) = R_X(m,n)$ is independent of (i,j) for all i, j, m, n.

Now we go back to the additive noise model as (3.3.1). Let us assume that F(i,j) and N(i,j) are zero-mean wide sense stationary random processes with known power spectra $P_F(u,v)$ and $P_N(u,v)$ given by

$$P_F(u,v) = \sum_{m=-\infty}^{+\infty} \sum_{n=-\infty}^{+\infty} R_F(m,n) e^{-imu-inv}$$

$$P_N(u,v) = \sum_{m=-\infty}^{+\infty} \sum_{n=-\infty}^{+\infty} R_N(m,n) e^{-imu-inv}$$

where $R_F(m,n)$ and $R_N(m,n)$ are correlation functions of F and N respectively. For the LSI system

$$Y(i,j) = h \star X(i,j)$$

The frequency response of the filter h(i,j) is defined by

$$H(u,v) = \sum_{m=-\infty}^{+\infty} \sum_{n=-\infty}^{+\infty} h(m,n) e^{-imu-inv}.$$

A Wiener filter is a filter h(i,j) that has frequency response $H(u,v)$ given by

$$H(u,v) = \frac{P_F(u,v)}{P_F(u,v) + P_N(u,v)}. \tag{3.3.2}$$

For the Wiener filter, we have the following theorem:

Theorem 3.3.2: Let F(i,j) and N(i,j) in the additive model (3.3.1) are zero-mean stationary random processes. Suppose that N(i,j) and F(i,j) are uncorrelated. Let h(i,j) be the Wiener filter whose frequency response is given by (3.3.2). Then $\hat{F}(i,j) = h \star G(i,j)$ is the linear and shift-invariant minimum mean square error (LSMS) estimator of F(i,j).

Remark. The LSMS estimator $\hat{F}(i,j)$ in Theorem 3.3.2 minimizes the mean square error among all estimators with the form of linear convolution $\hat{F}(i,j) = h \star G(i,j)$. The LSMS estimator is an estimator whose impulse response h(i,j) has frequency response as (3.3.2); while the minimum mean square error estimator $\hat{F}(i,j)$ in Theorem 3.3.1 minimizes the mean square error among all estimators which are functionals of some entries of $\{G(i,j), \text{ all } i,j)\}$ (The functional does not necessarily possess a form of $\hat{F}(i,j) = h \star G(i,j)$.) The second difference is that Theorem 3.3.2 assumes that the images are modeled as a stationary random process but Theorem 3.3.1 does not make such assumption.

Proof. Let e(i,j)=$\hat{F}$(i,j)-F(i,j), then the LSMS estimator of F(i,j) is an estimator that can be expressed as the form of $\hat{F}(i,j) = h \star G(i,j)$ and that minimizes $E|e(i,j)|^2$. We want to show that the frequency response of h(i,j) is given by (3.3.2).

From estimation theory, the linear minimum mean square error estimator $\hat{F}(i,j)$ of F(i,j) must follow the orthogonal principle, that is,

$$E(e(i,j)G^*(m,n)) = 0 \quad \text{for} \quad \text{all} \quad (i,j) \quad \text{and} \quad (m,n).$$

where $*$ denotes complex conjugate. Substituting $e(i,j) = \hat{F}(i,j) - F(i,j)$ into the preceding formula, it follows that

$$E(F(i,j)G^*(m,n)) = E(\hat{F}(i,j)G^*(m,n)).$$

Substituting $\hat{F}(i,j) = h \star G(i,j)$ into the preceding formula, we obtain

$$E(F(i,j)G^*(m,n)) = E[(h \star G(i,j))G^*(m,n)]$$

$$= E[(\sum_{k=-\infty}^{+\infty} \sum_{l=-\infty}^{+\infty} h(k,l) G(i-k, j-l)) G^*(m,n)]$$

$$= \sum_{k=-\infty}^{+\infty} \sum_{l=-\infty}^{+\infty} h(k,l) E[G(i-k, j-l) G^*(m,n)]. \tag{3.3.3}$$

Since F(i,j) and N(i,j) are assumed stationary, G(i,j) = F(i,j) +N(i,j) is also stationary and thus the preceding (3.3.3) can be written as

$$R_{FG}(i-m, j-n) = \sum_{k=-\infty}^{+\infty} \sum_{l=-\infty}^{+\infty} h(k,l) R_G(i-m-k, j-n-l).$$

This implies that

$$R_{FG}(i-m, j-n) = h \star R_G(i-m, j-n)$$

and then

$$R_{FG}(i,j) = h \star R_G(i,j).$$

Since the power spectrum of the linear convolution of two random processes is equal to the products of the power spectra, it follows that

$$P_{FG}(u,v) = H(u,v) P_G(u,v).$$

Therefore,

$$H(u,v) = \frac{P_{FG}(u,v)}{P_G(u,v)}. \tag{3.3.4}$$

Since F and N are uncorrelated and both have zero mean, it follows

$$R_{FG}(i,j) = R_F(i,j)$$

and

$$R_G(i,j) = R_F(i,j) + R_N(i,j).$$

Therefore, it follows that

$$P_{FG}(u,v) = P_F(u,v)$$

and

$$P_G(u,v) = P_F(u,v) + P_N(u,v).$$

From (3.3.4), we obtain

$$H(u,v) = \frac{P_F(u,v)}{P_F(u,v) + P_N(u,v)}.$$

QED

For the case where F has a known nonzero mean u_F and N has a known nonzero mean u_N, we can apply the Wiener filter to the image $G(i,j) - u_F - u_N$ to obtain the LSMS estimator $\tilde{F}$ for $F - u_F$. Hence, $\tilde{F} + u_F$ is the LSMS estimator of F.

Even though we have seen that the Wiener filter is the optimal filter in mean square sense under LSI and stationarity constraints, it does mean that the Wiener filter is the optimal choice for our PET models; even it may not even be suitable for the model. The reason is as follows: First, the Wiener filter requires knowledge of the signal power spectrum $P_F(u,v)$ and noise power spectrum $P_N(u,v)$ but the power spectra are in fact unknown and must be estimated. Due to the estimation error, it becomes a question whether the optimal property still remains for the Wiener filter after using estimated power spectra. Second, the PET image, as other images, cannot be modeled as a stationary random process since a stationary random process is assumed to have a constant mean for all pixels. Therefore, the PET image does not satisfy the constraint for the Wiener filter to achieve optimal estimation. Third, even for a stationary image, the estimator obtained by Wiener filter is the optimal only among LSI estimators but not all estimators. It would be better if one could find an optimal estimator among all estimators based on observation. Since it may be a more appropriate approach to model a image as a nonstationary random process, we will address how to design filters for nonstationary modeled images in the section "Adaptive Filtering". On the other hand, the Wiener filter does not distinguish whether or not the noise is signal-dependent or signal-independent. Since PET noise is signal-dependent noise, it is also desirable to to analyze such noise further before designing a filter for PET images. This issue will also be addressed in the section "Adaptive Filtering". Next, we want to discuss lowpass filtering, a basic technique in image restoration.

3.4 Lowpass Filter and Its Improvement

Consider an input-output system

$$Y(i,j) = h \star X(i,j) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} h(i-m, j-n)X(m,n) \tag{3.4.1}$$

where h(i,j) denotes a filter. The filter h(i,j) is said to be a lowpass filter if its frequency response satisfies H(u,v)$>$ 0 for (u,v)$\in$ Ω where Ω is a connected region containing the origin. For example, define h(i,j) by

$$h(i,j) = \begin{cases} \frac{1}{5} & i = j = 0; \\ \frac{1}{10} & i, j = -1, 0, 1, \quad (i,j) \neq (0,0); \\ 0 & \text{otherwise.} \end{cases} \tag{3.4.2}$$

Then, its frequency response H(u,v) is calculated by

$$H(u,v) = \frac{1}{10}(2 + e^{-iu} + e^{-iv} + e^{+iu} + e^{+iv} + e^{iu-iv} + e^{iu+uv} + e^{-iu+iv} + e^{-(iu+iv)})$$

$$= \frac{1}{5}(1 + \cos u + \cos v + \cos(u-v) + \cos(u+v))$$

$$= \frac{1}{5}(1 + \cos u + \cos v + 2\cos u \cos v)$$

where $|u| \leq \pi$ and $|v| \leq \pi$. It is not easy to explicitly express the maximum connected region where H(u,v)>0. However, we can see that H(u,v)>0 if $|u| < \pi/2$ and $|v| < \pi/2$. Therefore, The h(i,j) defined by (3.4.2) is a low-pass filter.

Lowpass filters may be the most commonly used filters for smoothing a noisy image. The filters are easy to implement. Sometimes, they are used together with other filters. The underlying assumption for using a lowpass filter is that the energy of a typical image is primarily concentrated in its low-frequency components and that the energy of a random noise is more spread out over the whole frequency domain. After applying a lowpass filter, the low-frequency components are preserved while the high-frequency components are removed. Consequently, it reduces a large amount of noise at the cost of losing some amount of signal at its high frequency part.

Often, the gross structure of an image appears as low-frequency components while the details of an image appear as high-frequency components. Therefore, the lowpass filtering causes some amount of detail loss of the original image. For some applications such as video image enhancement where such detail loss is not so critical, the lowpass filter is suitable. However, for PET image restoration, the details may contain the most important messages in the image, and it is not necessarily a wise choice to filter out high-frequency details.

Numerically, only a finite number of filter coefficients h(i,j) are nonzero, and in this case, a lowpass filter smoothes an image by moving a window throughout the image and taking "weighted average" for the pixels inside the windowed area. Observe that the neighboring pixels will be affected very much if some pixel inside the window is contaminated by a noise with very high variance. To reduce this effect, image too much. one may transform (nonlinearly) the image into an image which has less variance, apply lowpass filter for the transformed image and then take the inverse nonlinear transform for the smoothed image. This approach will reduce the effect from the pixel which contains high variance of noise. In Yang's master thesis [YANG91], a filter called square root filter is applied as the nonlinear transform before employing a lowpass filter to reduce the effect of high variance of noise occurring in PET image. The basic idea of square root filter is as follows:

Following the analysis in the Section 1 of this chapter, we take the number of photons N in each raysum to be a Poisson random variable with mean λ and variance λ. Expanding $\sqrt{N}$ around λ by a binomial series, we obtain:

$$\sqrt{N} = \sqrt{\lambda}\sqrt{1 + \frac{N-\lambda}{\lambda}}$$

$$= \sqrt{\lambda} + \frac{1}{2}\lambda^{-\frac{1}{2}}(N-\lambda) - \frac{1}{8}\lambda^{-\frac{3}{2}}(N-\lambda)^2 + \frac{1}{16}\lambda^{-\frac{5}{2}}(N-\lambda)^3 + ... + \lambda^{-n+\frac{1}{2}}O((N-\lambda)^n). \tag{3.4.3}$$

