Single-Shot Image Deblurring with Modified Camera Optics
カメラ光学系の加工による単一画像からのボケの除去

by

Yosuke Bando
坂東 洋介

A Doctoral Dissertation
博士論文

Submitted to
The Graduate School of Information Science and Technology
The University of Tokyo
on December 15, 2009
in Partial Fulfillment of the Requirements
for the Degree of Doctor of Information Science and Technology
in Computer Science

Thesis Supervisor: Tomoyuki Nishita 西田 友是
Title: Professor of Computer Science

Abstract

The recent rapid popularization of digital cameras allows people to capture a large number of digital photographs easily, and this situation makes automatic avoidance and correction of "failure" photographs important. While exposure and color issues have been mostly resolved by the improvement in automatic corrective functions of cameras, defocus, motion, and camera shake blur can be handled only in a limited fashion by current cameras. Camera shake blur can be alleviated by an anti-camera shake mechanism installed in most cameras; but for focus, although a particular scene depth can be focused with an auto-focus function, objects at different depths cannot be captured sharply at the same time. Moreover, defocused images can often result due to the failure of auto-focusing. In addition, blur caused by object motion, i.e., motion blur, is only avoided by increasing the shutter speed and sensor sensitivity when a camera detects motion in a scene.

This dissertation proposes a method for removing defocus and motion blur for digital cameras. Since deblurring is generally an ill-posed problem, and hence an image processing approach alone has limitations, the proposed method includes modifications of camera optics. In this regard, this dissertation pursues low cost and compact implementation, aiming at applications to consumer products. That is, small modifications to existing cameras or mechanisms that can be directly derived from existing ones will be adopted.

In order to set a baseline performance achievable without modifying camera optics, this dissertation first proposes an image deblurring method that is purely based on an image processing approach, which consists of fast image deconvolution for efficient deblurring, and local blur estimation for handling spatially-varying blur. Additionally, a set of intuitive user interfaces are provided with which the user can interactively change the focus settings of photographs after they are captured, so that she/he can not only obtain an all-in-focus image but also create images focused to different depths.

For removal of defocus blur, a method is proposed for estimating the defocus blur size in each image region by placing red, green, and blue color filters in a camera lens aperture. As captured image will have depth-dependent color misalignment, the scene depth can be estimated by solving a stereo correspondence problem between images recorded with different wavelengths. Since the depth is directly related to the defocus blur size, deblurred images can be produced by deconvolving each region with the estimated blur size. The modification requires only inexpensive color filters.

For motion blur removal, this dissertation proposes to move the camera image sensor circularly about the optical axis during exposure, so that the attenuation of high frequency image content due to motion blur can be prevented, facilitating deconvolution.

The frequency domain analysis of the circular sensor motion trajectory in space-time shows that the degradation of image quality is equally reduced for all objects moving in arbitrary directions with constant velocities up to some predetermined maximum speed. The proposed method may be implemented using an existing sensor-shift system of an anti-camera shake mechanism.

要旨

近年のデジタルカメラの普及により、誰もが容易かつ大量にデジタル写真を撮影することができるようになったことを受けて、「失敗」写真の自動回避・修正が重要な課題になっている。露出や色味の問題はカメラの自動補正機能の充実により解決されてきている一方で、焦点ボケ、被写体の動きによるブレ、手ブレを含むボケは、現在のカメラでは対策に限界がある。手ブレについては手ブレ補正機構のカメラ搭載が標準的になり問題は軽減されたが、焦点ボケについては、オートフォーカス機能を用いてある奥行きに焦点を合わせることはできても、異なる奥行きを持つ物体を同時に鮮明に捉えることはできない。また、オートフォーカスが失敗してピンボケ写真が撮れることも多い。さらに、被写体の動きによるブレすなわちモーションブラーに対するデジタルカメラの対策は、動きを検知してシャッター速度と感度を上げるという回避策にとどまっている。

本論文はデジタルカメラのための焦点ボケとモーションブラーの除去法を提案する。ボケの除去は一般に不良設定問題であり、画像処理だけでの解決には限界があるため、提案法はカメラ光学系の加工を伴う。ただし、民生品への応用を視野に、低コストかつ小規模な実装を目指す。すなわち、既存カメラに対する軽微な変更か、既存のカメラ機構の直接的な応用にとどめるものとする。

カメラ光学系の加工をせずに達成可能な基準性能を定めるために、本論文はまず純粋に画像処理のみに基づく画像のボケ除去法を提案する。これにはボケ除去を効率的に行うための高速な逆畳み込み法と、画像中で一様でないボケを扱うための局所ボケ推定法が含まれる。加えて、撮影後に写真の焦点設定を対話的に変更することのできる一連の直観的なユーザインタフェースを提供する。これによりユーザは全焦点画像だけでなく、異なる奥行きに焦点を合わせた画像を作成することができる。

焦点ボケの除去については、カメラのレンズ開口部に赤緑青の色フィルタを挿入することにより画像の領域ごとの焦点ボケの量を推定する方法を提案する。撮影される画像にはシーンの奥行きに応じた色ズレが生じるため、異なる波長で記録された画像間のステレオマッチングとして奥行きを求めることができる。奥行きは直接ボケ量に対応しており、推定値に基づいて各領域を逆畳み込みすることでボケを除去した画像を生成することができる。加工は色フィルタの追加だけでよいため安価に実現可能である。

モーションブラーの除去については、本論文は露光中にカメラの撮像センサを光軸に垂直に円運動させることにより、モーションブラーによる画像の高周波成分の減衰を低減し、

逆畳み込みによるボケの除去を容易化する手法を提案する。センサの円運動の時空間中の軌跡を周波数解析することで、ある既定の上限速度までは、任意の方向に等速運動する全ての物体の劣化を同程度に低減できることが示せる。本手法は手ブレ補正機構の１つである撮像センサシフト方式の応用により実現できると考えられる。

Acknowledgments

I would like to express my deepest gratitude to Professor Tomoyuki Nishita, who has been my research and thesis supervisor since I was undergraduate. For over 10 years, he has eagerly taught me how to conduct research for my bachelor, master, and this doctoral theses, and also for my research at TOSHIBA Corporation. I owe every part of my career to his valuable advices founded on his honorable academic background and achievements.

I am also deeply grateful to Associate Professor Bing-Yu Chen for his useful discussions on this work and for providing me with necessary information and environment. Major papers of mine were coauthored by him.

I would also like to thank my thesis committee: Professor Katsushi Ikeuchi, Professor Mikio Shinya, Associate Professor Takeshi Naemura, Associate Professor Shigeo Takahashi, and Associate Professor Takeo Igarashi. They provided me with critical comments and questions that enabled me to strengthen the final content of this dissertation. Besides, Associate Professor Takeshi Naemura and Associate Professor Shigeo Takahashi have given me helpful suggestions since the beginning of this work.

Moreover, I would like to thank TOSHIBA members including Mitsuo Saito, Kenichi Mori, and Atsushi Kunimatsu for providing me with the opportunity and support for this research; Kenji Shirakawa and Hisashi Kazama for their helpful advices; and Shingo Yanagawa, Mayumi Yuasa, and Tomokazu Wakasugi for their help.

I would also like to acknowledge important suggestions from Dr. Yusuke Iguchi, Dr. Yasutaka Furukawa, Dr. Tsuneya Kurihara, Dr. Takashi Imagire, Dr. Yoshihiro Kanamori, Yonghao Yue, and Takuya Saito.

In addition, I would like to thank all the present and past members of Nishita Laboratory for their discussions and encouragement. In particular, I would like to express my appreciation to Paulo Silva, Pablo Garcia Trigo, Zoltan Szego, Johanna Wolf, Napaporn Metaaphanon, and Yusuke Tsuda for their help.

Furthermore, I would like to thank Saori Horiuchi for her dedicated help and support.

Finally, I would like to thank my friends and family for their endless encouragement and support.

Contents

Chapter 1

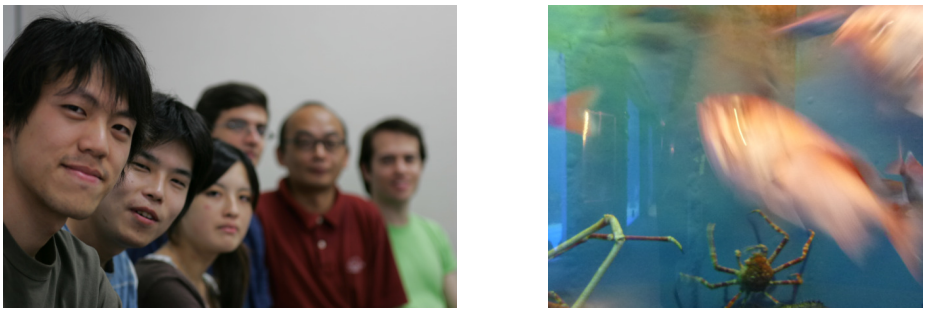
Introduction

This chapter first describes the motivation for this work, then the background and preliminary information on image deblurring, followed by a review of the related work. Finally, the contributions and organization of this dissertation are presented.

1.1 Motivation

The recent rapid popularization of digital cameras allows people to capture a large number of digital photographs easily. As the number of casual photographers increases, so does the number of "failure" photographs including over/under-exposed, noisy, blurred, and unnaturally-colored images. This situation makes automatic avoidance and correction of failure photographs important. In fact, automatic corrective functions of digital cameras including auto-exposure, automatic white balance, and noise reduction capabilities steadily improve to resolve exposure, color, and noise issues.

On the other hand, current digital cameras appear to handle image blur only in a limited fashion; they only directly address camera shake blur, but not defocus and motion blur. For camera shake blur, most of the recent cameras are equipped with an anti-camera shake mechanism that moves either the lens or the image sensor to compensate for camera motion obtained from an accelerometer. For defocus blur, however, although a particular scene depth can be focused with an auto-focus function, objects at different depths cannot be captured sharply at the same time (*depth-of-field* effects, see Fig. 1.1(a)). Moreover, defocused images can be commonly seen in personal photo collections due to the failure of auto-focusing. In addition, blur caused by object motion, i.e., motion blur (see Fig. 1.1(b)), can only be avoided by increasing the shutter speed and sensor sensitivity when a camera detects motions in a scene, at the expense of an increased noise level.



(a) (b)

Figure 1.1: Examples of blurred photographs. (a) Photograph with a shallow depth-of-field, in which only the faces at the focused depth are sharply captured, and the others are subject to defocus blur. (b) Photograph containing motion-blurred fish.

To overcome the above-described situation, this dissertation proposes methods for removing defocus and motion blurs in photographs. Since deblurring is generally an ill-posed problem as will be explained in Sec 1.2, image processing techniques alone can often suffer from noise amplification and ringing artifacts in deblurred images, which result from attenuation of high frequency image content at capture time and also from misidentification of blur kernels during image processing. Therefore, in addition to image processing techniques for deblurring, the proposed method includes modifications of camera optics that alter the image capture process of traditional cameras in order to achieve high frequency preservation and to facilitate blur kernel identification. In this regard, this dissertation pursues low cost and compact hardware implementation, aiming at applications to consumer digital cameras. That is, small modifications to existing cameras or mechanisms that can be directly derived from existing ones will be adopted.

This dissertation focuses on a single-shot approach. That is, we try to recover an unseen sharp image given a single blurred image, and do not resort to taking multiple photographs. Although one could benefit from an increased amount of information from multiple images, images must be registered in some way, and dynamic scenes and/or hand-held image capture without a tripod can introduce additional sources of errors. Of course, one could use multiple synchronized cameras to alleviate this issue, but that is not only expensive but also an unrealistic usage scenario for casual photographers. Another option might be to use a high-speed camera to minimize motion between frames to facilitate registration, but each frame will have an increased noise level due to reduced exposure time, and the memory bandwidth required to transfer image data from the sensor to the storage device will become large, making the obtainable image resolution small. Moreover, we would like to note that a single-shot approach and a multi-shot approach can complement each other; multi-shot approaches could benefit from improved deblurring results of the proposed single-shot methods, and vice versa.

Prior to proposing camera hardware-assisted deblurring methods, we would like to set a baseline performance achievable without modifying camera optics. To this end, we first explore an image deblurring method that is purely based on an image processing approach. After that, we propose defocus and motion deblurring methods with modified camera optics.

1.2 Image Deblurring

Image deblurring can be formulated as the process of inverting image blurring. This section first introduces a model of image degradation, and then presents problem definitions of image deblurring with their basic solution strategies and associated difficulties.

1.2.1 Image Degradation Model

Image degradation due to blur can be locally modeled as convolution [43]:

$$g(x,y) = h(x,y) * f(x,y) + n(x,y), \tag{1.1}$$

where g is an observed degraded image, h is a blur kernel or a *point-spread function* (PSF), f is an unknown (*latent*) original sharp image, n is a noise term, and $*$ denotes a convolution operator. A defocus blur kernel can be often modeled as a pillbox function (see Fig. 1.2(b) top):

$$h(x,y) = \begin{cases} 1/\pi r^2 & \text{for } \sqrt{x^2+y^2} \leq r \\ 0 & \text{otherwise} \end{cases}, \tag{1.2}$$

where r is the radius of a circle of confusion. A motion blur kernel for a horizontally moving object can be modeled as a box function (see Fig. 1.2(b) bottom):

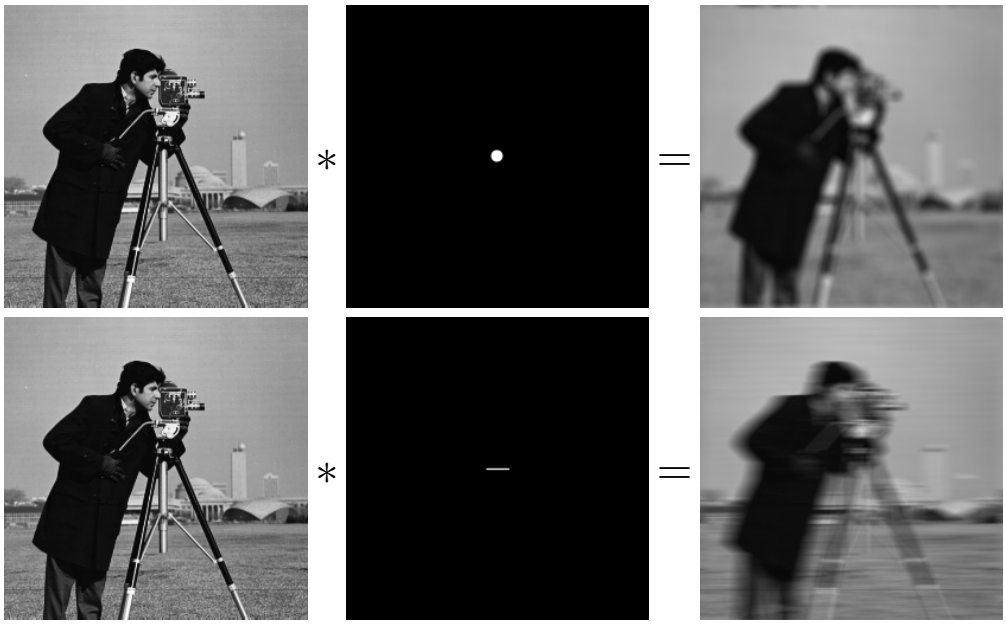
$$h(x,y) = \begin{cases} \delta(y)/2L & \text{for } |x| \leq L \\ 0 & \text{otherwise} \end{cases}, \tag{1.3}$$

where $2L$ is the length by which the object travels during exposure, and $\delta(\cdot)$ is a Dirac delta function.

Fig. 1.2 shows examples of this image degradation process, where the blur is assumed to be uniform for the entire image.

1.2.2 Image Deconvolution

Image deconvolution is a problem of estimating the latent image f given an observation g and a blur kernel h. This is known to be already an ill-posed problem, because high



(a) (b) (c)

Figure 1.2: Image degradation process (the noise term is omitted). (a) Original sharp image f. (b) Blur kernel h. The top row corresponds to defocus blur Eq. 1.2, and the bottom row to motion blur Eq. 1.3. (c) Blurred image g.

frequency information in the original image f is attenuated by blur. This is apparent if we consider Eq. 1.1 in the frequency domain:

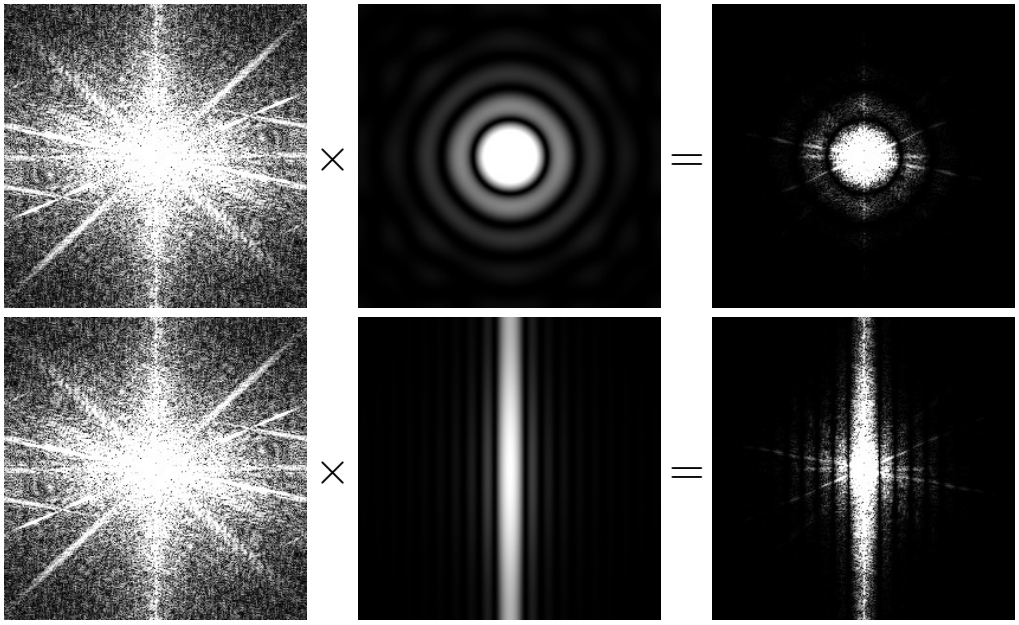
$$\hat{g}(\omega_x, \omega_y) = \hat{h}(\omega_x, \omega_y)\hat{f}(\omega_x, \omega_y) + \hat{n}(\omega_x, \omega_y), \tag{1.4}$$

where the *hat* notation $\hat{f}$ denotes the Fourier transform of f, and (ω_x, ω_y) denotes spatial frequencies. Note that convolution in the spatial domain becomes multiplication in the frequency domain. Fig. 1.3 shows frequency domain representation of Fig. 1.2. As can be seen, the high frequency content (corresponding to the values in the outer region of each frequency domain image in Fig. 1.3) of the original image is significantly attenuated after multiplication by the blur kernel, because the blur kernel decays rapidly for higher frequencies. In addition, the shown blur kernels have periodic zeros that make the corresponding frequency content completely lost. Therefore, image deconvolution is a process of recovering weakened or lost signals and essentially involves signal amplification. This is apparent in the following equation that implements naive deconvolution, known as *pseudo-inverse deconvolution*.

$$\hat{f}'(\omega_x, \omega_y) = \frac{\hat{h}^*(\omega_x, \omega_y)}{|\hat{h}(\omega_x, \omega_y)|^2 + \varepsilon}\ \hat{g}(\omega_x, \omega_y), \tag{1.5}$$

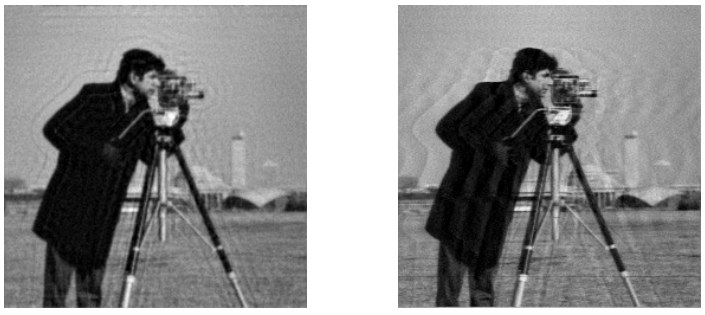
where $\hat{f}'$ denotes a deblurred image (in the frequency domain), $\hat{h}^*$ the complex conjugate of $\hat{h}$, and ε is a small number to avoid zero division. The fraction of the right-hand side of

the equation is essentially division by $\hat{h}$, meaning that weakened signals will be amplified back accordingly. As a result, naive deconvolution also amplifies noise in a blurred image, which leads to ringing artifacts as shown in Fig. 1.4.



(a) (b) (c)

Figure 1.3: Image degradation process viewed in the frequency domain. (a) Log magnitude of the Fourier transform $\hat{f}$ of an original sharp image f. (b) Log magnitude of the Fourier transform $\hat{h}$ of a blur kernel h. (c) Log magnitude of the Fourier transform $\hat{g}$ of a blurred image g.



(a) (b)

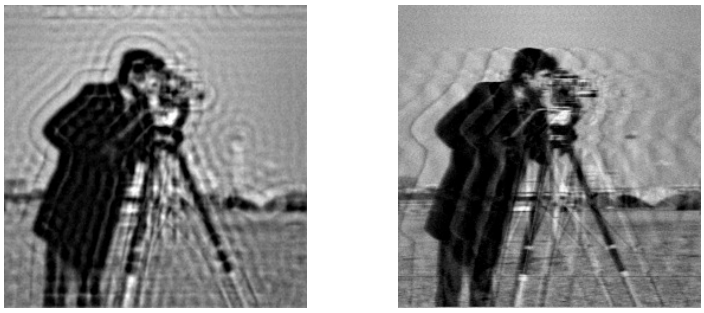
Figure 1.4: Results of pseudo-inverse deconvolution for the blurred images shown in Fig. 1.2. (a) Result for the defocus blurred image. (b) Result for the motion blurred image.

1.2.3 Blind Image Deconvolution

Blind image deconvolution is a problem of estimating the latent image f given only an observation g [43]. As we must also estimate a blur kernel h from the blurred image g, the

problem becomes under-constrained since the number of unknowns (the number of pixels in f plus the number of values in h) exceeds the number of observations (the number of pixels in g, which is equal to that of f). The problem is solved either by first estimating the blur kernel h and then applying (non-blind) image deconvolution to obtain f, or by iteratively improving the estimates of h and f until convergence.

Frequency attenuation can even more adversely affect recovered image quality than for non-blind deconvolution, since errors in the estimated kernel can excessively amplify the frequency content that is not significantly attenuated by the true blur kernel. Fig. 1.5 shows examples of pseudo-inverse deconvolution results with blur kernels having slightly different sizes from the true ones.



(a) (b)

Figure 1.5: Results of pseudo-inverse deconvolution with wrong blur kernels for the blurred images shown in Fig. 1.2. (a) Result for the defocus blurred image. (b) Result for the motion blurred image.

1.2.4 Target Problem

This dissertation addresses an image deblurring problem in which each local image region can be treated as being subject to the blind image deconvolution problem. That is, blur kernels can vary across the image region but can be assumed to be piecewise uniform. While we assume a parametric form of blur as in Eqs. 1.2 and 1.3, blur kernels are *not* assumed to be *a priori* known (i.e., blind).

1.3 Previous Work

Researches on image deblurring and restoration have a long history [14, 43, 11], and they have mainly focused on an image processing approach. We first review some of the traditional methods and also explain recent advances in this field in Sec. 1.3.1.

While image processing approaches handle images in a post-capture manner, recent years have seen an emergence of techniques called *computational photography* [76] that change the image capture process of cameras to acquire scene information unobservable with traditional cameras, and/or to facilitate post-capture image processing. The methods involving camera optics modifications proposed in this dissertation belong to this field of research. Sec. 1.3.2 introduces computational photography methods for image deblurring, and also some work that is not directly related to deblurring but is relevant to this dissertation.

Finally, the relation of the previous work to ours is summarized in Sec. 1.3.3.

1.3.1 Image Processing Approach

Image Deconvolution

Solving an image deconvolution problem all comes down to coping with its ill-posedness described in Sec.1.2.2, which, in one aspect, manifests itself as zero division in the frequency domain. While the simplest remedy is to add a fixed small number to the denominator as in Eq. 1.5, a more theoretically meaningful way is to add $|\hat{n}(\omega_x, \omega_y)|^2/|\hat{f}(\omega_x, \omega_y)|^2$ to the denominator, known as Wiener deconvolution [14]. Of course the power spectra of the noise $|\hat{n}|^2$ and original image $|\hat{f}|^2$ are not known, estimated values are used instead.

Apart from such relatively simple *regularization* (i.e., to make ill-posed problems well-posed ones), one can use some prior knowledge of natural images. The difficulty inherent in ill-posed problems is that the space of possible solutions is large; a blurring process can produce similar images from different images by losing the signals that make the original images differ from each other. Therefore, if we enforce some *a priori* properties on deblurred images, the solution space can be confined to a smaller one. A simple but effective piece of prior knowledge of natural images is that their pixel values are bounded; they cannot take negative or unlimitedly large values. Richardson-Lucy deconvolution iteratively updates the estimated deblurred image such that its pixel values are always kept positive [79, 58].

Another form of such regularization is to minimize $|c(x,y) * f'(x,y)|^2$, which is known as Tikhonov regularization [88]. A regularizing operator c is often set as a high-pass filter to rule out images with a large high frequency content, so that ringing-free images are favored. However, it may also rule out sharp images by its nature. Minimizing a squared norm of high-pass filtered images can be viewed as enforcing Gaussian smoothness priors to the filtered signal distributions, but recent researches on natural image statistics

show that they obey so-called *heavy-tailed* distributions [29], which have a narrower peak and a broader foot than Gaussians (also known as *sparseness priors*). Therefore, recent methods exploit heavy-tailed priors to allow occasional discontinuities (such as edges) in restored images [30, 68, 10, 15]. These methods use *discrete wavelet transform* (DWT) as band-pass filters, but since restored images suffer from blocky artifacts arising from the dyadic image partitioning in DWT, they use *translation-invariant DWT* (TI-DWT) [24], also known as *stationary* DWT, to reduce such artifacts at the cost of significant increase in computational complexity.

Blind Image Deconvolution

Blind image deconvolution techniques restore the original sharp image from an observed degraded image without precise knowledge of a point-spread function (PSF) [43]. There are two main approaches to this: 1) first estimate the PSF, and then apply a non-blind deconvolution method with that PSF; 2) iteratively estimate the PSF and the original sharp image.

For the approach that estimates the PSF first, some traditional methods payed attention to the frequency zero patterns in a blur kernel [20]. For example, the Fourier transform of a box function as shown in Eq. 1.3 is given as $\hat{h}(\omega_x, \omega_y) = \mathrm{sinc}(L\omega_x)$, meaning that it has periodic zeros at $\omega_x = k\pi/L$ for a non-zero integer k. From Eq. 1.4, we can expect that the Fourier transform $\hat{g}$ of the observed image has the same zero pattern if we can ignore noise. However, such methods are not practical in the presence of noise. Another approach is to take a set of candidate PSFs, and to choose the one that best explains the observed image. The selection criteria differ from method to method, such as residual spectral matching [80] and generalized cross validation [78].

For the approach that iteratively estimates the PSF and the sharp image, Ayers and Dainty proposed to iterate the process of updating the PSF from the estimated sharp image in the Fourier domain, imposing image space constraints on the PSF (non-negativity, for example), updating the sharp image from the PSF in the Fourier domain, and imposing constraints on the sharp image [9]. More recent methods took a conceptually similar approach and estimated a camera shake PSF from a single image by incorporating natural image statistics. Fergus *et al*. imposed a sparseness prior for image derivative distributions, and used an ensemble learning approach to solve the otherwise intractable optimization problem [28]. Shan *et al*. introduced a more sophisticated noise model and a local smoothness prior [81].

Some researchers used multiple images. Rav-Acha and Peleg [77] showed that using

two motion blurred images can produce better deconvolution results; Yuan *et al*. [101] used a long-exposure blurred image and a short-exposure noisy image, so that PSF estimation can benefit from the short-exposure image which is not subject to blur.

Handling Spatially-Variant Blur

The methods described above all assume a PSF to be spatially-invariant (uniform). A spatially-variant PSF is usually estimated by sectioning the image and by assuming it to be approximately spatially-invariant within each section [89, 72, 44]. This means that the blur is assumed to be only slowly varying across the image, as each section should be large enough to make reliable estimation. This is also true for non-blind spatially-variant deconvolution methods [66, 46, 96].

A few methods exist that can estimate a spatially-variant PSF with abrupt changes across the image. Levin identified spatially-variant motion blur by examining the difference between the image derivative distribution along the motion direction and that along its perpendicular direction for the case of 1D linear motion [47]. You and Kaveh [100] also addressed the problem of removing spatially-variant motion blur, but only a synthetic horizontal motion blur example was presented.

Depth-from-focus/defocus techniques generate a depth map of a scene by estimating the amount of defocus blurs in images. Hence they can be viewed as spatially-variant PSF estimation methods. Existing methods either use multiple images [73, 99, 67], or make an estimate at edges in a single image by assuming that a blurred ramp edge is originally a sharp step edge [73, 85, 45].

Depth-of-field Synthesis

Some methods are designed to create and alter the depth-of-field effects from images, rather than to remove blurs from them. Burt and Kolczynski fused multiple images with different focus to generate images with an extended depth-of-field [19]. Kubota and Aizawa used two images, and generated arbitrarily focused images by assuming that a scene consisted of two depth layers, each of which was in focus in either image [42]. The "Lens Blur" filter of Adobe Photoshop CS [2] creates depth-of-field effects from a single sharp photograph with a user-provided depth map.

Matting

In image editing, matting is an important technique for extracting foreground objects in an image so that they can be composited over other images [22]. In addition to image editing purposes, matting is also important in the context of image deblurring, for separating image regions so that each region can be deconvolved independently.

The traditional approach to matting is to use a blue or green screen as a background [91, 82]. Extracting a matte from a single *natural* image (i.e., an image with general unknown background colors) requires user intervention, a typical form of which is a *trimap* that segments an image into "strictly foreground," "strictly background," and "unknown" regions. Fractional alpha values are computed in the "unknown" region based on the information from the other two regions [23, 86, 92, 49, 50, 93]. To automate matting, previous approaches used multiple images. Smith and Blinn [82] captured images of a foreground object with two different known background colors. Alternatively, Wexler *et al*. [97] used a sequence of images of a translating/rotating object. Xiong and Jia [98] captured images from two viewpoints, and computed their stereo correspondences taking into account alpha values of a foreground object. Several methods used synchronized cameras to capture multiple images of an object [60, 61, 39, 59, 104].

1.3.2 Hardware-Assisted Approach

Defocus Blur Removal

For defocus blur removal, a *wavefront coding* method [27] incorporates a cubic phase plate in the imaging system, so that the defocus blur is independent of the scene depth. The defocus blur can be removed with a single known PSF, but this technique requires a custom optical system that can be expensive. Several researchers have introduced *coded aperture* techniques [48, 90] which places a patterned mask in the camera lens aperture to change the frequency characteristics of defocus blur in order to facilitate blur estimation and removal. These methods offer portable imaging systems with minimal modifications to the conventional camera, but as the blur estimation solely relies on defocus cues, some manual intervention may be required, and there is ambiguity between depths farther and nearer than the focused depth.

Several camera designs have been proposed to estimate scene depth using defocus cues, which can also be viewed as defocus PSF estimation methods. Hiura and Matsuyama used a modified camera with multiple pinholes [36] to capture images in which duplicated scene textures are displaced and superimposed in a depth-dependent manner.

