UNIVERSITY OF CALIFORNIA

SANTA CRUZ

MEASURING SPATIALLY VARYING BLUR AND ITS APPLICATION IN DIGITAL IMAGE RESTORATION

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

DOCTOR OF PHILOSOPHY

in

ELECTRICAL ENGINEERING

by

Xiang Zhu

June 2013

The Dissertation of Xiang Zhu
is approved:

Professor Peyman Milanfar, Chair

Professor Alex Pang

Professor Claire Gu

Professor Alyson Fletcher

Tyrus Miller
Vice Provost and Dean of Graduate Studies

Table of Contents

List of Figures

List of Tables

Abstract

Measuring Spatially Varying Blur and its Application in Digital Image Restoration

by

Xiang Zhu

While digital imaging systems have been widely used for many applications including consumer photography, microscopy, aerial photography, astronomical imaging, etc., their output images/videos often suffer from spatially varying blur caused by lens, transmission medium, post processing algorithms, and camera/object motion. Measuring the amount of blur globally and locally is an important issue. It can help us in removing the spatially varying blur, and enhancing the visual quality of the imaging system outputs. It can also provide useful information about the scene, such as saliency and depth map.

In this work, we study the blur measurement problem for different scenarios. We first analyze the behavior of a local gradient-based sharpness metric in the presence of spatially varying blur and spatially constant noise, and develop two multiframe restoration systems based on this metric. The first one is a focus-stacking method developed for multifocus image sequences. It selects sharp local patches from the input sequence through the metric, and fuses them together to generate a high quality image. Different from existing stacking methods, this one utilizes image color information to correct fusion errors caused by inaccurate sharpness measurement. The second system aims to restore videos distorted by air turbulence. Air turbulence produces random blur and geometric deformation. Our system first reduces the space and time varying

deblurring problem to a shift invariant one through a registration and fusion procedure, and then removes the blur using a blind deconvolution step. Experiments illustrate that this approach can effectively alleviate turbulence distortions, and recover details of the scene.

We then consider the situation where both blur and noise are spatially changing. A quality metric based on singular value decomposition of local image gradient matrix is proposed. Compared with existing sharpness metrics that cannot distinguish quality decay against noise, the proposed one is properly correlated with the noise level and blurriness of a given image. Ample simulated and real data experiments illustrate that this metric is capable of capturing the trend of quality change during the denoising process, and can be used to automatically select the denoising filter parameters that show excellent visual performance in balancing between noise suppression and detail preservation.

Finally, we propose a method capable of estimating the accurate amount of local defocus blur from a single image. This method measures the probability of local defocus level. It also takes smoothness and color edge information into consideration to generate a coherent map indicating the amount of blur at each pixel. Real data experiments illustrate its good performance, and its successful applications in foreground/background segmentation.

To my family and my girlfriend.

Acknowledgments

Above all, I would like to express my deepest gratitude to my advisor, Prof. Peyman Milanfar, for his guidance, caring, patience, encouragement, and providing me and all his students with an excellent atmosphere for doing research. I feel very lucky to have an advisor like him who cares so much about not only my academic work but also my personal development. I really enjoyed my five-year time in his MDSP lab in UC Santa Cruz. I also want to thank my other dissertation committee members, Prof. Alex Pang, Prof. Claire Gu and Prof. Alyson Fletcher for their invaluable advice.

My sincere thanks goes to Dr. Scott Cohen and Mr. Stephen Schiller for being my mentors during the summer internship at Adobe Systems in 2011, where I started the research on estimating local defocus blur from a single image. Without their help and contributions to this project I cannot get the estimation algorithm with state-of-the-art performance given in Chapter 4.

Many thanks to the researchers who share the precious video data about air turbulence illustrated in Chapter 3. I thank Prof. Mikhail A. Vorontsov from the Intelligent Optics Lab of the University of Maryland for allowing our lab to use the video *Water Tower*, and thank Mr. Faisal A. Salem from University of Michigan and Dr. Joseph M. Zawodny from NASA Langley Research Center for providing us with the video *Moon Surface*. I also thank Mr. M. Hirsch and Dr. S. Harmeling from Max Plank Institute for Biological Cybernetics for sharing the sequences *Chimney* and *Building*.

I am thankful for the former members of MDSP lab, Dr. Hiroyuki Takeda, Dr. Hae-Jong Seo, Dr. Priyam Chatterjee, Dr. Filip Šroubek, and the current students, Hossein, Amin, Sujoy, Rob and Dara. They are excellent researchers and great friends.

Hopefully I can still meet with them every year in Peyman's annual BBQ party after my graduation.

Finally, I would like to thank my parents and my girlfriend Tianying. Without their support and encouragement I cannot concentrate on my work and overcome the difficulties.

Santa Cruz, California

June 9th, 2013

Xiang Zhu

Chapter 1

Introduction

Abstract - In this chapter we discuss various types of image blur sources and their properties. Different degradation models are introduced to represent both constant and spatially varying blurring process. Importance of image blur measurement and its potential applications are explained. We also discuss limitations of most existing blur estimation algorithms and sharpness metrics, and give a brief description about the contributions of this thesis in the area of spatially varying blur measurement.

1.1 Blur in Imaging Systems

Digital imaging devices along with post processing algorithms are very popular in many imaging areas, including consumer photography, microscopy, macro photography, aerial photography, astronomical imaging, medical imaging, etc. However, all these imaging systems suffer from two common distortions, which are blur and noise. Compared with noise, which is mainly caused by the sensor and circuitry of a digital camera and could be approximately described through some standard statistical models (e.g.

Gaussian distribution, Poisson distribution), blur has more sources and its form can be highly complicated. How to measure and remove various kinds of blur along with noise is a significant problem not only in the image/video restoration area but also in many other applications in the fields of image processing, computer vision, and computational photography.

1.1.1 Sources of Blur

According to its sources, image blur can be generally categorized into four groups: motion blur, lens blur, blur due to transmission medium (e.g. turbulence), and post processing blur (see examples illustrated in Fig. **1.1**).

Either camera or object movement during the exposure period would lead to motion blur. This phenomenon is very common especially for consumer digital cameras. For example, cell phone cameras often cannot be held sufficiently steady, and thus it is easy to generate camera shaking blur. Fast exposure could reduce the blur amount to some degree. However, because the aperture size is also relatively small for mobile cameras, shorter exposure time would introduce more noise. This problem can be prominent if the scene is dim, which means we have to maintain a relatively long exposure time and sacrifice image sharpness to suppress noise level. Ideally, if the scene is static with uniform depth and if the camera motion is 2-D translational, then the motion blur can be viewed as spatially invariant (see Fig. **1.1** (a)), which could be estimated and removed through a *blind deconvolution* procedure [7, 8, 9]. However, in practice camera motion includes more complex motion. Objects inside the scene can also be moving. Besides, the depth is probably spatially changing. Any of these three

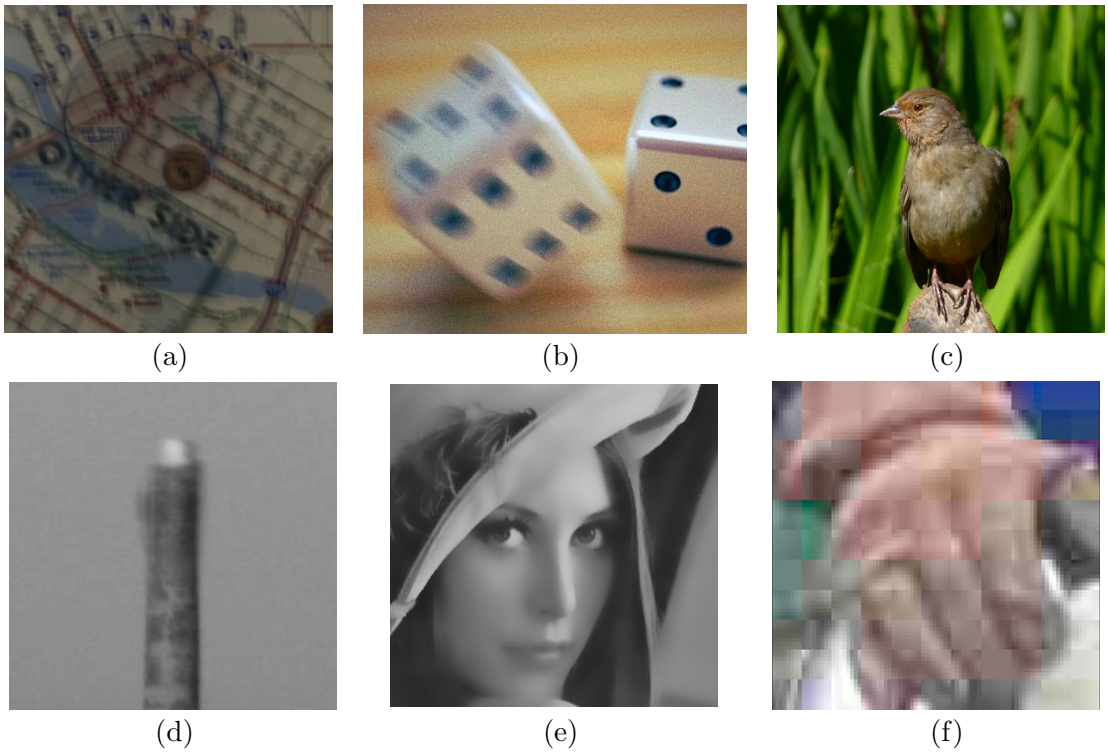


Figure 1.1: Examples for different types of blur: (a) uniform motion blur caused by camera shaking; (b) non-uniform motion blur caused by object movement; (c) defocus blur in the background; (d) spatially varying blur caused by atmospheric turbulence; (e) an image over-smoothed by Bilateral filtering; (f) JPEG compression artifacts.

factors could make motion blur *spatially varying* (see Fig. **1.1** (b)), which makes its estimation and removal highly difficult [10, 11].

Incorrect lens setting or limited depth of filed would produce defocus blur (see example in Fig. **1.1** (c)), which is an important type of lens blur. Besides, even if the scene is perfectly in focus and no matter how well the lens is corrected, in most optical imaging systems there always exists a fundamental resolution limit due to diffraction, which is called *diffraction-limited blur* [12]. On the other hand, commonly used cameras add an anti-aliasing filter into the lens system to remove high-frequency components beyond the Nyquist limit of the digital camera sensor. Such kind of blur can actually

enhance the visual quality of captured images. Generally speaking, diffraction-limited blur and anti-aliasing blur can be approximately viewed as spatially invariant, while the spatial variance of defocus blur depends on both the depth of field of the lens and depth of the scene [13, 14].

For long-distance imaging systems (such as ground-based astronomical imaging systems) atmospheric turbulence, which generates variation of refractive index along the optical transmission path and distort light wavefront, can strongly affect the image quality [15, 16, 17, 18, 19]. Turbulence of other media (e.g. water) also have similar effects and thus can degrade the performance of related imaging systems (such as underwater cameras) [20]. Generally speaking turbulence in the transmission medium produces two major distortions: geometrical deformation and blur, where both are temporally and spatially changing (see Fig. **1.1** (c)). Removing such effects is very important for many applications and meanwhile it is quite challenging [21, 20, 22, 23, 18].

Post processing methods (denoising, deblurring, super-resolution, etc.) can also introduce blur (see Fig. **1.1** (d)), though most of them are used to enhance image/video quality. One example is denoising filter, which smoothes the input image to reduce noise level but meanwhile produces blur as a side effect. If the filter is spatially invariant, such as a Gaussian smoothing filter, then the blur is also uniform. However, most of the time locally adapted denoising filters are implemented (e.g. Bilateral filter, non-local means (NLM), BM3D) in practice since these filters more effectively suppress noise and meanwhile maintain most of the useful image structure [24, 25, 26, 27, 28]. Blur caused by these filters is again spatially varying. Besides, compression is another procedure which generates non-uniform blur since it kills some high-frequency compo-

nents to reduce the file size (see the example in Fig. **1.1** (e)) [29, 30].

1.1.2 Degradation Model

Though generated by various sources, different kinds of blur can be described by similar models. Let us first consider a shift invariant convolution model, which can be written as:

$$\begin{aligned} g[x,y] &= h[x,y] \otimes f[x,y] + n[x,y] \\ &= \int_{\mathrm{R}^2} h[x-u, y-v] f[u,v] du dv + n \end{aligned} \tag{1.1}$$

where $g[x,y]$, $f[x,y]$ denote the observed blurry image and the latent sharp image, respectively. $h[x,y]$ represents the point-spread-function (PSF) of blur, $\otimes$ denotes 2-D shift-invariant convolution, and n is additive noise.

This model can be used to describe spatially uniform blur, where the same $h[x,y]$ is applied over the whole image. For example, in the camera motion blur case as we explained earlier that if the scene is static and if the camera motion is 2D translational, then the blur PSF can be viewed as spatially invariant (see an example in Fig. **1.2**). Besides, diffraction-limited blur and anti-aliasing filter blur can also be treated as spatially invariant.

Under some circumstances we may have multiple frames taken for the same scene, where each frame has a unique PSF [31]. The corresponding imaging model becomes:

$$g_k[x,y] = h_k[x,y] \otimes f[x,y] + n_k, \quad (k = 1, 2, \ldots, K) \tag{1.2}$$

here K denotes the number of frames and each frame is indexed by k. In this model it is assumed that the scene is static across the frame sequence, and the PSF is varying

Figure 1.2: A camera motion blurred image example from [1], where trajectories of the light spots reveal the shape of the local motion PSFs. We can see that the PSFs are basically invariant across the image region.

between frames, but within one frame $h_k[x, y]$ is still spatially invariant. Such situation may happen when we have a burst of images and each one is distorted by different motion blur [32, 33, 34, 35].

If blur PSF is spatially changing, which is more common in practice, a more general image model can be used:

$$g[x, y] = \int_{\mathrm{R}^2} h[x - u, y - v; u, v] f[u, v] du dv + n \tag{1.3}$$

where $h[x, y; u, v]$ represents the PSF of the point located at $[u, v]$. It can be seen that this is no longer a shift-invariant convolution, and almost all the spatially changing blur cases in Sec 1.1.1 can be represented using this model. In discrete imaging systems, this model can also be written as:

$$\mathbf{g} = \mathbf{H}\mathbf{f} + \mathbf{n} \tag{1.4}$$

where $\mathbf{g}$, $\mathbf{f}$ and $\mathbf{n}$ denote observed digital image, latent sharp image and noise in vector forms, and $\mathbf{H}$ represents a blurring matrix determined by $h[x, y; u, v]$. Note that $\mathbf{H}$ can also be generated from a space-invariant PSF $h[x, y]$.

For multi-frame cases, the above model becomes:

$$\mathbf{g}_k = \mathbf{H}_k\mathbf{f} + \mathbf{n}_k \quad (k = 1, 2, \ldots, K) \tag{1.5}$$

Again, each frame is indexed by k, and K denotes the total number of frames. $\mathbf{H}_k$ represents a spatially and temporally changing blurring matrix. This formula is useful in modeling videos and image sequences distorted by atmospheric turbulence [36].

For some types of blur (such as lens blur) their PSF can be approximated through a parametric model (e.g. disc, Gaussian) and characterized by a single parameter indicating its scale (radius, standard deviation, etc.). For other situations (e.g. motion blur, turbulence blur) the PSF can have a somewhat arbitrary form with a high degree of freedom, which makes its measurement quite challenging.

1.2 Measuring Image Blur

Measuring blur is very important in the fields of image processing and computer vision. For example, it can help us in recovering the latent sharp images (deblurring) or enhancing visual quality (sharpening). In this thesis the term *image blur measurement* includes two kinds of procedures: *PSF estimation* and *sharpness measurement*. The first one tells about the accurate blur form, and the second one measures the effect of blur.

1.2.1 Importance of Blur Measurement

The most direct way of measuring blur is to estimate its PSF. Ideally, once the blur matrix $\mathbf{H}$ is estimated, we can then calculate the latent image $\mathbf{f}$ through a

non-blind deconvolution[1] algorithm such as:

$$\hat{\mathbf{f}} = \arg\min_{\mathbf{f}} \|\mathbf{g} - \mathbf{H}\mathbf{f}\|^2 + \lambda R(\mathbf{f}) \tag{1.6}$$

where term $R(\mathbf{f})$ represents a prior constraint about the latent image $\mathbf{f}$. Usually a sparsity term is employed here [37, 38].

Even without knowing the PSF matrix $\mathbf{H}$, as long as the degree of blur, or say the sharpness of local image content is available, we can still use it to improve the image visual quality. For example, local sharpness can be used to adaptively adjust the strength of a sharpening filter. In practice, the following unsharp-masking filter is often used in image sharpness enhancement:

$$\mathbf{g}_{sharp} = \mathbf{g} + \lambda(\mathbf{I} - \mathbf{B})\mathbf{g} \tag{1.7}$$

where $\mathbf{B}$ is a low-pass filtering matrix, and $\mathbf{I}$ denotes Identity. So $\mathbf{I} - \mathbf{B}$ basically represents a high-pass filter. The high-frequency components $(\mathbf{I} - \mathbf{B})\mathbf{g}$ are scaled by a constant factor λ and are added back to the input image. Compared with deconvolution that usually requires an accurate PSF estimate and an iteration-based optimization procedure, unsharp-masking provides a fast way to enhance details. This method works reasonably well with spatially invariant and mild defocus blur. However, when implementing it on an image where part of it is in-focus and the rest out-of-focus, it fails. The in-focus region would contain unwanted over-sharpening artifacts (e.g. ringing, halo). An example is given in Fig. **1.3**. In this case, if local sharpness level can be measured, we can adjust λ accordingly so that it has small value in sharp regions (to avoid artifact) and large value in blurry regions (to enhance sharpness).

[1] If both the observed blurred image and the PSF are available, the latent image restoration is referred as "non-blind". If only the observed image is known, then the latent image restoration is referred as "blind".

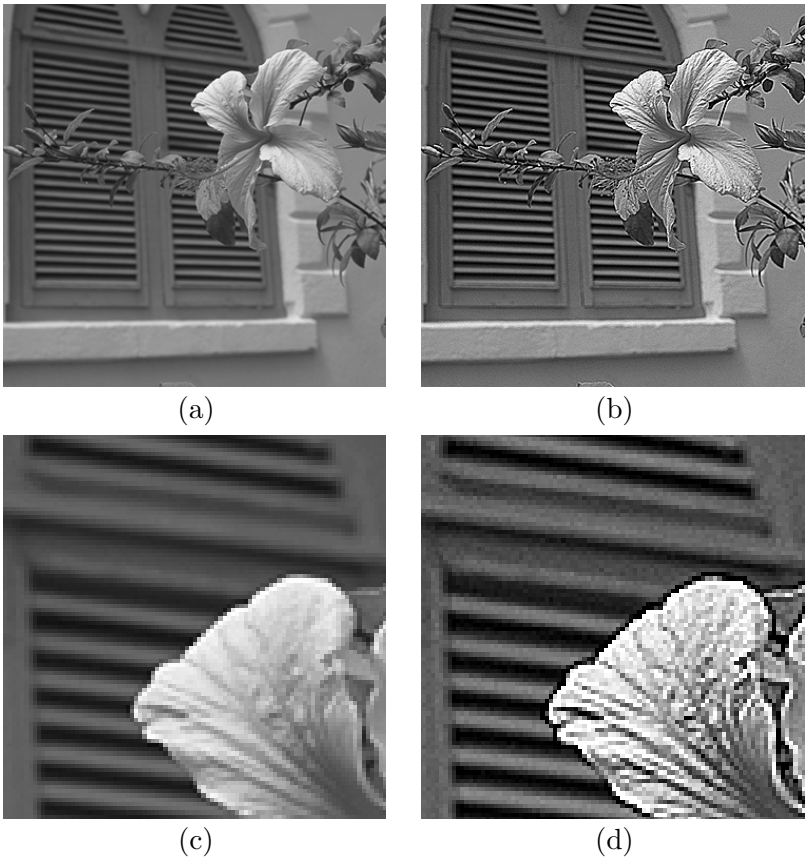
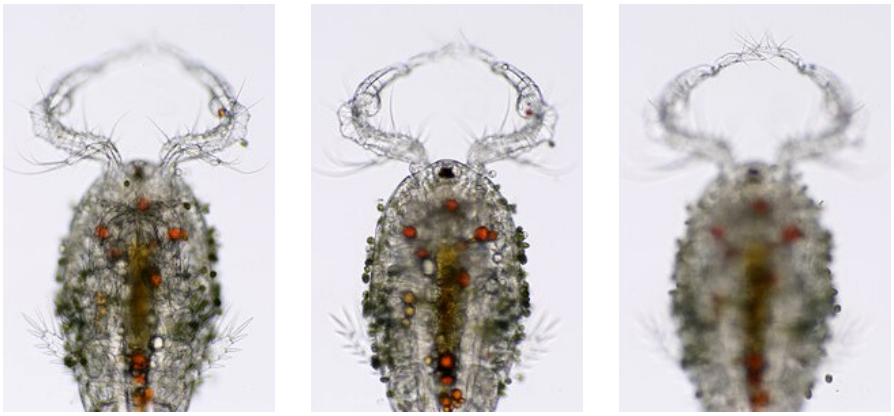


Figure 1.3: Unsharp-masking example. (a) Input image that contains in-focus foreground and defocused background. (b) Output of spatially invariant unsharp-masking. Image (b) shows that sharpness of the background is enhanced, but halo artifacts also appear in the foreground area. (c), (d) Zoomed part of (a), (b).

There are many other applications where measuring local sharpness is required for image restoration. One example is bright-field microscopy, which usually has a limited depth of field. Commonly, the surface profile of the object is beyond the focal range, since in microscopy the object is placed close to the lens. Hence, in a single shot only part of the surface within the depth of field can be sharp (see Fig. **1.4**). To extend the depth of field, a common method is to take a series of multi-focus images by gradually moving the focal plane, so that different parts of the object can be in focus and sharp in different images. Then, the sharpest region in each position is selected



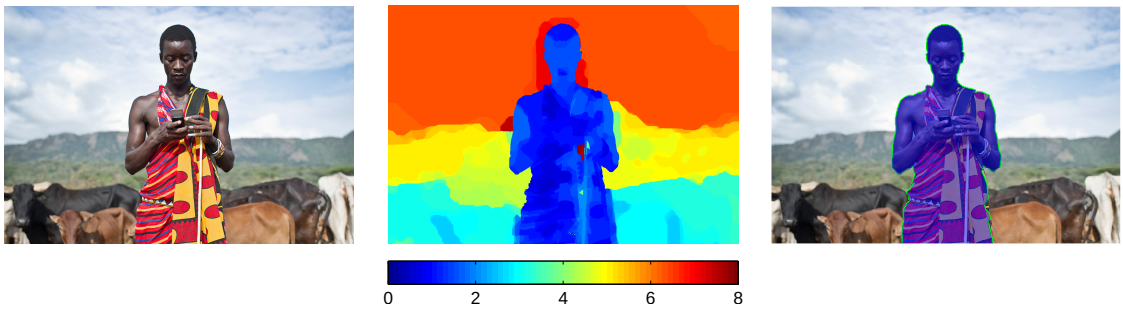
(a) (b) (c)

Figure 1.4: Three microscopic images taken with different camera settings. For each image, only part of the object is in-focus.

through a local sharpness metric, and all these regions are fused together to get an all-in-focus image [39, 40, 2].

Monitoring the amount of sharpness can also be useful in optimizing image reconstruction filters. Image reconstruction filter always contain some tuning parameters which can strongly affect their performance, such as smoothing parameter for bilateral filter, regularization parameter for most maximum-a-posteriori (MAP) based denoising/deblurring filters. If the value of such parameter is too large we may end up with over-smoothed images. On the other hand, if the value is too small we may get images that are noisy and sometimes with artifacts. Ideally, the setting of these parameters should be adapted to the image content. So, if we can efficiently measure the amount of blur along with other artifacts (noise, ringing, etc.) then it is possible to automatically optimizing such tuning parameters in a black-box manner for any reconstruction filter.

The degree of local blur may also reveal other useful information. For example, defocus blur level is intimately related with depth of the scene, so local sharpness measurement can provide an important cue for depth estimation. The computation of



(a) (b) (c)

Figure 1.5: Defocus blur measuring experiment. (a) Input test image. (b) Measured local blur map, where higher value represents more blurry. (c) Automatic foreground/background segmentation based on blur map (b).

depth information typically requires two photos of the same scene taken at the same time but from slightly different vantage points, i.e. a stereo pair [41]. However, in most practical applications only one image is available. Measuring local sharpness allows us to reconstruct a 3-D scene from a single photograph. Fig. **1.5** illustrates an example using one of our proposed methods (which we will discuss in detail in Chapter 4). Such measurement also provides information for foreground/background segmentation, which can help a photo editor to edit the subject of interest or the background, separately.

1.2.2 Difficulties in Blur Measurement

Unfortunately, in practice given observed images only it is never trivial to measure blur, especially in the spatially varying case.

Given only the observed image, estimating the blur PSF is an ill-posed problem even for the spatially invariant convolution model (1.1), because the degree of unknown $\dim(f) + \dim(h)$ is larger than the degree of known $\dim(g)$. So solving this problem requires prior constraints on both the latent image f and the blur PSF h. Recent

research on natural image statistics has shown that sharp image gradients obey *heavy-tailed* distributions that have most of their mass on small values but give significantly more probability to large values than Gaussian distributions [7]. Based on these studies, several *sparse* regularization methods have been introduced to solve the space-invariant blind deconvolution problem [7, 37, 8], all of which share a similar formula as:

$$< \hat{f}, \hat{h} >= \arg\min_{f,h} \|g - h \otimes f\|^2 + \lambda_1 R_f(f) + \lambda_2 R_h(h), \tag{1.8}$$

where R_f and R_h are the regularization terms based on prior knowledge about the latent sharp image f and the PSF h.

These methods have been implemented for estimating and removing uniform camera shaking blur. However, none of them can solve the general spatially varying blur estimation problem with imaging model (1.3), where $\dim(h) \gg \dim(g)$ indicating a highly ill-posed situation. In certain circumstances space-invariant PSF changes smoothly over the image space, and thus it can be locally treated as spatially invariant (such as atmospheric turbulence caused blur [15, 18, 19]). So some researchers have applied some methods based on (1.8) locally to see if they can estimate nonuniform PSF at each local region [11]. Unfortunately, it turns out that because blind deconvolution (1.8) relies on image statistics, it generally requires a large amount of observation, while local estimation can hardly provide sufficient image data. That is why turbulence distortion removal is still a challenging issue.

Similarly, sharpness measurement is not easy. Although various kinds of sharpness metrics have been proposed in the past two decades, they still are not routinely employed to achieve some of the applications mentioned in Section 1.2.1.

One limitation for these metrics is noise. Most sharpness metrics are based on

local image energy [42, 43] or edge width [44, 45, 4]. They can hardly distinguish image quality decay against high frequency behavior due to noise [46]. One example is the Riemannian tensor based metric [47], whose value drops when test image becomes more blurry. However, the value of this metric rises if the image noise variance is increased. Such metric can not be used to monitor image quality in the case where both blur and noise level changes (for example, the filter parameter optimization problem).

Another important limitation is about measuring relative sharpness score. Many sharpness metrics actually estimate the energy of image structure [42, 48, 43]. They are, of course, correlated with the energy of blur PSF: if the energy of PSF drops, which means more blurry, the metric value also changes monotonically. However, their values are also determined by the energy of the latent image content. A blurry high contrast edge region may have a higher sharpness score than a sharp low contrast one. Thus, though such metrics are useful in comparing sharpness of different versions of the same image content, they cannot be used to compare sharpness between different image contents. In this thesis we call them *absolute metrics*.

On the contrary, the value of a *relative metric* should be irrelevant to the latent image contrast. Ideally, such metric should be able to provide a relative sharpness score of a given image region with respect to its latent sharp one. Applications like adaptive sharpening and depth map estimation need *local* relative sharpness metrics, which are more difficult to achieve since in practice the latent image is not available. So we must use some assumptions about the latent image to serve as a reference. For example, in article [44] edge profile is employed to indicate sharpness, which is based on the assumption that all latent edges are ideal step edges. Unfortunately, edge profile based

metrics are only valid in specific regions, and thus so far they are often used for *global* sharpness measurement. How to achieve a reliable local relative sharpness metric is still a problem.

1.3 Contributions

To break the limits of existing metrics, in this thesis we propose several algorithms for measuring image sharpness. Some application systems based on these algorithms are also introduced, which can be use to solve the real problems related with spatially varying blur. The thesis is structured as follows:

- **Chapter 2 - Measuring Local Sharpness in the Presence of Noise [49, 50, 51]**

  In this chapter we analyze some simple absolute sharpness metrics based on local image activity and their performances in fusion-based image restoration algorithms, including Focus-stacking in microscopy, where noise level is stable over the image space. We also consider the atmospheric turbulence problem in long-distance imaging systems and propose a restoration system based on these simple absolute metrics. This system first reduces the space and time varying deblurring problem to a shift invariant one through a registration and fusion procedure, and then it removes the blur using a blind deconvolution step. Experiments using real data illustrate that this approach can effectively alleviate blur and geometric deformation, recover details of the scene and significantly improve visual quality.

- **Chapter 3 - Assessing Blur and Noise in Images [52, 53, 54, 55]**

  We propose a new absolute metric extended from a simple sharpness metric de-

scribed in Chapter 1. However, different from Chapter 1, both blur and noise are considered in this chapter and the new metric responds reasonably to these two factors. In other words, the value of this metric drops as the given image becomes either blurry or noisy. Though it is still an absolute sharpness metric, we found that it is highly correlated with HVS and thus could be used in optimizing denoising filters where we need to carefully balance between bias (blur) and variance (noise) of the estimate.

- **Chapter 4 - Estimating Local Defocus Blur [56]**

  In this chapter measuring relative sharpness is considered, and we propose a single image local defocus blur level estimation method. This method is capable of measuring the probability of local blur level in the continuous domain by analyzing the localized Fourier spectrum. It also takes smoothness and color edge information into consideration to generate a coherent blur map indicating defocus blur level at each pixel. Simulated and real data experiments illustrate excellent performance, and its successful applications in foreground/background segmentation.

- **Chapter 5 - Conclusions and Future Works**

  In the final chapter we conclude the thesis and discuss possible topics for future research.

Chapter 2

Measuring Local Sharpness in the Presence of Noise

Abstract - In this chapter we first analyze some simple absolute sharpness metrics based on local image activity and their performances in comparing sharpness in the presence of noise with fixed spatial strength. Then, we propose two restoration systems that utilize such absolute metrics, including a focus-stacking algorithm for multi-focus image fusion and an air turbulence removal approach for long-distance imaging systems. The focus-stacking algorithm employs a local sharpness metric to roughly select the sharpest pixel within observed images at each position. Different from other stacking methods, this algorithm refines the rough selection map to correct possible errors. Experiments show that such refinement could effectively reduce fusion artifacts and preserve boundary regions between objects from different focal planes. The air turbulence removal approach first reduces the space and time varying deblurring problem to a shift invariant one through a registration and fusion procedure, and then it removes the blur using a blind deconvolution step. Experiments using real data illustrate that this approach

can effectively alleviate blur and geometric deformation, recover details of the scene and significantly improve visual quality.

2.1 Local Sharpness Metrics

Although there are many sharpness metrics proposed in the past two decades, the term *sharpness* is somehow vague. This is probably because the term often refers to subjective human vision perceptual. So far we cannot find any standard mathematical definition for image sharpness. However, we believe that visual sharpness is determined by blur PSF. Consider the three PSF examples illustrated in Fig. **2.1**. There is no doubt that given the same latent image content, the output convolved by function (a) looks sharper than the one blurred by function (b) or (c), since the peak of (a) is much higher than the other two. Meanwhile, it is visually difficult to compare sharpness between the images blurred by functions (b) and (c), though (b) and (c) have completely different forms.

Intuitively sharpness should be correlated with the peakedness (or flatness) of the PSF. There are several sharpness metrics defined based on this idea. One example is the *Strehl ratio*, which is popular in Astronomical imaging [57, 58, 43]. Given a measured PSF $h[x, y]$ and a perfect diffraction-limited PSF $h_d[x, y]$ of the same optical system, the Strehl ratio is defined as the ratio between the peak intensity of $h[x, y]$ and $h_d[x, y]$:

$$S = \frac{\max(h[x, y])}{\max(h_d[x, y])}. \tag{2.1}$$

This metric provides a way of measuring the peakedness of $h[x, y]$. Strehl ratio is commonly used in the analysis of adaptive optics (AO) system performance. One example

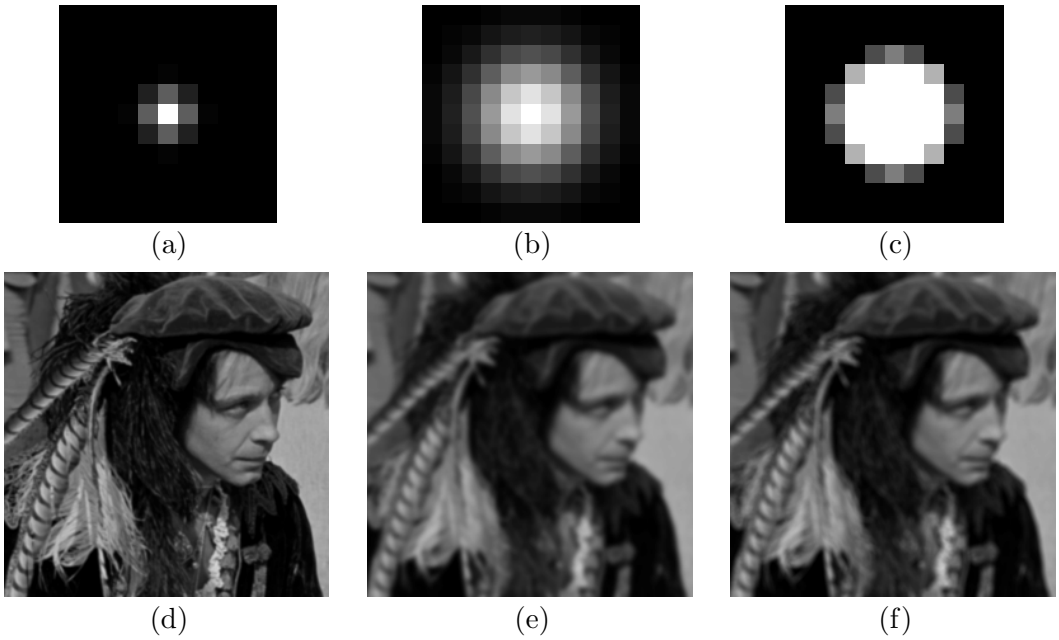


Figure 2.1: Examples of PSFs with different blur level and their corresponding blurry images. (a)-(c): Blur PSFs. (d)-(f): Images convolved by (a)-(c).

is AO system tuning, where the Strehl ratio is measured repeatedly for the same source while parameters of the AO system are adjusted to maximize the Strehl ratio value [58]. Actually in this case the absolute value of Strehl ratio does not matter so much as the relative change of the metric value. In other words, just measuring $S = \max(h[x, y])$ is enough, since $h_d[x, y]$ can be treated as a constant.

There are other ways to measure the peakedness of PSF. For example, because PSF is usually assumed to be normalized $\sum h[x, y] = 1$, and in most cases it should be positive $h[x, y] \geq 0$, its peakedness is roughly proportional to its energy:

$$S = \sum_{x,y} h^2[x, y]. \tag{2.2}$$

Note that convolution in the spatial domain is equivalent to point-wise multiplication in the Fourier domain:

$$\mathcal{F}(h \otimes f)[\omega, \upsilon] = \mathcal{F}(h)[\omega, \upsilon] \cdot \mathcal{F}(f)[\omega, \upsilon], \tag{2.3}$$

and that Fourier transform preserves energy

$$\sum h^2[x,y] = \sum \mathcal{F}(h)^2[\omega,\upsilon]. \tag{2.4}$$

So the energy of PSF measures the attenuation of the image spectrum due to blur:

$$S = \sum_{\omega,\upsilon} \mathcal{F}(h)^2[\omega,\upsilon] \tag{2.5}$$

In astronomical imaging systems, a widely used PSF energy-based sharpness metric is called *encircled energy factor* (EEF), which measures the amount of energy contained within a circle of specific radius [59]. It often serves as an index of optical system performance [59].

One problem for these PSF based sharpness metrics is that they require a pre-estimate of the local PSF. This requirement is reasonable for most astronomical systems, where it is easy to find a *reference star* within or near the region of interest, which can be assumed as a point source and its observation provides a direct measurement of local PSF. Unfortunately, this is not the case for general imaging systems.

In this section, we will discuss some sharpness metrics calculated directly from observed blurry images. These metrics are capable of indicating the change of local blur level, and thus can be implemented in some fusion based applications.

2.1.1 Sharpness from Image Content

As we have explained in Chapter 1, estimating local PSF is highly difficult. To avoid PSF estimation many sharpness metrics directly calculated from an input blurry image $g[x,y]$ have been proposed [44, 45, 4]. One simple example is the variance metric

[60, 61, 62, 63, 42, 43]:

$$S_{var} = \frac{1}{N-1} \sum_{x,y} \left(g[x,y] - \bar{g}\right)^2 \tag{2.6}$$

where $\bar{g}$ represents the mean value of the image (or image region) g, and N denotes the number of pixels within g. S_{var} provides an unbiased estimation of the variance of g. Intuitively, as the blur level increases, image contents (e.g. edges, textures) are smoothed making the transitions between image intensities reduced, and the value of image variance drops [4].

Ignoring the normalization factor $\frac{1}{N-1}$, we can see that the variance metric basically calculates the total energy of the image region except its DC component $\bar{g}$. In the Fourier domain metric S_{var} can be written as

$$\begin{aligned} S_{var} &= \frac{1}{N-1} \sum_{\omega,\upsilon \neq 0} \mathcal{F}(g)^2[\omega,\upsilon] \\ &= \frac{1}{N-1} \sum_{\omega,\upsilon \neq 0} \mathcal{F}(f)^2[\omega,\upsilon] \cdot \mathcal{F}(h)^2[\omega,\upsilon]. \end{aligned} \tag{2.7}$$

Compared with (2.5), the above metric provides a weighted sum of the PSF energy, where the weights are determined by the latent image content $\mathcal{F}(f)^2[\omega,\upsilon]$, and thus it cannot be used as a relative sharpness metric. However, if we just need to compare sharpness between different blurred versions of a same piece of image content, (for example, compare sharpness among images (d)-(f) in Fig. **2.1**,) then this simple variance metric should be enough for such comparison [22, 64, 43]. In [43] it was shown that this variance metric has a unique relationship with the Strehl ratio and can be used for applications such as sharpest patch selection.

There are some other ways to measure image sharpness. A summary of existing no-reference sharpness metrics can be found in [4]. Besides variance, these methods

can generally be categorized as gradient based metrics [48], spectrum kurtosis metrics [60, 61], histogram based metrics [62, 63], and edge width based metrics [44, 45, 4]. Most of them are absolute sharpness metrics, which can be used to compare sharpness among different blurred versions of the same image. A few of them are relative sharpness metrics (e.g. edge width based metrics), which are able to compare sharpness between two different images.

In this chapter the issues we focus on are fusion-based restoration systems. Specifically, we assume that a given imaging system can generate a sequence of images of the same scene, and each image has different regions of the scene in high quality. The goal is to detect all the high quality regions and fuse them together to get a single restored image. For this specific scenario, we do not care about relative sharpness measurement, since we just need to compare sharpness among various copies of the same image region. However, the required sharpness metric should be valid locally. In other words, it should not rely on a large amount of image samples to get the sharpness score. Due to this reason, metrics based on histogram or spectrum statistics (like spectrum kurtosis) cannot be employed. Also, the metric should be valid for general types of image content, which means edge width based metrics can not be implemented. In the following subsection, we will analyze the performance of a gradient-based metric, which is more sensitive to blur compared with the variance metric.

2.1.2 A Gradient-based Metric

A simple gradient-based metric is defined as:

$$S_{grad} = \frac{1}{N} \sum_{x,y} g_x^2[x,y] + g_y^2[x,y] \tag{2.8}$$

where $g_x[x, y]$ and $g_y[x, y]$ denote the horizontal and vertical derivatives, respectively. The two derivatives can be calculated by applying, for example, the following high-pass filters:

$$\frac{1}{2} \times \begin{bmatrix} 0 & 0 & 0 \\ -1 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}, \quad \frac{1}{2} \times \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \tag{2.9}$$

Comparing the gradient-based metric S_{grad} and the variance metric S_{var} we can see that both represent (normalized) energies of the test image region g, while S_{grad} concentrates more on its high-frequency components. This is consistent with human visual system (HVS), whose blur sensitivity is higher in high frequency components compared with low frequency ones [65].

To illustrate the performances of the two metrics, a simulated experiment is carried out. In this experiment, 4 test patches (of size 11×11) are generated representing some typical local image contents, including linear slope, quadratic, edge and square (see Fig. **2.2** (a)-(d)). Then, we apply a Gaussian blur kernel (of size 5×5) with a steadily growing standard deviation σ_b to the these patches (and output examples are given in Fig. **2.2** (e)-(p)). In this step, noise is not considered. We can see that though convolved by the same set of blur kernels, changes of sharpness in the structured patches (edge patch and square patch) can be easily observed, while in the smooth patches (linear slope patch and quadratic patch), which lacks high-frequency components, it is visually hard to tell changes of the blur level.

The corresponding metric values are recorded and plotted in Fig. **2.3**. Because the ranges of the two metrics are quite different, all the metric curves are normalized by their first (also largest) values so that we can easily compare their behaviors. Both

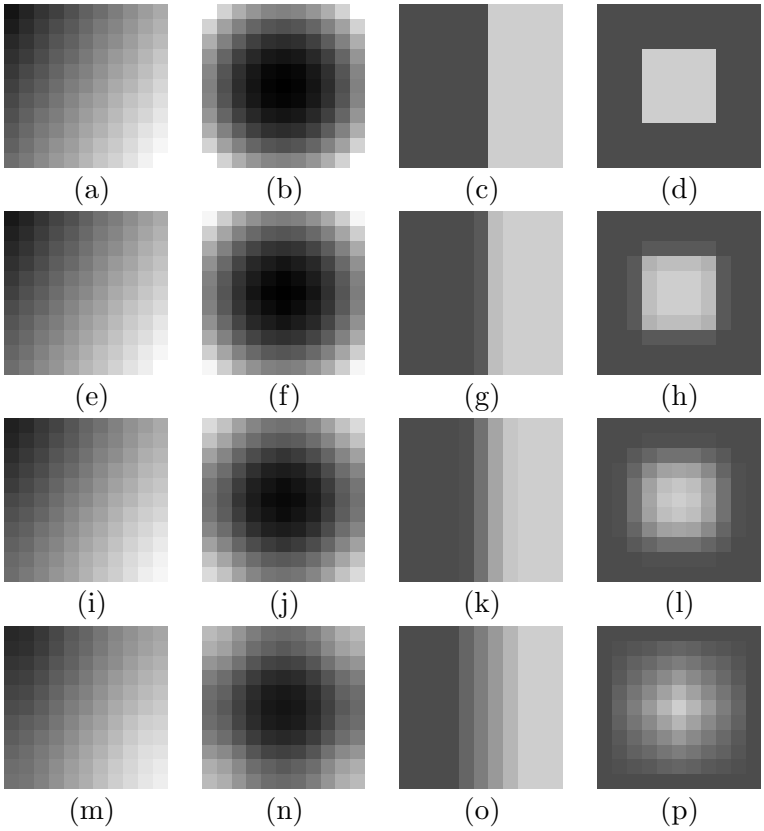


Figure 2.2: Patches used in the simulated blurring experiment and some selected blurred outputs. Gray levels are normalized to the range [0,1]. Columns from left to right: linear patch, quadratic patch, edge patch, square patch. Rows from top to bottom: original patches, patches blurred with kernel $\sigma_b = 0.5$, $\sigma_b = 1.0$, $\sigma_b = 5.0$

S_{grad} and S_{var} monotonically drops as all the patches get more and more blurry as expected. For all the patches S_{grad} has a sharp top, and the general slope of its curves is steeper than S_{var} indicating its higher sensitivity to blur. This is because most of the attenuation due to blur happens in high-frequency channels, and the high-pass filtering in S_{grad} magnifies such attenuation effect. That explains why S_{grad} is favored in many fusion based applications [22, 23, 21, 19].

