CORRECT SPATIALLY VARYING IMAGE BLUR BY PROJECTIVE MOTION RICHARDSON-LUCY ALGORITHM AND BLURRED IMAGE ALIGNMENT

Gao Long

NATIONAL UNIVERSITY OF SINGAPORE

2010

CORRECT SPATIALLY VARYING IMAGE BLUR BY PROJECTIVE MOTION RICHARDSON-LUCY ALGORITHM AND BLURRED IMAGE ALIGNMENT

Gao Long
(B.Eng.(Hons.), NUS)

A THESIS SUBMITTED

FOR THE DEGREE OF MASTER OF ENGINEERING

DEPARTMENT OF ELECTRICAL & COMPUTER ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

2010

Acknowledgments

My sincere appreciation goes to my supervisor Assistant Professor Tan Ping. Not only has he guided me along the projects, his inspiration and devotion has been my constant source of motivation.

My heartfelt gratitude also goes to my sponsor and supervisor Associate Professor Ashraf Kassim, for his encouragements and support.

I would like to thank Mr. Tai Yu-Wing, who was Prof. Brown's PhD student, currently an Assistant Professor at Korean Advanced Institute of Science and Technology (KAIST), and who I worked closely with on the project.

I am also thankful to Associate Prof. Michael Brown from School of Computing, NUS, who enlightened me on new ways of tackling problems.

Content

Summary

Blur is a typical image artifact when pictures are taken with long exposure or with moving objects. A lot of research has been carried out on the topic of image deblurring. Many methods assume each pixel in the image undergoes the same amount of blur. However, the relative motion between a camera and the scene often causes spatially varying blur which is different at every pixel. Although it is observed that most of the image blurs are spatially variant in recent works (1)*, there is no existing model to represent or to reduce spatially varying blurs.*

This thesis addresses the problem of modeling and correcting spatially varying image blurs caused by rigid camera motion. It first presents a new Projective Motion Blur Model which models a blurred image as an integration of a sequence of projectively transformed clear images. These projective transformations describe the camera's motion during exposure. This formulation is derived according to the physical cause of the blurring effect and also offers a compact representation of the spatially varying blur. Subsequently, we propose the Projective Motion Richardson-Lucy (RL) algorithm to recover a clear image from an image undergoing spatially varying blur. We also incorporated state-of-the-art regularization image priors to improve deblurring results.

We further investigated the deblurring problem when multiple blurred images of the same scene are available. To make use of complementary information in different *images, these blurred images must be registered to each other. However, existing image registration algorithms only apply on clear images. Hence, we propose a method to align blurred images. The algorithm in this thesis uses frequency domain properties of the blurred images for alignment, which is both efficient and effective. The key feature of this method is that it could align motion blurred images.*

List of Figures

List of Tables

Chapter 1.

Introduction

1.1. Modeling and Correcting Spatially Varying Motion Blur

Motion blur is an artifact in photography caused by relative motion between the camera and an imaged scene during exposure. Ignoring the effects of defocusing and lens distortion, each point in the scene is imaged into its point spread function (PSF), sometimes also called the “blur kernel”, which describes the relative motion trajectory at that pixel position. Image deblurring, to recover a clear image from a blurry input, is a well studied problem and is known to be ill-posed that multiple different clear images can produce the same blurred image by convolution with an appropriate kernel. Nonetheless, this problem has received extensive study because of its utility in photography. A common assumption in motion deblurring is that the motion is spatially invariant so that each pixel, regardless of its position, shares the same global PSF. This assumption implies that the motion blur effect is caused by the camera’s translational motion. Then the blurred image can be expressed as a convolution between the clear image *I* and blur kernel *K*, plus noise *n*:

$$B = I \otimes K + n \tag{ 1-1}$$

The goal of image deblurring is to reconstruct the clear image *I* from a degraded image *B*. If both kernel and true image are unknown, the deblurring is called blind deconvolution; if only the true image is unknown and blur kernel is known, the deblurring technique is called non-blind deconvolution. Well known non-blind deblurring algorithms include Richardson-Lucy (RL) algorithm (2) (3) and Wiener filter (4). While these algorithms can produce deblurred results, they often suffer from artifacts such as ringing due to frequency loss and poor PSF estimation. Recent works focus on how to improve deblurring results by imposing various image priors to better estimate of the PSF and to suppress ringing artifacts. *Fergus et al* (5) proposed a blind deconvolution technique by using natural image statistics to estimate a more accurate PSF. Several regularization terms were used to reduce artifacts during deblurring process, *Dey et al* (6) introduced total variation regularization in the RL algorithm, *Levin et al* (7) introduced the gradient sparsity prior and *Yuan et al* (8) proposed bilateral regularization on progressive multi-scale approach. Shan et al (9) introduced regularization based on high order partial derivatives to constrain image noise. An evaluation on the state-of-the art deblurring algorithms was presented in *Levin et al.* (1). Other recent methods improved deblurring results relying on multiple images or special hardware setups. For example, *Yuan et al* (10) used noisy and blurred images pair, *Rav-Acha and Peleg* (11) and *Chen et al* (12) used blurred images pair. *Raskar et al* (13), (14) coded the exposure to make the PSF more suitable for deconvolution. *Ben-Ezra and Nayar* (15) (16) proposed a hybrid camera to capture a high resolution blurred image together with a low resolution video which was used to provide an accurate estimate of PSF.

All these works assumed the blur is spatially invariant. However, in many cases, the PSF varies spatially over the degraded image. Early work by *Sawchuk* (17) addressed spatially varying motion blur by image coordinate transformations, e.g. log-polar transformation, which transformed rotational motion into a spatially invariant representation and then solved it by conventional spatially invariant deconvolution algorithms. Another strategy was to segment the blurred image into multiple regions, each with a constant PSF (7) (18) (19) (20). However, this is difficult even in simple cases of rotation or zooming of the camera. *Shan et al* (21) handled spatially varying blur by restricting the relative motion to global rotation. *Joshi et al* (22) estimated a PSF at each pixel for images with defocus blur. Special hardware setups were also used to handle spatially varying blur. *Tai et al* (23) (25) extended the hybrid camera framework in (15) (16) to estimate a PSF at each pixel using optical flow.

On the other hand, as recently discussed by *Levin et al* (1), the global PSF assumption for blurred images caused by camera motion is inaccurate for practical purposes. In their experiments, images taken by a handhold camera with motion blur exhibited notable amounts of camera rotation which causes the motion blur to be spatially varying within the image. As a result, *Levin et al* (1) advocated the need for better motion blur models as well as image priors to improve the deblurred results.

In view of the need of dealing with spatially varying blur, we propose a new model which is able to describe spatially varying blur caused by camera motion during exposure. The new blur model is referred to as “Projective Motion Blur Model”. It represents the degraded image as an integration of the clear scene under a sequence of projective motions. The benefits of this model are that 1) it describes the spatially varying blur on the whole image and 2) it does not require storing a different PSF at every pixel to describe the blur. During deblurring, it is not necessary to segment the image into regions of locally invariant blur as in several previous works. However this blur model also has a drawback. Because this approach is not based on convolution with a PSF, it is very hard to be analyzed in the frequency domain.

Another contribution of this project is a non-blind deblurring algorithm according to the proposed blur model. We extended the Richardson-Lucy (RL) algorithm to correct spatially varying motion blur under this novel “Projective Motion Blur Model”. The modified RL algorithm is referred to as “Projective Motion Richardson-Lucy algorithm”. To reduce the ringing artifacts in the result, regularizations derived from image priors can be incorporated, as in the conventional RL algorithm.

In summary, this thesis focuses on developing the projective motion blur model and the non-blind deblurring algorithm. The assumptions are: 1) the motion of the camera is known; 2) the scene is distance and static, so that the camera motion satisfies our Projective Motion Blur Model. Potential methods for accurately estimating the projective motion are briefly discussed as well.

1.2. Blurred Image Alignment

Multiple blurred images of the same scene provide complementary information for deblurring, e.g. two blurred images were used to estimate a clear image in (12), and blurred/noisy image pair in (10). To overcome the difficulty of spatially varying blur, we propose to use multiple images as well. But this requires an alignment (registration) among these images in preprocessing stage. Image alignment is a fundamental task that is used widely in many multi-image applications, such as image stabilization, image enhancement, panorama, satellite photo stitching and many other graphical applications. Traditional image alignment techniques include pixel-based method and feature-based method. A comprehensive review of image alignment can be found in (25). However traditional image alignment methods only work with clear images. Applying them to blurred images will lead to incorrect alignment especially when the blur is significant.

The alignment on blurred images is non-trivial. One approach is to first deblur the image. *Rav-Acha et al* (26) (11) proposed that by limiting the blur kernel to be one dimension, alignment and deblurring can be solved simultaneously. *Flusser and Suk* (27) proposed a moment-based invariant to align blurred images with a centrally symmetric blur kernel. However, centrally symmetric kernel is more likely to happen in out-of-focus blur and is very rare in motion blur. So far there is no blur-invariant feature for an arbitrary shape kernel. To align arbitrary motion blurred image, *Yuan et al* (28) introduced a method to align Blurred/Non-Blurred image pair using the kernel's sparseness prior. *Yuan et al* (28) also observed the need for new method to align two blurred images.

A simple and effective method is presented in this thesis to align two blurred images. Neither deblurring the images nor estimating the blur kernel, our method makes use of the fact that there are three color channels in one color image and all channels must have the same blur kernel. Using this property, this project shows the alignment of two blurred images can be solved by aligning two unknown kernels in Fourier Domain. The blur kernel is assumed to be spatially invariant, and this method can align two blurred images up to any affine transformation in principal. In this thesis, only alignment under similarity transformation is tested, due to limitation of the computational power,

1.3. Organization of this Thesis

This thesis is organized in the following manner: Chapter 2 is a literature review on the research topics on image alignment and deblurring, Chapter 3 describes the Projective Motion Blur Model and Projective Motion RL algorithm with implementation details. Chapter 4 provides an analysis on deblurring result, including the convergence properties of proposed algorithm and its sensitivity to noise. Projective Motion RL algorithm's limitation and further research directions are also discussed in Chapter 4. Chapter 5 presents the Blurred image alignment algorithm with analysis on its accuracy. This thesis concludes in Chapter 6.