Two research groups used color filters in the lens aperture to capture the displaced images in separate color channels [6, 21]. Moreno-Noguer *et al*. used a projector and estimated depth from projector defocus [64].

If multi-view images are available, not only scene depth can be estimated, but also synthesis of depth-of-field effects, or *refocusing*, can be performed. There are several camera designs to do multi-view image capture through a single main lens. Adelson and Wang [1] showed that light rays entering a camera can be captured separately depending on their incident angle by placing a microlens array on the image sensor, and they estimated depth from multi-view images captured through the main lens. Ng *et al*. [70] realized this idea in a hand-held camera, and proposed a post-exposure refocusing method by noting that the captured multi-view images correspond to the light field inside the camera [69]. Multi-view images can also be captured by placing an attenuation mask on the image sensor [90], or by splitting light rays at the aperture [35, 55, 54] or outside the main lens [31].

Motion Blur Removal

For motion blur removal, Raskar *et al*. [75] developed a *coded exposure* technique to prevent attenuation of high frequencies due to motion blur at capture time by opening and closing the shutter during exposure according to a pseudo-random binary code. Agrawal and Xu [4] presented another type of code that enables PSF estimation in addition to high frequency preservation. Levin *et al*. [51] proposed to move the camera image sensor with a constant 1D acceleration during exposure, and showed that this sensor motion can render motion blur *invariant* to 1D linear object motion (e.g., horizontal motion), and that it evenly distributes the fixed frequency "budget" to different object speeds. That is, objects moving at different speeds can be deblurred equally well.

Some researchers proposed to move sensors for different purposes. Ben-Ezra *et al*. [13] moved the sensor by a fraction of a pixel size between exposures for video super-resolution. Mohan *et al*. [63] moved the lens and sensor to deliberately introduce motion blur that acts like defocus blur. Nagahara *et al*. [65] moved the sensor along the optical axis to make defocus blur depth-invariant.

Camera Shake Removal

For camera shake removal, Ben-Ezra and Nayar [12] attached a low resolution video camera to a main camera, and estimated a camera shake PSF from video frames to remove blur

from a main camera image. Recent cameras typically have an anti-camera shake mechanism that moves either the lens or the image sensor to compensate for camera motion obtained from an accelerometer.

1.3.3 Relation to the Proposed Method

Fig. 1.6 shows four stages in a generic processing flow of image deblurring. We first capture an image, and then segment the image into regions each of which can be assumed to have a uniform blur. After that, for each local region, we estimate the blur kernel and finally use it to deconvolve the image. Some methods may perform segmentation and blur estimation simultaneously. Some may iterate blur estimation and deconvolution.

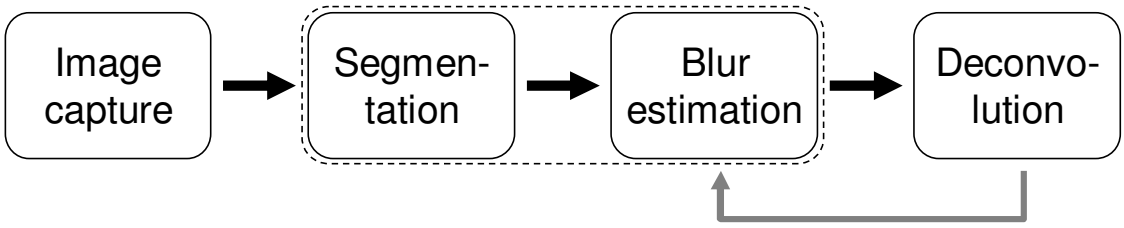


Figure 1.6: Processing flow of image deblurring.

Table 1.1 summarizes the relationship between the proposed method and some of the previous work for three of the above four stages and for the three blur types, namely defocus, motion, and camera shake blur. We set aside the image capture stage because it is trivial for methods purely based on an image processing approach, and for methods involving optics modifications, the (modified) image capture stage can facilitate one, two, or all of the succeeding three stages depending on the methods. Therefore, the table has two rows for each blur type, one for methods involving optics modifications, and the other for pure image processing methods.

Image Processing Approach to Image Deblurring

Chapter 2 proposes a method for defocus blur removal purely based on an image processing approach. For image segmentation and blur estimation, we propose a method that can handle abrupt blur changes across images, while previous methods such as [72] assume slowly varying blur. For deconvolution, we speed up the computation involving heavy-tailed priors by building upon one of the state-of-the-art methods called *WaveGSM* [15] and make it 10 times faster. The deconvolution method can be used for removing motion and camera shake blur as well. Additionally, we provide users with a set of intuitive interfaces with which the user can interactively change the focus settings of photographs

Table 1.1: Summary of the relationship between the proposed method and some of the previous work. Only a few representative methods that use single images are cited for brevity.

		Segmentation	Blur estimation	Deconvolution
Defocus blur	Modified optics	Wavefront coding [27] / Coded aperture [48, 90]		Chapter 3
	Image processing	Ozkan et al. 1991 [72]		WaveGSM [15] *
Motion blur	Modified optics		Motion-invariant photography [51] / Coded exposure photography [75, 4]	
	Image processing	Levin 2006 [47]	Chapter 4	(common to the above field *)
Camera shake blur	Modified optics	Not required	Ben-Ezra and Nayar 2004 [12]	Image processing alone will suffice
	Image processing		Fergus et al. 2006 [28]	(common to the above field *)

Chapter 2

after they are captured, so that she/he can not only obtain an all-in-focus image but also create images focused to different depths. To our knowledge, techniques that synthesize refocused images from a single conventional photograph have not been reported in the literature.

Defocus Blur Removal using a Color-Filtered Aperture

As will be shown in Chapter 2, image processing alone does not necessarily produce satisfactory results, and we propose to modify camera optics. In Chapter 3, we present a method for simultaneously performing segmentation and defocus blur estimation by placing red, green, and blue color filters in a camera lens aperture. Although wavefront coding [27] can cover all the latter three stages for image deblurring, it requires special lenses that can be expensive, whereas the modification of the proposed method requires only inexpensive color filters. The coded aperture methods [48, 90] also cover the three stages, but some issues remain for the segmentation and blur estimation stages as described in Sec. 1.3.2. As deconvolution quality can be considerably improved by the coded aperture, this dissertation focuses on facilitating the segmentation and blur estimation stages, and we use a color-filtered aperture to exploit parallax cues rather than to directly use defocus cues, which addresses the above-mentioned issues.

The idea of using color filters in the aperture itself has been proposed previously. For a stereo correspondence measure between the color planes, Amari and Adelson [6] used a squared intensity difference with high-pass filtering. As they discussed in their paper, however, this measure was insufficient to compensate for intensity differences between the color planes. Their prototype was not portable, and only a single result for a textured planar surface was shown. Chang *et al*. [21] normalized the intensities within a local window in each color plane before taking the sum of absolute differences between them. But as their camera was equipped with a flashbulb for projecting a speckle pattern onto the scene in order to generate strong edges in all the color planes, the performance of their correspondence measure in the absence of flash was not shown. They also had to capture another image without flash to obtain a "normal" image. We propose a better correspondence measure between the color planes.

As compared to the existing camera designs for single-lens multi-view image capture, our method splits light rays at the aperture similarly to [35, 55, 54], but uses only color filters as additional optical elements to the lens without requiring multiple exposures. Although this comes with a price of a reduced number of views (only three) each having only a single color plane, we can still obtain useful information for defocus deblurring and post-exposure image editing.

As for matting, our method can automatically extract alpha mattes with a single hand-held camera in a single exposure, and to the best of our knowledge, such capability has not been reported previously.

Motion Blur Removal using Circular Sensor Motion

While a method for segmenting and identifying 1D motion blur (e.g., horizontal motions) in a single image is reported in the literature [47], it still seems difficult to handle general 2D (i.e., in-plane) motions in a pure image processing framework. Chapter 4 proposes to move the camera image sensor circularly about the optical axis during exposure, so that the attenuation of high frequency image content due to motion blur can be prevented, facilitating deconvolution. This is an extension of motion-invariant photography [51] so that it can handle 2D linear object motion, although that leaves the segmentation stage an open problem.

The most closely related work to the proposed approach includes coded exposure photography [75, 4] and motion-invariant photography [51]. Table 1.2 summarizes qualitative comparisons among these methods and ours. Refer also to [3] for detailed comparison between the coded exposure and motion-invariant strategies.

The motion-invariant strategy best preserves high frequencies for target object motion range, but it does not generalize to motion directions other than the one it assumes. The coded exposure strategy can handle any direction, and its performance only gradually decreases for faster object motion. Our circular motion strategy can treat any direction and speed up to some assumed limit, and it achieves better high frequency preservation for target object speed than the coded exposure strategy in terms of deconvolution noise. Similar to the motion-invariant strategy, the circular motion strategy degrades static scene parts due to sensor motion, but it can partially track moving objects so that they are recognizable even before deconvolution. Unlike the other strategies, the circular motion strategy has no 180° motion ambiguity in PSF estimation; it can distinguish rightward object motion from leftward one.

Table 1.2: Summary of the trade-offs among various image capture strategies for motion deblurring.

	Static camera	Coded exp. [75, 4]	Motion-inv. [51]	Circular (ours)
High frequency preservation*	Bad	Good	Good	Good
Direction generalization	–	Yes	No	Yes
Speed generalization	–	Yes	Yes†	Yes†
Static scene parts unblurred	Yes	Yes	No	No
Image recognizable w/o deconv.	No	No	Yes†	Yes†
180° motion discrimination	No	No‡	No	Yes

∗ For target object motion, Motion-invariant > Circular > Coded exposure.
† Within some assumed motion range. ‡ As in [4].

1.4 Contributions

The major contributions of this dissertation are threefold.

1. **Image Processing Approach to Image Deblurring** (Chapter 2): In order to set a baseline performance achievable without modifying camera optics, this dissertation first proposes a set of methods for image deblurring that is purely based on an image processing approach.

   We propose a method for speeding up deconvolution computation for efficient deblurring. Deconvolution quality is known to be improved by taking into account derivative distributions of natural images. While existing methods take time to repeat derivative and convolution operations, the proposed method achieves similar image quality with 1/10 computation time by taking derivatives beforehand and by working in the gradient domain.

   We also present a method for estimating defocus blur that can handle spatially-variant blur having abrupt changes across the image. We propose to use color image segmentation rather than the traditional rectangular segmentation in order to divide the image into regions each having a similar defocus blur size. A criterion is elaborated to choose the best blur size from a set of candidates for each region, and means are also provided to correct the estimated blur size in a user-assisted manner.

   Moreover, besides producing all-in-focus images as the result of defocus removal, we provide users with means to interactively control the focus settings of photographs after they are captured, since defocus blur can serve to enhance artistic impression of photographs.

2. **Defocus Blur Removal using a Color-Filtered Aperture** (Chapter 3): We propose a method for estimating defocus blur sizes in each image region by placing red, green, and blue color filters in a camera lens aperture in order to facilitate defocus blur removal. The camera modification requires only inexpensive color filters.

   As a captured image will have depth-dependent color misalignment, the scene depth can be estimated by solving a stereo correspondence problem between images recorded with different wavelengths. We devise a novel stereo correspondence measure that can robustly identify disparities between the RGB color channels. The disparities are directly related to the defocus blur sizes, and deblurred images can be produced by deconvolving each region with the estimated blur size.

   We also present a matting method for extracting an in-focus foreground object so that the unblurred part of the scene can remain unaffected by the deconvolution

process. Color misalignment cues introduced by the filters serve to constrain the space of possible mattes that would otherwise contain erroneous mattes when foreground and background colors are similar. We propose a novel matting algorithm exploiting the color misalignment cues to obtain better mattes than ones that can be produced by the previous matting methods.

3. **Motion Blur Removal using Circular Sensor Motion** (Chapter 4): We propose to move the camera image sensor circularly about the optical axis during exposure, so that the attenuation of high frequency image content due to motion blur can be prevented, facilitating motion blur removal. That is, although no object may be photographed sharply at capture time, differently moving objects can be deconvolved with similar quality. The proposed method may be implemented using a sensor-shift system of an anti-camera shake mechanism.

   We analyze the frequency characteristics of circular sensor motion in relation to linear object motion, and show that this sensor motion nearly evenly distributes the fixed frequency "budget" to different object speeds, meaning that the degradation of image quality is equally reduced for all objects moving in arbitrary directions with constant velocities up to some predetermined speed.

   We also present a method for estimating motion blur based on the fact that, for a set of PSFs resulting from circular sensor motion, deconvolution by wrong PSFs causes ringing artifacts, which is not always the case for other image capture strategies. This allows us to take a simple hypothesis testing approach for PSF estimation, and we propose to detect such ringing based on image sparseness priors.

1.5 Organization

As listed in Sec. 1.4, Chapter 2 describes an image deblurring method that is purely based on an image processing approach, and Chapters 3 and 4 present methods that exploit camera optics modifications for defocus blur and motion blur removal. Finally, conclusion and future work are presented in Chapter 5.

Chapter 2

Image Processing Approach to Image Deblurring

This chapter explores a method for removing blur in images that is purely based on an image processing approach. Although the goal of this dissertation is to facilitate the deblurring process by modifying the capture process of traditional cameras, we would first like to know the achievable performance without modifications of camera optics. To this end, this chapter deals with defocus deblurring in the context of *digital refocusing*, in which images are not only deblurred, but also *refocused* as if they were focused to different depths.

2.1 Introduction

Digital refocusing, a technique that generates photographs focused to different depths (distances from a camera) after a single camera shot as shown in Fig. 2.1, is attracting the attention of the computer graphics community and others in view of its interesting and useful effects. The technique is originally based on the light field rendering [52, 32], and exploits the fact that a photograph is a 2D integral projection of a 4D light field [69], as was simulated by Isaksen *et al*. [37]. Ng *et al*. made this technique practical with their hand-held plenoptic camera [70], eliminating the need for large and often expensive apparatus such as a camera array or a moving camera that was traditionally required to capture light fields. Since then, other novel camera designs have been emerging in order to improve the resolution of images and/or to reduce the cost of optical equipment attached to a camera [31, 48, 54].

In an attempt to perform digital refocusing without modifying camera optics, in this chapter we are interested in developing an image processing method for synthesizing

(a) (b) (c)

Figure 2.1: From a single input photograph, images focused to different depths can be obtained. (a) A single input photograph, focused on the person in the left. (b) Created image, refocused on the person in the middle. (c) Created image, refocused on the person in the right.

refocused images from a single photograph taken with a conventional camera. If we had a sharp, all-in-focus photograph with a depth map of the scene, it would be straightforward to create depth-of-field effects by blurring the input photograph according to the depth, as some of the existing image-editing software do (e.g., the *Lens Blur* filter of Adobe Photoshop CS [2]). Therefore, we must first estimate "a sharp image with a depth map" from an input photograph. In other words, we must first estimate and remove defocus blur in a photograph.

To achieve this goal, we assume that spatially-variant defocus blur in an input photograph can be locally approximated by a uniform blur, and we restore a sharp image by stitching multiple deconvolved versions of an input photograph. And we also propose a local blur estimation method applicable to irregularly-shaped image segments in order to handle abrupt blur changes at depth discontinuities due to object boundaries. To create desired refocusing effects, we present several means of determining the amount of blur to be added to a restored sharp image based on the estimated blur, by which users can change focus and depth-of-field interactively and intuitively.

2.2 Image Processing Flow

Fig. 2.2 shows a block diagram of our method. From an input defocused photograph $g(x,y)$, we first restore a *latent image* $l(x,y)$, which would have been observed if defocus blur had not been introduced by the camera lens system (i.e., deblurred sharp image). We use the standard pillbox PSF parameterized by radius r of the circle of confusion, referred to as *blur radius*, as a defocus blur model [11]:

$$h(x,y;r) = \begin{cases} 1/\pi r^2 & \text{for } \sqrt{x^2+y^2} \leq r \\ 0 & \text{otherwise} \end{cases}, \tag{2.1}$$

and we generate multiple differently deblurred images $f_j(x,y)$ by deconvolving an input photograph with each of the predetermined $M+1$ blur radii $\{r_j | j = 0, 1, \cdots, M\}$. That is, we remove uniform defocus blur with blur radius r_j from $g(x,y)$ to obtain $f_j(x,y)$. This amounts to solving the image deconvolution problem

$$g(x,y) = h(x,y;r_j) * f_j(x,y) + n(x,y), \tag{2.2}$$

whose solution is given in Sec. 2.3. The $M+1$ blur radii are arranged in ascending order as $r_0 < r_1 < r_2 < \cdots < r_M$, and $r_0 = 0$ so that $f_0(x,y) \equiv g(x,y)$. We typically use $r_j = 0.5j$, and $r_M = 10.0$ (in pixels).

From the deblurred images $f_j(x,y)$, we locally select the "best" image and stitch them together to obtain the latent image $l(x,y)$, the approach known as *sectional method* [89]. More precisely, we first estimate a blur radius field $r_{org}(x,y)$ which describes with what blur radius the input photograph is originally blurred around each pixel location (x,y), as described in Sec. 2.4. We then linearly blend the deblurred images as

$$l(x,y) = \frac{r_{j+1} - r_{org}(x,y)}{r_{j+1} - r_j} f_j(x,y) + \frac{r_{org}(x,y) - r_j}{r_{j+1} - r_j} f_{j+1}(x,y), \tag{2.3}$$

where j is appropriately chosen for each pixel (x,y) such that $r_j \leq r_{org}(x,y) \leq r_{j+1}$.

Now that we obtained the latent image $l(x,y)$, we create an output refocused image $o(x,y)$ by blurring the latent image. The reason for first obtaining the latent image is that, as convolution of two disc PSFs does not result in another disc PSF, refocused images cannot be obtained by directly convolving/deconvolving an input photograph. Sec. 2.5 presents a method for determining a new blur radius field $r_{new}(x,y)$ to be added to the latent image based on $r_{org}(x,y)$ in order to meet desired refocusing effects. To perform the synthesis in real-time, we again employ the sectional method, and we prepare multiple differently blurred images as $b_j(x,y) = h(x,y;r_j) * l(x,y)$ in the preprocessing stage. Again, $b_0(x,y) \equiv l(x,y)$. In the interactive refocusing stage, we perform linear interpolation similar to Eq. 2.3 for a new blur radius field $r_{new}(x,y)$ and the blurred images $b_j(x,y)$ and $b_{j+1}(x,y)$ as:

$$o(x,y) = \frac{r_{j+1} - r_{new}(x,y)}{r_{j+1} - r_j} b_j(x,y) + \frac{r_{new}(x,y) - r_j}{r_{j+1} - r_j} b_{j+1}(x,y), \tag{2.4}$$

where j satisfies $r_j \leq r_{org}(x,y) \leq r_{j+1}$ for each pixel (x,y).

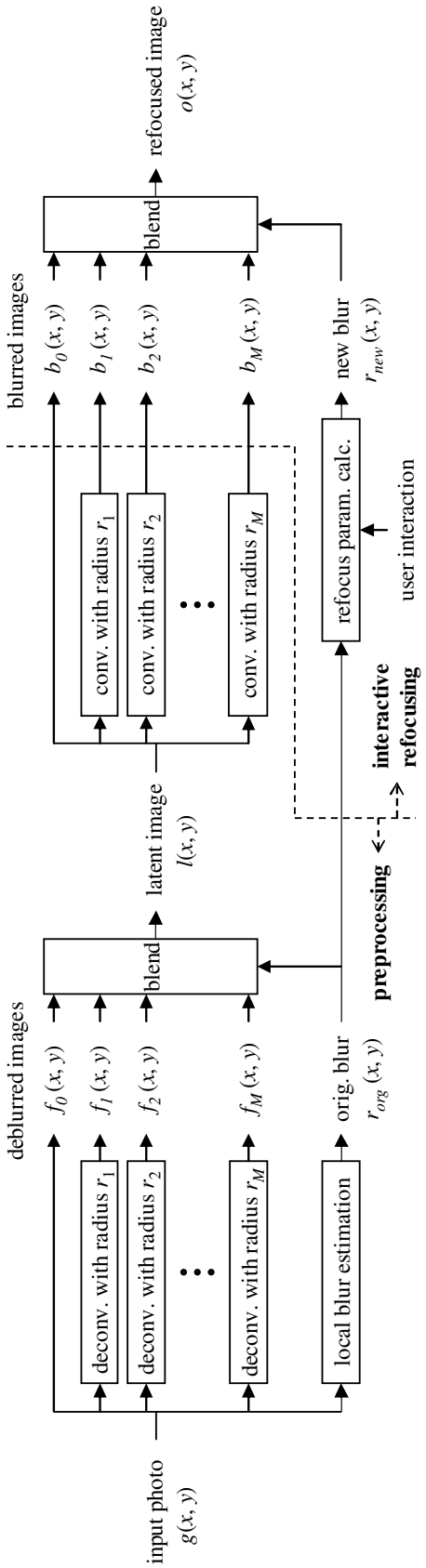


Figure 2.2: Block diagram of our defocus deblurring and digital refocusing process.

2.3 Image Deconvolution

In Sec. 1.2.2 we briefly reviewed the difficulties lying in the image deconvolution problem. Here we take a closer look. For notational convenience, this chapter uses a matrix-vector version of Eq. 2.2 as follows [11].

$$\mathbf{g} = \mathbf{H}\mathbf{f} + \mathbf{n}, \tag{2.5}$$

where $\mathbf{g}$, $\mathbf{f}$, and $\mathbf{n}$ are P-vectors representing $g(x,y)$, $f_j(x,y)$, and $n(x,y)$, respectively, with lexicographic ordering of P discretized pixel locations, and $\mathbf{H}$ is a $P \times P$ matrix representing convolution by a PSF $h(x,y;r_j)$. Dependence on j is omitted for brevity.

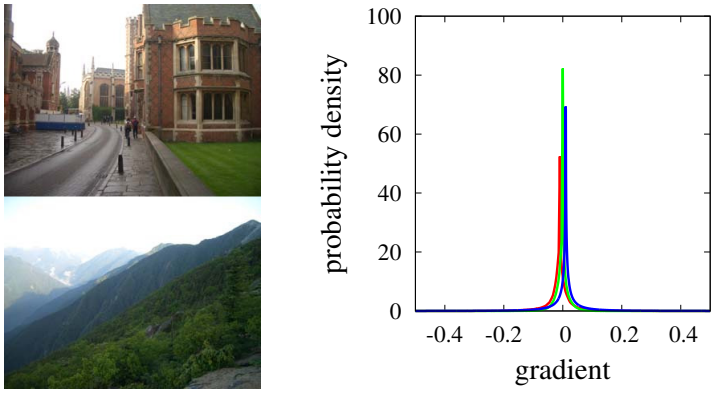
Since solving Eq. 2.5 for $\mathbf{f}$ as a least squares problem of minimizing $\|\mathbf{g} - \mathbf{H}\mathbf{f}\|^2$ is known to be ill-posed due to ill-conditioned matrix $\mathbf{H}$, one needs prior knowledge about which images are more likely to occur in nature. However, frequently-used Gaussian smoothness priors are not suitable for restoring sharp (hence not necessarily smooth) images, even though they are computationally tractable. Therefore, recent methods exploit so-called *heavy-tailed priors*, according to which the distributions of band-pass filter outputs of (sharp) natural images have a narrower peak and a broader foot than Gaussians as shown in Fig. 2.3, allowing occasional discontinuities (such as edges) in restored images [30, 15]. These methods use *discrete wavelet transform* (DWT) as band-pass filters, but since restored images suffer from blocky artifacts arising from the dyadic image partitioning in DWT, they use *translation-invariant DWT* (TI-DWT) [24], also known as *stationary* DWT, to reduce such artifacts at the cost of significant increase in computational complexity.

We avoid this problem by using derivative filters instead of DWT, since they are translation-invariant and do not involve dyadic image partitioning. Specifically, we bring Bioucas-Dias's wavelet domain method (WaveGSM) [15] into the gradient domain, because the *Gaussian scale mixture* (GSM) representation of heavy-tailed priors used in WaveGSM is also applicable to speeding up the non-linear optimization involving heavy-tailed priors in the gradient domain.

Following Tappen *et al*. [87], we use a generalized Laplacian distribution as a heavy-tailed prior model for image gradients:

$$p(\mathbf{f}_x[i]) \propto \exp\left(-|\mathbf{f}_x[i]|^{\alpha}/\beta\right), \tag{2.6}$$

where $\mathbf{f}_x[i]$ denotes the i-th element of the derivative of $\mathbf{f}$ with respect to x, and $p(\cdot)$ denotes a probability density function of an argument variable. We set $\alpha = 0.3$ and $\beta = 0.085$ with pixel values in range [0, 1], so that Eq. 2.6 approximates sample gradient distributions as shown in Fig. 2.3. We use the same prior for y derivatives, $\mathbf{f}_y[i]$.



(a) (b)

Figure 2.3: (a) Sample sharp images. (b) Gradient distributions of the top image (red) and of the bottom image (green), and the generalized Laplacian distribution we use (blue). For visibility, these plots are horizontally displaced. They all actually peak at zero.

Taking derivatives of Eq. 2.5 leads to the following two gradient domain deconvolution equations:

$$\mathbf{g}_x = \mathbf{H}\mathbf{f}_x + \mathbf{n}_x, \qquad \mathbf{g}_y = \mathbf{H}\mathbf{f}_y + \mathbf{n}_y. \tag{2.7}$$

For brevity, we will only deal with the x component in what follows. Assuming that the noise $\mathbf{n}_x$ in the gradient domain can be modeled as a Gaussian with variance w and that the prior is independently applicable to each pixel location i, the posterior distribution of a latent gradient $\mathbf{f}_x$ given an observation $\mathbf{g}_x$ is given as

$$p(\mathbf{f}_x|\mathbf{g}_x) \propto p(\mathbf{g}_x|\mathbf{f}_x)p(\mathbf{f}_x) \propto \exp\left(-\frac{\|\mathbf{g}_x - \mathbf{H}\mathbf{f}_x\|^2}{2w}\right)\prod_{i=1}^{P} p(\mathbf{f}_x[i]). \tag{2.8}$$

The latent gradient is estimated as the maximizer of (the logarithm of) Eq. 2.8 as

$$\mathbf{f}'_x = \arg\max_{\mathbf{f}_x}\left\{-\frac{\|\mathbf{g}_x - \mathbf{H}\mathbf{f}_x\|^2}{2w} + \sum_{i=1}^{P} \ln p(\mathbf{f}_x[i])\right\}, \tag{2.9}$$

leading to non-linear optimization because the prior term is not quadratic: $\ln p(\mathbf{f}_x[i]) = -|\mathbf{f}_x[i]|^\alpha/\beta$ with $\alpha = 0.3$ (see Eq. 2.6).

In order to solve Eq. 2.9 efficiently, we follow the WaveGSM approach, and we represent the heavy-tailed prior as a Gaussian scale mixture (GSM) as

$$p(\mathbf{f}_x[i]) = \int_0^\infty p(\mathbf{f}_x[i]|s)p(s)ds, \tag{2.10}$$

where $p(\mathbf{f}_x[i]|s)$ is a zero-mean Gaussian with scale (or variance) s, weighted by $p(s)$. Regarding s as a "missing variable," Eq. 2.9 is turned into an expectation maximization (EM) iteration as

$$\mathbf{f}_x^{m+1} = \arg\max_{\mathbf{f}_x}\left\{-\frac{\|\mathbf{g}_x - \mathbf{H}\mathbf{f}_x\|^2}{2w} + \sum_{i=1}^{P} E_i{}^m\left[\ln p(\mathbf{f}_x[i]|s)\right]\right\}, \tag{2.11}$$

where m is an iteration count, and $E_i^m[\cdot]$ denotes the expectation with respect to $p(s|\mathbf{f}_x^m[i])$, the probability density of scale s given the current (m-th) estimate $\mathbf{f}_x^m[i]$ of the latent gradient. Since $p(\mathbf{f}_x[i]|s)$ is a Gaussian, the prior term in Eq. 2.11 now becomes

$$E_i^m\left[\ln p(\mathbf{f}_x[i]|s)\right] = E_i^m\left[-\frac{(\mathbf{f}_x[i])^2}{2s}\right] = -\frac{(\mathbf{f}_x[i])^2}{2} E_i^m\left[\frac{1}{s}\right], \tag{2.12}$$

which is quadratic with respect to $\mathbf{f}_x[i]$ since $E_i^m[s^{-1}]$ is fixed during m-th EM iteration (see [15] for more details):

$$E_i^m\left[\frac{1}{s}\right] = \frac{\alpha}{\beta|\mathbf{f}_x^m[i]|^{2-\alpha}}. \tag{2.13}$$

Now that the objective function to be maximized in Eq. 2.11 is quadratic, taking its derivative with respect to $\mathbf{f}_x$ and setting it to zero leads to the following system of linear equations:

$$(\mathbf{H}^T\mathbf{H} + w\mathbf{S}^m)\mathbf{f}_x = \mathbf{H}^T\mathbf{g}_x, \tag{2.14}$$

where $\mathbf{H}^T$ is the transpose of $\mathbf{H}$, $\mathbf{S}^m$ is a diagonal matrix representing the prior term whose i-th element is given by Eq. 2.13, and w serves as a weighting coefficient for it, which we treat as a user-specified value (typically around 10^{-3}). The solution to Eq. 2.14 for $\mathbf{f}_x$ becomes the next estimate $\mathbf{f}_x^{m+1}$, from which $\mathbf{S}^{m+1}$ is computed, and this process is iterated. Eq. 2.14 can be solved rapidly by the *second-order stationary iterative method* [8], with the use of *fast Fourier transform* (FFT) for matrix multiplication by $\mathbf{H}$ and $\mathbf{H}^T$. We set the observation as an initial estimate: $\mathbf{f}_x^0 = \mathbf{g}_x$. Following Welk *et al*. [95], for better noise suppression in the early stage of deconvolution, we first run a few (around 3) EM iterations with w being twice the user-specified value, and then we run another few EM iterations with the original value. The y part of Eq. 2.7 is solved similarly. After obtaining estimated latent gradients $\mathbf{f}_x'$ and $\mathbf{f}_y'$, we reconstruct the deblurred image $\mathbf{f}'$ by solving a Poisson equation [74] with a multigrid solver. As we use FFT, periodic boundary conditions are assumed. Edge tapering is performed to reduce boundary effects, and the DC component lost by the derivative filters is restored from the input photograph. As with WaveGSM, positivity of pixel values is not enforced so far. We clamp any negative pixel values to a small positive value (1/255), and run additional (around 10) Richardson-Lucy iterations [58, 79] in the image domain.

The time complexity of our method is $O(P\log P)$ in the number P of pixels owing to the use of FFT convolution for matrix multiplication, which remains the same as that of WaveGSM. However, the amount of computation is significantly reduced in two respects as illustrated in the flowcharts shown in Fig. 2.4. First, we have only $O(P)$ derivative coefficients to be updated, in contrast to $O(P\log P)$ TI-DWT coefficients. Second, WaveGSM performs $O(P\log P)$ TI-DWT and its inverse for each iteration, whereas our method performs derivative and its inverse (i.e., integral) operations only at the beginning

(by deriving Eq. 2.7 from Eq. 2.5) and at the end (by solving a Poisson equation) of the whole deconvolution process.