Although in this chapter we concentrate on the scenarios where noise level is assumed to be constant everywhere, noise effect is still a problem because it brings additional variation to the local sharpness metric. Ideally, the value of S_{grad} monoton-

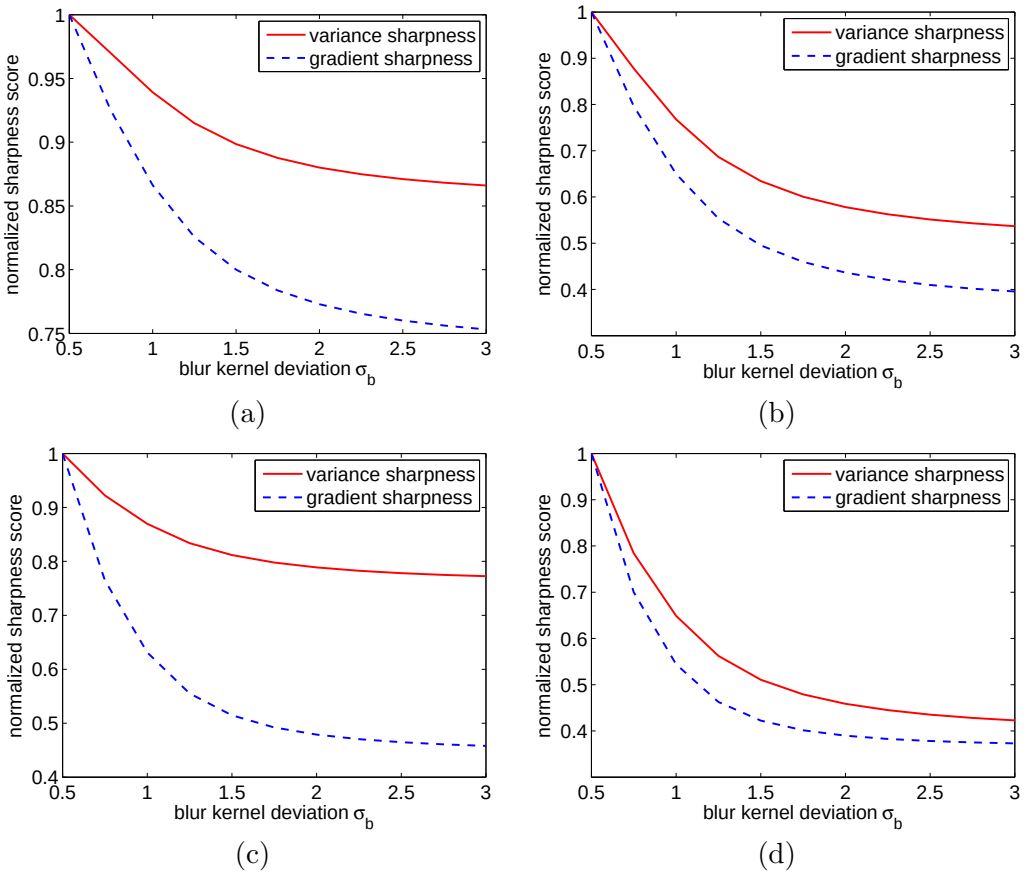


Figure 2.3: Plots of S_{var} and S_{grad} in blurring process for different patches. The blur kernel is Gaussian shaped, and the value of its standard deviation σ_b is raised steadily to make each patch more and more blurry. (a) Linear patch; (b) quadratic patch; (c) edge patch; (d) square patch.

ically drops as a test image region gets more and more blurry. In practice, however, its monotonicity can be affected by noise.

Let us now consider a degradation model similar to (1.1), which is rewritten here in a vector form:

$$\mathbf{g} = \mathbf{H}\mathbf{f} + \mathbf{n} = \mathbf{g}^* + \mathbf{n} \tag{2.10}$$

where the $N \times 1$ vector $\mathbf{g}^*$ denotes the noise-free blurry observation. When calculating the horizontal and vertical derivatives of $\mathbf{g}$ through the filters in (2.9), the filtering

process can also be represented by two $N \times N$ matrices $\mathbf{D}_x$ and $\mathbf{D}_y$:

$$\mathbf{g}_x = \mathbf{D}_x\mathbf{g} = \mathbf{g}_x^* + \mathbf{n}_x, \quad \mathbf{g}_y = \mathbf{D}_y\mathbf{g} = \mathbf{g}_y^* + \mathbf{n}_y \tag{2.11}$$

where, $\mathbf{g}_x^*$ and $\mathbf{g}_y^*$ denote noise-free horizontal and vertical derivative vectors, respectively. The derivative noise vectors $\mathbf{n}_x = \mathbf{D}_x\mathbf{n}$, $\mathbf{n}_y = \mathbf{D}_y\mathbf{n}$ are horizontal and vertical components of the noise $\mathbf{n}$ that is usually assumed as zero-mean independent and identically distributed (IID) .

Strictly speaking, Neither $\mathbf{n}_x$ nor $\mathbf{n}_y$ is IID. Take the horizontal derivative vector $\mathbf{n}_x$ for example. Suppose the variance of $\mathbf{n}$ is σ^2, then the expected vector $E(\mathbf{n}_x) = 0$, and the covariance matrix of $\mathbf{n}_x$ is

$$\begin{aligned}
\text{cov}(\mathbf{n}_x) &= E\left(\left(\mathbf{n}_x - E(\mathbf{n}_x)\right)\left(\mathbf{n}_x - E(\mathbf{n}_x)\right)^T\right) \\
&= E\left(\mathbf{D}_x\mathbf{n}\mathbf{n}^T\mathbf{D}_x^T\right) \\
&= \mathbf{D}_x E\left(\mathbf{n}\mathbf{n}^T\right)\mathbf{D}_x^T \\
&= \sigma^2\mathbf{D}_x\mathbf{D}_x^T.
\end{aligned} \tag{2.12}$$

$\mathbf{D}_x\mathbf{D}_x^T$ is clearly not an identity matrix $\mathbf{I}$. Suppose we use the filter in (2.9) to make the derivative matrix $\mathbf{D}_x$, and define $\mathcal{D} = \mathbf{D}_x\mathbf{D}_x^T$, then in the i-th row of $\mathcal{D}$ we have $\mathcal{D}_{i,i} = \frac{1}{2}$, and there are another two entries with the value of $-\frac{1}{4}$, and the rest are zeros. In other words, given any row of $\mathcal{D}$ we have

$$|\mathcal{D}_{i,i}| = \sum_{i \neq j} |\mathcal{D}_{i,j}|, \tag{2.13}$$

and the same situation happens in each column of $\mathcal{D}$. Such matrix is *diagonally dominant* [66], and it is close to an identity. For simplicity, in the rest of this section both $\mathbf{n}_x$ and $\mathbf{n}_y$ are approximated as IID noise vectors with covariance matrices $\text{cov}(\mathbf{n}_x) = \text{cov}(\mathbf{n}_y) = \frac{1}{2}\sigma^2\mathbf{I}$. We also treat $\mathbf{n}_x$ and $\mathbf{n}_y$ to be uncorrelated with each other.

According to equation (2.8) the gradient sharpness metric S_{grad} can be written as

$$S_{grad} = \frac{1}{N}\left(\mathbf{g}_x^T\mathbf{g}_x + \mathbf{g}_y^T\mathbf{g}_y\right) \tag{2.14}$$

and hence the variance $var(S_{grad}) = \frac{1}{N^2}var(\mathbf{g}_x^T\mathbf{g}_x) + \frac{1}{N^2}var(\mathbf{g}_y^T\mathbf{g}_y)$. The form of $var(S_{grad})$ depends on the distribution model of noise $\mathbf{n}_x$ and $\mathbf{n}_y$. For example, if they are Gaussian distributed, then the gradient energy $\mathbf{g}_x^T\mathbf{g}_x$ follows a *noncentral chi-squared distribution* with N degrees of freedom. It can be derived that

$$var(\mathbf{g}_x^T\mathbf{g}_x) = \frac{1}{2}N\sigma^4 + 2\sigma^2\mathbf{g}_x^{*T}\mathbf{g}_x^*. \tag{2.15}$$

So the variance of S_{grad} becomes

$$var(S_{grad}) = \frac{\sigma^4}{N} + \frac{2\sigma^2}{N^2}\left(\mathbf{g}_x^{*T}\mathbf{g}_x^* + \mathbf{g}_y^{*T}\mathbf{g}_y^*\right) \tag{2.16}$$

Such variation caused by noise could affect the performance of sharpness metric S_{grad}. An example is given in Fig. **2.4**, where we use the edge patch in Fig. **2.2** to do the blurring simulation similar to Fig. **2.3**. White Gaussian noise is also added into the blurry patch sequence. The intensity difference between the two sides of the edge is 0.5, and the noise standard deviation $\sigma = 0.05$. Patch size $N = 121$, and the patch SNR = 21dB. It can be seen that noise distorts the monotonicity of the curve in certain degree (though it did not change the priority of the sharpest patch in this specific example). One way to reduce noise effect is by increasing the patch size N. However, this would involve more surrounding pixels, which may have completely different blur level. This issue could affect the performance of fusion based restoration algorithms. One example is the classic focus-stacking methods discussed in the next section, where we will propose a new image fusion framework to alleviate this problem.

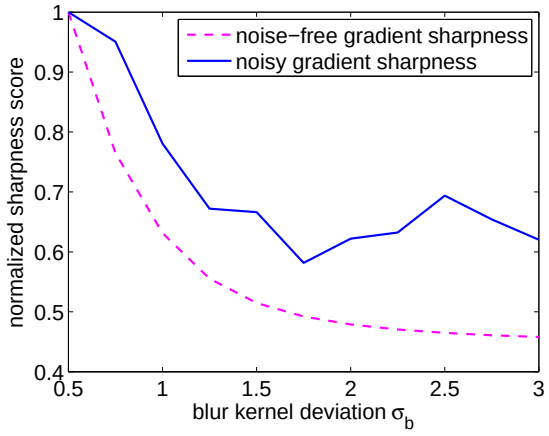


Figure 2.4: Plots of normalized S_{grad} for the edge patch (in Fig. **2.2** (c)) in a blurring process with/without noise. The blur kernel is Gaussian modeled, and the value of its standard deviation σ_b is raised steadily to make each patch more and more blurry. The dashed magenta curve is the plot of noise-free S_{grad}, and the solid blue curve represents the sharpness scores contaminated by a Gaussian noise realization (SNR = 21dB).

2.2 Application I: Focus-Stacking

Focus-stacking has been widely used to extend the depth-of-field of a conventional imaging system [76, 75]. For example, in bright field microscopy and macro photography it is very common that surface profile of the observed object is beyond the focal range since the object is placed very close to the lens. So in a single shot only part of the surface within the depth of field can be sharp, while the rest areas are blurred. To overcome this problem, one usually takes a series of multi-focus images by gradually moving the focal plane, so that different parts of the object can be in focus from different images. Then, all the in-focus regions are merged together through a process called focus-stacking to generate an all-in-focus image [67, 40, 2].

Because such imaging process are usually carried out under a controlled circumstance, where both the target object and the camera are static, there is almost no motion among the observed images. So image alignment is not a problem in this case.

Though various focus-stacking methods have been proposed, given a multi-focus image set $\{g_k\}$ most of them follow a general and common scheme, which contains three major steps [2]:

1. **Sharpness measurement.** For each pixel $g_k[x, y]$ the local sharpness $s[x, y, k]$ is measured through a specific metric (e.g. S_{grad}). The metric is usually calculated based on a small window (with a fixed size) centered at each given pixel. Sometimes a Gaussian smoothing procedure is also applied to the calculated sharpness map to suppress noise effect [2].

2. **Decision map estimation.** Most focus-stacking methods follow the maximum selection rule [68, 2], which only selects the sharpest pixel from all the observations at each position. The decision map $I[x, y]$ stores the index of the selected image for each position: $I[x, y] = \arg\max_k s[x, y, k]$.

3. **Image Fusion.** Finally, the fused image $\hat{f}$ is calculated pixel-by-pixel as follows:

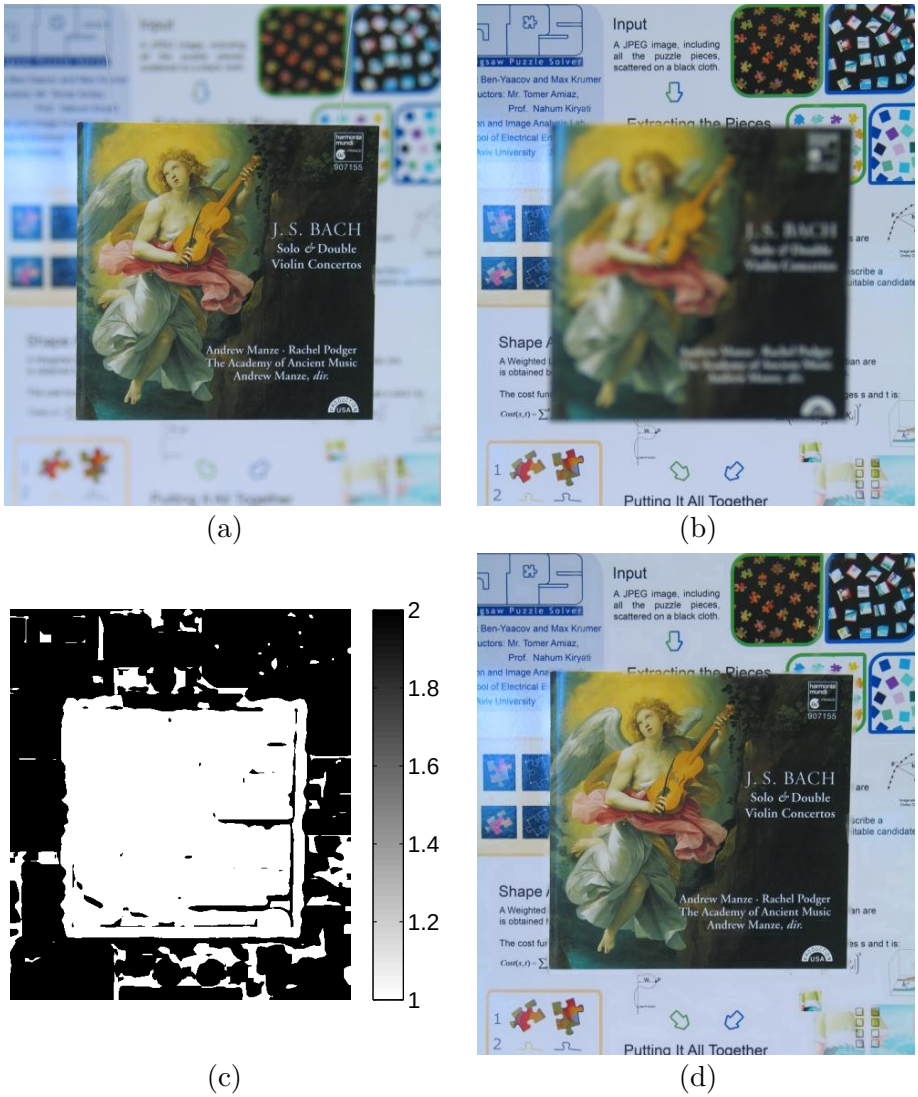
$$\hat{f}[x, y] = g_{k^*}[x, y], \quad \text{where} \quad k^* = I[x, y]. \tag{2.17}$$

A significant problem of the above scheme is that in practice we always need to carefully adjust the window size of the sharpness metric to balance between the stability and accuracy of the algorithm. Sharpness can not in practice be measured only based on a single pixel value; it always needs a piece of local region with unique blur level to calculate a reliable sharpness score. The larger the window is, the more reliable the sharpness score will be, with less noise effect (as explained in (2.16)). However, with a large window it is highly likely that neighboring objects with different blur level could be

included into the sharpness analysis window, which could strongly affect the sharpness score.

Such problems frequently appear in the boundary regions between two objects of different focus level. For example, we have two observed images in Fig. **2.5**, where input image (a) has the foreground CD case in focus, while image (B) has the background in focus. Panel (c) shows a decision map based on the gradient metric S_{grad}. This map is not consistent with the depth of the scene. The boundary area between foreground and background does not match the outline of the CD case. Specifically, in image (a) defocused background regions close to the CD case may have a higher sharpness score than the corresponding (in-focus) regions in image (b), since the sharp edge of the case in (a) could increase their local sharpness scores. As a result, we can see several fusion artifacts in the output image (d). For instance, some sharp details on the background, which are close to the case, are excluded from the fused output. Meanwhile, there also exist edges of the case that are blurred due to their neighboring strong background structures.

To alleviate this problem, we introduce a decision map refinement step into the focus-stacking scheme. This step incorporates sharpness information and image color information together to correct errors around the boundaries on the decision map, making the map consistent with the observed images. Experiments show that this method could effectively reduce fusion artifacts and generate high quality stacked images.



(a) (b)

(c) (d)

Figure 2.5: Multi-focus image pair example. (a) and (b) are the input image pair. (c) is a roughly calculated decision map based on local sharpness measurement. (d) is the corresponding fused image.

2.2.1 Focus Stacking Framework

Diagram of the proposed method is given in Fig. **2.6**. Given observed images, we first calculate local sharpness using S_{grad} with a relatively larger window (to stabilize the map). Then, a rough decision map is generated by maximizing each local sharpness. After that, given the rough map and the observed images, a refinement step

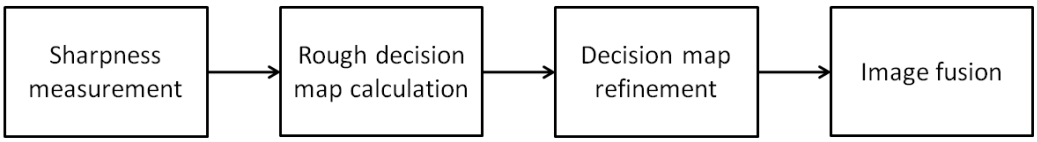


Figure 2.6: Diagram of the proposed focus-stacking algorithm.

is implemented to correct the decision map. Finally the fused output is produced using the refined decision map.

Most steps of this framework are the same as the common focus-stacking scheme. The only difference, which is also the key step of the method, is the additional decision map refinement. In our method, an α matting algorithm is implemented to refine the decision map [69, 70].

Let us first consider a simple two frame case, where we have a foreground in-focus image A and a background in-focus image B (such as Fig. **2.5** (a) and (b)). Because sharp boundaries are preserved in A, we choose A as the "reference" for the refinement. Suppose image A is synthesized by an ideal all-in-focus image C and an all-defocused image D through the following combination

$$A_i = \alpha_i C_i + (1 - \alpha_i) D_i \tag{2.18}$$

where A_i denotes the i-th pixel located at position $[x, y]$, and α_i represents the contribution of C_i. The α map is very similar to a decision map except that α ranges in $[0, 1]$ while the decision map is binary.

Suppose we already have a rough decision map $\tilde{I}$ and convert it into a binary alpha map $\tilde{\alpha}$. Given $\tilde{I}$ along with the observed image A we can estimate a refined α

map vector by minimizing the following cost function proposed by Levin [69]:

$$E(\alpha) = \alpha^T \mathbf{L} \alpha + \lambda(\alpha - \tilde{\alpha})^T(\alpha - \tilde{\alpha}) \tag{2.19}$$

where the first term on the right hand side is a smoothness term and the second one is a data term. λ is a regularization parameter and $\mathbf{L}$ denotes the Matting Laplacian matrix with its (i, j) element defined as [69]

$$\mathbf{L}_{i,j} = \sum_{m|(i,j)\in w_m} \left(\delta_{i,j} - \frac{1}{|w_m|} \left(1 + (\mathbf{A}_i - \mu_m)^T (\Sigma_m + \frac{\epsilon}{|w_m|}\mathbf{I}_3)^{-1} (\mathbf{A}_j - \mu_m) \right) \right). \tag{2.20}$$

$\mathbf{A}_i$ and $\mathbf{A}_j$ represent RGB color vectors of image A at the i-th and j-th pixels. $\delta_{i,j}$ is the Kronecker delta. μ_m and Σ_m are the mean and covariance matrix of the colors in window w_m. $\mathbf{I}_3$ denotes a 3×3 identity matrix. ϵ is a regularization parameter, and $|w_m|$ represents the number of pixels inside a local window w_m [69]. Once the refined α map is calculated, we can easily convert it to a decision map by a hard thresholding, where the threshold is set as 0.5.

This dual-image decision map refinement method can also be extended to the multi-image case. Given a multi-image sequence, (which are ordered according to the shift of focal plane,) we first estimate a rough decision map and do the α matting refinement for each neighboring image pair, and then stack all the dual-image refined maps together to get a multi-image decision map.

2.2.2 Experimental Results for Image Fusion

Real data experiments are carried out to test the performance of our algorithm. For all the tests, we fixed the window size of local sharpness measurement to be 11×11.

For the α matting Laplacian we set the analysis window w_m of size 3×3, and the regularization parameter $\epsilon = 0.0001$.

We first test the proposed algorithm on the real image example given in Fig. **2.5**. The corresponding results are illustrated in Fig. **2.7**. Starting from a rough decision map (Fig. **2.7** (a)) and observed input images, the estimated α map (b) successfully removed almost all the decision errors and made the boundary exactly match the observed object outline. However, because the range of α is $[0, 1]$ and its value is affected by activity of local image content, this map still contains mild variation in the regions where there is no foreground-to-background transition (see (b)). Such variations are removed after thresholding and the refined binary decision map, which accurately depicts the outline of the foreground CD case, is shown in (d). By using the refined decision map, we can see that fusion artifacts caused by the previous map (a) have been effectively suppressed (see (d)-(m)). Sharp edges of the CD case are preserved, and meanwhile high-frequency details on the background, which are close to the CD case, are also kept in the output. These improvements make the foreground-to-background transition look quite natural.

To quantitatively test the accuracy of the refined decision map and to compare it with other advanced focus-stacking methods, we tried another image pair given in Fig. **2.8** (a) and (b). These two gray-scale images are from paper [2] proposed by Redondo et al., where a ground truth decision map (see image (c)) is also provided[1]. Both the map generated by algorithm [2] through careful window size adjustment and proposed one are given in (d) and (e), respectively. It can be seen that, though very clean, the

[1]The ground truth decision map is constructed by occluding the background with a piece of black felt and subsequent thresholding [2].

outline in (d) is still bigger than the real foreground object. Because the input images are gray scale, where we don't have strong color information to distinguish objects, it somewhat limits the performance of α matting. However, we can still see that the proposed decision map (e) is more closer to the ground truth (c) compared with (d) as it shows detail of the outline of foreground object. The mean-squared-error (MSE) of (e), the proposed method, with respect to (c) is 0.0087, while the one of (d) is 0.0163, or about twice as large.

From the experiments we can conclude that by applying the α matting based refinement step, accuracy of the decision map could be improved. This refinement alleviates the algorithm's sensitivity to the window size, and meanwhile effectively deduces the amount of fusion artifacts.

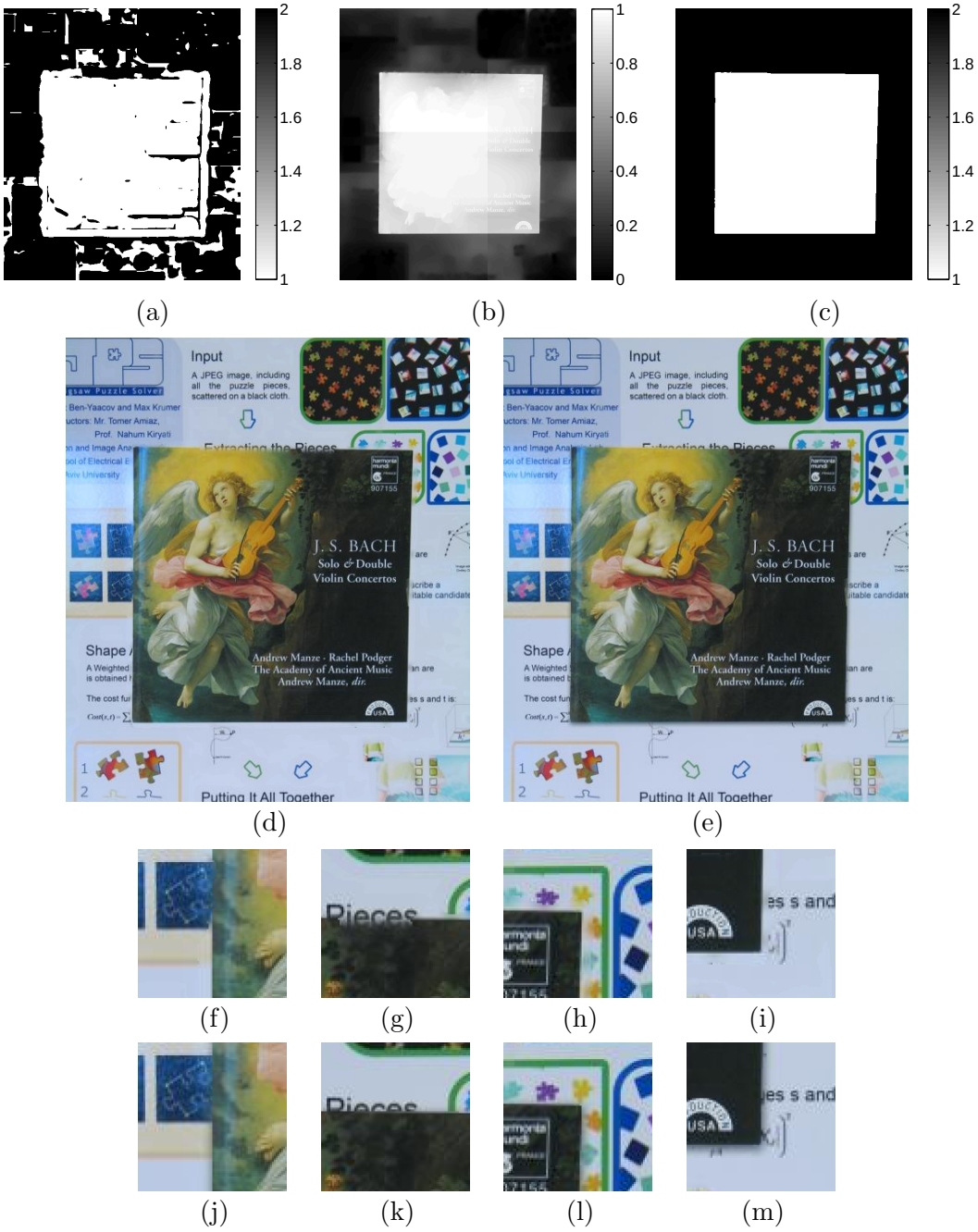


Figure 2.7: Multi-focus image fusion example. Input image pair are given in Fig. **2.5** (a) and (b). (a) Roughly calculated decision map. (b) Refined α map. (c) Refined decision map converted from (b). (d) Output based on (a). (e) Output based on (c). (f)-(i) Zoomed parts of (d). (j)-(m) Zoomed parts of (e).

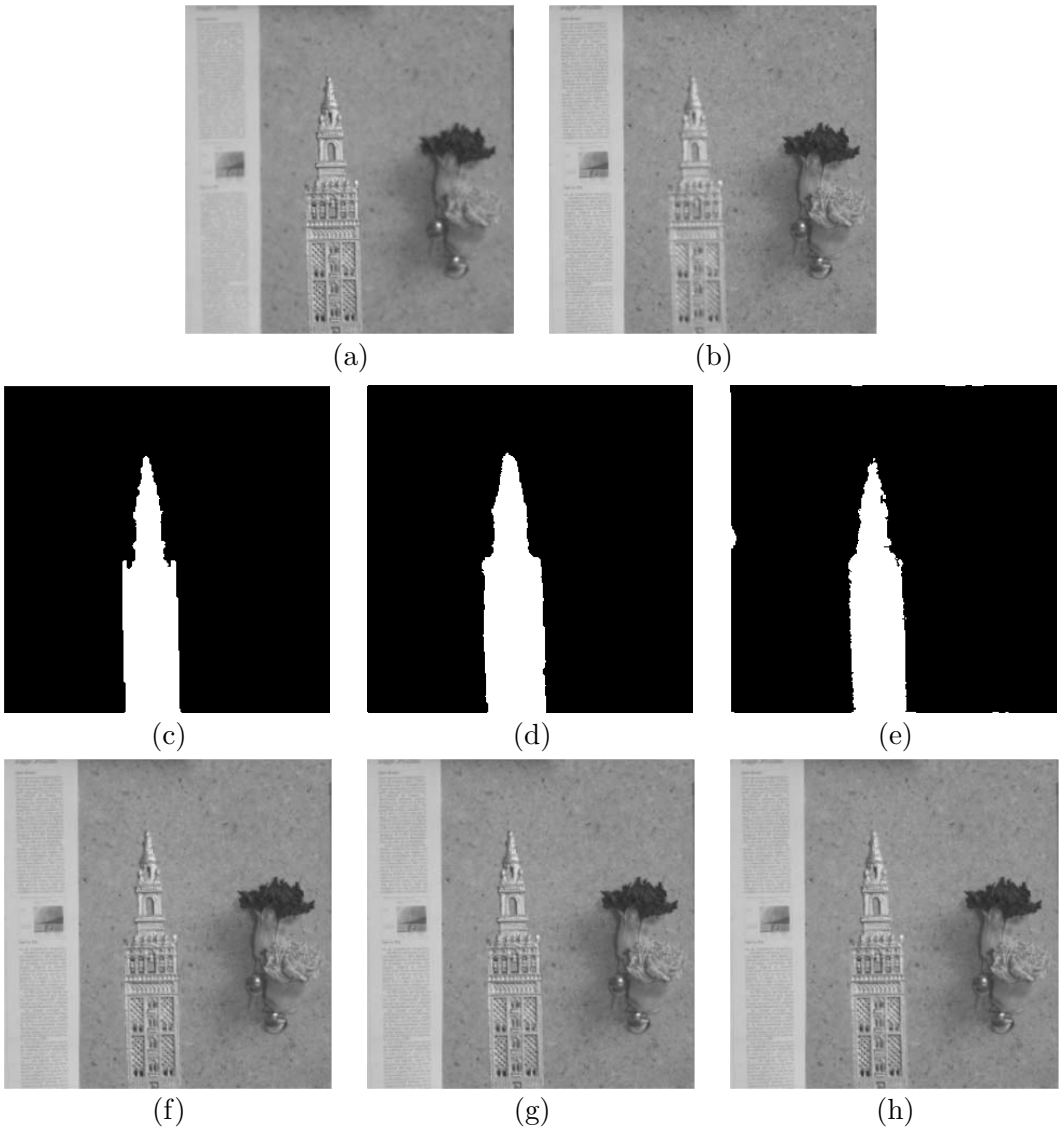


Figure 2.8: Multi-focus image pair example. (a)-(b) Input images. (c) Ground truth decision map. (d) Decision map of method [2]. (e) Proposed decision map. (f)-(h) Corresponding outputs of (c)-(e).

2.3 Application II: A Turbulence Removal System

In this section, we will discuss another important issue related with measuring spatially varying blur, which is about removing image distortions caused by atmospheric turbulence. A long-distance imaging system can be strongly affected by atmospheric

turbulence, which randomly changes the refractive index along the optical transmission path. Compared with the multi-focus imaging scenarios in Section 2.2 situations here are much more complicated. Air turbulence can generate different kinds of distortions simultaneously including geometric deformation (motion), space and time varying blur, and sometimes even motion blur if the exposure time is not sufficiently short [15, 16, 17, 18, 19]. Besides, generally speaking noise level in long-distance imaging systems is also higher than the multi-focus imaging cases (e.g. macro photography) that are usually carried out under controlled circumstances with good lightening conditions.

Aside from hardware-based adaptive optics approaches [71], several signal processing approaches have been proposed to solve the atmospheric turbulence caused imaging problem [72, 21, 18, 19, 51, 36]. These approaches attempt to restore a single high-quality image from an observed frame sequence distorted by air turbulence. As with these other works based on videos or image sequences, we work under the assumption that the scene and the image sensor are both static, and that observed motions are due to the air turbulence alone. The imaging process can be modeled as [73, 36]:

$$G_k[\mathbf{x}] = (F \otimes h_{k,\mathbf{x}} \otimes h)[\mathbf{x}] + N_k[\mathbf{x}] \tag{2.21}$$

where $\otimes$ represents a 2-D convolution operator. F, G_k and N_k denote the ideal image, the k-th observed frame and sensor noise, respectively. Because in this section we need to analyze both *global* and *local* imaging models frequently, to easily distinguish the two situations we use capital letters (e.g. F, G, N) to denote image-wise models, and lowercases (e.g. f, g, n) to represent patch-wise local models. The vector $\mathbf{x} = [x, y]$ denotes a 2-D spatial location. $h_{k,\mathbf{x}}$ represents the space-varying (air turbulence-caused) point spread function (PSF) for the position $\mathbf{x}$ in the k-th frame, which includes both

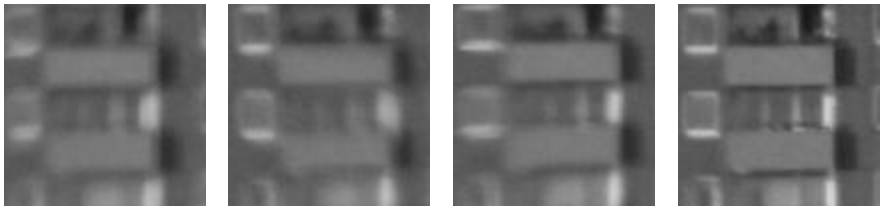
blur and motion effects [36]. h denotes lens PSF, which is mainly composed of space-invariant diffraction-limited PSF.

Because the unknown $h_{k,\mathbf{x}}$ is *spatially and temporally changing* in (2.21), restoring the high-quality image F is not trivial. Some multi-frame reconstruction approaches [18, 51] first employ a non-rigid image registration technique to register each observed frame with respect to a fixed reference grid, and use the registration parameters to estimate the corresponding motion field for each frame. Then, a sharp image is formed through a Bayesian reconstruction method. The main problem for such algorithms is that they do not estimate the actual PSF $h_{k,\mathbf{x}}$. Both [18] and [51] employ a fixed Gaussian model to approximate the PSF, which strongly limits their performance.

Recently a method called Efficient Filter Flow (EFF) was introduced by Hirsch *et al.* for space-varying blind deconvolution and has been applied for astronomical imaging to alleviate turbulence distortion, and also for magnetic resonance imaging (MRI) to reduce blur caused by object motion [36]. This method first divides each frame into overlapping patches. Because the size of these patches are small, they can be viewed as *isoplanatic* regions – small regions containing space-*invariant* blur [71, 19], and thus can be processed through a multi-frame blind deconvolution algorithm [31]. Given an isoplanatic patch sequence of a local region extracted from the input video, the blind deconvolution algorithm estimates the PSF *separately for each patch*. Final output is then generated by a non-blind image estimation step. Though capable of suppressing turbulence effects, the EFF method cannot remove diffraction-limited blur since the blind deconvolution step does not utilize much prior knowledge of the ideal sharp image except for a non-negativity constraint and a simple Tikhonov regularizer.

Besides, due to the limited accuracy of the local PSF estimation, and probably also due to the sensor noise effect, the results can contain strong deblurring artifacts, such as ringing [31].

Another class of approaches called "lucky imaging" employ image selection and fusion methods to reduce the blurring effects caused by turbulence [22, 23, 21, 19]. These approaches are very similar to the focus-stacking methods mentioned in Section 2.2. The image selection technique attempts to find frames of the best quality (lucky frames) from a short-exposure video stream by a local sharpness measurement, which is similar to the method used in Section 2.2. The output image is produced by fusing these lucky frames together [22, 23]. This method is based on the observation that for short-exposure images, turbulence creates "mutations" in image quality, and randomly makes some images sufficiently sharp (see examples in Fig. **2.9**) [23, 74, 75, 76]. This strategy is favored in many astronomical imaging applications where the image of the object of interest (e.g. a star) is usually taken inside an *isoplanatic* angle. In [76] Vorontsov *et al.* proposed a "lucky region" restoration approach for *anisoplanatic* scenarios. Small lucky (isoplanatic) regions (which can be viewed as being blurred only by the diffraction-limited PSF,) are detected by a local sharpness metric (similar to S_{grad}) and are fused to produce a large high quality image. In another similar method developed by Joshi *et al.* [77] a local block-based image alignment is first carried out to reduce geometric distortion caused by turbulence; then a lucky-imaging-based weighting scheme is employed to generate a single image that is sharp everywhere. The weighting scheme is carefully designed to balance between noise reduction and sharpness preservation. A dehazing process is finally used to enhance the visual quality.



(a) frame 1 (b) frame 4 (c) frame 11 (d) frame 23

Figure 2.9: Isoplanatic patches from short exposure image sequence *Building* taken through hot air, where a variation of blur can be observed. (d) illustrates a lucky frame example, which is much sharper than the others.

One difficulty with such method is that even though turbulence caused blur is strongly alleviated through the lucky imaging process, the output still suffers from the blur caused by the diffraction-limited PSF [19, 73]. Besides, the blur in lucky regions would inevitably be increased once temporal averaging is used for noise suppression [77].

Due to the isoplanatism of turbulence effects (including both blurring and geometric deformation) in small local regions, Lucky imaging focuses on patch-wise restoration. However, limited patch size also limits the restoration performance. As Levin *et al.* pointed out in [38], PSF estimation can be much improved as the size of the observed patch increases. In other words, if PSF can be estimated from a large region (e.g. the whole image), then we may be able to achieve better deconvolution performance. However, to achieve this goal the PSF spatial variation needs to be reduced.

Another important factor that may affect restoration is sensor noise. High shutter speed is favored in long-distance imaging systems in general to avoid motion blur caused by turbulence, and this would inevitably increase the noise level. Noise can strongly affect the performance of deblurring, as it distorts image details and produces ringing artifacts [8]. Fortunately, the frames within a sequence are highly correlated with each other, and such correlation can be used to advantage for denoising in the

temporal domain.

In this section, a framework is proposed for restoring a single image from an image sequence acquired in the general anisoplanatic scenarios. The 3-D physical scene is assumed to be static, as is the imaging sensor, while the air between the scene and sensor is affected by atmospheric turbulence. Our approach is designed to reduce the spatial variation of PSFs over the whole image space, so that the blur can be approximately treated as *spatially invariant*, and the latent image content can be estimated globally instead by local deconvolution. By doing this, we can improve the estimation accuracy; and more importantly, natural image statistics can be invoked so that diffraction-limited blur can be effectively removed. An image with reduced PSF variation (which we call the *near-diffraction-limited image*,) is generated through a fusion process. Similar to the focus-stacking method proposed in Section 2.2 we use the local sharpness metrics discussed in Section 2.1 to select sharp patches for each position. To avoid noise effects in the subsequent deconvolution, temporal kernel regression is employed in the fusion process for denoising, and this is different from the maximum selection rule used in focus-stacking methods.

2.3.1 Restoration Framework

The proposed restoration framework contains three main steps (see the diagram in Fig. **2.10**):

A. Non-rigid image registration;

B. Near-diffraction-limited image reconstruction;

C. Single image blind deconvolution.

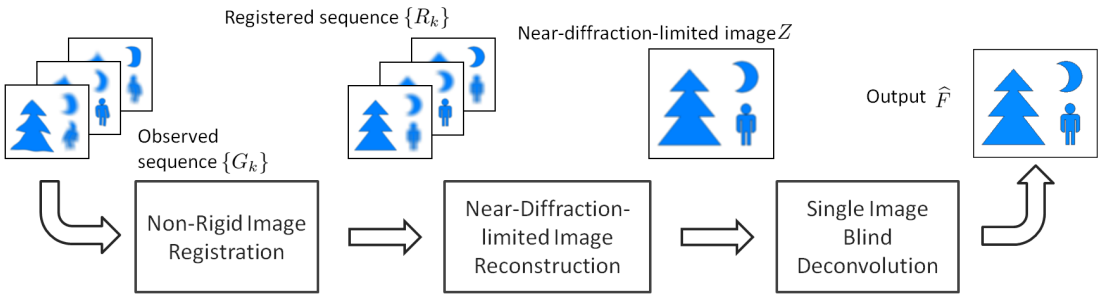


Figure 2.10: Block diagram for the proposed restoration framework.

Given an observed sequence $\{G_k\}$, the step A of our proposed approach registers each frame onto a fixed reference grid, generating a new sequence $\{R_k\}$ without geometric deformation. This process makes sure that for a given pixel located at a position say $\mathbf{x}$, the most correlated pixels in each frame across time are aligned at this same position. This step improves the performance of the subsequent temporal filtering.

Step B restores a single image Z from the registered $\{R_k\}$. For each local region, the sharpest patch (convolved by a diffraction-limited PSF) can be detected from the patch sequence using the gradient-based metric introduced in Section 2.1.2. Next, patch-wise temporal kernel regression is carried out to reduce the noise level. An image Z is then generated by fusing all the denoised sharp patches together. This image is still blurred by a diffraction-limited PSF, which can be approximately viewed as *spatially invariant*.

Finally, a single image blind deconvolution algorithm based on natural image statistics is implemented on Z to further remove the diffraction-limited blur and to enhance image quality. Details of each step are given in the following subsections.

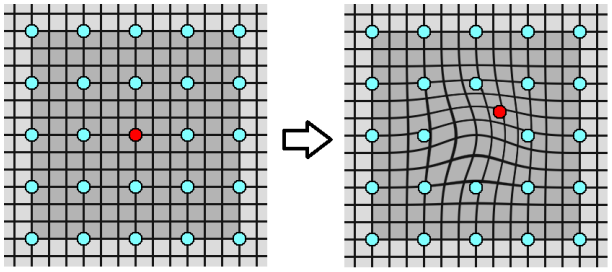


Figure 2.11: Shift of a control point and its influential area.

2.3.2 Non-Rigid Image Registration

In [51] we introduced a B-spline based registration algorithm to estimate the motion field in each observed frame. This method can be implemented in the proposed framework to remove geometric deformation. In this subsection we give a review of this registration algorithm. Assume G represents a given observed distorted image, and R denotes a reference image without turbulent deformation (which can be obtained by averaging the frame sequence [18]). A non-rigid deformation model can be employed to describe the geometric distortion between these two images. In this model, the complex motion is represented by the movement of m control points, whose initial positions $\hat{\mathbf{x}}_{0i} = (\hat{x}_{0i}, \hat{y}_{0i})^T$, $i = 1, \cdots, m$ are equally spaced on the reference image R. The displacement of all the control points on the given image G is denoted as the deformation vector: $\overrightarrow{\mathbf{p}} = [\Delta\hat{x}_1, \ldots, \Delta\hat{x}_m, \Delta\hat{y}_1, \ldots, \Delta\hat{y}_m]^T$, where $\Delta\hat{x}$ and $\Delta\hat{y}$ (also called deformation parameters) are the horizontal and vertical displacement from the initial position [78, 18]. The deformed position of any given pixel located at $\mathbf{x} = [x, y]^T$ from image R can then be described as:

$$\mathbf{W}(\mathbf{x}; \overrightarrow{\mathbf{p}}) = \mathbf{x} + \mathbf{A}(\mathbf{x})\overrightarrow{\mathbf{p}}, \tag{2.22}$$

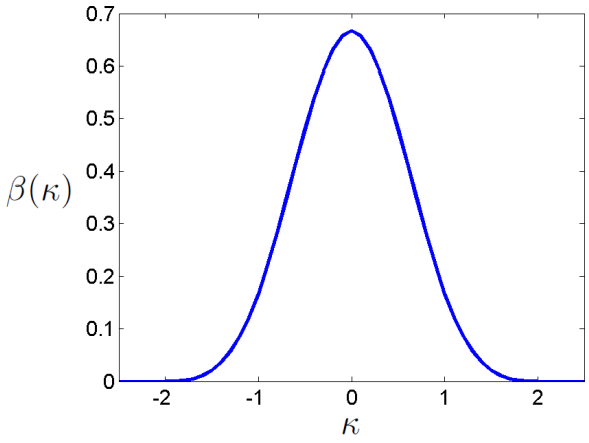


Figure 2.12: Spline function.

where $\mathbf{A}(\mathbf{x})$ denotes the basis function matrix for $\mathbf{x}$:

$$\mathbf{A}(\mathbf{x}) = \begin{bmatrix} c_1 & \dots & c_m & 0 & \dots & 0 \\ 0 & \dots & 0 & c_1 & \dots & c_m \end{bmatrix}, \tag{2.23}$$

and $\mathbf{A}(\mathbf{x})\overrightarrow{\mathbf{p}}$ is the motion vector, which is a linear combination of the movements of all control points. The weight (or spline basis) c_i is determined by the distance between $\mathbf{x}$ and $\hat{\mathbf{x}}_{0i}$ using B-spline functions (see Fig. **2.12**):

$$c_i = \beta\left(\frac{x - \hat{x}_{0i}}{\epsilon_x}\right)\beta\left(\frac{y - \hat{y}_{0i}}{\epsilon_y}\right). \tag{2.24}$$

$$\beta(\kappa) = \begin{cases} 2/3 - (1 - |\kappa|/2)\kappa^2 & \text{, if } 0 \le |\kappa| \le 1 \\ (2 - |\kappa|)^3/6 & \text{, if } 1 < |\kappa| < 2 \\ 0 & \text{, otherwise} \end{cases} \tag{2.25}$$

where ϵ_x and ϵ_y are the horizontal and vertical intervals between neighboring control points. This model guarantees local smoothness of the turbulence-caused motion field.

In the classic B-spline based registration approach, the deformation vector is

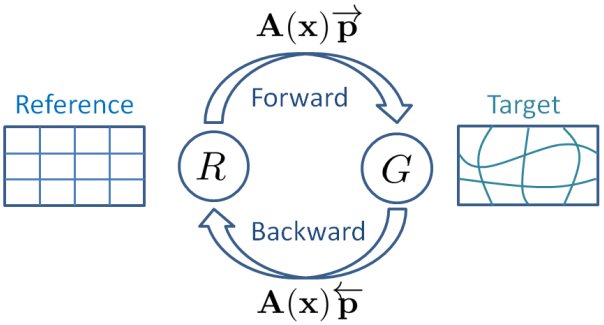


Figure 2.13: Symmetry constraint, where forward motion vector and backward motion vector are the inverse of each other.

estimated by minimizing the following cost function [78]:

$$C(\overrightarrow{\mathbf{p}}) = \sum_{\mathbf{x}} \left| G(\mathbf{W}(\mathbf{x}; \overrightarrow{\mathbf{p}})) - R(\mathbf{x}) \right|^2 \tag{2.26}$$

However, because the above formulation lacks prior constraint, the resulting estimate is unstable and gets easily stuck in local minima. To improve this method, a stabilization constraint is proposed in [18], which makes the estimated deformation parameters remain small in the regions that have less texture (low image gradient). However, in the present application, we know that the deformation caused by atmospheric turbulence is independent from image content. In other words, such stabilization constraint would bring unnecessary bias into the estimation.