Chapter 2.

Literature review

2.1. Image Coordinates

To facilitate working on images with different resolutions, it is better to normalize the coordinates to [-1, 1] along x axis and [-a, a] along y axis, where ‘a’ is the aspect ratio. As shown in Figure 2-1, an image with size W × H and pixel coordinates $\bar{\boldsymbol{x}} = (\bar{x}, \bar{y})$ is mapped to normalized coordinates$\tilde{\boldsymbol{x}} = (x, y)$**,** or $\boldsymbol{x} = (x, y, 1)$ in homogeneous coordinates. Throughout this thesis, normalized homogeneous coordinates is used when referring to pixel coordinates.

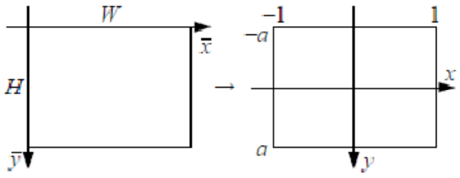


Figure 2-1: Mapping from pixel coordinates to normalized coordinates

2.2. Planar Transformations

Various 2D planar transformations are shown in Figure 2-2 (29).

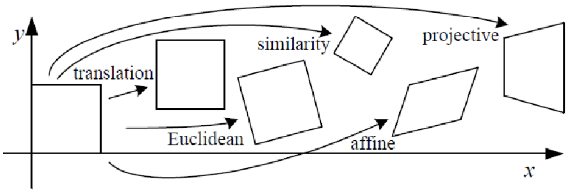


Figure 2-2 Planar transformations

Translation: 2D translations can be written as $\boldsymbol{x}' = \boldsymbol{x} + \boldsymbol{t}$ or

$$\boldsymbol{x}' = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{t} \\ \boldsymbol{0} & 1 \end{bmatrix} \boldsymbol{x} \tag{2-1}$$

where $\boldsymbol{I}$ is the 2×2 identity matrix. and $\boldsymbol{t} = (t_x, t_x)'$

Rotation + translation: This is also known as Euclidean transformation, since Euclidean distance is preserved. It can be written as

$$\boldsymbol{x}' = \begin{bmatrix} \boldsymbol{R} & \boldsymbol{t} \\ \boldsymbol{0} & 1 \end{bmatrix} \boldsymbol{x}$$

$$\text{where} \qquad R = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \tag{2-2}$$

is an orthonormal rotation matrix with $\boldsymbol{R}\boldsymbol{R}^{\boldsymbol{T}} = \boldsymbol{I}$ and $|\boldsymbol{R}| = 1$

Rotation + translation + scaling: Also known as the similarity transform, it preserves angles between lines which can be written as

$$\boldsymbol{x}' = \begin{bmatrix} \boldsymbol{sR} & \boldsymbol{t} \\ \boldsymbol{0} & 1 \end{bmatrix} \boldsymbol{x} = \begin{bmatrix} a & -b & t_x \\ b & a & t_y \\ 0 & 0 & 1 \end{bmatrix} \boldsymbol{x} \tag{2-3}$$

where $\boldsymbol{s}$ is an arbitrary scaling factor.

Affine Transformation: Parallel lines remain parallel under affine transformations.

$$\boldsymbol{x}' = \begin{bmatrix} a_{00} & a_{01} & a_{02} \\ a_{10} & a_{11} & a_{12} \\ 0 & 0 & 1 \end{bmatrix} \boldsymbol{x} \tag{2-4}$$

Projective Transformation: This is also known as perspective transformation or homography.

$$\boldsymbol{x}' = \begin{bmatrix} a_{00} & a_{01} & a_{02} \\ a_{10} & a_{11} & a_{12} \\ a_{20} & a_{21} & a_{22} \end{bmatrix} \boldsymbol{x} \tag{2-5}$$

2.3. Traditional Pixel-based Image Alignment (Registration)

Traditional Image alignment can usually be categorized into pixel-based and feature-based alignments. Feature-based registration is more robust in general. However, when dealing with blurred image, most feature-based methods will fail as it is often hard to extract features or to keep the features invariant under blur, e.g. Harris corners and SIFT are not blur invariant according to a recent work by ***Yuan el al*** (28) "so far there is no blur-invariant feature for an arbitrary shape kernel". Therefore this section will only talk about pixel-based alignment.

Error metrics is the simplest way to align two images with relative translational motion. Given a template image $\boldsymbol{I_0(x)}$ sampled with discrete pixel locations { $x_i = (x_i, y_i)$ } , the solution is to find the minimum of the cost function

$$E(t) = \sum_i f\big(I_1(x_i + t) - I_0(x_i)\big) = \sum_i f(e_i) \tag{2-6}$$

where e_i is called residual error. $f(\cdot)$ can be absolute, square or other robust functions.

Normalized cross-correlation is to find the maximized product of the two aligned image

$$E(t) = \frac{\sum_i |I_1(x_i + t) - \bar{I}_1| \cdot |I_0(x_i) - \bar{I}_0| =}{\sqrt{\sum_i |I_1(x_i + t) - \bar{I}_1|^2 \cdot |I_0(x_i) - \bar{I}_0|^2}} \qquad (2\text{-}7)$$

where $\bar{I}_t = \frac{1}{N}\sum_i I_t(x_i)$ is the mean of the image. This method is invalid when either image has zero variance, as in that case denominator becomes zero. In fact, performance degrades in the case of low-contrast and noisy images.

Fourier Domain

If two images are related by a translation $\boldsymbol{t} = (t_x, t_y)$**,** which can be represented by $I_1(x + t_x, y + t_y) = I_0(x, y)$. In Fourier Domain, it becomes

$$F_1(u, v) = e^{-j2\pi(ut_x + vt_y)} F_0(u, v) \qquad (2\text{-}8)$$

The cross-power spectrum is defined as

$$R(u, v) = \frac{F_1 F_0^*}{|F_1 F_0^*|} = e^{j2\pi(ut_x + vt_y)} \qquad (2\text{-}9)$$

where * denotes complex conjugate. The translation can be found by applying inverse Fourier transform.

$$(t_x, t_y) = arg \max_{x,y} F^{-1}(R(u, v)) \qquad (2\text{-}10)$$

This technique is called phase correlation (29).

If the two images also cantain rotations and scalings, it also can be solved in log-polar representations (30).

In case of rotation, $I_1(x, y) = I_0(x\cos\theta' + y\sin\theta' - t_x\,, y\cos\theta' - x\sin\theta' - t_y)$. In Fourier Domain it becomes

$$F_1(u, v) = e^{-j2\pi(ut_x + vt_y)} F_0(u\cos\theta\,' + v\sin\theta', v\cos\theta' - u\sin\theta') \qquad (2\text{-}11)$$

By taking the magnitudes of F_0 and F_1,

$$M_1(u,v) = M_0(u\cos\theta' + v\sin\theta', v\cos\theta' - u\sin\theta') \tag{2-12}$$

The magnitudes of the two spectra differ by a rotation. It can be converted to the translational case by using polar representation

$$M_1(\rho,\theta) = M_0(\rho,\theta-\theta') \tag{2-13}$$

In a similar way, the scaling case $I_1(x,y) = I_0(ax,by)$ transforms into Fourier domain as

$$F_1(u,v) = \frac{1}{|ab|}F_0(\frac{u}{a},\frac{v}{b}) \tag{2-14}$$

which can be converted to translation movement using log representation, ignoring the multiplication factor $1/ab$

$$F_1(u',v') = F_0(u'-a',v'-b') \tag{2-15}$$

where $u' = \log(u)$, $v' = \log(v)$, $a' = \log(a)$ and $b' = \log(b)$.

Combining the translation, rotation and scaling (by single factor, i.e. $\mathrm{a} = b$), the Fourier translation can be related by power magnitude in log-polar representation as:

$$M_1(\rho',\theta) = M_0(\rho'-a',\theta-\theta') \tag{2-16}$$

where $\rho' = \log(\rho)$ and $a' = \log(a)$. After removing rotation θ and scaling factor a in the images, translation can be recovered by phase correlation easily.

The advantage of Fourier domain alignment is that it is more computationally efficient than that in pixel domain, especially after rotation and scaling transformation, but the two images have to be largely overlapped to make this algorithm work.

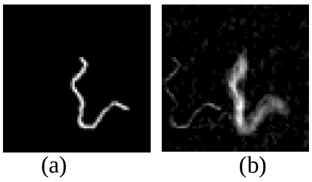
2.4. Non-Blurred /Blurred Image Pair Alignment.

As traditional alignment only works on clear images, *Yuan et al* (28) proposed this method to align a Clear/Blurred image pair, under the assumption that the image blur is caused by camera motion instead of out of focus blur. The main idea was to use the kernel's sparseness prior to solve the alignment. Because clear image I and blurred image B were known, the blur kernel K can be solved by

$$K = arg\min_{K} \|B - I \otimes K\|^2 + \|K\|^2$$

$$\text{subject to } k_i \geq 0 \text{ and } \sum_{i} k_i = 1 \qquad (\ 2\text{-}17)$$

If the blur is caused by camera motion and the alignment is correct, the kernel K should be sparse, as shown in Figure 2-3 (a). On the other hand, if alignment is incorrect, e.g. with an error of 0.7 degree, then estimated kernel by Equation (2-17) could be much more noisy, as shown in Figure 2-3 (b). It was also claimed that the true alignment produces the most sparse kernel (28). The alignment can then be solved by a coarse-to-fine brute force search. In (28), the alignment problem was resolved under affine transform by 4D search.