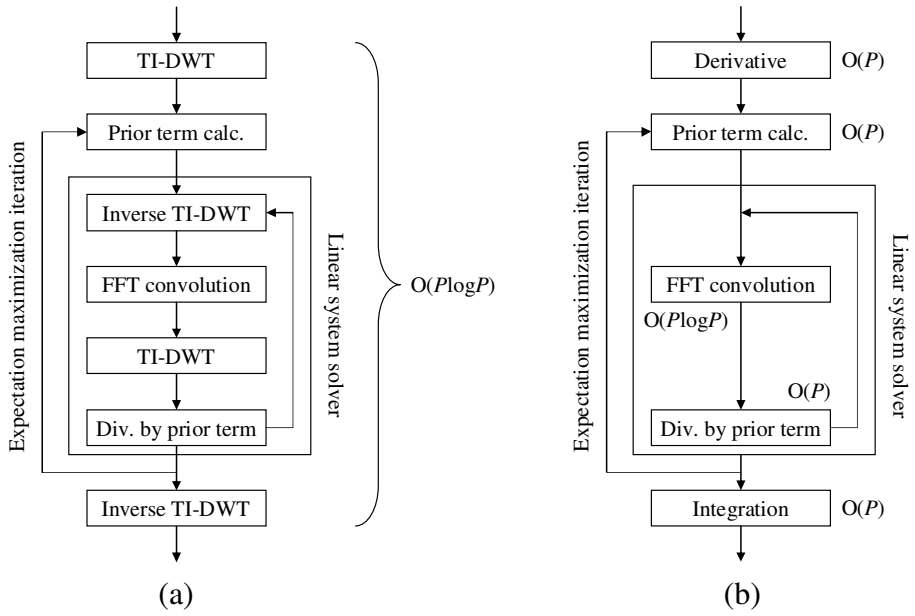


Figure 2.4: Processing flows of deconvolution algorithms. (a) WaveGSM [15]. (b) Proposed method.

2.4 Local Blur Estimation

Similar to the existing spatially-variant PSF estimation techniques, we divide an image into segments, and we assume the blur to be uniform within each segment. However, rectangular segmentation as in [72, 44] can produce segments that violate this uniformity assumption, as the blur radius can change abruptly due to depth discontinuities at object boundaries. Therefore, we perform color image segmentation [25] so that segments conform to the scene content. In what follows, we present a blur radius estimation method that is applicable to non-rectangular segments.

Our approach is to select the blur radius from the predetermined $M+1$ candidate blur radii $\{r_j\}$ that gives the "best" deblurred image for each segment. Unfortunately, *focus measures* [84, 40] are not suitable for this selection criterion, because digitally deconvolved images with wrong blur radii have different image statistics from optically misfocused images. Instead, we measure the amplitude of oscillatory artifacts in deblurred images due to overcompensation of blur (examples can be seen in Fig. 2.11). For simplicity, we explain this phenomenon using the 1D version of Eq. 2.2:

$$g(x) = h(x;r) * f(x) + n(x), \tag{2.15}$$

where the PSF is given by the following box function:

$$h(x;r) = \begin{cases} 1/2r & \text{for } |x| \leq r \\ 0 & \text{otherwise} \end{cases} . \tag{2.16}$$

In the frequency domain, Eq. 2.15 is rewritten as

$$\hat{g}(\omega) = \text{sinc}(r\omega)\hat{f}(\omega) + \hat{n}(\omega), \tag{2.17}$$

where the *hat* notation (such as $\hat{g}$) denotes the Fourier transform of a given signal, and ω denotes frequency. The Fourier transform of $h(x;r)$ is $\text{sinc}(r\omega)$ [20]. Neglecting the noise, an approximate solution to Eq. 2.17 can be given by the following equation, known as pseudo-inverse filtering:

$$\hat{f}'(\omega) = \frac{\text{sinc}(r\omega)}{\text{sinc}^2(r\omega) + \varepsilon}\, \hat{g}(\omega), \tag{2.18}$$

where ε is a small number (around 10^{-3}) to avoid zero division at $\omega = k\pi/r$ ($k = \pm 1, \pm 2, \cdots$). If $\hat{g}(\omega)$ is non-zero at these frequencies, it is overly amplified (scaled by $1/\varepsilon$), which results in oscillation in the deblurred image. As it is often the case that $|\hat{g}(\omega)|$ is a decreasing function with respect to $|\omega|$, major oscillation occurs at $\omega = \pm\pi/r$, which emerges as striped artifacts with an interval of $2r$ pixels.

Suppose we deblur a signal that has been blurred with radius r by a pseudo-inverse filter with radius R. Then at the major oscillation frequency $\omega = \pi/R$, we obtain the following equation from Eqs. 2.17 and 2.18 (similar for $\omega = -\pi/R$):

$$\hat{f}'(\pi/R) = \frac{1}{\varepsilon}\left(\text{sinc}(\pi r/R)\hat{f}(\pi/R) + \hat{n}(\pi/R)\right). \tag{2.19}$$

Fig. 2.10(a) shows a plot of $|\hat{f}'(\pi/R)|$ as a function of R, assuming that $|\hat{f}(\omega)|$ is also a decreasing function and that $|\hat{n}(\omega)|$ is constant (white noise) and is small compared to $|\hat{f}(\omega)|$ except for high frequencies. From this plot we can expect to observe large oscillation in deblurred images for $R > r$. Therefore, the maximum radius with which pseudo-inverse filtering does not produce large oscillation is estimated to be the true blur radius. The above discussion is also applicable to the 2D case, as the Fourier transform of Eq. 2.1 has a similar shape to circular sinc functions [20].

For each candidate radius r_j, we apply pseudo-inverse filtering to an input photograph with that radius, and we measure the amplitude of oscillation by the ratio of the number of pixels within each segment whose values are out of range $[\theta_{c,min} - \delta, \theta_{c,max} + \delta]$, where $[\theta_{c,min}, \theta_{c,max}]$ is the original range of pixel values within that segment of an input photograph for each color channel c, and δ is a small number around 0.1. This *oscillation measure* can be easily computed for arbitrarily-shaped segments. For reliability, however,

we exclude too small or thin segments (e.g., under 100 pixels). From a set of blur radii $\{r_j\}$, we identify the maximum radius having the oscillation measure below a certain threshold as the true blur radius. If this measure never exceeds the threshold, which typically occurs for segments with minimal color variance, we do not make an estimate for those segments.

A blur radius field $r_{org}(x,y)$ is obtained by stitching the estimated blur radii. Values in segments where no estimate was made as described above are interpolated from surrounding segments. We apply some smoothing to $r_{org}(x,y)$ in order to suppress occasional spurious estimates, and also to reduce step transitions that could lead to discontinuities in refocused images.

From a blur radius field $r_{org}(x,y)$ and deblurred images $f_j(x,y)$, we can reconstruct a latent image $l(x,y)$ by Eq. 2.3. As we cannot guarantee the blur estimation to be perfect, we provide users with a simple drawing interface in which pixel intensity corresponds to the size of a blur radius as shown in Fig. 2.5, so that users can interactively modify the estimated blur radius field. Modification to the blur radius field is immediately reflected in the latent image.

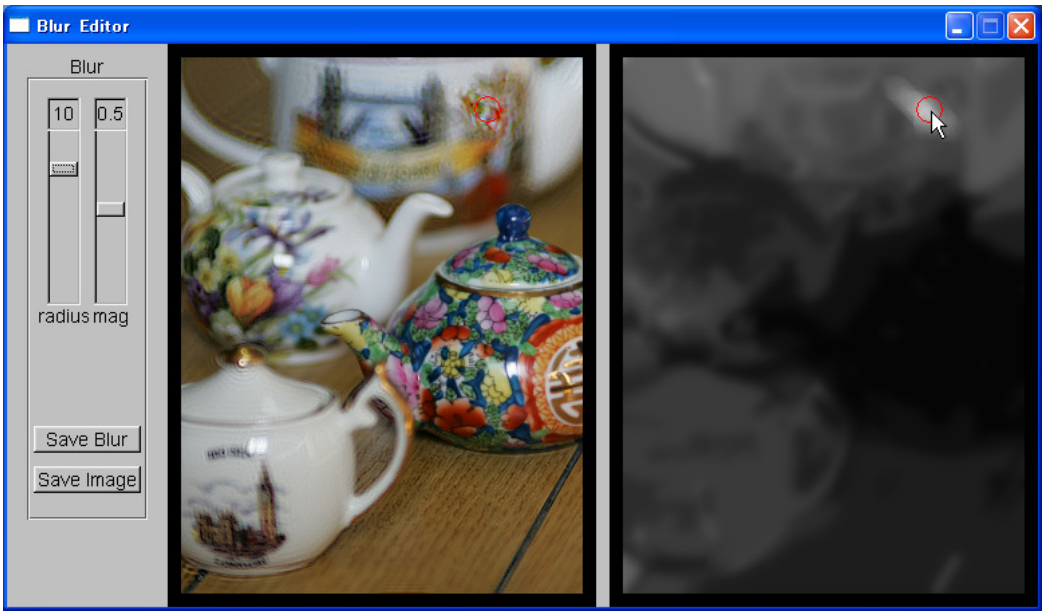


Figure 2.5: Screenshot of the blur field editing interface. The user draws on the grayscale image on the right which represents a blur radius field. In this example, the user is increasing the values around the upper right corner of the image, which are immediately reflected in the deblurred image on the left. The blur radius to be specified and the magnitude with which the specified radius influences the blur field can be adjusted by the sliders on the left.

2.5 Interactive Refocusing

Since defocus blur can serve to enhance artistic impression of photographs, in this section we aim to provide users with means to interactively control the focus settings of photographs after they are captured, based on the deblurred image and estimated blur radius field. Fig. 2.6 shows our example implementation of the proposed user interface. By the sliders on the left, users can make the depth-of-field of an input photograph wider or narrower, and the focused depth nearer or farther. They can also click on the image to bring the specified point in focus, in analogy with auto-focusing of a real camera. The three *refocus modes* on the top left will be explained later.

Figure 2.6: Screenshot of the interactive refocusing interface. The focused depth is immediately adjusted to the point specified by a mouse click.

To determine a new blur radius field $r_{new}(x,y)$ to be applied to the latent image in order to create desired refocusing effects, we associate a depth map $z(x,y)$ of the scene with the original blur radius field $r_{org}(x,y)$ through the ideal thin lens model [73] (see Fig. 2.7):

$$z(x,y) = \frac{F_0 v_0}{v_0 - F_0 - q_{org}(x,y) f_0}, \tag{2.20}$$

F_0, f_0, and v_0 are the original camera parameters, which represent the focal length, the f-number, and the distance between the lens and the image plane, respectively, and $q_{org}(x,y)$ is the original *signed* blur radius field, such that $r_{org}(x,y) = |q_{org}(x,y)|$. The sign of

$q_{org}(x,y)$ is related to the original focused depth $z_0 = F_0 v_0/(v_0 - F_0)$ as: $q(x,y) < 0$ for $z(x,y) < z_0$, and $q(x,y) > 0$ for $z(x,y) > z_0$.

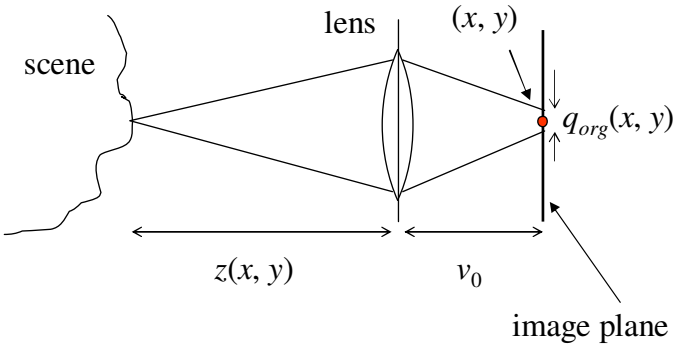


Figure 2.7: Thin lens defocus model.

As we can only estimate $r_{org}(x,y)$, we let users draw binary masks to specify the sign as shown in Fig. 2.8. Rough masks seem sufficient. If the scene depth is greater or smaller than the focused depth everywhere, users have only to declare so. For other cases, our drawing interface shown in Fig. 2.9 provides users with graph-cut image segmentation capability [17], so that the user has only to draw strokes sparsely on the image.

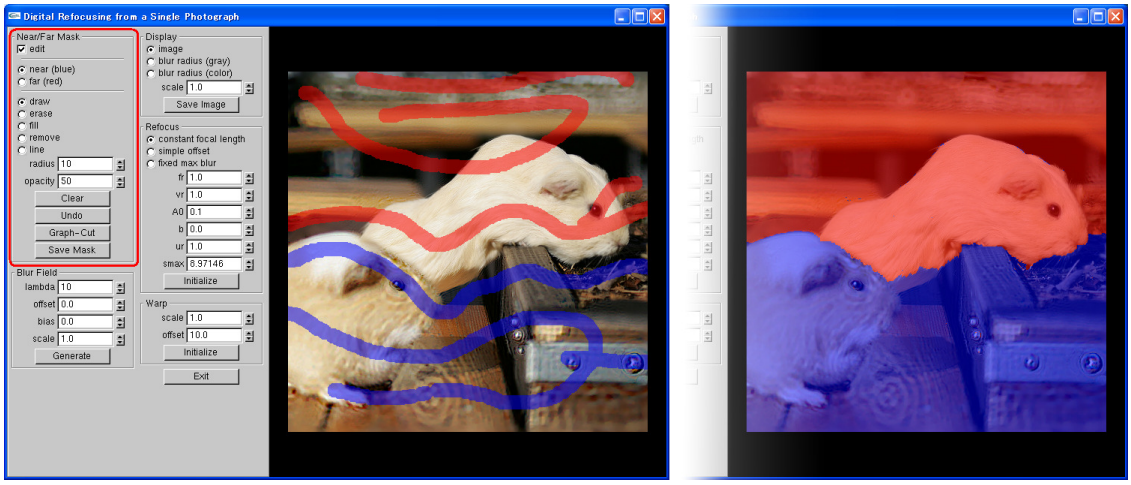
Figure 2.8: Binary masks specifying the sign of blur radius field $q_{org}(x,y)$. Top row: Input photographs. Bottom row: Corresponding masks. White indicates negative (nearer than the original focused depth), and black positive (farther).

Suppose that we change the camera parameters to F, f, and v, then a new *signed* blur radius field $q_{new}(x,y)$ is derived by using Eq. 2.20 as

$$\frac{F_0 v_0}{v_0 - F_0 - q_{org}(x,y) f_0} = \frac{Fv}{v - F - q_{new}(x,y) f}, \tag{2.21}$$

where we eliminated $z(x,y)$ to directly associate $q_{new}(x,y)$ with $q_{org}(x,y)$. Solving Eq. 2.21 for $q_{new}(x,y)$ leads to

$$q_{new}(x,y) = \frac{f_0 vF}{f v_0 F_0} q_{org}(x,y) + \frac{v_0 F_0 (v - F) - vF(v_0 - F_0)}{f v_0 F_0}, \tag{2.22}$$



(a) (b)

Figure 2.9: Screenshot of the signed mask specification interface. Only the portion of the panel enclosed by the red rectangle is used for this purpose. (a) The user drew some strokes to roughly specify the regions nearer than the focused depth (blue) and those farther (red). (b) The system automatically fill in the other regions using graph-cut image segmentation [17].

from which a new (unsigned) blur radius field to be added to the latent image is obtained as $r_{new}(x,y) = |q_{new}(x,y)|$.

The original camera parameters F_0, f_0, and v_0 may be obtained from EXIF data [38] embedded in JPEG files created by most of the recent digital cameras. However, some parameters are often unavailable, and EXIF data itself may not be available from converted or edited image files. In addition, it is not necessarily intuitive to manipulate the actual values when handling an image, not a camera. Therefore, we present three simplified versions of Eq. 2.22, in which relative camera parameters are used.

Constant Focal Length assumes the focal length to be constant: $F = F_0$. Then Eq. 2.22 simplifies to

$$q_{new}(x,y) = \frac{1}{f_r}\left(v_r q_{org}(x,y) + A_0(v_r - 1)\right), \tag{2.23}$$

where $f_r \equiv f/f_0$ is a relative f-number, $v_r \equiv v/v_0$ is a relative image plane distance, and $A_0 \equiv F_0/f_0$ is the original aperture. This equation has a good analogy to changing focus using a real camera.

Simple Offset assumes $vF = v_0F_0$. Eq. 2.22 becomes

$$q_{new}(x,y) = \frac{1}{f_r}(q_{org}(x,y) + q_{ofs}), \tag{2.24}$$

where $q_{ofs} \equiv ((v-F)-(v_0-F_0))/f_0$ is a blur radius offset. Though it is not realistic to change the parameters in this manner when handling a real camera, this equation provides

users with a simple and intuitive way of manipulating blur.

Fixed Max Blur assumes $v - F = v_0 - F_0$. Then,

$$q_{new}(x,y) = \frac{1}{f_r}\left(u_r\, q_{org}(x,y) + q_{max}(1-u_r)\right), \tag{2.25}$$

where $u_r \equiv vF/v_0F_0$, and $q_{max} \equiv (v_0 - F_0)/f_0$ is the maximum blur radius, corresponding to $z = \infty$ in Eq. 2.20. This equation is useful for refocusing among near objects while keeping far objects from becoming too sharp or blurry.

Using any one of the above three *refocusing modes*, users can interactively do the following three types of refocusing operations.

Changing depth-of-field. This operation can be done by changing relative f-number f_r. Increasing f_r extends the depth-of-field, whereas decreasing f_r makes it shallower.

Changing focus. This can be done by changing v_r, q_{ofs}, or u_r depending on the refocusing mode in use. The other parameters A_0 and q_{max} can also be adjusted, which we typically set to $\max\{r_{org}(x,y)\}$ for good refocusing effects.

Auto-focusing. Users can simply specify a point in a photograph which they want to be in focus. An appropriate value is automatically computed for the parameter of the selected refocusing mode so that $q_{new}(x,y) = 0$ at the specified point (x_s, y_s) as:

$$\begin{aligned} v_r &= A_0/(A_0 + q_{org}(x_s,y_s)), \\ q_{ofs} &= -q_{org}(x_s,y_s), \\ u_r &= q_{max}/(q_{max} - q_{org}(x_s,y_s)). \end{aligned} \tag{2.26}$$

In summary, from the user's point of view, the user will take the following steps to perform refocusing.

1. The user inputs a photograph. The system automatically produces a blur radius field and a set of deblurred images. This takes about 20 minutes for an image size of 512×512.

2. The user edits the blur radius field if it has noticeable errors, using the interface shown in Fig. 2.5. This typically takes a few to ten minutes.

3. The user draws a signed mask via the interface shown in Fig. 2.9. This takes less than a minute.

4. The user can now interactively refocus the photograph using the interface shown in Fig. 2.6.

2.6 Results

All of the input photographs shown in this paper were taken with a Canon EOS-1D Mark II camera and a Canon EF 28-70mm wide aperture (F2.8) lens. The image format was JPEG with sRGB color space (gamma-corrected with $\gamma = 2.2$). We inverted this gamma-correction during deconvolution and blur estimation.

2.6.1 Estimation and Removal of Uniform Blur

We first demonstrate the performance of our blur estimation and deconvolution methods for uniform defocus blur. For the images shown in Figs. 2.12(a)(f), in which the scenes have approximately uniform depths, we plotted their oscillation measure in Fig. 2.10(b), treating the whole image as one segment. The arrows show the estimated blur radii with a threshold of 0.01, which are 11 pixels for Fig. 2.12(a) and 7 pixels for Fig. 2.12(f). These results conform to visual inspection as shown in Fig. 2.11. Fig. 2.11 also shows that the number of out-of-range pixels (see Sec. 2.4) begins to increase as the pseudo-inverse filter radius exceeds the true blur radius.

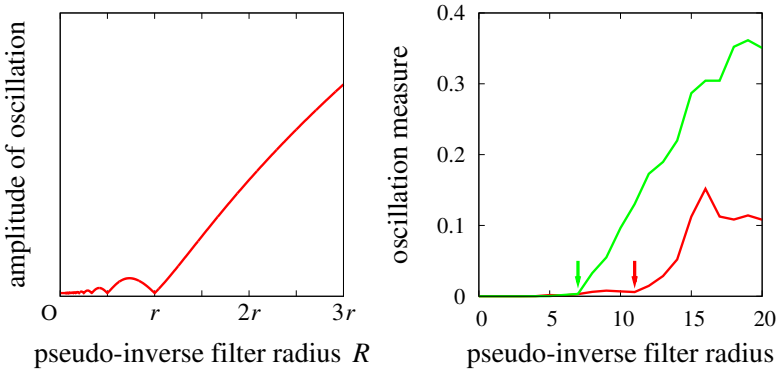


Figure 2.10: (a) Plot of the amplitude of oscillation $|\hat{f}'(\pi/R)|$ as a function of pseudo-inverse filter radius R. (b) Plots of the oscillation measure for Fig. 2.12(a) (red) and Fig. 2.12(f) (green), treating the whole image as one segment. The arrows show the estimated blur radii with a threshold of 0.01.

Based on the estimated blur radii, we applied our deconvolution method, along with other methods including Richardson-Lucy [58, 79], WaveGSM with ordinary DWT, and that with TI-DWT. Fig. 2.12 shows the results. Since Richardson-Lucy does not exploit explicit image priors, it produced less sharp images with noise (between the alphabets in Fig. 2.12(b)) and halo artifacts (around the hair and face in Fig. 2.12(g)). WaveGSM with DWT resulted in blocky images as expected (see Sec. 2.3). Our method produced

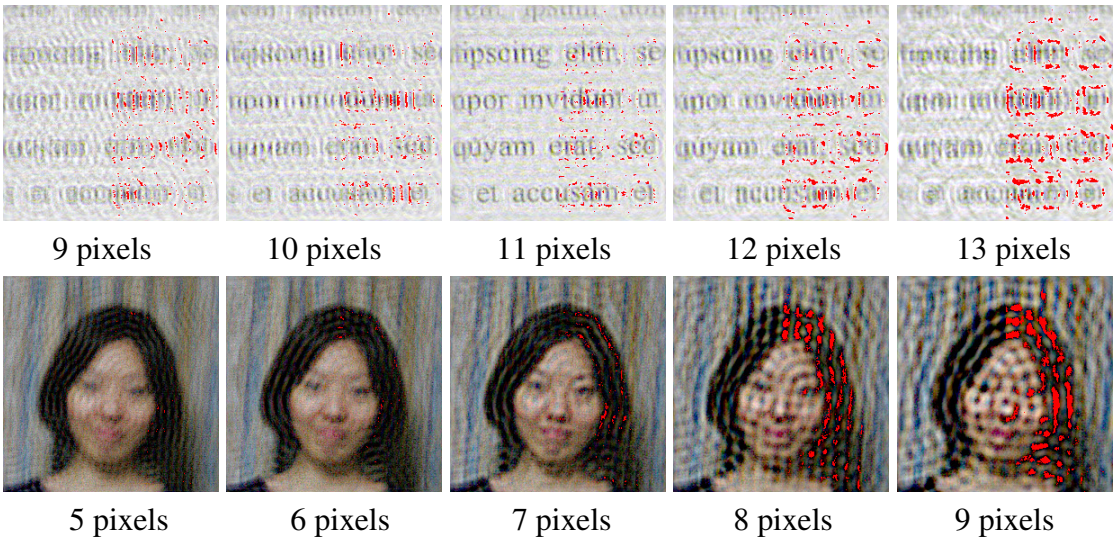


Figure 2.11: Results of pseudo-inverse filtering for Figs. 2.12(a)(f) with different blur radii. The out-of-range pixels are shown in red in the right half of each image.

better (for the alphabet image) or comparable (for the face image) results as compared to WaveGSM with TI-DWT, running about 10 times faster.

Since our deconvolution method does not assume a specific form of a PSF, we also applied it to the removal of camera shake from a photograph. Fergus *et al*. used Richardson-Lucy deconvolution to remove the camera shake PSF estimated by their method [28], and we replaced Richardson-Lucy by our deconvolution algorithm. Our method appears to produce less noisy results as shown in Fig. 2.13. In this example, we estimated the PSF by extracting the trajectory of a bright small object in the input photograph (not shown in the figure).

2.6.2 Estimation of Spatially-Variant Blur

Next, we show several local blur estimation results in Fig. 2.14. The input photographs are shown in Figs. 2.1(a), 2.16(a), 2.17(a), 2.18(a), and 2.19(a). We performed relatively fine segmentation to ensure estimation locality. The estimated radii approximately correspond to the scene depths. For comparison, we applied the spatially-variant blur estimation method by Özkan *et al*. [72]. This method is based on local Fourier transform, hence it employs rectangular segmentation. The results are shown in Figs. 2.15(a)(d). It failed in regions around object boundaries and also failed to identify small blur radii, leading to noisy latent images as shown in Figs. 2.15(b)(e). The corresponding latent images based on our blur estimation are shown in Figs. 2.15(c)(f).

2.6.3 User Intervention for Blur Estimation

Next, we show an example of user intervention for the estimated blur radius field mentioned in Sec. 2.4. Fig. 2.16(b) shows an image representing the estimated blur radius field after smoothing. Users can draw on this image to locally increase/decrease the values as shown in Fig. 2.16(c), for better visibility (Fig. 2.16(g) top) and ringing reduction (Fig. 2.16(g) middle and bottom). This can be done in an aesthetic sense to obtain a visually pleasing latent image, and the edited blur radius field needs not correspond to the scene depth. This user editing operation took from a few to ten minutes for our examples shown below.

2.6.4 Refocusing Results

Finally, we show several refocusing examples in Figs. 2.17, 2.18, and 2.19, in which we changed the depth-of-field and moved the focus nearer to or farther from the camera. Out-of-focus objects became sharp after they were brought into focus, as can be seen in the floret symbol at the bottom of the red crayon in Fig. 2.17(c) and the furry texture of the nearer marmot in Fig. 2.19 right.

When synthesizing Fig. 2.17(c) from Fig. 2.17(a), we used the refocusing equation Eq. 2.23, which simulates focus changes of a real camera (see Sec. 2.5). We obtained the synthesis result that well approximates a real photograph shown in Fig. 2.17(d). For Fig. 2.1, we used Eq. 2.24 for simple manipulation of blur radii. For Figs. 2.18 and 2.19, we used Eq. 2.25 to keep distant objects unaffected as they are too blurry to be fully restored.

For an image size of 512×512, our deconvolution described in Sec 2.3 took about 1 minute for each blur radius r_j, and the blur estimation 15 seconds on an Intel Pentium4 3.2GHz CPU. Although the theoretical time complexity is $O(P \log P)$, it seems $O(P)$ computation is dominant, and the deconvolution took 16 minutes and the blur estimation 4 minutes for a 4Mpixel image. Refocusing can be performed in real-time.

2.7 Summary

This chapter has presented a method for removing defocus blur in images in the context of digital refocusing, in which the goal is not only to perform deblurring but also to create images with different focus settings. This chapter has focused exclusively on an

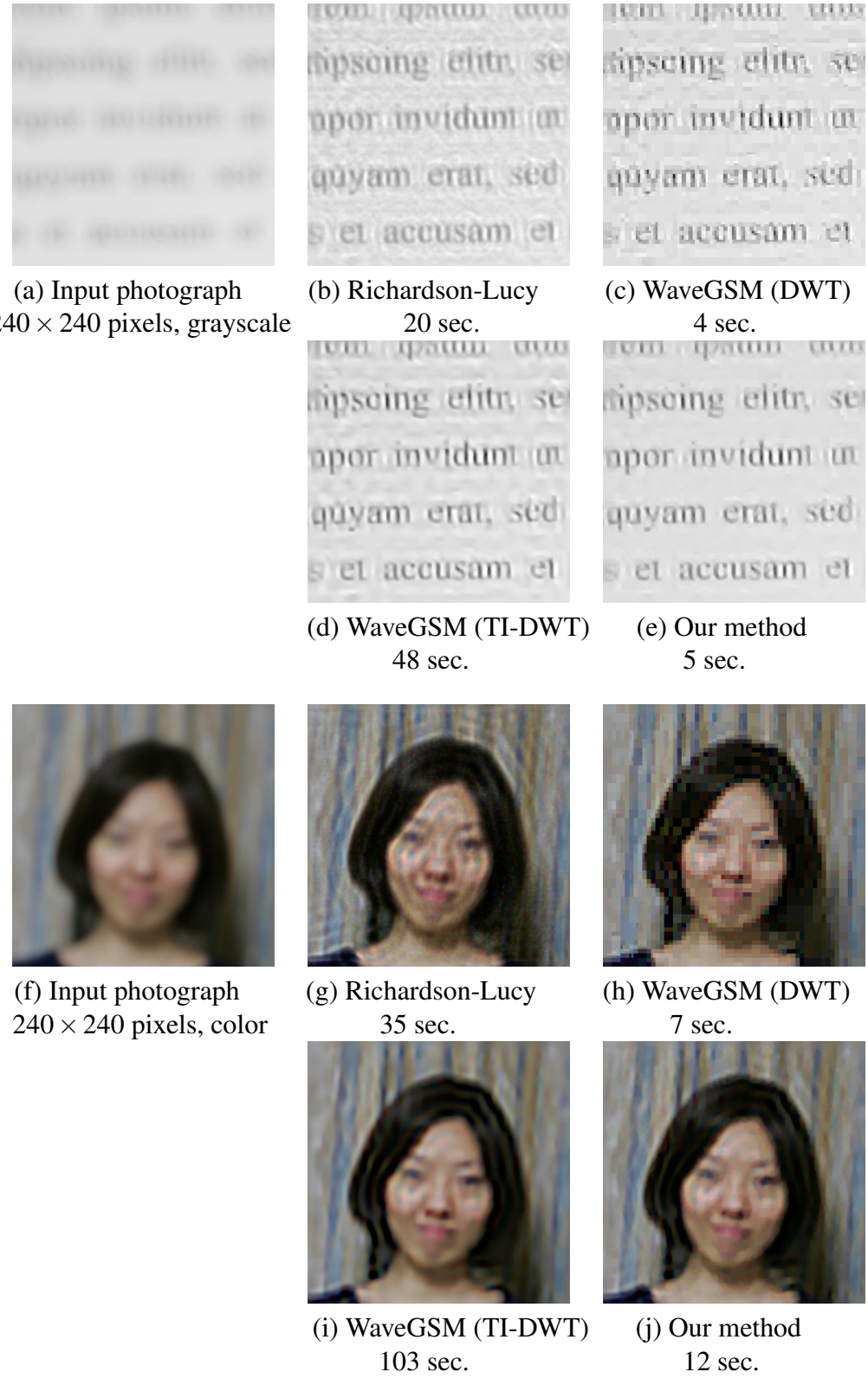
image processing approach without camera optics modifications, in order to set a baseline performance achievable without modifying the image capture process.

For image deconvolution, we have proposed a method for speeding up deconvolution computation while taking into account heavy-tailed priors for image derivative distributions. The proposed method achieves similar image quality with 1/10 computation time by taking derivatives beforehand and by working in the gradient domain. The proposed method can also be used for removing blur other than defocus blur.

For blur estimation, we have proposed a method which can handle abrupt blur changes at depth discontinuities due to object boundaries. Our method uses color image segmentation rather than the traditional rectangular segmentation to better divide the image into uniformly blurred regions, and the largest blur radius that does not cause ringing after deconvolution is selected for each region. Although the proposed method was shown to outperform the previous method, the estimated blur radius fields still needed to be retouched to obtain better deblurring and refocusing results.

For creating refocusing effects, we have presented several means of determining the amount of blur to be added to a restored sharp image based on the estimated blur, with which users can interactively control the focus settings of photographs after they are captured. While we have reduced the burden for the user by providing intuitive refocusing parameters and "auto-focusing" capability, it is still necessary for the user to draw a signed mask before performing refocusing operations.

In summary, we have found that we can achieve desired refocusing effects by exclusively relying on an image processing approach, but we had to introduce some user intervention, indicating that image processing alone is still not sufficient for making deblurring and refocusing processes completely automatic.