Instead, to accurately estimate the deformation (motion) vectors from image grid R to G, we introduce a more natural symmetry constraint [79, 80] into the B-spline registration algorithm. This constraint is based on the important property that the registration should be symmetric or inverse consistent [79]. Let $\overrightarrow{\mathbf{p}}$ denote the deformation vector that transforms reference grid R into G (forward deformation), and let $\overleftarrow{\mathbf{p}}$ denote the inverse vector that transforms G into R (backward deformation). Then approximately[2] we should have: $\overrightarrow{\mathbf{p}} = -\overleftarrow{\mathbf{p}}$ (see Fig. **2.13**). Combining the two vectors

[2]The forward and backward motion vectors for each point in the physical world should be the exact

into one: $\mathbf{p}^T = [\overrightarrow{\mathbf{p}}^T, \overleftarrow{\mathbf{p}}^T]$, the proposed cost function to minimize becomes:

$$C(\mathbf{p}) = \sum_{\mathbf{x}} \left| G(\mathbf{W}(\mathbf{x}; \overrightarrow{\mathbf{p}})) - R(\mathbf{x}) \right|^2 + \sum_{\mathbf{x}} \left| R(\mathbf{W}(\mathbf{x}; \overleftarrow{\mathbf{p}})) - G(\mathbf{x}) \right|^2 + \gamma (\overrightarrow{\mathbf{p}} + \overleftarrow{\mathbf{p}})^T (\overrightarrow{\mathbf{p}} + \overleftarrow{\mathbf{p}}) \tag{2.27}$$

where scalar γ controls the effect of the soft symmetry constraint.

The Gauss-Newton method is used to minimize $C(\mathbf{p})$, and the update of the parameter set $\mathbf{p}$ can be derived as follows:

$$\mathbf{p}^{l+1} = \mathbf{p}^l - \mathbf{E}^{-1}\mathbf{b}, \tag{2.28}$$

where

$$\mathbf{E} = \begin{bmatrix} \overrightarrow{\mathbf{E}} + \gamma \mathbf{I} & \gamma \mathbf{I} \\ \gamma \mathbf{I} & \overleftarrow{\mathbf{E}} + \gamma \mathbf{I} \end{bmatrix}$$

$$\overrightarrow{\mathbf{E}} = \sum_{\mathbf{x}} \overrightarrow{\mathbf{d}}(\mathbf{x}) \overrightarrow{\mathbf{d}}(\mathbf{x})^T, \quad \overleftarrow{\mathbf{E}} = \sum_{\mathbf{x}} \overleftarrow{\mathbf{d}}(\mathbf{x}) \overleftarrow{\mathbf{d}}(\mathbf{x})^T$$

$$\overrightarrow{\mathbf{d}}(\mathbf{x})^T = \frac{\partial G(\mathbf{W}(\mathbf{x}; \overrightarrow{\mathbf{p}}^l))}{\partial \mathbf{W}} \mathbf{A}(\mathbf{x})$$

$$\overleftarrow{\mathbf{d}}(\mathbf{x})^T = \frac{\partial R(\mathbf{W}(\mathbf{x}; \overleftarrow{\mathbf{p}}^l))}{\partial \mathbf{W}} \mathbf{A}(\mathbf{x})$$

and

$$\mathbf{b}^T = \left[(\overrightarrow{\mathbf{b}} + \overrightarrow{\mathbf{p}}^l + \overleftarrow{\mathbf{p}}^l)^T, (\overleftarrow{\mathbf{b}} + \overrightarrow{\mathbf{p}}^l + \overleftarrow{\mathbf{p}}^l)^T \right]$$

$$\overrightarrow{\mathbf{b}} = \sum_{\mathbf{x}} \overrightarrow{\mathbf{d}}(\mathbf{x}) [G(\mathbf{W}(\mathbf{x}; \overrightarrow{\mathbf{p}}^l)) - R(\mathbf{x})]$$

$$\overleftarrow{\mathbf{b}} = \sum_{\mathbf{x}} \overleftarrow{\mathbf{d}}(\mathbf{x}) [R(\mathbf{W}(\mathbf{x}; \overleftarrow{\mathbf{p}}^l)) - G(\mathbf{x})].$$

In the above algorithm, the computational cost for calculating the matrices $\overrightarrow{\mathbf{E}}$ and $\overleftarrow{\mathbf{E}}$ is tremendous if it is implemented directly. Instead, we designed a fast implementation

opposite of each other. However, in the registration model the motion field is indexed by pixel locations instead of physical points. Thus, the motion vectors for the same position in the reference and the target image are not exactly the opposite of each other since they belong to slightly different points. So do the deformation vectors.

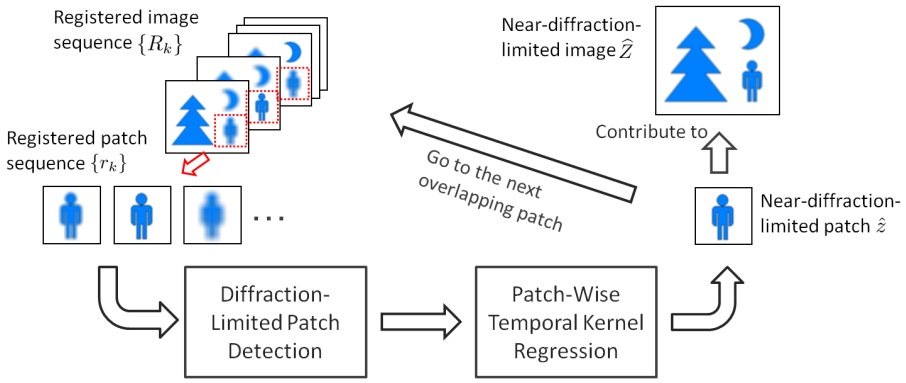


Figure 2.14: Block diagram for the near-diffraction-limited image reconstruction step.

method to alleviate this problem. Details can be found in [51]. Once the deformation vector $\mathbf{p}$ is estimated for each frame, a registered sequence $\{R_k\}$ is generated through bilinear interpolation.

2.3.3 Near-Diffraction-Limited Image Reconstruction

In this subsection, we estimate a *diffraction-limited* image Z from $\{R_k\}$, which can be globally deconvolved. A concise description of the reconstruction procedure is provided in Algorithm 1 (see diagram in Fig. **2.14**):

To better understand the procedure in Algorithm 1, let us consider a *patch-wise* imaging model under the isoplanatic scenario (see Fig. **2.14**):

$$\begin{aligned} g_k &= f \otimes h_k \otimes h + n_k \\ &= f \otimes h \otimes h_k + n_k \\ &= z \otimes h_k + n_k \end{aligned} \tag{2.29}$$

where g_k can denote any local patch of size $L \times L$ extracted from frame G_k, and its corresponding latent patch is denoted by f. Different from (2.21), turbulence-caused

Algorithm 1 Procedure for Restoring A Near-Diffraction-Limited Image from Registered Frames

1. Given a registered sequence $\{R_k\}$, divide each frame into $L \times L$ overlapping patches centered at each pixel, and calculate the intensity variance of each patch as a local sharpness measure.
2. For a patch sequence $\{r_k\}$ centered at location $\mathbf{x}$, detect the sharpest one $r_{\check{k}}$ by maximizing local sharpness measure (outliers need to be detected and excluded).
3. Set $r_{\check{k}}$ as a reference patch, and restore its center pixel value using temporal kernel regression. Assign this value to the pixel $Z[\mathbf{x}]$.
4. Go to the next pixel and return to step 2.

PSF h_k is now patch-wise constant and temporally changing. We call $z = f \otimes h$ the *diffraction-limited* patch, which is convolved by the space and time invariant h, and thus can be accurately deconvolved in a larger scene [38] (such as the whole image). n_k represents local noise, which is assumed to be zero-mean and IID.

The motion field estimated from the non-rigid image registration in Section 2.3.2 is smooth, and the registration process can be viewed approximately as patch-wise constant translational movement (see an example given in Fig. **2.15**). So the relationship between an observed local patch g_k and the corresponding registered one r_k extracted from R_k can be described as:

$$r_k = g_k \otimes \delta_{\Delta \mathbf{x}} \tag{2.30}$$

where $\delta_{\Delta \mathbf{x}}$ represents a 2-D Kronecker Delta function shifted by the local registration

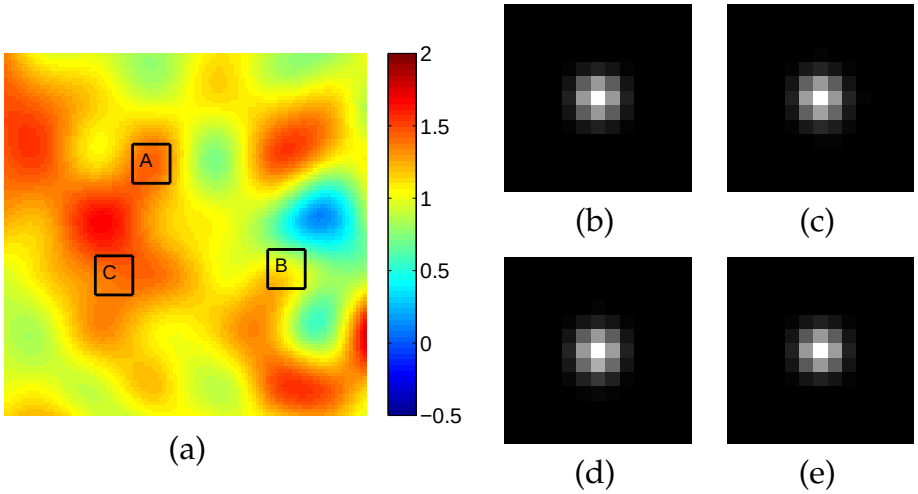


Figure 2.15: PSF distortion after the registration step. (a) Magnitude of a motion field estimated from a real frame distorted by air turbulence (video *Moon Surface* in Fig. **2.22**) using the proposed registration method. (b) a Gaussian shaped diffraction-limited PSF before alignment. (c)-(e) illustrate PSFs after the alignment using the local motion fields given in the squares A, B and C respectively from (a), where it can be observed that since the local movement could be treated as translational, the shapes of the PSFs are preserved.

vector $\mathbf{\Delta x}$. Hence, (2.29) and (2.30) can be combined as follows:

$$\begin{aligned} r_k &= z \otimes h_k \otimes \delta_{\mathbf{\Delta x}} + n_k \otimes \delta_{\mathbf{\Delta x}} \\ &= z \otimes \tilde{h}_k + \tilde{n}_k \\ &= r_k^* + \tilde{n}_k \end{aligned} \tag{2.31}$$

The PSF $\tilde{h}_k = h_k \otimes \delta_{\mathbf{\Delta x}}$ is simply a shifted version of h_k, which means the registration process preserves the shape (and spectrum) of local PSFs. Similarly, the shifted noise patch $\tilde{n}_k$ also has the same statistical properties as n_k. As mentioned before, the registration process increases the correlation among the pixels in the same position from different patches across time. Such correlation can be utilized for estimating the noise-free sequence $\{r_k^*\}$, as we will describe in Section 2.3.3.2.

2.3.3.1 Diffraction-Limited Patch Detection

As mentioned in earlier, once sufficient observations are collected, relatively sharp image patches that occasionally appear due to the turbulence variation can be found. Suppose a $\check{k}$-th patch is a diffraction-limited one: $r_{\check{k}} \approx z + \tilde{n}_{\check{k}}$, then simply denoising $r_{\check{k}}$ provides a good estimation of the patch z.

In practice, diffraction-limited patches can be detected through local sharpness metrics. Here we simply implement the gradient-based metric S_{grad} defined in Section 2.1.2. One problem is that sharp patch selection by maximizing this metric can be sensitive to outliers, such as those caused by objects incorrectly registered, or moving objects occasionally appearing in the scene (e.g. a bird flying through the field of view). Examples are illustrated in Fig. **2.16** (a), where some artifacts can be observed in flat regions (see patch A). These artifacts are caused by registration error, and can be strongly magnified (see (c)) after the consequential deblurring process (Section 2.3.4). In (e) we can observe extremely high values among the sharpness measures for patch A. Patch B gives another example (see (f)) where no outlier exists[3]. Histograms of these two patches are given in (g) and (h), where the sharpness values for the outlier-free patch have a roughly symmetric distribution, while outlier-contaminated distributions contain a long tail corresponding to high values.

To alleviate the outlier problem, Hampel's outlier identification method is employed [81]. We use a Gaussian model $\mathcal{N}(\eta, \sigma)$ to approximate the distribution of outlier-free samples (which are sharpness measurements in our case). A given sample

[3]Patch B contains strong structure (high SNR), and thus its sharpness measures have values much higher than patch A (see Fig. **2.16** (e) and (f)). This also explains why the registration accuracy is lower in patch A compared with patch B.

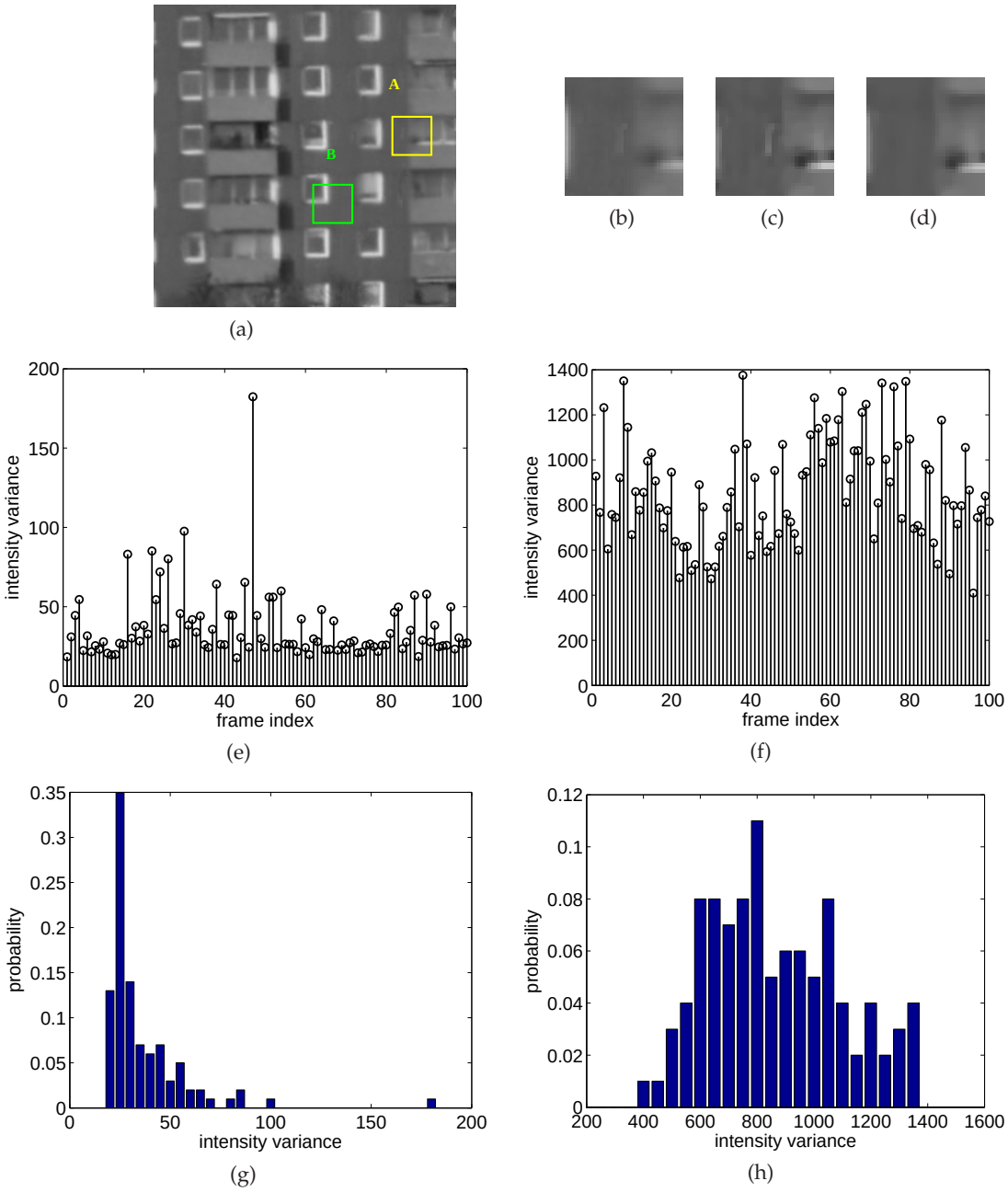


Figure 2.16: Outlier identification example. (a) A fused image without outlier removal, where patch A contains outlier pixels and patch B does not contain any outlier; (b) zoomed patch A from (a); (c) patch A after deblurring, where outlier effect is magnified; (d) deblurred patch A with outlier removal process; (e) intensity variance of patch A over 100 frames; (f) intensity variance of patch B over 100 frames; (g) histogram of (e); (h) histogram of (f).

s_k is identified as an outlier if

$$|s_k - \eta| > \varrho_{1-\alpha/2}\sigma \tag{2.32}$$

where ϱ_x is the x quantile of the $\mathcal{N}(0,1)$ distribution, and α denotes the confidence level. Hampel suggested using the median and median absolute deviation (MAD) method [82] to robustly estimate the data mean η and the standard deviation σ respectively [82]. In our experiment, we set $\varrho_{1-\alpha/2} = 6$ to strongly suppress the type I error. Any sample s_k that is above the threshold $\tau = 6\hat{\sigma}$ is detected as outlier and would be ignored when maximizing the local sharpness metric to select the diffraction-limited patch. In Fig. **2.16** (d) we can see that with outlier removal the artifacts caused by incorrect registration are successfully eliminated from the fused image.

2.3.3.2 Patch-wise Temporal Kernel Regression

To avoid possible artifacts that may appear in the subsequent deconvolution step, noise in the selected diffraction-limited patches needs to be suppressed. We formulate this denoising problem under a general temporal regression framework, where the imaging model is:

$$r_k = r_k^* + \tilde{n}_k \tag{2.33}$$

The value of a pixel at $\mathbf{x}$ in the l-th frame ($r_l^*[\mathbf{x}]$) can be estimated through zero-th order kernel regression [83, 84]:

$$\hat{r}_l^*[\mathbf{x}] = \arg\min_{r_l^*[\mathbf{x}]} \sum_k (r_k[\mathbf{x}] - r_l^*[\mathbf{x}])^2 U(\mathbf{x}; k, l), \tag{2.34}$$

where the weight $U(\mathbf{x}; k, l)$ is a positive function measuring the "similarity" between $r_l^*[\mathbf{x}]$ and $r_k^*[\mathbf{x}]$. One simple but effective way of defining $U(\cdot)$ is based on patch-wise

photometric distance and a Gaussian kernel function:

$$U(\mathbf{x};k,l)=\exp\left(\frac{-\|r_k^*-r_l^*\|^2}{L^2\mu^2}\right) \tag{2.35}$$

Here L^2 is the total number of pixels in the patch, and the scalar μ is called the *smoothing parameter* [83]. The noise-free photometric distance $\frac{1}{L^2}\|r_k^*-r_l^*\|^2$ can be unbiasedly estimated using the following function [85]:

$$\frac{\|r_k^*-r_l^*\|^2}{L^2}\approx\frac{\|r_k-r_l\|^2}{L^2}-2\sigma_n^2 \tag{2.36}$$

where the noise variance σ_n^2 can be estimated using, for example, the MAD method [82].

The solution to (2.34) is:

$$\hat{r}_l^*[\mathbf{x}]=\frac{\sum_k U(\mathbf{x};k,l)r_k[\mathbf{x}]}{\sum_k U(\mathbf{x};k,l)} \tag{2.37}$$

which is nothing but a filter generated by normalizing the kernel $U(\cdot)$. According to (2.35) the kernel value *within a patch* is independent of the pixel position and space-invariant, which means the spatial constancy of the PSF in the estimated patch $\hat{r}_l^*$ is preserved after the regression.

Of course, in this step we only estimate the center pixel in the selected $\check{k}$-th patch, which will be assigned to the output image Z according to the reconstruction procedure given in Algorithm 1. One important issue is that in the proposed regression, all the weights are positive, which means that when restoring the diffraction-limited value, the output is always more blurry than the observed $r_{\check{k}}[\mathbf{x}]$. That is why we call the restored image *near-diffraction-limited*.

2.3.4 Blind Deconvolution

Finally, a single image deblurring algorithm is required as a post-process to deconvolve the near-diffraction-limited image Z. The degradation model is:

$$Z = F \otimes h + \varepsilon \tag{2.38}$$

where ε represents error caused by the process generating the estimate of Z. Such blind deconvolution algorithm can be described generally using the following:

$$< \widehat{F}, \hat{h} >= \arg\min_{F,h} \|Z - h \otimes F\|^2 + \lambda_1 R_f(F) + \lambda_2 R_h(h), \tag{2.39}$$

where R_f and R_h are the regularization terms based on prior knowledge about the latent sharp image F and the PSF h. Recent research on natural image statistics has shown that image gradients obey *heavy-tailed* distributions that have most of their mass on small values but give significantly more probability to large values than Gaussian distributions [7]. Based on these studies, several *sparse* regularization methods have been introduced and have achieved great success in solving the blind deconvolution problem [7, 37, 8]. One example is the method proposed by Shan *et al.* [8], which is directly implemented in this step to calculate a final output. This method uses basically the same estimation form as (2.39) with the following sparse regularization term:

$$R_f(F) = \|\rho(F_x) + \rho(F_y)\|_1 \tag{2.40}$$

where F_x and F_y denote the derivatives of F in horizontal and vertical directions respectively, and

$$\rho(\kappa) = \begin{cases} -\theta_1|\kappa| & \kappa \le l_t \\ -(\theta_2\kappa^2 + \theta_3) & \kappa > l_t \end{cases} \tag{2.41}$$

Here l_t, θ_1, θ_2 and θ_3 are all fixed parameters [8]. Sparsity is also utilized in regularizing h [8]:

$$R_h(h) = \|h\|_1 \tag{2.42}$$

The cost function (2.39) is optimized by alternating the estimation of f and h [37, 8].

In what follows, we used the default parameter settings as described in the authors' project page[4] for [8] except the noise level parameter 'noiseStr', which is chosen in the range [0.01, 0.05] according to the actual noise level observed in the given data. We refer interested readers to [8] for details.

2.3.5 Experimental Results

In this subsection we will illustrate the performance of the proposed approach using both simulated and real image sequences. Throughout all the experiments, the intervals of the control points in the registration step are set as $\epsilon_x = \epsilon_y = 16$ pixels, and the symmetry constraint parameter: $\gamma = 5000$. In implementing the restoration step in Section 2.3.3, we set the patch size $L = 9$. Results of the lucky-region algorithm from [76], the multi-frame reconstruction approach from [51], and the EFF approach from [36] are also shown for comparison[5].

[4] http://www.cse.cuhk.edu.hk/~leojia/projects/motion_deblurring/index.html.

[5] In this section, the outputs of method [51] are generated using the original code. The EFF outputs are directly from the authors of [36]. The outputs of method [76] are produced by our own implementation.

2.3.5.1 Simulated Experiments

To quantitatively evaluate the algorithm performance, a set of image sequences with different degrees of turbulence and noise are generated. The latent sharp image[6] (300×300) is shown in Fig. **2.17**. The sequences are produced using the imaging model described in (2.21) and (2.29), where motion fields, spatially variant PSFs and spatially invariant diffraction-limited PSF are required. To produce the motion fields, we first randomly generate a set of deformation vectors through a Gaussian distribution, and then calculate a motion field for each frame through the B-spline model (2.22)-(2.25). The turbulence strength is determined by the variance of the Gaussian distribution. The spatially variant PSFs are simulated also using a Gaussian function, where the variance of a local PSF is proportional to the magnitude of the corresponding local motion vector. The diffraction-limited PSF is generated using a disc function. Three degrees (weak, medium and strong) of turbulence are produced (see Fig. **2.18** (a)-(c)). Also different levels of white Gaussian noise (with variance $\sigma_n^2 = 1, 9, 25$) are added into the sequences. Each sequence contains 100 frames.

Due to the space limit, we only provide results of three sets of sequences with noise variance $\sigma_n^2 = 1$ in Fig. **2.18**. It is clear that the proposed approach significantly improved the visual quality, and recovered many high frequency details of the image content.

Table **2.1** gives the Peak Signal-to-Noise Ratio (PSNR) values for all the outputs with 4 different restoration algorithms and the averaged PSNR values of each input sequence. It can be seen that the proposed approach outperforms in all test sequences in

[6]The pixel intensity range here is $[0, 255]$.

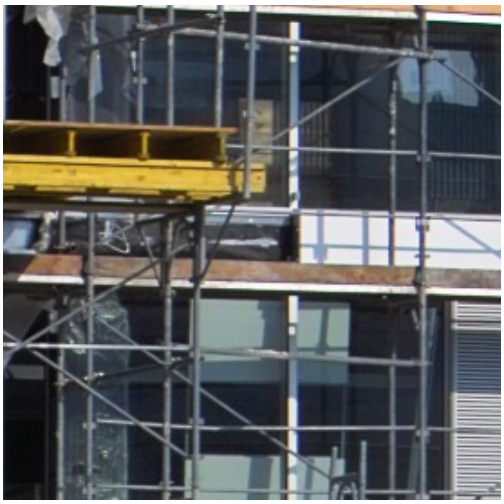


Figure 2.17: Latent sharp image used for simulation.

Table 2.1: Performance of the restoration approaches evaluated in PSNR values (dB).

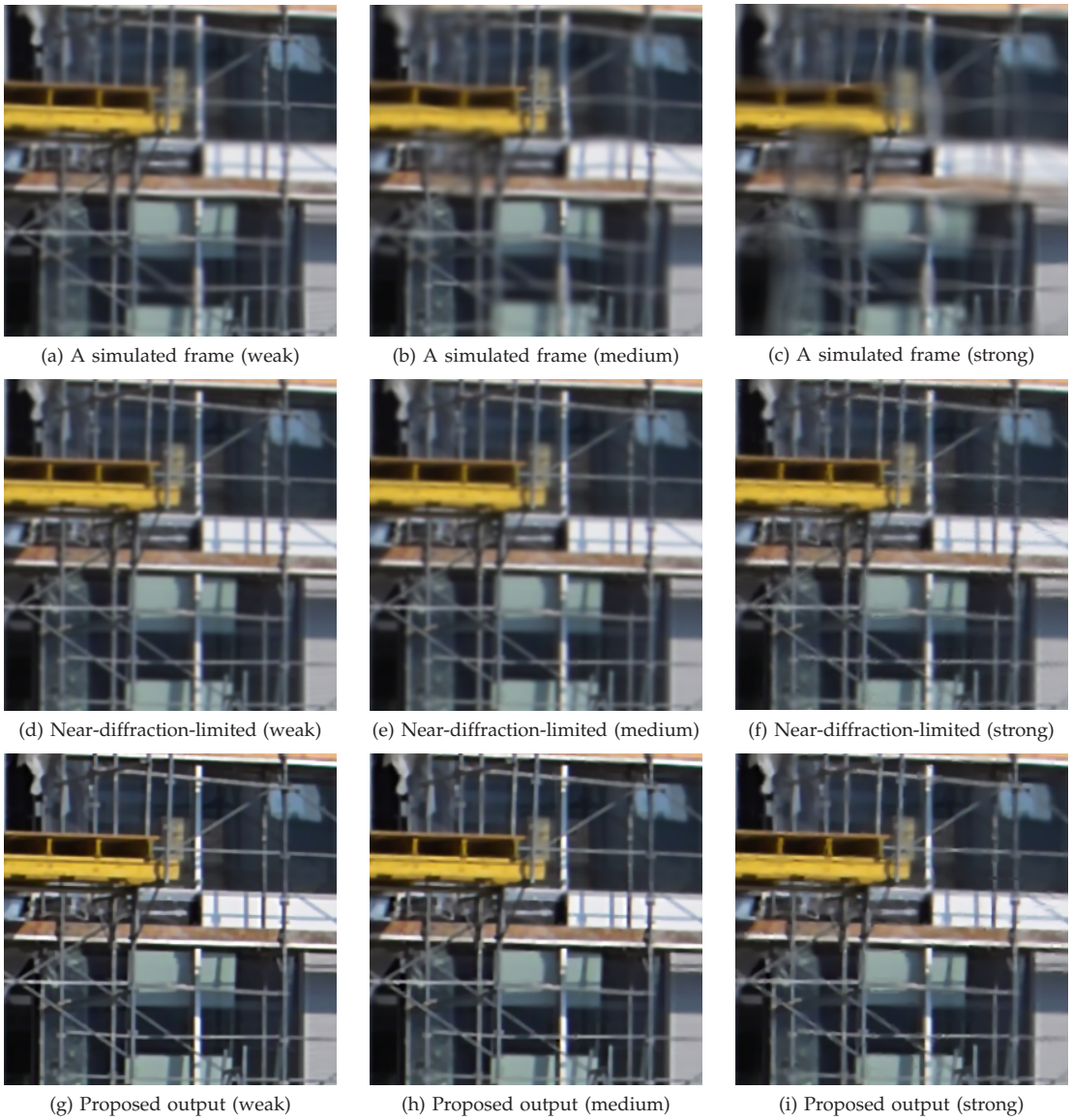
Method	$\sigma_n^2 = 1$	$\sigma_n^2 = 9$	$\sigma_n^2 = 25$
	weak / medium / strong	weak / medium / strong	weak / medium / strong
Proposed	**23.52** / **23.17** / **22.79**	**23.47** / **23.11** / **22.77**	**23.35** / **23.10** / **22.60**
Proposed NDL	22.22 / 21.67 / 21.64	22.24 / 21.70 / 21.63	22.29 / 21.74 / 21.62
Lucky Region [76]	22.44 / 21.32 / 21.28	22.40 / 21.29 / 21.27	22.33 / 21.23 / 21.21
Method [51]	21.80 / 20.67 / 18.81	21.77 / 20.59 / 18.80	21.67 / 20.52 / 18.77
EFF [36]	21.29 / 20.15 / 18.89	21.85 / 20.23 / 18.86	21.71 / 20.24 / 18.85
Averaged Input	20.67 / 19.33 / 18.06	20.61 / 19.28 / 18.03	20.49 / 19.19 / 17.96

terms of PSNR. Results of the near-diffraction-limited (NDL) images are also provided in this table, from which we can tell that the final blind-deconvolution step generally increases the PSNR by $1 \sim 1.5$ dB.

2.3.5.2 Real Video Experiments

Several real videos are tested to illustrate the performance of the proposed restoration framework in practical situations[7]. We first show results on data taken

[7]Videos and experimental results are given on the web page: http://users.soe.ucsc.edu/~xzhu/doc/turbulence.html.



(a) A simulated frame (weak) (b) A simulated frame (medium) (c) A simulated frame (strong)

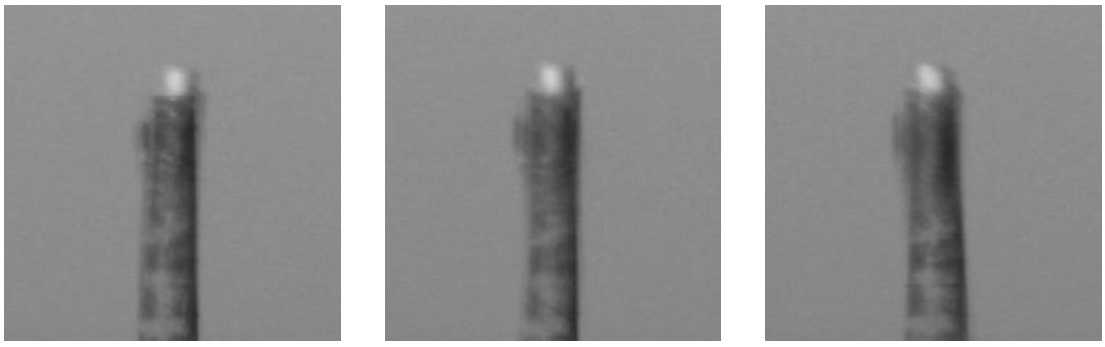
(d) Near-diffraction-limited (weak) (e) Near-diffraction-limited (medium) (f) Near-diffraction-limited (strong)

(g) Proposed output (weak) (h) Proposed output (medium) (i) Proposed output (strong)

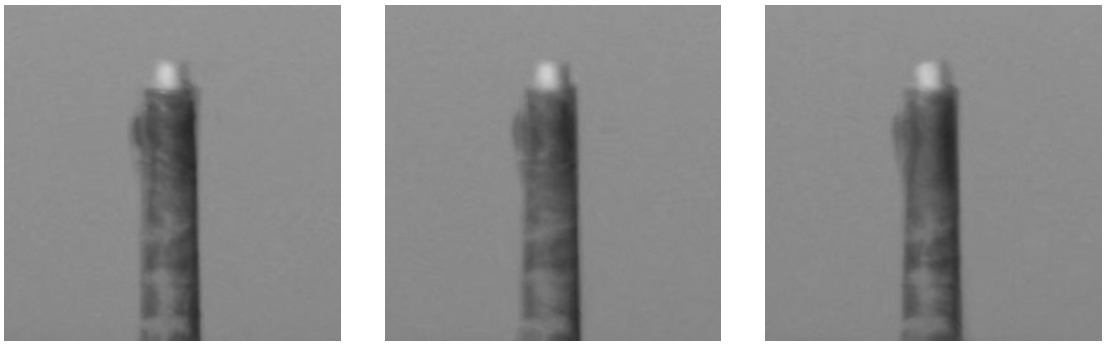
Figure 2.18: Simulated experiments. (a)-(c) Simulated frames with three turbulence levels (weak, medium, strong). (d)-(f) Near-diffraction-limited images from the three test sequences with noise variance $\sigma_n^2 = 1$ and turbulence level weak, medium and strong, respectively. (g)-(i) Final outputs deblurred from (d)-(f).

under controlled but real conditions. The two sets of videos (see *Chimney* and *Building* in Fig. **2.19**-**2.21**) of size 237×237 were captured through hot air exhausted by a building's vent, which could be closed to take "ideal" images of the same scene [36]. Each sequence consists of 100 frames degraded by strong turbulence effects (see Fig. **2.19** (a)-(c) and Fig. **2.21** (b)). In the *Chimney* experiment, some registration results are also given in Fig. **2.19** (d)-(f), where we can see geometric deformation with respect to the averaged image Fig. **2.20** (a) has been effectively removed. From the output images in Fig. **2.20** (c), (d), (f) we can observe that EFF (c) and the proposed method (f) provide the best restoration results and faithfully recover details of the object. The proposed result (f) looks even sharper, while the EFF output (c) contains some halo artifacts in the edge regions. Similar restoration results can be observed in the *Building* experiment shown in Fig. **2.21**. The near-diffraction-limited image generated from the *Chimney* sequence is also provided in Fig. **2.20** (e).

Additional sets of video data taken from long-distance imaging systems are processed next. The first set of images ($410 \times 380 \times 80$) show the moon's surface taken from a ground-based telescope (see Fig. **2.22** (a)). From (b) we can see that though the output image of [76] looks slightly sharper than one of the observed frames, it is still quite blurry probably due to the diffraction-limited blur and the limited number of frames. The method in [51] provides a better result but with some details (like small craters) vanished (Fig. **2.22** (c)). The proposed method gives a significant improvement in visual quality (Fig. **2.22** (d)). It successfully removed blur and meanwhile recovered many small craters on the surface (Fig. **2.22** (h)) that can hardly be seen from either original frame (Fig. **2.22** (e)), or the outputs of the other two methods (Fig. **2.22** (f),



(a) Observed frame 1 (b) Observed frame 2 (c) Observed frame 3

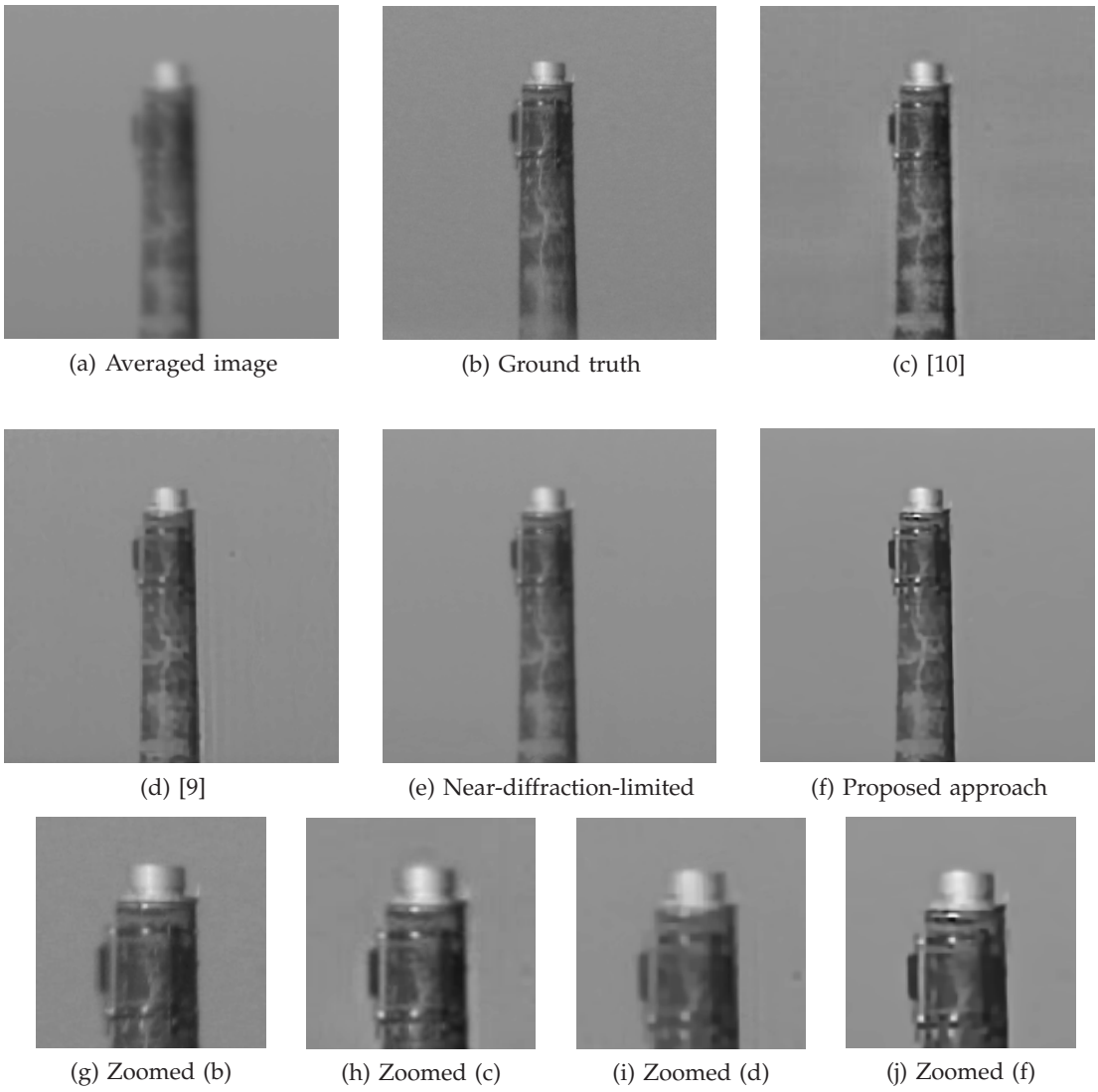


(d) Registered frame 1 (e) Registered frame 2 (f) Registered frame 3

Figure 2.19: Image registration results using 100 frames taken from the video *Chimney* distorted by real atmospheric turbulence.

(g)).

The scene of the next video stream consists of a water tower located above the ground, imaged at a (horizontal) distance of 2.4 kilometers. The (300×220) video is quite noisy and highly blurred (due to long exposure time). 80 frames were taken from the video to produce the result shown in Fig.**2.23**. Again, the lucky region method did not provide much improvement in the result and slightly changed the object shape (Fig. **2.23** (b)). The reconstruction method in [51] slightly increased the sharpness (Fig. **2.23** (c)). The output of the proposed method looks much sharper and clean, with many details well restored (Fig. **2.23** (d)).



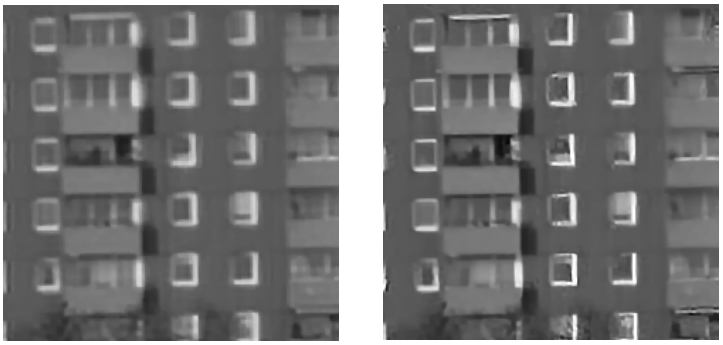
(a) Averaged image (b) Ground truth (c) [10]

(d) [9] (e) Near-diffraction-limited (f) Proposed approach

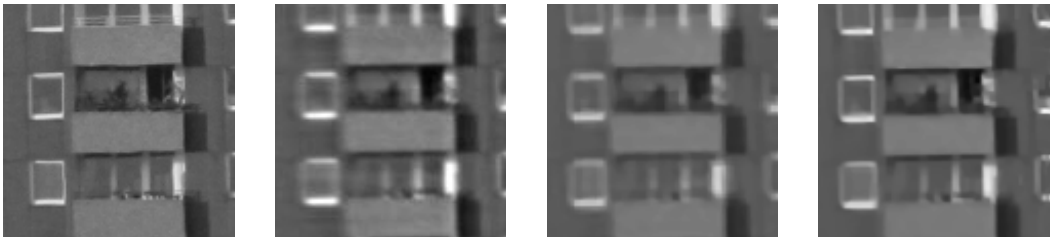
(g) Zoomed (b) (h) Zoomed (c) (i) Zoomed (d) (j) Zoomed (f)

Figure 2.20: Image reconstruction results using 100 frames taken from the video *Chimney* distorted by real atmospheric turbulence.

(a) Ground truth (b) One observed frame (c) [10]

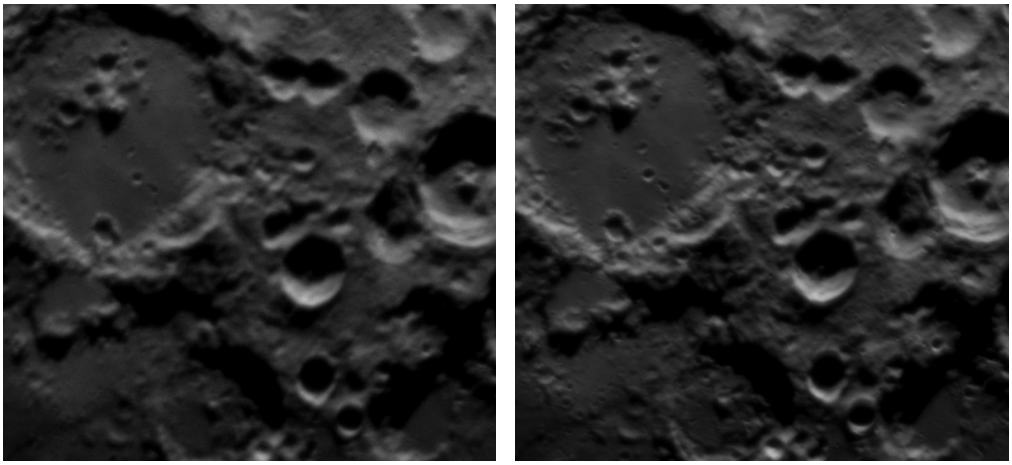


(d) [9] (e) Proposed approach



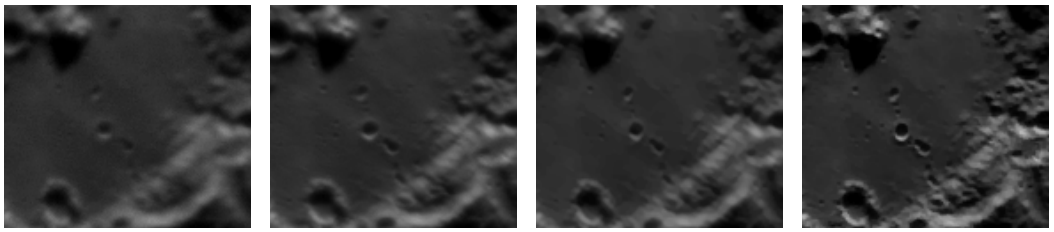
(f) Zoomed (a) (g) Zoomed (c) (h) Zoomed (d) (i) Zoomed (e)

Figure 2.21: Image reconstruction results using 100 frames taken from the video *Building* distorted by real atmospheric turbulence.