It is easy to verify $E(N-\lambda)^2 = \lambda$ and $E(N-\lambda)^3 = \lambda$. In general, from [HAIG67, pp. 7], Riordan (1937) provided the following iteration formula for $M_n = E(N-\lambda)^n$, $n > 0$:

$$M_{n+1} = \lambda n M_{n-1} + \lambda(\frac{d}{d\lambda})M_n.$$

Since $M_3 = \lambda$, from the preceding formula and by induction, it is easy to see that M_n is a polynomial of λ with order no more than (n-2) for $n \geq 3$. On the other hand, it is observed that the coefficient of the term $(N-\lambda)^n$ in (3.4.3) contains $\lambda^{-n+\frac{1}{2}}$. It follows that for $n \geq 3$, the expectation of the $(N-\lambda)^n$ term in (3.4.3) contains λ^α where $\alpha \leq -3/2$. Therefore, the mean of $\sqrt{N}$ is

$$E(\sqrt{N}) = \lambda^{1/2} - \frac{1}{8}\lambda^{-1/2} + \frac{1}{16}\lambda^{-3/2} + ... + \lambda^{-n+\frac{1}{2}}EO((N-\lambda)^n)$$

and the variance of $\sqrt{N}$ is

$$\begin{aligned} V(\sqrt{N}) &= EN - (E\sqrt{N})^2 \\ &= \lambda - [\lambda^{1/2} - \frac{1}{8}\lambda^{-1/2} + \frac{1}{16}\lambda^{-3/2} + ... + \lambda^{-n+\frac{1}{2}}EO((N-\lambda)^n)]^2 \\ &= \lambda - (\lambda - \frac{1}{4} + \frac{9}{64}\lambda^{-1} - \frac{1}{64}\lambda^{-2} + ...) \\ &\quad -(\lambda^{1/2} - \frac{1}{8}\lambda^{-1/2} + \frac{1}{16}\lambda^{-3/2} + ...)\lambda^{-n+\frac{1}{2}}EO((N-\lambda)^n) \\ &\quad -\lambda^{-2n+1}[EO((N-\lambda)^n)]^2 \end{aligned} \tag{3.4.4}$$

As noted above, $E(N-\lambda)^n$ is a polynomial of λ with order no more than (n-2) for $n \geq 3$. Therefore, the last two terms of (3.4.4) are bounded by λ^{-1}. Thus,

$$\begin{aligned} V(\sqrt{N}) &= \lambda - [\lambda - \frac{1}{4} + \frac{9}{64}\lambda^{-1} - \frac{1}{64}\lambda^{-2} + ... + O(\lambda^{-1})] \\ &= \frac{1}{4} + O(\lambda^{-1}). \end{aligned}$$

We conclude that the variance of $\sqrt{N}$ is approximately $\frac{1}{4}$ for a sufficient large λ so that the noise becomes nearly constant variance noise. This is the heuristic argument for using square root filter. Yang's [YANG91] numerical experiments also indicated that using a lowpass filter combined with a square root filter slightly improved the quality of smoothed image, compared with that by using only a lowpass filter.

3.5 Adaptive Filter

In a stationary image model, an image is assumed to have a constant mean of intensity and that the covariance between any two pixels is only related the distance between them, that is,

(i) $EX(i,j) = m_X$ for all (i,j);

(ii) EX(i,j)X(i+m, j+n) is independent of (i,j) for all i, j, m, n.

Obviously, the stationary model is oversimplified for a real-world image and such model based restoration algorithms usually blurs the image very much. To improve restoration, one may model images as nonstationary random processes by considering the abrupt changes of intensity and the characteristics in different regions of an image. To this end, a number of nonstationary models have been proposed, and a number of adaptive filters have been developed based on these models. Among them, Kuan *et al* [KUAN85] introduced a nonstationary mean, nonstationary variance image model and developed a linear minimum mean square error filter (LMS) based this model. In [KUAN85], the LMS filter was also implemented for PET image model. In this section, we illustrate adaptive filtering method by introducing the work done by Kuan and his co-workers.

For an image modeled as

$$G(i,j) = F(i,j) + N(i,j) \tag{3.5.1}$$

where F(i,j) denotes original image, G(i,j) denotes observed image and N(i,j) denotes noise. The linear minimum mean square error (LMS) estimator $\hat{F}$ of F is a estimator which minimizes $\sum_i \sum_j E(\hat{F}(i,j) - F(i,j))^2$ among all F satisfying $F = A + BG$ where A and B are two matrices. From regression theory, we have the linear minimum mean square error (LMS) estimator of F:

$$\hat{F}_{LMS} = E(F) + C_{FG}C_G^{-1}(G - E(G)) \tag{3.5.2}$$

where E(F) and E(G) are the mean of F and G, C_G is the covariance matrix of G and C_{FG} is the cross-covariance matrix of F and G. The following theorem summarizes the main result of [KUAN85] for the additive model (3.5.1).

Theorem 3.5.1: Assume F and N in the additive model (3.5.1) satisfy:

(i) the noise N(i,j) is uncorrelated, *i.e.*, E[N(i,j)N(r,s)]=0 for (i,j)$\neq$(r,s);

(ii) the components of F are uncorrelated, *i.e.*, the covariance matrix of F is diagonal;

(iii) the conditional mean of N given F is zero, *i.e.* $E(N|F) = 0$.

Then, the linear minimum mean square error estimator of F is given by

$$\hat{F} = w(i,j)E(F(i,j)) + (1 - w(i,j))G(i,j) \tag{3.5.3}$$

where

$$w(i,j) = \frac{\sigma_N^2(i,j)}{\sigma_F^2(i,j) + \sigma_N^2(i,j)}.$$

Proof. The cross covariance matrix of F and G can be calculated from

$$C_{FG} = E[(F - E(F))(G - E(G))^T]$$

$$= E[(F - E(F))(F + N - E(G))^T]$$

Since $E(N|F) = 0$, it follows $E(N) = E[E(N|F)] = E(0] = 0$ and

$$E(G) = E(F + N) = E(F) \tag{3.5.3a}$$

. Therefore,

$$C_{FG} = E[(F - E(F))(F - E(F) + N)^T]$$

$$= C_F + E[(F - E(F))N^T] \tag{3.5.4}$$

where C_F is the covariance matrix of F. Similarly, the covariance matrix of G is given by

$$C_G = E[(G - E(G))(G - E(G))^T]$$

$$= E[(F - E(F) + N)(F - E(F) + N)^T]$$

$$= C_F + E[(F - E(F))N^T] + E[N(F - E(F))^T] + C_N \tag{3.5.5}$$

where C_N is the covariance matrix of N. Since $E(N|F) = 0$ implies $E(N(i,j)|F) = 0$ for all i,j, it follows that E(N^T|F)=0 and

$$E[(F - E(F))N^T] = E_F[E_N[(F - E(F))N^T|F]]$$

$$= E_F[(F - E(F))E[N^T|F]]$$

$$= E_F[0] = 0. \tag{3.5.6}$$

From (3.5,4), (3.5.5) and (3.5.6), we obtain

$$C_{FG} = C_F$$

$$C_G = C_F + C_N. \tag{3.5.7}$$

Since N(i,j) is uncorrelated, C_N is a diagonal matrix with variance $\sigma_N^2(i,j)$ as its diagonal entries. Assuming that C_F is diagonal, the equation (3.5.7) implies that C_G is also diagonal with $\sigma_F^2(i,j) + \sigma_N^2(i,j)$ as its diagonal entries. Substituting these results back into (3.5.2), we obtain

$$\hat{F} = E(F(i,j)) + \frac{\sigma_F^2(i,j)}{\sigma_F^2(i,j) + \sigma_N^{(}i,j)}[G(i,j) - E(G(i,j))]. \tag{3.5.8}$$

Let

$$w(i,j) = \frac{\sigma_N^2(i,j)}{\sigma_F^2(i,j) + \sigma_N^{(}i,j)},$$

(3.5.3) follows.

QED

$$\frac{1}{w(i,j)} = \frac{\sigma_F^2(i,j) + \sigma_N^{(}i,j)}{\sigma_N^2(i,j)}$$

is referred to as signal-to-noise (SNR) ratio in literature. From the formula (3.5.3), we see that for an image with low signal-to-noise (SNR) ratio, the LMS estimate of F puts more weight to the mean of F(i,j) since it is more difficult to estimate F(i,j) from the highly contaminated observation data G(i,j). Conversely, for a high SNR image, the LMS estimate puts more weights to the observed data G(i,j). Since the mean and variance of F(i,j) is unknown in practice, the local mean and local variance of G(i,j) are used to estimate the mean and variance of G(i,j) and to calculate the mean and variance of F(i,j) by (3.5.3a) and (3.5.7). Since the local statistics are involved in practice, the estimate of F(i,j) is also referred to as local LMS estimate of F(i,j).

Next, we consider if the adaptive filter can be applied to PET image restoration problem. To this end, we check if all conditions in the preceding theorem are satisfied. We know from Section 3.2 that for a PET raysum image can be modeled as Poisson model as

$$G(i,j) = \lambda(i,j) + N(i,j)$$

where G(i,j) denotes noisy raysum and $\lambda(i,j)$ denotes mean of G(i,j). Since raysum G(i,j) are independent random variables, N(i,j), $N(i,j) = G(i,j) - \lambda(i,j)$, are also independent so that the covariance matrix C_N is diagonal. It means the condition (i) of Theorem 3.5.1 is satisfied. Since each $\lambda(i,j)$ is a constant, it follows that the components of λ are uncorrelated and then the condition (ii) holds. Observe $E[N|\lambda] = E[N] = E[G - \lambda] = \lambda - \lambda = 0$. It follows that the condition (iii) of Theorem 3.5.1 is also satisfied. Therefore, all conditions in Theorem 3.5.1 are satisfied and the adaptive filter can be applied to the Poisson model based PET problem. For the Cox model, the condition (i) and (iii) are easy to verify. If we further assume that the random mean measure $\Lambda(i,j)$'s are independent, then all conditions in the theorem are satisfied and thus the theorem can be applied to the Cox model based PET raysum image to obtain LMS estimator of F.

3.6 Concluding Remarks

The filtering methods considered in this chapter estimate the original images from observation data in the presence of noise. Wiener filter is the optimal mean square filter under LSI and stationary constraints, while the adaptive filter in

Section 3.5 is the optimal mean square filter in a special linear and nonstationary model. Low-pass filter smoothes an image by preserving the most of its energy.

A L_2 norm based optimal filter often emphasizes global results but may neglect some local features; while a lowpass filter often preserve most energy in the image but may lose some amount of signal with high frequency. For PET image application, some high frequency parts and local features maybe contains the most important signal in the image and are deserved to be preserved. To preserve features present in the image, it is desirable to define what may characterize the features and if possible find an algorithm to detect them. A boundary where a rapid intensity change occurs is considered to contain some features in the image. By detecting these changes ("edges"), one can understand the gross structure of the image; if the changes occurs in some local area very frequently, one may think there may exist some singularity. We will discuss these idea further in the next chapter and try to develop a filtering scheme which can smooth images while preserving features contained in the image in the remainder of this thesis.

CHAPTER IV
EDGE DETECTION

4.1 Introduction

An edge in an image is a boundary where a significant change occurs in some physical aspect of the image, such as intensity, color or texture. We are mainly concerned with the change of intensity. Since edges represent the basic structure of an image, detecting edges is very important for image analysis and is also very useful in some applications such as object identification, image segmentation and low bit-rate image coding. In this chapter, we study several edge detection methods.

The gradient magnitude method is by its name an edge detection method based on the gradient magnitude (modulus). The Laplacian method is a method of detecting edges by finding zero crossings of the Laplacian of a image. Both methods can be derived from the observations in fundamental calculus. They are also used as basic ingredient of other more sophisticated edge detecting schemes. Marr and Hildreth [MARR80] introduced the concept of multiscale edges and used the Laplacian method to detect edges for each scale. Gregson [GREG93] proposed an edge detection scheme of using angular dispersion of gradient direction. We discuss all these methods in this chapter. Mallat and Zhong [MALL92a] studied the characteristics of multiscale edges in the setting of wavelets and we will discuss them in the next chapter.