(a) Input photograph
240 × 240 pixels, grayscale

(b) Richardson-Lucy
20 sec.

(c) WaveGSM (DWT)
4 sec.

(d) WaveGSM (TI-DWT)
48 sec.

(e) Our method
5 sec.

(f) Input photograph
240 × 240 pixels, color

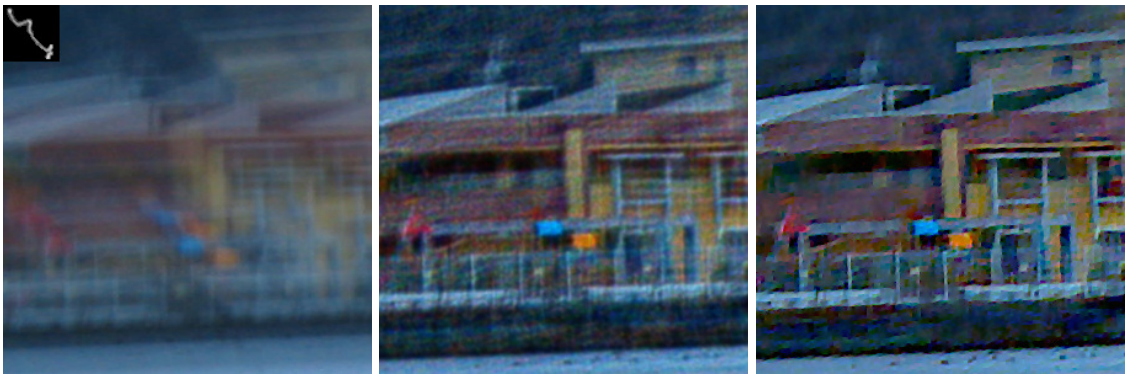
(g) Richardson-Lucy
35 sec.

(h) WaveGSM (DWT)
7 sec.

(i) WaveGSM (TI-DWT)
103 sec.

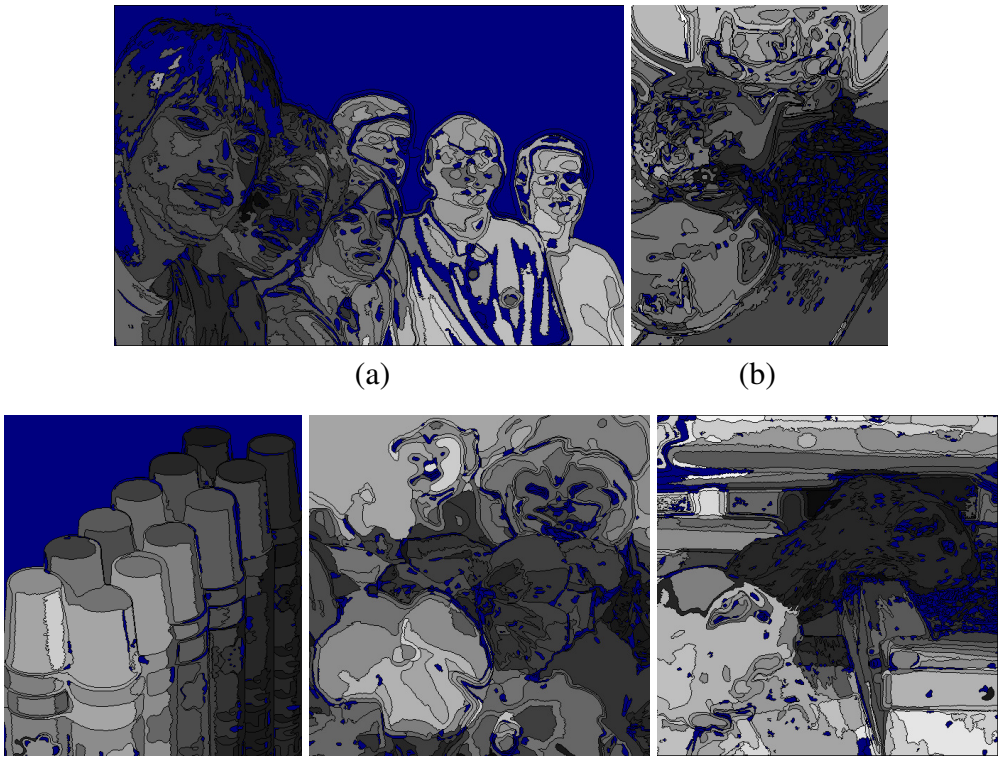
(j) Our method
12 sec.

Figure 2.12: Comparison of four deconvolution methods and their computation times.



(a) (b) (c)

Figure 2.13: (a) Photograph spoiled by camera shake. The estimated PSF is shown in the top left corner. (b) Result of Richardson-Lucy deconvolution. (c) Result of our deconvolution method.



(a) (b)

(c) (d) (e)

Figure 2.14: Results of our local blur estimation shown in gray-level. The maximum intensity (white) corresponds to a blur radius of 10 pixels. The blue regions indicate that no estimate was made there. The black lines show the segmentation boundaries.

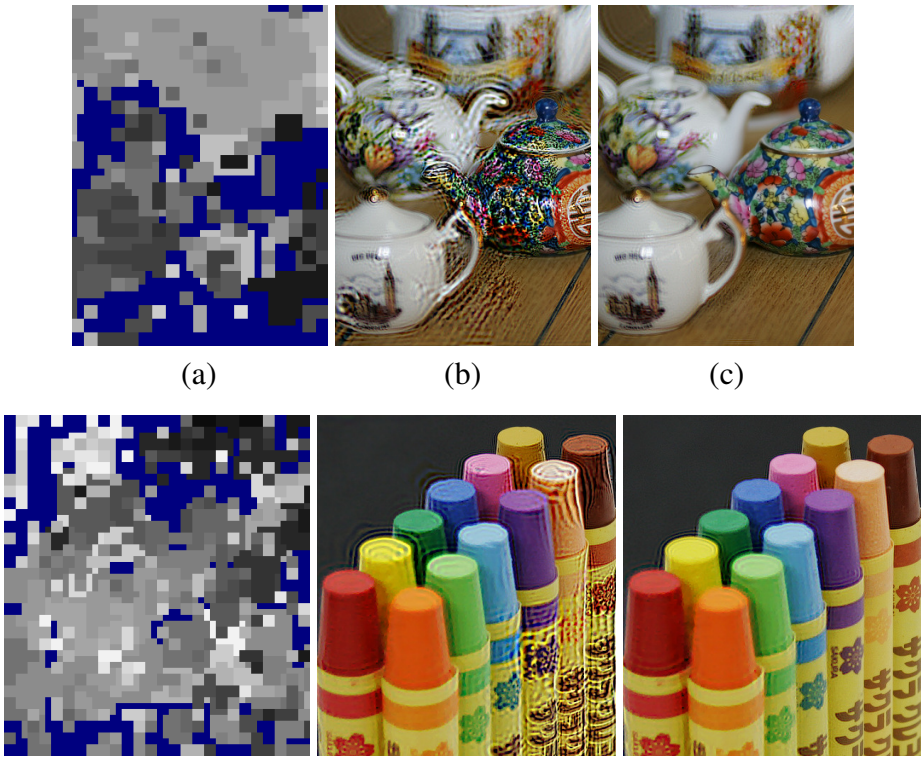


Figure 2.15: Comparison with the existing blur estimation method [72]. (a) Estimation result for the teapot image shown in Fig. 2.16(a). (b) Latent image based on (a). (c) Latent image based on our estimate shown in Fig. 2.14(b). (d) Estimation result for the crayon image shown in Fig. 2.17(a). (e) Latent image based on (d). (f) Latent image based on our estimate shown in Fig. 2.14(c).

(a) (b) (c)

(d) (e) (f) (g) (h)

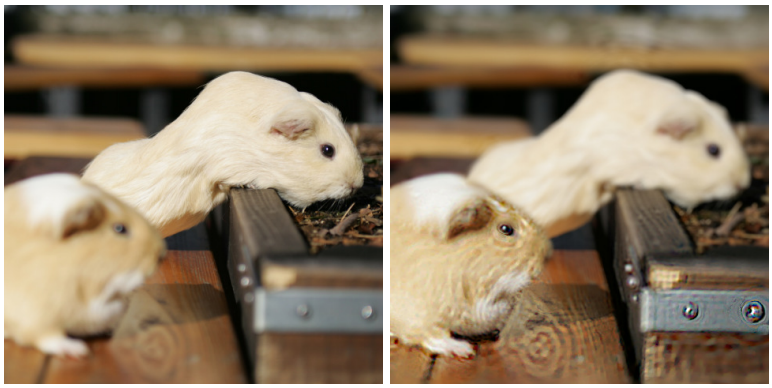
Figure 2.16: Example of user intervention for a blur radius field. (a) Input photograph. (b) Blur radius field after filling in the undefined (blue) regions in Fig. 2.14(b) and after smoothing. (c) Edited blur radius field. The red circles indicate the edited regions. (d) Latent image based on (b). This is the same as Fig 2.15(c). (e) Latent image based on (c). (f) Magnified crops from the red rectangles in (d) (before editing). (g) Magnified crops from the corresponding red rectangles in (e) (after editing). (h) Refocused image, created by using the image (e) as a latent image.

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

Figure 2.17: (a) Input photograph, focused on the brown crayon. (b) Created image with a shallow depth-of-field. (c) Created image, refocused on the orange crayon. (d) Ground truth photograph, focused on the orange crayon.

(a) (b)

Figure 2.18: (a) Input photograph, focused on the flower in the center. (b) Created image, refocused on the flower in the top right corner.



(a) (b)

Figure 2.19: (a) Input photograph, focused on the farther marmot. (b) Created image, refocused on the nearer marmot.

Chapter 3

Defocus Blur Removal using a Color-Filtered Aperture

The previous chapter has shown that automatic defocus deblurring that is solely based on image processing is still not feasible, and user intervention for blur estimation was necessary. This chapter describes a method for defocus deblurring consisting of a camera lens modification with color filters and associated image processing techniques in order to achieve automatic defocus blur estimation.

3.1 Introduction

Wide aperture lenses are efficient in increasing the amount of incoming light so as to improve the signal-to-noise ratio of captured images. However, they make the depth-of-field shallow, and only objects located at a limited range of depth can be focused sharply.

In this chapter we intend to obtain an all-in-focus image by removing defocus blur whose size is dependent on the scene depth. While coded aperture techniques [48, 90] facilitate both blur estimation and deconvolution, the blur estimation has to rely solely on defocus cues, requiring some manual intervention and also resulting in ambiguity between depths farther and nearer than the focused depth (this ambiguity was also present in Chapter 2, and we let the user draw a mask as shown in Fig. 2.8). Instead, we propose to use a color-filtered aperture mask to exploit parallax cues which escape the above-mentioned depth ambiguity.

By dividing the aperture into three regions through which only light in one of the RGB color bands can pass, we can acquire three shifted views of a scene in the RGB planes of a captured image in a single exposure. This allows us to take stereo correspondence

between the RGB planes to estimate the scene depth, which is directly related to the defocus blur size. A challenge we must address in using a color-filtered aperture is that, as a scene is captured with three different bands of wavelength, corresponding points in the RGB planes generally have different signal intensity levels. We develop a color alignment measure to find correspondence between such RGB signals. Moreover, we propose a method for extracting the matte of an in-focus foreground object, so that the extracted foreground will be free from possible degradation due to deconvolution. Color misalignment cues introduced by the color filters serve to constrain the space of possible mattes that would otherwise contain erroneous mattes when foreground and background colors are similar.

The proposed imaging system is portable, and it only requires off-the-shelf color filters for cameras as additional optical elements. The downsides of using a color-filtered aperture are that objects having only a single pure R, G, or B color cannot be handled, and that the visual quality of images is spoiled by color misalignment. We will show, however, that our method can handle many real-world objects, and we also present how to reconstruct color-aligned images using extracted depth and matte.

3.2 Color-Filtered Aperture

Fig. 3.1(a) shows our prototype camera lens with color filters in the aperture. We arranged the RGB regions so that their displacement with respect to the optical center of the lens aligns with the X and Y axes of the image sensor, as indicated by the arrows in Fig. 3.1(b). By this arrangement, a scene point farther than the focused depth is observed with a rightward shift in the R plane, an upward shift in the G plane, and a leftward shift in the B plane. A scene point nearer than the focused depth will be shifted in the opposite directions. Note that these color shifts come from geometric optics, not from chromatic aberration. Fig. 3.2 illustrates this phenomenon in 2D where the aperture is split into two (R and G) regions.

(a) (b)

Figure 3.1: (a) Camera lens with color filters placed in the aperture. (b) Filter arrangement.

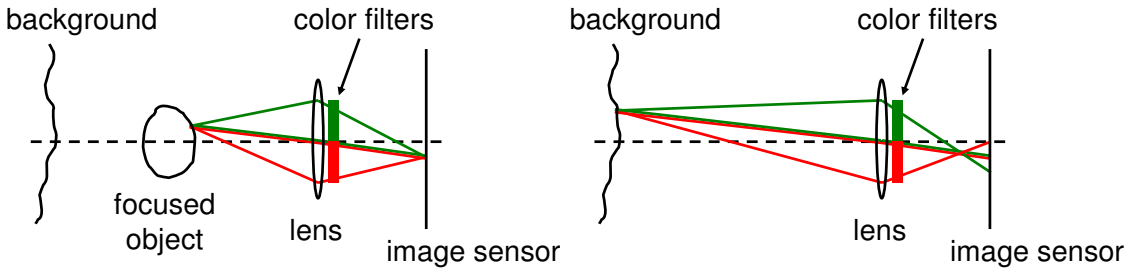


Figure 3.2: 2D illustration of the interactions between light rays from a scene point and a color-filtered aperture. (a) For a scene point at the focused depth, light rays in the R band and those in the G band converge to the same point on the image sensor. (b) For a scene point off the focused depth, light rays in the two bands reach different positions on the image sensor, resulting in a color shift.

Fig. 3.3 shows an example photograph and its separated RGB planes. Due to the higher transmittance of the R filter, captured images are relatively reddish.

Figure 3.3: Example photograph taken with our lens, and its separated RGB planes. The white lines are superimposed to highlight the background color shifts. See Fig. 3.16(a) for a closeup view.

3.3 Defocus Blur Estimation

The RGB planes I_r, I_g, and I_b of a captured image $\mathbf{I}$ correspond to three views of a scene. If we take a virtual center view (*cyclopean view*) as a reference coordinate system, the R, G, and B planes are shifted to rightward, upward, and leftward according to the arrangement of the aperture color filters. Meanwhile, the defocus PSF for each color plane is an oblique square corresponding to the filter shape shown in Fig. 3.1. Since the size of the PSF measured by the half diagonal of the square (the length of the arrows in Fig. 3.1(b)) is equal to the disparity between the RGB planes, we can estimate blur sizes as a stereo correspondence problem between the RGB planes. Therefore, letting d be a hypothesized disparity at (x,y), we need to measure the quality of a match between $I_r(x{+}d,y), I_g(x,y{-}d)$, and $I_b(x{-}d,y)$.

Clearly, we cannot expect these three values to have similar intensities because they are recorded with different bands of wavelength. To cope with this issue, inspired by Levin *et al.*'s matting approach [50], we exploit the tendency of colors in natural images to form elongated clusters in the RGB space (*color lines model*) [71]. We assume that pixel colors within a local window $w(x,y)$ around (x,y) belong to one cluster, and we use the magnitude of the cluster's elongation as a correspondence measure. More specifically, we consider a set $S_I(x,y;d)$ of pixel colors with hypothesized disparity d as $S_I(x,y;d) = \{(I_r(s{+}d,t), I_g(s,t{-}d), I_b(s{-}d,t)) \mid (s,t) \in w(x,y)\}$, and search for d that minimizes the following *color alignment measure*:

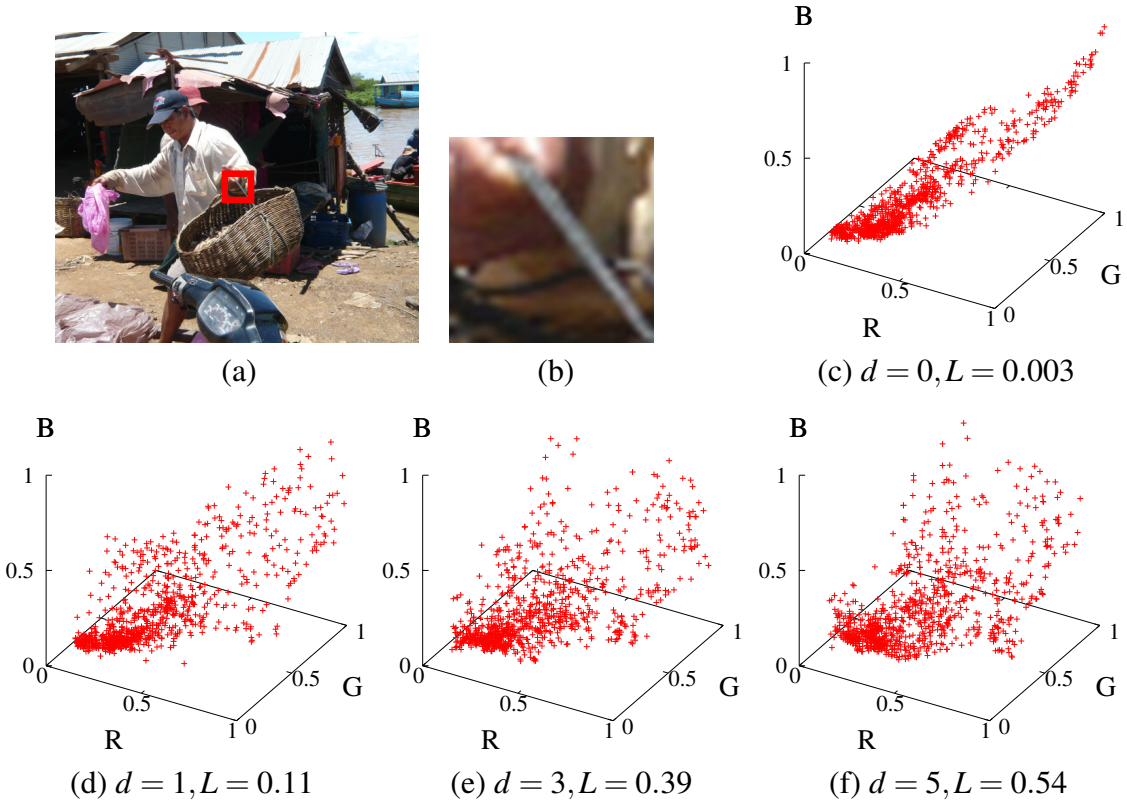
$$L(x,y;d) = \frac{\lambda_0\lambda_1\lambda_2}{\sigma_r^2\sigma_g^2\sigma_b^2}, \tag{3.1}$$

where λ_0, λ_1, and λ_2 ($\lambda_0 \geq \lambda_1 \geq \lambda_2 \geq 0$) are the eigenvalues of the covariance matrix Σ of the color distribution $S_I(x,y;d)$, and σ_r^2, σ_g^2, and σ_b^2 are the diagonal elements of Σ. Note that the dependence on $(x,y;d)$ of the right-hand side of Eq. 3.1 is omitted for brevity. $L(x,y;d)$ is the product of the variances of the color distribution along the principal axes, normalized by the product of the variances along the RGB axes. It gets small when the cluster is elongated (i.e., $\lambda_0 \gg \lambda_1, \lambda_2$) in an oblique direction with respect to the RGB axes, meaning that the RGB components are correlated. In fact, this measure can be interpreted as an extension of *normalized cross-correlation* (NCC) [53] so that it is applicable to three images simultaneously (see Appendix A). $L(x,y;d)$ is in the range [0, 1], with the upper bound given by Hadamard's inequality [33], since $\lambda_0\lambda_1\lambda_2 = \det(\Sigma)$.

To illustrate the effect of this measure, we use a sample image shown in Fig. 3.4(a), taken with a conventional camera lens. Since its RGB planes are aligned, the true disparity is $d = 0$ everywhere, and colors within the local window indicated by the red rectangle

in Fig. 3.4(a) actually form an elongated cluster, as shown in Fig. 3.4(c). If we deliberately misalign the RGB planes by $d = 1, 3$, and 5 pixels, the distribution becomes more isotropic, and the color alignment measure becomes larger, as shown in Figs. 3.4(d-f).

Now that we can evaluate the quality of a match between the RGB planes, we can find the disparity d that minimizes $L(x, y; d)$ at each pixel (x, y), from a predetermined set of disparity values (-5 to 10 in our implementation). As local estimates alone are prone to error, we use the standard energy minimization framework using graph-cuts [16] to impose spatial smoothness constraints.



(a) (b) (c) $d = 0, L = 0.003$

(d) $d = 1, L = 0.11$ (e) $d = 3, L = 0.39$ (f) $d = 5, L = 0.54$

Figure 3.4: (a) Sample photograph taken with a conventional camera lens. (b) Closeup of the local window indicated by the red rectangle in (a). (c-f) Plots of the pixel colors within the local window in the RGB space. The values d and L shown below each plot are the simulated disparity and the value of Eq. 3.1.

3.4 Matting

This section describes a method for extracting the matte of an in-focus foreground object, so that the extracted foreground will be free from possible degradation due to deconvolution. Matting is a problem of solving for foreground opacity $\alpha(x,y)$ at each pixel (x,y) in the following *matting equation*.

$$\mathbf{I}(x,y) = \alpha(x,y)\mathbf{F}(x,y) + (1-\alpha(x,y))\mathbf{B}(x,y), \tag{3.2}$$

which models an observed image $\mathbf{I}$ as a convex combination of a foreground color $\mathbf{F}$ and a background color $\mathbf{B}$. By capturing an image so that a foreground object is in focus, we can assume that $\alpha(x,y)$ is aligned between the RGB planes. More precisely, regions with fractional alpha values (i.e., the silhouette of a foreground object) should be within the depth-of-field of the lens. Slight violation of this assumption however does not lead to severe degradation of extracted mattes, as will be shown in Sec. 3.6.

Solving Eq. 3.2 based only on the observation $\mathbf{I}$ is an under-constrained problem, since we have only three measurements (I_r, I_g, and I_b) for seven unknowns (α, F_r, F_g, F_b, B_r, B_g, and B_b) at each pixel. Therefore, to incorporate additional constraints, we use a trimap which we automatically generate from the disparity map, and we also leverage the difference in misalignment between foreground and background colors to iteratively optimize the matte.

3.4.1 Matte Optimization Flow

Algorithm 3.1 shows our iterative matte optimization procedure. For initialization, we first roughly divide the image into foreground and background regions by thresholding the disparity map, and we dilate their border to construct a trimap having a conservatively wide "unknown" region (50-70 pixels in our implementation), as shown in Fig. 3.5(a). We then initialize the alpha values using a trimap-based matting method, for which we used Levin *et al.*'s *Closed-Form Matting* [50]. While this often gives already good results, errors can remain where foreground and background colors are similar (see Fig. 3.12(a) as an example). We detect and correct these errors in the subsequent iterative optimization using color misalignment cues. To determine how the foreground and background colors are misaligned in the "unknown" region, we make a two-layer assumption for the scene around the foreground silhouette. And we propagate the disparity values from the "strictly foreground" region to obtain foreground disparity map $d_F(x,y)$ as shown in Fig. 3.5(b). Similarly we obtain background disparity map $d_B(x,y)$ from the "strictly background" region (Fig. 3.5(c)).

In the iterative optimization, letting n denote an iteration count, we first estimate foreground and background colors $\mathbf{F}_n$ and $\mathbf{B}_n$ based on the current matte α_n, by minimizing a quadratic cost function $\sum_{(x,y)} ||\mathbf{I}(x,y) - \alpha_n(x,y)\mathbf{F}_n(x,y) - (1-\alpha_n(x,y))\mathbf{B}_n(x,y)||^2$ derived from Eq. 3.2, plus smoothness constraints on foreground and background colors, similar to [50]. These estimated colors $\mathbf{F}_n$ and $\mathbf{B}_n$ have errors in the same regions as α_n has errors. We detect these erroneous regions by measuring how consistent the estimated colors are with the foreground and background disparity maps $d_F(x,y)$ and $d_B(x,y)$, as we will describe in Sec. 3.4.2. We then correct the alpha values around the detected regions to obtain the matte α_{n+1} for the next iteration (Sec. 3.4.3). We iterate this process until change in the matte is sufficiently small. Fig. 3.6 illustrates each step of the iterative optimization.

Algorithm 3.1: Matte optimization algorithm.

Initialization

1. Construct a trimap from the disparity map.
2. Find an initial matte α_0 based on the trimap.
3. Propagate the disparity values to obtain foreground and background disparity maps d_F and d_B.

Iterative optimization

1. Estimate foreground color $\mathbf{F}_n$ and background color $\mathbf{B}_n$ based on the current α_n.
2. Compute consistency measures C_{F_n} and C_{B_n} (Sec. 3.4.2).
3. Update α_{n+1} based on C_{F_n} and C_{B_n} (Sec. 3.4.3).
4. Repeat until convergence.

3.4.2 Measuring Consistency with Disparity Maps

Similar to the color alignment measure in Eq. 3.1, we consider a set $S_F(x,y;d)$ of pixel colors within a local window $w(x,y)$, in this case for the foreground color $\mathbf{F}(x,y)$, not for the input image $\mathbf{I}(x,y)$ as $S_F(x,y;d) = \{(F_r(s+d,t), F_g(s,t-d), F_b(s-d,t)) \mid (s,t) \in w(x,y)\}$ with hypothesized disparity d, and we define a foreground *color lines model error* as follows.

$$e_F(x,y;d) = \frac{1}{N}\sum_{i=1}^{N} l_i^2, \tag{3.3}$$

(a) (b) (c)

Figure 3.5: (a) Trimap for the toy dog image in Fig. 3.3, constructed from the disparity map shown in Fig. 3.10(a) top. White: strictly foreground. Black: strictly background. Gray: unknown. (b) Propagated foreground disparity map $d_F(x,y)$. Blue indicates an undefined region. (c) Propagated background disparity map $d_B(x,y)$.

where $N = |S_F(x,y;d)|$, and l_i is the distance of the i-th color in $S_F(x,y;d)$ from the line fitted to the elongated color cluster (i.e., the first principal axis). Intuitively, we examine whether the colors in a local window fit the color lines model. Therefore, $e_F(x,y;d)$ becomes large when d is a wrong disparity. We define the background color lines model error $e_B(x,y;d)$ similarly. See Appendix B for more details.

As we have two possible disparities $d_F(x,y)$ and $d_B(x,y)$ at each pixel (x,y) in the "unknown" region, we define foreground and background *color consistency measures* by incorporating two values of color lines model errors at these two disparities:

$$\begin{aligned} C_F(x,y) &= \exp\left\{(e_F(x,y;d_F) - e_F(x,y;d_B))/\kappa_s\right\}, \\ C_B(x,y) &= \exp\left\{(e_B(x,y;d_B) - e_B(x,y;d_F))/\kappa_s\right\}, \end{aligned} \tag{3.4}$$

where κ_s is a scale parameter. If the estimated foreground color around (x,y) erroneously contains the (true) background color, $C_F(x,y)$ will be large around that region because $e_F(x,y;d_F)$ will be large and $e_F(x,y;d_B)$ will be small. The effect of the background counterpart $C_B(x,y)$ can be similarly explained.

3.4.3 Solving for the Matte

Following Wang and Cohen's *Robust Matting* approach [93], we solve for $\alpha(x,y)$ as a *soft graph-labeling* problem, where each pixel (regarded as a node in a graph) has *data weights* $W_F(x,y)$ and $W_B(x,y)$, and each pair of neighboring pixels has an *edge weight* $W_e(x_0,y_0;x_1,y_1)$. The data weight $W_F(x,y)$ is responsible for pulling $\alpha(x,y)$ toward 1, whereas $W_B(x,y)$ pulls it toward 0. The edge weights impose spatial smoothness constraints on alpha values by the *Matting Laplacian* [50]. This formulation is beneficial in that it can be solved as a sparse linear system [34], not graph-cuts, and that it guarantees $\alpha(x,y)$ to be in the range [0, 1] without additional hard constraints.

While Wang and Cohen [93] used color samples gathered from the "strictly foreground" and "strictly background" regions to set the data weights, we instead iteratively update the data weights according to the consistency measures $C_{F_n}(x,y)$ and $C_{B_n}(x,y)$ computed for the current estimate of the foreground and background colors $\mathbf{F}_n$ and $\mathbf{B}_n$, as follows.

$$\begin{aligned} W_{F_n}(x,y) &= \kappa_\alpha \alpha_n(x,y) &&+ \kappa_c(C_{B_n}(x,y) - C_{F_n}(x,y)), \\ W_{B_n}(x,y) &= \kappa_\alpha(1-\alpha_n(x,y)) &&+ \kappa_c(C_{F_n}(x,y) - C_{B_n}(x,y)), \end{aligned} \tag{3.5}$$

where κ_α and κ_c are constants. We clamp $W_{F_n}(x,y)$ and $W_{B_n}(x,y)$ at 0 to keep them nonnegative. When the foreground consistency measure $C_{F_n}(x,y)$ is smaller (i.e., more consistent) than the background counterpart $C_{B_n}(x,y)$, the foreground data weight $W_{F_n}(x,y)$ is increased while the background data weight $W_{B_n}(x,y)$ is decreased, so that $\alpha(x,y)$ is pulled toward 1 from the current value $\alpha_n(x,y)$. Conversely, $\alpha(x,y)$ will be pulled toward 0 if $C_{F_n}(x,y) > C_{B_n}(x,y)$.