(a) (b)

Figure 2-3. Estimated kernel

(a) by correct alignment, (b) with alignment error of 0.7° in rotation

2.5. Richardson-Lucy Deconvolution

According to (2) (3), given the motion blurred image B and blur kernel K, the clear image I can be computed by Bayesian estimation. As all image intensities can be normalized to range between 0 and 1, and the kernel entries sum to 1 (Figure 2-4). The pixel intensities can be treated as probabilities. As a result the blurred image $B = I \otimes K$ can be represented as

$$P(B_x) = \sum_y P(B_x|I_y)\,P(I_y) \tag{2-18}$$

where $P(B_x) = B(x)$, $P(I_y) = I(y)$ and $P(B_x|I_y) = K(x,y)$ is the value at y position in the blur kernel centered at x position as shown in Figure 2-4

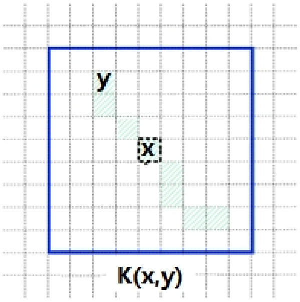


Figure 2-4 Blur kernel K for pixel location x, $\sum_y K(x,y) = 1$

Therefore the pixel value of the clear image $P(I_x)$ can be computed according to pixel values of the blurred image $P(B_y)$ by the following formula.

$$P(I_x) = \sum_y P(I_x|B_y)\,P(B_y) \tag{2-19}$$

By Bayes' rule, $P(I_x|B_y)$ can be computed and $P(I_x)$ can be written as:

$$P(I_x) = \sum_y \frac{P(B_y|I_x)P(I_x)}{\sum_z P(B_y|I_z)\,P(I_z)} P(B_y) \tag{2-20}$$

Both sides of this equation contain $P(I_x)$, which complicates the computation. To break the dependency, a general and acceptable practice is to make the best out of a bad situation and to use a current estimation of $P(I_x)$ to approximate $P(I_x|B_y)$. Hence an iterative algorithm is used as:

$$P^{t+1}(I_x) = \sum_y \frac{P(B_y|I_x)P^t(I_x)}{\sum_z P(B_y|I_z)\,P^t(I_z)} P(B_y)$$

$$= P^t(I_x)\sum_y P(B_y|I_x)\frac{P(B_y)}{\sum_z P(B_y|I_z)\,P^t(I_z)} = P^t(I_x)\sum_y P(B_y|I_x)E_y \qquad (2\text{-}21)$$

where t is an iteration index. Under the conventional blur model that $B = I \otimes K$, $P(B_y|I_x) = K(y,x)$ is the PSF at location y and $\sum_x K(y,x) = 1$, $B' = \sum_z P(B_y|I_z)\,P^t(I_z) = I^t \otimes K$ is the prediction of the blurred image according to estimated clear image I^t. $E = \frac{B}{B'}$ is the residual error (pixel-wise division) between the real blurred image and the predicted blurred image. $\sum_y P(B_y|I_x)E_y$ is the amendment term according to the residual, which can be computed by $K * E$. Here $\otimes$ and $*$ are convolution and correlation operators respecively. Hence the previous equation can be written as:

$$I^{t+1} = I^t \times K * \frac{B}{I^t \otimes K} \qquad (2\text{-}22)$$

The Richardson-Lucy algorithm essentially minimizes the difference between the original blurred image B and the predicted blurred image B' according to the Poisson noise model:

$$\arg\min_{\mathrm{I}} N(B', B) \qquad (2\text{-}23)$$

where $N(\lambda,k) = e^{-\lambda}\lambda^k/k!$, $\lambda = B'$and $k = B$ is the probability distribution function (pdf). Image noise is further assumed to be independent and identically distributed (i.i.d.). In (3), Lucy showed that based on Poisson noise distribution, $K * \frac{B}{I^t \otimes K}$ converged to 1 and hence $I^{t+1} = I^t$ as $t \to \infty$, which is taken as a proof that the Richardson-Lucy algorithm converges.

Other noise models, such as Gaussian model, also can be used. In this project, the approach used in (23) (31) is used to assume Gaussian noise model, taking log on both side of equation (2-22) and re-define the variables to obtain

$$I^{t+1} = I^t + K * (B - I^t \otimes K) \tag{2-24}$$

The convergence of equation (2-24) was proved in (31).

2.6. Regulations

The result derived in the previous section is a maximum likelihood with respect to Poisson or Gaussian noise models and imposes no regularization on the solution. The main problem is artifacts such as ringing. Recent deblurring works (6), (32), (8) have shown that imposing certain images priors can significantly reduce the ringing artifact.

Total variation

The total variation (TV) regularization was introduced by Dey et al (6). The purpose of this regularization is to suppress image noise amplified during the deconvolution process by minimizing the magnitude of gradients in deblurred image:

$$R_{TV}(I) = \int \sqrt{\|\nabla I(x)\|^2} dx \qquad (\ 2\text{-}25\ )$$

where $\nabla I(x)$ is the first order derivative of $I(x)$ in x and y directions. Substituting this regularization term into Richardson-Lucy algorithm, the updating rule Equations (2-22), (2-24) become

$$I^{t+1} = \frac{I^t}{1 - \lambda \nabla R_{TV}(I)} \times K * \frac{B}{I^t \otimes K} \qquad (\ 2\text{-}26\ )$$

and

$$I^{t+1} = I^t + K * (B - I^t \otimes K) + \lambda \nabla R_{TV}(I) \qquad (\ 2\text{-}27\ )$$

where $\nabla R_{TV}(I) = -\nabla \frac{\nabla I^t}{|\nabla I^t|}$ and λ is a parameter controlling the weight of regularization. As reported in (6), $\lambda = 0.002$ was used in their experiments.

Laplacian Regularization

The Laplacian regularization, which is also called the sparseness prior, states that for the natural images their histogram of gradient magnitudes should follow a heavy-tailed distribution such as the Laplacian distribution. The Laplacian regularization, suggested by (32) takes the following form:

$$R_L(I) = exp\,(-\frac{1}{\eta}|\nabla I|^d) \qquad (\ 2\text{-}28\ )$$

where d is a parameter controlling the shape of distribution, and term η is the variance of the image noise. In (32), d is set to 0.8 and $\eta = 0.005$.

The updated rule is similar to Equations (2-26) and (2-27), by replacing $\nabla R_{TV}(I)$ with $\nabla R_L(I)$

$$\nabla R_L(I) = -\frac{1}{\eta} exp\,(-\frac{1}{\eta}|\nabla I|^d)|\nabla I|^{d-1}\nabla^2 I \tag{2-29}$$

The implementation of (32) used a slightly different weighting scheme:

$$\nabla R_L(I) = -\frac{1}{|\nabla I|^{d-2}}\nabla^2 I \tag{2-30}$$

The effects of the two weighting schemes in Equations (2-29) and (2-30) are similar with larger regularization weight given to smaller gradient and vice versa. The effects of Laplacian regularization are to suppress noise and to reduce small ringing artifacts. It produces sharper edges comparing with the total variation method which tends to produce over-smooth results.

Bilateral regularization

In order to suppress ringing artifacts while preserving sharp edges, *Yuan el al.* (8) proposed an edge preserving bilateral regularization cost:

$$R_B(I) = \sum_{\boldsymbol{x}} \sum_{\boldsymbol{y}\in\boldsymbol{\Omega}(\boldsymbol{x})} g_s(\|\boldsymbol{x}-\boldsymbol{y}\|^2)(1-g_r(\|I(\boldsymbol{x})-I(\boldsymbol{y})\|^2)) \tag{2-31}$$

where $\Omega(\mathrm{x})$ is the local neighborhood of x, and $\mathrm{g_s}(\cdot)$, $\mathrm{g_r}(\cdot)$ are two Gaussian functions with zero mean and variance of σ_s^2 and σ_r^2 respectively.

The updated rule is also similar to Equations (2-26) and (2-27), by replacing $\nabla R_{TV}(I)$ with$\nabla R_B(I)$. According to (8) $\nabla R_B(I)$ is derived as

$$\nabla R_L(I) = \sum_{y\in\Omega(x)} I_y^d - I_y^d D_y \tag{2-32}$$

$$I_y^d(\boldsymbol{x}) = g_1(\|\boldsymbol{x}-\boldsymbol{y}\|^2)g_2(\|I(\boldsymbol{x})-I(\boldsymbol{y})\|^2)\cdot\frac{I(\boldsymbol{x})-I(\boldsymbol{y})}{\sigma_r} \tag{2-33}$$

The term D_y is a displacement operator which shifts the entire image I_y^d by the displacement vector $(\boldsymbol{x}-\boldsymbol{y})$. Equation (2-32) actually calculates the overall intensity difference within a local neighborhood; therefore it is also referred to as "long range gradient". The bilateral regularization reduces ringing artifacts significantly in the final deblurring result.

2.7. Deblurring Blurred/Noisy Image Pair

This method was introduced by (10). Because the image blur is usually caused by long exposure time under low light environment, it was proposed to take one more image with higher ISO and shorter exposure time. Shorter exposure will provide a sharper image. However, high ISO often results in very noisy image because noise is also amplified. By using both blurred and noisy images, a high quality image can be produced. A clear image can be represented as the noisy image I_N with a lost detail layer ΔI , that is

$$I = I_N + \Delta I \tag{2-34}$$

The the blurred image is

$$B = I \otimes K \tag{2-35}$$

By combining Equations (2-34) and (2-35) , the following is obtained

$$\Delta B = \Delta I \otimes K \tag{2-36}$$

where $\Delta B = \mathrm{B} - I_N \otimes K$.

By using $I = I_N$ as initial value, blur kernel K can be estimated by Equation (2-17). ΔI in Equation (2-36) is the only unknown and can be solved by Richardson-Lucy deconvolution. The final result can hence be solved using iterative method. In (10), the noisy image was also used to perform a gain control and to add details. Therefore, instead of Equation (2-35), the update rule becomes

$$I^{t+1} = I^t + I_{gain}\Delta I^{t+1} \tag{2-37}$$

$$I_{gain} = (1-\alpha) + \alpha\sum_l \left\|\nabla I_N^l\right\| \tag{2-38}$$

Here, according to (10), ∇I_N^l is the gradient of the denoised image at the *l*th level of Gaussian pyramid with standard deviation of 0.5 and $\alpha = 0.2$, ΔI^{t+1} is solved by Equation (2-36) using Richardson-Lucy algorithm. $I^0 = I_N$

2.8. Deblurring Blurred/Blurred Images

This method was introduced by *Chen et al* (12). Two blurred images will provide more information and constraints than a single image. Their approach also used an iterative method by first estimating blur kernels, then estimating clear image and finally back to refine blur kernels.