4.2 Gradient Magnitude Method and Laplacian Method

We illustrate these two methods starting from the one dimensional case. Let f(x) be a nonnegative function with second continuous derivative. f(x) can be interpreted as a signal in signal processing applications. From calculus, we know that if $|f'(x)|$ is very large then f(x) is changing very rapidly and thus the intensity (magnitude, or modulus) of f(x) is also changing rapidly. If $|f'(x)|$ is greater than a certain threshold, the point x is a candidate edge point. The gradient magnitude method is derived from this observation. The task of detecting all edges in a signal is to find all values of x such that $|f'(x)|$ is larger than the threshold. On the other hand, $f'(x)$ can be considered possibly large when it reaches a local extremum. We also know from calculus that $f'(x)$ reaches a local extremum when $f''(x)$ has a zero crossing. Therefore, the task of detecting edges may be fulfilled by finding zero crossings of $f''(x)$. This is the basic idea of Laplacian method.

Let us discuss gradient magnitude method first. The two dimensional counterpart of $f'(x)$ is the gradient $\bigtriangledown$f(x,y) given by

$$\bigtriangledown f(x,y) = \frac{\partial f(x,y)}{\partial x}\hat{i}_x + \frac{\partial f(x,y)}{\partial y}\hat{i}_y \tag{4.2.1}$$

where $\hat{i}_x$ and $\hat{i}_y$ are unit vectors in x and y directions respectively. Detecting edges in a two-dimensional image is equivalent to finding all values of (x,y) where the magnitude of $\bigtriangledown f(x,y)$ is greater than a certain threshold. For numerical implementation, $\frac{\partial f(x,y)}{\partial x}$ may be replaced (for example) by

$$f(i,j) - f(i-1,j), \tag{4.2.2}$$

$$(f(i+1,j) - f(i-1,j))/2, \tag{4.2.3}$$

or

$$[f(i+1,j+1)+f(i+1,j)+f(i+1,j-1)-f(i-1,j+1)-f(i-1,j)-f(i-1,j-1)]/6 \tag{4.2.4}$$

and $\frac{\partial f(x,y)}{\partial y}$ can be discretized in a similar manner. From the discretized forms, we are able to calculate gradient magnitude of each point and compare it with the threshold and determine if the pixel is an edge point.

Next, we discuss Laplacian method for detecting edges in an image. A generalization of the second derivative to a two-dimensional function f(x,y) is Laplacian $\bigtriangledown^2 f(x,y)$ given by

$$\bigtriangledown^2 f(x,y) = \frac{\partial^2 f(x,y)}{\partial x^2} + \frac{\partial^2 f(x,y)}{\partial y^2}. \tag{4.2.5}$$

Detecting edges is equivalent to finding all zero crossings of $\bigtriangledown^2 f(x,y)$. For numerical implementation, we use (4.2.2) for $\frac{\partial f(x,y)}{\partial x}$ and discretize $\frac{\partial^2 f(x,y)}{\partial x^2}$ by

$$f_x(i+1,j) - f_x(i,j) = f(i+1,j) - 2f(i,j) + f(i-1,j). \tag{4.2.6}$$

From (4.2.6), and by discretizing $\frac{\partial^2 f(x,y)}{\partial y^2}$ in a similar manner, we obtain a discrete form of $\bigtriangledown^2 f(x,y)$:

$$f(i+1,j) + f(i-1,j) + f(i,j+1) + f(i,j-1) - 4f(i,j). \tag{4.2.7}$$

The linear convolution of an image f(i,j) and a filter h(i,j) is defined by

$$h * f(i,j) = \sum_{k=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} h(i-k, j-l) f(k,l).$$

The operations on the discretized image f(i,j) in the above paragraphs can be expressed as the linear convolution of f(i,j) with some filter h(i,j). For example, the filter corresponding to (4.2.4) is

$$h_1(i,j) = \frac{1}{6} \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}$$

and the filter for (4.2.7) is

$$h_2(i,j) = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}.$$

The above $h_1(i,j)$ is called a differencing filter and $h_2(i,j)$ is called the Laplacian filter (even though there are other ways of discretizing the Laplacian).

Both the gradient magnitude method and the Laplacian method can be extended to the case where an image is modeled as a stochastic process rather than a deterministic function. However, the gradient magnitude method is sensitive to the variance of noise. We use the filter of (4.2.2) to illustrate this fact. For simplicity, we consider a noisy "flat" image modeled as

$$F(i,j) = u + N(i,j) \tag{4.2.8}$$

where u is a real number and $\{N(i,j)\}$'s are independent noise with zero mean and constant variance σ^2. The original image does not have any edges but false edges may be generated in the observed image due to the noise. Next, we want to see how the number of false edges is related to the variance of noise. For the gradient magnitude method using filter of (4.2.2), the gradient magnitude of F(i,j) is given by

$$M(i,j) = \sqrt{(N(i,j) - N(i-1,j))^2 + (N(i,j) - N(i,j-1))^2}$$

The mean square of M(i,j) is given by

$$EM(i,j)^2 = E((N(i,j) - N(i-1,j))^2 + (N(i,j) - N(i,j-1))^2)$$

$$= 4\sigma^2.$$

Thus, the mean square of the gradient magnitude is proportional to the variance of noise. Since each gradient magnitude is compared with a given threshold and a pixel is considered to be an edge if the the magnitude is larger than the threshold, it follows that the larger the variance of the noise is, the more edges will be likely to possess magnitude greater than the threshold and thus the more false edges will be likely to be generated.

The Laplacian method is also sensitive to noise. We use the same noisy "flat" image model defined in (4.2.8) as an example to illustrate this fact. The original image does not have any edges and we want to see how many false edges will be generated by using the Laplacian edge detection scheme. For the Laplacian method using filter $h_2(i,j)$, the Laplacian for pixel (i,j) is

$$\bigtriangledown^2 F(i,j) = N(i+1,j) + N(i-1,j) + N(i,j+1) + N(i,j-1) - 4N(i,j). \tag{4.2.9}$$

Since each N(.,.) is a random variable with zero mean and variance σ^2 and each N(.,.) is independent to each other, $\bigtriangledown^2 F(i,j)$ is a random variable with zero mean and variance $20\sigma^2$. It suggests that each $\bigtriangledown^2 F(i,j)$ oscillates around zero and that $\bigtriangledown^2 F(i,j)$ may change signs frequently. To justify the statement above, consider

$$\bigtriangledown^2 F(i+3,j) = N(i+4,j)+N(i+2,j)+N(i+3,j+1)+N(i+3,j-1)-4N(i+3,j).$$

Observe that $\bigtriangledown^2 F(i+3,j)$ also has mean zero and variance $20\sigma^2$. Since $\bigtriangledown^2 F(i+3,j)$ is a linear combination of noises which do not appear in (4.2.9), $\bigtriangledown^2 F(i+3,j)$ and $\bigtriangledown^2 F(i,j)$ are independent. Thus, the probability that $\bigtriangledown^2 F(i,j)$ and $\bigtriangledown^2 F(i+3,j)$ have different signs is $\frac{1}{2}$. The same conclusions can be drawn for the pixel (i-3, j), (i, j+3), (i, j-3) and others. Here we see that there is a sign change every four pixel in a raw with probability $\frac{1}{2}$. and that there is a sign change every four pixel in a column with probability $\frac{1}{2}$. Therefore, zero crossings and thus false edges are likely to be generated by using the Laplacian edge detection method.

There are many variations of both methods for the purpose of reducing noise sensitivity. Some noise smoothing methods discussed in the previous chapter may be desirable prior to applying the gradient magnitude method to reduce the number of false edges. For example, if we take averages for the pixels inside the 3×3 windows centered at (i,j) for the image model defined in (4.2.8), the mean square of the gradient magnitude at (i,j) becomes

$$E\bar{M}(i,j)^2 = E((\bar{N}(i,j) - \bar{N}(i-1,j))^2 + (\bar{N}(i,j) - \bar{N}(i,j-1))^2)$$

$$= E(\frac{1}{9}(\sum_{k=-1}^{1}\sum_{l=-1}^{1} N(i+k,j+l) - \sum_{k=-1}^{1}\sum_{l=-1}^{1} N(i-1+k,j+l))^2$$

$$+E(\frac{1}{9}(\sum_{k=-1}^{1}\sum_{l=-1}^{1} N(i+k,j+l) - \sum_{k=-1}^{1}\sum_{l=-1}^{1} N(i+k,j-1+l))^2$$

$$= \frac{1}{81}E(N(i+1,j+1) + N(i+1,j) + N(i+1,j-1)$$

$$-N(i-2,j+1) - N(i-2,j) - N(i-2,j-1))^2$$

$$+\frac{1}{81}E(N(i+1,j+1)+N(i,j+1)+N(i-1,j+1)$$
$$-N(i-1,j-2)-N(i,j-2)-N(i-1,j-2))^2$$
$$=\frac{12}{81}\sigma^2.$$

Here we see that after applying averaging filter, the gradient magnitude is significantly reduced. Therefore, according to the analysis above, the number of false edges will be significantly reduced in the gradient magnitude edge detection method. For the Laplacian method, since many false edges may be generated by even small perturbations, it is anticipated to reduce the number of false edges by thresholding the local variance of an image. That is, the edge pixel with small local variance is considered to be generated by noise and thus the edge is considered to be a false edge. The next section gives a different approach to reducing noise sensitivity.

4.3 Gradient Angular Method

The gradient magnitude method and the Laplacian method discussed in the previous section are found to be sensitive to noise. Gregson [GREG93] proposed an approach to edge detection by using angular dispersion of the gradient direction. The method is found to be less sensitive to the variance of noise but more dependent on the signal-to-noise ratio (SNR) of an image. In this chapter, we discuss the basic idea of this approach.

Usually, signal-to-noise ratio (SNR) is used to measure the degree of noise in an image and to measure the efficiency of a restoration algorithm by comparing SNR ratio before and after employing such algorithm. Consider the additive noise model given by

$$G(i,j)=F(i,j)+N(i,j) \tag{4.3.1}$$

where F(i,j) denotes the original image, N(i,j) denotes zero-mean noise and G(i,j) denotes the observed image. For this model, the signal-to-noise ratio (SNR) of the image is usually defined by

$$10\log_{10}\frac{E(F(i,j))^2}{E(N(i,j))^2}$$

for each pixel, and

$$10\log_{10}\frac{\sum_{i,j}E(F(i,j)-\bar{F}(i,j))^2}{\sum_{i,j}E(N(i,j))^2}$$

for the whole image, where

$$\bar{F}(i,j)=\sum_{i,j}F(i,j).$$

For a single image $\{f(i,j)\}$, assuming that the variance of noise at each pixel (i,j) is known as $\sigma^2(i,j)$, then we use each pixel value $f(i,j)$ to estimate $F(i,j)$ and calculate the SNR for pixel (i,j) by

$$10\log_{10}\frac{f^2(i,j)}{\sigma^2(i,j)}$$

and the SNR for the whole image by

$$10\log_{10}\frac{\sum_{i,j}(f(i,j)-\bar{f}(i,j))^2}{\sum_{i,j}\sigma^2(i,j)}$$

where

$$\bar{f}(i,j)=\sum_{i,j}f(i,j).$$

This SNR is in fact a mean-square measurement and it does not take edge information into account. Gregson [GREG93] proposed a slightly different SNR ratio for the purpose of edge detection, which will be introduced below.