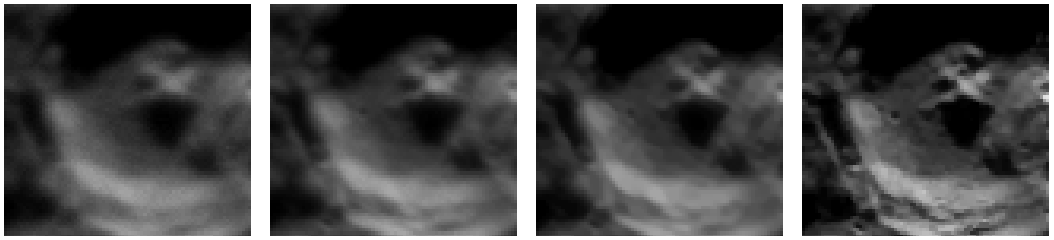


(a) One observed frame (b) [17]

(c) [9] (d) Proposed approach

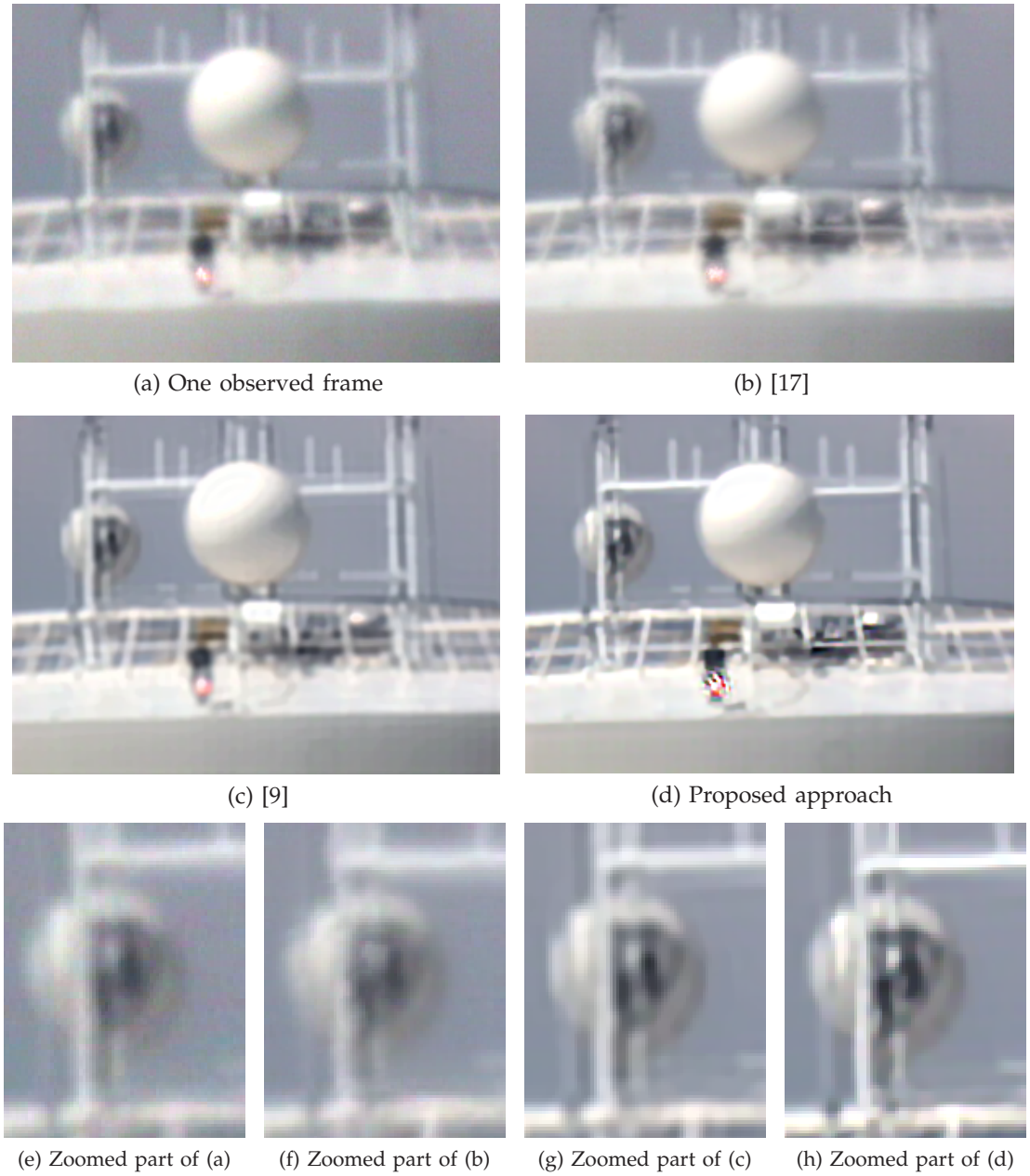


(e) Zoomed part of (a) (f) Zoomed part of (b) (g) Zoomed part of (c) (h) Zoomed part of (d)



(i) Zoomed part of (a) (j) Zoomed part of (b) (k) Zoomed part of (c) (l) Zoomed part of (d)

Figure 2.22: Image reconstruction result using 80 frames taken from the video *Moon Surface* distorted by real atmospheric turbulence.



(a) One observed frame (b) [17]

(c) [9] (d) Proposed approach

(e) Zoomed part of (a) (f) Zoomed part of (b) (g) Zoomed part of (c) (h) Zoomed part of (d)

Figure 2.23: Image reconstruction result using 80 frames taken from the video *Water Tower* distorted by real atmospheric turbulence.

2.3.6 Discussions

In this section we proposed a approach for restoring a single high-quality image from an image sequence distorted by air turbulence. The proposed algorithm first registers the frames to suppress geometric deformation using B-spline based non-rigid image registration, which incorporates a symmetry constraint to effectively improve the estimation accuracy. Next, a regression-based process is carried out to produce an image convolved with a near-diffraction-limited PSF, which can be viewed as spatially invariant. Finally, a blind deconvolution algorithm is implemented to remove diffraction-limited blur from the fused image to generate the final output. Experiments using controlled and real data illustrate that this approach is capable of alleviating geometric deformation and space-time varying blur caused by turbulence, recovering unprecedented details of the scene and significantly improving visual quality.

Another technique which may be related to the present work is *seeing through water* [3, 20]. Compared with air turbulence, in most cases the geometric warping effect induced by water is much stronger, but the blur effect is relatively milder. In the following experiments, we replace the image registration step in Section 2.3.2 with a robust iterative registration algorithm from a seeing through water approach [3]. The sequence *Chimney* with severe turbulence motion is tested. Results are given in Fig. **2.24**, where (a) shows the direct output of method [3], which is highly blurry since it is just the temporal mean of the registered sequence. We then take the registered sequence to generate a near-diffraction-limited image (b) using the temporal regression step described in Section 2.3.3, this intermediate result is much sharper than (a). The final deblurred output is given in (c), which is visually very close to the proposed result

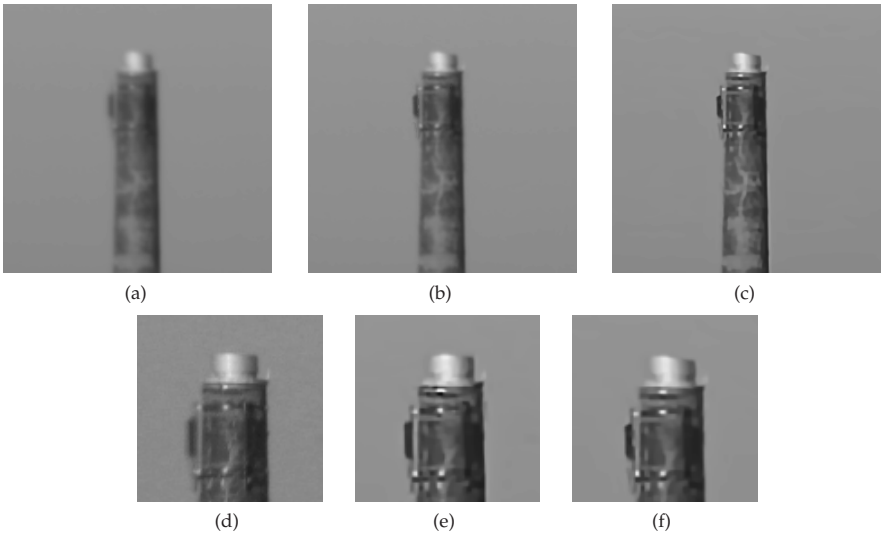


Figure 2.24: Experiments using seeing through water method [3]. Video *Chimney* distorted by real atmospheric turbulence is tested. (a) Output of method [3]. (b) Near-diffraction-limited image generated using the sequence registered by [3]. (c) Image deblurred from (b). (d) Zoomed ground truth. (e) Zoomed proposed result (same as Fig. **2.20** (j)). (f) Zoomed (c).

given in Fig. **2.20** (f) except for a mild shape change at the top part of the Chimney. Similar experiments using test video *Building* are illustrated in Fig. **2.25**. Probably the robust registration algorithm addressing water distortions is capable of correcting strong geometric deformations quite well, but it seems that the registration step in Section 2.3.2 is sufficient for handling most air induced motion.

2.4 Summary

In this chapter we analyzed the behavior of a local gradient based sharpness metric in the presence of noise with spatially constant strength, and proposed two restoration systems designed to remove spatially varying blur, including a focus-stacking algorithm for multi-focus image fusion and an air turbulence removal approach

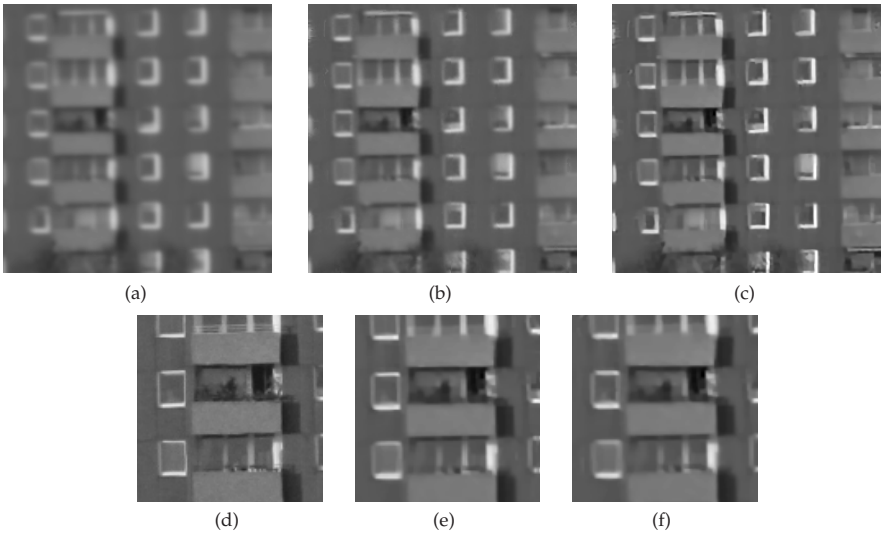


Figure 2.25: Experiments using seeing through water method [3]. Video *Building* distorted by real atmospheric turbulence is tested. (a) Output of method [3]. (b) Near-diffraction-limited image generated using the sequence registered by [3]. (c) Image deblurred from (b). (d) Zoomed ground truth. (e) Zoomed proposed result (same as Fig. **2.21** (i)). (f) Zoomed (c).

for long-distance imaging systems. Both systems utilize the gradient based metric to select the sharpest local patches. The focus-stacking algorithm employs an α matting algorithm to correct fusion error caused by the metric, and experiments showed that this refinement can reduce the amount of fusion artifacts and preserve boundary regions. The air turbulence removal approach first registers each frame to suppress geometric deformation. Then, it uses a fusion process based on the gradient metric to produce a single image that can be viewed as being convolved with a spatially invariant blur. Finally, a blind deconvolution algorithm is implemented to generate the sharp output. Experiments using real data illustrated that this approach is capable of alleviating blur and geometric deformation caused by turbulence, recovering details of the scene and significantly improving visual quality.

Chapter 3

Assessing Blur *and* Noise in Images

Abstract - In this chapter we propose a new absolute metric extended from the gradient sharpness metric described in Chapter 2. Different from Chapter 2, here both blur and noise are considered and the new metric performs reasonably to these two factors. In other words, the value of this metric drops as the given image becomes either more blurry or more noisy. Though it is still an absolute sharpness metric, the proposed measure is highly correlated with HVS and thus could be used in optimizing denoising filters so that bias (blur) and variance (noise) of the filtered image can be automatically balanced generating visually the best denoised output.

3.1 Analysis on Gradient-based Sharpness Metric

In Chapter 2 we analyzed the gradient based sharpness metric S_{grad} in the presence of noise with spatially constant strength. Generally speaking, the value of S_{grad} monotonically drops as the test image (or image patch) gets more and more blurry, though the monotonicity of S_{grad} could be affected by its variation (caused by

random noise) to a certain degree. This metric is utilized in selecting sharpest patches in fusion based image restoration algorithms. However, if the noise strength is not stable in a given application, S_{grad} then cannot be applied to assess image quality.

Recall that S_{grad} measures local gradient energy. Consider a small patch w within a given image $g[x, y]$. If the patch is contaminated by noise, then we have

$$\begin{aligned} S_{grad} &= \frac{1}{N}\left(\mathbf{g}_x^T\mathbf{g}_x + \mathbf{g}_y^T\mathbf{g}_y\right) \\ &= \frac{1}{N}\left(\left(\mathbf{g}_x^* + \mathbf{n}_x\right)^T\left(\mathbf{g}_x^* + \mathbf{n}_x\right) + \left(\mathbf{g}_y^* + \mathbf{n}_y\right)^T\left(\mathbf{g}_y^* + \mathbf{n}_y\right)\right) \\ &= \frac{1}{N}\left(\mathbf{g}_x^{*T}\mathbf{g}_x^* + 2\mathbf{g}_x^{*T}\mathbf{n}_x + \mathbf{n}_x^T\mathbf{n}_x + \mathbf{g}_y^{*T}\mathbf{g}_y^* + 2\mathbf{g}_y^{*T}\mathbf{n}_y + \mathbf{n}_y^T\mathbf{n}_y\right) \end{aligned} \tag{3.1}$$

where, again $\mathbf{g}_x^*$ and $\mathbf{g}_y^*$ denote the noise-free version of the $N \times 1$ local gradient vectors $\mathbf{g}_x$ and $\mathbf{g}_y$. N represents total number of pixels within w. $\mathbf{n}_x$ and $\mathbf{n}_y$ are the horizontal and vertical components of the gradient of the zero-mean IID noise vector $\mathbf{n}$ with its variance denoted as σ^2. Specifically, $\mathbf{n}_x = \mathbf{D}_x\mathbf{n}$ and $\mathbf{n}_y = \mathbf{D}_y\mathbf{n}$, where $N \times N$ derivative filtering matrices $\mathbf{D}_x$ and $\mathbf{D}_y$ are the gradient filters.

Statistically, because $E(\mathbf{n}) = \mathbf{0}$ and $\text{cov}(\mathbf{n}) = \sigma^2\mathbf{I}$, the expected noise gradient vectors are:

$$E(\mathbf{n}_x) = \mathbf{D}_x E(\mathbf{n}) = \mathbf{0}, \quad E(\mathbf{n}_y) = \mathbf{D}_y E(\mathbf{n}) = \mathbf{0}, \tag{3.2}$$

and the expected value of the energy of the noise derivative vector becomes:

$$\begin{aligned} E(\mathbf{n}_x^T\mathbf{n}_x) &= E\left(\mathbf{n}^T\mathbf{D}_x^T\mathbf{D}_x\mathbf{n}\right) \\ &= E\left(tr\left(\mathbf{D}_x\mathbf{n}\mathbf{n}^T\mathbf{D}_x^T\right)\right) \\ &= \sigma^2 tr(\mathbf{D}_x\mathbf{D}_x^T) \end{aligned} \tag{3.3}$$

The value of the trace $tr(\mathbf{D}_x\mathbf{D}_x^T)$ depends upon the specific filter used for gradient calculation. It can be shown that if we choose the filter in (2.9), then $tr(\mathbf{D}_x\mathbf{D}_x^T) = \xi N$,

where $\xi = \frac{1}{2}$. If we choose the following gradient filters:

$$\tfrac{1}{8} \times \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \quad \tfrac{1}{8} \times \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}, \tag{3.4}$$

then $\xi = \frac{3}{16}$. Similarly, we have $E(\mathbf{n}_y^T \mathbf{n}_y) = \xi N \sigma^2$.

Because vectors $\mathbf{g}_x^*$ and $\mathbf{g}_y^*$ are deterministic, the expected value of S_{grad} becomes

$$\begin{aligned} E(S_{grad}) &= \frac{1}{N} \left(\mathbf{g}_x^{*T} \mathbf{g}_x^* + \mathbf{g}_y^{*T} \mathbf{g}_y^* + 2\mathbf{g}_x^{*T} E(\mathbf{n}_x) + 2\mathbf{g}_y^{*T} E(\mathbf{n}_y) + E(\mathbf{n}_x^T \mathbf{n}_x) + E(\mathbf{n}_y^T \mathbf{n}_y) \right) \\ &= \frac{1}{N} \left(\mathbf{g}_x^{*T} \mathbf{g}_x^* + \mathbf{g}_y^{*T} \mathbf{g}_y^* + 2\xi N \sigma^2 \right). \end{aligned} \tag{3.5}$$

If the noise variance increases, according to the above equation the value of S_{grad} should also be raised. This property would cause confusion in image quality assessment. For example, when the value of S_{grad} rises, we cannot tell whether such change is caused by an enhancement of image sharpness (quality improvement) or by an increase of noise strength (quality decay). Similar problems happen in most local image sharpness metrics [47].

In this chapter, we design a new metric that is capable of measuring the amount of *true image content*[1] in the presence of noise. Its value should monotonically drop as the image quality deteriorates (more blurry or more noisy). Meanwhile, the metric's sensitivity to random noise and blur should also be consistent to HVS. We will show later that such metric is important to image restoration algorithms, and that it can be

[1] By *true image content* we refer simultaneously to sharpness and local contrast, as manifested by visually salient geometric structures such as edges etc., which convey information about the nature of the physical objects in the scene. In this context, high frequency content such as that introduced by noise, or low frequency content such as that produced by blur of various types are not considered *true* image content.

used for automatic image filter optimization. Before designing new metrics, let us first analyze the spectral decomposition of S_{grad}.

3.1.1 SVD of Gradient Matrix

We define a gradient matrix $\mathbf{G} = [\mathbf{g}_x, \mathbf{g}_y]$, and its corresponding gradient covariance matrix $\mathbf{C} = \mathbf{G}^T\mathbf{G}$. It is well known that important information about the content of the image patch w can be derived from the gradient matrix $\mathbf{G}$ or the gradient covariance matrix $\mathbf{C}$. In particular, we can calculate the locally dominant orientation by computing the (compact) Singular Value Decomposition (SVD) of $\mathbf{G}$ [86, 87]:

$$\mathbf{G} = \mathbf{U}\mathbf{S}\mathbf{V}^T = \mathbf{U} \begin{bmatrix} s_1 & 0 \\ 0 & s_2 \end{bmatrix} \begin{bmatrix} \mathbf{v}_1 & \mathbf{v}_2 \end{bmatrix}^T \tag{3.6}$$

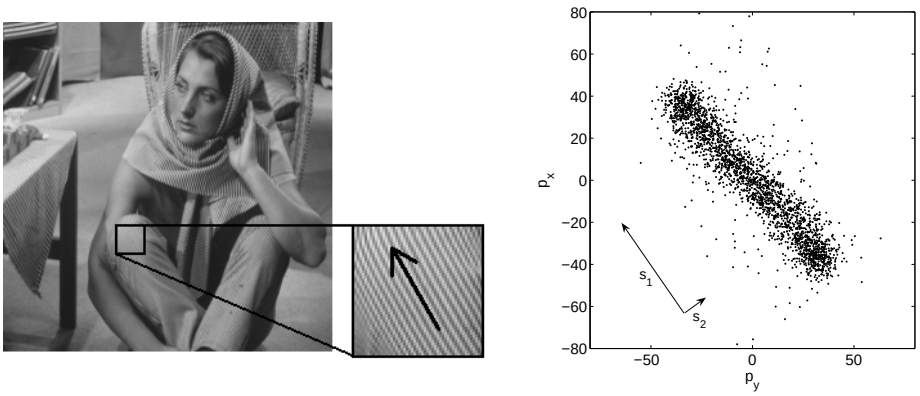
where $\mathbf{U}$ and $\mathbf{V}$ are both orthonormal matrices. The column vector $\mathbf{v}_1$ represents the dominant orientation of the local gradient field. Correspondingly, the second singular vector $\mathbf{v}_2$ (which is orthogonal to $\mathbf{v}_1$) will describe the dominant "edge orientation" of this patch. The singular values $s_1 \geq s_2 \geq 0$ represent the strength in the directions $\mathbf{v}_1$ and $\mathbf{v}_2$, respectively (see Fig. **3.1**).

The above quantities can equivalently be measured using the eigenvectors of $\mathbf{C}$, because

$$\mathbf{C} = \mathbf{V}\mathbf{S}^T\mathbf{S}\mathbf{V}^T = \mathbf{V} \begin{bmatrix} s_1^2 & 0 \\ 0 & s_2^2 \end{bmatrix} \mathbf{V}^T. \tag{3.7}$$

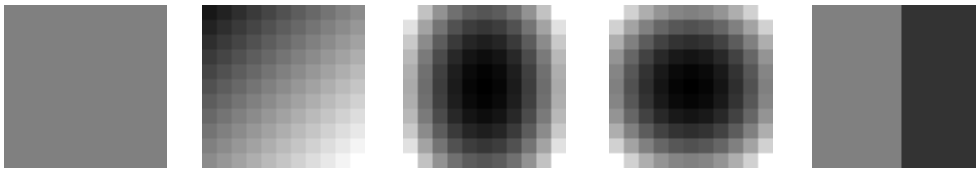
According to Equation (2.14) the relationship between the singular values and S_{grad} can be derived as:

$$NS_{grad} = tr\left(\mathbf{C}\right) = tr\left(\mathbf{V}\mathbf{S}^T\mathbf{S}\mathbf{V}^T\right) = s_1^2 + s_2^2. \tag{3.8}$$



(a) (b)

Figure 3.1: An example of local dominant orientation estimation. (b) plots the gradient of each pixel within the chosen patch in (a). s_1 and s_2 represent the energy in the dominant orientation and its perpendicular direction respectively.



(a) Flat (b) Linear (c) Quadratic (Anisotropic) (d) Quadratic (Isotropic) (e) Edge

Figure 3.2: Types of patches that are used in the experiments throughout this section. Gray levels are normalized to the range [0,1].

Ignoring the fixed window size N, we can see that s_1^2 and s_2^2 represent a decomposition of S_{grad}, and the decomposition is adaptive to the local image structure. So similar to the behavior of S_{grad}, the singular values s_1 and s_2 should be sensitive to blurring, and therefore, may be used to define a sharpness metric. To gain some useful intuition, we analyze the behavior of s_1 and s_2 on several types of idealized and noise-free patches which include flat, linear, quadratic, and edge regions (shown in Fig. **3.2**).

In the flat case, all points within the patch share a common intensity value:

$$g[x, y] = c \tag{3.9}$$

Both $g_x[x, y]$ and $g_y[x, y]$ are equal to 0 for $[x, y] \in w$, and $s_1 = s_2 = 0$. Naturally,

ignoring boundary effects arising from the finite nature of the window, a flat patch remains unchanged after being blurred. In what follows, we will apply a space invariant Gaussian blur function with a growing spread to the canonical regions shown in Fig. **3.2**, and observe how the singular values behave. In this sense, the flat region can be thought of as the asymptotic result as the spread of the blur function (or equivalently the strength of the blur) is made arbitrarily large.

In the linear patch, the gray value of each point can be modeled as:

$$g[x, y] = a(x \cos\theta + y \sin\theta) + b \tag{3.10}$$

where a decides the slope, θ decides the orientation, and b is the bias. It can be deduced that s_1 and s_2 have the following values:

$$s_1 = a\sqrt{N}, \quad s_2 = 0 \tag{3.11}$$

Both s_1 and s_2 are independent from the orientation, and s_1 is proportional to the slope given a fixed patch size, while s_2 remains at zero.

The quadratic patch is modeled as:

$$g[x, y] = a_1(x - x_o)^2 + a_2(y - y_o)^2 \tag{3.12}$$

where (x_o, y_o) is the center point. This kind of patch is called *isotropic* when $a_1 = a_2$, and *anisotropic* otherwise (see Fig. **3.2** (c) and (d)). The singular values of its gradient matrix are:

$$\begin{aligned} s_1 &= a_{max}\sqrt{\tfrac{N(N-1)}{3}} \\ s_2 &= a_{min}\sqrt{\tfrac{N(N-1)}{3}} \end{aligned} \tag{3.13}$$

where

$$\begin{aligned} a_{max} &= max(a_1, a_2) \\ a_{min} &= min(a_1, a_2) \end{aligned} \tag{3.14}$$

Here s_1 and s_2 reflect the values of a_1 and a_2, which determine the patch slope at each point, and thus determine the sharpness and contrast of the region.

Another type of image region that is very sensitive to blurring is the ideal edge patch. In the interest of convenience we just look at an ideal vertical edge:

$$g[x,y] = \begin{cases} b+c & x > x_o \\ \\ b & otherwise \end{cases} \tag{3.15}$$

where, without loss of generality, c is a positive constant. The corresponding singular values are:

$$\begin{aligned} s_1 &= \tfrac{c}{\sqrt{2}} N^{1/4} \\ s_2 &= 0 \end{aligned} \tag{3.16}$$

Only s_1 here reflects the value of parameter c, which gives the intensity difference (contrast) between the two sides of the edge.

In general, and regardless of type, rotating a patch by an arbitrary angle θ will not change the singular values of the gradient matrix. To see this, we note the relationship between the rotated gradient matrix $\mathbf{G}_\theta$ and the unrotated $\mathbf{G}$:

$$\mathbf{G}_\theta = \mathbf{G}\mathbf{R}_\theta^T \tag{3.17}$$

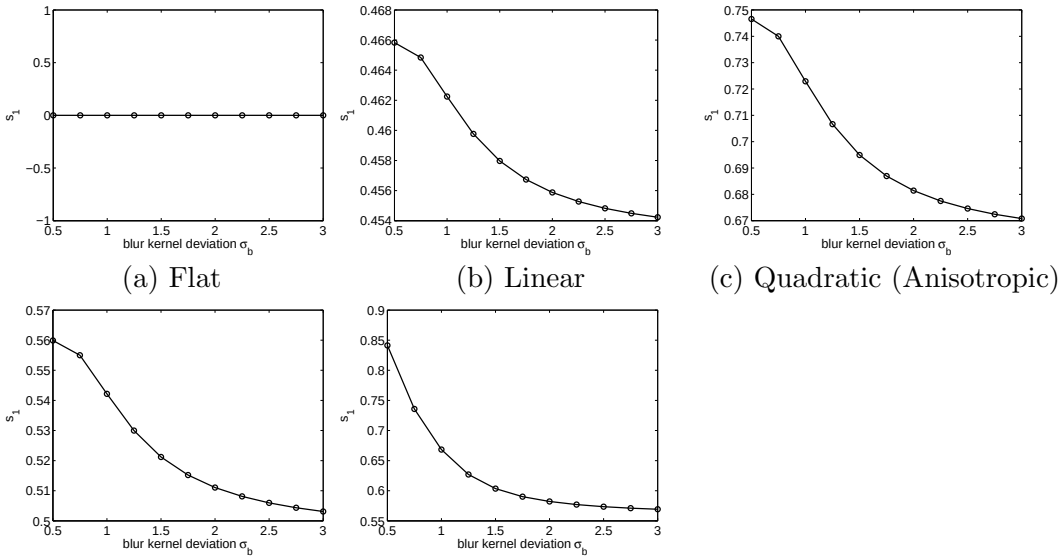
where $\mathbf{R}_\theta$ is the (orthonormal) rotation matrix:

$$\mathbf{R}_\theta = \begin{bmatrix} \cos\theta & \sin\theta \\ \\ -\sin\theta & \cos\theta \end{bmatrix} \tag{3.18}$$

Therefore, the SVD of $\mathbf{G}_\theta$ becomes:

$$\mathbf{G}_\theta = \mathbf{U}\mathbf{S}(\mathbf{R}_\theta\mathbf{V})^T, \tag{3.19}$$

which illustrates that the directions $\mathbf{v}_1$ and $\mathbf{v}_2$ are correspondingly rotated, but the singular values s_1 and s_2 remain unchanged.



(a) Flat (b) Linear (c) Quadratic (Anisotropic)

(d) Quadratic (Isotropic) (e) Edge

Figure 3.3: Plots of s_1 in blurring process for different patches. The blur kernel is Gaussian, and the value of its standard deviation σ_b is raised steadily to make each patch more and more blurred.

It is observed through the above analysis that the singular value s_1 is quite intimately related with the sharpness and contrast of the local region. This is valid not only in regions with strong direction and contrast (edged patch (3.16)), but also in regions which may be isotropic (quadratic patch (3.13), (3.14), where $a_1 = a_2$), or very smooth (linear patch (3.11)).

To verify the usefulness of s_1 in the presence of blur, we applied a Gaussian blur kernel (of size 5×5) with a steadily growing standard deviation σ_b to the above patch types (of size 11×11) and recorded the resulting s_1 values, which are shown in Fig. **3.3**. It is observed that, similar to S_{grad}, as the value of σ_b grows (more blurry) s_1 for all the non-flat patches drops steadily as expected.

3.1.2 Effect of Noise on Singular Values

Next, we address what happens to s_1 if the image (or patch) is corrupted by white (though not necessarily Gaussian) noise. Again, consider the white noise image patch (with pixel-wise variance σ^2) denoted in column-stacked vector format as an $N \times 1$ vector $\mathbf{n}$. Its gradient matrix (denoted as $\mathbf{G}_n$) can be produced by applying the derivative filters:

$$\mathbf{G}_n = \begin{bmatrix} \mathbf{D}_x\mathbf{n} & \mathbf{D}_y\mathbf{n} \end{bmatrix} \tag{3.20}$$

Because the noise is zero-mean, the expected value of $\mathbf{G}_n$ is:

$$E(\mathbf{G}_n) = \begin{bmatrix} \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{3.21}$$

and the expected gradient covariance matrix becomes:

$$\begin{aligned} E(\mathbf{C}_n) &= E(\mathbf{G}_n^T\mathbf{G}_n) \\ &= E\left(\begin{bmatrix} \mathbf{n}^T\mathbf{D}_x^T\mathbf{D}_x\mathbf{n} & \mathbf{n}^T\mathbf{D}_x^T\mathbf{D}_y\mathbf{n} \\ \mathbf{n}^T\mathbf{D}_y^T\mathbf{D}_x\mathbf{n} & \mathbf{n}^T\mathbf{D}_y^T\mathbf{D}_y\mathbf{n} \end{bmatrix}\right) \end{aligned} \tag{3.22}$$

where according to Equation (3.3) the first entry can be deduced as:

$$E(\mathbf{C}_n)_{1,1} = \sigma^2 tr(\mathbf{D}_x\mathbf{D}_x^T) \tag{3.23}$$

and similarly we have:

$$E(\mathbf{C}_n)_{1,2} = \sigma^2 tr(\mathbf{D}_y\mathbf{D}_x^T), \quad E(\mathbf{C}_n)_{2,1} = \sigma^2 tr(\mathbf{D}_x\mathbf{D}_y^T)$$

$$E(\mathbf{C}_n)_{2,2} = \sigma^2 tr(\mathbf{D}_y\mathbf{D}_y^T)$$

Recall that if we choose the filters (2.9) or (3.4) to make $\mathbf{D}_x$ and $\mathbf{D}_y$, the expected $\mathbf{C}_n$ will have the form:

$$E(\mathbf{C}_n) = \begin{bmatrix} \xi N^2\sigma^2 & 0 \\ 0 & \xi N^2\sigma^2 \end{bmatrix} \tag{3.24}$$

where $\xi = \frac{1}{2}$ if we use filters of (2.9), and $\xi = \frac{3}{16}$ for filters of (3.4).

Now consider how the value of s_1 changes when an ideal clean image $\mathbf{g}^*$ is corrupted by the white noise image denoted by $\mathbf{n}$. The gradient matrix of the noisy image $\mathbf{g}$ would become:

$$\mathbf{G} = \mathbf{G}^* + \mathbf{G}_n \tag{3.25}$$

Since $\mathbf{G}^*$ is deterministic, the expected $\mathbf{C}$ would have the form:

$$\begin{aligned} E(\mathbf{C}) &= E(\mathbf{G}^T\mathbf{G}) \\ &= \mathbf{G}^{*T}\mathbf{G}^* + E(\mathbf{G}_n^T\mathbf{G}_n) + 2\mathbf{G}^{*T}E(\mathbf{G}_n) \\ &= \mathbf{V}\begin{bmatrix} s_1^{*2} + \xi N^2\sigma^2 & 0 \\ 0 & s_2^{*2} + \xi N^2\sigma^2 \end{bmatrix}\mathbf{V}^T \end{aligned} \tag{3.26}$$

So on average the dominant singular value s_1 of the noisy image can approximately be written as:

$$s_1 \approx \sqrt{s_1^{*2} + \xi N^2\sigma^2} \tag{3.27}$$

This equation tells us that s_1 is determined by both s_1^* and σ^2. Given a fixed σ^2, the value of s_1 drops as s_1^* is decreased, or say when the image $\mathbf{g}^*$ becomes more blurry. Unfortunately, similar to S_{grad}, s_1 is also monotonically increasing with the noise variance σ^2.

3.1.3 Normalizing for the Noise Variance

To alleviate this problem, we define an intermediate metric H as follows:

$$H = \frac{s_1}{\sigma^2} \tag{3.28}$$

For now, we assume that the noise variance σ^2 is known, or at least can be estimated. For a fixed σ^2, the behavior of H is basically the same as s_1. If σ^2 is sufficiently large,

s_1 becomes approximately proportional to the standard deviation σ:

$$s_1 \approx \xi^{\frac{1}{2}} N\sigma \tag{3.29}$$

And therefore, the value of $H \approx \frac{\xi^{\frac{1}{2}} N}{\sigma}$ drops to zero with increasing σ, as desired. We note that in (3.28) the value of H estimated from patches in an image can be used to decide whether those patches contain real content (based on sharpness and contrast) as opposed to noise. Said another way, H can be thought of as a rough indicator of the signal to noise ratio.

Although the metric H has well-behaved characteristics in the presence of both blur and noise, it still suffers from the shortcoming that the variance of noise is assumed known or reliably estimated, which may not be the case in many real applications [88]. As a practical example, if the image of interest is the output of a spatially adaptive denoising filter [89], it is difficult to estimate how much noise still remains, because the denoising effect varies with the local content in different parts of the image. So we need a metric which implicitly contains an estimate of the local noise variance as well. This is what we set out to do next.

3.2 Metric Q

We define the image content metric Q as:

$$Q = s_1 \frac{s_1 - s_2}{s_1 + s_2} \tag{3.30}$$

Compared with H in (3.28), it can be seen that in the definition of Q the factor $\frac{1}{\sigma^2}$ is replaced by another quantity, which we call the *coherence*:

$$R = \frac{s_1 - s_2}{s_1 + s_2} \tag{3.31}$$

As we briefly describe below, this replacement will not only do away with the explicit need to know the noise variance as a priori, but also enables us to measure content even if the noise variance is non stationary across the image.

3.2.1 Behavior of Coherence with Noise

Recall that for a noisy image patch $\mathbf{g}$, its singular values s_1 and s_2 can approximately be written as:

$$s_1 \approx \sqrt{s_1^{*2} + \xi N^2 \sigma^2}$$
$$s_2 \approx \sqrt{s_2^{*2} + \xi N^2 \sigma^2}$$

where s_1^* and s_2^* stand for singular values of a patch of the noise-free image $\mathbf{g}^*$. So the coherence can be written as:

$$\begin{aligned} R &\approx \frac{\sqrt{s_1^{*2} + \xi N^2 \sigma^2} - \sqrt{s_2^{*2} + \xi N^2 \sigma^2}}{\sqrt{s_1^{*2} + \xi N^2 \sigma^2} + \sqrt{s_2^{*2} + \xi N^2 \sigma^2}} \\ &= \frac{s_1^{*2} + \xi N^2 \sigma^2 - s_2^{*2} - \xi N^2 \sigma^2}{(\sqrt{s_1^{*2} + \xi N^2 \sigma^2} + \sqrt{s_2^{*2} + \xi N^2 \sigma^2})^2} \\ &= \frac{s_1^{*2} - s_2^{*2}}{s_1^{*2} + s_2^{*2} + 2\xi N^2 \sigma^2 + 2\sqrt{(s_1^{*2} + \xi N^2 \sigma^2)(s_2^{*2} + \xi N^2 \sigma^2)}} \end{aligned} \tag{3.32}$$

The above illustrates that in a noisy image patch, the computed value of coherence R is roughly inversely proportional to the local noise variance σ^2 when $s_1^* > s_2^*$ (which is true whenever the underlying patch is anisotropic.)

This is not the first time that coherence R has been used to analyze local image characteristics. Indeed, Bigun et al. [87] used this quantity to measure the locally dominant orientation of textures in a deterministic setting. In [86], we used this measure in a statistical framework to estimate dominant orientations in a multi-scale setting. As mentioned before, s_1 and s_2 represent the energy in both the dominant

direction and its perpendicular direction. So basically R measures their relative size. Considering a noise-free region with strong anisotropic geometric structure (such as the edged patch), the difference between s_1^* and s_2^* is very large, and in the absence of noise, the value of R is near 1. If white noise is added, the resulting R would be reduced, indicating that this region has become less structured, or the strength of the dominant direction has been reduced. It is worth repeating that metric Q is valid as an approximation of H only when the noise-free patch contains a dominant orientation (where $s_1^* > s_2^*$). The behavior of Q when the patch is isotropic (where $s_1^* = s_2^*$) will be discussed in the next subsection.

So far, the definition and descriptions of measures Q and R were quite general in the sense that the only assumption on noise was that it is white. It is instructive to study the statistical behavior of Q and R in a specific (Gaussian) noise setting. As may be expected, the statistical distribution of these metrics in the presence of noise of arbitrary distribution is in general very complex. However, their distributions are tractable when the noise is restricted to be white and Gaussian. For the sake of completeness therefore, we next discuss the statistical behavior of R when the image patch is purely white Gaussian noise (WGN).[2]

If the image patch $\mathbf{g}$ is pure WGN ($s_1^* \approx s_2^* \approx 0$), according to (3.32) coherence $R \approx 0$, but in practice in a finite window size, R has a small positive value. Put another way, while in theory, a white noise images contains (by definition) no dominant orientation, patches with limited size lead to a small relative difference between s_1 and s_2, leading to a nonzero coherence. More specifically, given a patch of white Gaussian

[2]For the sake of completeness, the statistical distribution of Q is also given in the Appendix.

noise pixels, the corresponding R is a random variable whose density function is (See Fig. **3.5** and Appendix for detailed derivation):

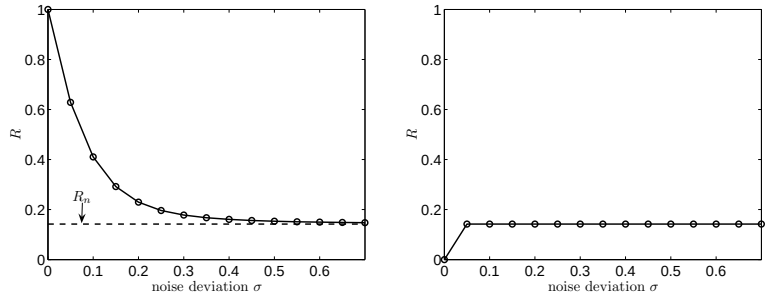
$$f_R(r) = 4(N-1)r\frac{(1-r^2)^{N-2}}{(1+r^2)^N}, \tag{3.33}$$

It is interesting to note that the pdf of R is independent of the mean and variance of the white noise; instead, it is a strong function of the patch size N.

We computed the coherence R on an edged patch (as described in (3.15), where $b = 0.3$ and $c = 0.5$) and a flat patch (as described in (3.9), where $c = 0.5$) to illustrate its behavior in random noise. A sample of WGN $\mathbf{n}$ is added to each patch with its standard deviation σ ranging from 0 to 0.7. The plots of the coherence R of each noisy edged patch are shown in Fig. **3.4** (a). Since $s_1^* > s_2^* = 0$ in the clean edged patch, $R = 1$ when $\sigma = 0$. We can see that as σ increases, the value of R drops as desired, and approaches the value of R_n, which is the coherence of the noise-only sample $\mathbf{n}$. This is reasonable, because when $\sigma \to \infty$, the patch looks more and more like pure WGN. In the flat case (see Fig. **3.4** (b)), where no anisotropic structure exists, $R = R_n$ for all the nonzero noise standard deviations[3]. The random variable R_n is distributed according to the pdf (3.33).

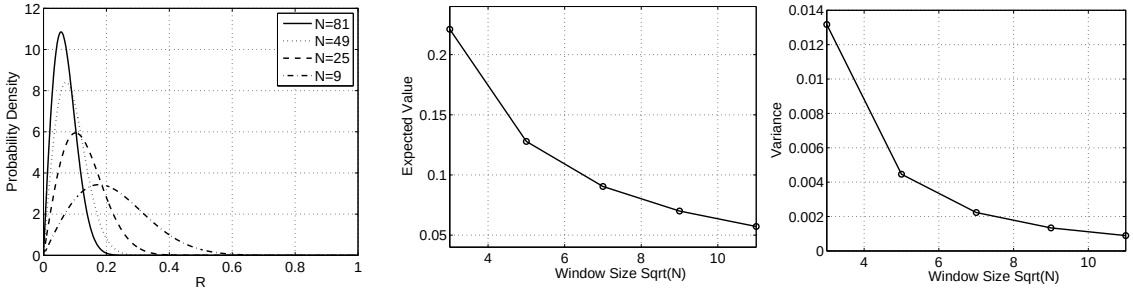
The pdf of R for a variety of N is plotted in Fig. **3.5**, where the change of the first and second moments of $f_R(r)$ versus $\sqrt{N}$ are also shown. We can see that the expected value of R decreases as N increases. This coincides with the fact that as the patch size grows, asymptotically there is no dominant direction in a noise patch. The reader may be wondering why we bother to derive and illustrate the statistical distribution of R as we have done here. This indeed turns out to be helpful in the latter

[3]For the flat patch, $s_1 = s_2 = 0$ when $\sigma = 0$. In such situations, we define $R = 0$.



(a) Edged patch (b) Flat patch

Figure 3.4: Plots of the coherence R of a noisy edged patch (a) and a noisy flat patch (b) versus the noise standard deviation σ. R_n stands for the coherence of the noise sample.



(a) Probability density functions (b) Expected values (c) Variances

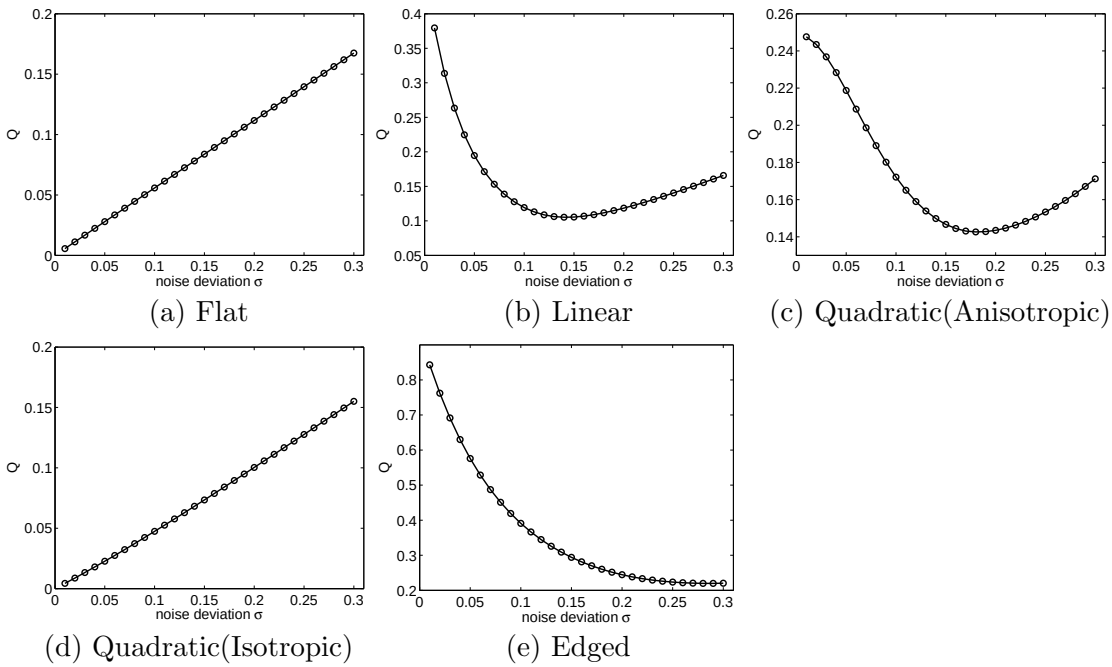
Figure 3.5: The statistical properties of local coherence measure R for white Gaussian noise: (a) Probability density functions with different patch sizes. (b) The expected values.(c) The variances.

part of this section, where the density function of R will be used in a significance test to tease apart isotropic patches from anisotropic ones. This distinction will then be employed in reliably calculating the metric Q for the whole image.

3.2.2 Metric Q in Patches

To further understand the performance of the image content metric Q in different types of patches in the presence of Gaussian noise, we employ Monte-Carlo simulations. White Gaussian noise with a variety of σ, ranging[4] from 0.01 to 0.3, was added

[4]The pixel value is on a scale of $[0, 1]$ gray level.