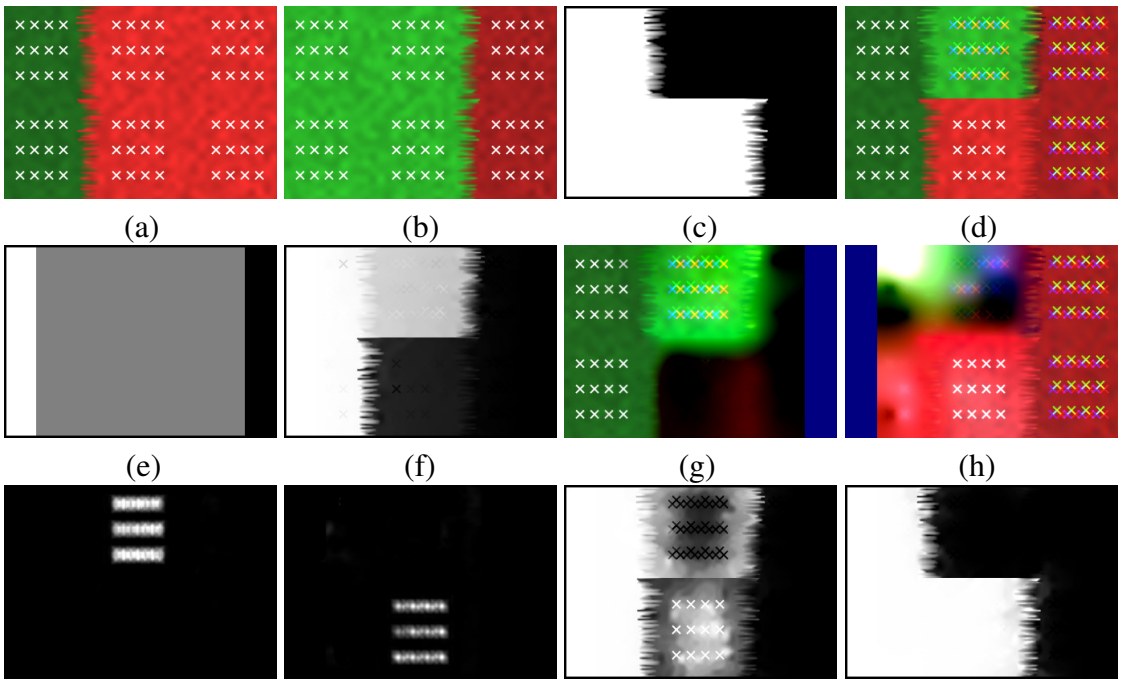
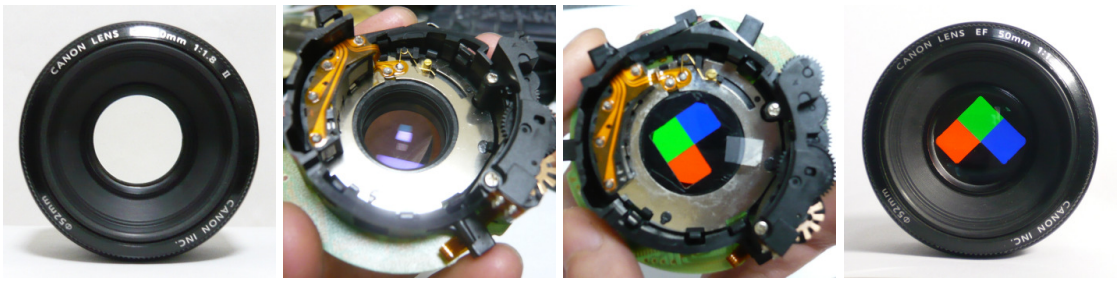


Figure 3.6: Synthetic toy example demonstrating how our matte optimization works. (a) Ground truth foreground color. (b) Ground truth background color. (c) Ground truth matte. (d) Composite image from (a-c) with the background color misaligned by 5 pixels. This image is input to our matting algorithm. (e) Trimap. In this example we manually drew it in order to leave a wide "unknown" region. (f) Initialized matte α_0. The center image region has large errors because the foreground and background colors are similar. These errors will be corrected in the subsequent steps using color misalignment cues from the 'x' shaped textures. (g) Estimated foreground color $\mathbf{F}_0$ based on α_0 in (f). Blue indicates an undefined region. (h) Estimated background color $\mathbf{B}_0$ based on α_0 in (f). (i) Foreground color consistency C_{F_0} computed for $\mathbf{F}_0$ in (g). The disparity of (g) around the top center region is 5, which is inconsistent with the true foreground disparity of 0. Therefore, C_{F_0} became large around there. (j) Background color consistency C_{B_0} computed for $\mathbf{B}_0$ in (h). The disparity of (h) around the bottom center region is 0, which is inconsistent with the true background disparity of 5. Therefore, C_{B_0} became large around there. (k) Updated matte. The alpha values were pulled toward 0 where C_{F_0} in (i) is large, and toward 1 where C_{B_0} in (j) is large. (l) Final matte after convergence, which is close to the ground truth matte (c).

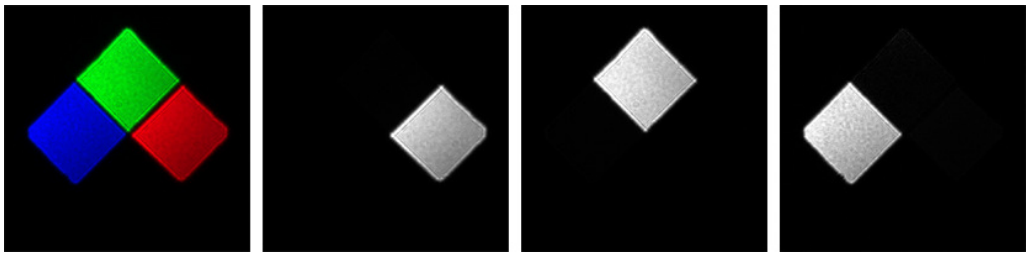
3.5 Camera Hardware Implementation

For a prototype camera lens, we cut out a disc with a triple-square-shaped hole from a piece of black cardboard, glued color filters (Fujifilter SC-58, BPB-53, and BPB-45) to it, and attached it immediately in front of the aperture diaphragm of a Canon EF 50mm f/1.8 II lens (see Fig. 3.7). This fabrication was done in a few hours with a box cutter and a micro screwdriver. We used an unmodified Canon EOS40D DSLR as a camera body. Fig. 3.8 shows the point-spread function (PSF) of the prototype lens, which is an image of a defocused point light source. The square shape of each filter is observed mostly as-is, with only slightly rounded corners at the horizontal extremities due to occlusion by the lens housing. Fig. 3.8 also shows that the three color bands are well separated. We achieved this by applying a linear transform to RGB sensor response so as to minimize crosstalk between the aperture filters and the image sensor (see Appendix C for details).



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

Figure 3.7: Prototyping process of a color-filtered aperture lens. (a) Original Canon EF 50mm f/1.8 II lens. (b) The aperture part of the disassembled lens. (c) Color filters are attached to the aperture. (d) The lens after reassembly.



Color Red Green Blue

Figure 3.8: Point-spread function of our lens and its RGB components. The positions of the R and B regions are opposite to those in Figs. 3.1 and 3.7(d), as the viewpoint is behind the lens in this figure.

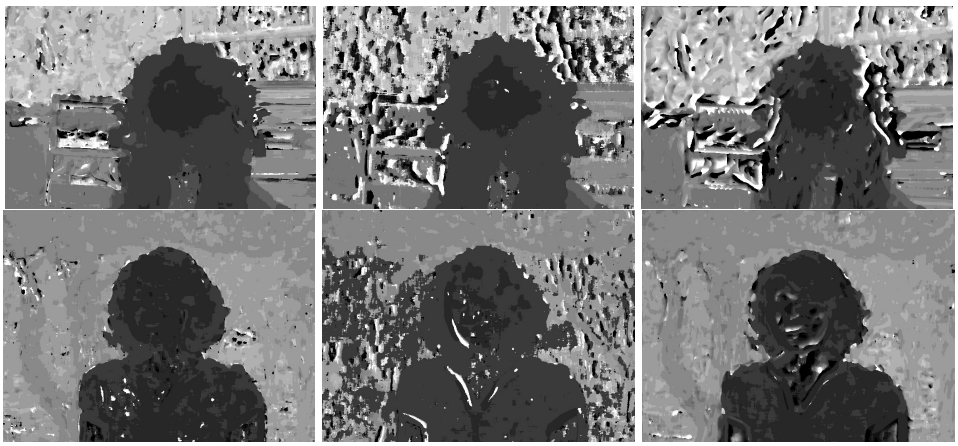
To align the RGB regions with the image sensor axes, manual adjustment was sufficient. Once this is done, pixel disparities will always align with the X and Y axes of captured images, requiring no further calibration and rectification at capture time or during post-processing.

3.6 Results

For all of the results shown below, we set the local window size to 15×15 pixels, $\kappa_s = 0.1$, $\kappa_\alpha = 0.01$, and $\kappa_c = 0.02$. The matte optimization converged in about 20 iterations. The computation time for a 720×480 image was 10 sec. for blur estimation, and 2 min. for matting on an Intel Pentium 4 3.2GHz with 2GB RAM. We used *summed-area tables* [26] to rapidly compute covariance matrices in local windows.

3.6.1 Blur Estimation Results

We first demonstrate the performance of our RGB correspondence measure used for disparity/blur estimation. We compare our disparity estimation results with those of the previous methods [6, 21] in Figs. 3.9(a-c). In order to reveal raw performance, we show local window estimates without graph-cut optimization. As Amari and Adelson's method [6] relies on high-pass filtering, it mostly failed to detect disparities of the defocused scene backgrounds (Fig. 3.9(b)). Chang *et al.*'s method [21] performed better, but it handled color edges and gradations poorly, presumably because these may not be accounted for by a single intensity normalization factor within a local window (Fig. 3.9(c)). Our method produced better results than the previous methods (Fig. 3.9(a)).



(a) (b) (c)

Figure 3.9: Comparison of correspondence measures between the RGB planes (local estimate). Larger intensities indicate larger disparities. Top row: results for the toy dog image in Fig. 3.3. Bottom row: results for the woman image in Fig. 3.17. (a) Our method. (b) Amari and Adelson [6]. (c) Chang *et al.* [21].

We also compare our results with a mutual information-based method by Kim *et al.* [41], which can handle broad types of intensity relationships between images. Since their method is coupled with iterative graph-cut optimization, our results after (single) graph-

cut optimization are also shown in Fig. 3.10(a). Because their correspondence measure is defined for two images, we take the average of the values for the three pairs of RGB planes (RG, GB, and BR). Their method performed well in view of the fact that it does not assume *a priori* knowledge of the intensity relationships. However, some portions of the foreground objects were not detected (Fig. 3.10(b)).

(a) (b)

Figure 3.10: Comparison of correspondence measures between the RGB planes (global estimate). Larger intensities indicate larger disparities. Top row: results for the toy dog image in Fig. 3.3. Bottom row: results for the woman image in Fig. 3.17. (a) Our method (after graph-cut optimization). (b) Kim *et al.* [41].

3.6.2 Matting Results

Next we show our matting results. Fig. 3.11(a) shows the extracted matte for the toy dog image in Fig. 3.3. The hairy silhouette was extracted successfully. We can use this matte to refine the boundary of the foreground and background regions in the disparity map as shown in Fig. 3.11(b), by compositing the foreground and background disparity maps shown in Figs. 3.5(b, c). In Fig. 3.12, we applied existing natural image matting methods, Closed-Form Matting [50] and Robust Matting [93], with the trimap given by our method. These results are not for comparison because the previous methods are designed for color-aligned images, but the matte errors seen in Fig. 3.12 are indicative of the importance of our color consistency measure in suppressing them.

For proper comparison, we used a ground truth matte shown in Fig. 3.13(a) obtained by capturing an object in front of a simple background and by using *Bayesian Matting* [23], followed by manual touch-up where needed. We created a synthetic "natural" image as shown in Fig. 3.13(b) by compositing the object over a new background image. We also created its color-misaligned version as shown in Fig. 3.13(c) by shifting the back-

(a) (b)

Figure 3.11: (a) Extracted matte for the toy dog image in Fig. 3.3. (b) Refined disparity map. Compare this with Fig. 3.10(a) top.

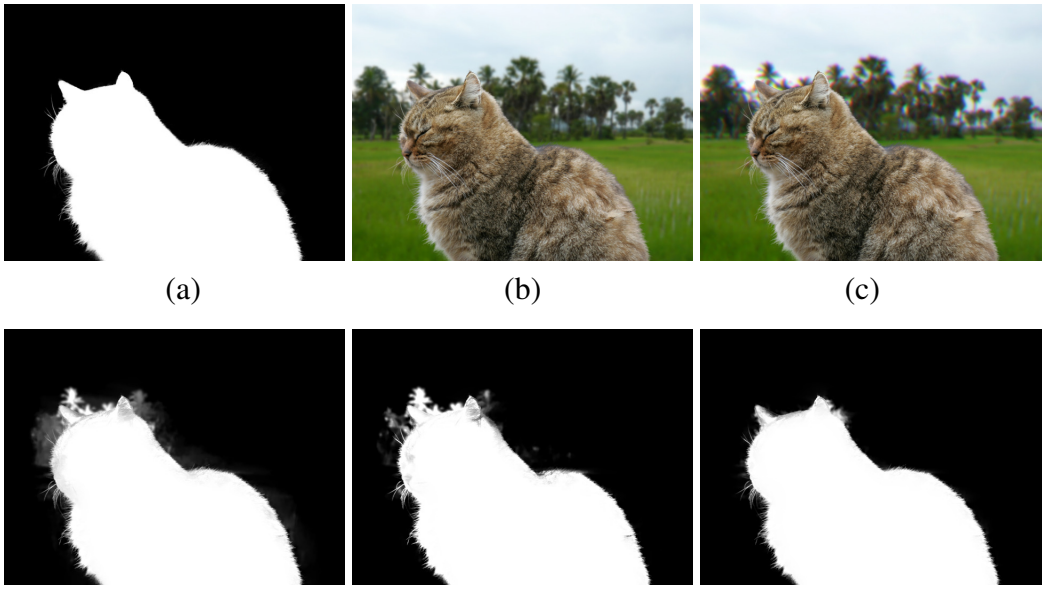
(a) (b)

Figure 3.12: Results of existing natural image matting methods. (a) Closed-Form Matting [50]. (b) Robust Matting [93].

ground color by 3 pixels before composition. We applied the previous methods to the color-aligned synthetic image, and our method to the color-misaligned one. Though not perfect, our method produced a better matte as shown in Figs. 3.13(d-f). For quantitative evaluation, we conducted the same experiment for five more examples shown in Fig. 3.14, and we computed the mean squared errors (MSE) against the ground truth mattes, which we plotted in Fig. 3.15. Our method reduced MSE values by 33-86% compared to the other two methods.

3.6.3 Defocus Deblurring Results

We show defocus blur removal results based on the estimated blur and matte. First, we restore a color-aligned image as shown in Fig. 3.16(b), by re-compositing the foreground and background colors after canceling their color misalignment based on the foreground and background disparity maps. Specifically, if the foreground disparity at (x,y) is d, the aligned foreground color at that point is restored as: $(F_r(x{+}d,y),F_g(x,y{-}d),F_b(x{-}d,y))$. We then restore an all-in-focus background color by stitching deconvolved images based on the estimated blur size as illustrated in Fig. 2.2, and compose it with the extracted foreground as shown in Fig. 3.16(c). In addition, by reblurring the deblurred image differently as was done in Chapter 2, we can synthetically refocus the image as shown in



(a) (b) (c)

(d) (e) (f)

Figure 3.13: Comparison using a ground truth matte. (a) Ground truth matte. (b) Synthetic natural image. (c) Color-misaligned version of (b). (d) Closed-Form Matting (applied to (b)). (e) Robust Matting (applied to (b)). (f) Our method (applied to (c)).

Fig. 3.16(d). In the presence of hairy foreground objects, alpha mattes are indispensable for the above operations to give plausible results. Fig. 3.17 shows another deblurring result for an outdoor photograph.

3.6.4 Additional Results

Fig. 3.18 shows additional color misalignment cancellation results.

Fig. 3.19 shows an example where a portion of the foreground object (the hip of the sheep) is slightly out of the depth-of-field of the lens, violating the assumption that $\alpha(x,y)$ is aligned between the RGB planes in Eq. 3.2. However, degradation of the extracted matte around the region was small, as shown in Fig. 3.19(d).

The extracted mattes can also be used for composing the foreground objects onto different background images as shown in Fig. 3.20.

Due to the use of color filters, our method cannot handle entirely pure-red objects (as well as entirely pure-green or pure-blue objects). But this does not mean that objects must not be mostly red. To prove that our method works for mostly red objects, we applied our method to a photo of a red chair. This chair is mostly red but not entirely pure-red because: 1) it is not entirely red as it has a silver frame, 2) it is not purely red as this orangish red

Figure 3.14: Synthetic natural images and their ground truth mattes.

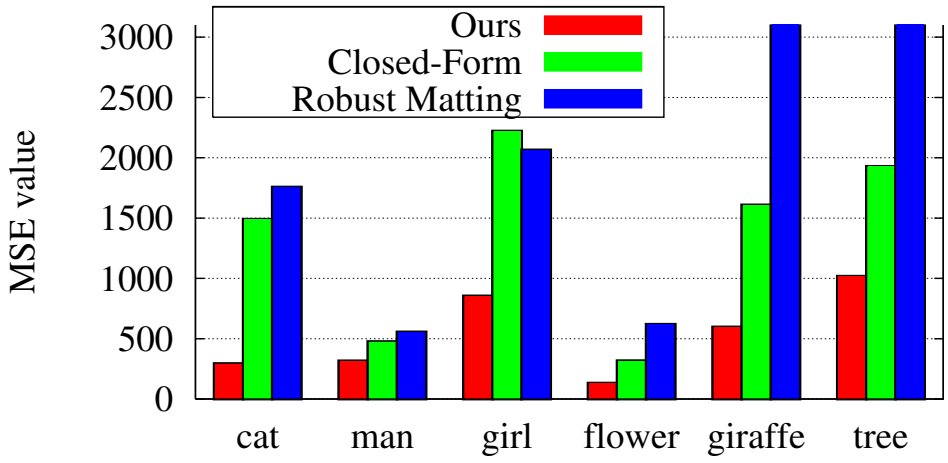


Figure 3.15: MSE values of the mattes produced by our method and the previous methods for the images shown in Figs. 3.13 and 3.14.

has a sufficient green component. Our method indeed worked, as shown in Fig. 3.21.

Fig. 3.22 shows additional results for a photograph of two big names in the computer graphics community.

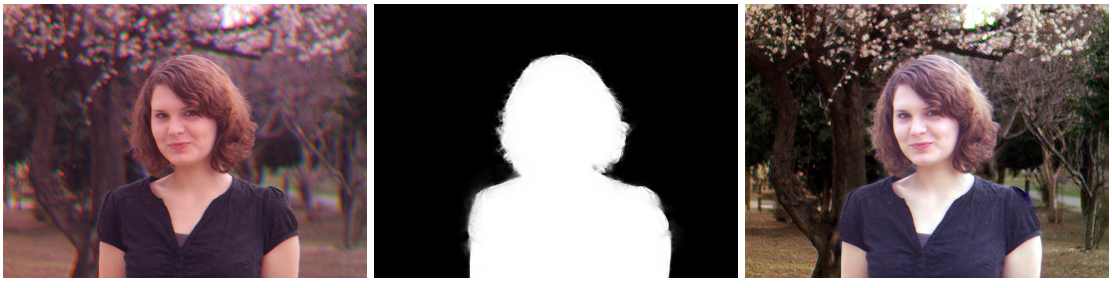
Using the rapid shooting capability of the camera, we also performed *video matting* as shown in Fig. 3.23. We applied our method to each frame in the video independently without taking into account temporal coherence.

3.7 Summary

This chapter has presented a method for estimating defocus blur sizes by placing red, green, and blue color filters in a camera lens aperture. By dividing the aperture into three

Figure 3.16: Defocus blur removal and refocusing. The right column shows closeup views of the left one. (a) Captured image. The colors are misaligned. (b) Color misalignment canceled. (c) Defocus blur removed. (d) Refocused.

regions through which only light in one of the RGB color bands can pass, we can acquire three shifted views of a scene in the RGB planes of a captured image in a single exposure. This allows us to take stereo correspondence between the RGB planes to estimate the scene depth, which is directly related to the defocus blur size. We have also presented a matting method for extracting an in-focus foreground object so that the unblurred part of the scene can remain unaffected by the deconvolution process. Our method only modifies a camera lens with off-the-shelf color filters, and utilizes the RGB planes of the image sensor of a conventional camera body to capture multi-view images in a single exposure. We have proposed an effective correspondence measure between the RGB planes, and a method for employing color misalignment cues to improve the matte. By showing results for outdoor scenes and/or hairy foreground objects, we have demonstrated the portability



(a) (b) (c)

(d) (e)

Figure 3.17: Defocus blur removal for an outdoor scene. (a) Captured image. (b) Extracted matte. The estimated blur/disparity map is shown in Fig. 3.10(a) bottom. (c) Deblurred image. The reddish color shade seen in (a) due to the aperture filters is also corrected. (d) Closeup of (a). (e) Closeup of (c).

(a) (b) (c)

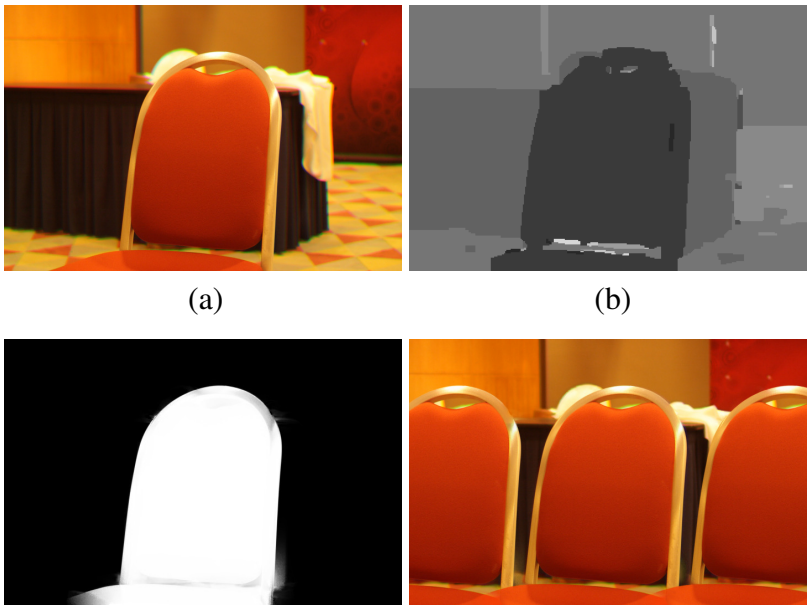
Figure 3.18: More color misalignment cancellation results. (a) Restored images. (b) Closeups of (a). (c) Closeups of the original.

of our device and the effectiveness of our method in defocus blur removal, as well as in post-exposure image editing including digital refocusing and composition over different backgrounds.

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

Figure 3.19: Results for a sheep. (a) Captured image. (b) Blur/disparity map. (c) Matte. (d) Closeup from the red rectangle in (c).

Figure 3.20: Composition onto different background images.



(a) (b)

(c) (d)

Figure 3.21: Results for a red chair. (a) Captured image. (b) Estimated depth. (c) Extracted matte. (d) Composite image, where the extracted chair is added back to the input image multiple times.

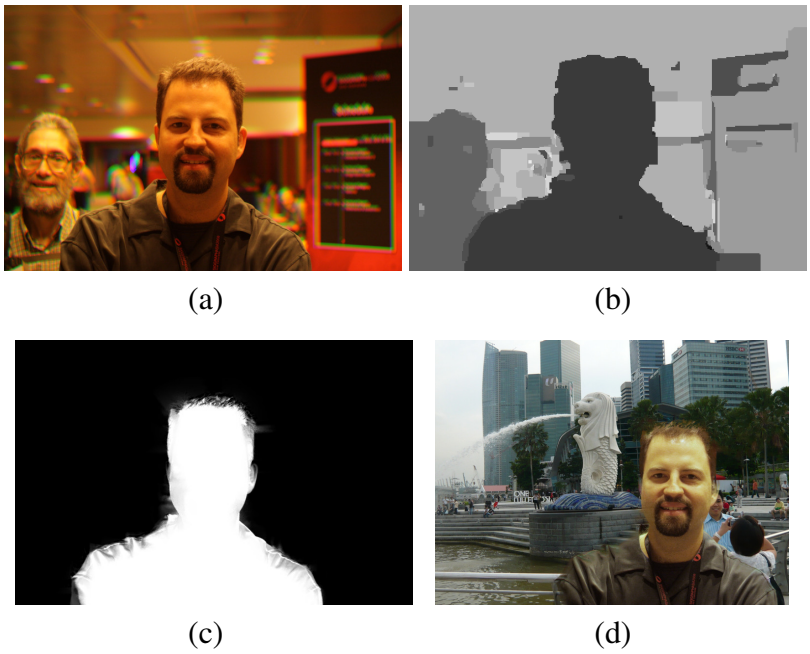


Figure 3.22: Results for two big names in the computer graphics community. (a) Captured image. (b) Estimated depth. (c) Extracted matte. (d) Composite image.

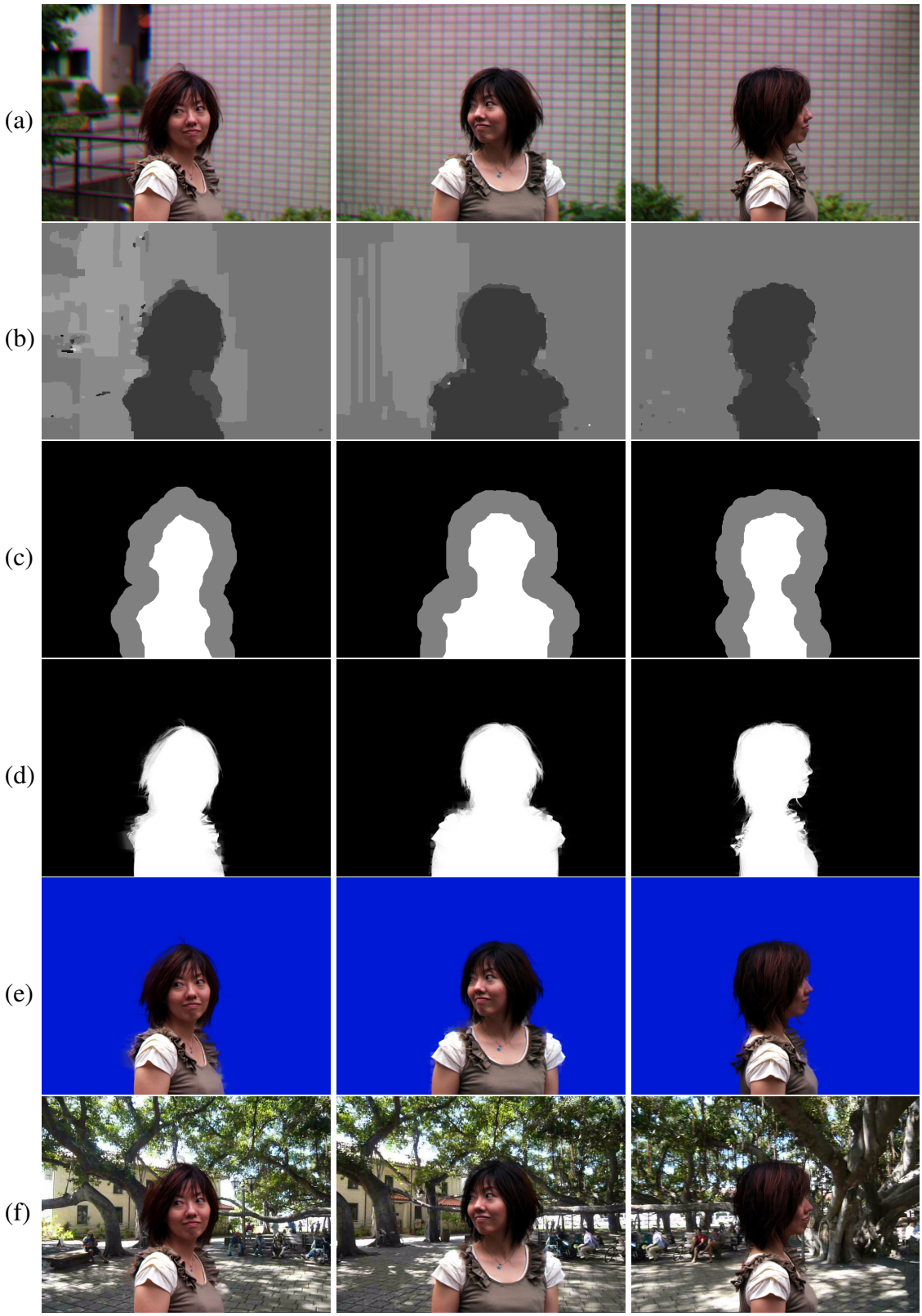


Figure 3.23: Some frames from a video matting result, from left to right. (a) Captured images. (b) Depth maps. (c) Trimaps. (d) Alpha mattes. (e) Composition over a blue background. (f) Composition over another video.

Chapter 4

Motion Blur Removal using Circular Sensor Motion

So far, this dissertation has dealt with defocus blur. This chapter focuses on motion blur, and describes a method for motion deblurring exploiting a modified image capture process which involves translation of the camera image sensor during exposure.

4.1 Introduction

Motion blur, while being useful for depicting object motion in still images, often spoils photographs by losing image sharpness. The frequency band that can be recovered by deconvolution easily becomes narrow for fast object motion as high frequencies are severely attenuated and virtually lost.

Follow shot, a photographing technique in which a photographer pans a camera to track an object during exposure, can capture sharp images of a moving object as if it were static. However, there are cases where follow shot is not effective: 1) when object motion is unpredictable; 2) when there are multiple objects with different motion. This is because follow shot favors particular motion that a photographer has chosen to track, as much as a static camera favors "motion" at the speed of zero (i.e., static objects): objects moving differently from favored motion degrade.

This chapter explores camera sensor motion during exposure that treats a wide range of in-plane linear object motion in any direction and up to some predetermined speed. That is, although no object may be photographed sharply at capture time, differently moving objects can be deconvolved with similar quality. This idea is inspired by Levin *et al*. [51], who proved that constantly accelerating 1D sensor motion can render motion

blur *invariant* to 1D linear object motion (e.g., horizontal motion), and showed that this sensor motion evenly distributes the fixed frequency "budget" to different object speeds. We intend to extend their budgeting argument to 2D (i.e., in-plane) linear object motion by sacrificing motion-invariance. We propose to translate a camera sensor circularly about the optical axis, and we analyze the frequency characteristics of circular sensor motion in relation to linear object motion.

By losing motion-invariance, we inevitably reintroduce two issues inherent to the classical motion deblurring problem, which [51] resolved for 1D motion. Firstly, we need to estimate a point-spread function (PSF) of motion blur as it depends on object motion. Fortunately, for a set of PSFs resulting from circular sensor motion, deconvolution by wrong PSFs causes ringing artifacts, which is not always the case for other image capture strategies. This allows us to take a simple hypothesis testing approach for PSF estimation. Secondly, we need to segment an image into regions with different motion in order for deconvolution to be applicable. This is still a challenging problem which has only been partially addressed by state-of-the-art methods (e.g., [47] for 1D motion), and this chapter assumes user-specified motion segmentation.

4.2 Circular Image Integration

Fig. 4.1(a) shows the proposed motion of a camera image sensor. We *translate* the sensor along a circle perpendicular to the optical axis while keeping its orientation. We use the phrase "circular motion" to emphasize that we do not *rotate* the sensor itself.

During exposure time $t \in [-T, +T]$, the sensor undergoes one revolution with constant angular velocity $\omega = \pi/T$. Letting the radius of circular motion be R, the sensor moves along the circle with constant speed $R\omega$, which corresponds to the target object speed S *in the image space*. The corresponding object speed in the world space (i.e., actual speed in a scene) is determined by the camera optics and the distance to the object from the camera. Given exposure time $2T$ and the target object speed S, the appropriate radius is therefore $R = ST/\pi$. Taking an xy plane on the sensor, the sensor motion goes through a spiral in the xyt space-time volume as shown in red in Fig. 4.1(b).