The initial kernel estimation for two blurred images can be obtained by minimizing the energy function

$$E(K_1,K_2) = E_d(K_1,K_2) + \alpha\sum_{i=1}^{2} E_{reg}(K_i) \tag{2-39}$$

$$E_d(K_1, K_2) = \rho(B_2 \otimes K_1 - B_1 \otimes K_2)$$

Here $\rho(\mathrm{r}) = \log(1 + \frac{1}{2}\left(\frac{\mathrm{r}}{\varepsilon}\right)^2)$ is a robust cost function. $E_{reg}(K_i)$ is the kernel prior, including sparseness and continuity. More detail about $E_{reg}(K_i)$ can be found in (12).

With the kernel estimation, clear image can be reconstructed by minimizing deconvolution energy

$$E(I) = \sum_{i=1}^{2} \rho(I \otimes B_k - B_k) + E_r(I) \tag{2-40}$$

where $E_r(I)$ is the regularization term, and sparseness prior as (12) is used. The final result was obtained by repeating the kernel estimation and clear image reconstruction. This method can be applied to multiple input images as well. However, to make this algorithm work well, an accurate alignment on input blurred image is essential.

2.9. Hybrid Camera

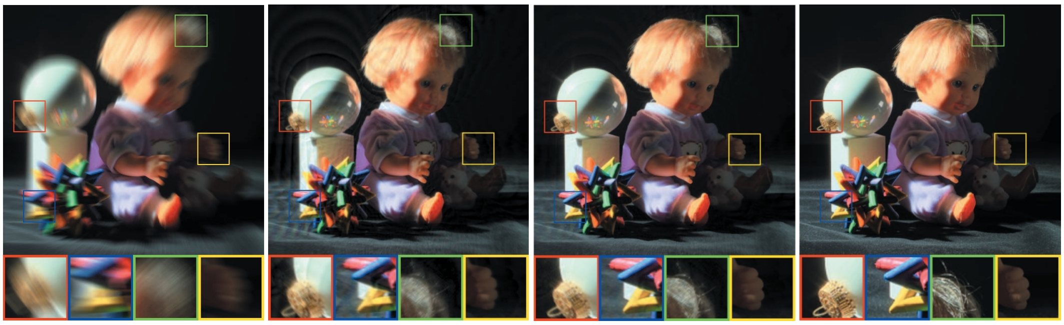
One of the most difficult tasks in image deblurring is to estimate the blur kernel. *Ben-Ezra et al* (15) (16) proposed that the blur kernel resulted from camera motion can be captured by using auxiliary hardware. The idea is to capture a video with low resolution but fast frame-rate during the image exposure so that the camera motion can be captured and blur kernel can be reconstructed. It was originally designed to solve spatially invariant deblurring problem. In latest development, (23) (25) extended the hybrid frame work to solve spatially varying blur by estimating a PSF at each pixel.

Chapter 3.

Projective Motion Richardson-Lucy (RL) algorithm

A recent research work (1) found the blur often varies over the image. Hence, we propose a novel blur model in this chapter to handle spatially varying blur. At the same time, we also derive a non-blind deblurring algorithm by extending the Richardson-Lucy algorithm.

Figure 3-1 is an example of spatially varying blur and the deblurring result by PMRL.



(a) RMS 33.4480 (b) RMS 7.3261 (c) RMS 3.3100 (d) Ground Truth

Figure 3-1 (a) An image degraded by spatially varying motion blur, (b) result by basic PMRL algorithm, (b) Result with regularizations. (d) Ground truth. The RMS errors are also shown below each image.

3.1. Projective Motion Blur Model

This section describes the projective motion blur model and later this model is used to derive the deblurring algorithm.

An image consists of many pixels and each pixel's intensity is determined by the amount of light received on the sensor (CCD/CMOS) over the exposure time. This can be represented as

$$I(x) = \int_0^T \Delta I(x,t)dt \approx \sum_{i=1}^{N} \Delta I(x,t_i) \qquad (3\text{-}1)$$

where $I(x)$ is the image recorded after exposure, $\Delta I(x,t)$ is an image captured by the sensor within infinite small time interval dt at the time instance t, $[0,T]$ is the total exposure time and x is a 3x1 vector indicating the homogenous pixel coordinates. In this model, if N is large enough then the difference between continuous integration and the discrete integration can be neglected. When there is no relative motion between the camera and the scene during the exposure, $\Delta I(x,t_0) \cong \Delta I(x,t_1) \cong \cdots \cong \Delta I(x,t_N)$ and a clear image is generated $I(x) \cong N\Delta I(x,t_0) \triangleq I_0(x)$. When there is relative motion, $I(x)$ becomes blur as a result of the summation of multiple unaligned images $\Delta I(x,t_i)$. As illustrated in Figure 3-2, motion blur is generated by integrating a clear image signal over the exposure time.

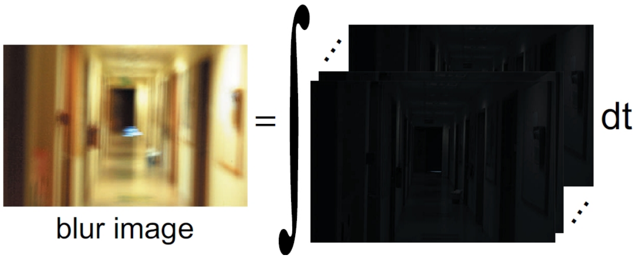


Figure 3-2 Blurred image is considered as the integration of an image scene under projective motion

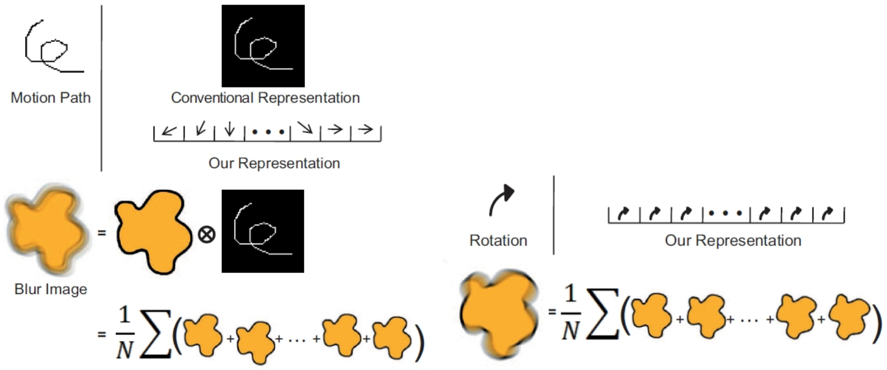
For a rigid motion and static scene, it is reasonable to assume the relative motion causes a 2D projective transformation in the image plane. i.e. $\Delta I(x, t_i) = \Delta I(h_i x, t_{i-1})$. Here h_i is the homography defined by a 3×3 non-singular matrix. Supposing all h_i are known, we can express $\Delta I(x, t_i)$ in $I_0(x)$ using the following formula

$$\Delta I(x, t_i) = \Delta I\left(\prod_{j=1}^{i} h_j\, x, t_0\right) = \frac{1}{N} I_0(H_i x) \tag{3-2}$$

where $H_i = \prod_{j=1}^{i} h_j$ is also a homography. Hence, the projective motion blur can be expressed as

$$B(x) = \sum_{i=1}^{N} \Delta I(x, t_i) = \frac{1}{N} \sum_{i=1}^{N} I_0(H_i x) \tag{3-3}$$

where $B(x)$ is the motion blurred image and $I_0(x)$ is the clear image to be estimated later in the deblurring process. According to this model, the blurred image is generated by averaging multiple clear images, where each image is a projective transformed version of the clear image $I_0(x)$. Figure 3-3 illustrates the relationship between the projective motion blur model and the conventional representation.


Motion Path
Conventional Representation
Our Representation
Blur Image
=
⊗
$= \frac{1}{N}\sum(+ + \cdots + +)$
Rotation
Our Representation
$= \frac{1}{N}\sum(+ + \cdots + +)$

(a) Spatially invariant blur in both blur models (b)Spatially variant blur in Projective motion blur model

Figure 3-3 Compare of projective motion blur model and conventional model.

The conventional spatial invariant representation is a special case of the projective motion blur model, in that each h_i only contains a translation. The conventional PSF can have intensity variations, which are modeled by the different magnitude of translations in h_i. In the planar translational motion, the conventional kernel-based model is a significantly more compact representation for the motion blur. However, in the case of other motions, e.g. rotation and zooming, the projective motion blur model is compact and intuitive.

3.2. Projective Motion Richardson Lucy (RL) Algorithm

Projective Motion RL algorithm is derived from the conventional RL algorithm. As describe in Section 2.5. , original RL algorithm minimizes the difference between the original blurred image B and the predicted blurred image B' according to the Poisson noise model (3)(4).

$$\arg\min_{\mathrm{I}} N(\|B - B'\|^2) \qquad (3\text{-}4)$$

Given the motion blurred image B, the clear image I can be computed by Bayesian estimation. Because all image intensities are normalized to range from 0 to 1, and the kernel entries sum to 1, the pixel intensities can be regarded as probabilities. Therefore the pixel value *I(x)* can be represented as probability $P(I_x)$, which can be computed according to pixel values of the blurred image $P(B_y)$ by the following formula.

$$P(I_x) = \sum_y P(I_x|B_y)\,P(B_y)$$
$$= \sum_y \frac{P(B_y|I_x)P(I_x)}{\sum_z P(B_y|I_z)\,P(I_z)} P(B_y) \qquad (\text{ 3-5 })$$

The solution can be obtained by an iterative update procedural algorithm

$$P^{t+1}(I_x) = \sum_y \frac{P(B_y|I_x)P^t(I_x)}{\sum_z P(B_y|I_z)\,P^t(I_z)} P(B_y)$$
$$= P^t(I_x)\sum_y P(B_y|I_x)\frac{P(B_y)}{\sum_z P(B_y|I_z)\,P^t(I_z)}$$
$$= P^t(I_x)\sum_y P(B_y|I_x)E_y^t \qquad (\text{ 3-6 })$$

To understand Equation (3-6), $B' = \sum_z P(B_y|I_z)\,P^t(I_z)$ is the prediction of a blurred image according to the current estimation of clear image I^t. $E_y = B_y/B'_y$ is the residual errors. $\sum_y P(B_y|I_x)E_y^t$ is the amendment term according to the residual, which integrates the residual errors distributed within the local windows of PSF according to $P(B_y|I_x)$. Note that the index of summation is different from that of the generation of predicted blurred image.