For the purpose of simplicity, we assume that F(x,y) is a real number rather than a random valuable and that two gradient components

$$G_X=\frac{\partial G(x,y)}{\partial x}$$

$$G_X=\frac{\partial G(x,y)}{\partial y}$$

are independent and normally distributed with means

$$u_X=EG_X$$

$$u_Y = EG_Y$$

and variances

$$\sigma_X^2 = V(G_X) = \sigma_Y^2 = V(G_Y) = \sigma^2.$$

The joint density function of g_X and g_Y is therefore

$$f(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x-u_X)^2+(y-u_Y)^2}{2\sigma^2}}$$

Denoting $u_R = \sqrt{u_X^2 + u_Y^2}$ and $u_\theta = \arccos \frac{u_X}{u_R}$ and converting to polar coordinates, it follows that

$$f_{R,\Theta}(r, \theta) = \frac{r}{2\pi\sigma^2} e^{\frac{-u_R^2}{2\sigma^2}} e^{-\frac{r^2 - 2ru_R\cos(\theta-u_\theta)}{2\sigma^2}}$$

$$= \frac{r}{2\pi\sigma^2} e^{-\frac{u_R^2 \sin^2(\theta-u_\theta)}{2\sigma^2}} e^{-\frac{r-u_R\cos(\theta-u_\theta)^2}{2\sigma^2}}. \tag{4.3.2}$$

Letting $\beta = \theta - u_\theta$ and integrating (4.3.2) with respect to r, we obtain

$$f_B(\beta) = \frac{1}{2\pi\sigma^2} e^{-\frac{u_R^2 \sin^2\beta}{2\sigma^2}} \int_0^{+\infty} re^{-\frac{(r-u_R\cos\beta)^2}{2\sigma^2}} dr.$$

Denote

$$\delta = \frac{u_R}{\sigma}. \tag{4.3.3}$$

In [GREG93], this δ is defined to be the SNR ratio of the pixel (x,y) in the image for the purpose of edge detection. Since the modulus of the gradient magnitude u_R reflects the local intensity changes of an image, it can be used to find edge pixels: a pixel is considered to be an edge pixel if u_R is greater than some threshold. In other words, u_R reflects the edge information (or structured component) of the image. Since σ reflects the noise level of the gradient components, it is reasonable to treat δ as SNR ratio in the context of edge detection.

Substituting δ into $f_B(\beta)$, it follows

$$f_B(\beta) = \frac{1}{2\pi} e^{-\frac{\delta^2 \sin^2\beta}{2}} [\delta \cos\beta \int_{-\delta\cos\beta}^{+\infty} e^{-\frac{r^2}{2}} dr + \int_{-\delta\cos\beta}^{+\infty} re^{-\frac{r^2}{2}} dr]$$

$$= \frac{1}{2\pi}e^{-\frac{\delta^2}{2}} + \frac{1}{2\pi}e^{-\frac{\delta^2 \sin^2\beta}{2}} \delta \cos\beta \int_{-\delta\cos\beta}^{+\infty} e^{-\frac{r^2}{2}} dr$$

$$= \frac{1}{2\pi}e^{-\frac{\delta^2}{2}} + \frac{1}{\sqrt{2\pi}}e^{-\frac{\delta^2 \sin^2\beta}{2}} \delta \cos\beta \Phi(\delta\cos\beta)$$

$$= \frac{1}{2\pi}e^{-\frac{\delta^2}{2}} + \frac{1}{\sqrt{2\pi}}e^{\frac{-\delta^2+(\delta\cos\beta)^2}{2}} \delta \cos\beta \Phi(\delta\cos\beta) \tag{4.3.4}$$

where Φ denotes the distribution function of a standard normal random variable. From (4.3.4), we see that the density of β is a function of δ and β. If δ is reduced to zero, the density function of β becomes the density function of a uniformly distributed random variable. As δ increases, the first term in (4.3.4) decreases to zero and the second term tends to be near zero if $\beta \neq 0$, the probability density is concentrated near $\beta = 0$. Heuristically, this implies that the variance of β increases as the value of δ decreases. Let us look at it in a little more detail. Since $E\beta = 0$,

$$\sigma_\beta^2 = \int_0^{2\pi} \beta^2 [\frac{1}{2\pi}e^{-\frac{\delta^2}{2}} + \frac{1}{\sqrt{2\pi}}e^{\frac{-\delta^2+(\delta\cos\beta)^2}{2}} \delta \cos\beta \Phi(\delta\cos\beta)] d\beta$$

$$= \frac{1}{2\pi}e^{-\frac{\delta^2}{2}} \int_0^{2\pi} \beta^2 (1 + \sqrt{2\pi}e^{\frac{(\delta\cos\beta)^2}{2}} \delta\cos\beta\Phi(\delta\cos\beta)) d\beta$$

$$= \frac{\pi^2}{3}e^{-\frac{\delta^2}{2}} + e^{-\frac{\delta^2}{2}} \frac{1}{\sqrt{2\pi}} \int_0^{2\pi} \beta^2 e^{\frac{(\delta\cos\beta)^2}{2}} \delta\cos\beta\Phi(\delta\cos\beta) d\beta$$

We do not know whether it is possible to show rigorously that σ_β^2 monotonically decreases as δ increases. In [GREG93], a plot of σ_β^2 versus δ from numerical calculations indicates that σ_β^2 decreases as δ increases. Since δ is interpreted as signal to noise ratio (SNR) in the context of edge detection, it follows that the density of the deviation β is dependent on SNR in that a higher SNR corresponds to a lower variance of gradient direction and *vice versa*. If the variance of the gradient direction β in a pixel is high, the SNR of the pixel is low and then the pixel contains less structured component of the image and should not be considered as an edge pixel. On the other hand, if the variance of the gradient direction β is low, the SNR of the pixel is high and then the pixel contains more structured component of the image and is likely to be an edge pixel. Therefore, we can distinguish noise from true signal be comparing the variance of gradient direction β with a given threshold.

Next, we address implementation issue. Since u_X and u_Y are generally unknown, the density function in (4.3.4) must be estimated by observed discrete values and the variance of gradient direction is calculated from the estimated density function. In [GREG93], Gregson used the following estimation formula for the variance of gradient direction:

$$\hat{\sigma}^2 = 2(1 - \frac{|\Sigma G|}{\Sigma|G|}) \tag{4.3.5}$$

where $|G|$ denotes gradient magnitude and the summation is taken over a neighborhood of the given pixel.

Finally, we make some comments on Gregson's approach. This approach differs from other methods such as the gradient magnitude method and the Laplacian methods in the manner that it exploits the characteristics of angles as opposed to magnitude. The numerical experiments by Gregson also indicated that it performs better than gradient magnitude method. However, several issues need to be investigated further. First, the Gaussian assumption for the gradient components is not commonly accepted and needs to be further explored. Let us take an image modeled as (4.3.1) as an example. Let F(i,j) denote the discretized F(x,y) and N(i,j) denotes discretized N(x,y). Suppose that F(i,j) and N(i,j) are Gaussian distributed and

$$h_x(i,j) = \frac{1}{4}\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ 1 & 0 & -1 \end{bmatrix}$$

and

$$h_y(i,j) = \frac{1}{4}\begin{bmatrix} 1 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & -1 \end{bmatrix}$$

is used as the differencing filter for x and y directions. Then, the discretized gradient components are Gaussian distributed and independent (the argument for this can be seen in the proof of Theorem 6.3.2 in Chapter 6). This indicates that the Gaussian assumption for the gradient components is reasonable if a Gaussian model of the image is reasonable. For other types of images such as Poisson modeled images in PET applications, the gradient components need not be Gaussian and therefore how to use angular information to detect edges is still an open issue. This problem will be addressed in Chapter VI and is one of main results of this dissertation. Second, the estimation formula (4.3.5) seems have no solid justification. It is desirable to obtain a reasonable estimate of the variance of the gradient dispersion β. Third, the pixel-wise thresholding strategy seems have some shortcomings. The scheme distinguishes a false edge from a true edge by comparing the variance of gradient direction at the pixel with some given threshold. It neglects the coherence between the pixel and its neighboring pixels along edge curves. Edge detection could apparently be improved by an edge tracking procedure which looks at neighboring pixels in a direction orthogonal to the gradient direction. This idea has been implemented in our numerical experiments and is discussed in the next chapter.

4.4 Multiscale Edge Detection

Marr and Hildreth [MARR80] observed the intensity changes of smoothed versions of an image at different scales and argued that the changes at different scales contain important information about the image. Their approach is called multiscale edge detection.

In later 70's, Marr and other researchers tried to develop a science of vision called "robotic (computer) vision" to simulate the human vision system. One goal was to develop computer algorithms for object recognition. A fundamental question raised in the research was how to distinguish the boundaries of blurry shadow regions and the contours of fine detail regions present in the same image. Marr and Hildreth suggested that it could be done by lowpass filtering the original image at several different cutoff frequencies (scales) and detecting edges for each of the smoothed images. The contours of different types of regions could then be determined by examining the edge maps of different scales. Generally, the edges of large shadow regions can be found in the edge maps of large scales while the edges of fine details can be found in that of small scales. This is the basic idea of multiscale edge detection.

The lowpass filter proposed in [MARR80] is the discretized Gaussian function $h_\sigma(x, y)$ given by

$$h_\sigma(x, y) = e^{-(x^2+y^2)/(2\pi\sigma^2)}$$

where σ is a scaling parameter. Since the frequency response (Fourier transform) of h(x,y) is given by

$$H(u, v) = 2\pi^2\sigma^2 e^{-\pi\sigma^2(u^2+v^2)/2},$$

it follows that σ effectually determines the cutoff frequency – a larger σ corresponds to a lower cutoff frequency. For the smoothed image, they used the Laplacian method to detect edges at each scale.

Marr and Hildreth's method can be numerically implemented as follows. First, obtain the discretized Gaussian function $h_{\sigma_k}(i, j)$ for σ_k. Second, for each scale, perform linear convolution for f(i,j) and h_{σ_k} to obtain the smoothed versions of the image

$$f_{\sigma_k}(i, j) = h_{\sigma_k} * f(i, j)$$

$$\sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} h_{\sigma_k}(m, n) f(i-m, j-n).$$

Then, apply the Laplacian filter

$$h_2(i, j) = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}.$$

to each smoothed image and find edges using the Laplacian method discussed in Section 4.2.

In Marr and Hildreth's approach, the values of $\frac{\partial^2 h_\sigma * f(x,y)}{\partial x^2}$ and $\frac{\partial^2 h_\sigma * f(x,y)}{\partial y^2}$ were not saved. They conjectured that an image can be perfectly reconstructed from the multiscale edge locations but they did not give an explicit reconstruction algorithm; they did not even give a method of obtaining an approximation to the original image. This conjecture was disproved by Meyer [MEYE93] later. A natural question is: can the original image be reconstructed if not only edge locations but also information about the first or second derivatives at edge locations are saved. This problem has been addressed by Mallat *et al* in the setting of wavelets and will be discussed further in the next chapter.

To conclude the discussions of this section, we make the following comments on Marr and Hildreth's approach:

First, Gaussian functions are very smooth and in fact infinitely differentiable. A smooth function is more likely to preserve edges that are present in the original images and less likely to introduce edges that are not present in the original images. However, Lindeberg [LIND90] pointed out that even though it is theoretically sound, the method of using Gaussian functions generates undesirable spurious edges after discretizing h(x,y), especially at fine levels of scale. It is still an open problem if these undesirable extra edges can be avoided or reduced by using suitable wavelets.