Fig. 4.2 shows simulated motion blur PSFs and their power spectra of various object motions observed from a static camera, the coded exposure camera [75], the motion-invariant camera [51], and our circular motion camera. As can be seen in the figure, while the power spectrum for a static object observed from a static camera is perfectly broadband, those for moving objects become quickly narrowband as the object speed increases. The coded exposure camera makes power spectra broadband at the cost of losing

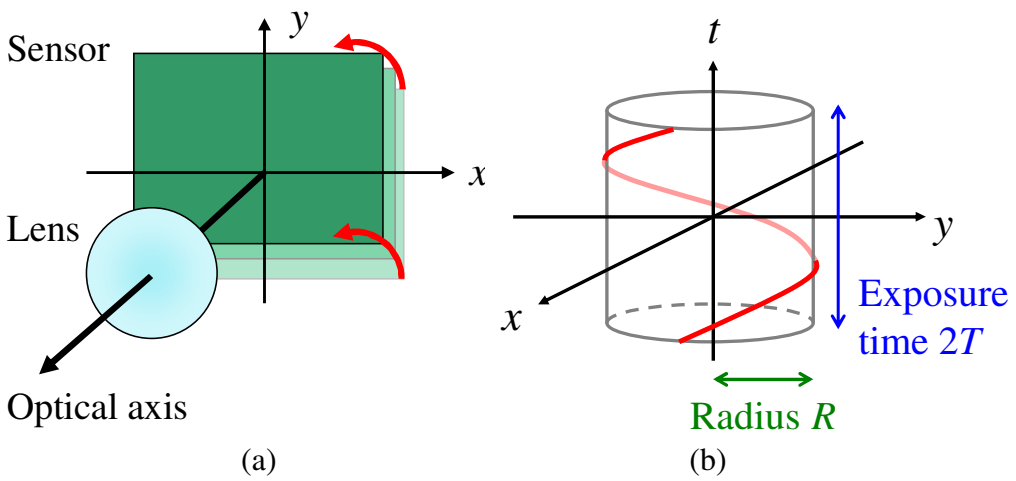


Figure 4.1: Circular sensor motion. (a) The sensor is translated circularly about the optical axis. (b) Trajectory of the sensor motion in the space-time volume (shown in red).

light blocked by the shutter, but the tendency of bandwidth narrowing for faster motion remains. The motion-invariant camera produces similarly broadband power spectra for horizontal motions (they are not completely identical due to the *tail clipping* effect [51]), but vertical frequencies are sacrificed as motion direction deviates from horizontal. The circular motion camera produces power spectra that extend to high frequency regions in all cases. Although they have striped frequency zeros, these zeros facilitate PSF estimation as described in Sec. 4.4.

We evaluated the quality of these PSFs in a similar way to [103] by simulating motion blur for a set of 12 natural images, and by measuring the mean squared errors (MSE) between the deconvolved images (using pseudo-inverse deconvolution) and the original unblurred images. Fig. 4.3 plots the deconvolution noise increase in decibels as $10\log_{10}(MSE/\sigma^2)$, where we assumed noise corruption for motion blur to be Gaussian of standard deviation $\sigma = 10^{-3}$ for [0, 1] pixel values. As shown in the plots, the motion-invariant camera shows excellent constant performance for horizontal motion up to the target speed S, but for other motion directions, deconvolution noise increases for faster object motion. The coded exposure camera and ours do not have such directional dependence. The coded exposure camera performs almost as perfectly as a static camera for static objects, and deconvolution noise gradually increases for faster object motion. The circular motion camera also maintains stable performance for all directions up to and slightly beyond S. It moderately favors the target object speed S, where it has lower deconvolution noise than the other cameras except for the motion-invariant camera for horizontal object motion. The downside of our image capture strategy is the increased deconvolution noise for static objects.

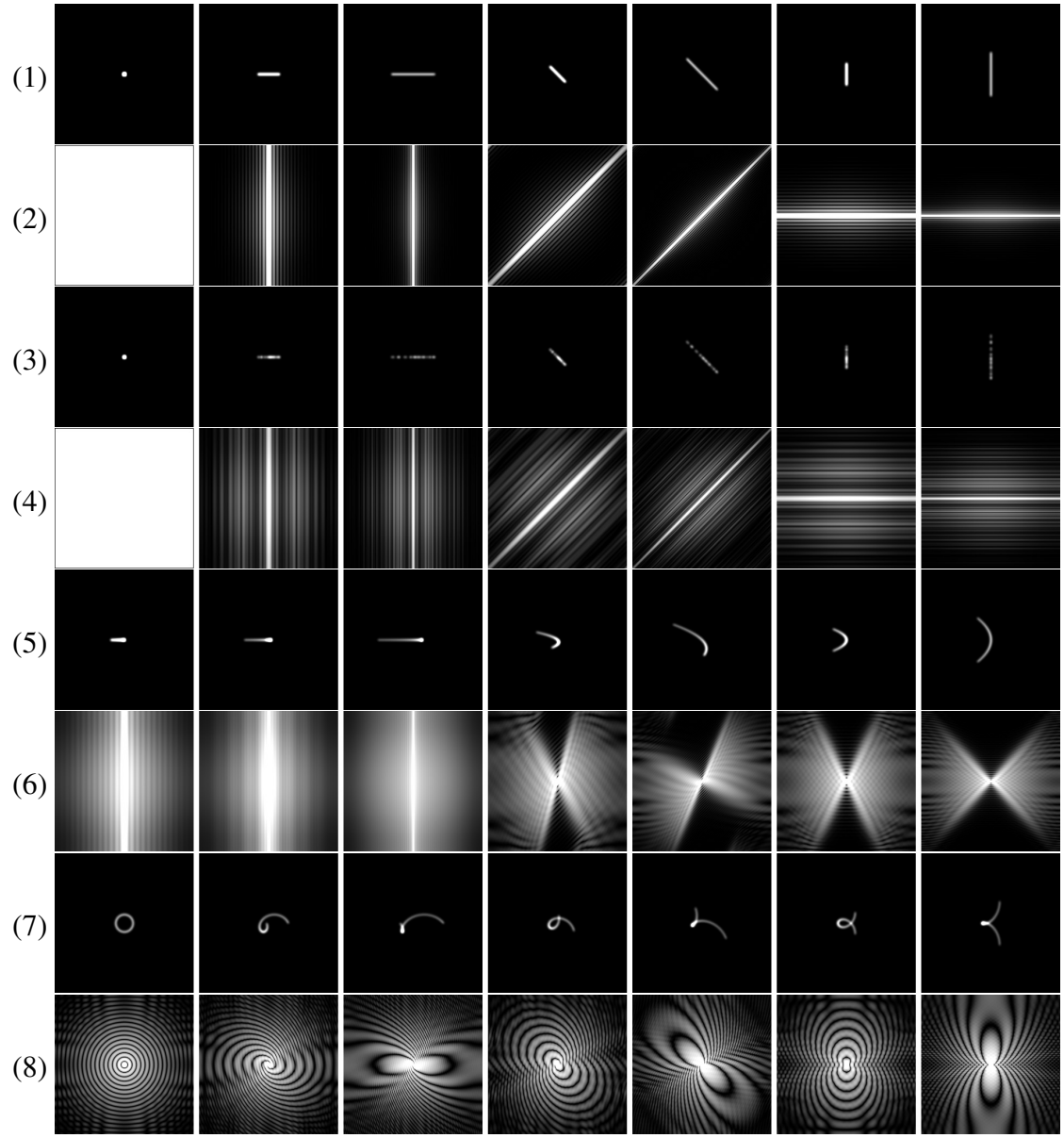


Figure 4.2: Motion blur PSFs and their corresponding log power spectra. Rows: (1) PSFs and (2) power spectra for a static camera. (3)(4) For the coded exposure camera. (5)(6) For the motion-invariant camera. (7)(8) For the circular motion camera. Columns: (a) Static object. (b)(c) Horizontal object motion at different speeds. (d)(e) Oblique object motion. (f)(g) Vertical object motion.

Fig. 4.4 demonstrates the above-mentioned trade-offs, showing synthetically motion blurred objects and their deblurred images.

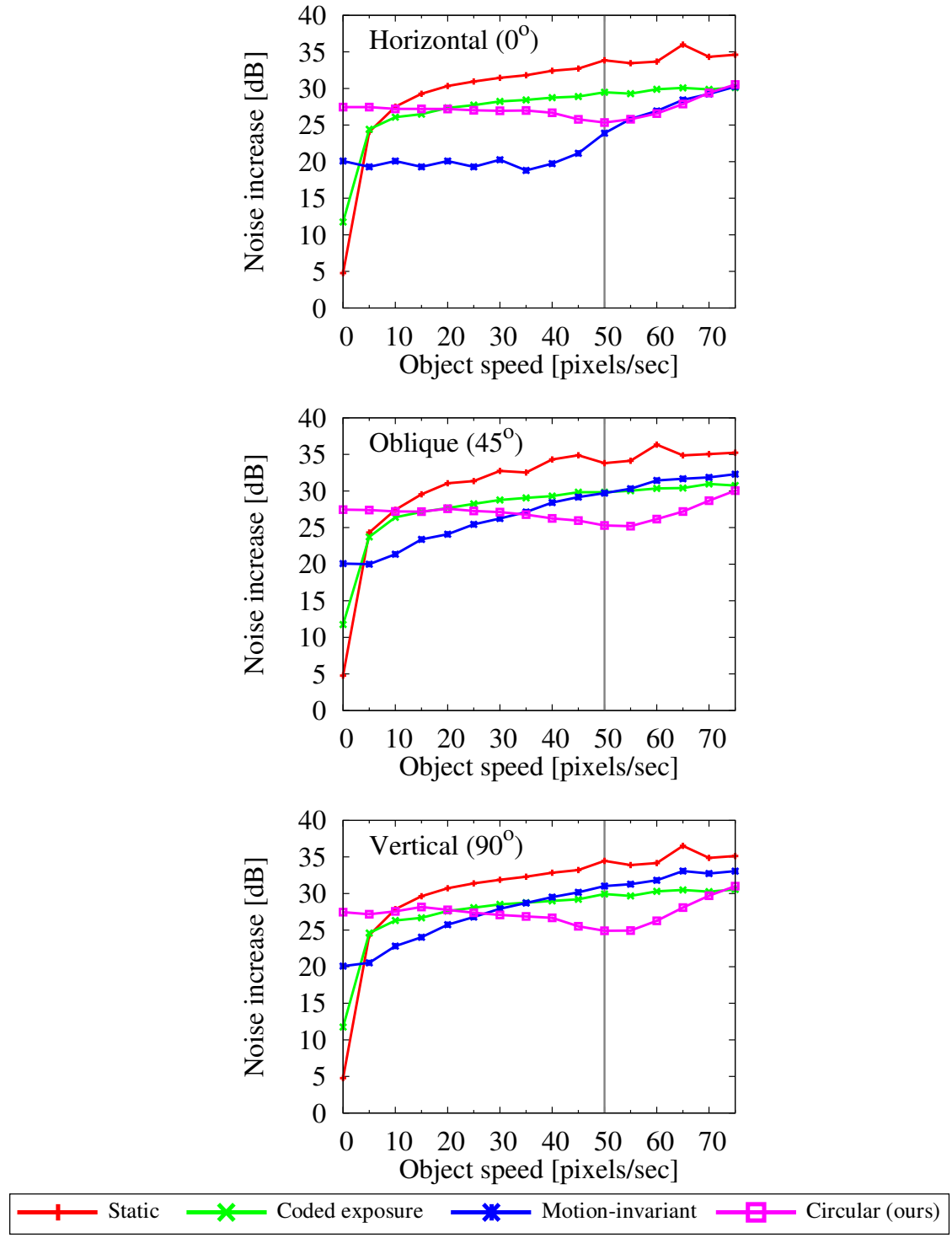


Figure 4.3: Plots of deconvolution noise increase for different object speeds and directions. Pseudo-inverse deconvolution is used. The exposure time is 1 sec for all the cameras. The vertical gray lines indicate the target (maximum) object speed $S = 50$ pixels/sec for the motion-invariant camera and ours. The length 50 code containing 25 '1's [5] was used for the coded exposure camera (half the light level).

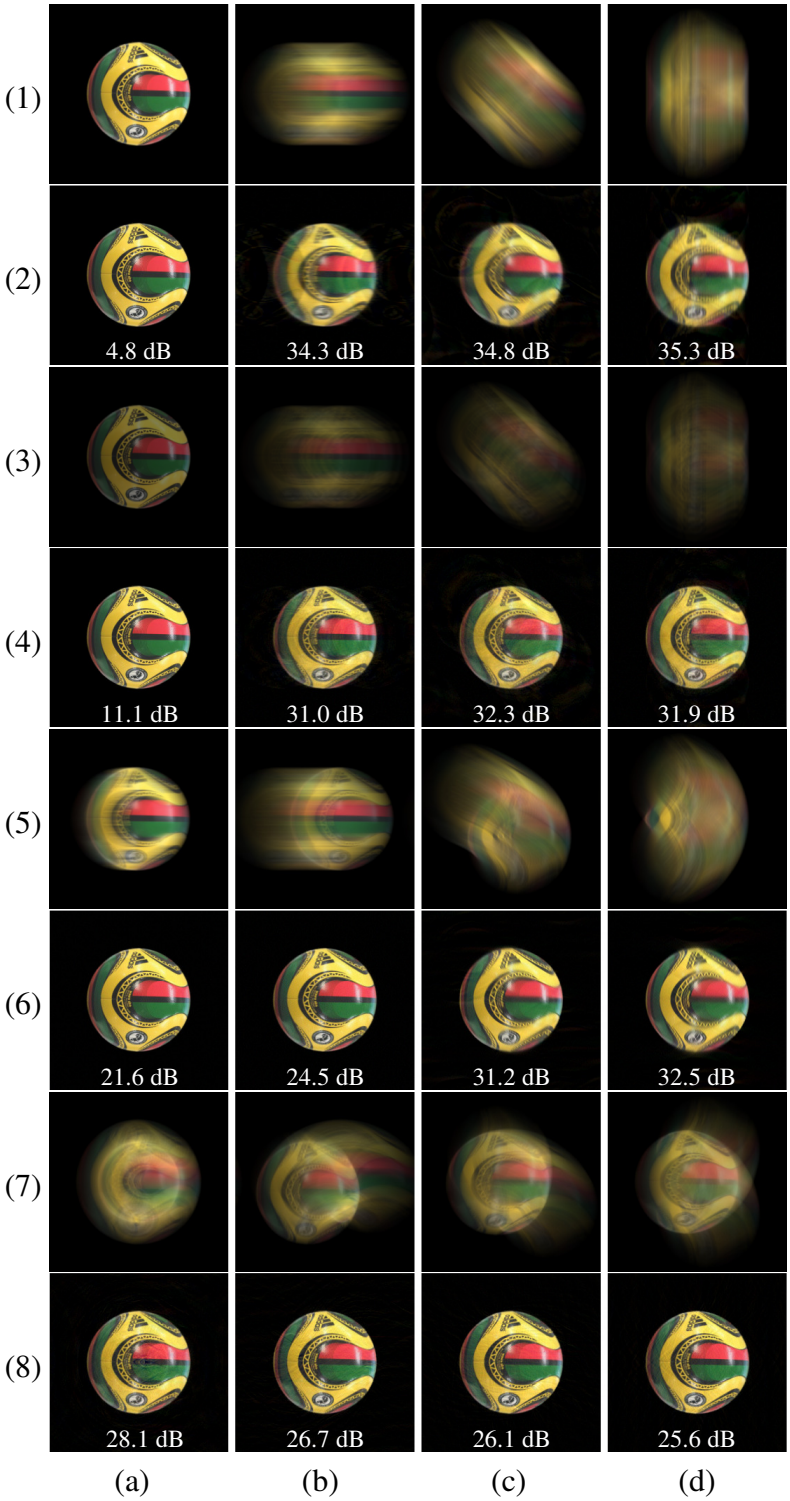


Figure 4.4: Simulated motion blurred images of a colorful soccer ball and their pseudo-inverse deconvolution results. The values indicate deconvolution noise increase. Rows: (1) Blurred and (2) deblurred images for a static camera. (3)(4) Coded exposure camera. (5)(6) Motion-invariant camera. (7)(8) Circular motion camera. Columns: (a) Static object. (b)(c)(d) Horizontal, oblique, and vertical object motion at the target speed S.

4.3 Analysis of Circular Sensor Motion

Levin *et al*. [51] proved that constantly accelerating 1D sensor motion (going through a parabola $x = at^2$ in xt space-time) is the only sensor motion that makes PSF invariant to 1D linear object motion. Based on this finding, we can derive the following proposition.

Proposition 1: *There is no sensor motion that makes PSF invariant to 2D linear object motion.*

Proof: Suppose there exists such sensor motion $\mathbf{m}(t) = (m_x(t), m_y(t))$. As it is invariant to 2D linear object motion, for any constant object velocity $\mathbf{v} = (s_x, s_y)$, there must exist c and $\mathbf{d}$ such that

$$\mathbf{m}(t) - \mathbf{v}t = \mathbf{m}(t+c) + \mathbf{d}, \tag{4.1}$$

which means that the object motion only translates the sensor motion path $\mathbf{m}(t)$. Differentiating Eq. 4.1 and rearranging, we have:

$$\frac{\partial \mathbf{m}(t+c)}{\partial t} - \frac{\partial \mathbf{m}(t)}{\partial t} = \mathbf{v}. \tag{4.2}$$

From this equation we can see that both $\partial m_x(t)/\partial t$ and $\partial m_y(t)/\partial t$ are linear functions of t, and therefore $m_x(t)$ and $m_y(t)$ are parabolas. However, letting $m_x(t) = at^2$ and $m_y(t) = bt^2$, Eq. 4.1 cannot be satisfied because

$$m_x(t) - s_x t = a\left(t - \frac{s_x}{2a}\right)^2 - \frac{s_x^2}{4a}, \tag{4.3}$$

$$m_t(t) - s_y t = b\left(t - \frac{s_y}{2b}\right)^2 - \frac{s_y^2}{4b}, \tag{4.4}$$

and c does not exist unless $s_x/2a = s_y/2b$, leading to a contradiction. Q.E.D.

Hence, we must abandon motion-invariance, and we seek to extend Levin *et al*.'s another finding that their sensor motion evenly and nearly optimally distributes the fixed frequency "budget" to different object speeds.

The intuitive explanation for optimality of constant camera acceleration for 1D case is as follows. Fig. 4.5(a) shows the range of speed $[-S, +S]$ that must be taken care of. We can cover the entire range by accelerating a camera beginning at speed $-S$ until it reaches $+S$. The camera tracks every speed at one moment during exposure. By extending to 2D, the range of *velocity* (speed + direction) we must cover becomes a disc as shown in green in Fig. 4.5(b). We are no longer able to fill the entire disc by a finite sensor motion path, and we opt to trace only the circumference of the disc (shown in blue), which can be achieved by moving a sensor circularly. The reasons for doing so are threefold.

1. It makes theoretical analysis easier. Although full frequency analysis of 3D *xyt* space-time is difficult, we were able to draw some insights of frequency characteristics of circular sensor motion.
2. Tracing the circumference alone can be shown to deal with velocity in the interior of the disc fairly well.
3. It makes implementation of camera hardware easier.

As for Reason 2, we have already seen empirically in Fig. 4.3 that the circular sensor motion favors the target speed but this tendency is not too pronounced. To further treat different object speeds evenly, one can consider sampling the interior of the velocity disc by a set of concentric circles. However, this does not bring in significant improvement of PSF power spectra, since the phases of the Fourier transform of multiple circular motions cancel each other when superimposed, resulting in a qualitatively similar set of power spectra to the one shown in the bottom row of Fig. 4.2 (see Appendix D).

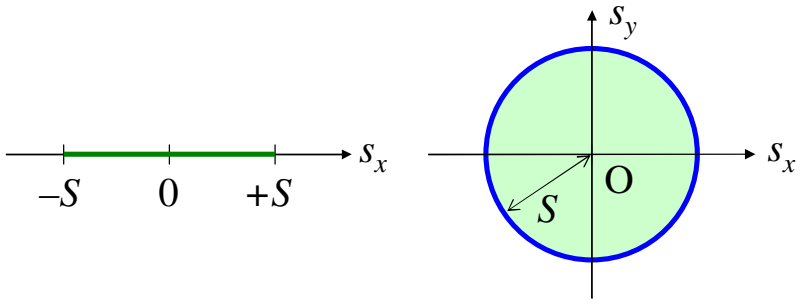


Figure 4.5: The range of velocity (s_x, s_y) that must be covered by sensor motion for (a) 1D case and (b) 2D case (shown in green). We trace only the circumference of the disc (shown in blue).

4.3.1 Frequency Budgeting

Now we review the frequency budgeting argument of [51] for the case of 2D object motion.

We consider a camera path in the *xyt* space-time volume.

$$p(\mathbf{x},t) = \begin{cases} \delta(\mathbf{x}-\mathbf{m}(t)) & \text{for } t \in [-T,+T] \\ 0 & \text{otherwise} \end{cases}, \tag{4.5}$$

where $\mathbf{x} = (x,y)$, $\mathbf{m}(t)$ specifies the camera position at time t, and $\delta(\cdot)$ is a delta function. We would like to consider its 3D Fourier transform, denoted by $\hat{p}$:

$$\hat{p}(\mathbf{f}, f_t) = \int_{\Omega}\int_{-T}^{+T} \delta(\mathbf{x}-\mathbf{m}(t)) e^{-2\pi i(\mathbf{f}\cdot\mathbf{x}+f_t t)} dt d\mathbf{x}, \tag{4.6}$$

where $\mathbf{f} = (f_x, f_y)$ is a 2D spatial frequency, f_t is a temporal frequency, and Ω spans the entire xy plane.

It can be shown that the 2D Fourier transform of a motion blur PSF for object velocity $\mathbf{v}$ is a 2D slice of $\hat{p}(\mathbf{f}, f_t)$ along the plane of $f_t = -\mathbf{v}\cdot\mathbf{f} = -s_x f_x - s_y f_y$ (Fourier projection-slice theorem [18], see Appendix E). Therefore, given a maximum speed S, the volume in the 3D $f_x f_y f_t$ frequency domain that these slices can pass through is confined to the outside of the cone as $|f_t| \leq S|\mathbf{f}|$, as shown in blue in Fig. 4.6(a). We would like $|\hat{p}(\mathbf{f}, f_t)|$ to have as large value as possible within this volume, so that motion blur PSFs up to S have large power spectra. However, the budget is exactly $2T$ along each vertical line $\mathbf{f} = \mathbf{c}$ (the line shown in red and green in Fig. 4.6(a)) for any given spatial frequency $\mathbf{c}$: i.e., $\int |\hat{p}(\mathbf{c}, f_t)|^2 df_t = 2T$ (see Appendix F).

To assign the $2T$ budget so that any 2D linear object motion below S has the same amount of PSF spectral power, we consider the following two criteria.

Effectiveness: The budget should be assigned as much as possible within the line segment of $f_t \in [-S|\mathbf{c}|, +S|\mathbf{c}|]$ which is shown in red in Fig. 4.6(a). In other words, we would like to avoid assigning the budget to the other portions of the line (shown in green in Fig. 4.6(a)) as they correspond to object speeds beyond S and the budget will be wasted. Because the budget is exactly $2T$ unless we close the shutter during exposure, less assignment to some portion means more assignment to the other.

Uniformity: The budget should be distributed evenly across the line segment, so that every object motion PSF has an equal amount of spectral power.

Therefore, optimal assignment in which both effectiveness and uniformity are perfect gives $T/S|\mathbf{c}|$ to each point on the line segment.

4.3.2 Spectrum of Circular Sensor Motion

We take the 3D Fourier transform of the circular sensor motion $\mathbf{m}(t) = (R\cos\omega t, R\sin\omega t)$, a spiral in the xyt space-time as shown in Fig. 4.1(b). By integrating Eq. 4.6 with respect to t, we obtain:

$$\hat{p}(\mathbf{f}, f_t) = \int_\Omega \left(\frac{\delta(|\mathbf{x}| - R)}{R\omega} e^{-2\pi i f_t \mathbf{m}^{-1}(\mathbf{x})} \right) e^{-2\pi i \mathbf{f}\cdot\mathbf{x}} d\mathbf{x}, \tag{4.7}$$

since the integrand is non-zero only at $|\mathbf{x}| = R$ and at $t = \mathbf{m}^{-1}(\mathbf{x})$. Jacobian $|d\mathbf{m}(t)/dt| = R\omega$ is introduced in the denominator. By using polar coordinates as $x = r\cos\theta$ and $y =$

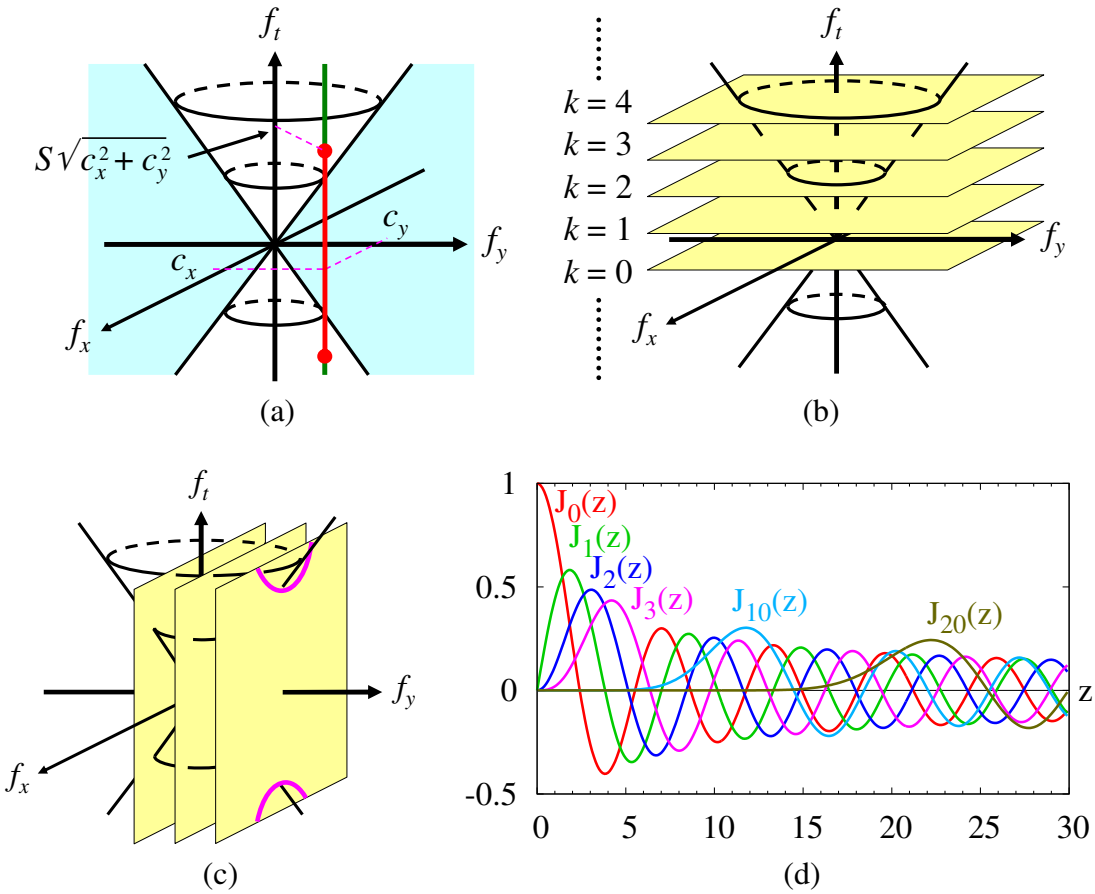


Figure 4.6: (a) The cone defining the volume (shown in blue) whose slices passing through the origin correspond to the power spectra of motion blur PSFs below the speed S. (b) Discrete f_t slices. (c) f_y slices. The hyperbolic intersections with the cone are shown in purple. (d) Plots of Bessel functions $J_k(z)$ of the first kind for some k, which correspond to the slices in (b).

$r\sin\theta$,

$$\hat{p}(\mathbf{f}, f_t) = \int_\Omega \left(\frac{\delta(r-R)}{R\omega} e^{-2\pi i f_t \theta/\omega} \right) e^{-2\pi i \mathbf{f}\cdot\mathbf{x}} d\mathbf{x}. \tag{4.8}$$

This is a hard-to-integrate expression, but we can proceed if we focus on a set of discrete f_t slices where $k = 2\pi f_t/\omega$ is an integer as shown in Fig. 4.6(b), as (see Appendix G):

$$|\hat{p}(\mathbf{f}, f_t)|^2 = 4T^2 J_k^2(2\pi R|\mathbf{f}|), \tag{4.9}$$

where $J_k(z)$ is the k-th order Bessel function of the first kind [94, 62], which is plotted for some k in Fig. 4.6(d).

We show the *effectiveness* and *uniformity* of this distribution as described in Sec. 4.3.1. For effectiveness, we show $|\hat{p}(\mathbf{f}, f_t)|^2$ is small inside the cone $|f_t| \geq S|\mathbf{f}|$, shown in white in Fig. 4.6(a). By simple algebraic manipulation, we have $2\pi R|\mathbf{f}| < k$ inside the cone. As can be observed in Fig. 4.6(d) particularly clearly for $k = 10$ and 20, Bessel functions

$J_k(z)$ start from zero at the origin (except for $k = 0$), and remain small until coming close to the first maximum value, which is known to be around $z = k + 0.808618k^{1/3} > k$ [94]. Therefore, $J_k(z)$ is small for $z < k$, which means $|\hat{p}(\mathbf{f}, f_t)|^2$ is small inside the cone.

Next, we show the uniformity of the distribution. For sufficiently large $z \gg k^2$, the Bessel function can be approximated as

$$J_k(z) \approx \sqrt{\frac{2}{\pi z}} \cos\left(z - \frac{k\pi}{2} - \frac{\pi}{4}\right). \tag{4.10}$$

Using this approximation, Eq. 4.9 can be written as:

$$|\hat{p}(\mathbf{f}, f_t)|^2 \approx \frac{4}{\pi} \frac{T}{S|\mathbf{f}|} \cos^2\left(2\pi R|\mathbf{f}| - \frac{k\pi}{2} - \frac{\pi}{4}\right). \tag{4.11}$$

This equation indicates that, at any given spatial frequency $\mathbf{f}$ which is sufficiently large, $|\hat{p}(\mathbf{f}, f_t)|^2$ is a sinusoidal wave with an amplitude of $(4/\pi)(T/S|\mathbf{f}|)$ irrespective of temporal frequency f_t. Hence, although undulating, the distribution is uniform along the f_t direction on an average. The amplitude is greater than the optimal assignment $T/S|\mathbf{f}|$ as described in Sec. 4.3.1, and by averaging the cosine undulation in Eq. 4.11, we can see that the assigned frequency power is $(2/\pi)(T/S|\mathbf{f}|)$ on an average, meaning that the circular sensor motion achieves $2/\pi$ (about 64%) of the optimal assignment (it achieves more around the target speed).

To verify the above argument, we show a numerically computed power spectrum of a spiral in Fig. 4.7 by three f_y slices as shown in Fig. 4.6(c), along with the power spectra of the other camera paths. The motion-invariant camera nearly optimally assigns the budget for the $f_y = 0$ slice corresponding to horizontal object motion, but it fails to deliver the budget uniformly for other cases. Our circular motion camera distributes the budget mostly evenly within the volume of interest, with condensed power around the cone surface corresponding to the maximum value of Bessel functions, which results in a tendency to favor the target speed.