In fact (3-6) does not assume the PSF to be the same at each pixel. When the PSF varies spatially and the variations satisfy Projective Motion Blur model, it satisfies

$$P(B_y|I_x) = \begin{cases} \frac{1}{N}, & y = H_i x \\ 0, & otherwise \end{cases} \qquad (\text{ 3-7 })$$

where $\sum_x P(B_y|I_x) = \sum_{i=1}^{N} P(y = H_i x)$=1. Note that in this definition, $P(B_y|I_x)$ does not necessary correspond to a discrete integer coordinate of a pixel location. Rather the location can be fractional according to the motion defined by $y = H_i x$. Therefore, bicubic interpolation is used to compute the value of $I(H_i x)$. Hence $\mathrm{B}' = \sum_{i=1}^{N} P(y = H_i x) I(x) = \frac{1}{N}\sum_{i=1}^{N} I(H_i x)$ which is the equation of the projective motion blur model Equation (3-3) for generating the prediction of blurred image according to the clear image I given camera motion in terms of H_i

After substitution and simplification, the update rule for the projective motion blur model becomes:

$$I^{t+1}(\mathrm{x}) = I^{t}(\mathrm{x}) \times \frac{1}{N}\sum_{i=1}^{N} E^{t}(H_i^{-1}x) \qquad (3\text{-}8)$$

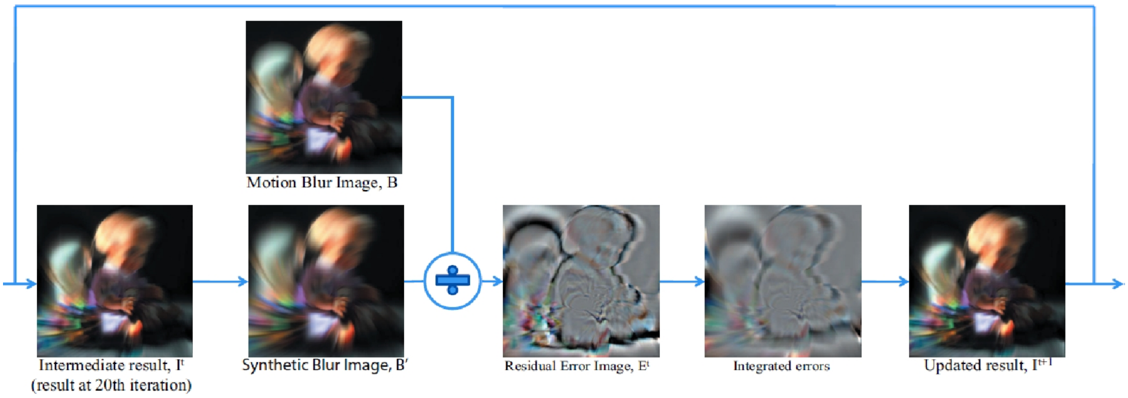


Figure 3-4 Overview of Projective Motion RL algorithm. (Eqn (3-8)

where $E^t(x) = \frac{B(x)}{B'(x)} = \frac{B(x)}{\frac{1}{N}\sum_{i=1}^{N} I^t(H_i x)}$. Equation (3-8) is demonstrated graphically in Figure 3-4.

Similarly, if the image noise model follows a Gaussian distribution, our iterative algorithm becomes

$$I^{t+1}(\mathrm{x}) = I^{t}(\mathrm{x}) + \frac{1}{N}\sum_{i=1}^{N} E(H_i^{-1}x) \qquad (3\text{-}9)$$

where $E(x) = B(x) - B'(x)$. This approach has essentially replaced the global correlation and convolution operators in the original RL algorithm with a sequence of forward projective motions and their inverses. It handles the entire image under a global rigid motion, which is spatially varying. However on scale of each pixel, it is equivalent to the original RL algorithm, the only difference is that each pixel has a unique PSF. Therefore the convergence should remain the same as original RL algorithm.

3.3. Adding Regularization

Similar to the conventional RL algorithm, the result derived in the previously section is a maximum likelihood estimation with respect to Poisson or Gaussian noise model and imposes no regularization on the solution. However, due to the similarity, regularization can be easily added into Projective Motion RL algorithm as follows.

In Poisson Noise distribution:

$$I^{t+1} = \frac{I^{t}}{1 - \lambda \nabla R_{reg}(I)} \times \frac{1}{N}\sum_{i=1}^{N} E^{t}(H_i^{-1}x) \qquad (3\text{-}10)$$

In Gaussian Noise distribution:

$$I^{t+1} = I^t + \frac{1}{N}\sum_{i=1}^{N} E^t(H_i^{-1}x) + \lambda\nabla R_{reg}(I) \qquad (\ 3\text{-}11\ )$$

R_{reg} can be several different regularization priors such as Total Variation, Laplacian regularization or Bilateral regularization. The derivation of ∇R_{reg} can be found in Section 2.6.

3.4. Motion Estimation

This section describes two methods to estimated projective motion during the exposure time. The first one makes use of auxiliary hardware during image capturing, while the second one assumes uniform motion. General algorithm for motion estimation is to be developed in future research.

Hybrid Camera

A simple and direct method to estimate camera motion in terms of homographies during the exposure is to use the hybrid camera (16)(17)(24), as it can capture an auxiliary high frame rate low resolution video. In the auxiliary video, motion trajectory between each adjacent frame can be computed by image registration algorithm, and can be used as h_i as described in equation (3-2) .

Uniform Motion Assumption

In this method, a constant projection motion over the exposure time is assumed.

Hence, $h_1 = h_2 = \cdots = h_n$. According to the definition of $H_i = \prod_{j=1}^{i} h_j$,

$$h_i = \sqrt[N]{H_N} \qquad 1 \le i \le N \tag{3-12}$$

Thus h_i is computed as the N-th matrix root of H_N, and the estimation of series of h_i is reduced to the estimation of H_N, which can be estimated by techniques used in (22) that simply relies on users to supply image correspondences to establish the transformation. Another highly promising technique was proposed by *Dai and Wu* (33) to use the blurred objects alpha matte to estimate H_N. In my experiments, the approach in (22) was used.

Blind Deblurring on Non-Uniform Rotational Motion

This method was purposed by *O. Whyte et al* (39). By simplifying the Projective Motion Blur Model to purely rotational motion, blind deblurring algorithm in (6) can be used to solve the motion trajectories.

If the motion only contains rotational motion, the homography H can be express as:

$$H = KRK^{-1} \tag{3-13}$$

where K is camera's intrinsic matrix and R describes the rotation in "angle-axis" representation by the vector $\boldsymbol{\theta} = (\theta_x, \theta_y, \theta_z)^{\mathrm{T}}$. R becomes

$$\mathrm{R}(\theta) = \mathrm{e}^{[\boldsymbol{\theta}]_\times}$$

$$[\boldsymbol{\theta}]_\times = \begin{bmatrix} 0 & -\theta_z & \theta_y \\ \theta_z & 0 & -\theta_x \\ -\theta_y & \theta_x & 0 \end{bmatrix} \tag{3-14}$$

The blur model then becomes

$$\begin{aligned} B(x) &= \int_0^T I(H_t x)dt \\ &= \int_R I(H_{\boldsymbol{\theta}} x)\omega(\boldsymbol{\theta})d\boldsymbol{\theta} \end{aligned} \qquad (\text{ 3-15})$$

where $\omega(\boldsymbol{\theta})$ is the weight value, which corresponds to the time duration at the orientation $\boldsymbol{\theta}$**.** In Eqn (3-16), the only unknown variable is $\omega(\boldsymbol{\theta})$, which is three dimensional in space $(\theta_x, \theta_y, \theta_z)$. Similar to traditional PSF, sparseness and continuous priors also apply to $\omega(\boldsymbol{\theta})$. Then blind deblurring method in (6), which originally solves PSF in 2D, can be used to solve the weight $\omega(\boldsymbol{\theta})$ in 3D with proper modifications regarding to this new blur model.

Chapter 4.

Experiment Results and Discussion

4.1. Convergence Analysis

While the conventional RL algorithm guarantees convergence, convergence of Projective motion RL algorithm was examined empirically. At each iteration, the RMS error of the current result against ground truth image was computed. We ran our algorithm for a total of 5,000 iterations for each case. The convergence rate of our projective motion RL for both the Poisson noise distribution and the Gaussian noise model were compared. Figure 4-1 shows the graphs plotting the RMS error against the number of iterations.

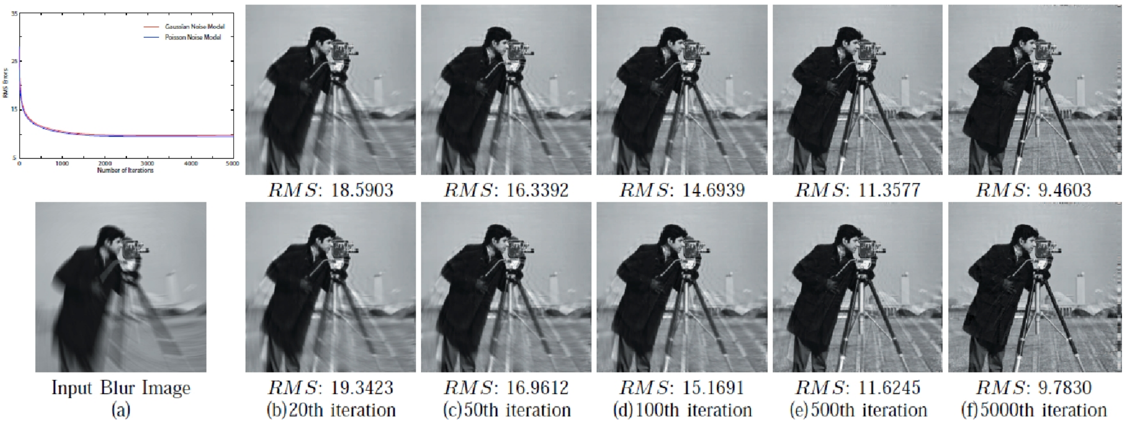


Figure 4-1 Convergence rates of Projective Motion RL algorithm
(a) the plot of RMS error against number of iterations and the input of motion blurred image.
(b-f) Intermediate results at different iterations, Poisson noise Model is on top.