Second, Gaussian functions are relatively localized in the sense that they achieve minimum product of effective duration and effective bandwidth [PEEB76]. Let f(t) $\in L^2$ denote a signal and F(ω) denote its Fourier transform. The effective duration of f(t) is defined by

$$\Delta t = \sqrt{\frac{\int t^2 |f(t)|^2 dt}{\int |f(t)|^2 dt}}$$

and the effective bandwidth of f(t) is defined by

$$\Delta\omega = \sqrt{\frac{\int \omega^2 |F(\omega)|^2 d\omega}{\int |F(\omega)|^2 d\omega}}.$$

It is well known that the effective duration and effective bandwidth of a signal satisfy Heisenberg's Inequality [PEEB76]:

$$\Delta t \Delta \omega \geq \frac{1}{2}.$$

and that Gaussian function is the unique function for which the equality holds [PEEB76]. However, the Gaussian function does not have compact support in either time (space) domain or frequency domain. This limitation makes it undesirable in some applications. For example, since the Gaussian function does not have compact support in the space domain, if the original image contains noise then the noise in each pixel will spread to all pixels after taking the Gaussian lowpass filter. It makes more difficult to detect edges in an image where noise is present. Some wavelets with compact support may provide hope to solve this problem.

Third, Gaussian functions at different scales do not constitute an orthogonal basis for L^2 space. Orthogonality is desirable to avoid redundancy and to avoid introducing noise correlations in image processing applications. We will see in the next chapter that wavelet filters possess desirable orthogonality properties.

Finally, Marr and Hildreth used the Laplacian method discussed in the previous section to detect edges for the smoothed images. This choice also has some drawbacks. First, zero-crossings of the derivative of a function correspond to the extrema but not necessary maxima of the derivative and thus are not necessarily edge points. Thus the Laplacian method may generate false edges. Second, as mentioned before, the Laplacian method is sensitive to small perturbations of the signal and it will become worse for the smoothed images at fine scales. To overcome these drawbacks, Mallat *et al* [MARR91, 92a, 92b] proposed an edge detection scheme by using the modulus maxima of wavelet transforms. As will be seen in the next chapter, their approach is in essence a gradient magnitude method for multiscale edge detection using wavelets rather than Gaussian kernel in the filters at different scales.

4.5 Concluding Remarks

We studied several edge detection methods in this section. The gradient magnitude method and the Laplacian method are two basic edge detection schemes. They are found to be sensitive to noise. To reduce the noise sensitivity, a smoothing filter may be applied before employing edge detecting algorithms. The gradient angular method is an edge detection scheme which exploits the characteristics of gradient direction. Multiscale edge detection methods try to analyze an image by filtering the image at several scales and detecting edges for the filtered images.

Marr and Hildreth's multiscale edge detection method is helpful for analyzing the structure of an image. The method would be better if the original image could be reconstructed or approximated from multiscale edge information. This issue will be addressed in the following section by using wavelets. Gregson's gradient angular method provides an interesting alternative approach to edge detection. But as mentioned in Section 4.3, how to justify the method for more realistic image models, including the PET image model, is still an open issue. It is also desirable to find a better estimate for the angular dispersion and to incorporate the angular method with the multiscale edge detection method. These problems will be addressed in the following chapters.

CHAPTER V
WAVELET BASED RESTORATION METHODS

5.1 Introduction

Wavelet theory is a unified framework for a number of techniques in signal processing applications including the constant-Q method in non-stationary signal analysis, subband coding in signal compression and multiscale edge detection in computer vision. In the middle 80's, French researchers lead by Morlet, Grossmann and Meyer built strong mathematical foundations for the subject and named their work "wavelets". Recent results include the Balian-Low theorem, which states roughly that while there are orthonormal wavelet bases with good time-frequency localization, good time-frequency localization cannot be achieved by windowed Fourier transform techniques in classical non-stationary signal analysis [DAUB90]. Recently, also, Mallat *et al* [MALL91, 92a, 92b] applied wavelet methods to image processing in areas such as edge detection, image coding and image restoration. The development of wavelet theory and applications has been rapid. A comprehensive review on the recent development in the field can be found in [RIOU91] and [LIN94]. In this chapter, we focus on how to apply wavelet methods to signal and image restoration.

This chapter is intended to illustrate wavelet methods for signal and image restoration applications in a self-contained and rigorous manner and without being too lengthy. To this end, we limit our discussions to discrete signals with finite length so that it will only require linear algebra as a prerequisite. To avoid complicated notations, we are only concerned with one-dimensional case (we will extend the derived results to the two-dimensional case in the following chapter). Also, we mostly consider wavelet transforms under Haar basis throughout the chapter for the purpose of simplicity. During the discussion, we will refer to some references for more general and detailed discussions.

5.2 Signal Decomposition and Reconstruction

A one-dimensional discrete signal with finite length N can be expressed as a vector $f = (f_0, f_1, f_2, ..., f_{N-1})$ in an N-dimensional vector space. (A two-dimensional discrete signal or an image can be expressed as a matrix and will be discussed in the next chapter.) suppose G and H are $K \times N$ matrices and G^+ and H^+ are $N \times K$ matrices with the property that

$$G^+G + H^+H = I \tag{5.2.1}$$

where I denotes an $N \times N$ identity matrix. Then, the signal f can be decomposed into two components u and v via linear transforms

$$\begin{cases} u = Gf \\ v = Hf, \end{cases} \tag{5.2.2}$$

and f can be reconstructed from the components u and v via

$$f = G^+u + H^+v. \tag{5.2.3}$$

There exist many matrices which can be used for such a decomposition and reconstruction. For some matrices G and H, (5.2.1) has interesting properties which can be exploited. We give three examples here. The first example is associated with the fast Fourier transform (FFT) algorithm, which is one of the most important algorithms in signal processing. This example will be used to compare with wavelet transform. The second example is called Haar transform, which leads to the discrete Haar wavelet transform, one of the most important discrete wavelet transforms (DWT) in applications. The third example is very similar to the Haar transform but the matrices possess additional symmetry properties. The Haar transform in the second example is associated with an orthogonal Haar wavelet, the symmetric Haar transform in the third example is associated with a non-orthogonal Haar wavelet. The third example will be actually used in our numerical algorithm.

Example 1. Fast Fourier Transform

Suppose $N = 2^m$ where m is a positive integer. Let $\hat{f}$ denote the discrete Fourier transform (DFT) of signal f, that is,

$$\hat{f}_k = \sum_{l=0}^{N-1} e^{-j2\pi kl/N} f_l, k = 0, 1, ..., N-1.$$

For k, l = 0, 1, 2, ..., N-1, set

$$G = (g_{kl}) = \begin{cases} 0 & \text{if l is odd} \\ e^{-j2\pi kl/N} & \text{if l is even} \end{cases} \tag{5.2.4}$$

and

$$H = (h_{kl}) = \begin{cases} 0 & \text{if l is even} \\ e^{-j2\pi kl/N} & \text{if l is odd.} \end{cases} \tag{5.2.5}$$

If we set u =Gf, v = Hf, then $\hat{f} = u + v$, that is, the DFT of f can be decomposed into "even component" u and "odd component" v. On the other hand, if we set

$$G^+ = (g^+_{kl}) = \begin{cases} 0 & \text{if l is odd} \\ \frac{1}{N} e^{j2\pi kl/N} & \text{if l is even} \end{cases} \tag{5.2.6}$$

and

$$H^+ = (h^+_{kl}) = \begin{cases} 0 & \text{if l is even} \\ \frac{1}{N} e^{j2\pi kl/N} & \text{if l is odd,} \end{cases} \tag{5.2.7}$$

it can be easily verify that G, H, G^+ and H^+ satisfy (5.2.1). Therefore, the original signal f can be reconstructed from components u and v.

Let us introduce the basic idea of the FFT algorithm. For k=0, 1, ..., N/2-1,

$$u_{k+N/2} = \sum_{l=0}^{N/2-1} f_{2l} e^{-j2\pi(2l)(k+N/2)} = \sum_{l=0}^{N/2-1} f_{2l} e^{-j2\pi(2l)k} = u_k$$

It can also be verified that $v_{k+N/2} = -v_k$. Therefore, the DFT $\hat{f}$ of f, k=0, 1, ..., N-1, can be calculated by

$$\hat{f}_k = u_k + v_k, k = 0, 1, 2, ..., N/2 - 1$$

$$\hat{f}_{k+N/2} = u_k - v_k, k = 0, 1, 2, ..., N/2 - 1$$

Observe from (5.2.4) that u_k is the DFT coefficients of the even entries of f and observe from (5.2.5) that v_k can be obtained by multiplying $e^{-j2\pi k/N}$ to the DFT coefficients of the odd entries of f, that is,

$$u_k = \hat{f^e},$$

$$v_k = e^{-j2\pi k/N} \hat{f^o}$$

where $\hat{f^e}$ denotes the DFT of $f^e = (f_0, f_2, ..., f_{N-2})$ and $\hat{f^o}$ denotes the DFT of $f^o = (f_1, f_3, ..., f_{N-1})$. Since N is assumed to be a power of 2, the process stated above can be applied to u_k and v_k, each of them has length $N/2$. Therefore, the DFT of a signal f of length N can be obtained by computing two DFT's of signals of length $N/2$ and in addition $N/2$ multiplications. And the process can be repeated since N is a power of 2. This is the basic idea of the FFT algorithm.

Suppose that the cost (the number of multiplications) of calculating $\hat{f}$ is C_N. Then, it follows that

$$C_N = 2C_{N/2} + N/2 \qquad (5.2.8).$$

By induction, it can be seen that

$$C_N = \frac{N}{2} \log_2 N.$$

In fact, by a direct calculation, we know that $C_{2^m} = 1$ for m = 1. Suppose it is true for m, that is, $C_N = \frac{N}{2} \log_2 N$ where $N = 2^m$. Then, for m+1, using (5.2.8), it follows

$$C_{2N} = 2C_N + N = 2\frac{N}{2} \log_2 N + N$$

$$= \frac{(2N)}{2} \log_2(2N).$$

That is, $C_N = \frac{N}{2} \log_2 N$ is also true for $N = 2^{m+1}$. By induction, it is true for all $m \geq 1$. Therefore, the cost of FFT algorithm is $\frac{N}{2} \log_2 N$.

From the discussion above, we see that the key step in FFT method is to decompose the DFT of a signal into two components, "even component" and "odd component". By exploiting features of G and H, the DFT of f can be computed with the cost of $\frac{N}{2}\log_2 N$ multiplications instead of N^2 multiplications, which is the cost for a direct matrix and vector multiplication.