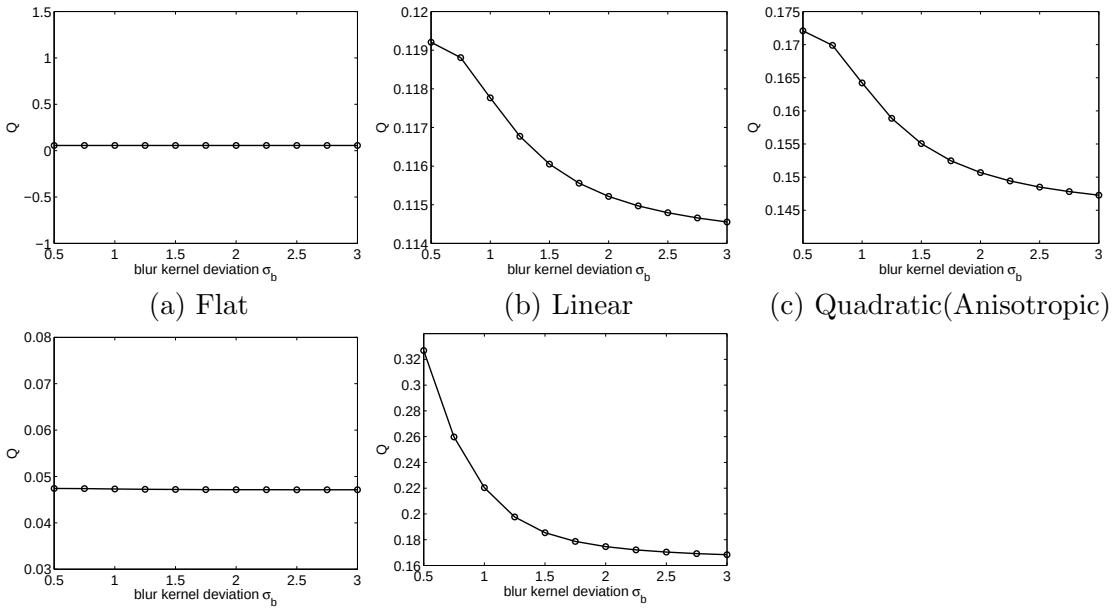
(a) Flat (b) Linear (c) Quadratic(Anisotropic)

(d) Quadratic(Isotropic) (e) Edged

Figure 3.6: Plots of the mean metric Q versus the noise standard deviation σ in Monte-Carlo simulations for different patches. 100 different noise realizations were used for each σ to get the averaged Q.

to the test patches shown earlier. For each σ, 100 Monte-Carlo simulations are carried out with independent noise realizations. Fig. **3.6** shows the plots of the averaged Q across these experiments, versus the standard deviation of noise. In this experiment, we distinguish quadratic patches into isotropic ($a_1 = a_2$) and anisotropic ($a_1 \neq a_2$) types. It can be observed that the behavior of Q is consistent across all anisotropic patch types (including linear, anisotropic quadratic and edge), but different in the isotropic patches (including flat, and isotropic quadratic.)

It is no surprise to see that in isotropic cases, Q goes up steadily when σ rises, because the coherence part in Q cannot play the role of $\frac{1}{\sigma^2}$ as we mentioned before. Take the flat patch for example. The coherence R does not change with the noise variance (see Fig. **3.4**(b)), while $\hat{s}_1$ is proportional to σ (see equation (3.29)). So in this case, Q



(a) Flat (b) Linear (c) Quadratic(Anisotropic)

(d) Quadratic(Isotropic) (e) Edged

Figure 3.7: Monte-Carlo simulations using both random noise and blur for different patches. Each patch was blurred first and then white Gaussian noise with $\sigma = 0.1$ was added. After each blurring process, 100 independent noise simulations were applied respectively, and averaged value of metric Q was calculated. The size of the Gaussian smoothing kernel is 5×5.

also becomes proportional to σ.

For anisotropic cases, Q behaves reasonably, since the coherence part decreases as described in (3.32) with respect to the increase of the noise standard deviation σ. But when σ goes to infinity, the image patch looks more and more like pure noise. So the coherence part approaches the expected value of the random variable R characterized by the density function (3.33). This value is usually small, but not zero. While, on the other hand, the increase of the s_1 part in Q is approximately proportional to σ, hence the value of Q rises again. Fortunately, this becomes a problem only when the noise standard deviation becomes extremely large.

Next, we take the blurring process into account. Blurred patches are obtained by applying a Gaussian smoothing filter with a growing standard deviation σ_b. After

that, white Gaussian noise with variance σ^2 is added respectively. In Fig.**3.7** we can see that basically for all the anisotropic cases, the value of metric Q drops when the test region gets more and more blurred in the presence of noise as expected. While in isotropic cases, Q does not show significant change.

So what do we learn from the above simulations? Namely, we can see that generally for anisotropic patches, where a dominant orientation exists, the proposed metric Q is able to detect both blur and random noise. So in practice, when measuring the true content of an image as a whole using Q, the anisotropic areas are detected and used to compute a global measure for the whole image. Put another way, isotropic patches should be avoided in the calculation of the overall image content metric Q. Specifically, one way to distinguish isotropic from anisotropic areas is by employing significance testing based on local coherence R, whose statistics in the "noise-only" case were described earlier in Section 3.2.1.

Defining the null hypothesis $\mathcal{H}_0$ as: "The given patch is isotropic with white Gaussian noise," we can calculate the metric R and use it as a test statistics to decide whether to reject the null hypothesis $\mathcal{H}_0$. Numerically, the test is carried out by comparing the calculated R for the patch to a preselected threshold τ. If $R \geq \tau$, then the hypothesis is rejected. For its part, τ is determined by a significance level $0 < \delta < 1$, which is the probability of rejecting the null hypothesis when this hypothesis is in fact true. The relationship between δ and τ, which comes directly from integrating the pdf of R, is[5]:

[5]Strictly speaking, the density function used in the above formula should be a valid pdf for R in all types of isotropic patches with WGN added; but this is not practical. Instead, for simplicity, we employ the pdf for pure WGN (see equation (3.33)) as an approximate and quite reasonable alternative.

$$\delta = \text{Prob}(R \geq \tau | \mathcal{H}_0) = \int_{\tau}^{1} f_R(r) dr$$

The integral of $f_R(r)$ above can be computed in closed form as:

$$\delta = \int_{\tau}^{1} f_R(r) dr = \left[-\left(\frac{1-r^2}{1+r^2} \right)^{N-1} \right]_{\tau}^{1} = \left(\frac{1-\tau^2}{1+\tau^2} \right)^{N-1} \tag{3.34}$$

For a desired δ, to determine the threshold, we can simply solve the above for τ. For example, if the patch size $N = 64$, and we want the significance level δ to be 0.001, then the corresponding threshold becomes $\tau = 0.2340$. If the local coherence $R \geq \tau$, the test patch would be labeled as "anisotropic".

3.2.3 Calculating Q for a Whole Image

To summarize the above discussions, we provide here a concise description of the algorithmic procedure for calculating the metric Q from a given image:

Algorithm 2 Algorithmic Procedure for Computing Q

1. Given a noisy image, divide it into M non-overlapping patches of size N, and calculate the local coherence R_j using equation (3.31) for each patch $j = 1, \cdots, M$.

2. Find (say $m \leq M$) anisotropic patches by thresholding the local coherence values as $R_j \geq \tau$. The threshold τ is determined by solving the equation (3.34) with a given significance level δ.

3. Calculate the local metric Q_j using equation (3.30) on each anisotropic patch identified in step 2.

4. Output the value $Q = \frac{1}{M} \sum_{j=1}^{m} Q_j$ as the metric for the whole image.

A simulated experiment using the natural image *Man* has been carried out to illustrate the above procedure. The data in the test image set are generated through

the following model:

$$\mathbf{g} = \mathbf{H}\mathbf{f} + \mathbf{n} \tag{3.35}$$

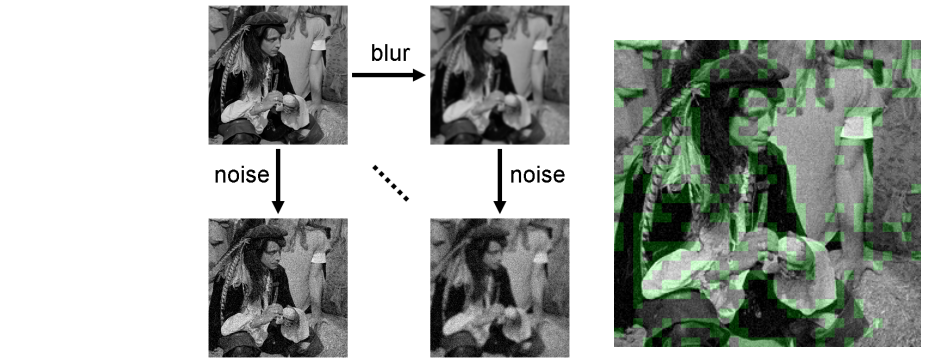
where $\mathbf{f}$ stands for the clean and sharp image, $\mathbf{H}$ represents the blurring matrix generated by a Gaussian blurring kernel, and $\mathbf{n}$ is the WGN.

We arranged the simulated images into a grid as shown in Fig. **3.8** (a), where images get more and more blurred from left to right (by applying a 9×9 Gaussian blur kernel with growing standard deviation σ_b), and more and more noisy from top to bottom. The corresponding metric Qs (with the patch size $N = 8$ and $\delta = 0.001$) are given in Fig. **3.8** (d). Full-reference metric mean-squared error (MSE) and a no reference sharpness metric JNB [4] are also tested for comparison. It can be observed that, like MSE, the change of the metric Q successfully reflects the change of the image quality, while the sharpness metric JNB (whose value drops as image becomes blurry,) failed in distinguishing quality decay against noise.

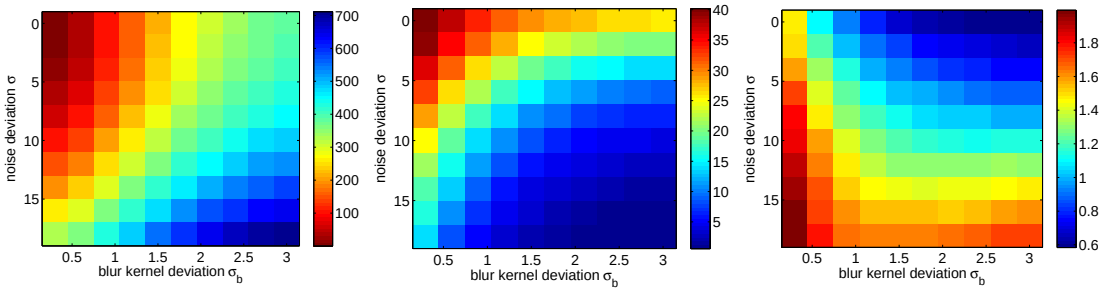
3.3 Extending Q to Isotropic Regions

So far we have developed a no-reference image content metric Q based on the singular value decomposition (SVD) of local image gradients. Simulated experiments show that this metric is capable of measuring true image content for local anisotropic patches (e.g. edge patch in Fig. **3.9** (a)). Its value drops monotonically when such patches becomes either blurry or noisy. We have also shown that Q is capable of globally monitoring the change of image quality by concentrating on the anisotropic regions.

However, one problem for metric Q is that it fails in structured regions that do



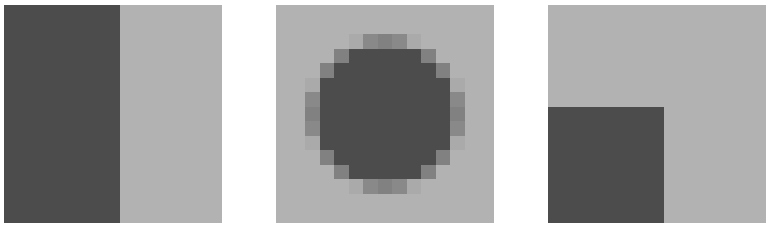
(a) Test image set (b) Anisotropic patches



(c) MSE (d) Metric Q (e) JNB [4]

Figure 3.8: Evaluations of Q, MSE and JNB [4] on the image **Man** (256×256), and its successively degraded versions. In (b), the green areas illustrate the anisotropic patch set used in measuring Q. In (c), we inverted the color scale just to show the similarity between MSE and Q in capturing the trend of image quality change.

not contain a single dominant direction. One example is the *isotropic* circular patch in Fig. **3.9** (b), which has a circular shape and the energy in every orientation is basically the same. Actually any axially symmetric structure in a region will yield singular values $s_1 = s_2$ (see Fig. **3.9** (c)). If a metric cannot assess the quality change of such strongly structured patches, it may fail in monitoring the whole image. To solve this problem, we extend current metric Q in this section. The updated metric is called Q_{pro}, which implements SVD in *transformed coordinates* that are adapted to local image content. We will see that the metric Q in Section 3.2 is a special case of Q_{pro}, and Q_{pro} is valid for most structured regions, including isotropic ones.



(a) Edge (b) Circular (c) Corner

Figure 3.9: Patches with strong structures, including (a) edge, (b) circular and (c) corner. (a) is a typical anisotropic patch. (b) is a typical isotropic patch. Though (c) looks like anisotropic, its $s_1 = s_2$ and thus it is isotropic.

3.3.1 SVD in Transformed Coordinates

Consider a typical isotropic patch example in Fig. **3.10**, where in traditional Cartesian coordinate system we have $s_1 = s_2$. Our goal is to find a way to collect the gradient energy along the circular edge direction and separate it from the one in its perpendicular direction. One way is through the application of a rotation transform. For example, consider the j-th point on the circle and its pixel-wise gradient vector $\nabla \mathbf{g}[\mathbf{x}_j] = [g_x[\mathbf{x}_j], g_y[\mathbf{x}_j]]^T$, (where $\mathbf{x}_j = [x_j, y_j]$,). To distinguish the gradient component in the normal direction to the edge from the one in the tangent direction to the edge, we can rotate the local Cartesian coordinates (with origin at $\mathbf{x}_j$) by an angle θ_j (see Fig. **3.10**) and calculate the rotated vector $\nabla \mathbf{g}'[\mathbf{x}_j]$ using the rotation matrix $\mathbf{R}_{\theta_j}$ defined in equation (3.18). Specifically, we have:

$$\nabla \mathbf{g}'[\mathbf{x}_j] = \mathbf{R}_{\theta_j} \nabla \mathbf{g}[\mathbf{x}_j] \tag{3.36}$$

where $\nabla \mathbf{g}'[\mathbf{x}_j] = [g_s[\mathbf{x}_j], g_t[\mathbf{x}_j]]^T$. $g_s[\mathbf{x}_j]$ represents the gradient component in the normal direction to the edge of the circle at $\mathbf{x}_j$, and $g_t[\mathbf{x}_j]$ denotes the component in the tangent direction.

Note that the angle θ_j is determined by both the pixel position $\mathbf{x}_j$ and the

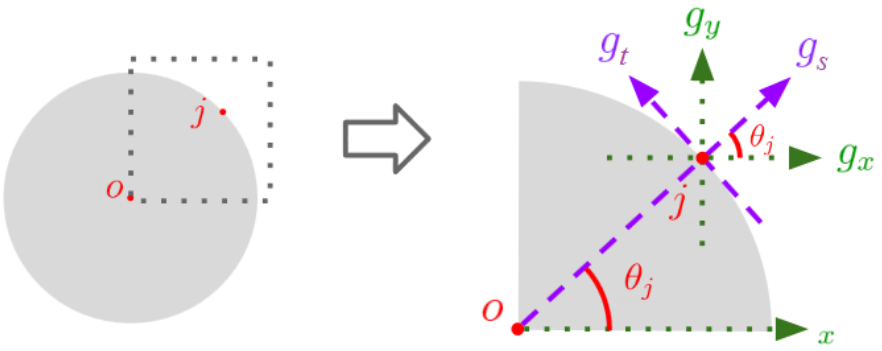


Figure 3.10: Rotational transformation example.

center position (denoted as $\mathbf{x}_o = [x_o, y_o]$). We implement this transform on the patch (b) from Fig. **3.9**, and its corresponding gradients are given in Fig. **3.11**, where we can see that after applying the transform, most gradient energy is transferred to g_s, which means the rotational transform of local coordinates produces a dominant direction in the gradient space.

Because the rotation matrix is orthogonal, it does not change the correlation structure of random noise that is assumed to be zero-mean IID. So all the statistical analysis on Q in Section 3.2 is still valid. We define the new metric Q_{pro} as:

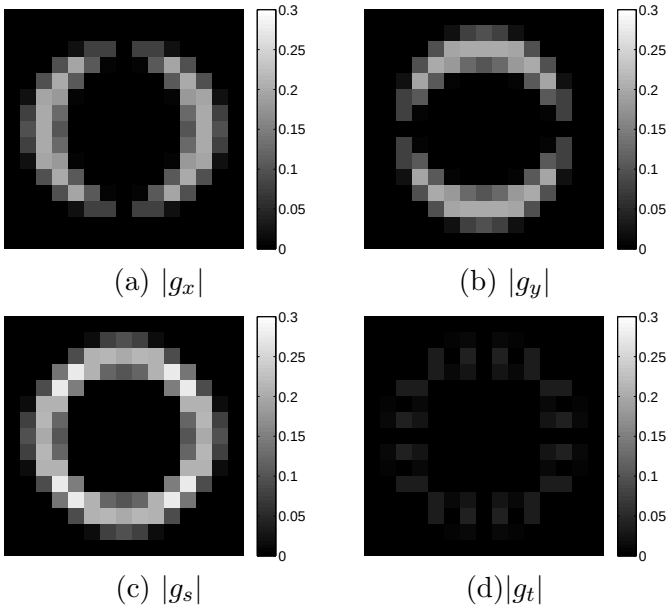
$$Q_{pro} = s'_1 \frac{s'_1 - s'_2}{s'_1 + s'_2} \tag{3.37}$$

where s'_1 and s'_2 are the singular values of the transformed gradient matrix $\mathbf{G}'$ in local window w:

$$\mathbf{G}' = \begin{bmatrix} \vdots & \vdots \\ g_s[\mathbf{x}_j] & g_t[\mathbf{x}_j] \\ \vdots & \vdots \end{bmatrix}, \quad j \in w \tag{3.38}$$

Because we have

$$s_1^2 + s_2^2 = tr(\mathbf{G}^T\mathbf{G}) = \sum_{j \in w} \nabla \mathbf{g}^T[\mathbf{x}_j] \nabla \mathbf{g}[\mathbf{x}_j], \tag{3.39}$$



(a) $|g_x|$ (b) $|g_y|$

(c) $|g_s|$ (d)$|g_t|$

Figure 3.11: (a)-(b): Magnitude of the horizontal and vertical derivatives of Fig. **3.9** (b) before the rotational transform. (c)-(d): Magnitude of the derivatives in the normal direction to the edge and its perpendicular direction.

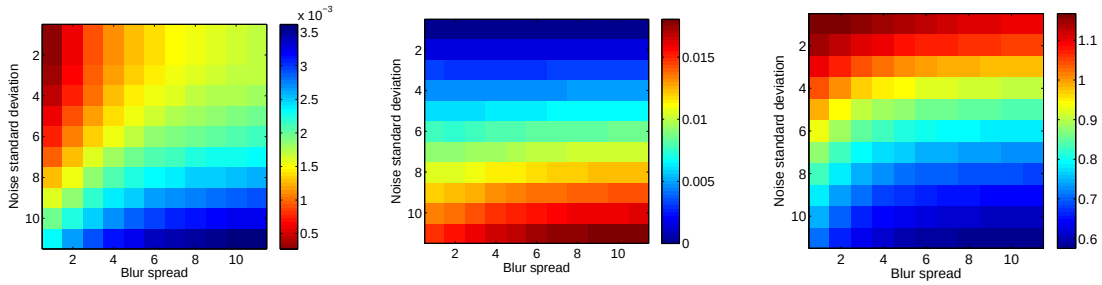
and also because $\nabla \mathbf{g}'[\mathbf{x}_j] = \mathbf{R}_{\theta_j} \nabla \mathbf{g}[\mathbf{x}_j]$, hence the total gradient energy:

$$\begin{aligned} s'^2_1 + s'^2_2 &= \sum_{j \in w} \nabla \mathbf{g}'^T[\mathbf{x}_j] \nabla \mathbf{g}'[\mathbf{x}_j] \\ &= \sum_{j \in w} \nabla \mathbf{g}^T[\mathbf{x}_j] \mathbf{R}^T_{\theta_j} \mathbf{R}_{\theta_j} \nabla \mathbf{g}[\mathbf{x}_j] \\ &= \sum_{j \in w} \nabla \mathbf{g}^T[\mathbf{x}_j] \nabla \mathbf{g}[\mathbf{x}_j] \\ &= s_1^2 + s_2^2, \end{aligned} \tag{3.40}$$

which means the rotational transform of the local coordinates does not change the total energy of gradient field, but it produces a dominant direction, which helps Q_{pro} to capture the main structure of the isotropic patch.

3.3.2 Behavior of Q_{pro} in An Isotropic Patch

To verify the validity of Q_{pro} with respect to both blur and noise, a simulated experiment similar to the one in Section 3.2.3 is carried out. The test isotropic patch in Fig. **3.9** (b) is distorted through the same model (3.35). Both noise variance and blur kernel spread could be varied to alter the distortion level. The corresponding values of metric Q and Q_{pro} are given in Fig. **3.12** (b) and (c). Full-reference metric MSE is also plotted in (a) for comparison. In the plots given in Fig. **3.12** the test patch gets more and more blurred from left to right, and more and more noisy from top to bottom. It can be observed that, like MSE, the change of the metric Q_{pro} successfully reflects the change of the image quality: Q_{pro} monotonically drops as the distortion level rises, while the old Q fails.



(a) MSE (b) Q (c) Q_{pro}

Figure 3.12: Evaluations of MSE, Q and Q_{pro} on patch Fig. **3.2** (b), and its successively degraded versions. In (a) we inverted the color scale just to show the similarity between MSE and Q_{pro} in capturing the trend of image quality change.

3.3.3 Q_{pro} Calculation

One problem for Q_{pro} is that for each patch it is sensitive to the center position $\mathbf{x}_o$, while the isotropic content is not always centered at the middle of the patch. In other words, before calculating Q_{pro} we need to detect the proper center position first. On the

other hand, in practice anisotropic patches appear more frequently than isotropic ones, which means given a patch we may need to decide which metric should be employed.

In Section 3.2.2 we have discovered that the validity of Q depends on the relative difference between s_1 and s_2, which can be measured through its coherence R. Similarly, given a pair of singular values from transformed coordinates, we have the corresponding coherence:

$$R' = \frac{s'_1 - s'_2}{s'_1 + s'_2} \tag{3.41}$$

Higher coherence value indicates more structure captured by s'_1, and thus the corresponding Q_{pro} should perform better. In other words, we can also employ the coherence R' to determine the best center position. Given a patch, since its coherence is determined by the center position $\mathbf{x}_o$, we can denote R' as $R'(\mathbf{x}_o)$.

Assume that we have a set of candidate positions $\{\mathbf{x}_j\}$, the one with the highest $R'(\mathbf{x}_j)$ could be selected as the proper center position. Note that this position can be either inside or outside the patch. Specifically, if the center position is infinitely far from the patch ($\mathbf{x}_\infty = [-\infty, 0]$), the corresponding rotation matrix would become identity, and its $Q_{pro} = Q$. In other words, Q is nothing but a special version of Q_{pro} with its center point located at $\mathbf{x}_\infty$.

In practice, given an image patch inside window w, we calculate its Q_{pro} through the following scheme:

1) Collect the candidate center position set $\{\mathbf{x}_j\}$ including all the pixel positions inside w and the infinity position $\mathbf{x}_\infty$. Calculate the corresponding coherence set $\{R'(\mathbf{x}_j)\}$.

2) Select the position with the highest coherence value as the proper center position $\mathbf{x}_o = \arg\max_{\mathbf{x}_j} R'(\mathbf{x}_j)$.

3) Calculate the Q_{pro} based on the position $\mathbf{x}_o$.

Note that by using the above scheme, if the measured patch is a typical anisotropic one (e.g. the edge patch in Fig. **3.9**), then $\mathbf{x}_\infty$ will be selected as the proper center position (since $\{R'(\mathbf{x}_\infty)\}$ has the highest value), and thus Q_{pro} will become Q. In other words, Q is automatically included by Q_{pro}, and there is no need to decide which metric (Q or Q_{pro}) should be used.

Though Q_{pro} is valid for both isotropic and anisotropic patches, given an arbitrary patch we still need to decide whether it contains sufficiently strong structure or not. For example, given a flat patch neither Q or Q_{pro} is valid, since no mater how we rotate the coordinates we always have $s'_1 = s'_2 = 0$. So, given a whole image, before calculating its Q_{pro} we still need to examine each local patch to see if it is valid for Q_{pro} measurement. The validity could again be determined by a significance test based on $R'(\mathbf{x}_o)$. In other words, except for an extra step of selecting $\mathbf{x}_o$ for all the non-overlapping patches, the way of calculating Q_{pro} for a given whole image is basically the same as the way used for calculating metric Q (see Algorithm 3).

3.4 Correlation with HVS

In the previous section we have shown that as a given image gets more blurry or noisy, its Q and Q_{pro} monotonically drop. However, we have not yet provided any evidence to illustrate whether they behave close to HVS. Specifically, we want to see if their sensitivity to blur and noise is consistent with HVS. In this section, we provide some tests based on TID2008 database [90] to illustrate the correlation between subjective ratings and the two proposed metrics. TID2008 database contains 25 reference images

Algorithm 3 Algorithmic Procedure for Computing Q_{pro}

1. Given a noisy image, divide it into M non-overlapping patches of size N.
2. For each patch $j = 1, \cdots, M$, select the proper center position $\mathbf{x}_o$ and record the corresponding coherence $R'(\mathbf{x}_o)$.
3. Find (say $m \leq M$) valid structured patches by thresholding the local coherence values as $R'(\mathbf{x}_o) \geq \tau$. The threshold τ is determined by solving the equation (3.34) with a given significance level δ.
4. Calculate the local metric $Q_{pro}[\mathbf{x}_j]$ on each structured patch identified in step 3.
5. Output the value $Q_{pro} = \frac{1}{M} \sum_{j=1}^{m} Q_{pro}[\mathbf{x}_j]$ as the metric for the whole image.

(some are illustrated in Fig. **3.13** (a)-(c)) and each reference image $\mathbf{f}_i$ $(i = 1, 2, ..., 25)$ is degraded by several types of distortions (Gaussian blur, Gaussian noise, quantization noise, JPEG compression, etc.), where each type has 4 distortion levels. For each distorted version $\mathbf{g}_{ik}$ $(k = 1, 2, ...)$, the corresponding mean opinion score (MOS) is also recorded and included in the database. In this section, we consider both Gaussian noise and Gaussian blur distortions as these are related to our application of interest. So for a given reference $\mathbf{f}_i$, we consider a set of corresponding degraded images $\varphi_i = \{\mathbf{g}_{ik} | k = 1, 2, ..., 8\}$ including 4 noisy and 4 blurry images.

All 25 image sets from the database were tested. When calculating both Q and Q_{pro} we fix the patch size $N = 8$, and the significance level $\delta = 0.001$ (or the threshold $\tau = 0.2340$). The no-reference sharpness metric JNB [4] is also tested for comparison. For each set φ, Spearman rank-order correlation coefficient (SROCC) [91] is employed to measure the correlation between MOS and the target metrics, including Q, Q_{pro}, MSE and JNB. The average SROCC scores and the standard deviations are given in Table

Table 3.1: Mean and standard deviation of the SROCC values of full-reference (FR) and no-reference (NR) metrics using 25 image sets from TID2008 database.

SROCC	blurry images		noisy images		blur + noise	
	mean	std.	mean	std.	mean	std.
MSE (FR)	-1.0	0.0	-0.992	0.039	**-0.930**	**0.091**
Metric Q (NR)	1.0	0.0	0.976	0.086	**0.921**	**0.080**
Metric Q_{pro} (NR)	1.0	0.0	0.992	0.040	**0.937**	**0.083**
JNB (NR)	1.0	0.0	-0.840	0.339	**0.452**	**0.171**

3.1 (where test results using blurry image sets and noisy image sets separately are also provided). It can be observed that the performance of the proposed metric Q and Q_{pro} is as good as MSE: the magnitude of the corresponding SROCC for Q, Q_{pro} and MSE are above 0.9 with low variances. Q_{pro} is even better than MSE with higher magnitude of mean SROCC and lower variance. Meanwhile, Q and Q_{pro} outperform JNB [4] since JNB fails in distinguishing quality decay against high frequency components due to noise. As specific illustrations, plots of MOS versus the tested metrics corresponding to the sample images in Fig. **3.13** (a)-(c) are illustrated in Fig. **3.13** (d)-(l), which show that both metric Q and Q_{pro} respond to blur and noise in a way close to the subjective quality perception. For some images (such as Fig. **3.13** (c)), SROCC of Q_{pro} (see (l)) is slightly higher than the one of Q (see (i)). This is probably because Q_{pro} captures more structured information than Q.

3.5 Application: Optimizing Denoising Filters

In any leading denoising algorithm, there always exist some tuning parameters. Such as smoothing parameter of Bilateral filter, noise level estimate of BM3D, analysis window size of NLM, etc. How to automatically adjust the tuning parameters with

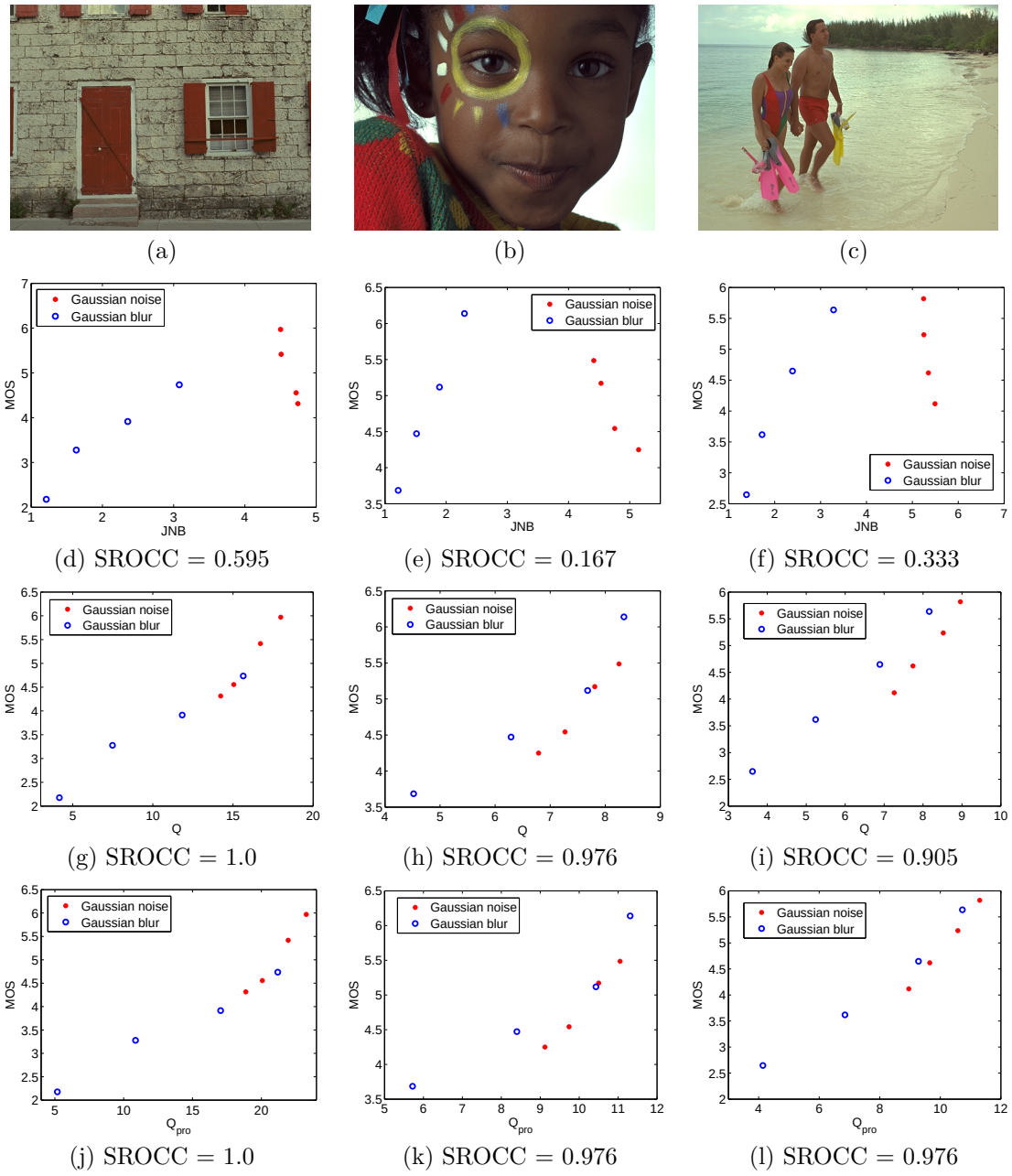


Figure 3.13: Sample reference images (a)-(c) from TID2008 database. (d)-(f) show plots of MOS versus JNB [4] of the image sets degraded from (a)-(c) respectively, where each set contains images distorted (using Gaussian blur and Gaussian noise) from a single reference. (g)-(i) show plots of MOS versus metric Q. (j)-(l) show plots of MOS versus metric Q_{pro}.

respect to the visual quality of the output image is a significant problem.

3.5.1 Existing Parameter Selection Methods

First, let us briefly summarize the relevant existing literature in this area. A commonly used optimization method is through full-reference or reduced-reference metrics [92]. Full-reference metrics need a complete reference image, and what they calculate is basically the similarity between the target and reference images. Such measures of similarity include the classical MSE and and the recently introduced Structural Similarity (SSIM) [92]. Reduced-reference metrics require the reference image to be partially available, which is usually in the form of a set of extracted features [92]. However, in most practical applications the reference image is unavailable. Therefore, in applications the (full-reference) quality metrics MSE or SSIM can not be directly used to optimize the parameters of algorithms.

Several (no-reference) approaches have been developed to address the parameter optimization problem. Generalized cross-validation (GCV) [93], [94] and the L-curve method [95], [96] have been widely used in choosing the regularization parameters for various restoration applications. More recently, methods based on Stein's unbiased risk estimate (SURE) were proposed for the denoising problem [97], [98], which provide a means for unbiased estimation of the MSE without requiring the reference image. Useful as they are, these methods are far from ideal. Namely, aside from their computational complexity, they address the parameter optimization problem without direct regard for the visual content of the reconstructed images. Instead, they compute or approximate quantities such as MSE (or the related cross-validation cost), which are not necessarily very good indicators of visual quality of the results. As a particular example, for instance, Ramani *et al.*'s Monte-Carlo SURE [98], which can be used for arbitrary de-

noising algorithms, is based on the idea of probing the denoising operator with additive noise and manipulating the response signal to estimate MSE. This approach is also only appropriate when the noise is assumed to be Gaussian, and generally requires an accurate estimation of the noise variance as well.

In image restoration, as is the case for any estimation problem generally, it can be observed that selecting parameters amounts to a tradeoff between bias and variance in the final estimate. A canonical example is the regularization parameter in MAP-based restoration algorithms [93], [96]. Generally, the larger the parameter is, the more smooth the image content becomes (small variance), while more useful detail and edges are flattened or blurred (larger bias). In other words, an ideal no-reference measure that is useful for the parameter optimization problem should take both blur and noise on the reconstructed image into account [99]. However, most sharpness metrics [47], [4], [44], [60] can hardly distinguish image quality decay against high frequency behavior due to noise. Take the approach in [4] for example, whose value drops when the image is increasingly more blurred. The value of this measure also rises if the variance of noise is increased (see Fig. **3.8** (e)). For the metrics based on edge detection and edge width estimation [44], the performance stability can easily suffer in the presence of noise. Such problems are precisely what our proposed metric Q and its extension Q_{pro} are intended to address.

On a related note, we mention that some no-reference image quality metrics have been developed to detect blur and noise simultaneously. One example is the metric based on the image anisotropy [100] proposed by Gabarda and Cristóbal [46]. They calculate the Rényi entropy [101] pixel by pixel along different directions, and use the

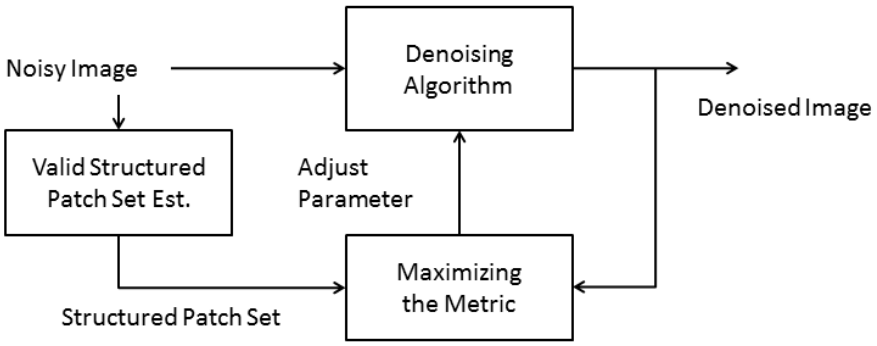


Figure 3.14: Selecting the tuning parameter using metric Q_{pro} or Q.

variance of the entropy to index visual quality. However, such metrics require uniform degradation across the whole image, and do not work well if the random noise or blur varies spatially, which is the case, for instance, in images denoised by spatially adaptive filters.

3.5.2 Automatic Parameter Tuning System Based on Proposed Metrics

In this section, we will provide evidence that the proposed metric Q_{pro} (also Q) can be used to optimize the parameters of denoising algorithms. In particular, like [98], the application of Q_{pro} to any "black-box" denoising algorithm with parameters in need of tuning is possible.

In order to obtain a practical procedure for parameter setting, the strategy we take for computing the proposed metric Q_{pro} is slightly different from what we described in Section 3.3.3. Namely, we use the original noisy *input* image to estimate a valid structure patch set, and use this reference set to compute the metric Q_{pro} on the *output* of the respective denoising filter. We vary the value of the tuning parameter

while observing the resulting Q_{pro}. The "best" value of the algorithm parameter is then selected as that which maximizes the metric Q_{pro} on the output (see Fig. **3.14**). In all the following experiments for parameter optimization, we fix the patch size $N = 8$, and the significance level $\delta = 0.001$ (or the threshold $\tau = 0.2340$). From a practical point of view, insofar as computing Q_{pro} is concerned, these values are universally good and do not need to be varied across images or in different context (or else this would defeat the very purpose of automatic parameter selection!) So we can compute the metric Q_{pro} with the same patch size and significance level in all the various experiments shown in this section. Same settings are also applied to metric Q for comparison.

3.5.3 Experimental Results

We focus on two recent state of the art denoising algorithms. One is the Locally Adaptive Regression Kernels (LARK) method of Takeda *et al.* [89], where there are two main parameters to tune: the global smoothing parameter h, and the iteration number. The effect of these parameters is generally interdependent in that the smaller the h is, the more iterations are needed to achieve the best output image [89]. Hence in practice, in makes sense to set h to a fixed value (we set it to 2.0 throughout) and to attempt to optimize the iteration number within a reasonable range (which we select to be between 1 and 20.)

The second algorithm is the BM3D algorithm [28], which is considered to be the state of the art denoising algorithm at the time of this writing. In the BM3D filter, a Wiener filter is employed for collaborative filtering, which requires the estimate of a (variance) parameter σ_{est}^2. The value of the parameter σ_{est} can strongly affect the

output, and thus needs to be tuned. In the following experiments, we optimize this parameter in the broad range of 1 to 30.

In what follows, both simulated and real data are considered. In the simulated experiments, where the reference image is available (but not used in our case), MSE is also computed for the sake of comparison even though this calculation is not practical, and MSE is not a good visual metric anyway. Monte-Carlo SURE [98], on the other hand, offers a way to access MSE without a reference, and it is used for our experiments with a real noisy image.

3.5.3.1 Simulated Experiments

In this set of experiments, we simulated noisy data by adding white Gaussian noise to three clean images shown in Fig. **3.15**; namely, *Squares* (256×256), *Motorcycle* (384×384) and *Lighthouse* (384×384). Experiments are conducted at peak signal-to-noise ratios[6] (PSNR) of 30dB, and 23dB, to test the performance of the metrics in a range of noise strengths.

Plots of the experimental results are given in Fig. **3.16-3.21**. We can observe that both metric Q and Q_{pro} were consistently effective in capturing the changing trend of quality in the output as the (LARK and BM3D) algorithm parameters were varied. As a result, the maximum values of Q and Q_{pro} yielded very good results in almost every case in a completely unsupervised fashion and without access to a reference image or an estimate of the MSE.

The behavior of the two proposed metrics provides not only optimized values for the algorithm parameters, but also tends to reveal interesting behavioral character-

[6]Peak signal-to-noise ratio is defined as $10\log_{10}(255^2/\sigma^2)$, where σ^2 is the variance of noise.

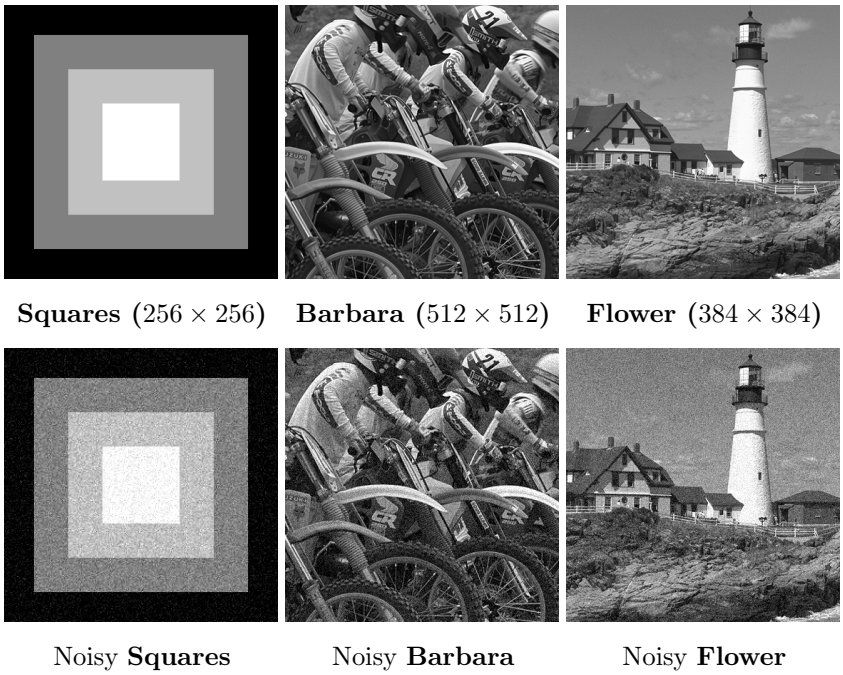


Figure 3.15: Original images that are used in the simulated experiments, and the corresponding noisy images whose PSNR = 23dB.

istics of the algorithms to which it is applied. As a case in point, we observe that as the value of the parameter being optimized (number of iterations for LARK, and σ_{est} for BM3D) increases, the overall visual quality of the output image rises first due to the suppression of random noise, and then goes down because of the blurring effect of the filter. This phenomenon happens most strongly in the experiments using two natural images (*Motorcycle* and *Lighthouse*) – see Figures **3.18 - 3.21**. In the *Squares* case, where the image content is relatively simple, the edges are successfully preserved by the BM3D filter even when the σ_{est} is set to be very large. This feature is also seen in the curves of MSE, Q and Q_{pro} (see Fig. **3.16**), where we can see the curves flattening out as the image quality changes little. The LARK algorithm, on the other hand, is more sensitive to increasing number of iterations, even when the content is relatively simple as in the *Squares* case, as can be seen in Fig. **3.17**. Meanwhile, we can also

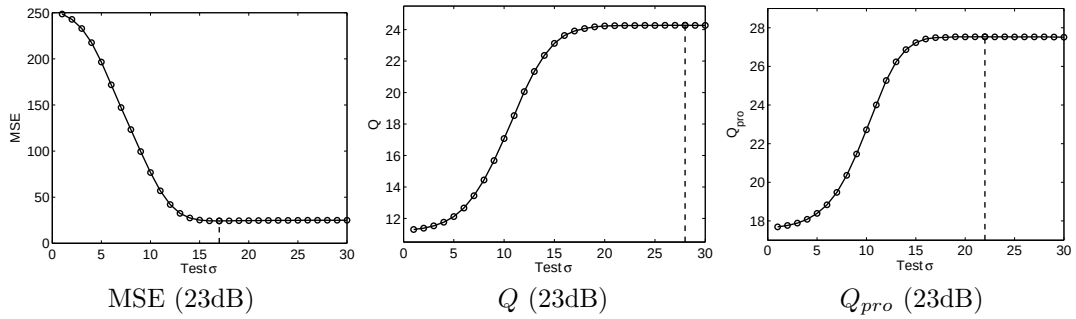
observe that Q_{pro} performs better than Q in preserving image details in some cases. For example, in the *Lighthouse* image related experiments (see Fig. **3.21** and **3.20**), Q optimized images are slightly over-smoothed, while Q_{pro} optimized images are visually good in both noise suppression and image detail preservation. This is probably due to the weak ability of structure capturing of Q. Fig. **3.22** illustrates the valid patch maps of *Lighthouse* for both Q and Q_{pro}, where we can see that the map for Q_{pro} includes more structured patches than the one for metric Q.

For the sake of completeness, we also tried the no-reference sharpness metric JNB [4] to test its behavior and found that in general it fails to capture the trend of quality change since it cannot handle noise well. One example is provided in Fig. **3.23**.

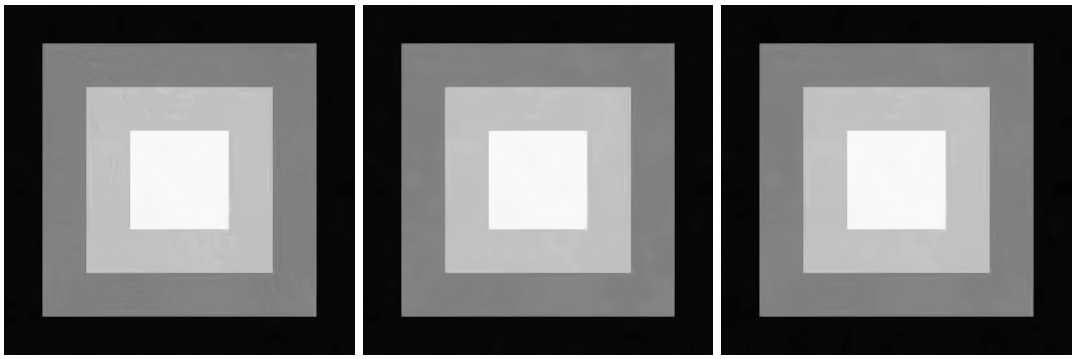
3.5.3.2 Real Noise Experiments

Fig. **3.24** illustrates a test image *JFK* (367×343) that suffers from real noise. The noise comes from film grain, scanning and compression processes, and is not Gaussian – indeed its variance is space varying. For lack of a better yardstick, we use the Monte-Carlo SURE [98] method for comparison, where the standard deviation of the noise (assumed to be globally constant) is estimated through the median absolute deviation (MAD) method [82]. The measured values for the above test image is $\sigma = 4.2$. In implementing the Monte-Carlo SURE, the standard deviation of the probing noise is set to be 0.1 as recommended in [98].