Typically, the result converges within 300 to 400 iterations for both the Poisson noise model and Gaussian noise model with the RMS errors difference between successive iterations less than 0.01. The Poisson noise model produce slightly better result in terms of RMS errors than that of Gaussian noise model, however, the two results are visually indistinguishable. As the number of iterations increases, the difference of RMS errors between successive iterations decreases. However, after 500 iterations, the quality in the deblurred result becomes unnoticeable.

4.2. Qualitative Analysis

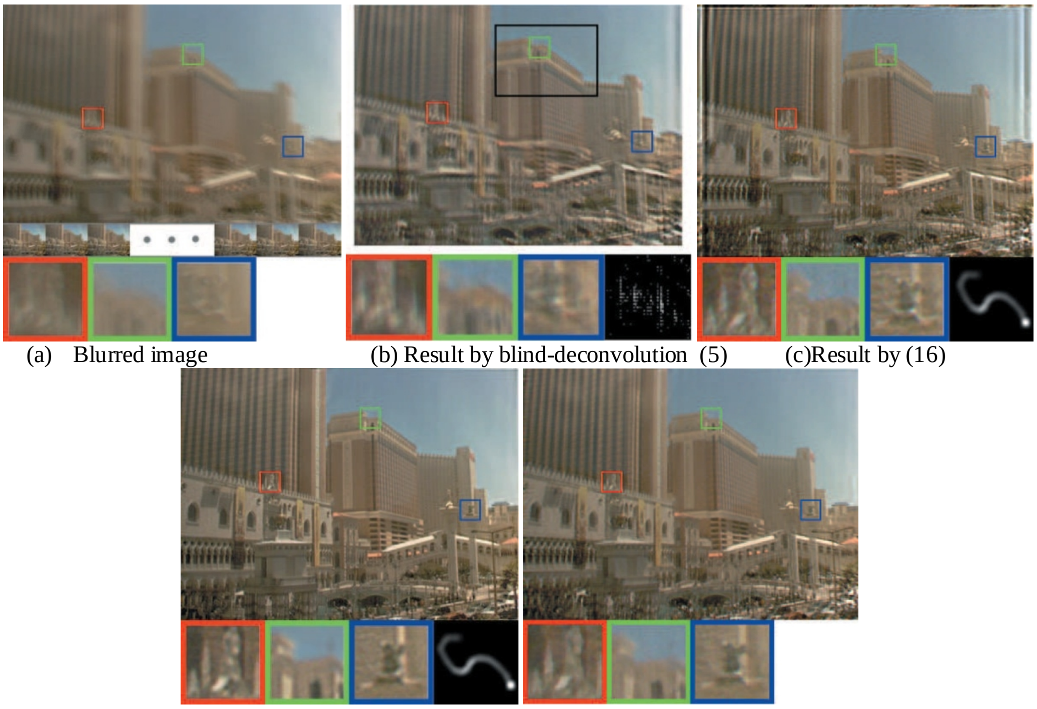
To demonstrate the effectiveness of the Projective Motion RL algorithm, a series of experiments were carried out on synthetic images, real images and one hybrid camera data set. Figure 3-1 already showed a synthetic example of spatially varying motion blur where h_i is known. The Projective Motion RL algorithm is effective in dealing with the rigid spatially varying motion blur, especially with regularization added. To further evaluate the algorithm, a testing set with 4 different images is tested: Mandrill, Lena, Cameraman and Fruits. The Mandrill example contains significant high frequency details in the hair regions. The Lena example contains high frequency details, smooth regions and step edges. The Cameraman is a gray level image. The Fruits example also contains many high frequency details and smooth regions. Each testing image is blurred by fifteen different motions from simple translational motion, rotation, zooming to full projective motion. Table 4-1 shows, from left to right, the resulting RMS of the blurred image, deblurred image with basic Projective Motion

RL algorithm and deblurred result with regularization. Some image results of these test cases are shown in the Appendix B.

	Mandrill			Lena			Cameraman			Fruits		
	I:RMS	B:RMS	R:RMS	I:RMS	B:RMS	R:RMS	I:RMS	B:RMS	R:RMS	I:RMS	B:RMS	R:RMS
T1	30.65	14.56	12.94	25.79	7.06	4.46	27.84	8.40	4.79	18.68	6.08	3.33
T2	36.09	15.20	13.70	29.08	8.74	6.07	29.25	8.82	5.61	22.41	7.95	5.46
T3	36.86	6.58	5.60	29.79	5.36	3.39	40.06	9.31	5.57	22.62	3.99	2.86
T4	36.67	16.32	15.00	32.88	9.92	6.84	34.35	11.44	7.19	24.12	8.53	5.82
T5	36.88	12.83	8.99	31.04	6.22	4.40	38.87	8.53	5.89	23.11	5.31	3.76
T6	37.24	14.11	11.85	32.74	9.70	7.88	37.63	12.48	9.80	24.25	8.85	7.68
T7	38.37	16.56	14.32	34.97	11.72	9.22	36.61	11.38	11.72	25.07	9.37	7.98
T8	27.08	19.95	19.44	21.28	11.47	10.93	12.21	6.27	5.41	15.51	9.01	8.13
T9	33.24	19.09	18.98	29.66	11.33	10.78	29.37	9.57	7.89	21.80	9.68	8.80
T10	36.27	17.35	17.26	29.86	11.02	10.54	29.72	8.69	7.28	23.14	9.50	8.54
T11	37.37	13.26	12.07	33.10	8.73	7.70	41.99	7.34	6.33	23.52	7.09	6.54
T12	37.28	18.87	18.85	34.72	12.37	11.76	34.52	11.22	8.80	26.03	10.53	9.50
T13	38.09	14.75	13.95	34.95	10.83	10.11	40.59	6.56	6.11	24.86	7.09	6.70
T14	39.31	17.01	15.30	37.37	9.51	8.51	39.11	10.73	9.63	26.94	11.11	10.41
T15	40.18	17.44	16.16	38.62	12.42	11.46	37.71	10.29	10.18	28.03	12.16	11.72

Table 4-1 pixel RMS error for different example under different motion in synthetic test
I:RMS is RMS of blurred image, **B:RMS** is deblurred using basic PMRL and **R:RMS** is with regularization

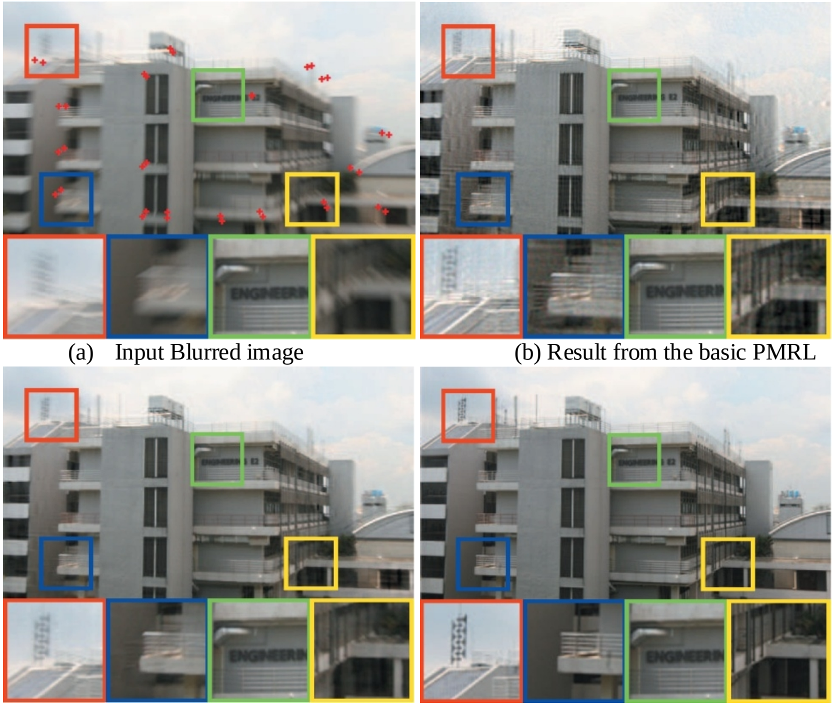
To test the Projective Motion RL algorithm on real images, the motion trajectories are obtained by two different approaches in Section 3.4. Figure 4-2 shows an example of blurred image taken by hybrid camera obtained from (23). For this example, it shows the Projective Motion RL algorithm is more accurate than the conventional method, as there are some rotational motions involved in the motion which causes the PSF to vary slightly across the whole image. The result is comparable with (23), in which the optical flow for each pixel was calculated as PSF and low resolution images were also used as regularizations. Meanwhile Projective Motion RL algorithm obtains the result only using the motion trajectory which is recovered from the low resolution images.


(a) Blurred image (b) Result by blind-deconvolution (5) (c)Result by (16)

(d) Result by (23) (e) Result by Projective Motion RL

Figure 4-2. Image Deblurring on Hybrid Camera data set, with Close-up views and estimated blur kernels

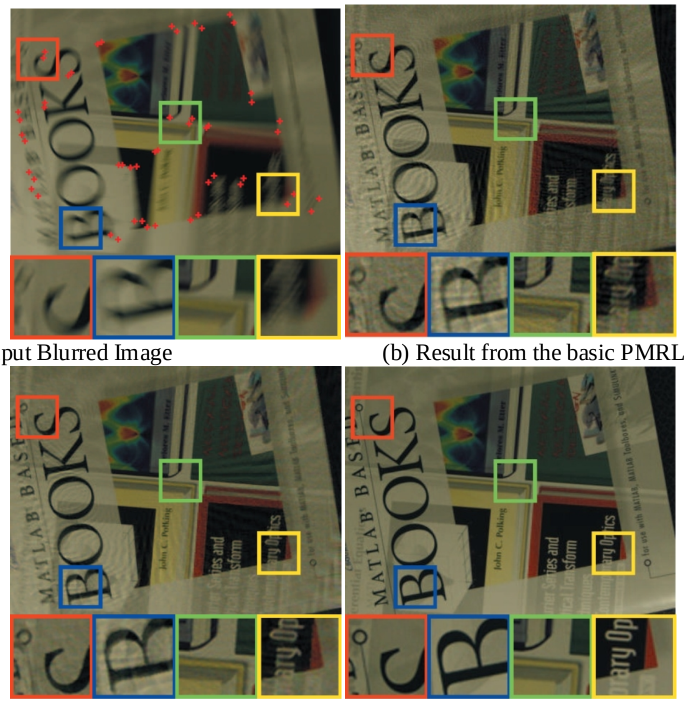
Figure 4-3 and Figure 4-4 show two real examples with zooming motion and rotational motion respectively, the overall blur motion $\boldsymbol{H_N}$ is obtained by fitting the transformation matrix with user manually providing correspondences as shown in Figure 4-3a. The sequence of h_i is then computed by assuming the motion to be uniform. The results of the Projective Moton RL algorithm without and with regularization are shown in (b) and (c) respectively. The suggested ground truth images are shown in (d) for comparison. In these real image examples, the ringing artifacts are larger than that in the synthetic examples. This is mainly because of error in estimation of h_i.