Example 2. Haar Transform

Let G and H be two $(N/2) \times N$ matrices with entries

$$G = (g_{kl}) = \begin{cases} \frac{1}{\sqrt{2}} & \text{if l = 2k, 2k+1, k=0, 1, 2,...,N/2-1} \\ 0 & \text{otherwise} \end{cases} \tag{5.2.9}$$

and

$$H = (h_{kl}) = \begin{cases} \frac{1}{\sqrt{2}} & \text{if l = 2k where k=0, 1, 2,...,N/2-1} \\ -\frac{1}{\sqrt{2}} & \text{if l = 2k+1 where k=0, 1, 2,...,N/2-1} \\ 0 & \text{otherwise.} \end{cases} \tag{5.2.10}$$

In the form of matrix,

$$G = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 & 0 & 0 & \ldots & 0 & 0 \\ 0 & 0 & 1 & 1 & \ldots & 0 & 0 \\ & & & & \ldots & & \\ 0 & 0 & 0 & 0 & \ldots & 1 & 1 \end{bmatrix}$$

and

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & -1 & 0 & 0 & \ldots & 0 & 0 \\ 0 & 0 & 1 & -1 & \ldots & 0 & 0 \\ & & & & \ldots & & \\ 0 & 0 & 0 & 0 & \ldots & 1 & -1 \end{bmatrix}$$

The matrices G and H defined in (5.2.9) and (5.2.10) are referred to as a Haar matrix pair. $\begin{pmatrix} Gf \\ Hf \end{pmatrix}$ is called the Haar transform of signal f. Gf is referred to as the lowpass component and Hf is referred to as the highpass component of the Haar transform of f. Since matrix $\begin{pmatrix} G \\ H \end{pmatrix}$ has dimension $N \times N$, it follows that the Haar transforms of all N-dimensional vectors also constitute an N-dimensional vector space, which is called a Haar transform space.

We observe some properties of the Haar transform matrix $\begin{pmatrix} G \\ H \end{pmatrix}$. First, it can be verified by a straightforward calculation that G and H satisfy the following equations:

$$GG^T = I \tag{5.2.11}$$

$$HH^T = I \tag{5.2.12}$$

$$GH^T = HG^T = 0 \tag{5.2.13}$$

$$G^TG + H^TH = I. \tag{5.2.14}$$

where G^T and H^T denote the transpose matrices of G and H respectively. These equations are equivalent to the statement that the matrix $\begin{pmatrix} G \\ H \end{pmatrix}$ is unitary. By (5.2.14), a signal f can be reconstructed from its Haar transform by applying (G^T, H^T) to $\begin{pmatrix} Gf \\ Hf \end{pmatrix}$.

Second, observe that the Haar matrices G and H have only a few non-zero entries which appear consecutively in each row, while there is a non-zero entry in every two consecutive entries in each row of the Fourier matrices G and H defined in (5.2.4) and (5.2.5). This means that for each entry in a signal, the Haar transform only takes its neighboring entries into account but the Fourier transform takes the entries spread over the whole signal into account. The Haar transform is said to be localized since it possesses such a feature. This is a significant difference between Fourier transform and a wavelet transform such as the Haar transform.

Third, observe that the lowpass component Gf of the Haar transform of the signal f has $\frac{N}{2}$ entries. Assume $N = 2^m, m \geq 2$ where m is a positive integer. Define

$$G_1 = (g_{kl}) = \begin{cases} \frac{1}{\sqrt{2}} & \text{if l = 2k, 2k+1, k=0, 1, 2,...,N/4-1} \\ 0 & \text{otherwise} \end{cases}$$

and

$$H_1 = (h_{kl}) = \begin{cases} \frac{1}{\sqrt{2}} & \text{if l = 2k where k=0, 1, 2,...,N/4-1} \\ -\frac{1}{\sqrt{2}} & \text{if l = 2k+1 where k=0, 1, 2,...,N/4-1} \\ 0 & \text{otherwise.} \end{cases}$$

By applying $\begin{pmatrix} G_1 \\ H_1 \end{pmatrix}$ to the lowpass component Gf, we obtain the Haar transform of Gf, with the lowpass component G_1Gf and the highpass component H_1Gf. This process can be continued until the lowpass component contains only one entry. Thus, the signal f can be decomposed into a sequence of components. This is an example of discrete wavelet transform, which will be discussed in the following section.

Fourth, we consider the computational cost of the Haar transform. For a signal with length $N = 2^m$ where $m > 0$, we can repeat the decomposition process upto m steps. For each step, we only need $2^i, 0 < i \leq m$ multiplications (multiplying $\frac{1}{\sqrt{2}}$ to each component obtained by addition and subtraction operations). So the total cost is bounded by

$$2^m + 2^{m-1} + ... + 2 = 2(2^m - 1) = 2N - 2.$$

That is, the cost of Haar transform is O(N).

To conclude the discussions for the Haar transform, we make some comparisons between the Haar transform and the preceding fast Fourier transform. Both transforms generate a sequence of coefficients from which the original signal can be reconstructed, the DFT coefficients of the signal for the FFT and the Haar wavelet coefficients of the signal for the Haar transform. These two types of coefficients may be exploited for different applications. For the FFT, if the DFT coefficients with lower indices contain much more information about the signals than the coefficients with higher indices, one can preserve most of the energy in the original signal by just keeping the coefficients with lower indices. On the other hand, as discussed above, the Haar transform is localized and the FFT is globalized. If the original signal is contaminated by noise, the noise will only spread to the neighboring locations after performing Haar transform but the noise will spread over all locations after performing FFT. Second, both transforms are fast: $O(N)$ operations for the Haar transform and $O(N \log_2 N)$ operations for the FFT. The fact that both the FFT and the Haar transform are fast makes these transforms useful in practice.

Example 3. Symmetric Haar Transform

We give a pair of matrices G and H, which are similar to those in (5.2.6) and (5.2.7). Let G and H be two $N \times N$ circulant matrices with entries given by

$$G = \frac{1}{2}\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & \ldots & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & \ldots & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & \ldots & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & \ldots & 0 & 0 \\ & & & & & \ldots & & \\ 1 & 0 & 0 & 0 & 0 & \ldots & 1 & 0 \end{bmatrix} \tag{5.2.15}$$

and

$$H = \frac{1}{2}\begin{bmatrix} 0 & -1 & 0 & 0 & 0 & \ldots & 0 & 1 \\ 1 & 0 & -1 & 0 & 0 & \ldots & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & \ldots & 0 & 0 \\ 0 & 0 & 1 & 0 & -1 & \ldots & 0 & 0 \\ & & & & & \ldots & & \\ -1 & 0 & 0 & 0 & 0 & \ldots & 1 & 0 \end{bmatrix} \tag{5.2.16}$$

Observe that G is symmetric and H is skew symmetric. The matrices G and H defined in (5.2.15) and (5.2.16) are referred to as a symmetric Haar matrix pair. Similar to the matrix pair given in Example 2, the symmetric Haar matrix pair is also localized. For signal f, the lowpass component Gf can be further decomposed into GGf and HGf and so forth. Since

$$GG^T = \frac{1}{4}\begin{bmatrix} 2 & 0 & 1 & 0 & 0 & \ldots & 1 & 0 \\ 0 & 2 & 0 & 1 & 0 & \ldots & 0 & 1 \\ 1 & 0 & 2 & 0 & 1 & \ldots & 0 & 0 \\ 0 & 1 & 0 & 2 & 0 & \ldots & 0 & 0 \\ & & & & & \ldots & & \\ 0 & 1 & 0 & 0 & 0 & \ldots & 0 & 2 \end{bmatrix}$$

and

$$HH^T = \frac{1}{4}\begin{bmatrix} 2 & 0 & -1 & 0 & 0 & \ldots & -1 & 0 \\ 0 & 2 & 0 & -1 & 0 & \ldots & 0 & -1 \\ -1 & 0 & 2 & 0 & -1 & \ldots & 0 & 0 \\ 0 & -1 & 0 & 2 & 0 & \ldots & 0 & 0 \\ & & & & & \ldots & & \\ 0 & -1 & 0 & 0 & 0 & \ldots & 0 & 2 \end{bmatrix},$$

it follows

$$G^T G + H^T H = I.$$

Therefore, the signal f can be reconstructed by

$$f = G^T u + H^T v.$$

But equations. (5.2.11) through (5.2.13) do not hold for this symmetric Haar matrix pair. For example,

$$HG^T = \frac{1}{4}\begin{bmatrix} 0 & 0 & -1 & 0 & 0 & \ldots & 1 & 0 \\ 0 & 0 & 0 & -1 & 0 & \ldots & 0 & 1 \\ 1 & 0 & 0 & 0 & -1 & \ldots & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & \ldots & 0 & 0 \\ & & & & & \ldots & & \\ 0 & -1 & 0 & 0 & 0 & \ldots & 1 & 0 \end{bmatrix}.$$

Another difference between the symmetric Haar matrices and the non-symmetric Haar matrices is the dimensions of the matrices: the matrices G and H of the symmetric matrix pair each have dimension of $N \times N$ but the matrices G and H of the nonsymmetric matrix pair each have dimension of $N/2 \times N$.

We have seen some similarities and some differences between the symmetric Haar matrix pair and the nonsymmetric Haar matrix pair. The symmetric Haar matrix pair is used in the numerical experiments conducted in this study even though the nonsymmetric Haar matrix pair is a classic example in wavelet theory. Next section makes further comparisons for these two matrix pairs and gives the reason for why we select the symmetric Haar matrix pair rather than the nonsymmetric pair in our numerical experiments.

5.3 Symmetric and Nonsymmetric Haar Wavelets

5.3.1 Discrete Wavelets. Let $G_i, H_i, i = 1, 2, ...$ be $N_i \times N_{i-1}$ matrices, where $N_i, i = 0, 1, 2, ...$ are positive integers. If there are $N_{i-1} \times N_i$ matrices G_i^+ and H_i^+ associated with G_i and H_i satisfying

$$G_i^+ G_i + H_i^+ H_i = I_i, \tag{5.3.1}$$

where I_i is an $N_{i-1} \times N_{i-1}$ identity matrix, G_i and H_i is called a discrete wavelet pair. The wavelet pair is said to be orthonormal if $G_i^+ = G_i^T$, $H_i^+ = H_i^T$ and

$$G_i G_i^T = I_i \tag{5.3.2}$$

$$H_i H_i^T = I_i \tag{5.3.3}$$

$$G_i H_i^T = H_i G_i^T = 0. \tag{5.3.4}$$

For a signal $f = (f_1, f_2, ..., f_N)$ and discrete wavelet pairs G_i and H_i with $N_0 = N$, where i=1,2,...,J and J is a positive integer. we define the collection of signals

$$(G_J G_{J-1} ... G_1 f, H_J G_{J-1} ... G_1 f, ..., H_3 G_2 G_1 f, H_2 G_1 f, H_1 f)$$

the (generalized) discrete wavelet transform up to scale J. $G_J G_{J-1} ... G_1 f$ is called the smooth part of the (generalized) wavelet transform and $H_j G_{j-1} ... G_1 f$, $1 \geq j \leq J$, is called the (generalized) wavelet transform at scale j.

Remark I: We defined wavelet transform in a very general form. This definition includes dyadic wavelet transform, where N_i is assumed to be $N_{i-1}/2$ (You can see an example of such transform in the next subsection). If no confusion occurs, we omit the term "generalized" in the later discussions.

Remark II: The orthonormal condition (5.3.2) through (5.3.4) seems somewhat unnatural. But this orthonormality condition can be naturally derived from the theory of multiresolution analysis, where "orthogonality" is defined in a conventional sense in a Hilbert space. For the details of the derivation, we refer the reader to [LIN94].