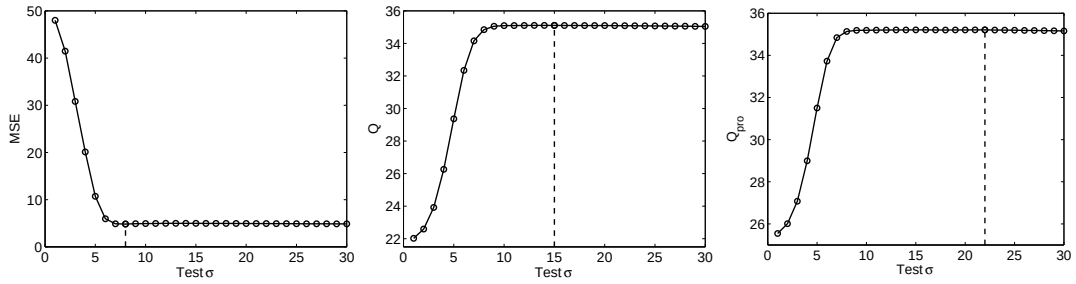
Again, LARK and BM3D filters are employed in this set of experiments. The plots of SURE, Q and Q_{pro} versus the tuning parameter for the LARK algorithm, and the corresponding optimized output images are provided in Fig. **3.26**. For the LARK



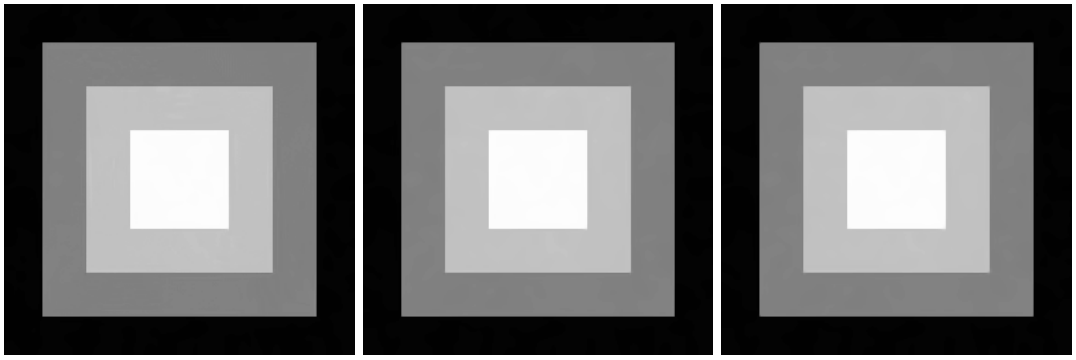
MSE (23dB) Q (23dB) Q_{pro} (23dB)



MSE optimized, $\sigma_{est} = 17$ Q optimized, $\sigma_{est} = 28$ Q_{pro} optimized, $\sigma_{est} = 22$

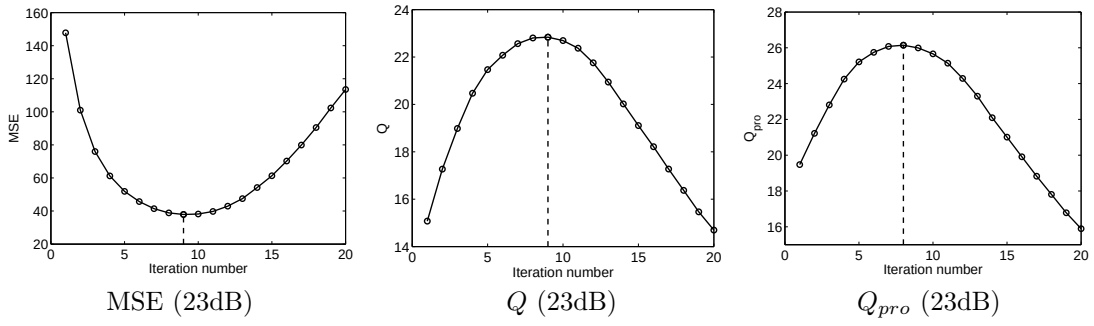


MSE (30dB) Q (30dB) Q_{pro} (30dB)



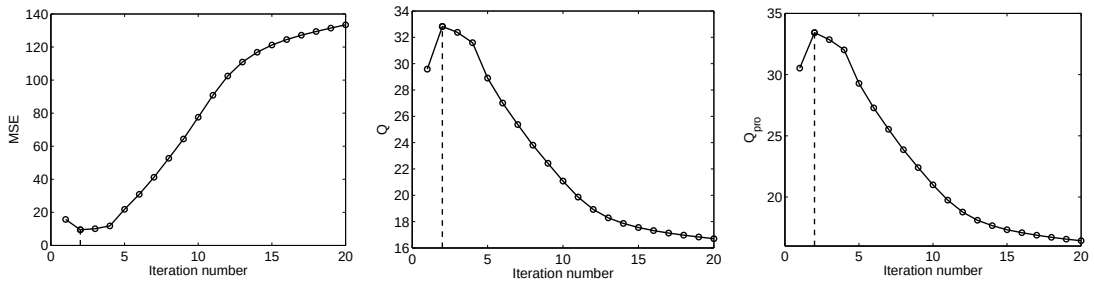
MSE optimized, $\sigma_{est} = 8$ Q optimized, $\sigma_{est} = 15$ Q_{pro} optimized, $\sigma_{est} = 22$

Figure 3.16: Plots of MSE , metric Q and Q_{pro} versus the tuning parameter in **BM3D** denoising using image **Squares** (with PSNR 23dB, 30dB), and the optimized filtered images.

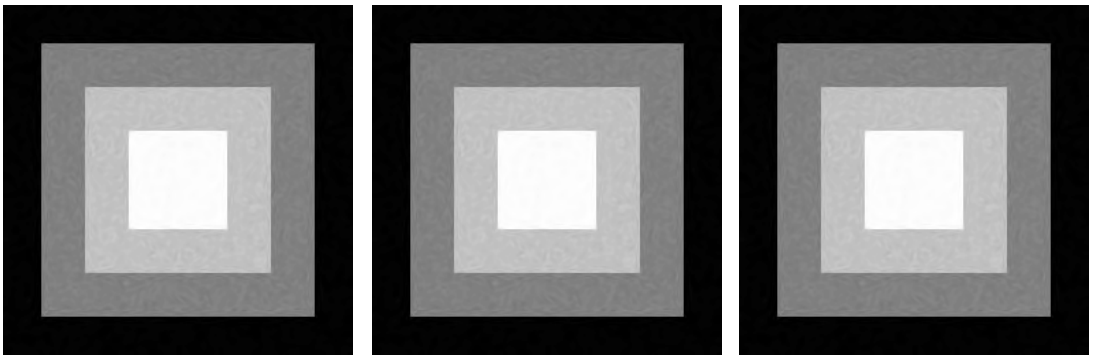


MSE (23dB) Q (23dB) Q_{pro} (23dB)

MSE optimized, 9th iteration Q optimized, 9th iteration Q_{pro} optimized, 8th iteration

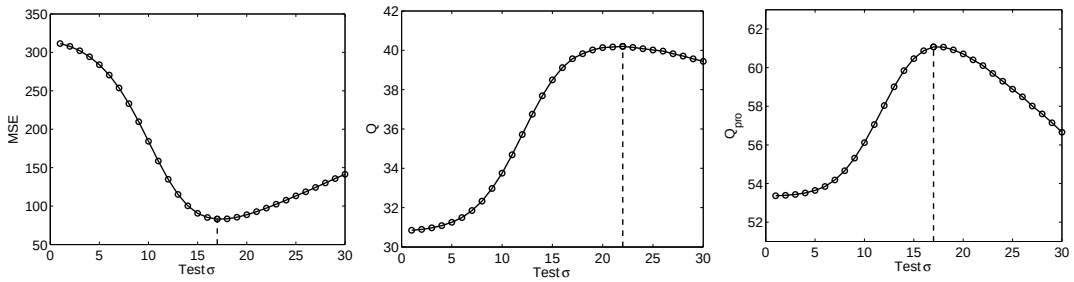


MSE (30dB) Q (30dB) Q_{pro} (30dB)



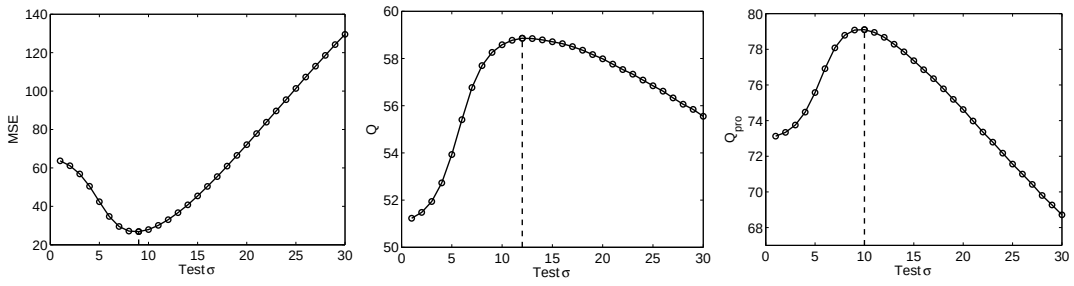
MSE optimized, 2nd iteration Q optimized, 2nd iteration Q_{pro} optimized, 2nd iteration

Figure 3.17: Plots of MSE , metric Q and Q_{pro} versus the tuning parameter in LARK denoising using image **Squares** (with PSNR 23dB, 30dB), and the optimized filtered images.



MSE (23dB) Q (23dB) Q_{pro} (23dB)

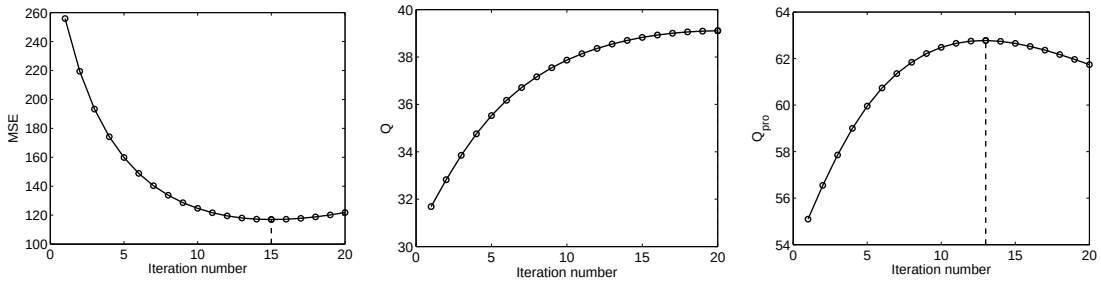
MSE optimized, $\sigma_{est} = 17$ Q optimized, $\sigma_{est} = 22$ Q_{pro} optimized, $\sigma_{est} = 17$



MSE (30dB) Q (30dB) Q_{pro} (30dB)

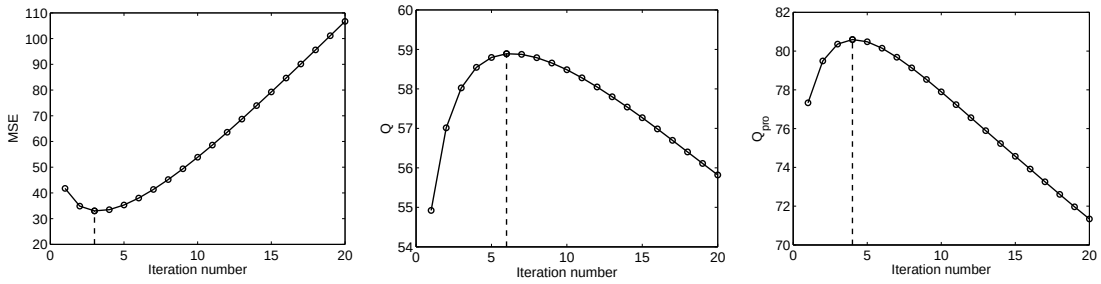
MSE optimized, $\sigma_{est} = 9$ Q optimized, $\sigma_{est} = 12$ Q_{pro} optimized, $\sigma_{est} = 10$

Figure 3.18: Plots of MSE , metric Q and Q_{pro} versus the tuning parameter in **BM3D** denoising using image **Motocycle** (with PSNR 23dB, 30dB), and the optimized filtered images.



MSE (23dB) Q (23dB) Q_{pro} (23dB)

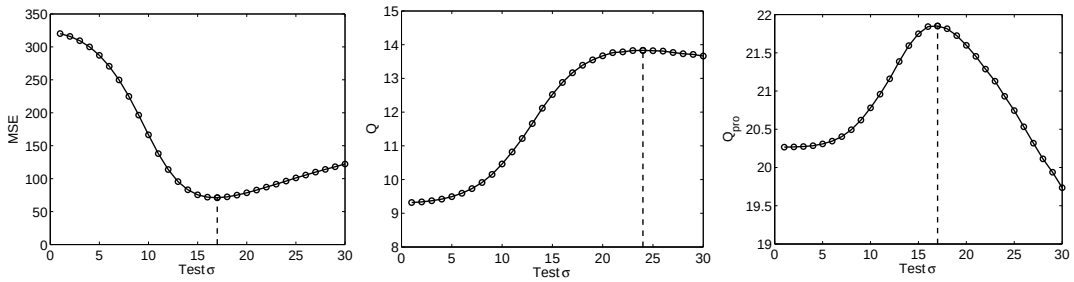
MSE optimized, 15th iteration Q optimized, 20th iteration Q_{pro} optimized, 13th iteration



MSE (30dB) Q (30dB) Q_{pro} (30dB)

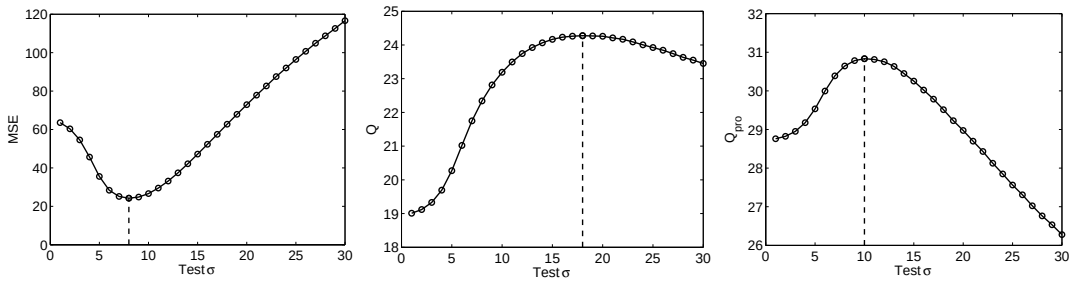
MSE optimized, 3rd iteration Q optimized, 6th iteration Q_{pro} optimized, 4th iteration

Figure 3.19: Plots of MSE , metric Q and Q_{pro} versus the tuning parameter in LARK denoising using image **Motocycle** (with PSNR 23dB, 30dB), and the optimized filtered images.



MSE (23dB) Q (23dB) Q_{pro} (23dB)

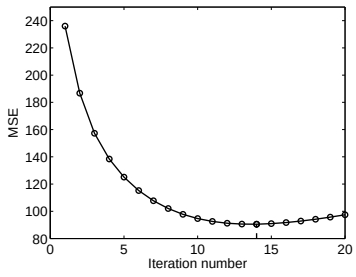
MSE optimized, $\sigma_{est} = 17$ Q optimized, $\sigma_{est} = 24$ Q_{pro} optimized, $\sigma_{est} = 17$



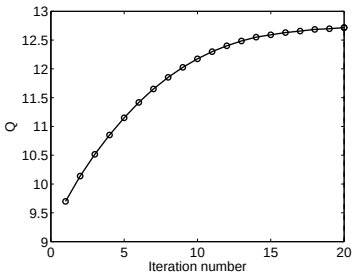
MSE (30dB) Q (30dB) Q_{pro} (30dB)

MSE optimized, $\sigma_{est} = 8$ Q optimized, $\sigma_{est} = 18$ Q_{pro} optimized, $\sigma_{est} = 10$

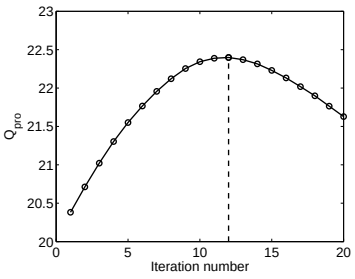
Figure 3.20: Plots of MSE , metric Q and Q_{pro} versus the tuning parameter in **BM3D** denoising using image **Lighthouse** (with PSNR 23dB, 30dB), and the optimized filtered images.



MSE (23dB)



Q (23dB)

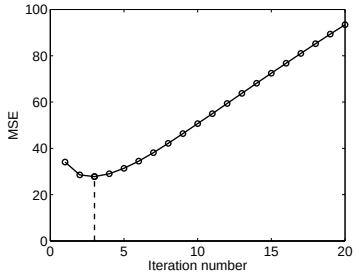


Q_{pro} (23dB)

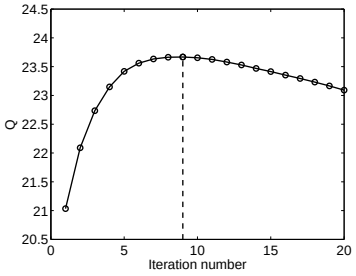
MSE optimized, 14th iteration

Q optimized, 20th iteration

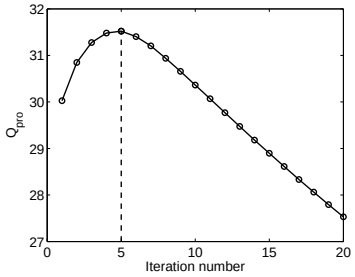
Q_{pro} optimized, 12th iteration



MSE (30dB)



Q (30dB)



Q_{pro} (30dB)

MSE optimized, 3rd iteration

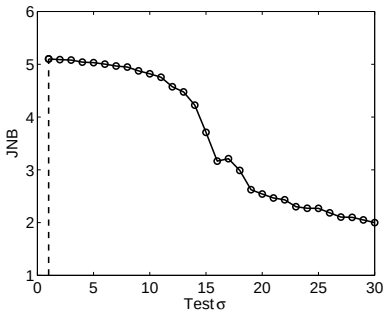
Q optimized, 9th iteration

Q_{pro} optimized, 5th iteration

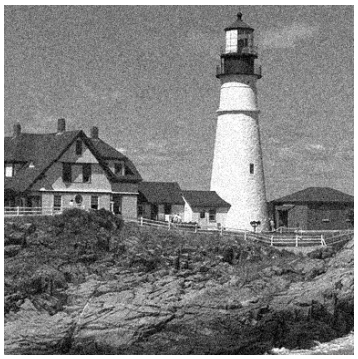
Figure 3.21: Plots of MSE , metric Q and Q_{pro} versus the tuning parameter in LARK denoising using image **Lighthouse** (with PSNR 23dB, 30dB), and the optimized filtered images.

(a) (b)

Figure 3.22: Examples of valid patch maps of Q (a) and Q_{pro} calculated from the noisy **Lighthouse** image (23dB) .



(a) JNB (23dB)



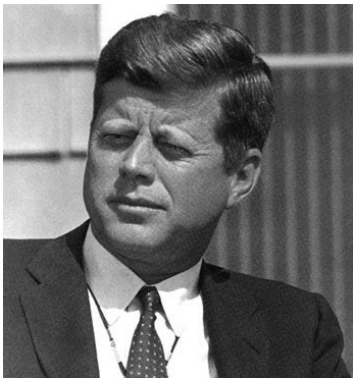
(b) JNB optimized, $\sigma_{est} = 1$

Figure 3.23: Plot of JNB [4] versus the tuning parameter in **BM3D** denoising using image **Lighthouse** (with PSNR 23dB), and the optimized filtered image.

method, the optimal iteration number suggested by SURE is 1, which is clearly inadequate for denoising, as obvious noise can be observed in the output image. Meanwhile, images using the Q optimized iteration number 4 and Q_{pro} optimized number 3 show good visual performance in balancing between denoising and detail-preservation.

Plots of SURE, Q and Q_{pro} versus the tuning parameter for the BM3D algorithm, and the corresponding optimized output images are provided in Fig. **3.25**. Again, SURE optimized image looks noisy. Q optimized image is much cleaner, but it is also mildly over-smoothed. Q_{pro} optimized image provides excellent visual quality in terms of both noise suppression and detail-preservation.

Through the above real noise data experiments, it can be seen that the Monte-Carlo SURE method did not give adequate parameters for the filters when testing on the image *JFK*. We note that this example is not entirely fair to the SURE method since the assumptions underlying that method are violated in this example: the noise is not white Gaussian, and the estimate of its standard deviation is likely inaccurate. However, the experiment does illustrate that Q and Q_{pro} are nevertheless able to maintain their stable performance, indicating that our proposed metrics can be useful for a more general variety of practical situations. Meanwhile, it shows that Q_{pro} generally performs closer to HVS than Q since it captures more structured patches from the test image. This is consistent to the results of the perceptual data experiments in Section 3.4.

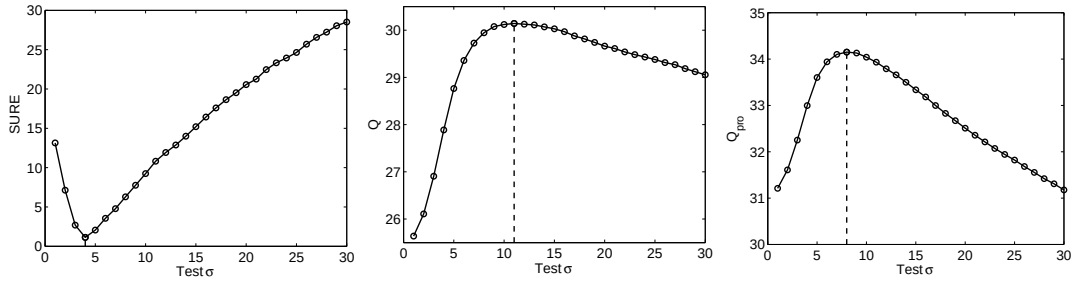


JFK

Figure 3.24: Image corrupted by real noise.

3.6 Summary

In this chapter we first analyzed the behavior of the singular values of local image gradient matrix in the presence of both changing blur and changing noise, and introduced the metric Q that responses reasonably to both blur and noise in anisotropic image regions. We then extended the measuring idea to isotropic structures and proposed the metric Q_{pro}, which is valid to both isotropic and anisotropic regions as long as they contain sufficiently strong structures. Simulations illustrate that by monitoring local structured regions, Q_{pro} is capable of globally assessing image quality. Tests using the TID2008 database show that the proposed metric Q_{pro} correlates well with subjective quality evaluations for both blur and noise distortions. Ample simulated and real data experiments also illustrate that this metric can be used to automatically optimize the tuning parameters of any given denoising filters so that they can generate visually the best output images.



SURE $\qquad$ Q $\qquad$ Q_{pro}

SURE optimized, $\sigma_{est} = 4$ $\qquad$ Q optimized, $\sigma_{est} = 11$ $\qquad$ Q_{pro} optimized, $\sigma_{est} = 8$

SURE optimized, $\sigma_{est} = 4$ $\qquad$ Q optimized, $\sigma_{est} = 11$ $\qquad$ Q_{pro} optimized, $\sigma_{est} = 8$

Figure 3.25: Plots of SURE, metric Q and Q_{pro} versus the tuning parameter in **BM3D** denoising experiments using image **JFK**, and optimized filtered images. The 3rd row images are zoomed parts of the 2nd row images.

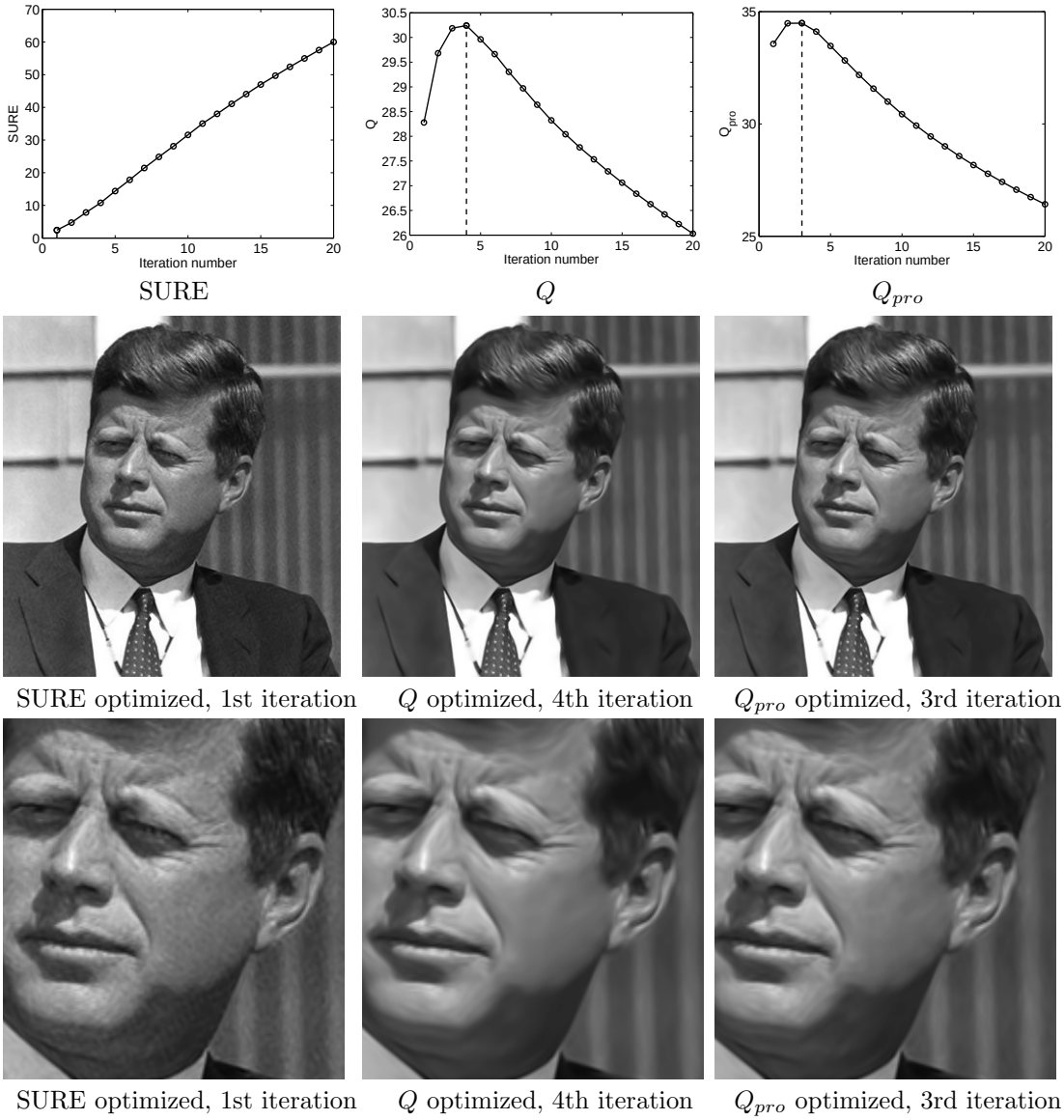


Figure 3.26: Plots of SURE, metric Q and Q_{pro} versus the tuning parameter in LARK denoising using image **JFK**, and the optimized filtered images. The 3rd row images are zoomed parts of the 2nd row images.

Chapter 4

Estimating Local Defocus Blur

Abstract - A PSF of defocus blur can be specified by a single parameter indicating its scale. In this chapter we introduce an algorithm capable of estimating a defocus scale map from a single image. This method is able to measure the probability of local defocus scale in the *continuous* domain. It also takes smoothness and color edge information into consideration to generate a coherent defocus scale map indicating the amount of blur at each pixel. Simulated and real data experiments illustrate its good performance, and its successful applications in foreground/background segmentation.

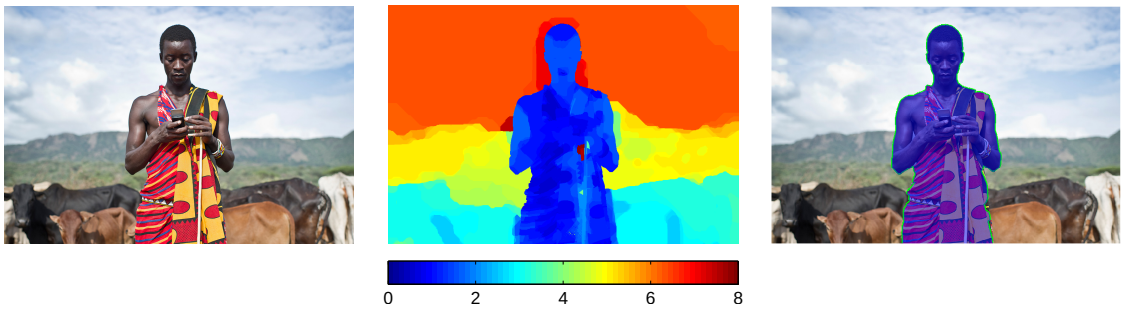
4.1 Introduction

As described in Chapter 1, the value of a *relative metric* should be independent of the latent image contrast. For example, given two patches with different contrast levels or even with completely different image contents, a relative metric should be able to compare their sharpness with respect to the peakedness of their latent PSFs. In this chapter, we address the relative sharpness measurement problem for defocus blur, given

a single image.

Optical imaging systems have a limited depth of field, which may lead to defocus blur. One way of measuring local defocus blur is by estimating the defocus PSFs directly. Most single image blind deconvolution algorithms focus on estimating spatially invariant PSFs, or shift-varying PSFs that can be treated as projections of a globally constant blur descriptor caused by camera shake [7, 8, 38, 9, 10]. However, estimating defocus blur is a challenging task mainly because the corresponding PSFs are spatially varying and cannot be represented by any global descriptor. Indeed, spatially varying defocus PSFs for a given camera can be pre-calibrated and described typically through a simple model (e.g. disc, Gaussian), which is characterized by a single parameter indicating its scale (e.g. radius, standard deviation, etc.). For an image, we call a 2-D map of the scale parameter the *defocus blur map*, which indicates the level of local blur at each pixel (see an example in Fig. **4.1**). Because the defocus blur scale is directly related to the peakedness of the PSF, such blur map also indicates local relative sharpness. In other words, given a defocus PSF model, estimating local blur scale is equivalent to measuring local relative sharpness score.

Defocus blur map estimation has several potential applications. For example, it can be employed to detect and segment in-focus subjects from the out-of-focus background, helping a photo editor or artist to edit the subject of interest (usually in focus) or the background, separately. Besides that, since defocus blur level is intimately related with depth of the scene, a blur map also provides important information for depth estimation. The computation of depth information typically requires two photos of the same scene taken at the same time but from slightly different vantage points,



(a) (b) (c)

Figure 4.1: Defocus blur map estimation experiment using a real image. (a) Test image. (b) Estimated defocus blur map. (c) Automatic foreground/background segmentation.

i.e. a stereo pair [41]. However, in most cases only one image is available. A blur map allows one to reconstruct a 3-D scene from a single photograph as long as the camera settings (focal length, aperture settings, etc.) are known. Even without these camera settings, an estimated blur map can still (at least coarsely) depict the 3-D geometric information (see Fig. **4.1** (b)). For image restoration applications, if both the defocus PSF calibration and blur map estimation are made, we can reconstruct an all-in-focus image through a non-blind spatially varying deblurring process (see the example in Fig. **4.8**).

In [37] Levin et al. proposed an algorithm that simultaneously restores a sharp image and a depth map from a single input. This method locally selects the best PSF by evaluating its deconvolution errors. It requires a specially designed aperture filter for the camera, which strongly limits its domain of application. Instead of estimating the optimal blur scale in the continuous domain, it can only identify the most likely candidate from a finite number of calibrated PSFs with somewhat limited accuracy. Chakrabarti et al. suggested a method estimating the likelihood function of a given

candidate PSF based on local frequency component analysis without deconvolution [102]. In their paper the method is applied to detect simple motion blur, but it can also be employed for defocus blur identification. Again it can only detect optimal PSFs from a finite number of candidates.

In this chapter we propose a new method for estimating defocus PSF scale at each pixel. The estimation is based on local frequency analysis similar to [102], but is significantly more general since it is carried out in the continuous domain. Smoothness constraints and image color edge information are also taken into consideration to generate a map that is smooth and meanwhile allows discontinuity in the boundary regions[1] between objects. This algorithm does not rely on any specific functional model of the PSFs and is therefore very generally applicable. It can be implemented using any PSF model that is a function of a single parameter. As we illustrate in Section 4.4, even without accurate PSF calibration and modeling the method can still roughly tell local blur level for real images by employing the disc function as a generic model.

A similar estimation method was proposed by Bae and Durand [5] that is used to magnify focus differences, but their blur estimation is done only at edges. Their blur map is essentially interpolated elsewhere. Their first step is an explicit edge detection step, which may not be very robust to either strong blur, or noise. Since the goal in [5] is magnifying focus differences, the case of a background that is too blurry for reliable edge detection is not mentioned. On the other hand, our statistical models are applicable everywhere as long as there exists some image contrast, even where there is no clear edge that can be localized. Thus, we produce a dense set of probability distributions

[1] One example of the boundary regions is the boundary between sharp foreground objects and blurry background. Such boundary issue is also discussed in the multi-image fusion problem in Section 2.2. However, here we assume that only one observed image is be available.

versus blur radii over the image. Besides, the method of [5] models only step edges with a Gaussian blur PSF, while ours does not need such requirement on blur PSF model.

Our continuous blur radius modeling discussed in section 4.2 leads to a very accurate estimate of local blur, which in turn provides for better discrimination than [102] in separating the effects of defocus blur over noise and image content. A second important improvement over [102] is that we find and enhance 2nd and 3rd local maxima in the blur radius probability distribution at each pixel. When the global maximum does not give the correct blur radius, the 2nd or 3rd highest local maxima almost always do (see Fig. **4.5**). Our smoothness constraint then allows the method to choose the proper radius, thereby significantly reducing errors in the blur radius map.

The rest of this chapter is organized as follows. Section 4.2 gives an analysis on local image statistics to motivate the basic estimation idea. The proposed algorithm is described in Section 4.3. Simulated and real data experiments are given in Section 4.4 to show the algorithm performance. We also provide application examples in this section, focusing mainly on automatic foreground/background segmentation. Known shortcomings are discussed in Section 4.6.

4.2 Local Analysis of Image Statistics

Recall that an imaging system suffering from spatially changing blur and random noise can be generally modeled as:

$$g[\mathbf{x}] = (h_{\mathbf{x}} \otimes f)[\mathbf{x}] + n[\mathbf{x}] \tag{4.1}$$

where $\otimes$ denotes a 2-D convolution operator. f and g represent the ideal all-in-focus image and the observed blurry image (in gray level), respectively. $h_{\mathbf{x}}$ is the spatially

varying blur kernel at position $\mathbf{x}$, and n denotes random noise that is assumed to be IID Gaussian: $n[\mathbf{x}] \sim \mathcal{N}(0, \sigma_n^2)$.

Because both f and $h_{\mathbf{x}}$ are unknown, the blur estimation is highly ill-posed, and thus prior knowledge about the latent image f is required. Although the distribution of f is difficult to describe, we assume that its gradient field can be locally modeled as white Gaussian. Specifically, in a small analysis window w of size $\sqrt{N} \times \sqrt{N}$ we have

$$\nabla f[\mathbf{x}] = (\nabla \otimes f)[\mathbf{x}] \sim \mathcal{N}(0, \varsigma_{\mathbf{x}}), \quad \forall \mathbf{x} \in w \tag{4.2}$$

where ∇ denotes a derivative operator in a particular direction (horizontal or vertical). $\varsigma_{\mathbf{x}}$ represents the local variance of the signal derivatives in the window w around $\mathbf{x}$. We assume that blur kernel $h_{\mathbf{x}}$ is spatially invariant inside w. For simplicity, in the rest of this chapter we use h and ς to replace $h_{\mathbf{x}}$ and $\varsigma_{\mathbf{x}}$, respectively.

It is known that information about blur can be conveniently analyzed by means of a frequency spectrum given the observed g. We first define a *localized* 2-D Fourier filter basis $\{t_i\}$, which is a set of functions over the same spatial extent as the analysis window w. Each such function represents a different spatial frequency, or a group of related spatial frequencies. Specifically, a Gabor filter is employed, which is the product of a pure sinusoid with a 2-D Gaussian function. For example, for the i-th frequency $[\omega^{(i)}, \upsilon^{(i)}]$, the function value at position $\mathbf{x} = [x, y]$ is

$$t_i[\mathbf{x}] = m[\mathbf{x}] \exp\left(-2\pi j \left(x\omega^{(i)} + y\upsilon^{(i)}\right)\right). \tag{4.3}$$

Here the 2-D Gaussian function $m[\mathbf{x}]$ is centered in the analysis window w and its standard deviation is $1/4$ of the diameter $\sqrt{N}$ of the window size. This has the advantage of tapering values down to 0 as they approach the boundary of the window. Otherwise

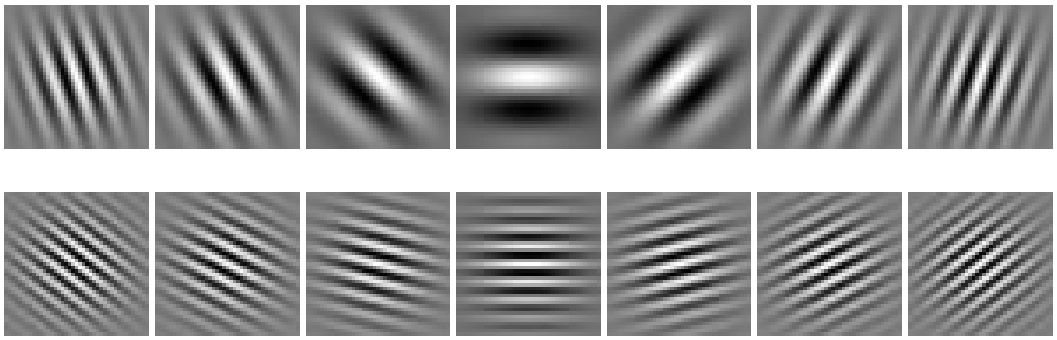


Figure 4.2: A subset of the cosine filters in our Gabor filter bank for $\sqrt{N} \times \sqrt{N}$ windows of size $\sqrt{N} = 41$.

the window boundary will appear to be sharp in the image and mask the true frequency response. A subset of the real (cosine) filters in our Gabor filter bank for $\sqrt{N} = 41$ is shown in Fig. **4.2**.

If we impose such localized Fourier analysis onto image g within the window centered at $\mathbf{x}$:

$$\nabla g_i[\mathbf{x}] = (\nabla g \otimes t_i)[\mathbf{x}], \tag{4.4}$$

then, using [102] as a starting point we can derive the likelihood function of the modulus squared of these coefficients as:

$$p(\{\nabla g_i[\mathbf{x}]\}|h) = \prod_i \mathcal{N}\left(\nabla g_i[\mathbf{x}] \,|\, 0, \varsigma\sigma_{hi}^2 + \sigma_{ni}^2\right) \tag{4.5}$$

where $\{\sigma_{hi}^2\}$ is called the *blur spectrum* for blur kernel h defined by:

$$\sigma_{hi}^2 = \sum_{\mathbf{x}} |(h \otimes t_i)[\mathbf{x}]|^2 \tag{4.6}$$

and $\{\sigma_{ni}^2\}$ is the noise spectrum:

$$\sigma_{ni}^2 = \sigma_n^2 \sigma_{\nabla i}^2 \quad \text{with} \quad \sigma_{\nabla i}^2 \triangleq \textstyle\sum_{\mathbf{x}\in w} |\nabla \otimes t_i[\mathbf{x}]|^2. \tag{4.7}$$

Note that in [102] no Gaussian window is used. Instead, a hard rectangular window is implicitly imposed on the image data. One advantage of using a hard window

is that the local power spectra can be better localized in space. However, as we know from the convolution theorem, multiplication in one domain (spatial or frequency) corresponds to a convolution in the other domain. Thus, by using a hard window in the spatial domain the frequency data is convolved with the Fourier transform of the hard window: a 2-D tensor product of sinc functions. In contrast, multiplication by a Gaussian in the spatial domain corresponds to convolution with a Gaussian in the frequency domain. Since the power of $\mathrm{sinc}(\omega)$ function falls off as $1/|\omega|$ its power is much more spread out than that of a Gaussian, and thus there is more mixing of components of the spectrum from the hard window. In real experiments with both hard and soft windows we have found that using the Gaussian window gives in more accurate and less noisy results.

Assume that the defocus PSF model h is given, and that it can be indexed by its scale parameter r: $h = h(r)$. Theoretically an optimal $\hat{r}$ could be selected by maximizing the likelihood function (4.5) if both ς and σ_n^2 are given:

$$\hat{r} = \arg\max_r p\left(\{\nabla g_i[\mathbf{x}]\} \,|\, h(r), \varsigma\right) \tag{4.8}$$

The noise variance σ_n^2 is assumed to be spatially constant and can be estimated by many approaches, for example [103]. However, the variance of the latent image gradients ς is unknown and is difficult to estimate directly.

From (4.5) we estimate the conditional likelihood function as

$$p\left(\{\nabla g_i[\mathbf{x}]\}|h\right) \propto \max_\varsigma \prod_i \mathcal{N}\left(\nabla g_i[\mathbf{x}] \,|\, 0, \varsigma\sigma_{hi}^2 + \sigma_{ni}^2\right) \tag{4.9}$$

where the optimal $\hat{\varsigma}$ that maximizes the conditional likelihood is selected for each given h. In other words, maximizing the likelihood function (4.9) is equivalent to optimizing

it over both ς and r simultaneously:

$$< \hat{r}, \hat{\varsigma} >= \arg\max_{r,\varsigma} p\left(\{\nabla g_i[\mathbf{x}]\} \mid h(r), \varsigma\right) \tag{4.10}$$

However, it is still not quite clear why optimizing (4.5) with respect to both r and ς is a reasonable way to select the scale r, since we do not have any prior knowledge about r or ς.

To further analyze the behavior of the likelihood function (4.5) over r and ς, a simulated experiment is carried out and shown in Fig. **4.3**, where (a) shows the latent test image patch of size 41×41. We use a disc function to simulate the defocus PSF and its radius to define the scale value r. The radius of the true h that convolves the image patch is set as: $r^* = 5$. White Gaussian noise with $\sigma_n^2 = 10^{-7}$ is also added according to (4.1).[2] Then, we decompose the simulated patch (b) through equation (4.4) (where horizontal derivative filter $\nabla = [1, -1]$ is used) and calculate the likelihood value p with different ς and r based on equation (4.5). The results are plotted in (d), where a global maximum is located in the point with the true radius value. In this case, maximizing function (4.10) in the continuous domain can generate the correct r.

However, it is not guaranteed that the global maximum always indicates the true radius. Fig. **4.4** illustrates another example where we implement the same simulation as Fig. **4.3** but with a different patch (see Fig. **4.4** (a)). At this time there still exists a local maximum around the true radius value, but it is no longer the global maximum.

We repeat this experiment on overlapping patches centered at every pixel of Fig. **4.5** (b), which is uniformly convolved by the disc function with $r^* = 5$. For

[2]The pixel intensity range here is $[0, 1]$.

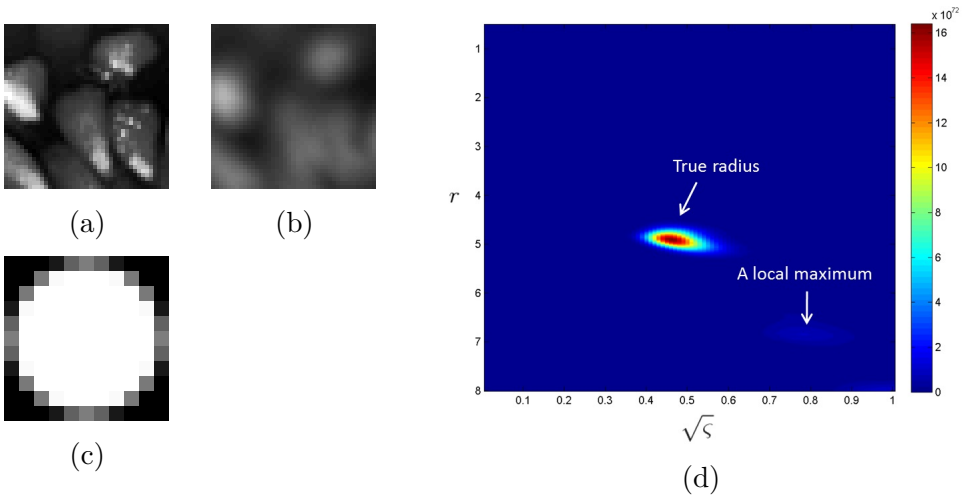


Figure 4.3: Simulated experiment based on a local patch. (a) Latent test patch. (b) Simulated blurry patch. (c) True PSF using the disc model. (d) Plot of the conditional likelihood values of (4.5) with different r and ς.

each patch, the radius $\hat{r}_1$ corresponding to the global maximum, and the radius $\hat{r}_2$ corresponding to the second highest local maximum are detected and illustrated in Fig. **4.5** (c) and (d), respectively. From (c) we can see that for most pixels the latent radii are correctly captured, but meanwhile there exist some "holes" where the maximum likelihood estimation failed (see circled regions for example). At the same time, for most of these holes the correct radii values are captured by the second highest maxima (see (d)).