(a) Input Blurred image (b) Result from the basic PMRL

(c) Result from PMRL with regularization (d) a clear image taken the same place

Figure 4-3 Example of Zooming blur



(a) Input Blurred Image (b) Result from the basic PMRL

(c) Result with regularization (d) a clear image

Figure 4-4 Example of Rotational blur

4.3. Run Time Analysis

Current implementation takes about 15 minutes to run 500 iterations on a 512 ×512 image with Intel(R)CPU L2400@1.66Ghz. The run time of Projective Motion RL algorithm depends on several factors, including the image size, number of discrete samplings (number of homographies) and number of iterations. Most of the run time in the current implementation is spent on the bicubic interpolation during the generation of predicted blurred image and the integration of the residual errors. One possible solution to speed up this process is to use GPU for the projective transformation at each *dt.*

4.4. Limitations

The key advantage of the Projective Motion Blur Model is that it is in accordance with to the physical phenomena causing the blur effect. Hence it offers a compact representation for the spatially varying blurs. The Projective Motion RL algorithm is derived directly from conventional RL algorithm with a similar formulation. It also incorporates with state-of-the-art regularizations.

Currently the Projective Motion RL algorithm has several limitations. The first limitation is inherent from Projective Motion Blur Model that the blur is assumed to be motion blur instead of out-of-focus blur. Furthermore, the scene is assumed to be static without motion nor deformable object, and lacks significant depth variation.

4.5. Further directions

There can be several major future directions of this work. The first direction is to overcome the current limitations. There are several other causes that may generate spatially varying blur, such as lens distortion, combining motion with out-of-focus blur and moving object. It is possible to extend the current Projective Motion Blur Model to other distortion. Current blur is a sequence of transformations during exposure, other than projective transformation $\Delta I(x, t_i) = \frac{1}{N} I_0(H_i x)$, transformations $f_i(x)$ that describe the other causes of the blur could also be used by replacing H_i. To deal with moving object and significant depth variation, image matting could be added to separate the object/depth layer. In order to keep each segmentation in accordance with the Projective Motion Blur Model, additional constraints could be imposed regarding the object's motion.

Other directions are to explore existing algorithms and hardware (e.g. coded exposure and coded aperture) to use this blur formulation instead of assuming a global PSF. It is possible to include more than one motion blurred images, e.g. noisy/blurred image pair or video deblurring. On video deblurring, the motion between adjacent frames can be used to estimate the blur motion, replacing user provided corresponding points. Then by developing a kernel refinement algorithm, the image and projective motion could be updated in an iterative manner. A challenging problem is to estimate blur motion under the Projective Motion Blur Model and to perform blind-deconvolution, as estimating a piecewise projective camera motion is very different from estimating global PSF in a conventional deblurring algorithm.

Chapter 5.

Blurred Image Alignment

5.1. Background

Multiple blurred images of the same scene provide complementary information for deblurring. Therefore, deblurring result can be improved by using multiple images. For example, two blurred images were used to estimate a clear image in (12), and blurred/noisy images pair was used in (10). To take advantage of multiple blurred images, these images should be aligned precisely. In this chapter, we propose a method to align two blurred images.

Image alignment is a necessary and useful pre-processing step in quite a number of multi-image and video processing applications, e.g. image stabilization, image enhancement, video summarization and many other graphic applications. It is also well studied because of its applications in image processing. However, the traditional methods are designed for clear images and not suitable for blurred images, such as blurry photo from hand-held camera in low-light condition.

Aligning blurred image is non-trivial. Feature-based methods have problems extracting features from blurred images as popular feature descriptors such as SIFT (34) is not invariant to blur. Pixel-based methods are also infeasible when the blur kernel is large. The previous approaches for aligning blurred images are very limited as either the blur had to be one-dimension (26) (11), or the kernel had to be centrally symmetric (27), e.g. Gaussian blur kernel. For kernels caused by camera motion, the work in (28) proposed a method to align blurred/non-blurred image pairs using kernel sparseness. (28) also claimed there was no blur invariant feature for an arbitrarily shaped kernel.

A simple and effective method is introduced in this chapter that uses the characteristics of the blurred image in the Fourier domain for alignment. The accuracy is comparable to (28) but the implementation and computation is much simpler. The key contribution is the ability to align two blurred images with arbitrarily shaped kernels.

5.2. Blurred image Alignment - Theory

Suppose two blurred images of the same scene with different blur kernels can be aligned by an unknown transform H, a 3x3 homography matrix. In spatial domain it can be represented mathematically as follows:

$$\begin{gathered} B_1 = \mathrm{I}(\boldsymbol{x}) \otimes psf_1 \\ B_2 = \mathrm{I}(H\boldsymbol{x}) \otimes psf_2 \end{gathered} \tag{5-1}$$

where B_x is the blurred image, $\mathrm{I}(x)$ is the clear scene and psf_x is the point spread function. $\otimes$ is the convolution operator.

The goal is to find H so that the B_2 can be aligned to B_1 using H^{-1}, which could be represented as

$$B_2' = B_2(H^{-1}x) = \mathrm{I}(H^{-1}H\cdot\boldsymbol{x}) \otimes psf_2(H^{-1}\boldsymbol{x}) = I(\boldsymbol{x}) \otimes psf_2' \qquad (5\text{-}2)$$

Now, B_2' and B_1 are both convolutions of the same clear image $I(\boldsymbol{x})$**.** So B_2' is considered as the result of aligning B_2 to the image B_1 . Here H is assumed to be an affine transformation and psf_2' is the transformed version of psf_2 .

Hence ***B₁*** and B_2' can be transformed into the Fourier domain,

$$\boldsymbol{B}_1 = I(\boldsymbol{x}) \odot \boldsymbol{psf}_1$$

$$\boldsymbol{B}_2' = \boldsymbol{I}(\boldsymbol{x}) \odot \boldsymbol{psf}_2' \qquad (5\text{-}3)$$

By dividing these two equations in Eqn.(5-3), after some manipulations, we obtain

$$\frac{\boldsymbol{B}_1}{\boldsymbol{B}_2'} = \frac{\boldsymbol{psf}_1}{\boldsymbol{psf}_2'} = \frac{\boldsymbol{B}_{1r}}{\boldsymbol{B}_{2r}'} = \frac{\boldsymbol{B}_{1g}}{\boldsymbol{B}_{2g}'} = \frac{\boldsymbol{B}_{1b}}{\boldsymbol{B}_{2b}'} \qquad (5\text{-}4)$$

where $\boldsymbol{B}_{1r}, \boldsymbol{B}_{1g}, \boldsymbol{B}_{1b}$ and $\boldsymbol{B}_{2r}, \boldsymbol{B}_{2g}, \boldsymbol{B}_{2b}$ are the three different color channels in $\boldsymbol{B}_1$ and $\boldsymbol{B}_2$ respectively. If the alignment is correct, the ratio between each channel of the blurred images should be $\frac{\boldsymbol{psf}_1}{\boldsymbol{psf}_2'}$ which only depends on blur kernels and is independent of the scene, as in the same image the blur kernel should be identical in each color channel. Therefore the alignment can be solved by minimizing the following equation:

$$H = arg \min_H \left(\sum_{i=r,g,b} (\bar{R} - R_i)^2 \right)$$

$$= \arg\min_H (Var[R_r(H), R_g(H), R_b(H)])$$

$$\text{where} \quad R_x(H) = \frac{\boldsymbol{F}(B_{1x})}{\boldsymbol{F}(B_{2x}(H^{-1}x))} \qquad (5\text{-}5)$$

$$\bar{R} = \frac{1}{3} \sum_{i=r,g,b} R_i$$

where $\boldsymbol{F}(\cdot)$ denotes the Fourier Transform.

5.3. Implementation and Acceleration

This section describes the search for H when it is a similarity transformation. Similar method can be applied to search for a full affine transformation matrix H. For similarity transform

$$H(s,\theta,tx,ty)=\begin{bmatrix} s\cdot\cos(\theta) & \sin(\theta) & tx \\ -\sin(\theta) & s\cdot\cos(\theta) & ty \\ 0 & 0 & 1 \end{bmatrix} \tag{5-6}$$

here s is the scale factor, θ is the rotation angle between the two images, and *tx* and *ty* are the translation. Meanwhile, translation (tx,ty) in the spatial domain does not affect the magnitude in the Fourier domain. Hence by first ignoring the translation, the minimization can be simplified to a 2D search for (s,θ). During the experiments, the result shows the energy function is locally continuous, an example of energy function vs. scaling and rotation is shown in Figure 5-1(*d)*. Therefore after a good initialization, searching on scaling and rotation can be further separated. In other words, from the second iteration onward, when searching for rotation angles, the scaling factor is fixed to the current best result. The same method is used to search for the scaling factor with the rotation angle fixed. In this way, the searching space is further reduced from 2D to 1D.

To accelerate the searching process, multi-scaled coarse-to-fine implementation was used. The idea was to have a rough alignment on a smaller image as initialization, and then work on a bigger image with higher accuracy. Each level scales by a factor of $\sqrt{2}$ and usually 3 to 4 different scales are used.