5.3.2 Nonsymmetric Haar Wavelets. Suppose $N = 2^m$ where m is a positive integer. Let $N_0 = N$ and $N_i = N_{i-1}/2$. Let G_i and H_i, where i=1,2,...,m, be two $N_i \times N_{i-1}$ matrices with entries

$$G_i(k,l) = \begin{cases} \frac{1}{\sqrt{2}} & \text{if l = 2k, 2k+1, k=0, 1, 2,...,Ni-1} \\ 0 & \text{otherwise} \end{cases}$$

and

$$H_i(k,l) = \begin{cases} \frac{1}{\sqrt{2}} & \text{if l = 2k, k=0, 1, 2,...,Ni-1} \\ -\frac{1}{\sqrt{2}} & \text{if l = 2k+1, k=0, 1, 2,...,Ni-1} \\ 0 & \text{otherwise.} \end{cases}$$

From the discussions in the previous section, we see that G_i, H_i, G_i^T and H_i^T satisfy

$$G_i^T G_i + H_i^T H_i = I_i,$$

where I_i is a $N_i \times N_i$ identity matrix, and G_i^T and H_i^T denotes the matrix transposes of G_i and H_i respectively. Therefore, G_i's and H_i's are wavelet pairs. They are referred to as the nonsymmetric Haar wavelet pair. We also saw in the previous section that G_i, H_i, G_i^T and H_i^T satisfy

$$G_i G_i^T = I_i$$

$$H_i H_i^T = I_i$$

$$G_i H_i^T = H_i G_i^T = 0.$$

It means that the nonsymmetric Haar wavelet is an orthogonal wavelet.

5.3.3 Symmetric Haar Wavelets. Suppose $N = 2^m$ where m is a positive integer. Let G and H be two $N \times N$ circulant matrices given by (5.2.15) and (5.2.16). Let $G_i = G$, $H_i = H$, $G_i^+ = G^T$ and $H_i^+ = H^T$ for i =1,2, ...,m. Then, it was seen in the previous section that

$$G_i^T G_i + H_i^T H_i = I$$

for each i =1,2,...m, where I is a $N \times N$ identity matrix. Therefore, each G_i and H_i is a wavelet pair. They are referred to as a symmetric Haar wavelet pair.

In the preceding example, each wavelet pair is identical. Next, we give another type of symmetric Haar wavelets where each pair is differed from each other but has similar form. To describe such matrices, we need to define circulant matrix in a rigorous way.

Let Z denote the collection of all integers and N denote a positive number. A sequence $s = \{s_i\}_{i \in Z}$ is said to be N-length periodic if $s_i = s_{i+N}$ for all $i \in Z$. A $N \times N$ matrix S with entries

$$S_{nl} = s_{l-n}, \qquad n, l = 0, 1, 2, ..., N-1,$$

is called a circulant matrix. $s_i, 0 \leq i < N$ is called the generating sequence.

From the definition above, the symmetric $N \times N$ Haar matrices G and H defined in (5.2.15) and (5.2.16) are generated by

$$g = \frac{1}{2}(0, 1, 0, 0, ..., 0, 1)$$

and

$$h = \frac{1}{2}(0, -1, 0, 0, ..., 0, 1)$$

respectively. Let $N = 2^m$ where m is a positive integer. A sequence of circulant matrices G_i and H_i, $1 \leq i \leq k, k < m$, is generated by

$$g_1 = \frac{1}{2}(0, 1, 0, 0, 0, 0, ..., 0, 1)$$

$$h_1 = \frac{1}{2}(0, -1, 0, 0, 0, 0, ..., 0, 1)$$

$$g_2 = \frac{1}{2}(0, 0, 1, 0, 0, 0, ..., 1, 0)$$

$$h_2 = \frac{1}{2}(0, 0, -1, 0, 0, 0, ..., 1, 0)$$

$$g_3 = \frac{1}{2}(0, 0, 0, 0, 1, 0, ..., 1, 0, 0, 0)$$

$$h_3 = \frac{1}{2}(0, 0, 0, 0, -1, 0, ..., 1, 0, 0, 0)$$

$$...... \tag{5.3.5}$$

In general, g_i has 1/2 at the $2^{i-1}th$ term , 1/2 at the $(2^m - 2^{i-1})th$ term and 0 elsewhere. h_i has -1/2 at the $2^{i-1}th$ term , 1/2 at the $(2^m - 2^{i-1})th$ term and 0 elsewhere. Let g_i^+ be the sequence with 1/2 at the $2^{i-1}th$ term , 1/2 at $(2^m - 2^{i-1})th$ term and 0 elsewhere, h_i^+ be the sequence with 1/2 at the $2^{i-1}th$ term , -1/2 at $(2^m - 2^{i-1})th$ term and 0 elsewhere. Let G_i^+ and H_i^+, $1 \leq i \leq k$, be circulant matrices generated by g_i^+ and h_i^+, $1 \geq i \leq k$. Then, by direct calculation as in Section 5.2, we can show that

$$G_i^T G_i + H_i^T H_i = I.$$

(In Section 6.2, this formula will be shown in an easier way using a property of Fourier transform of circular convolution.) Therefore, G_i and H_i is a wavelet matrix pair. It was found in our numerical experiments that the PET raysum images can be restored better using these wavelet matrix pairs than using the wavelet matrix pairs which are identical for all scales.

5.3.4 Differencing Operator for Edge Detection. It is observed in Section 4.2 that edges in a signal f(x) can be detected by finding modulus maxima of $f'(x)$ and that $f'(x)$ can be implemented by differencing operator

$$\Delta_1 f_i = f_{i+1} - f_i \tag{5.3.6}$$

or

$$\Delta_2 f_i = (f_{i+1} - f_{i-1})/2. \tag{5.3.7}$$

where $f_i = f(i\Delta x)$ and Δx denotes increment. The local modulus maxima of $f'(x)$ are determined by comparing $|\Delta_k f_i|$ with $|\Delta_k f_{i+1}|$ and $|\Delta_k f_{i-1}|$ where k=1 or 2 (see next section.) In this subsection, we discuss how to associate the nonsymmetric Haar wavelet with the differencing operator in (5.3.6) and the symmetric Haar wavelet with the differencing operator in (5.3.7).

Let $f = (f_0, f_1, f_2, ..., f_{N-1})$, where N is a positive even number, denote a signal and let H denote the Haar matrix defined in (5.2.10). The ith entry of u=Hf is $u_i = (Hf)_i = \frac{1}{\sqrt{2}}(f_{2i} - f_{2i+1})$. Here we observe that $|u_i|$ is equal to $|\Delta_1 f_{2i}|$ multiplying a constant $\sqrt{2}$, which is not important since it appears for each entry and can be crossed out when one entry is compared with its neighboring entry.

On the other hand, for the Haar matrix H defined in (5.2.13), the ith entry of u=Hf is $u_i = (Hf)_i = \frac{1}{2}(f_{i-1} - f_{i+1})$. Here we observe that $|u_i|$ is equal to $|\Delta_2 f_i|$. Therefore, both nonsymmetric and symmetric Haar wavelet matrices can be used as differencing operators for edge detection. In the next subsection, we compare these two types of wavelet matrices.

5.3.5 Comparison: Symmetric with Nonsymmetric. In this subsection, we summarize the discussions above and give the reason for why we select the symmetric Haar transform for our signal restoration algorithm.

1. Both the nonsymmetric Haar transform and the symmetric Haar transform can be used for multiscale signal decomposition and reconstruction. Both are localized and both can be easily implemented. Both are associated with differencing operators for edge detection.

2. The nonsymmetric Haar transform is orthogonal while the symmetric Haar transform is non-orthogonal. For signal restoration application, orthogonality can be used to avoid introducing correlation of noise in transformed signal since the noise in the entries of the transformed signal is uncorrelated after an orthogonal transform is performed for an i.i.d. noise. This seems to be an advantage of the nonsymmetric Haar transform over the symmetric Haar transform. But for some noise such as Gaussian noise and a special type of Poisson noise (which is assumed as PET noise), the symmetric Haar transform can also avoid introducing noise

correlation in the transformed signal (the details of how to do it will be seen in the next chapter). Thus, choosing the symmetric Haar transform is not a drawback for the PET image restoration applications.

3. The dimension of the transformed signal space by symmetric Haar transform is 2N; but the dimension of the transformed signal space by non-symmetric Haar transform is N. From the view of saving storage space, this is another advantage of the non-symmetric Haar transform over the symmetric one. This feature is exploited in data compression. For the nonsymmetric Haar transform, since the dimension does not increase from each scale to the next and since a signal can be reconstructed by many combinations of its Haar transforms at different scales, it is anticipated that the cost of coding a signal will be reduced if there exists a combination which has lowest coding cost. This is exploited in lossless data compression and is discussed in the Section 1.7 of [LIN94]. Here, efficient storage is not the principal goal.

4. The symmetric Haar transform has some advantages over the nonsymmetric Haar transform from the view of implementation. For the symmetric Haar transform, since f, Gf, Hf, HGf and GGf have the same dimension, it is easy to compare Hf and HGf for each location. (This feature will be exploited in one of our denoising algorithms, which distinguishes noises from true signals by comparing the wavelet moduli and phases of the noisy edges from one scale to the next.) On the other hand, for the nonsymmetric Haar wavelet pair, edge locations are shifted from scale to scale so that it will be difficult for index manipulation.

In summary, it is our opinion that the symmetric Haar transform is superior to the non-symmetric Haar transforms in signal restoration applications while the non-symmetric Haar transform is superior to the symmetric Haar transform in data compression applications. Since we are mostly concerned with the signal restoration applications in this thesis, we limit our discussions on the symmetric Haar transform from now on. In the following section, we discuss in detail how to restore a signal contaminated by noise using the symmetric Haar wavelet transform.

5.4 Wavelets and Multiscale Edge Detection

In this section, we use the symmetric Haar wavelet to illustrate wavelet application in multiscale edge detection. For the case of using other wavelets and for more general discussions, we refer the reader to [MALL92a], Chapter 8 in [MEYE93] and Section 1.5 in [LIN94].

In Section 4.4, we introduced Marr and Hildreth's approach to multiscale edge detection. They provided a scheme which filtered the original signal via a series of low-pass filters with different cutoff frequencies (scales) and detected edges for each filtered signal. In their approach, a Gaussian-type lowpass filter is used as the lowpass filter and the Laplacian method is used to find edge locations for each filtered signals. They also conjectured that a signal in L^2 can be perfectly reconstructed from the information of its multiscale edge locations at all scales. But this fascinating conjecture was disproved by Meyer [MEYE93].

According to [MEYE93], Mallat refined the Marr and Hildreth's conjecture. He conjectured that a signal in L^2 can be perfectly reconstructed from the locations and the modulus maxima and the wavelet transforms at different scales together with the values of the wavelet transform coefficients at the modulus maxima locations. This conjecture is also disproved by Meyer [MEYE93]. In spite Meyer's counterexamples, however, many experiments indicate that the reconstruction of a signal from the modulus maxima locations and values gives an excellent approximation to the original signal. The reconstruction technique was suggested by Mallat and Zhong [MALL92a]. Next, we illustrate the reconstruction algorithm using the symmetric Haar wavelets.

For a signal $f = (f_0, f_1, f_2, ..., f_{N-1})$ where $N = 2^m$ and m is a positive integer, the reconstruction algorithm can be seen from following steps:

Step 1. Obtain the discrete wavelet transform of signal f,

$$W_J = (G^J f, HG^{J-1} f, ..., HGf, Hf)$$

where G and H is the symmetric Haar matrix given by (5.2.15) and (5.2.16) and $G^j = G_j G_{j-1} ... G_1$.

Step 2. For each scale j, $0 \leq j \leq J-1$, and and each location i, $1 \leq i \leq N$, determine if the location is an edge (or modulus maximum) location by comparing $HG^j f_i$ with $HG^j f_{i+1}$ and $HG^j f_{i-1}$: the location i is an edge location at scale j if

$$|HG^j f_i| > |HG^j f_{i-1}| \text{ and } |HG^j f_i| \geq |HG^j f_{i+1}|$$

or

$$|HG^j f_i| > |HG^j f_{i+1}| \text{ and } |HG^j f_i| \geq |HG^j f_{i-1}|.$$

Step 3. Obtain the estimate $\hat{HG^j} f_i$ of $HG^j f_i$ for each i and j from the wavelet coefficients at the edge locations obtained in Step 2 by using an interpolation technique.