4.4 Motion Blur Estimation

As shown in the bottom row of Fig. 4.2, the power spectra of PSFs resulting from circular sensor motion have different frequency zeros depending on object motion, serving as cues for PSF estimation [48]. As a result, deconvolution with wrong PSFs will result in ringing artifacts as shown in Fig. 4.8, which we detect by the following equation:

$$E(\mathbf{v}) = \frac{1}{N} \sum_j \left[\log q\left(\frac{\partial d_j(\mathbf{v})}{\partial x}\right) + \log q\left(\frac{\partial d_j(\mathbf{v})}{\partial y}\right) \right], \tag{4.12}$$

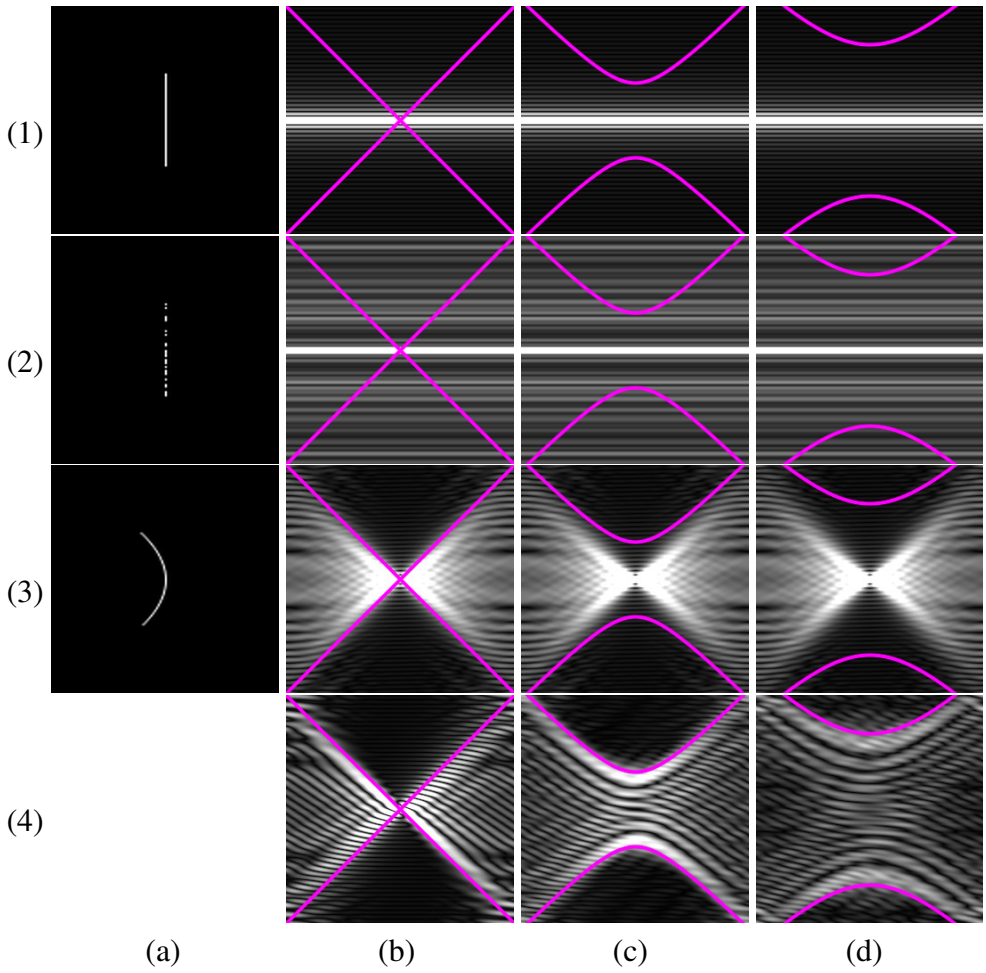


Figure 4.7: Camera paths in the space-time and 2D slices of their 3D log power spectra. Purple curves show the intersections with the cone of target speed S. Rows: (1) Static camera. (2) The coded exposure camera. (3) The motion-invariant camera. (4) The circular motion camera. Columns: (a) Camera path in the xt space-time. See Fig. 4.1(b) for the circular sensor motion path. (b) Slice at $f_y = 0$. (c)(d) Slices off the $f_x f_t$ plane ($f_y \neq 0$).

where N is the number of pixels, j is a pixel index, $d(\mathbf{v})$ denotes a deconvolved image with pseudo-inverse filtering using the PSF corresponding to object motion $\mathbf{v}$, and $q(\cdot)$ is a sparseness prior for natural image derivatives learned from sample images. Images with ringing artifacts have many large derivative values inconsistently with the prior, making Eq. 4.12 small. We search all possible (discretized) object motion directions and speeds up to $1.5S$, and pick the motion $\mathbf{v}$ (equivalently the PSF) that gives the largest value for Eq. 4.12 as a true PSF.

As shown in Fig. 4.8, the simple hypothesis testing approach described above is valid only for the circular motion camera PSFs. Figs. 4.8(c)(e) also show that our PSF estimation can clearly distinguish opposite motion directions.

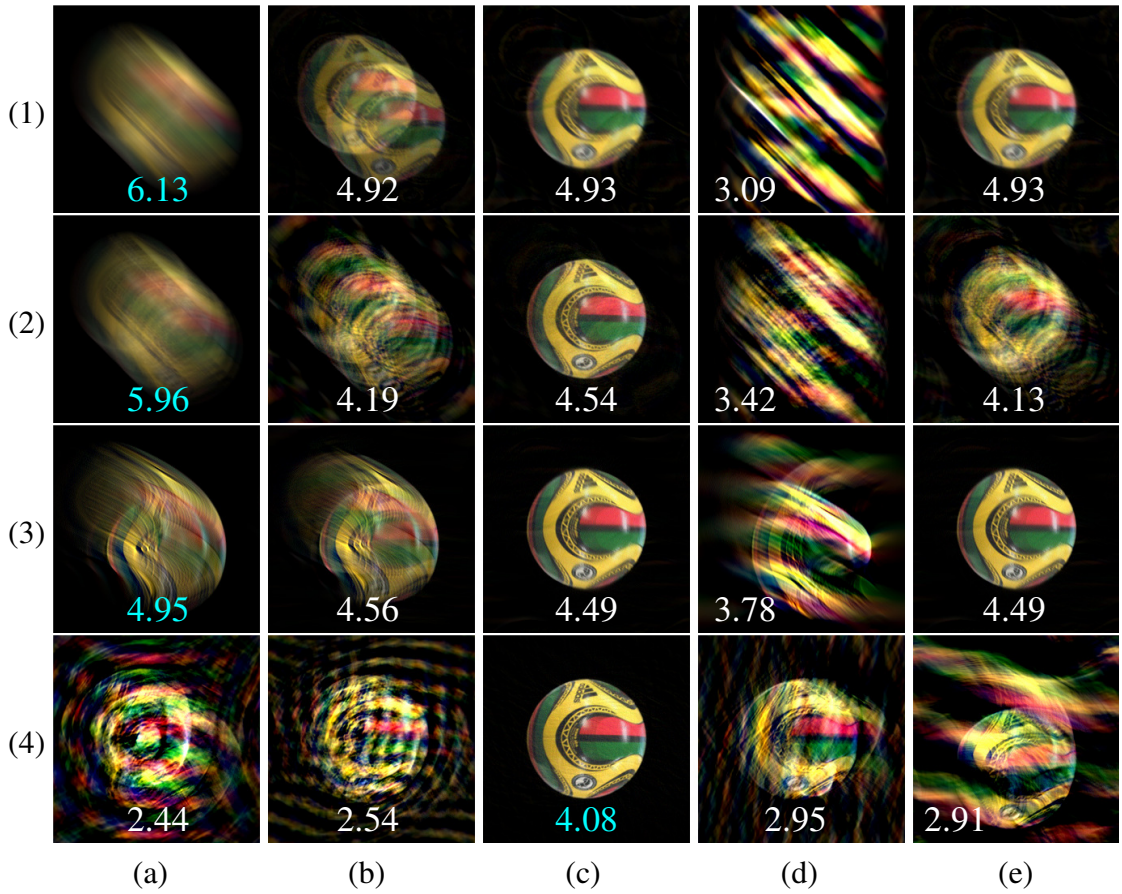


Figure 4.8: Deconvolution results for a synthetically motion-blurred ball and their corresponding values of the sparseness prior Eq. 4.12. The largest value for each camera is written in cyan, which identifies the correct PSF only for the circular motion camera. Rows: (1) Static camera. (2) Coded exposure camera. (3) Motion-invariant camera. (4) Circular motion camera. Columns: (a) Deconvolution results with static object PSFs, (direction, speed) = $(0°, 0)$. (b) Incorrect speed, $(45°, S/2)$. (c) Correct PSF, $(45°, S)$. (d) Incorrect direction, $(90°, S)$. (e) Opposite direction, $(225°, S)$.

4.5 Camera Hardware Implementation

While we believe that circular sensor motion can be implemented with sensor shift mechanisms for image stabilization, they are not currently accessible to users, and we made the following two prototypes by taking different approaches to implementing circular motion.

For the first prototype shown in Fig. 4.9, we opt to translate the entire camera body mechanically using stepper motors. The circular motion radius is R = 1.1cm, and the exposure time is set to $2T$ = 1.0sec. Translation of the whole camera body makes the target object speed S equal to $R\omega$ *in the world space* (about 7cm/sec in our case), and the size of PSFs becomes depth-dependent (i.e., far objects are not affected by the camera motion). Therefore, this prototype only works for scenes up to 50cm away from the

camera. Precision issues also kept us from using it for farther scenes, because the PSF for static objects deviates from a circle as the optical axis slightly tilts during circular motion.

To overcome the above-described issues, for the second prototype, we place a tilted acrylic plate inside the camera lens mount as shown in Fig. 4.10, and *rotate* it so that refracted light rays move circularly. The plate is 3mm thick with a refraction index of 1.49, and the tilt angle is 7.7 degrees, making the circular motion radius R to be 0.13mm. This radius corresponds to 5 pixels in our experiments, and hence the target object speed is $S = 31.4$ pixels/sec with the exposure time $2T = 1.0$ sec. We used this second prototype for all of the results shown below.

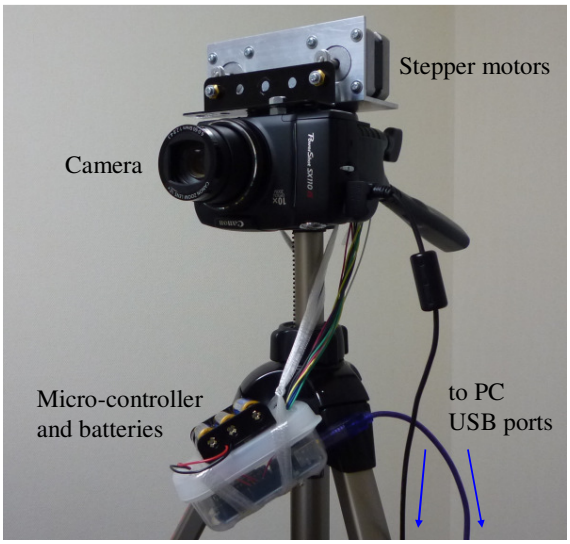


Figure 4.9: Prototype camera based on a Canon PowerShot SX110. The whole camera body is translated by two stepper motors.

4.6 Results

For deblurring, we performed the PSF estimation described in Sec. 4.4 for each user-segmented object, and applied deconvolution with the estimated PSF. In order to reduce ringing due to boundary effects, we made a rectangular image containing a cropped object, and smoothly fill in the outside of the object region with periodic boundary condition similarly to [57]. The deblurred objects and the background are blended back together. The PSF estimation using a multi-resolution approach similar to [28] took 20 min for a 512×512 image on a desktop PC with an Intel Pentium 4 3.2GHz CPU and 2GB RAM. User intervention for motion segmentation took less than a minute. An example of segmentation can be seen in Fig. 4.13(d).

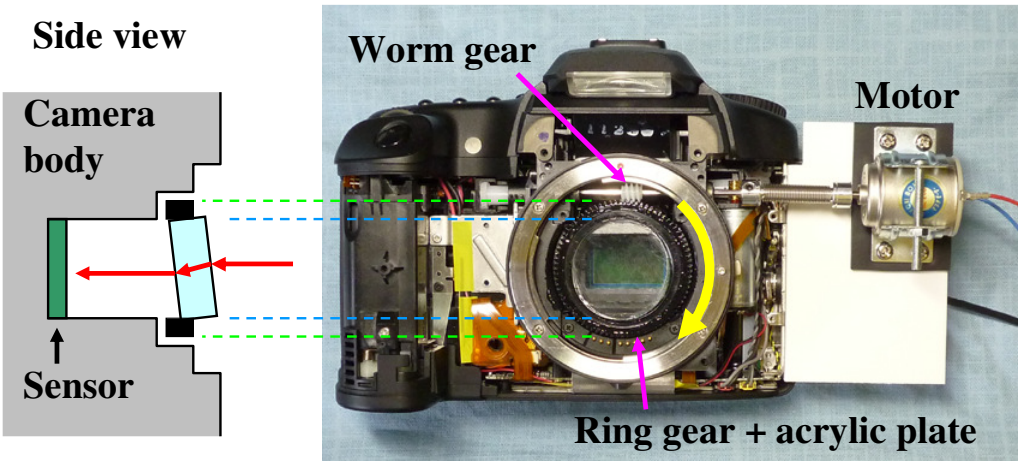
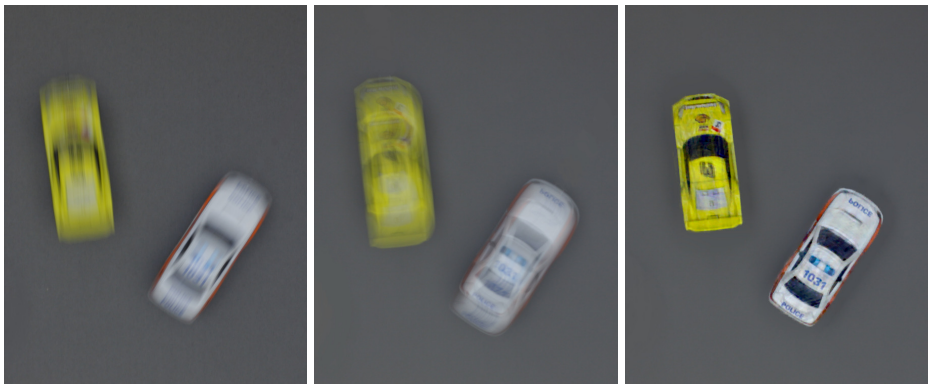


Figure 4.10: Prototype camera based on a Canon EOS 40D DSLR, where the lens is detached to reveal the modified lens mount. We attach an ordinary Canon EF 50mm f/1.8 lens for image capture. After passing through the lens, incoming light (shown in red) is displaced via the tilted acrylic plate, and the displacement sweeps a circle on the sensor while the plate rotates (yellow arrow).

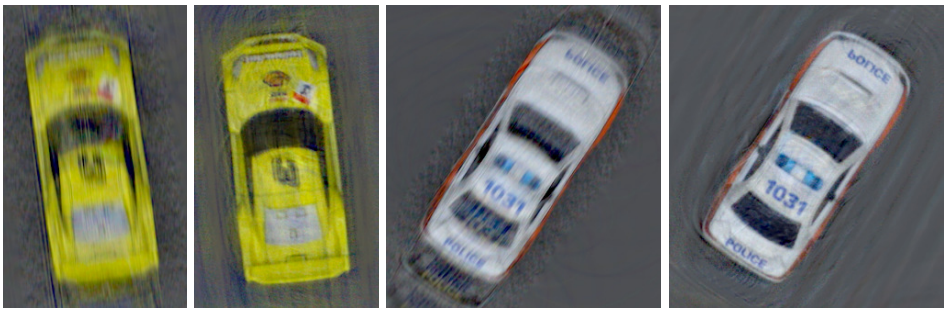
4.6.1 Motion Deblurring Results

Fig. 4.11 shows an example of multiple objects moving in different directions and at different speeds. The digits and marks on the cars are visible in the deblurred image. For comparison, we also show closeups of the deconvolution results in Fig. 4.12 for both the static camera image and the circular motion camera image. Note that, for Fig. 4.12, we used simpler, pseudo-inverse deconvolution to better demonstrate high frequency preservation. As shown, more details are recovered for the circular motion camera image with less deconvolution noise.



(a) (b) (c)

Figure 4.11: Toy cars. (a) From a static camera. (b) From the circular motion camera. (c) Deblurring result of (b).



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

Figure 4.12: Comparison of pseudo-inverse deconvolution results for the toy car example. (a)(c) Results for the static camera image. (b)(d) Results for the circular motion camera image.

Fig. 4.13 shows an example of an object whose parts are moving differently. Fig. 4.13 (d) shows the user-specified motion segmentation. The regions overlap in order to stitch them smoothly at the borders after deconvolution. Details such as fingers and wrinkles on the clothes were recovered.

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

Figure 4.13: Squat motion. (a) From a static camera. (b) From the circular motion camera. (c) Deblurring result of (b). (d) User-specified motion segmentation. Four regions are enclosed by differently-colored lines.

Fig. 4.14 shows an example with a textured background. Due to occlusion boundaries, artifacts can be seen around the silhouettes of the people, but the deblurred faces are clearly recognizable. It is worth mentioning that the circular motion camera tells us that the man was moving downward while the woman was moving leftward (not upward or rightward), which is unavailable information from the static camera image in Fig. 4.14(a) and also from the other capture strategies. We also note that, as the sensor partially tracks object motion during exposure, details such as facial features are already visible in the captured image even before deconvolution as shown in Fig. 4.14(b). To demonstrate this, we applied a facial feature point detector [102] to Figs. 4.14(a-c). As shown in Fig. 4.15, facial feature points were successfully detected without deconvolution. These motion identification and recognizable image capture capabilities may be useful for surveillance

purposes.

(a) (b) (c)

Figure 4.14: Moving people in front of a textured background. (a) From a static camera. (b) From the circular motion camera. (c) Deblurring result of (b).

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

Figure 4.15: Results of facial feature point detection [102] for Fig. 4.14. (a)(d) Detection failed for the static camera image in Fig. 4.14(a), as the faces are severely blurred. (b)(e) Detection succeeded for the circular motion camera image in Fig. 4.14(b) even before they were deblurred, since the facial features are already visible. (c)(f) Detection also succeeded for the deblurred image in Fig. 4.14(c).

Fig. 4.16 shows an example of a license plate of a motorbike. The digits and characters are legible in the deblurred image Fig. 4.16(c). The motorbike is identified as moving rightward (not leftward), which is unavailable information from the static camera image in Fig. 4.16(a). This information may be useful for traffic accident investigation (e.g., to identify whether the motorbike crashed into another car on the left or it was trying to avoid being hit by that car).

(a) (b) (c)

Figure 4.16: License plate of a motorbike. (a) From a static camera. (b) From the circular motion camera. (c) Deblurring result of (b).

4.6.2 Comparisons using a High-speed Camera

For comparison with the other capture strategies, we used high-speed camera images of a horizontally moving resolution chart provided online [5]. Blurred images are simulated by averaging 150 frames from the 1,000 fps video, resulting in a 39-pixel blur. The length 50 code was used for the coded exposure camera, spending 3 msec for each chop of the code. For fair comparison, the motion-invariant and circular motion cameras were targeted to an object speed of 50 pixels (not 39 pixels) per exposure time. We tilted the camera by 90° to simulate the "vertical" object motion relative to the camera. As shown in Fig. 4.17, the coded exposure deblurring produced a less noisy image than the static camera, although oblique streaks of noise can still be seen. The motion-invariant camera produced an even less noisy image for horizontal object motion, but the result for vertical object motion exhibits severe noise. The circular motion camera produced clean images for both motion directions, although they are not artifact-free, either.

We also used high speed camera images to demonstrate the recognizability of captured images even before deconvolution, as compared to the other image capture strategies. Examples of a vertically moving face are shown in Fig. 4.18. The facial feature point detection succeeded only for the circular motion camera image of Fig. 4.18(d), as shown in Fig. 4.18(e).

Fig. 4.19 shows examples of a license plate. They are also simulated from high speed camera images (note that Fig. 4.16 is a real example, not a simulated one). Large digits "72-14" are legible for all of the capture strategies, but the characters above these digits are hard to recognize in the static and coded exposure images shown in Figs. 4.19(a)(b). Legibility for the motion invariant camera image (c) is not as good as that for the circular motion camera image (d) as the motion direction is slightly off the horizontal.

4.7 Summary

This chapter has proposed a method for removing motion blur by translating a camera sensor circularly about the optical axis during exposure, so that high frequencies can be preserved for a wide range of in-plane linear object motion up to some target speed. We analyzed the frequency characteristics of circular sensor motion in relation to linear object motion, and investigated its trade-offs between other capture strategies. We have also presented a blur estimation method that can be applied to a set of PSFs resulting from circular sensor motion, based on a simple observation that deconvolution by wrong PSFs causes ringing artifacts, which is not always the case for other image capture strategies.

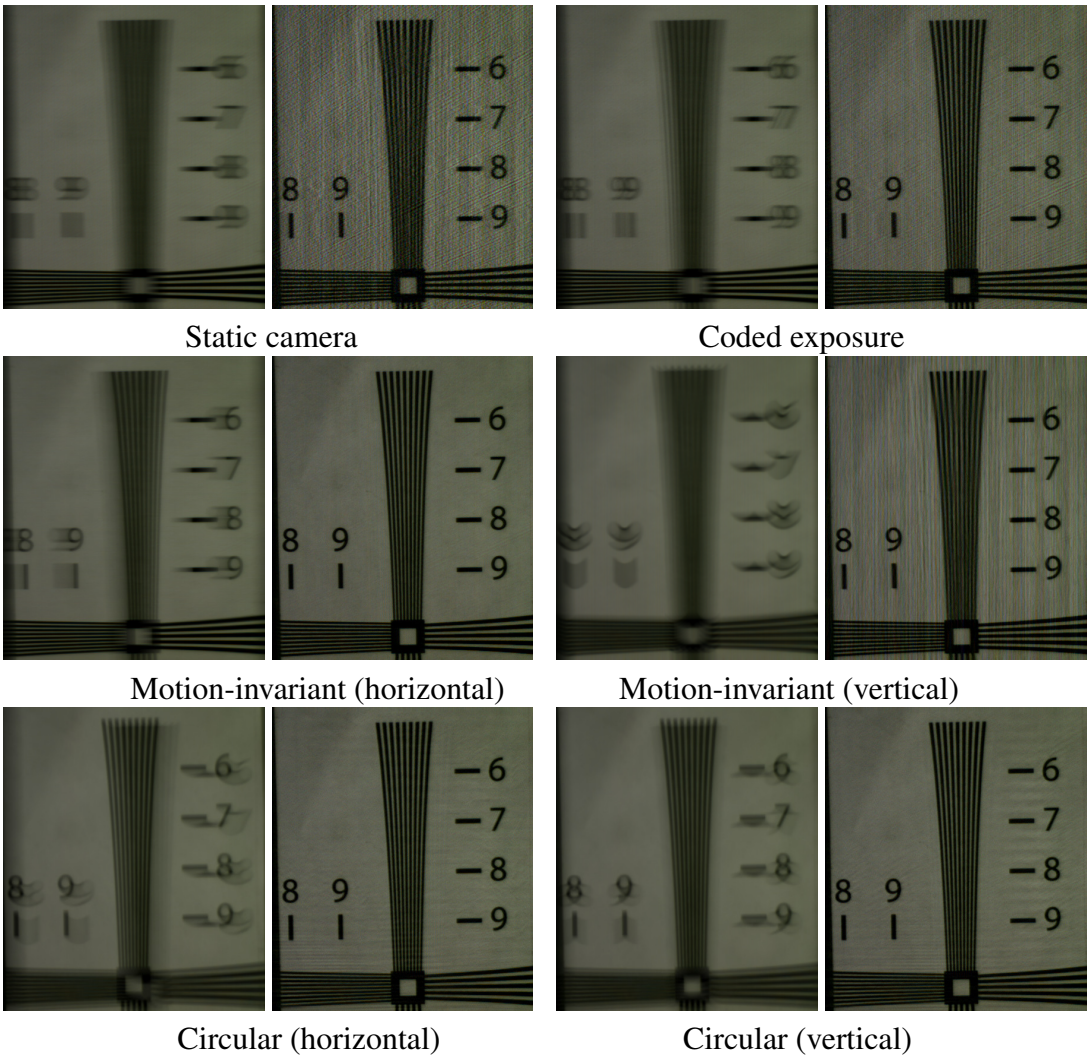


Figure 4.17: Comparison using high-speed camera images. For each pair of shown images, the left one is a simulated blurred image, and the right one is its deconvolution result.

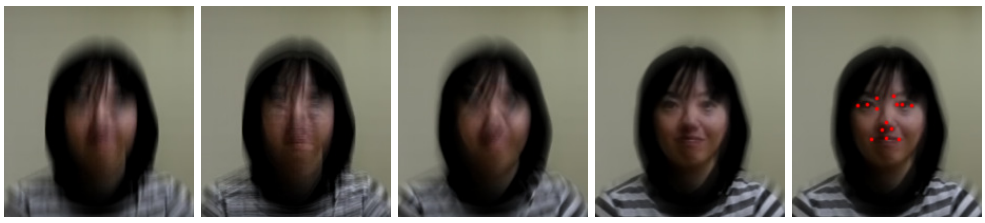
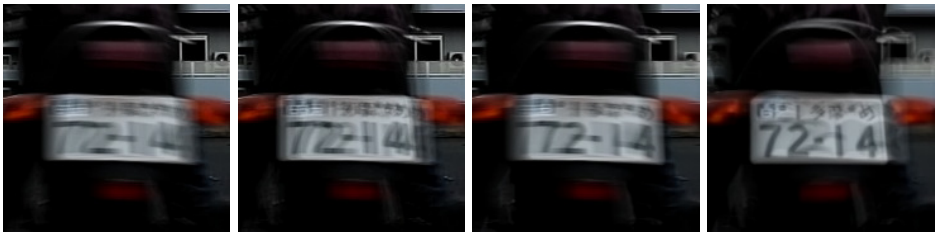


Figure 4.18: Motion-blurred face simulated from high speed camera images. (a) Static camera. (b) Coded exposure camera. (c) Motion-invariant camera. (d) Circular motion camera. (e) Facial feature point detection succeeded for the circular motion camera image (d) without deconvolution (and failed for the others (a-c)).



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

Figure 4.19: Motion-blurred license plate simulated from high speed camera images. (a) Static camera. (b) Coded exposure camera. (c) Motion-invariant camera. (d) Circular motion camera.

We have shown deconvolution results for simulated images as well as real photographs captured by our prototype camera, and demonstrated that objects moving in different directions at different speeds can be deblurred equally well.

Chapter 5

Conclusions and Future Work

This dissertation has proposed methods for removing defocus and motion blurs in photographs. Since deblurring is generally an ill-posed problem, the proposed method includes modifications of camera optics that alter the image capture process of traditional cameras in order to achieve high frequency preservation and to facilitate blur kernel identification. Aiming at applications to consumer digital cameras, this dissertation proposed low cost hardware implementation which adopted small modifications to existing cameras and mechanisms that can be directly derived from existing ones.

5.1 Common Issues

This section describes several issues common to various parts of the proposed methods, which result from our assumption that spatially-variant blur in an input photograph can be locally approximated by a uniform blur. This directly leads to the following limitations, which we would like to address in the future.

First, in order for blur estimation to be reliable, objects in a photograph should be larger than the blur size around them, so that local segments contain a uniform blur with enough sample pixels. Hence, estimation can be erroneous for small or thin objects (e.g., a strand of hair).

Second, since blur is locally modeled as convolution by a single PSF, translucent objects are not accounted for. A similar problem occurs around occlusion boundaries [7], which we alleviated by blending deblurred images. The quality of deblurred images will degrade particularly if occlusion boundaries frequently appear in a scene (e.g., bars of a cage).

Third, sources of image degradation other than blur, such as image compression artifacts can disrupt our blur estimation and deconvolution algorithms. Over/under-exposures also lead to loss of information, breaking the linear relationship between pixel values and captured light intensities. Blur estimation can still be conducted by excluding affected regions, but deconvolution will produce artifacts around there as shown in Fig. 5.1. Transparent objects and specular highlights will also induce similar artifacts as they distort the PSF shape.

(a) (b)

Figure 5.1: (a) Saturated input photograph. (b) Result of deblurring.

5.2 Image Processing Approach to Image Deblurring

We have presented a method for removing defocus blur in images in the context of digital refocusing, in which the goal is not only to perform deblurring but also to create images with different focus settings. The proposed method relies exclusively on an image processing approach without camera optics modifications, in order to set a baseline performance achievable without modifying the image capture process. The proposed method consists of a fast image deconvolution method for efficient deblurring, a local blur estimation method which can handle abrupt blur changes at depth discontinuities due to object boundaries, and a set of user interfaces for interactive refocusing.

Although the gradient domain approach made the deconvolution process faster, we are no longer able to directly impose positivity constraints on variables, which are known to be effective in regularizing the solution. Currently we fix values after bringing them back to the image domain, but we would like to seek a way to incorporate such constraints into the deconvolution process. The degree of ringing suppression of our deconvolution method depends on the choice of parameter w, which is related to the image noise level. We would like to consider determining the parameter automatically based on noise estimation methods [56].

We used a simple pillbox PSF model, which seems sufficient for the defocus blur of

the lens we used. Nevertheless, it is worth considering the use of more complex models and calibrated PSFs depending on a target imaging system.

It would be interesting to consider applying heavy-tailed priors also to blur estimation, which we did not because we knew that the defocus PSF was a pillbox, which is much stronger prior knowledge about the PSF shape; and we assumed the blur to be uniform within each segment, which may be interpreted as a heavy-tailed prior that allows discontinuities in a blur radius field occasionally at segment boundaries. For better blur estimation, it would also be useful to improve segmentation quality.

We provided a means of modifying a blur radius field to fix ringing artifacts that may still remain. Skilled retouching software users could further improve the quality by directly working on the latent images. We would like to consider developing example-based touch-up tools for ordinary users.

5.3 Defocus Blur Removal using a Color-filtered Aperture

We have presented a method for estimating defocus blur sizes and for extracting the alpha matte of an in-focus foreground object in order to facilitate defocus blur removal. Our method only modifies a camera lens with off-the-shelf color filters and utilizes the RGB planes of the image sensor of a conventional camera body to capture multi-view images in a single exposure. We have proposed an effective correspondence measure between the RGB planes, and a method for employing color misalignment cues to improve the matte.

The major limitation of our approach is that it does not work for objects having only a single pure R, G, or B color. Combining with depth-from-defocus methods may partially solve this problem. However, this does not mean that objects must have achromatic colors all over. For example, the disparity of the red box in Fig. 3.3 is correctly identified as shown in Fig. 3.10(a), thanks to the alphabets and the pictures of chocolates printed on the box. Therefore, our requirement is that objects must not be purely colored *entirely*, and we think there are many real-world objects satisfying this requirement. We would like to further investigate this limitation.

In our imaging system, the f-number is fixed to 1.8 (full aperture of our prototype lens) because a large aperture increases disparities and thus increases depth resolution. Since disparities also increase when the lens is focused near, our system typically works well for foreground objects at 0.5 to 2.5 meters away from the camera with a sufficiently distant (about twice as far away) background. For farther scenes, depth resolution will gradually

decrease, and the matte quality will also deteriorate as difference between foreground and background color misalignment will be small. At a certain point, there will be no disparity, and the system will not work at all.

By introducing color filters, the amount of incident light is decreased. Increasing the aperture filter area to compensate for this introduces more defocus. While this degrades depth estimation accuracy at defocused regions, it suppresses background clutters, which is beneficial for matting. Color filters may also affect color demosaicing for the image sensor, although we did not observe any loss of quality in our experiments, mainly because we downsampled the captured images for tractable computation time.

While our blur estimation works fairly robustly, our matting fails when the foreground and background colors are similar with little texture, as shown in Fig. 5.2(b), since we have few color misalignment cues. Another failure mode is that, as we use a relatively large window (15×15), we cannot recover small/thin features such as hair strands and holes in foreground objects, once they are missed in the course of optimization, as shown in Fig. 5.2(d). We would like to address the above issues in the future.

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

Figure 5.2: Failure cases of the proposed matting algorithm. Major errors are indicated by the arrow and circles. (a) Captured image. (b) Matte from (a). (c) Closeup of the ground truth matte for the girl image in Fig. 3.14. (d) Our result.