This phenomenon is further illustrated in Fig. **4.6** where we see plots of $\log(p(r))$, assuming optimal ς as in equation (4.9), at three different patches in the image in Fig. **4.5** blurred with the disc of radius 5. The blue plot is for one patch, where the maximum likelihood estimation gives the correct radius. The red and green plots are for the patches where the maximum likelihood estimation fails, but where there is a clear local maxima at $r = 5$.

At this point we should make a few comments as to why the maximum like-

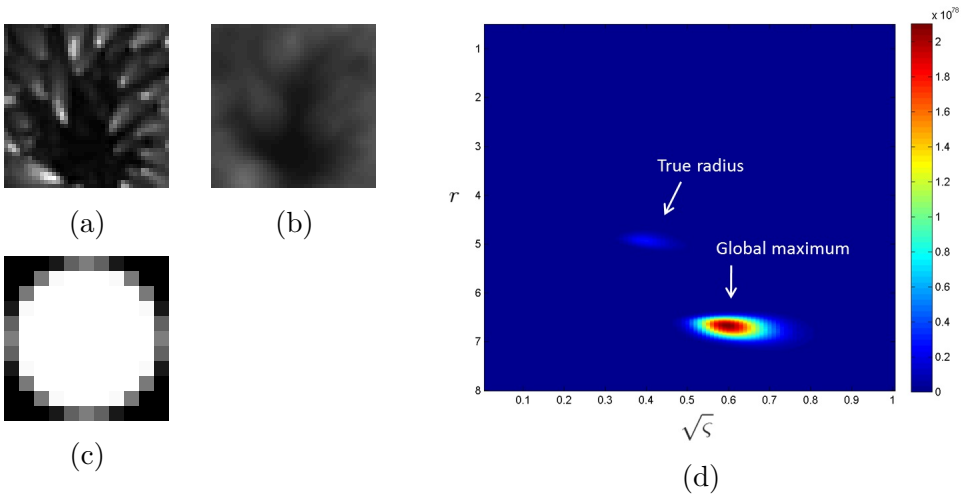


Figure 4.4: Simulated experiment based on a local patch. (a) Latent test patch. (b) Simulated blurry patch. (c) True PSF using the disc model. (d) Plot of the conditional likelihood values of (4.9) with different r and ς.

lihood estimation fails in some cases. One factor is that we make the assumption that the latent image gradient is Gaussian distributed. However, according to recent studies on natural image statistics the gradient magnitude follows a "heavy-tailed" distribution [7], which is better approximated through a Gaussian mixture model. Another factor is the tension between wanting a small window size for localizing the blur estimate to a smaller spatial region, but also needing enough data in that window for the power spectra estimation to be statistically robust. The best tradeoff will sacrifice some in both areas. Thus, we end up using a window size that gives power spectra that are nosier than ideal. This, in turn, is another cause for occasional failure of the maximum likelihood estimation.

From the above simulations, we can conclude that:

1. Function (4.5) is non-convex over r and ς.

2. In many cases, the global maximum point of (4.5) corresponds to the latent r^*,

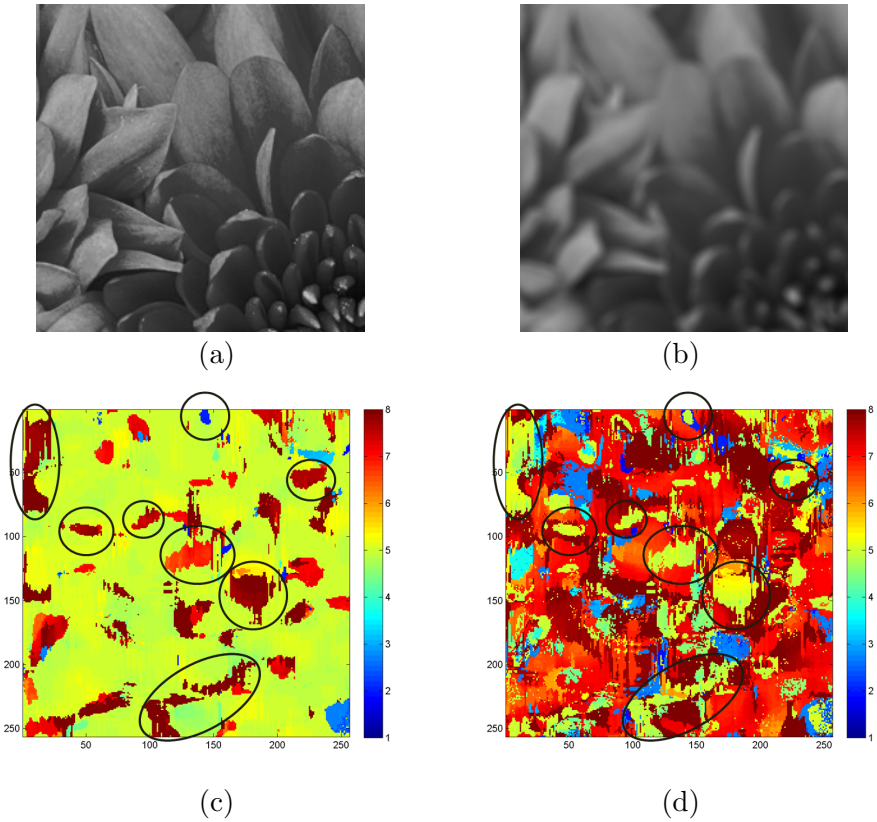


Figure 4.5: Simulated experiment based on an image. (a) Latent in-focus image. (b) Simulated blurry image convolved by a disc function with radius $r^* = 5$. (c) Estimated radii map corresponding to the global maxima. (d) Estimated radii map corresponding to the second highest local maxima. In the circled regions, the true radii values are missed by the global maxima, but captured by the second highest local maxima.

but this is not guaranteed.

3. For most cases, the true radius value r is located in a local/global maximum with a relatively high probability.

For the maximum likelihood estimation in (4.5), because we don't have any prior on either r or ς, its accuracy is limited. However, function (4.5) still provides candidate r's for most patches. If priors or constraints about r can be taken into account, then it is possible to improve the quality of blur map estimation further as we describe in the next section.

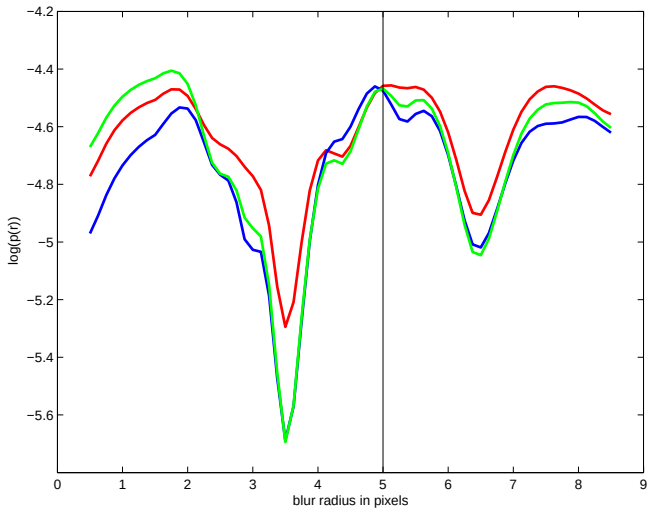


Figure 4.6: $\log(p)$ versus r at three different points in the image of Fig. **4.5** in blurred with disc of radius $r^* = 5$.

4.3 Proposed Algorithm

Our blur map estimation approach includes two main steps:

1. Local PSF probability estimation;

2. Coherent map labeling.

Given an input color image, the first step estimates up to 3 candidate scale r values for every pixel in its luminance channel: the candidate r values correspond to the global/local maxima of function (4.5) with the highest likelihood values, and they are calculated in the continuous domain. The second step creates a coherent blur map based on the estimate of the first step, image derivative information, and a smoothness prior.

4.3.1 Local PSF Probability Estimation

To find the most important local maxima of (4.5) we use a fixed point iteration, namely, calculating the optimal r or ς iteratively with the other variable fixed.

In the defocus blur situation the blur spectrum $\{\sigma_{hi}^2\}$ is solely determined by r, i.e. $h = h(r)$. Thus, in this section we use the notation $\sigma_i^2(r)$ to describe σ_{hi}^2. It has been deduced in [102] that given a fixed set of blur spectra $\{\sigma_i^2(r)\}$, the optimal $\hat{\varsigma}$ maximizing (4.5) can be found through the following fixed point iteration:

$$\hat{\varsigma} = \left(\sum_i \rho_i(\hat{\varsigma}) \right)^{-1} \sum_i \rho_i(\hat{\varsigma}) \frac{|\nabla g_i[\mathbf{x}]|^2 - \sigma_{ni}^2}{\sigma_i^2(r)}, \tag{4.11}$$

where

$$\rho_i(\hat{\varsigma}) = \left(1 + \frac{\sigma_{ni}^2}{\hat{\varsigma}\sigma_i^2(r)} \right)^{-2}.$$

Although it is possible to analytically compute the blur spectrum for a disc kernel, we want to keep our system general enough to handle any blur kernel model (such as the somewhat polygonal blur kernels arising from the leaf shutters of some cameras). We therefore *fit* the function $\sigma_i^2(r)$ under a reasonably mild constraint.

Consider a given domain of r (e.g. $r \in [0, 8]$). We select samples equally spaced over the domain with a relatively small interval, say $\Delta r = 0.1$. Then a set of sample PSFs can be generated according to the PSF model, and their blur spectrum $\{\sigma_i^2(r)\}$ can be calculated by equation (4.6) for each basis function t_i. We note that these discrete samples are only used to generate the continuous fitting functions $\{\sigma_i^2(r)\}$.

Then, for each frequency i we fit the following function of r to the samples

$$\sigma_i^2(r) = \exp\left(\alpha_{i,p} r^p + \alpha_{i,p+1} r^{p+1} + \cdots + \alpha_{i,0} r^0 + \cdots + \alpha_{i,q} r^q\right) \tag{4.12}$$

For defocus PSFs, blur spectrum are likely to be close to zero in some domain locations, in which case a mild fitting error may be exaggerated when calculating the likelihood (4.5). So an exponential function is used here to promote the fitting accuracy for the

small values. A least squares criterion is used to get the best fitting function $\tilde{\sigma}_i^2(r)$ for each frequency basis function t_i.

Once the function set $\{\tilde{\sigma}_i^2(r)\}$ is available, given a fixed ς, the optimal $\hat{r}$ can be generated by maximizing (4.5), or equivalently by minimizing the following:

$$\hat{r} = \arg\min_r \sum_i \left(\frac{|\nabla g_i[\mathbf{x}]|^2}{\varsigma \tilde{\sigma}_{hi}^2(r) + \sigma_{ni}^2} + \log\left(\varsigma \tilde{\sigma}_{hi}^2(r) + \sigma_{ni}^2\right) \right) \tag{4.13}$$

Because $\tilde{\sigma}_{hi}^2(r)$ is differentiable, (4.13) can be optimized through a gradient descent algorithm. Here a steepest descent procedure is employed.

However, in a gradient descent optimization process calculating the spectrum values and their derivatives directly from function (4.12) is computationally expensive, since this process needs to be carried out for every frequency basis function at every pixel. To reduce the cost, look-up tables, which store the spectrum values, their first and second order derivatives, are employed to replace the runtime computation. Experiments show that using look-up tables takes only one-tenth the time of the runtime computation.

We can also generate these tables directly from the analytic description of the blur model, if it is available. However, in practice we may not have a parametric model for the PSFs of a given lens. In the calibration step it is impractical to get a large number of PSF samples to generate the dense look-up tables. It is easier to collect fewer PSF samples through calibration, fit the spectrum curves from the sparse samples using function (4.12), and finally get the dense look-up tables through the fitted curves. Hence the benefit of the fitting and re-sampling. Since the fitting function does not depend on any specific function of h, our system can be implemented given any PSF model indexed by a single scalar r as long as the blur spectrum $\{\sigma_{hi}^2\}$ are smooth over r.

For example, it can be a model based on data collected from a particular lens used on a particular camera. A new model can be easily implemented in our system by simply replacing the fitting function set $\{\tilde{\sigma}_i^2(r)\}$.

The above fixed point iteration process optimizing (4.5) at each pixel $\mathbf{x}$ is summarized as follows:

1. Set $l = 0$, and initialize r^l.
2. Compute $\varsigma^{l+1} = \arg\max_{\varsigma} p\left(\{\nabla g_i[\mathbf{x}]\}|h(r^l), \varsigma\right)$ by (4.11).
3. Compute $r^{l+1} = \arg\max_{r} p\left(\{\nabla g_i[\mathbf{x}]\}|h(r), \varsigma^{l+1}\right)$ by (4.13).
4. $l \leftarrow l + 1$.
5. End if stopping criterion is met, otherwise go to Step 2.

This optimization is sensitive to the initial guess r^0 since (4.5) is non-convex. To cover most local maxima, we make a set of initial values. For example, we choose the integers $1, 2, \cdots, 8$ as the initial guess and run the optimization procedure for all these values, so that most local maxima over the domain $[0, 8]$ could be captured. After such searching step, only the top 3 optimal scales $\{\hat{r}_1, \hat{r}_2, \hat{r}_3\}$ and their corresponding likelihood values $\{\hat{p}_1, \hat{p}_2, \hat{p}_3\}$ are stored for each pixel $\mathbf{x}$. These data will be sent to the following stage.

4.3.2 Coherent Map Labeling

This section discusses how to make a coherent blur map based on the previous probability estimation and other constraints (e.g. smoothness). This goal can be

achieved by minimizing the following energy function:

$$E(\mathbf{R}) = \sum_{\mathbf{x}} D_{\mathbf{x}}(r_{\mathbf{x}}) + \sum_{(\mathbf{x},\mathbf{v})\in\nu} \lambda_{\mathbf{x},\mathbf{v}} V(r_{\mathbf{x}}, r_{\mathbf{v}}), \tag{4.14}$$

which includes two major terms: a data term $D_{\mathbf{x}}(r_{\mathbf{x}})$ reflecting fidelity to the previous probability estimation at position $\mathbf{x}$, and a smoothness term $V(r_{\mathbf{x}}, r_{\mathbf{v}})$ regularizing the output. The smoothness parameter $\lambda_{\mathbf{x},\mathbf{v}}$ controls the strength of this constraint, and is adaptive to local image content. ν is the set of pairs of neighboring pixels. In our system, given pixel $\mathbf{x}$, only the 8 surrounding pixels are considered for the smoothness term. $\mathbf{R} = \{r_{\mathbf{x}}\}_{\mathbf{x}}$ denotes a solution over all positions.

Because the data term is highly non-convex, estimating the optimal solution in the continuous domain is not trivial. To use existing optimization techniques, without introducing too much error, a discrete labeling procedure is carried out. In the blur map labeling problem, labels are discrete r from a finite set φ of possible values. Note that as long as the possible labels within the required range are sufficiently dense, we can still get a good approximation to the continuous solution.

It may seem strange that we went though considerable effort in the preceding local probability estimation to obtain the exact r for the top three local maxima in the continuous domain and now switch to a discrete domain for r in this phase. However, the effort to estimate r in the continuous domain is not wasted. The values attained at various local maxima in the $p(r)$ function can be very close, as can be seen in Fig. **4.6**. A discrete sampling could miss a local maximum or return a lower $p(r)$ that is actually attained. It is the detection of these local maxima and the values attained at them that are most important; the exact value of r at which the maximum is attained does not require pinpoint accuracy. Thus, the information gained in the preceding probability

estimation step will not be lost if we round r, but not $p(r)$, to a discrete value.

Theoretically, the data term should give the fidelity cost of $r_{\mathbf{x}}$ assigning to $\mathbf{x}$ with respect to the likelihood values from equation (4.5). However, using the values directly from (4.5), such as $D_{\mathbf{x}}(r) = -\log p(h(r))$, does not perform well. It is computationally expensive, and it does not give sufficient prominence to the top of local maxima. So in our system for pixel $\mathbf{x}$, given the estimated candidates $\{\hat{r}_1, \hat{r}_2, \hat{r}_3\}$ and their corresponding likelihood values $\{\hat{p}_1, \hat{p}_2, \hat{p}_3\}$ from the first estimation step, an *artificial* discrete likelihood array $\tilde{p}_{\mathbf{x}}(r)$ are made through the following scheme (see Fig. **4.7**):

1. Create an empty array $p_{\mathbf{x}}(r) = 0$, where $r \in \varphi$.

2. Set $p_{\mathbf{x}}(\hat{r}_l) = \hat{p}_l$, $l = 1, 2, 3$.

3. Convolve $p(r)$ with a symmetric 1D kernel κ. Then, normalize $p_{\mathbf{x}}(r) \otimes \kappa$ to sum to 1 to get an array $\tilde{p}_{\mathbf{x}}(r)$.

We set $\kappa = [10^{-20}, 10^{-12}, 10^{-7}, 10^{-3}, 10^{-1}, 1, 10^{-1}, 10^{-3}, 10^{-7}, 10^{-12}, 10^{-20}]$.

This convolution array is just wide enough so that similar, but not exactly equal, r values that are at adjacent pixels do not incur a large penalty in the smoothness, V, term of the energy function.

Finally, we let $D_{\mathbf{x}}(r) = -\log \tilde{p}_{\mathbf{x}}(r)$. Since in the labeling problem only a finite set of labels need to be considered, such an array can sufficiently describe the data function $D_{\mathbf{x}}(r)$.

A simple and efficient V function for creating a coherent blur map is

$$V(r_{\mathbf{x}}, r_{\mathbf{v}}) = |r_{\mathbf{x}} - r_{\mathbf{v}}| \tag{4.15}$$

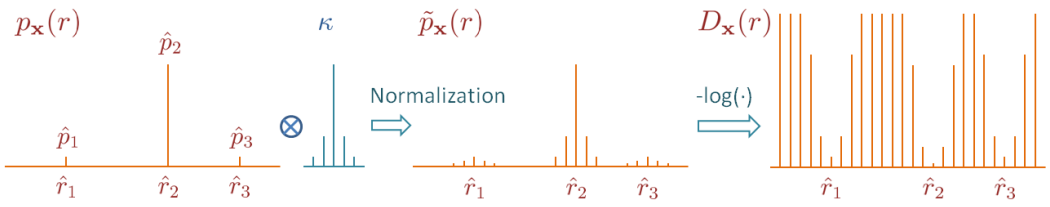


Figure 4.7: Making the artificial likelihood array.

The bigger the difference between the scales, the larger the penalty becomes. There is zero cost to setting adjacent pixels with the same scale value. This smoothness term can reduce the noise effect in the data term, correcting the errors caused in the first probability estimation stage.

However, such smoothness constraint may also blur the boundaries between different focus planes. To encourage the discontinuity of the blur map to fall along object edges, we define the smoothness parameter as:

$$\lambda_{\mathbf{x},\mathbf{v}} = \lambda_0 \exp\left(-\frac{\|\mathbf{A}_{\mathbf{x}} - \mathbf{A}_{\mathbf{v}}\|^2}{2\sigma_\lambda^2}\right) \tag{4.16}$$

Here λ_0 is a global parameter controlling the overall strength of the smoothness term. $\mathbf{A}_{\mathbf{x}}$ is a 3×1 vector containing the RGB values of pixel $\mathbf{x}$ of the input color image. Color is an important and effective feature for object distinguishing, because different objects tend to have different colors. $\|\mathbf{A}_{\mathbf{x}} - \mathbf{A}_{\mathbf{v}}\|^2$ measures the color difference between $\mathbf{x}$ and $\mathbf{v}$. σ_λ is another tuning parameter. In general, the value of $\lambda_{\mathbf{x},\mathbf{v}}$ decreases if the color distance between pixel $\mathbf{x}$ and $\mathbf{v}$ is large, protecting the boundaries between the objects in different focus planes.

In our system, α-expansion is used to minimize the energy function (4.14) [104].

4.3.3 Foreground/background segmentation

Besides blur map labeling, another interesting application of the proposed coherent map estimation method is foreground/background segmentation, which labels the in focus foreground subject from the rest of the input image. However, in this case only a binary labeling map is required. This goal can be easily achieved using the same labeling form as (4.14):

$$E(\phi) = \sum_{\mathbf{x}} D_{\mathbf{x}}(\phi_{\mathbf{x}}) + \sum_{(\mathbf{x},\mathbf{v})\in\nu} \lambda_{\mathbf{x},\mathbf{v}} V(\phi_{\mathbf{x}}, \phi_{\mathbf{v}}), \tag{4.17}$$

where $\mathbf{\Phi} = \{\phi_{\mathbf{x}}\}$ denotes a binary labeling solution. $\phi = 0$ is the blurry label, and $\phi = 1$ is the in-focus label. The data term in this case can be simplified as:

$$D_{\mathbf{x}}(0) = -\log \max_{r>\tau} \tilde{p}_{\mathbf{x}}(r),\ D_{\mathbf{x}}(1) = -\log \max_{r\leq\tau} \tilde{p}_{\mathbf{x}}(r) \tag{4.18}$$

where τ represents the in-focus threshold. So if there exists a large blur ($r > \tau$) with high probability, then there is a low cost $D_{\mathbf{x}}(0)$ of labeling pixel $\mathbf{x}$ as blurry. Similarly, if there exists a small blur $r \leq \tau$ with high probability, then there is a low cost $D_{\mathbf{x}}(1)$ of labeling pixel $\mathbf{x}$ as sharp.

The smoothness term here is defined the same as (4.15). Again, we only use the 8 surrounding pixels for ν.

4.4 Experimental Results

Both simulated and real data experiments are carried out to test the performance of the proposed defocus blur estimation framework. In the local probability estimation step, we use square windows with side length $\sqrt{N} = 41$. Our default noise setting is $\sigma_n^2 = 10^{-4}$. The coherent blur maps choose the blur radius r from the set

$\{0, 0.1, 0.2, \ldots, 7.9, 8\}$. Our default parameter settings for the coherent blur labeling are $\lambda_0 = 20$ and $\sigma_\lambda = 0.1$ (for intensities in the range $[0, 1]$). The settings for the binary foreground/background segmentation problem are $\tau = 2$, $\lambda_0 = 1000$ and $\sigma_\lambda = 0.04$.

Unless otherwise noted, the default parameter values are used. As can be seen in the results, the default settings work well for nearly all the test images shown in this section. In fact, the only parameter we varied in these experiments is the noise variance σ_n^2. In a few of the examples presented here, we found it useful to set $\sigma_n^2 = 10^{-6}$ (very low noise).

4.4.1 Simulated Experiments

A simulated experiment is illustrated in Fig. **4.8**, which allows us to quantitatively test the performance of the proposed method. The input image is generated according to the model in (4.1). Similar to the test in Fig. **4.3** disc functions are employed to simulate defocus PSFs. Variance of the additive white Gaussian noise is $\sigma_n^2 = 1 \times 10^{-6}$. The latent blur map is given in (c), where the blur radius continuously changes over the image space. This actually violates the assumption that local blur $h_{\mathbf{x}}$ is constant within local analysis window w. However, the proposed output seems to be robust to the violation of this assumption (see Fig. **4.8** (d)): trend of the blur change is successfully captured by our method. This is probably because we use overlapping windows. The mean-squared-error (MSE) of (d) with respect to the latent map (c) is 0.022.

Based on the blur map estimation $\{\hat{r}_{\mathbf{x}}\}$, and further the PSFs $\{\hat{h}_{\mathbf{x}}\}$ generated by the disc function, a spatially varying deconvolution procedure is carried out through

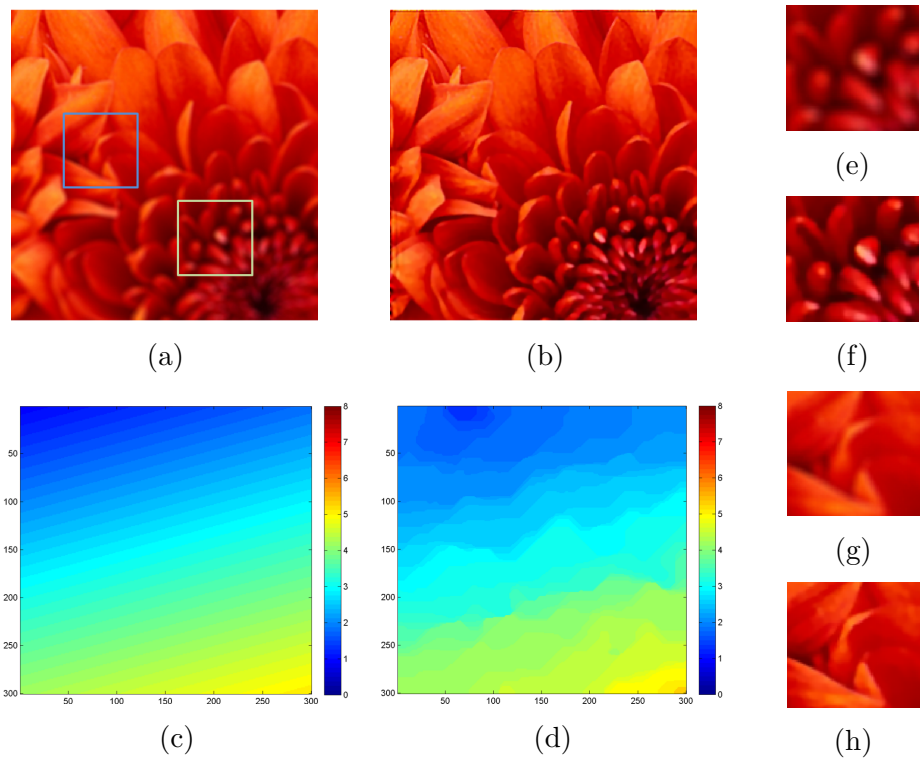


Figure 4.8: Simulated experiment for blur map estimation and spatially varying blind deblurring. (a): Simulated input image with spatially varying blur, whose PSNR is 27.3dB. (b): Deblurred image based on the proposed estimate in (d). Its PSNR is 32.3dB. (c): Latent blur map. (d): Estimated blur map. (e), (g): zoomed part of (a). (f), (h): zoomed part of (b).

the following optimization:

$$\hat{f} = \arg\min_{f} \sum_{\mathbf{x}} \left| \left(\hat{h}_{\mathbf{x}} \otimes f \right) [\mathbf{x}] - g[\mathbf{x}] \right|^2 + \lambda \sum_{\mathbf{x}} \|\nabla f[\mathbf{x}]\|_1 \tag{4.19}$$

The deblurred output $\hat{f}$ is given in Fig. **4.8** (b), where we can observe that spatially varying blurs have been successfully removed (see zoomed parts (e)-(h)). The PSNR of the original input (a) is 27.3dB, whereas the PSNR of (b) is 32.3dB with 5dB improved, which means in this experiment the accuracy of our estimation method is good enough for *blind* deblurring.

4.5 Real Data Experiments

Real image experiments are given in Fig. **4.1** and Fig. **4.9**. Because these data are collected from outside sources, the corresponding calibrated defocus PSFs are not available. However, it is known that blur from an ideal lens with a circular aperture could be modeled by the disc function in the absence of diffraction effects [105]. Since diffraction effects are almost always negligible once the blur is of visible size, we use the disc function to approximate the real PSFs in our experiments. Even though the actual blur PSF for cameras used for the test images are unknown, the disc approximation seems to be quite adequate. Our method still captures the amount of local defocus blur for all these test images, depicts 3-D geometric information for each scene, and does a good job in identifying in-focus subjects.

For example, Fig. **4.1** (a) contains four focal layers: the in-focus herdsman, the slightly defocused cattle, the background mountain and the highly blurry sky. These layers are all reflected in the output blur map (b), and the in-focus herdsman is also correctly labeled in (c). In Fig. **4.9** (a) the lizard and part of the rock are in-focus, which are correctly identified and labeled by Fig. **4.9** (b)-(c). Note that here we are not doing pure object segmentation, and that the segmentation only depends on local sharpness level (which means we are not trying to segment the lizard only from the rest of the image). Fig. **4.9** (j) illustrates another example with the blur smoothly varies over the space. Again, our blur map captures the progression of out-of-focus to in-focus to out-of-focus along the correct angle (lower left to upper right of the image) on the wood (see Fig. **4.9** (k)).

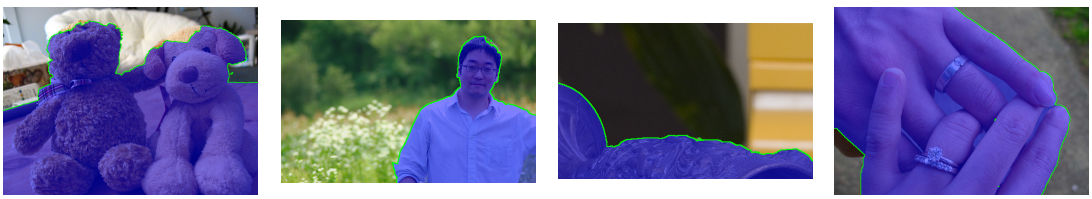
Additional real examples of our automatic binary segmentation algorithm into

in-focus and out-of-focus regions are illustrated in Fig. **4.10**.

Finally, in Fig. **4.11** we show some comparisons between the blur maps produced by Defocus Magnification [5] (DM) and by our method on some of the examples in [5]. In the DM blur maps in the middle column of Fig. **4.11**, the whiter the pixel the larger the standard deviation in their fitted Gaussian blur model and the higher the predicted blurriness. The DM approach estimates the blur only at image edges and then propagates the sparse blur estimates to the rest of the image by assuming pixels of similar intensity and color have similar blurriness.

In general, the DM blur estimation method tends to show the underlying image edges in places where the blur measure is actually smooth. Examples of this in the center column of Fig. **4.11**(b) include the nose of the dog, and the boundary of the legs of the stuffed animals. Our blur estimates are (correctly) much smoother in these areas.

In Fig. **4.11**(e), the DM result on the grass to the left of the subject has the same level of blur as the foreground subject, while ours is at two background levels: the grass closer to the subject and then the patch of white flowers further back that is clearly marked as significantly blurrier than the subject. We also capture the blur of the distant background as being much greater than other areas of the image. Automatic in-focus segmentations for the examples in Figure **4.11** are given in Fig. **4.12**.



(a) (b) (c) (d)

Figure 4.12: Automatic In-focus segmentations for the examples in Figure **4.11**.

4.6 Summary and Discussions

In this chapter we proposed a method for estimating a defocus blur map from a single image. It is capable of measuring the probability of local blur scale in the continuous domain by analyzing the *localized Fourier* (i.e. Gabor) spectrum. For each analysis window, not only the global optimum maximizing the likelihood function but also a few local optima are detected as candidate scales. Finally, color edge information and smoothness constraints are incorporated into the system to select the best candidates and generate a coherent map of the blur scale at each pixel. Experiments show that this method can be used to produce a blur map that coarsely depict the depth of the scene, help remove spatially varying blur, and segment in-focus subjects from defocused background.

One problem with our method for estimating blur maps is that we do not explicitly model the case where a window overlaps areas with different blur scales. In this case we have found that the sub-area of a window with the smallest blur size tends to dominate the probability analysis, as this sub-area contributes more power per pixel to the power spectrum. Thus, sharp areas would be enlarged by the radius of the analysis window in a blur map produced without the coherence labeling step. We rely on the coherence labeling step to snap the boundary back to the closest color boundary in the underlying image, which is usually where the actual depth discontinuity lies. But in cases of gradually changing blur, or in cases where there is not a good color boundary at the depth discontinuity, this may fail.

For the coherence labeling step to have the above correcting influence we have to set the λ parameter to a significantly large value. This can cause a slight over-

smoothing of the blur values and manifests as the blur values taking discrete steps in areas where the blur changes continuously as seen in Fig. **4.8**. The jaggedness of the contours in the same figure is partially a result of the inherent uncertainty present in the statistical computation.

The same comments apply to the foreground/background segmentation case, in that we are relying on coherence to snap the foreground mask to the nearest color boundaries in the image. Generally the results are quite good, but we occasionally see problems. Because there is increased energy to follow a serpentine contour sometimes long, thin parts of the subject of interest are cut off.

Currently our MATLAB implementation takes around 10 minutes to process a 500×500 image using a PC with a 2.70GHz CPU. Efficiency could be improved through C++ implementation. The runtime can be reduced further by using parallel computation. For example, the local optima searching process starting from the 8 different initial r values described in Section 4.3 could be done in parallel.

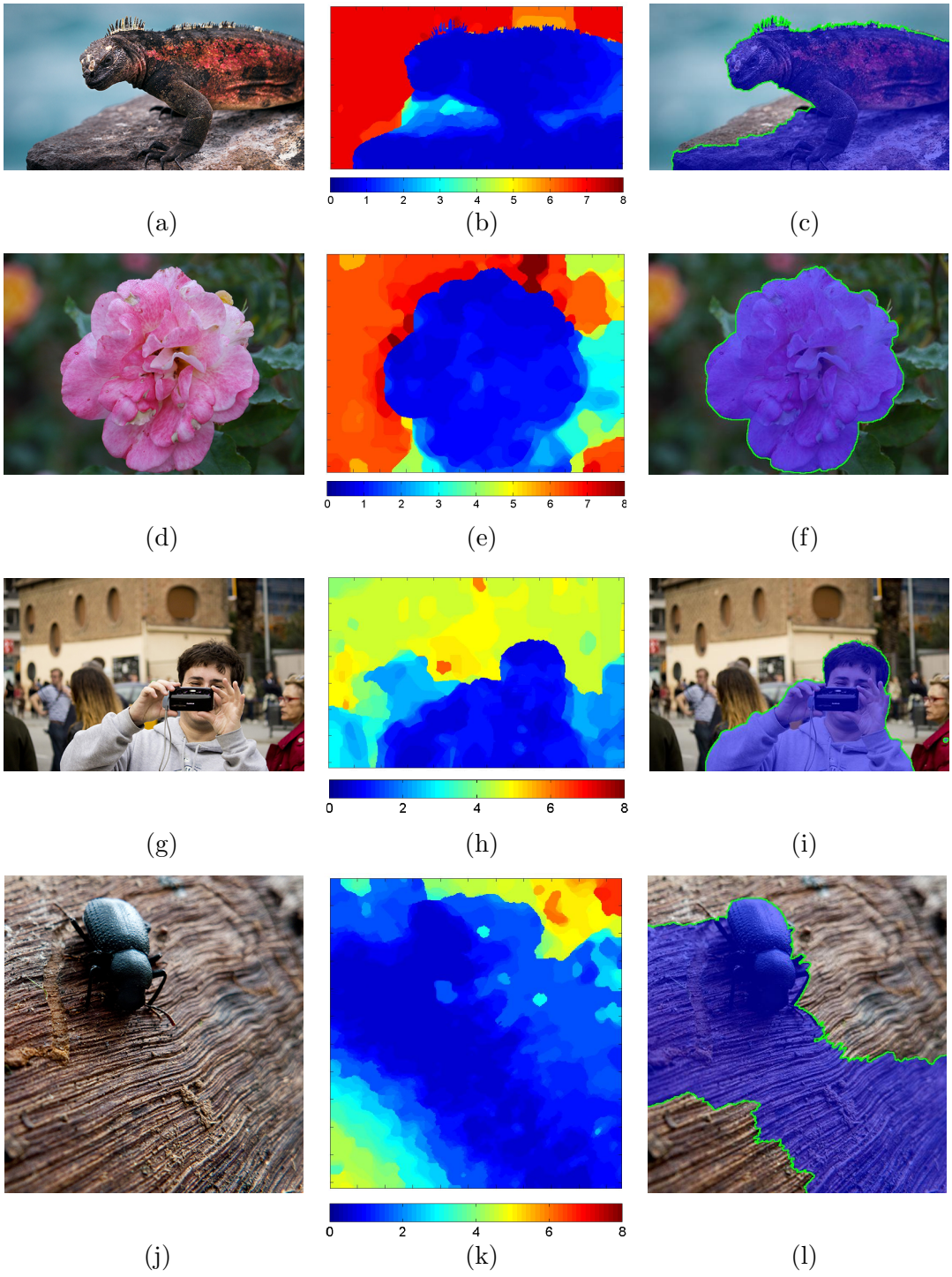


Figure 4.9: Defocus blur map estimation experiments using real images. Left column: input images. Middle column: estimated defocus blur maps. Right column: automatic foreground/background segmentation results.

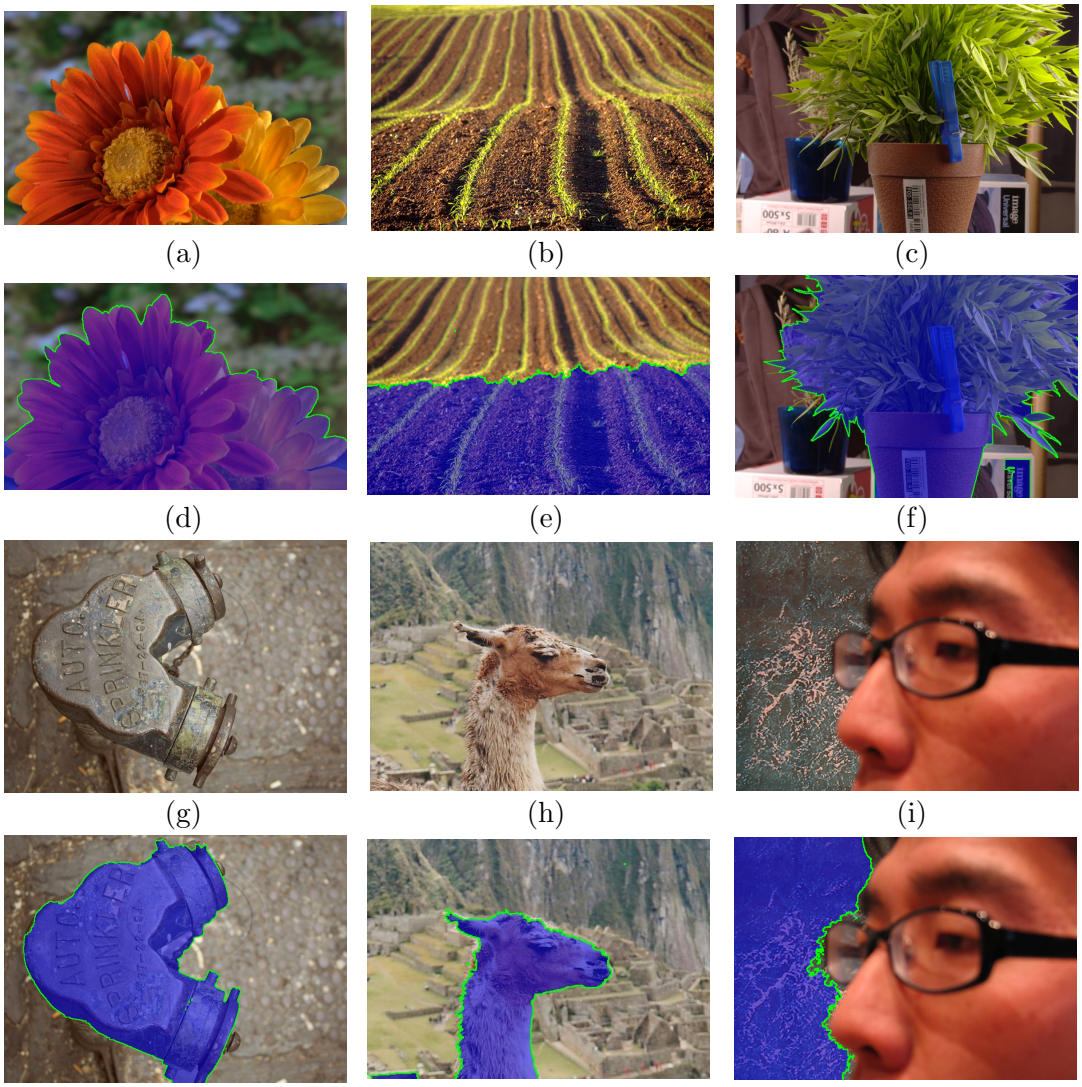


Figure 4.10: Automatic In-focus Segmentation Results. 1st and 3rd rows are input images. 2nd and 4th rows are corresponding results.

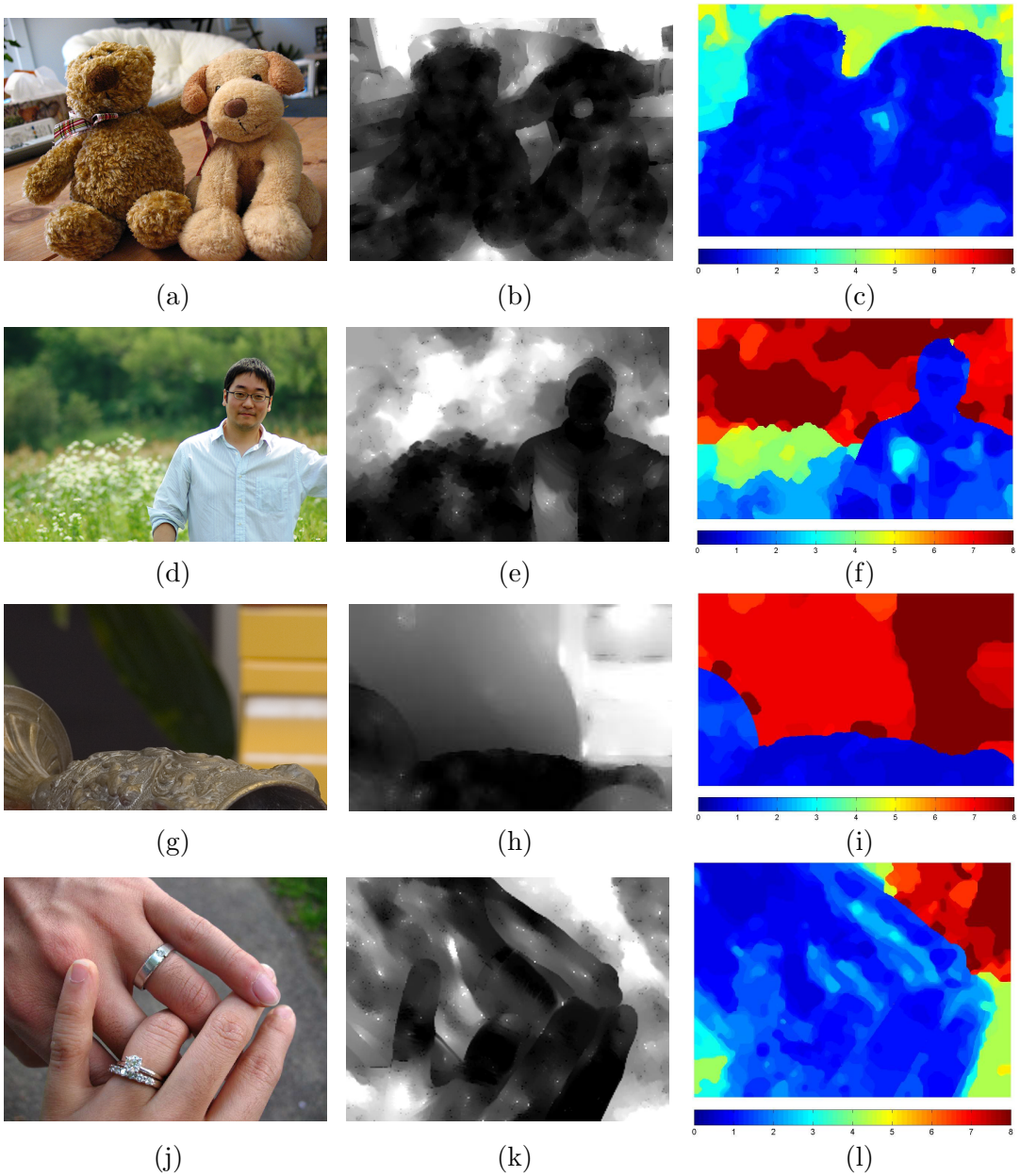


Figure 4.11: Comparison with Defocus Magnification [5] blur map results. (a) *IMG_0419*. (b) *man*. (c) *cup*. (d) *hands*.

Chapter 5

Contributions and Directions for Future Work

5.1 Summary of Contributions

In this thesis we studied the spatially varying blur measurement problem under different scenarios, introduced some image sharpness metrics, and proposed several applications based on these metrics to solve real problems.

- ▷ **Chapter 1** – We discussed different types of image blur sources and their properties. Image degradation models were introduced to represent both spatially invariant and spatially varying blurring processes. We explained the importance of image blur measurement and its potential applications. We also discussed limitations of existing blur estimation algorithms and sharpness metrics.

- ▷ **Chapter 2** – We focused on the problems related to multi-frame fusion, where a local metric capable of comparing sharpness between different copies of the same image content is required. We discussed a gradient-based sharpness metric, and

analyzed its behavior in the presence of noise with spatially uniform strength. We then proposed two fusion-based image restoration algorithms using this metric.

The first algorithm is a focus-stacking method for multi-focus image sequences. It employs the sharpness metric to roughly select the sharpest pixels within observed images at each position and merges them together to get an all-in-focus image. Different from existing focus-stacking methods, this algorithm refines its pixel selection map through an alpha matting procedure. Experiments demonstrated that such refinement could effectively reduce fusion artifacts and preserve boundary regions between objects from different focal planes.

The second proposed algorithm is about removing air turbulence caused distortions for long-distance imaging systems. Air turbulence generates spatially varying geometric deformation and blur. Our algorithm first registers each frame to suppress geometric deformation. Then, it uses a fusion process based on the sharpness metric to produce a single image that can be viewed as being convolved with a spatially invariant blur. Finally, a blind deconvolution algorithm is implemented to generate a sharp output. Experiments using real data illustrated that this approach is capable of alleviating blur and geometric deformation caused by turbulence, recovering details of the scene and significantly improving visual quality.