Step 4. Obtain the estimate $\hat{f}$ of f by taking inverse wavelet transform for $\hat{HG^j} f_i$, which is obtained in Step 3.

The interpolation technique in Step 3 is basically as follows: Let i_0 and i_1 be two consecutive edge locations for scale j. The wavelet coefficient $\hat{HG^j} f_i$, $i_0 < i < i_1$ is obtained by interpolating $HG^j f_{i_0}$ and $HG^j f_{i_1}$. The interpolation function suggested in [MALL92] is

$$\epsilon_j(i) = \alpha e^{2^{-j} i} + \beta e^{-2^{-j} i} \tag{5.4.1}$$

where α and β satisfy

$$\begin{cases} \alpha e^{2^{-j} i_0} + \beta e^{-2^{-j} i_0} = HG^j f_{i_0} \\ \alpha e^{2^{-j} i_1} + \beta e^{-2^{-j} i_1} = HG^j f_{i_1}. \end{cases} \tag{5.4.2}$$

In practice, the signal f and its wavelet transform are unknown and only $\{\hat{HG^j} f_i\}$'s in Step 3 are known and therefore the algorithm to reconstruct a signal from the modulus maxima information of its wavelet transform starts from Step 3. We add Step 1 and Step 2 for the purpose of simulation so that one can see the difference between the original signal f and its estimate $\hat{f}$ by the reconstruction algorithm.

The approximation error between $\hat{f}$ and f is difficult to estimate in an analytical manner. But it can be seen from the preceding algorithm that the error is caused by the interpolation. It is required to minimize the interpolation error in order to minimize the error between $\hat{f}$ and f. Mallat and Zhong selected the interpolation function (5.4.1) by reasoning based on the minimization of the interpolation error in Sobolov norm:

$$\sum_{j=0}^{J-1}(|\epsilon^j|^2 + 2^{2j}|\Delta\epsilon^j|^2)$$

where $|.|$ denotes absolute value, $\epsilon^j = \hat{HG^j}f - HG^jf$ and $\Delta\epsilon^j$ is defined by $\Delta\epsilon^j(i) = \epsilon^j(i) - \epsilon^j(i-1)$. For the details of the argument, we refer the reader to [MALL92a]. In the numerical experiments discussed below, we used Mallat and Zhong's interpolation schemes, since it seems to work well in practice.

5.5 Wavelet Modulus Based Denoising Scheme

One application of the wavelet based multiscale edge detection is in restoring a signal contaminated by noise. This was explored by Mallat and Hwang [MALL92b]. In this section, we discuss the signal restoration application using the symmetric Haar wavelet transform.

First, we extend wavelet transform to random signals. Let $X = (X_1, X_2, ..., X_N)$ denote a random signal where X_i, i=1,2, ... N, is a random variable. For the discrete wavelet matrices G_i and H_i, i=1,2,...,J-1, we define the collection of random signals

$$(G_J G_{J-1}...G_1X, H_J G_{J-1}...G_1X, ..., H_3G_2G_1X, H_2G_1X, H_1X)$$

the (generalized) discrete wavelet transform up to scale J. $G_J G_{J-1}...G_1 X$ is called the smooth part of the (generalized) wavelet transform and $H_j G_{j-1}...G_1 X$, $1 \leq j \leq J$, is called the (generalized) wavelet transform of X at scale j.

Next, we consider wavelet transform of noise. Let $\{\epsilon_i\}$'s, i=1,2,...N, denote *i.i.d.* random variables with mean 0 and variance σ^2. $\epsilon = (\epsilon_1, \epsilon_2, ..., \epsilon_N)$ is called an *i.i.d.* noise. One important observation about *i.i.d.* noise is that the variance (or second moment, since the noise has zero mean) of its Haar wavelet transform coefficients at scale j is $O(2^j \sigma^2)$. The argument is as follows: For the symmetric Haar matrix pair G and H given by (5.2.15) and (5.2.16), the variance of each entry in $H\epsilon$ is

$$\begin{aligned} V((H\epsilon)_i) &= V(\frac{1}{2}(\epsilon_{i-1} - \epsilon_{i+1})) \\ &= \frac{1}{4}(\sigma^2 + \sigma^2) \\ &= \frac{1}{2}\sigma^2. \end{aligned}$$

The variance of each entry in $HG\epsilon$ is

$$\begin{aligned} V((HG\epsilon)_i) &= V(\frac{1}{2}(\frac{\epsilon_{i-2} + \epsilon_i}{2} - \frac{\epsilon_i + \epsilon_{i+2}}{2})) \\ &= V(\frac{1}{4}(\epsilon_{i-2} - \epsilon_{i+2})) \\ &= \frac{1}{16}(\sigma^2 + \sigma^2) \\ &= \frac{1}{8}\sigma^2. \end{aligned}$$

Generally, we can show by induction that $(G^j\epsilon)_i$ contains at most 2^j terms of $\{\epsilon_i\}$'s and each term is multiplied by $\pm 2^{-j}$. Since $(G\epsilon)_i = \frac{1}{2}(\epsilon_{i+1} + \epsilon_{i-1})$, it is true for j=1. Suppose that it is true for scale j-1. That is, $(G^j\epsilon)_k$ contains at most 2^{j-1} terms of $\{\epsilon_i\}$'s and each term is multiplied by $\pm 2^{-j+1}$. Since

$$(G^j\epsilon)_i = 1/2[(G^{j-1}\epsilon)_{i-1} + (G^{j-1}\epsilon)_{i+1}],$$

it follows that $(G^j\epsilon)_i$ contains at most 2^j terms of $\{\epsilon_i\}$'s and each term is multiplied by $\pm 2^{-j}$. It is also true for j. Since

$$(HG^j\epsilon)_i = 1/2[(G^j\epsilon)_{i-1} - (G^j\epsilon)_{i+1}],$$

$(HG^j\epsilon)_i$ contains at most 2^{j+1} terms of $\{\epsilon_i\}$'s and each term is multiplied by $\pm 2^{-(j+1)}$. Therefore, the variance of each entry in $HG^j\epsilon$

$$V((HG^j\epsilon)_i) \leq (\frac{1}{2^{j+1}})^2(2^{j+1}\sigma^2)$$

$$= \frac{1}{2^{j+1}}\sigma^2$$

On the other hand, Mallat and Hwang [MALL92b] observed that the modulus maxima of the wavelet transform of a true signal will not decay as fast as that of a noise. We will not discuss their general results at length but give an example to illustrate the basic idea. For simplicity, we assume the signal is not a random signal but a deterministic signal. Also, periodical extension is employed at the boundary of the signal.

Example: Let f=(0, 0, 0, 0, 0.5, 1, 1, 1, 1) denote a signal with length 9. We want to use this short signal to illustrate how the modulus of the symmetric Haar wavelet transform changes from scale to scale. By a direct calculation, we obtain

$$Hf = (-0.5, 0, 0, 0.25, 0.5, 0.25, 0, 0, -0.5)$$

$$HGf = (-0.25, -0.25, 0.125, 0.25, 0.25, 0.25, 0.125, -0.25, -0.25)$$

$$HGGf = (-0.25, -0.0625, 0, 0.1825, 0.25, 0.1825, 0, -0.0625, -0.25)$$

$$HGGGf = (-0.15625, -0.125, 0.125, 0.125, 0.188, 0.125, 0.125, -0.125, -0.15625)$$

From the above calculation, it is seen that modulus maxima are 0.5, 0.25, 0.25, 0.188 for scale 1, 2, 3, 4 which are from Hf, HGf, HGGf and HGGGf respectively. The squares of the modulus maxima are 0.25, 0.0625, 0.0625, 0.035344. Suppose that the signal is contaminated by noise with variance 0.1. It follows from the discussion above that the variances (mean squares, since zero mean is assumed for the noise) of the wavelet transform of the noise is bounded above by 0.1/2=0.05, 0.1/4=0.025, 0.1/8=0.0125, 0.1/16=0.00625 for scale 1, 2, 3, 4. Here we observe the following facts: First, the mean square of the wavelet transform of noise is less than that of the signal at the modulus maximum location; Second, the mean square of the wavelet transform of the noise decays slightly faster than that of the signal at the modulus maximum location from scale 1 to scale 4 (note that the mean square of the wavelet transform of noise at scale 4 is 8 times less than that at scale 1, while the square of the wavelet modulus maximum of the signal at scale 4 is 7.07 times less than that at scale 1). Also observe that the mean square of the wavelet transform of noise at scale 3 is 2 times less than that at scale 2, while the square of the wavelet modulus maximum of the signal at scale 3 is equal to that at scale 2. These are the main differences between a noise and a true signal in their mean squares of wavelet transforms. These differences are exploited to distinguish noise from true signal in the wavelet modulus based denoising algorithm in [MALL92b]. The example indicates that the differences need not be as pronounced as one might wish. The algorithm can be summarized as follows:

Step 1. Take discrete wavelet transforms for the signal, which is contaminated by noise (This step is the same as the Step 1 in the wavelet based multiscale edge detection algorithm in the previous section).

Step 2. Find edge locations using the techniques in the Step 2 of the algorithm in the previous section.

Step 3. Removing those edges whose moduli are either relatively smaller than some given thresholds or decay faster than some given ratios.

Step 4. Obtain wavelet transforms from the information of the remaining edges using the Step 3 in the previous section .

Step 5. Reconstruct signal from the wavelet transforms obtained in Step 4 using inverse wavelet transform.

Observe that the algorithm is sensitive to the variance level of noise. Let us go back to the example given above. Suppose the variance of the noise is 0.5 instead of 0.1. Then the upper bounds of the mean squares of the wavelet transforms of the noise for the first 4 scales are 0.25, 0.125, 0.0625 and 0.03125. Observe that these values are not far from 0.25, 0.0625, 0.0625, 0.035344, which are the mean squares of the modulus maxima of the signal at the first 4 scales. Here we see that the wavelet modulus maxima based denoising scheme will not perform effectively if the variance of noise is high, which is exactly the case for PET images. Also observe that the decay of the wavelet modulus maxima of noise is not so significant from that of the signal for the first 4 scales (8 *vs.* 7.07 in the preceding example). Therefore, This example also suggests that a wavelet based denoising algorithm will not be very effective if it uses modulus information alone.

5.6 Concluding Remarks

In summary, we illustrated how to use wavelets to restore signals contaminated by noise. The wavelet method differs from a conventional lowpass filtering method in that the wavelet method does throw away all high frequency components but distinguishes noise from signal according to their wavelet transform properties. The symmetric Haar wavelet has been seen to be an useful wavelet in such application. To make our arguments easier to be grasped, we limited our discussions to the one-dimensional case. It is desirable to extend such discussions to the two-dimensional case, since an image is usually modeled as a two-dimensional array. This will be discussed in the beginning of the next chapter.

In essence, detecting edges by using modulus maxima of wavelet transforms is a method based on gradient magnitudes. As pointed in the previous chapter, it is sensitive to the variance of noise. Gregson's angular method improved the detection by using gradient angle information. This suggests that we may also improve denoising algorithm by incorporating the phase information of wavelet transforms. We will exploit this possibility in the next chapter.