5.4 Motion Blur Removal using Circular Sensor Motion

We have proposed to translate a camera sensor circularly about the optical axis during exposure, so that high frequencies can be preserved for a wide range of in-plane linear object motion up to some target speed. We analyzed the frequency characteristics of circular sensor motion in relation to linear object motion, and investigated its trade-offs between other capture strategies.

Our camera prevents capture-time loss of frequency content of images and also facilitates blur estimation. However, another issue of classical motion deblurring remains. That is, motion segmentation is left an open problem, for which we assumed user-intervention

in this dissertation. We also confined ourselves to considering only in-plain linear object motion. We would like to address these limitations in the future.

Another issue of our method is that static objects are also blurred. One way to alleviate this is to pause the sensor for a fraction of exposure time before or after the circular motion. We intend to investigate ways to control the degree to which static and moving objects are favored relative to each other.

5.5 Future Directions

In this dissertation we have presented two types of camera modifications. One is to place color filters in the camera lens aperture, and the other is to move the camera image sensor circularly. Both of the modifications have large design spaces. For the color-filtered aperture, we could change the square shape of each filter area into other shapes such as circles and hexagons, or we could change relative positions of the filters. For the circular sensor motion, we could move the sensor multiple times, move it with acceleration, or move it along a whorl-like path. We would like to investigate the pros and cons of various designs for each of the two modified image capture processes.

Another future direction is that, as we have focused on compact and low-cost implementation of camera hardware modifications, we are interested in making the existing computational photography techniques (including the ones we have proposed in this dissertation) more common to ordinary people. The first step we would like to take is to implement multi-sensor consumer digital cameras for light-field capture, which can leverage the abundant findings and knowledge from the recent advances in this field.

Bibliography

[1] E. H. Adelson and J. Y. A. Wang. Single lens stereo with a plenoptic camera. *IEEE Trans. Pattern Anal. Machine Intell.*, 14(2):99–106, 1992.

[2] Adobe Systems Inc. Photoshop CS. http://www.adobe.com/.

[3] A. Agrawal and R. Raskar. Optimal single image capture for motion deblurring. In *Proc. CVPR*, pages 1–8, 2009.

[4] A. Agrawal and Y. Xu. Coded exposure deblurring: optimized codes for PSF estimation and invertibility. In *Proc. CVPR*, pages 1–8, 2009.

[5] A. Agrawal, Y. Xu, R. Raskar, and J. Tumblin. Motion blur datasets and matlab codes. http://www.umiacs.umd.edu/˜aagrawal/MotionBlur/.

[6] Y. Amari and E. H. Adelson. Single-eye range estimation by using displaced apertures with color filters. In *Proc. Int. Conf. Industrial Electronics, Control, Instrumentation, and Automation*, volume 3, pages 1588–1592, 1992.

[7] N. Asada, H. Fujiwara, and T. Matsuyama. Seeing behind the scene: analysis of photometric properties of occluding edges by the reversed projection blurring model. *IEEE Trans. Pattern Anal. Machine Intell.*, 20(2):155–167, 1998.

[8] O. Axelsson. *Iterative Solution Methods*. Cambridge University Press, 1994.

[9] G. R. Ayers and J. C. Dainty. Iterative blind deconvolution method and its applications. *Optics Letters*, 13(7):547–549, 1988.

[10] M. R. Banham and A. K. Katsaggelos. Spatially adaptive wavelet-based multiscale image restoration. *IEEE Trans. Image Processing*, 5(4):619–634, 1996.

[11] M. R. Banham and A. K. Katsaggelos. Digital image restoration. *IEEE Signal Processing Magazine*, 14(2):24–41, 1997.

[12] M. Ben-Ezra and S. K. Nayar. Motion-based motion deblurring. *IEEE Trans. Pattern Anal. Machine Intell.*, 26(6):689–698, 2004.

[13] M. Ben-Ezra, A. Zomet, and S. K. Nayar. Video super-resolution using controlled subpixel detector shifts. *IEEE Trans. Pattern Anal. Machine Intell.*, 27(6):977–987, 2005.

[14] J. Biemond, R. L. Lagendijk, and R. M. Mersereau. Iterative methods for image deblurring. *Proceedings of the IEEE*, 78(5):856–883, 1990.

[15] J. M. Bioucas-Dias. Bayesian wavelet-based image deconvolution: a GEM algorithm exploiting a class of heavy-tailed priors. *IEEE Trans. Image Processing*, 15(4):937–951, 2006.

[16] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Anal. Machine Intell.*, 23(11):1222–1239, 2001.

[17] Y. Y. Boykov and M.-P. Jolly. *Interactive graph cuts* for optimal boundary & region segmentation of objects in N-D images. In *Proc. ICCV*, pages 105–112, 2001.

[18] R. N. Bracewell. *The Fourier transform and its applications*. McGraw-Hill, 1965.

[19] P. J. Burt and R. J. Kolczynski. Enhanced image capture through fusion. In *Proc. ICCV*, pages 173–182, 1993.

[20] M. Cannon. Blind deconvolution of spatially invariant image blurs with phase. *IEEE Trans. Acous., Speech, and Sig. Processing*, 24(1):58–63, 1976.

[21] I.-C. Chang, C.-L. Huang, W.-J. Hsueh, H.-C. Lin, C.-C. Chen, and Y.-H. Yeh. A novel 3-D hand-held camera based on tri-aperture lens. In *Proc. SPIE 4925*, pages 655–662, 2002.

[22] Y.-Y. Chuang. *New models and methods for matting and compositing*. PhD thesis, University of Washington, 2004.

[23] Y.-Y. Chuang, B. Curless, D. H. Salesin, and R. Szeliski. A Bayesian approach to digital matting. In *Proc. CVPR*, volume 2, pages 264–271, 2001.

[24] R. Coifman and D. Donoho. Translation-invariant de-noising. In *Wavelets and Statistics*, volume 103 of *Lecture Notes in Statistics*, pages 125–150. Springer-Verlag, 1995.

[25] D. Comaniciu and P. Meer. Robust analysis of feature spaces: color image segmentation. In *Proc. CVPR*, pages 750–755, 1997.

[26] F. C. Crow. Summed-area tables for texture mapping. *Computer Graphics (Proc. SIGGRAPH 84)*, 18(3):207–212, 1984.

[27] E. R. Dowski and G. E. Johnson. Wavefront coding: a modern method of achieving high performance and/or low cost imaging systems. In *Proc. SPIE 3779*, pages 137–145, 1999.

[28] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman. Removing camera shake from a single photograph. *ACM Trans. Gr.*, 25(3):787–794, 2006.

[29] D. J. Field. What is the goal of sensory coding? *Neural Computation*, 6:559–601, 1994.

[30] M. Figueiredo and R. Nowak. An EM algorithm for wavelet-based image restoration. *IEEE Trans. Image Processing*, 12(8):906–916, 2003.

[31] T. Georgiev, K. C. Zheng, B. Curless, D. Salesin, S. Nayar, and C. Intwala. Spatio-angular resolution tradeoff in integral photography. In *Proc. Eurographics Symposium on Rendering*, pages 263–272, 2006.

[32] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen. The lumigraph. In *Proc. ACM SIGGRAPH 96*, pages 43–54, 1996.

[33] I. S. Gradshteyn and I. M. Ryzhik. *Table of integrals, series, and products (sixth edition)*. Academic Press, 2000.

[34] L. Grady. Random walks for image segmentation. *IEEE Trans. Pattern Anal. Machine Intell.*, 28(11):1768–1783, 2006.

[35] P. Green, W. Sun, W. Matusik, and F. Durand. Multi-aperture photography. *ACM Trans. Gr.*, 26(3):68:1–68:7, 2007.

[36] S. Hiura and T. Matsuyama. Depth measurement by the multi-focus camera. In *Proc. CVPR*, pages 953–959, 1998.

[37] A. Isaksen, L. McMillan, and S. J. Gortler. Dynamically reparameterized light fields. In *Proc. ACM SIGGRAPH 2000*, pages 297–306, 2000.

[38] JEITA. EXIF: exchangeable image file format for digital still camera. http://www.exif.org/specifications.html.

[39] N. Joshi, W. Matusik, and S. Avidan. Natural video matting using camera arrays. *ACM Trans. Gr.*, 25(3):779–786, 2006.

[40] J. Kautsky, J. Flusser, B. Zitova, and S. Simberova. A new wavelet-based measure of image focus. *Pattern Recognition Letters*, 23(14):1785–1794, 2002.

[41] J. Kim, V. Kolmogorov, and R. Zabih. Visual correspondence using energy minimization and mutual information. In *Proc. ICCV*, volume 2, pages 1033–1040, 2003.

[42] A. Kubota and K. Aizawa. Reconstructing arbitrarily focused images from two differently focused images using linear filters. *IEEE Trans. Image Processing*, 14(11):1848–1859, 2005.

[43] D. Kundur and D. Hatzinakos. Blind image deconvolution. *IEEE Signal Processing Magazine*, 13(3):43–64, 1996.

[44] R. L. Lagendijk and J. Biemond. Block-adaptive image identification and restoration. In *Proc. Int. Conf. Acoustics, Speech, and Signal Processing*, pages 2497–2500, 1991.

[45] S.-H. Lai, C.-W. Fu, and S. Chang. A generalized depth estimation algorithm with a single image. *IEEE Trans. Pattern Anal. Machine Intell.*, 14(4):405–411, 1992.

[46] T. R. Lauer. Deconvolution with a spatially-variant PSF. In *Proc. SPIE 4847*, pages 167–173, 2002.

[47] A. Levin. Blind motion deblurring using image statistics. In *Proc. Advances in Neural Information Processing Systems (NIPS)*, 2006.

[48] A. Levin, R. Fergus, F. Durand, and W. T. Freeman. Image and depth from a conventional camera with a coded aperture. *ACM Trans. Gr.*, 26(3):70:1–70:9, 2007.

[49] A. Levin, D. Lischinski, and Y. Weiss. A closed form solution to natural image matting. In *Proc. CVPR*, pages 61–68, 2006.

[50] A. Levin, D. Lischinski, and Y. Weiss. A closed-form solution to natural image matting. *IEEE Trans. Pattern Anal. Machine Intell.*, 30(2):228–242, 2008.

[51] A. Levin, P. Sand, T. S. Cho, F. Durand, and W. T. Freeman. Motion-invariant photography. *ACM Trans. Gr.*, 27(3):71:1–71:9, 2008.

[52] M. Levoy and P. Hanrahan. Light field rendering. In *Proc. ACM SIGGRAPH 96*, pages 31–42, 1996.

[53] J. P. Lewis. Fast template matching. In *Proc. Vision Interface*, pages 120–123, 1995.

[54] C.-K. Liang, T.-H. Lin, B.-Y. Wong, C. Liu, and H. H. Chen. Programmable aperture photography: multiplexed light field acquisition. *ACM Trans. Gr.*, 27(3):55:1–55:10, 2008.

[55] C.-K. Liang, G. Liu, and H. H. Chen. Light field acquisition using programmable aperture camera. In *Proc. ICIP*, volume 5, pages 233–236, 2007.

[56] C. Liu, W. T. Freeman, R. Szeliski, and S. B. Kang. Noise estimation from a single image. In *Proc. CVPR*, pages 901–908, 2006.

[57] R. Liu and J. Jia. Reducing boundary artifacts in image deconvolution. In *Proc. ICIP*, pages 505–508, 2008.

[58] L. B. Lucy. An iterative technique for the rectification of observed distributions. *The Astronomical Journal*, 79(6):745–754, 1974.

[59] W. Matusik, H. Pfister, A. Ngan, P. Beardsley, R. Ziegler, and L. McMillan. Image-based 3D photography using opacity hulls. *ACM Trans. Gr.*, 21(3):427–437, 2002.

[60] M. McGuire, M. Matusik, H. Pfister, F. Durand, and J. Hughes. Defocus video matting. *ACM Trans. Gr.*, 24(3):567–576, 2005.

[61] M. McGuire, W. Matusik, and W. Yerazunis. Practical, real-time studio matting using dual imagers. In *Proc. Eurographics Symposium on Rendering*, pages 235–244, 2006.

[62] N. W. McLachlan. *Bessel functions for engineers*. Oxford University Press, 1934.

[63] A. Mohan, D. Lanman, S. Hiura, and R. Raskar. Image destabilization: programmable defocus using lens and sensor motion. In *IEEE Int. Conf. Computational Photography*, 2009.

[64] F. Moreno-Noguer, P. N. Belhumeur, and S. K. Nayar. Active refocusing of images and videos. *ACM Trans. Gr.*, 26(3):67:1–67:9, 2007.

[65] H. Nagahara, S. Kuthirummal, C. Zhou, and S. K. Nayar. Flexible depth of field photography. In *Proc. ECCV*, pages 60–73, 2008.

[66] J. G. Nagy and D. P. O'Leary. Restoring images degraded by spatially variant blur. *SIAM Journal on Scientific Computing*, 19(4):1063–1082, 1998.

[67] S. K. Nayar and Y. Nakagawa. Shape from focus. *IEEE Trans. Pattern Anal. Machine Intell.*, 16(8):824–831, 1994.

[68] R. Neelamani, H. Choi, and R. Baraniuk. ForWaRD: Fourier-wavelet regularized deconvolution for ill-conditioned systems. *IEEE Trans. Signal Processing*, 52(2):418–433, 2004.

[69] R. Ng. Fourier slice photography. *ACM Trans. Gr.*, 24(3):735–744, 2005.

[70] R. Ng, M. Levoy, M. Brédif, G. Duval, M. Horowitz, and P. Hanrahan. Light field photography with a hand-held plenoptic camera. Tech. Rep. CSTR 2005-02, Stanford Computer Science, Apr. 2005.

[71] I. Omer and M. Werman. Color lines: image specific color representation. In *Proc. CVPR*, volume 2, pages 946–953, 2004.

[72] M. K. Ozkan, A. M. Tekalp, and M. I. Sezan. Identification of a class of space-variant image blurs. In *Proc. SPIE 1452*, pages 146–156, 1991.

[73] A. P. Pentland. A new sense for depth of field. *IEEE Trans. Pattern Anal. Machine Intell.*, 9(4):523–531, 1987.

[74] P. Perez, M. Gangnet, and A. Blake. Poisson image editing. *ACM Trans. Gr.*, 22(3):313–318, 2003.

[75] R. Raskar, A. Agrawal, and J. Tumblin. Coded exposure photography: motion deblurring using fluttered shutter. *ACM Trans. Gr.*, 25(3):795–804, 2006.

[76] R. Raskar, J. Tumblin, A. Mohan, A. Agrawal, and Y. Li. Computational photography. In *Proc. Eurographics STAR*, 2006.

[77] A. Rav-Acha and S. Peleg. Two motion-blurred images are better than one. *Pattern Recognition Letters*, 26(3):311–317, 2005.

[78] S. J. Reeves and R. M. Mersereau. Blur identification by the method of generalized cross-validation. *IEEE Trans. Image Processing*, 1(3):301–311, 1992.

[79] W. H. Richardson. Bayesian-based iterative method of image restoration. *Journal of the Optical Society of America*, 62(1):55–59, 1972.

[80] A. E. Savakis and H. J. Trussell. Blur identification by residual spectral matching. *IEEE Trans. Image Processing*, 2(2):141–151, 1993.

[81] Q. Shan, J. Jia, and A. Agarwala. High-quality motion deblurring from a single image. *ACM Trans. Gr.*, 27(3):73:1–73:10, 2008.

[82] A. R. Smith and J. F. Blinn. Blue screen matting. In *Proc. ACM SIGGRAPH 96*, pages 259–268, 1996.

[83] E. M. Stein and G. Weiss. *Introduction to Foureir analysis on Euclidean spaces*. Princeton University Press, 1971.

[84] M. Subbarao, T. Choi, and A. Nikzad. Focusing techniques. *Optical Engineering*, 32(11):2824–2836, 1993.

[85] M. Subbarao and N. Gurumoorthy. Depth recovery from blurred edges. In *Proc. CVPR*, pages 498–503, 1988.

[86] J. Sun, J. Jia, C.-K. Tang, and H.-Y. Shum. Poisson matting. *ACM Trans. Gr.*, 23(3):315–321, 2004.

[87] M. F. Tappen, B. C. Russell, and W. T. Freeman. Exploiting the sparse derivative prior for super-resolution and image demosaicing. In *Proc. 3rd Int. Workshop on Statistical and Computational Theories of Vision*, 2003.

[88] A. N. Tikhonov and V. Y. Arsenin. *Solutions of ill-posed problems*. Wiley, 1977.

[89] H. J. Trussell and B. R. Hunt. Image restoration of space-variant blurs by sectional methods. *IEEE Trans. Acous., Speech, and Sig. Processing*, 26:608–609, 1978.

[90] A. Veeraraghavan, R. Raskar, A. Agrawal, A. Mohan, and J. Tumblin. Dappled photography: mask enhanced cameras for heterodyned light fields and coded aperture refocusing. *ACM Trans. Gr.*, 26(3):69:1–69:12, 2007.

[91] P. Vlahos. Electronic composite photography. U. S. Patent 3,595,987, 1971.

[92] J. Wang and M. F. Cohen. An iterative optimization approach for unified image segmentation and matting. In *Proc. ICCV*, volume 2, pages 936–943, 2005.

[93] J. Wang and M. F. Cohen. Optimized color sampling for robust matting. In *Proc. CVPR*, 2007.

[94] G. N. Watson. *A treatise on the theory of Bessel functions*. Cambridge University Press, 1922.

[95] M. Welk, D. Theis, T. Brox, and J. Weickert. PDE-based deconvolution with forward-backward diffusivities and diffusion tensors. In *Scale Space and PDE*

Methods in Computer Vision, volume 3459 of *Lecture Notes in Computer Science*, pages 585–597, 2005.

[96] M. Welk, D. Theis, and J. Weickert. Variational deblurring of images with uncertain and spatially variant blurs. In *Proc. DAGM-Symposium*, pages 485–492, 2005.

[97] Y. Wexler, A. Fitzgibbon, and A. Zisserman. Bayesian estimation of layers from multiple images. In *Proc. ECCV*, pages 487–501, 2002.

[98] W. Xiong and J. Jia. Stereo matching on objects with fractional boundary. In *Proc. CVPR*, 2007.

[99] Y. Xiong and S. A. Shafer. Depth from focusing and defocusing. In *Proc. CVPR*, pages 68–73, 1993.

[100] Y.-L. You and M. Kaveh. Blind image restoration by anisotropic regularization. *IEEE Trans. Image Processing*, 8(3):396–407, 1999.

[101] L. Yuan, J. Sun, L. Quan, and H.-Y. Shum. Image deblurring with blurred/noisy image pairs. *ACM Trans. Gr.*, 26(3):1:1–1:10, 2007.

[102] M. Yuasa and O. Yamaguchi. Real-time face blending by automatic facial feature point detection. In *IEEE Int. Conf. Automatic Face & Gesture Recognition*, pages 1–6, 2008.

[103] C. Zhou and S. Nayar. What are good apertures for defocus deblurring? In *IEEE Int. Conf. Computational Photography*, 2009.

[104] C. L. Zitnick, S.-B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski. High-quality video view interpolation using a layered representation. *ACM Trans. Gr.*, 23(3):600–608, 2004.

Appendix

A Color Alignment Measure and Normalized Cross Correlation

An equivalent of Eq. 3.1 in 2D (e.g., in the RG space) would be:

$$L(x,y;d) = \lambda_0\lambda_1/\sigma_r^2\sigma_g^2. \tag{A.1}$$

Let σ_{rg} be the covariance between the R and G components, then by $\lambda_0\lambda_1 = \det(\Sigma) = \sigma_r^2\sigma_g^2 - \sigma_{rg}^2$, we obtain:

$$L(x,y;d) = 1 - \sigma_{rg}^2/\sigma_r^2\sigma_g^2 \tag{A.2}$$

Since normalized cross-correlation (NCC) is given as $\text{NCC} = \sigma_{rg}/\sigma_r\sigma_g \in [-1,1]$, and $|\text{NCC}| \in [0,1]$ indicates the magnitude of correlation, the 2D version of the color alignment measure L has a one-to-one correspondence to $|\text{NCC}|$.

B Computing the Color Lines Model Error

Letting c_i be the i-th color in $S_F(x,y;d)$, μ be the mean color, and v_0 be a unit vector of the fitted line (the first principal eigenvector), trigonometry gives the distance l_i of the point c_i from the line as:

$$l_i^2 = |c_i - \mu|^2 - ((c_i - \mu)^T v_0)^2. \tag{B.1}$$

The average of the first term is, by definition, the variance:

$$\frac{1}{N}\sum_{i=1}^{N} |c_i - \mu|^2 = \sigma_r^2 + \sigma_g^2 + \sigma_b^2. \tag{B.2}$$

For the second term, we have:

$$\begin{aligned}\frac{1}{N}\sum_{i=1}^{N}((c_i-\mu)^T v_0)^2 &= \frac{1}{N}\sum_{i=1}^{N}((c_i-\mu)^T v_0)^T((c_i-\mu)^T v_0)\\ &= \frac{1}{N}\sum_{i=1}^{N} v_0^T(c_i-\mu)(c_i-\mu)^T v_0\\ &= v_0^T\left(\frac{1}{N}\sum_{i=1}^{N}(c_i-\mu)(c_i-\mu)^T\right)v_0\\ &= v_0^T\Sigma v_0 = v_0^T(\lambda_0 v_0) = \lambda_0(v_0^T v_0) = \lambda_0,\end{aligned} \tag{B.3}$$

by the definitions of the covariance matrix Σ and the eigenvector v_0. Therefore, the color lines model error can be computed as follows.

$$e_F(x,y;d) = \sigma_r^2 + \sigma_g^2 + \sigma_b^2 - \lambda_0. \tag{B.4}$$

This turns out to be similar to the color alignment measure of Eq. 3.1, but we found it more effective for matting to use this unnormalized, direct error measure. Since estimation errors of background disparities are typically larger than those of foreground disparities, we discount $e_B(x,y;d)$ by scaling it by around 0.7-0.9.

C Color Crosstalk Suppression

Let c_r, c_g, and c_b be the mean image colors of a sheet of white paper through the R, G, and B filters, respectively. For our prototype,

$$\begin{aligned}c_r &= (1.000, 0.335, 0.025)^T,\\ c_g &= (0.153, 1.000, 0.162)^T,\\ c_b &= (0.007, 0.190, 1.000)^T,\end{aligned} \tag{C.1}$$

where the values are normalized with respect to the maximum component. Letting $M = (c_r, c_g, c_b)$, we can decompose an observed color c_o into the three aperture filters' contributions by $M^{-1}c_o$.

D Multiple Revolutions of the Sensor

For the velocity disc shown in Fig. 4.5(b), we trace only the circumference of the disc by circularly moving the sensor once during exposure. Here we consider additionally tracing the interior of the disc by concentric circles as shown in Fig. D.1, meaning that the sensor

undergoes circular motion multiple times with different speeds during exposure. This may seem to fill in the frequency zeros of the motion blur PSF power spectra seen in the bottom row of Fig. 4.2. However, frequency zeros remain as shown in Fig. D.2 as the phases of the Fourier transform of PSFs each corresponding to a single revolution cancel each other when superimposed. Moreover, as the number of revolutions increases, PSFs become more like the ones resulting from a static camera or the coded exposure camera, and begin to favor static objects.

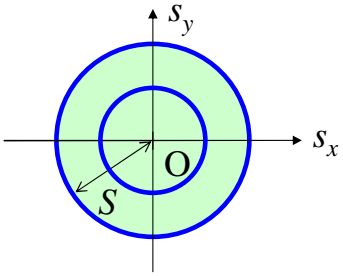


Figure D.1: Example of two circles (shown in blue) for sampling the velocity disc.

E The Slicing Relationship

As we are interested in motion blur PSF, we consider an object as a point light source moving at velocity $\mathbf{v}$ as $\delta(\mathbf{x}-\mathbf{v}t)$. An image of this object (i.e., PSF) observed from a camera moving according to $\mathbf{m}(t)$ during exposure time $[-T,+T]$ is given as:

$$h(\mathbf{x}) = \int_{-T}^{+T} \delta(\mathbf{x}-\mathbf{v}t+\mathbf{m}(t))dt. \tag{E.1}$$

Taking its 2D Fourier transform leads to

$$\hat{h}(\mathbf{f}) = \int_{\Omega}\int_{-T}^{+T} \delta(\mathbf{x}-\mathbf{v}t+\mathbf{m}(t))e^{-2\pi i\mathbf{f}\cdot\mathbf{x}}dtd\mathbf{x}. \tag{E.2}$$

By changing variable as $\mathbf{x}' = \mathbf{v}t-\mathbf{x}$, we obtain:

$$\hat{h}(\mathbf{f}) = \int_{\Omega}\int_{-T}^{+T} \delta(\mathbf{m}(t)-\mathbf{x}')e^{-2\pi i\mathbf{f}\cdot(-\mathbf{x}'+\mathbf{v}t)}dtd\mathbf{x}', \tag{E.3}$$

and integrating with respect to $\mathbf{x}'$ leads to:

$$\hat{h}(\mathbf{f}) = \int_{-T}^{+T} e^{-2\pi i\mathbf{f}\cdot(-\mathbf{m}(t)+\mathbf{v}t)}dt. \tag{E.4}$$

Meanwhile, by integrating Eq. 4.6 with respect to $\mathbf{x}$, we obtain:

$$\hat{p}(\mathbf{f},f_t) = \int_{-T}^{+T} e^{-2\pi i(\mathbf{f}\cdot\mathbf{m}(t)+f_t t)}dt. \tag{E.5}$$

Comparing this equation to Eq. E.4, we see that $\hat{h}(\mathbf{f}) = \hat{p}(-\mathbf{f},\mathbf{f}\cdot\mathbf{v})$, meaning that the Fourier transform of a motion blur PSF is a 2D slice of $\hat{p}(\mathbf{f},f_t)$ along the plane of $f_t = -\mathbf{v}\cdot\mathbf{f}$.

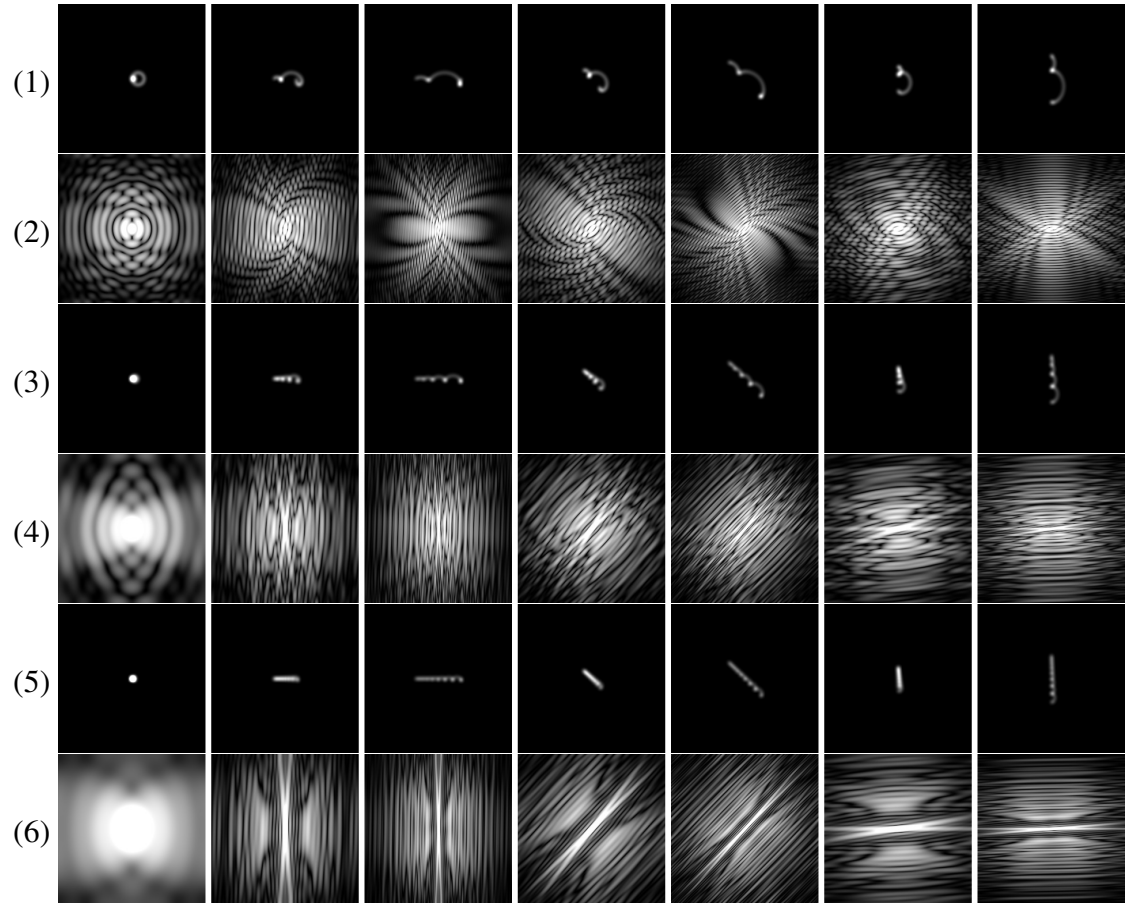


Figure D.2: Motion blur PSFs and their corresponding log power spectra. Rows: (1) PSFs and (2) power spectra resulting from two-revolution circular sensor motion. (3)(4) Five-revolution. (5)(6) Ten-revolution. Columns: (a) Static object. (b)(c) Horizontal object motion at different speeds. (d)(e) Oblique object motion. (f)(g) Vertical object motion.

F The Amount of the Frequency Budget

From Eq. E.5, we see that $\hat{p}(\mathbf{c}, f_t)$, when viewed as a function of f_t, is the (1D) Fourier transform of the following function:

$$b(t) = \begin{cases} e^{-2\pi i \mathbf{c}\cdot\mathbf{m}(t)} & \text{for } t \in [-T, +T] \\ 0 & \text{otherwise} \end{cases} . \tag{F.1}$$

Therefore, using the Parseval's theorem,

$$\begin{aligned} \int_{-\infty}^{+\infty} |\hat{p}(\mathbf{c}, f_t)|^2 df_t &= \int_{-\infty}^{+\infty} |b(t)|^2 dt \\ &= \int_{-T}^{+T} 1 \, dt = 2T. \end{aligned} \tag{F.2}$$

G Fourier Transform of a Spiral

According to [83], 2D Fourier transform of a function $g(r)e^{-ik\theta}$ is given as $G(f_r)e^{-ik\phi}$, where (r,θ) and (f_r,ϕ) are the polar coordinates in the primal and frequency domains, respectively (i.e., $f_r = |\mathbf{f}| \equiv |(f_x, f_y)|$), and we have:

$$G(f_r) = 2\pi i^{-k} \int_0^\infty g(r) J_k(2\pi f_r r) r dr. \tag{G.1}$$

Applying this theorem to Eq. 4.8 leads to:

$$\begin{aligned} \hat{p}(\mathbf{f}, f_t) &= 2\pi i^{-k} e^{-ik\phi} \int_0^\infty \frac{1}{R\omega} \delta(r-R) J_k(2\pi f_r r) r dr \\ &= 2\pi i^{-k} e^{-ik\phi} \frac{1}{\omega} J_k(2\pi R f_r). \end{aligned} \tag{G.2}$$

Hence we have:

$$\begin{aligned} |\hat{p}(\mathbf{f}, f_t)|^2 &= 4\pi^2 \frac{1}{\omega^2} J_k^2(2\pi R f_r) \\ &= 4T^2 J_k^2(2\pi R|\mathbf{f}|). \end{aligned} \tag{G.3}$$