▷ **Chapter 3** – We considered both blur and noise, and proposed a metric called Q_{pro} which reacts reasonably to these two factors. This metric is based on singular value decomposition of local gradient matrix, and it is capable of capturing the dominant structure of local patches. The value of this metric drops as a given

structured image patch gets either more blurry or more noisy. Simulations illustrated that by monitoring local structured regions, this metric is also capable of globally assessing image quality. Tests using TID2008 database showed that the proposed metric correlates well with subjective quality evaluations for both blur and noise distortions. We also proposed a parameter tuning system based on this metric. Ample simulated and real data experiments illustrated that this system can be used to automatically optimize the tuning parameters of any given denoising filters so that they can generate visually the best output images.

▷ **Chapter 4** – We proposed a method capable of estimating the scale of local defocus blur at each pixel given a single image. It measures the probability of local blur scale in the continuous domain by analyzing the localized Fourier (i.e. Gabor) spectrum. For an analysis window centered at each pixel, not only the global optimal blur scale maximizing the likelihood function but also a few local optima are detected as candidate scales. Color edge information and smoothness constraints are incorporated into the estimation system to select the best candidates and generate a coherent map of the blur scale. Simulated and real data experiments demonstrated that this method can be used to produce a blur map that coarsely depict the depth of the scene, help remove spatially varying blur, and segment in-focus subjects from defocused background.

5.2 Future Directions

The future directions we discuss below focus on (i) how to extend our turbulence removal system to videos with complex motion, (ii) how to apply the Q_{pro} based

parameter tuning system to deblurring algorithms, and (iii) how to remove spatially varying defocus blur given a single image.

5.2.1 Turbulence Removal with Complex Motion

In Chapter 2 we introduced a system capable of removing air turbulence caused distortions for long-distance imaging systems. One limitation of this system is that it requires the camera and the scene to be static, since it can only handle the random motion generated by air turbulence. However, in practice there may exist global motion due to camera shake and local motion of objects in the scene. How to register videos with complex motion is an important issue for a robust turbulence removal system.

Global motion due to camera movement is relatively easy to deal with. One way is to globally register each frame first using a parametric model (e.g. translational, affine), and then run the existing restoration system. We can also combine the global registration procedure and the non-rigid registration step in our system (proposed in Section 2.3.2) together. Take the affine global registration model for example. In this case, we just need to replace the deformation model in equation (2.22) with the following:

$$\mathbf{W}(\mathbf{x}; \overrightarrow{\mathbf{p}}, \mathbf{a}) = \mathbf{x} + \mathbf{A}(\mathbf{x})\overrightarrow{\mathbf{p}} + \begin{bmatrix} a_1 & a_2 \\ a_3 & a_4 \end{bmatrix} \mathbf{x} + \begin{bmatrix} a_5 \\ a_6 \end{bmatrix}, \tag{5.1}$$

where the affine parameter vector $\mathbf{a} = [a_1, a_2, a_3, a_4, a_5, a_6]^T$. By optimizing the corresponding cost function with respect to both the non-rigid deformation vector $\overrightarrow{\mathbf{p}}$ and the affine vector $\mathbf{a}$, it is possible to estimate the local turbulence motion and the global camera motion simultaneously.

If local objects' movement are also considered, then accurately registering each frame could be highly challenging. A recent article has proposed detecting moving

objects from videos with a static background [106]. If the static background is of interest, then accurately detecting the moving objects can help us to avoid possible registration errors and fusion artifacts. Further research needs to be carried out in this direction to enhance the practicability and reliability of our turbulence removal system.

5.2.2 Automatically Tuning Deblurring Filters

Parameter optimization is a common problem for image restoration algorithms. In Chapter 3 we have shown that the proposed metric Q and its extended version Q_{pro} can be used in parameter optimization for denoising filters. We are also curious to see whether these metrics can be applied in optimizing general restoration algorithms, such as image deblurring algorithms, and super-resolution algorithms.

In a recent paper [107], M. Almeida and M. Figueiredo show that our Q can be successfully applied in optimizing the parameters for arbitrary *non-blind* deconvolution algorithms. In their work, two recent algorithms SpaRSA (sparse reconstruction by separable approximation) [108] and SALSA (split augmented Lagrangian shrinkage algorithm) [109] are tested, and for each algorithm both the regularization parameter and the stopping iteration number are tuned. Experiments illustrated that our metric Q has achieved state-of-the-art performance, and for most test images it works even better than their proposed methods[1] with respect to *improvement in signal-to-noise ratio* (ISNR) [107]. However, Q does not work well for *blind* deconvolution algorithms [107].

The experimental results provided in [107] are not surprising. Take the regularization parameter optimization for example. For non-blind deconvolution, where the

[1]Details about the comparison can be found in Section IV.B from [107].

(a) (b)

Figure 5.1: Blind deconvolution example from [6]. (a) Blurry image and the latent PSF. (b) An output deconvolved by an incorrectly estimated PSF, where we can see strong ringing artifacts.

latent PSF is known, the major artifacts caused by an improper setting of the regularization parameter are magnified random noise and over-smoothing (blur), which are similar to the denoising case. When recovering the attenuated frequency components of the useful image signal, non-blind deconvolution without regularization would inevitably magnify noise [110]. Strong regularization can suppress noise magnification, but it generates blur. On the other hand, insufficient regularization strength may lead to an output even noisier than the input. Hence, tuning the regularization parameter of a non-blind deconvolution method is close in spirit to tuning the denoising parameters, and thus the system proposed in Section 3.5 could be applied directly.

However, in blind deconvolution the latent PSF is not available. Mild error in the estimated PSF could produce strong ringing and halo (see Fig. **5.1**), which are *structured* artifacts [8]. Currently neither Q or Q_{pro} are able to identify these artifacts. Actually ringing and halo could be counted as useful structure by our metrics. In other words, the more ringing the output contains, the higher the value of Q/Q_{pro} becomes. This explains why Q does not perform well for the blind deconvolution experiments in [107].

Note that our metrics can quantitatively measure the amount of structured

artifacts since they also have dominant directions. Suppose we can somehow find a way to identify the measured structures as either "useful signal" (Q_{pro}^{usf}) or "artifact" (Q_{pro}^{art}), and use the following metric

$$Q_{pro} = Q_{pro}^{usf} - Q_{pro}^{art}, \tag{5.2}$$

then, it would be possible to use such metric to accurately assess the quality of blindly deblurred images. How to identify the nature of the measured structures and how to extend our metrics to images with structured artifacts merit further study.

5.2.3 Blindly Removing Spatially Varying Defocus Blur

In Section 4.4 we provided a single image blind deblurring example with spatially varying defocus blur, where local blur PSFs are estimated using the method in Chapter 4 for measuring blur scale. For blur scale measurement only, there is no need to get an accurate parametric PSF model since the continuous PSF likelihood function could be approximated through a polynomial-based fitting procedure (as explained in Section 4.3). However, for blind deblurring an accurate parametric PSF model is required, so that given an estimated local scale r we can instantly calculate its PSF $h(\mathbf{x}\,|\,r)$. So far we have only implemented such deblurring in simulated experiments where the latent PSF model is known.

Studies on PSF calibration show that mild defocus blur could be modeled as Gaussian, while large scale defocus PSFs look more like discs [105, 111]. For PSF estimation with a wide range of scale r, we need a more general model. One possible example is the *exponential power class* formulated as [112]

$$h(\mathbf{x}\,|\,\zeta,\varsigma) = \Omega^{-1} \exp\left(-c(\zeta)\left(\frac{\mathbf{x}^T\mathbf{x}}{\varsigma^2}\right)^{\frac{1}{1+\zeta}}\right),\ \varsigma > 0,\ -1 < \zeta \leq 1, \tag{5.3}$$

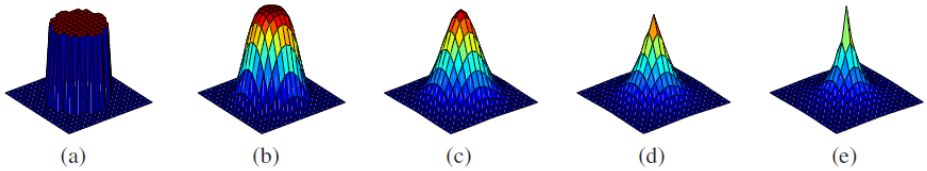


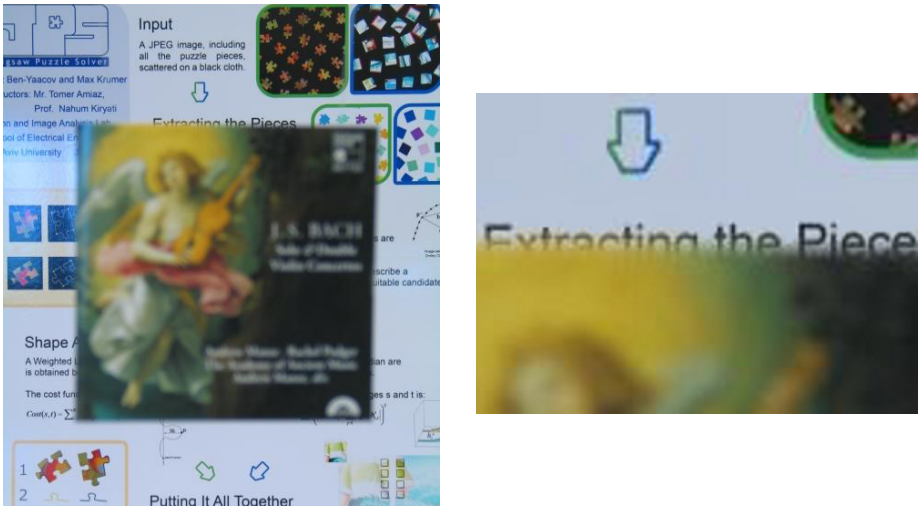
Figure 5.2: Examples of the parameterized PSF by the exponential power class with a variety of power parameter ζ: (a) -0.99 (disc blur), (b) -0.5, (c) 0 (Gaussian), (d) 0.5, and (e) 1.0 (Laplacian). The scale parameter $\varsigma = 3.0$ is used for all.

where ς determines the scale, and ζ determines the shape (see Fig. **5.2**). Ω is the normalization coefficient, and $c(\zeta)$ is given by

$$c(\zeta) = \left(\frac{\Gamma\left(\frac{3}{2}(1+\zeta)\right)}{\Gamma\left(\frac{1}{2}(1+\zeta)\right)} \right)^{\frac{1}{1+\zeta}}, \ \Gamma(\cdot): \text{ Gamma function.} \tag{5.4}$$

As Fig. **5.2** illustrates, a variety of shapes such as disc, Gaussian and Laplacian can be obtained by varying the parameter ζ. This function can be used for accurate PSF modeling. For example, given a camera lens, suppose we have collected PSF samples $\{h_k\}$ for a set of discrete scale values $\{r_k\}$ through a calibration procedure. We can first fit each PSF sample using (5.3) producing the corresponding parameters $\{\zeta_k\}$ and $\{\varsigma_k\}$. Secondly, fit the sample parameters through two polynomial functions $\zeta(r)$ and $\varsigma(r)$. Through these two steps, we may get an accurate PSF model $h(\mathbf{x} \,|\, \zeta(r), \varsigma(r))$ in the continuous scale domain. Given the parametric PSF model and an estimated blur scale map, we can generate the local PSF at each pixel position. Then, based on the spatially varying convolution model given in equation (1.3), the defocus blur could be removed through a spatially varying non-blind deconvolution procedure similar to equation (4.19).

However, such blind deblurring method is only valid for the image regions where there is no occlusion. In the boundary area between two objects in different focal



(a) (b)

Figure 5.3: Foreground defocused image example. (a) Observed image. (b) Zoomed part of (a), where in the background area that is close to the foreground/background boundary we can observe both the sharp background content and the blurry foreground edge that partially occludes the background.

planes, the local PSF model could be more complicated than the axially symmetric defocus model discussed in Chapter 4 [113]. For example, consider the image given in Fig. **2.5** (b) (also illustrated in Fig. **5.3** (a)), where its foreground object is defocused and the background is sharp. In the background area that is close to the boundary we can observe both the sharp background content and the blurry foreground edge that partially occludes the background (see Fig. **5.3**) (b)). In this case, the local PSF is definitely not a simple axially symmetric function.

Suppose the latent all-in-focus image is denoted as f, and let m_F represent a foreground mask, which has the value of 1 in the foreground area, and 0 in the background area (similar to the decision map given in Fig. **2.7** (c)). Then, we can define a foreground image $f_F = f \cdot m_F$, and a background image $f_B = f(1 - m_F)$. Consider the case of Fig. **5.3** where f_B is in focus (whose PSF equals to a δ function)

and f_F is defocused (with the corresponding PSF h_D). Given a background position $\mathbf{x}$ close to the boundary, if only the foreground object is available, the corresponding observation:

$$g[\mathbf{x}] = (f_F \otimes h_D)[\mathbf{x}] + n[\mathbf{x}] \tag{5.5}$$

where $n[\mathbf{x}]$ is the additive noise. On the other hand, if only the background is given, we have

$$g[\mathbf{x}] = (f_B \otimes \delta)[\mathbf{x}] + n[\mathbf{x}] = f_B[\mathbf{x}] + n[\mathbf{x}]. \tag{5.6}$$

If both f_F and f_B are available, as illustrated in Fig. **5.4** (a), due to the occlusion effect the amount of light from the background area is reduced. It can be deduced that the observed pixel value at location $\mathbf{x}$ becomes

$$g[\mathbf{x}] = (f_F \otimes h_D)[\mathbf{x}] + \varrho f_B[\mathbf{x}] + n[\mathbf{x}] \tag{5.7}$$

where ϱ is an occlusion factor within the range $[0, 1]$, and its value is determined by the distance between $\mathbf{x}$ and the boundary. If $\mathbf{x}$ is in the area completely occluded by the foreground, then $\varrho = 0$. If $\mathbf{x}$ is in the background area without any occlusion, then $\varrho = 1$. In other cases $0 < \varrho < 1$.

Actually the above imaging model can still be represented through the general spatially varying convolution model:

$$g[\mathbf{x}] = (f \otimes h)[\mathbf{x}] + n[\mathbf{x}]. \tag{5.8}$$

where it can be deduced that in the boundary region the local PSF is a combination of the local δ scaled by ϱ and an occluded defocus PSF h_D (see Fig. **5.4** (d)). How to deconvolve such boundary region is an interesting problem that merits further study.

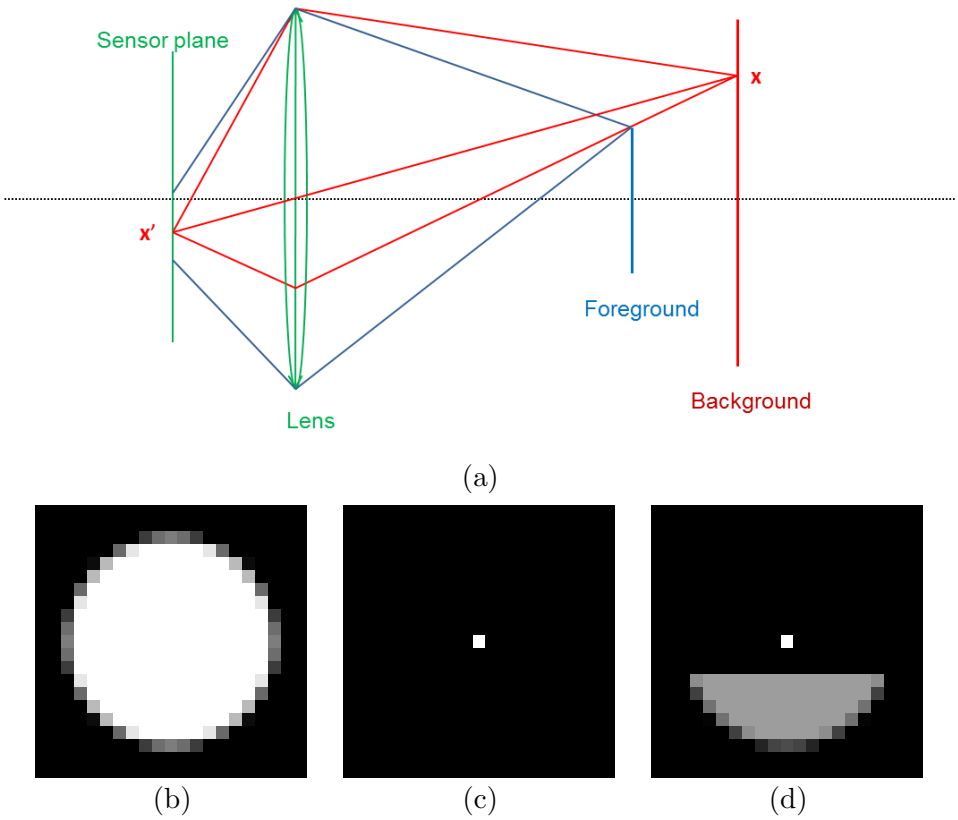


Figure 5.4: Occlusion illustration. (a) Figure illustrating the effect of occlusion in the foreground/background boundary region, where the amount of light from the background position $\mathbf{x}$ is reduced due to the foreground occlusion. Meanwhile, the foreground content is defocused on the sensor plane. (b) Defocus PSF of the foreground. (c) Defocus PSF of the sharp background, which is a δ function. (d) Example of a local PSF in the boundary region $\mathbf{x}$.

Appendix A

Calculation of Probability Density Functions of Local Coherence and Metric Q

If all components of the gradient matrix $\mathbf{G}$ are independent standard Normal variables, the joint probability density function for its singular values s_1 and s_2 is given by [114]:

$$f_{S_1,S_2}(s_1,s_2) = \frac{1}{(N-2)!} e^{-\frac{1}{2}(s_1^2+s_2^2)} s_1^{N-2} s_2^{N-2} (s_1^2 - s_2^2) \tag{A.1}$$

In a more general case, where the variance of a white Gaussian noise image (including N pixels) is σ^2, its $\mathbf{G}$ can be viewed as a $N \times 2$ Gaussian matrix with a variance equals to $\xi\sigma^2$, where ξ is determined by the filters that we choose to calculate the discrete image gradient. The joint density function of the singular values s_1 and s_2

can then be derived from (A.1) as:

$$f_{S_1,S_2}(s_1,s_2) = \frac{\nu^{2N}}{(N-2)!}e^{-\frac{\nu^2}{2}(s_1^2+s_2^2)} \cdot s_1^{N-2}s_2^{N-2}(s_1^2-s_2^2) \tag{A.2}$$

where $\nu = 1/(\sigma\xi^{\frac{1}{2}})$.

The marginal pdf of s_1 is:

$$\begin{aligned} f_{S_1}(s_1) &= \int_0^{s_1} f_{S_1,S_2}(s_1,t)dt \\ &= \frac{1}{2(N-2)!}e^{-\frac{s_1^2\nu^2}{2}}s_1^{2\beta-1}\nu^{2\beta} \\ &\cdot\left[2(s_1\nu)^{2\beta}e^{-\frac{s_1^2\nu^2}{2}} - 2^\beta\left(2\beta - s_1^2\nu^2\right)\gamma\left(\beta,\frac{s_1^2\nu^2}{2}\right)\right] \end{aligned} \tag{A.3}$$

where $\beta = (N-1)/2$, and $\gamma(a,x)$ stands for lower incomplete gamma function, which is defined as:

$$\gamma(\beta,x) = \int_0^x t^{\beta-1}e^{-t}dt \tag{A.4}$$

According to the definition of the sharpness metric H in equation (3.28) and the formula (A.3), the density function of H becomes:

$$\begin{aligned} f_H(h) = \frac{1}{2(N-2)!}e^{-\frac{h^2}{2\mu^2}}\frac{h^{2\beta-1}}{\mu^{2\beta}}\Bigg[2\left(\frac{h}{\mu}\right)^{2\beta}e^{-\frac{h^2}{2\mu^2}} \\ -2^\beta\left(2\beta-\frac{h^2}{\mu^2}\right)\gamma\left(\beta,\frac{h^2}{2\mu^2}\right)\Bigg] \end{aligned} \tag{A.5}$$

where

$$\mu = \frac{\xi^{\frac{1}{2}}}{\sigma+\epsilon/\sigma} \tag{A.6}$$

The ratio between s_1 and s_2 is called the condition number κ:

$$\kappa = \frac{s_1}{s_2} \tag{A.7}$$

whose pdf is given by [114]

$$f_K(\kappa) = 2^{N-1}(N-1)\frac{\kappa^2-1}{(\kappa^2+1)^N}\kappa^{N-2}, \quad 1 \leq \kappa \leq \infty. \tag{A.8}$$

We note that the coherence R is related to κ by $\kappa = \frac{1+R}{1-R}$. Using the formula of random variable transformation [115], the pdf of R can then be derived as:

$$f_R(r) = 4(N-1)r\frac{(1-r^2)^{N-2}}{(1+r^2)^N}, \quad 0 \leq r \leq 1. \tag{A.9}$$

Given the joint density function $f_{S_1,S_2}(s_1, s_2)$ in (A.2), the conditional density of s_2 given s_1 can be derived as:

$$f_{S_2|S_1}(s_2|s_1) = \frac{f_{S_1,S_2}(s_1, s_2)}{f_{S_1}(s_1)} \tag{A.10}$$

According to the definition (3.30), the relationship between s_2 and Q can also be written as:

$$s_2 = s_1\frac{s_1 - Q}{s_1 + Q} \tag{A.11}$$

Again, using the random variable transformation, the conditional density function of Q given s_1 becomes:

$$\begin{aligned} f_{Q|S_1}(q|s_1) &= f_{S_2|S_1}(s_2|s_1)\left|\frac{ds_2}{dq}\right| \\ &= f_{S_2|S_1}\left(s_1\frac{s_1-q}{s_1+q}\middle| s_1\right)\frac{2s_1^2}{(s_1+q)^2} \end{aligned} \tag{A.12}$$

Then the joint density function of s_1 and Q can be written as:

$$\begin{aligned} f_{Q,S_1}(q, s_1) &= f_{S_1}(s_1)f_{Q|S_1}(q|s_1) \\ &= f_{S_1,S_2}\left(s_1, s_1\frac{s_1-q}{s_1+q}\right)\frac{2s_1^2}{(s_1+q)^2} \end{aligned} \tag{A.13}$$

So the pdf of Q becomes:

$$\begin{aligned} f_Q(q) &= \int_q^\infty f_{Q,S1}(q,t)dt \\ &= \frac{8q\nu^{2N}}{(N-2)!}\int_q^\infty t^{2N+1}\frac{(t-q)^{N-2}}{(t+q)^{N+2}}e^{-t^2\nu^2\frac{(t^2+q^2)}{(t+q)^2}}dt \end{aligned} \tag{A.14}$$

Bibliography

[1] L. Yuan, J. Sun, L. Quan, and H. Shum, "Image deblurring with blurred/noisy image pairs," *ACM Transactions on Graphics (SIGGRAPH)*, 2007. vi, 6

[2] R. Redondo, F. Šroubek, S. Fischer, and G. Cristóbal, "Multifocus image fusion using the log-gabor transform and a multisize windows technique," *Inf. Fusion*, vol. 10, no. 2, pp. 163–171, 2009. vii, 10, 27, 28, 33, 36

[3] O. Oreifej, G. Shu, T. Pace, and M. Shah, "A two-stage reconstruction approach for seeing through water clearly," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011. viii, ix, 65, 66, 67

[4] R. Ferzli and L. J. Karam, "A no-reference objective image sharpness metric based on the notion of just noticeable blur (JNB)," *IEEE Trans. Image Processing*, vol. 18, pp. 717–728, 2009. ix, x, 13, 19, 20, 21, 87, 88, 95, 96, 97, 99, 104, 111

[5] S. Bae and F. Durand, "Defocus magnification," in *Computer Graphics Forum*, vol. 26, no. 3. Wiley Online Library, 2007, pp. 571–579. xii, 119, 120, 139, 144

[6] J. Cai, H. Ji, C. Liu, and Z. Shen, "Blind motion deblurring from a single image using sparse approximation," *CVPR*, 2009. xii, 150

[7] R. Fergus, B. Singh, A. Hertsmann, S. T. Roweis, and W. T. Freeman, "Removing camera shake from a single image," *ACM Transactions on Graphics (SIGGRAPH)*, 2006. 2, 12, 54, 117, 126

[8] Q. Shan, J. Jia, and A. Agarwala, "High-quality motion deblurring from a single image," *ACM Transactions on Graphics (SIGGRAPH)*, 2008. 2, 12, 40, 54, 55, 117, 150

[9] L. Xu and J. Jia, "Two-phase kernel estimation for robust motion deblurring," in *Computer Vision–ECCV 2010*. Springer, 2010, pp. 157–170. 2, 117

[10] O. Whyte, J. Sivic, A. Zisserman, and J. Ponce, "Non-uniform deblurring for shaken images," *International journal of computer vision*, vol. 98, no. 2, pp. 168–186, 2012. 3, 117

[11] S. Harmeling, M. Hirsch, and B. Schölkopf, "Space-variant single-image blind deconvolution for removing camera shake," *Advances in Neural Inform. Processing Syst*, vol. 6, 2010. 3, 12

[12] M. Born and E. Wolf, *Principles of optics: electromagnetic theory of propagation, interference and diffraction of light*. CUP Archive, 1999. 3

[13] P. Favaro and S. Soatto, "A geometric approach to shape from defocus," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 27, pp. 406–417, 2005. 4

[14] P. Favaro, "Seeing beyond occlusions (and other marvels of a finite lens aperture)," in *In IEEE CVPR*, 2003, pp. 579–586. 4

[15] W. E. K. Middleton, *Vision Through the Atmosphere.* Canada: University of Toronto Press, 1958. 4, 12, 37

[16] M. C. Roggemann and B. M. Welsh, "Imaging through turbulence," *CRC Press, Boca Raton, Fla.*, 1996. 4, 37

[17] D. Li, R. M. Mersereau, and S. Simske, "Atmospheric turbulence-degraded image restoration using principal components analysis," *IEEE Geoscience and Remote Sensing Letters*, vol. 4, no. 3, pp. 340–344, July 2007. 4, 37

[18] M. Shimizu, S. Yoshimura, M. Tanaka, and M. Okutomi, "Super-resolution from image sequence under influence of hot-air optical turbulence," *CVPR 2008*, June 2008. 4, 12, 37, 38, 43, 45

[19] R. N. Tubbs, "Lucky exposures: Diffraction limited astronomical imaging through the atmosphere," *Ph. D. Thesis, Cambridge University*, September 2003. 4, 12, 23, 37, 38, 39, 40

[20] Y. Tian and S. Narasimhan, "Seeing through water: Image restoration using model-based tracking," *IEEE International Conference on Computer Vision (ICCV)*, 2009. 4, 65

[21] M. A. Vorontsov and G. W. Carhart, "Anisoplanatic imaging through turbulent media: image recovery by local information fusion from a set of short-exposure images," *J. Opt. Soc. Am. A*, vol. 18, no. 6, pp. 1312–1324, June 2001. 4, 23, 37, 39

[22] M. C. Roggemann, C. A. Stoudt, and B. M. Welsh, "Image-spectrum signal-to-

noise-ratio improvements by statistical frame selection for adaptive-optics imaging through atmospheric turbulence," *Optical Engineering*, vol. 33, no. 10, pp. 3254–3264, October 1994. 4, 20, 23, 39

[23] D. L. Fried, "Probability of getting a lucky short-exposure image through turbulence," *Optical Society of America, Journal*, vol. 68, pp. 1651–1658, 1978. 4, 23, 39

[24] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," *Proceeding of the 1998 IEEE International Conference of Compute Vision, Bombay, India*, pp. 836–846, January 1998. 4

[25] S. Morillas, V. Gregori, and A. Sapena, "Fuzzy bilateral filtering for color images," in *Image analysis and recognition*, ser. Lecture Notes in Computer Science, vol. 4141. Springer-Verlag, 2006, pp. 138–145. 4

[26] W.-C. Kao and Y.-J. Chen, "Multistage bilateral noise filtering and edge detection for color image enhancement," *IEEE Transactions on Consumer Electronics*, vol. 51, no. 4, pp. 1346–1351, November 2005. 4

[27] P. Chatterjee and P. Milanfar, "Generalization of non-local mean via kernel regression," accepted in SPIE conference on Computational Imaging, 2008. 4

[28] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-D transform-domain collaborative filtering," *IEEE Transactions on Image Processing*, vol. 16, no. 8, pp. 2080–2095, August 2007. 4, 101

[29] A. Nosratinia, "Postprocessing of jpeg-2000 images to remove compression artifacts," *Signal Processing Letters, IEEE*, vol. 10, no. 10, pp. 296–299, 2003. 5

[30] Z. Wang, H. R. Sheikh, and A. C. Bovik, "No-reference perceptual quality assessment of jpeg compressed images," in *Proceedings of IEEE 2002 International Conferencing on Image Processing*, 2002, pp. 477–480. 5

[31] S. Harmeling, M. Hirsch, S. Sra, and B. Schölkopf, "Online blind deconvolution for astronomical imaging," *IEEE International Conference on Computational Photography (ICCP)*, April 2009. 5, 38, 39

[32] W. Li, J. Zhang, and Q. Dai, "Exploring aligned complementary image prior for blind motion deblurring," *CVPR*, 2011. 6

[33] G. B. Giannakis and R. W. Heath, "Blind identification of multichannel FIR blurs and perfect image restoration," *IEEE Transactions on Image Processing*, vol. 9, pp. 1877–1896, Nov. 2000. 6

[34] G. Harikumar and Y. Bresler, "Perfect blind restoration of images blurred by multiple filters: Theory and efficient algorithms," *IEEE Transactions on Image Processing*, vol. 8, no. 2, pp. 202–219, July 1999. 6

[35] F. Sroubek and P. Milanfar, "Robust multichannel blind deconvolution via fast alternating minimization," *Accepted for IEEE Transactions on Image Processing, 10.1109/TIP.2011.2175740.* 6

[36] M. Hirsch, S. Sra, B. Schölkopf, and S. Harmeling, "Efficient filter flow for space-

variant multiframe blind deconvolution," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 607–614, June 2010. 7, 37, 38, 55, 57, 59

[37] A. Levin, R. Fergus, F. Durand, and W. T. Freeman, "Image and depth from a conventional camera with a coded aperture," *ACM Transactions on Graphics (SIGGRAPH)*, 2007. 8, 12, 54, 55, 118

[38] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman, "Understanding and evaluating blind deconvolution algorithms," *CVPR*, 2009. 8, 40, 48, 117

[39] F. Aguet, D. Van De Ville, and M. Unser, "Model-based 2.5-d deconvolution for extended depth of field in brightfield microscopy," *IEEE Transactions on Image Processing*, vol. 17, no. 7, pp. 1144–1153, July 2009. 10

[40] M. Antunes, M. Trachtenberg, G. Thomas, and T. Shoa, "All-in-focus imaging using a series of images on different focal planes," *Lecture Notes in Computer Science*, vol. 3656, pp. 174–181, 2005. 10, 27

[41] A. N. Rajagopalan, S. Chaudhuri, and U. Mudenagudi, "Depth estimation and image restoration using defocused stereo pairs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 11, pp. 1521–1525, Nov. 2004. 11, 118

[42] S. Erasmus and K. Smith, "An automatic focusing and astigmatism correction system for the sem and ctem," *Journal of Microscopy*, vol. 127, no. 2, pp. 185–199, 1982. 13, 20

[43] J. C. Christou, K. J. Mighell, and R. B. Makidon, "Strehl ratio and image sharp-

ness for adaptive optics," *Advances in Adaptive Optics II. Proceedings of the SPIE*, vol. 6272, p. 62721Y, 2006. 13, 17, 20

[44] P. Marziliano, F. Dufaux, S. Winkler, and T. Ebrahimi, "A no-reference perceptual blur metric," in *Proceedings of the International Conference on Image Processing*, vol. 3, Rochester, NY, 2002, pp. 57–60. 13, 19, 21, 99

[45] S. Varadarajan and L. J. Karam, "An improved perception-based no-reference objective image sharpness metric using iterative edge refinement," in *Proceedings of the 15th IEEE International Conference on Image Processing*, October 2008, pp. 401–404. 13, 19, 21

[46] S. Gabarda and G. Cristóbal, "Blind image quality assessment through anisotropy," *J. Opt. Soc. Am. A*, vol. 24, no. 12, pp. B42–B51, 2007. 13, 99

[47] R. Ferzli and L. J. Karam, "A no-reference objective sharpness metric using Riemannian tensor," in *Third International Workshop on Video Processing and Quality Metrics for Consumer Electronics*, Scottsdale, Arizona, January 2007, pp. 25–26. 13, 70, 99

[48] C. F. Batten, "Autofocusing and astigmatism correction in the scanning electron microscope," in *M.Phil. Thesis, University of Cambridge*, August 2000. 13, 21

[49] X. Zhu and P. Milanfar, "Removing atmospheric turbulence via space-invariant deconvolution," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 35, no. 1, pp. 157 – 170, 2013. 14

[50] ——, "Stabilizing and deblurring atmospheric turbulence," *IEEE International Conference on Computational Photography (ICCP)*, April 2011. 14

[51] ——, "Image reconstruction from videos distorted by atmospheric turbulence," in *SPIE Electronic Imaging, Conference 7543 on Visual Information Processing and Communication*, San Jose, CA, January 2010. 14, 37, 38, 43, 47, 55, 57, 59, 60

[52] ——, "Automatic parameter selection for denoising algorithms using a no-reference measure of image content," *Image Processing, IEEE Transactions on*, vol. 19, no. 12, pp. 3116–3132, 2010. 14

[53] ——, "A no-reference sharpness metric sensitive to blur and noise," in *1st International Workshop on Quality of Multimedia Experience (QoMEX)*, San Diego, CA, July 2009. 14

[54] ——, "A no-reference image content metric and its application to denoising," in *Image Processing (ICIP), 2010 17th IEEE International Conference on.* IEEE, 2010, pp. 1145–1148. 14

[55] ——, "Qpro: An improved no-reference image content metric using locally adapted SVD," *International Workshop on Video Processing and Quality Metrics for Consumer Electronics (VPQM)*, Jan. 2013. 14

[56] X. Zhu, S. Cohen, S. Schiller, and P. Milanfar, "Estimating spatially varying defocus blur from a single image," *Submitted to IEEE Transactions on Image Processing.* 15

[57] J. C. Christou, E. K. Hege, D. Freeman, and E. Ribak, "Self-calibrating shift-

and-add technique for speckle imaging," *Journal of the Optical Society of America A-Optics Image Science and Vision*, vol. 3, no. 2, pp. 204–209, February 1986. 17

[58] L. C. Roberts Jr, M. D. Perrin, F. Marchis, A. Sivaramakrishnan, R. B. Makidon, J. C. Christou, B. A. Macintosh, L. A. Poyneer, M. A. van Dam, and M. Troy, "Is that really your strehl ratio?" in *Astronomical Telescopes and Instrumentation*. International Society for Optics and Photonics, 2004, pp. 504–515. 17, 18

[59] S. Sanyal and A. Ghosh, "The factor of encircled energy from the optical transfer function," *Journal of Optics A: Pure and Applied Optics*, vol. 4, no. 2, p. 208, 2002. 19

[60] J. Caviedes and F. Oberti, "A new sharpness metric based on local kurtosis, edge and energy information," *Signal Processing: Image Communication*, vol. 19, pp. 147–161, 2004. 20, 21, 99

[61] N. Zhang, A. Vladar, M. Postek, and B. Larrabee, "A kurtosis-based statistitcal measure for two-dimensional processes and its application to image sharpness," *Proceedings of Section of Physical and Engineering Sciences of American Statistical Society*, pp. 4730–4736, 2003. 20, 21

[62] L. Firestone, K. Cook, K. Culp, N. Talsania, and K. Preston, "Comparison of autofocus methods for automated microscopy," *Cytometry*, vol. 12, no. 3, pp. 195–206, 1991. 20, 21

[63] X. Marichal, W. Y. Ma, and H. J. Zhang, "Blur determination in the compressed domain using DCT information," in *Proceedings of the International Conference on Image Processing*, vol. 2, Kobe, Japan, October 1999, pp. 386–390. 20, 21

[64] R. F. Dantowitz, S. W. Teare, and M. J. Kozubal, "Ground-based high-resolution imaging of Mercury," *The Astronomical Journal*, vol. 119, pp. 2455–2457, May 2000. 20

[65] K.-C. Yang, C. C. Guest, and P. Das, "Perceptual sharpness metric (psm) for compressed video," in *Multimedia and Expo, 2006 IEEE International Conference on.* IEEE, 2006, pp. 777–780. 22

[66] G. H. Golub and C. van Loan, *Matrix Computations*, 3rd ed. Johns Hopkins University Press, 1996. 25

[67] G. Piella, "A general framework for multiresolution image fusion: from pixels to regions," *Information fusion*, vol. 4, no. 4, pp. 259–280, 2003. 27

[68] H. Li, B. Manjunath, and S. K. Mitra, "Multisensor image fusion using the wavelet transform," *Graphical models and image processing*, vol. 57, no. 3, pp. 235–245, 1995. 28

[69] A. Levin, D. Lischinski, and Y. Weiss, "A closed form solution to natural image matting," *CVPR*, 2006. 31, 32

[70] K. He, J. Sun, and X. Tang, "Single image haze removal using dark channel prior," *CVPR*, 2009. 31

[71] R. K. Tyson, *Principles of Adaptive Optics.* San Diego, USA: Academic Press, 1998. 37, 38

[72] M. A. Vorontsov, "Parallel image processing based on an evolution equation with

anisotropic gain: integrated optoelectronic architectures," *J. Opt. Soc. Am. A*, vol. 16, pp. 1623–1637, 1999. 37

[73] N. M. Law, "Lucky imaging: Diffraction-limited astronomy from the ground in the visible," *Ph. D. Thesis, Cambridge University*, May 2003. 37, 40

[74] M. Aubailly, M. A. Vorontsov, G. W. Carhart, and M. T. Valley, "Image enhancement by local information fusion with pre-processing and composed metric," *Proceedings of SPIE*, vol. 7090, 2008. 39

[75] S. John and M. A. Vorontsov, "Multiframe selective information fusion from robust error estimation theory," *IEEE Transactions on Image Processing*, vol. 14, no. 5, pp. 577–584, May 2005. 39

[76] M. Aubailly, M. A. Vorontsov, G. W. Carhat, and M. T. Valley, "Automated video enhancement from a stream of atmospherically-distorted images: the lucky-region fusion approach," *Proceedings of SPIE*, vol. 7463, 2009. 39, 55, 57, 59

[77] N. Joshi and M. Cohen, "Seeing Mt. Rainier: Lucky imaging for multi-image denoising, sharpening, and haze removal," *IEEE ICCP 2010*, March 2010. 39, 40

[78] R. Szeliski and J. Coughlan, "Spline-based image registration," *International Journal of Computer Vision*, vol. 22, no. 93, pp. 199–218, 1997. 43, 45

[79] S. Farsiu, M. Elad, and P. Milanfar, "Constrained, globally optimal, multi-frame motion estimation," *IEEE/SP 13th Workshop on Statistical Signal Processing*, pp. 1396–1401, July 2005. 45

[80] M. F. Beg and A. Khan, "Symmetric data attachment terms for large deformation

image registration," *IEEE Transactions on Medical Imaging*, vol. 26, no. 9, pp. 1179–1189, 2007. 45

[81] L. Davies and U. Gather, "The identification of multiple outliers," *Journal of the American Statistical Association*, vol. 88, pp. 782–792, Sep. 1993. 50

[82] F. R. Hampel, "The influence curve and its role in robust estimation," *Journal of the American Statistical Association*, vol. 69, pp. 383–393, 1974. 52, 53, 104

[83] M. P. Wand and M. C. Jones, *Kernel Smoothing*, ser. Monographs on Statistics and Applied Probability. London; New York: Chapman and Hall, 1995. 52, 53

[84] H. Takeda, S. Farsiu, and P. Milanfar, "Kernel regression for image processing and reconstruction," *IEEE Transactions on Image Processing*, vol. 16, no. 2, pp. 349–366, February 2007. 52

[85] A. Buades, B. Coll, and J. M. Morel, "A review of image denoising algorithms, with a new one," *Multiscale Modeling and Simulation (SIAM interdisciplinary journal)*, vol. 4, no. 2, pp. 490–530, 2005. 53

[86] X. Feng and P. Milanfar, "Multiscale principal components analysis for image local orientation estimation," in *Proceedings of the 36th Asilomar Conference on Signals, Systems and Computers*, vol. 1, Pacific Grove, CA, November 2002, pp. 478–482. 71, 79

[87] J. Bigun, G. H. Granlund, and J. Wiklund, "Multidimensional orientation estimation with applications to texture analysis and optical flow," *IEEE Transactions*

on Pattern Analysis and Machine Intelligence, vol. 13, no. 8, pp. 775–790, August 1991. 71, 79

[88] M. Lebrun, M. Colom, A. Buades, and J. Morel, "Secrets of image denoising cuisine," *Acta Numerica*, vol. 21, no. 1, pp. 475–576, 2012. 78

[89] H. Takeda, S. Farsiu, and P. Milanfar, "Kernel regression for image processing and reconstruction," *IEEE Transactions on Image Processing*, vol. 16, no. 2, pp. 349–366, February 2007. 78, 101

[90] N. Ponomarenko, V. Lukin, A. Zelensky, K. Egiazarian, M. Carli, and F. Battisti, "TID2008 - a database for evaluation of full-reference visual quality assessment metrics," *Advances of Modern Radioelectronics*, vol. 10, pp. 30–45, 2009. 94

[91] H. R. Sheikh, M. F. Sabir, and A. C. Bovik, "A statistical evaluation of recent full reference image quality assessment algorithms," *IEEE Trans. Image Processing*, vol. 15, pp. 3440–3451, 2006. 95

[92] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, April 2004. 98

[93] G. H. Golub, M. Heath, and G. Wahba, "Generalized cross-validation as a method for choosing a good ridge parameter," *Technometrics*, vol. 21, no. 2, pp. 215–223, May 1979. 98, 99

[94] D. A. Girard, "The fast Monte-Carlo cross-validation and c_l procedures: com-

ments, new results and application to image recovery problems," *Comput. Statist.*, vol. 10, pp. 205–231, 1995. 98

[95] P. C. Hansen, "Analysis of discrete ill-posed problems by means of the L-curve," *SIAM Rev.*, vol. 34, no. 4, pp. 561–580, 1992. 98

[96] P. C. Hansen and D. P. O'Leary, "The use of the L-curve in the regularization of discrete ill-posed problems," *SIAM J. Sci. Comput.*, vol. 14, no. 6, pp. 1487–1503, 1993. 98, 99

[97] C. Stein, "Estimation of the mean of a multivariate Normal distribution," *Ann. Statist.*, vol. 9, pp. 1135–1151, 1981. 98

[98] S. Ramani, T. Blu, and M. Unser, "Monte-Carlo SURE: A black-box optimization of regularization parameters for general denoising algorithms," *IEEE Transactions on Image Processing*, vol. 17, no. 9, pp. 1540–1554, September 2008. 98, 100, 102, 104

[99] Y. Qu, Z. Pu, H. Zhao, and Y. Zhao, "Comparison of different quality assessment functions in autoregulative illumination intensity algorithms," *Optical Engineering*, vol. 45, pp. 117–201, 2006. 99

[100] E. N. Kirsanova and M. G. Sadovsky, "Entropy approach in the analysis of anisotropy of digital images," *Open Systems and Information Dynamics*, vol. 9, pp. 239–250, 2002. 99

[101] A. Rényi, "Some fundamental questions of information theory," *Selected Papers of Alfréd Rényi*, vol. 3, pp. 526–552, 1976. 99

[102] A. Chakrabarti, T. Zickler, and W. T. Freeman, "Analyzing spatially-varying blur," *CVPR*, 2010. 119, 120, 122, 129

[103] D. Zoran and Y. Weiss, "Scale invariance and noise in natural images," *IEEE International Conference on Computer Vision*, Sep. 2009. 123

[104] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE International Conference on Computer Vision*, 1999. 134

[105] M. Potmesil and I. Chakravarty, "Synthetic image generation with a lens and aperture camera model," *ACM Transactions on Graphics*, vol. 1, no. 2, April. 138, 151

[106] O. Oreifej, X. Li, and M. Shah, "Simultaneous video stabilization and moving object detection in turbulence," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 35, no. 2, pp. 450–462, 2013. 149

[107] M. S. C. Almeida and M. A. T. Figueiredo, "Parameter estimation for blind and non-blind deblurring using residual whiteness measures," *Image Processing, IEEE Transactions on*, vol. 22, no. 7, July 2013. 149, 150

[108] S. J. Wright, R. D. Nowak, and M. A. Figueiredo, "Sparse reconstruction by separable approximation," *Signal Processing, IEEE Transactions on*, vol. 57, no. 7, pp. 2479–2493, 2009. 149

[109] M. V. Afonso, J. M. Bioucas-Dias, and M. A. Figueiredo, "Fast image recovery using variable splitting and constrained optimization," *Image Processing, IEEE Transactions on*, vol. 19, no. 9, pp. 2345–2356, 2010. 149

[110] R. C. Gonzalez and R. E. Woods, *Digital Image Processing.* Upper Saddle River, N.J.: Prentice Hall, 2002. 150

[111] E. Kee, S. Paris, S. Chen, and J. Wang, "Modeling and removing spatially-varying optical blur," in *Computational Photography (ICCP), 2011 IEEE International Conference on.* IEEE, 2011, pp. 1–8. 151

[112] G. E. Box and G. C. Tiao, *Bayesian inference in statistical analysis.* Wiley-Interscience, 2011. 151

[113] S. S. Bhasin and S. Chaudhuri, "Depth from defocus in presence of partial self occlusion," *ICCV*, 2001. 153

[114] A. Edelman, "Eigenvalues and condition numbers of random matrices," *SIAM Journal on Matrix Analysis and Aplications*, vol. 9, pp. 543–560, 1988. 156, 158

[115] P. G. Hoel, S. C. Port, and C. J. Stone, *Introduction to Probability Theory.* Houghton Mifflin Harcourt, 1971. 158