The multi-scaled iterative coarse-to-fine method is designed as follows:

1. Scale down both images to lower resolution.
2. Search minimization solution of Equation (5-5)in combination of rotation and scale (θ, s) with a sampling intervals ($\Delta\theta, \Delta s$);
3. Reduce sampling intervals to 1/3 ($\Delta\theta, \Delta s$) within the search ranges (θ + 2 $\Delta\theta$,s + 2 Δs) to increase accuracy;
4. Search on rotation and scaling separately, based on the previous solution;
5. Repeat step 3,4 once or twice to increase accuracy (optional), ;
6. Scale the images to higher resolution,;
7. Go to step 3 for next iteration until the result accuracy is satisfied.

Initial step length $\Delta\theta$ is 1-3 degree in rotation and Δs is 0.05-0.1 in scale depending on how different the images are. Usually 5 to 6 iterations lead to an accurate result.

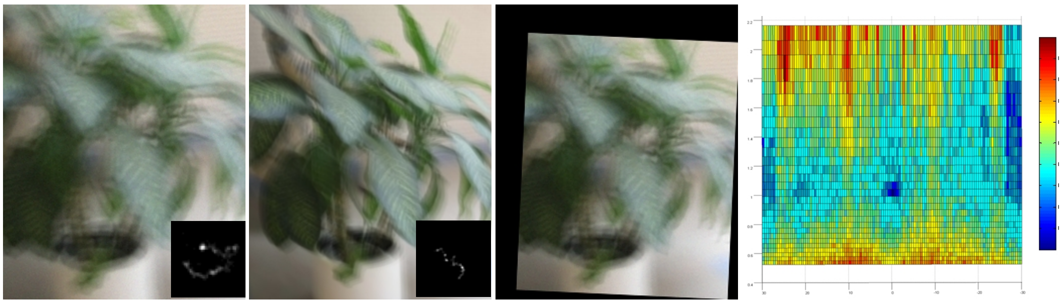
After the scale and rotation have been recovered, the translation *tx* and *ty* can be computed easily in spatial domain using conventional techniques such as normalized cross correlation (NCC). Due to the characteristics of digital image, the resolution of the NCC is only up to 1 pixel. However, it is usually enough for aligning blurred image, as the blur itself usually contains a translation motion and the error in translation can be compensated in blur kernel as well. If, in certain cases, a better resolution in translation alignment is required, image can be scaled to higher resolution by bi-cubic interpolation before NCC to increase accuracy, such as if the image is scaled by factor of 5, the accuracy can be increase to 0.2 pixels on the original resolution.

Furthermore, in order to obtain the desired solution, a few problems need to be handled properly.

1. **Image Out-of-Range.** During image transformation, it is very common to have empty region at the boundary of the image due to Out-of-Range effect. An example of the empty region (black region) is shown in Figure 5-1(c). These empty regions must be removed before alignment because they would significantly affect the result in the Fourier Domain.
2. **Edge Discontinuity**. Due to the discontinuity at the image boundary, very sharp responses in horizontal and vertical direction usually appear in Fourier Domain. To overcome this effect, the image boundary region has to be smooth and continuous, which could be done by matlab function *edgetaper*().
3. **Divide-by-Zeros.** In fact, a certain threshold must be set, not only to avoid divide-by-zeros but also to compensate the noise and the huge information lost in high frequency. Usually this threshold will filter out most of the high frequency regions, as for blurred image, the high frequency region is very weak in the Fourier domain.

5.4. Experiment and Result

The algorithm was first tested on synthetic data, which were generated by 20 different images and 10 different blur kernels. Each image was transformed by a similarity transformation, with random rotation between (-45° , 45°) and scaling between (0.7 , 1.8). Then 2 kernels were chosen randomly to generate blurred images. The experiment was also repeated for different noise levels.



(a) Blurred Image 1 (b) Blurred Image 2 (c) Registered Image 1 (d) energy function on rotation/scaling

Figure 5-1. Example of Alignment on Synthetic Data

An example of synthetic testing data set is in Figure 5-1. In the map of energy function (Figure 5-1.d), it shows the global minimum is at the true alignment. And the cost function is locally continuous so that the search on rotation and scaling can be further separated after the first initial.

Figure 5-2 shows a typical searching procedure on aligning 2 images by 5 iterations. Left column is the energy function vs. estimated scaling, right column is the rotation and each row is one iteration. In the first iteration, it performs a searching on a 2D space to find the initialization. From the second iteration onward, it performs searching on the scaling first followed by rotation to reduce the searching space from 2D to 2 × 1D.

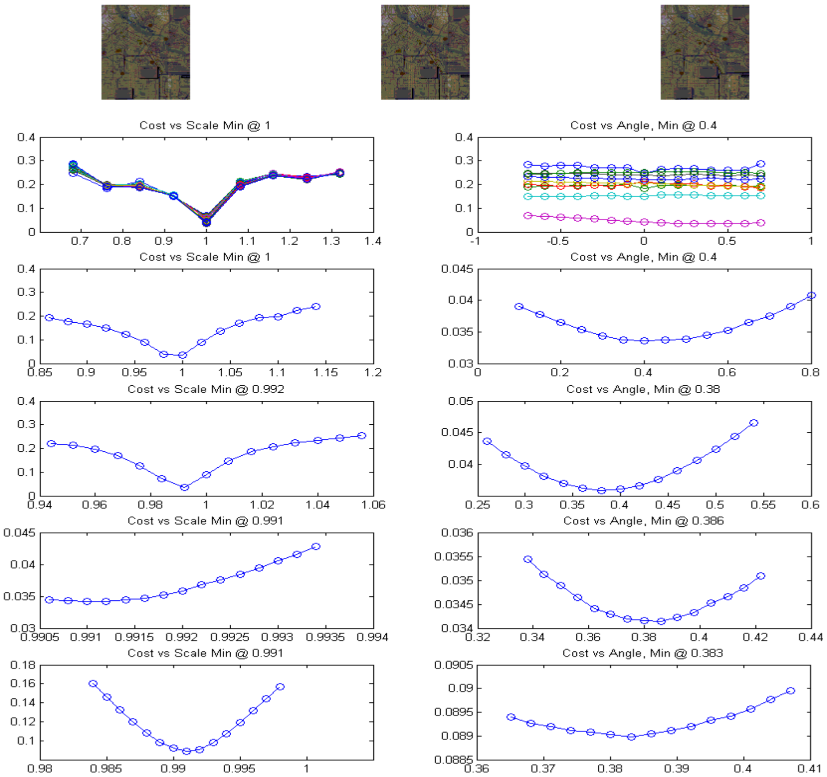


Figure 5-2. A Typical Search Routine for alignment

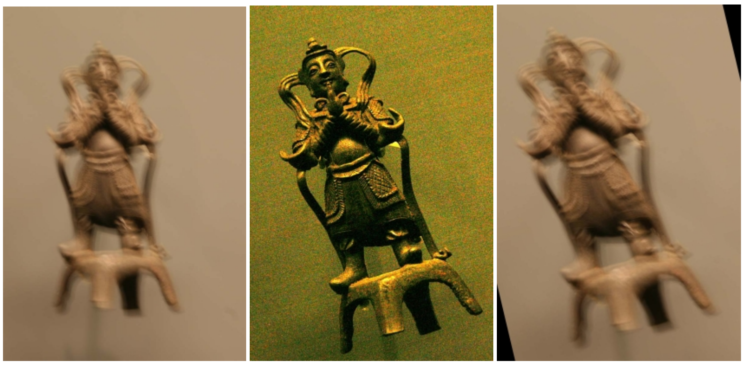
The accuracy of the estimation can be evaluated easily from synthetic examples, as the ground truth is known. For the example shown in Figure 5-1, the estimated parameters are (2.327˚, 1.1469) with respected ground truth (2.3377˚, 1.1459). The error is (0.01˚, 0.001). In this example, the blur kernel was 32x32 with the image size 320x240. The alignment error was sub-pixel.

Table 5-1 shows the alignment accuracy for 6 tests on first 9 testing images with different noise levels. The average of absolute error is also shown at the bottom. The testing images are shown in Appendix A. The scaling factors and the rotation angles for each test were generated randomly.

Noise	$\sigma^2 = 2$		$\sigma^2 = 5$		$\sigma^2 = 10$	
Image	T1 (x10^{-3})	T2(x10^{-3})	T3(x10^{-3})	T4(x10^{-3})	T5(x10^{-3})	T6(x10^{-3})
1	(-6.3°,-0.6)	(-6.8°,1.3)	(-14.9°,-0.1)	(-8.1°,-1.3)	(-25.3°,-0.3)	(-25.3°,-0.2)
2	(-9.0°,-1.6)	(-10.4°,1.3)	(-25.3°,-0.3)	(-25.3°,-0.2)	(-15.1°,-1.4)	(-22.1°,-1.3)
3	(5.7°,1.1)	(1.2°,-1.0)	(-28.5°,0.2)	(6.1°,-2.2)	(-24.1°,0.4)	(18.1°,1.0)
4	(-12.9°,0.1)	(-0.6°,0.8)	(-13.2°,-1.6)	(-3.8°,2.5)	(11.0°,1.4)	(-8.5°,1.1)
5	(5.4°,-0.1)	(-1.4°,0.8)	(25.0°,1.0)	(-10.7°,2.3)	(-10.9°,-1.6)	(-19.8°,1.2)
6	(6.8°,0.6)	(5.8°,-0.9)	(25.0°,-2.1)	(22.4°,-0.6)	(-34.8°,0.7)	(11.8°,-1.4)
7	(-7.2°,0.2)	(8.0°,-1.4)	(-7.8°,-0.4)	(-1.0°,-2.1)	(-24.9°,0.9)	(-27.1°,-0.9)
8	(4.6°,-0.7)	(12.8°,-1.2)	(2.7°,-0.8)	(-26.5°,-1.8)	(19.2°,-2.0)	(17.0°,-0.8)
9	(-9.6°,-1.3)	(-13.7°,0.0)	(10.1°,-1.4)	(15.5°,-1.1)	(18.0°,2.0)	(-15.5°,-0.9)
...	...	...	...	...	...	...
Average	(7.4°,0.9)	(9.8°,1.1)	(15.4°,1.2)	(14.4°,1.6)	(20.3°,1.6)	(18.3°,1.4)

Table 5-1. Error on synthetic images with different noise level.

Although the formulation was derived from two blurred images, our method can also be applied to align a blurred image to a noisy image. We tested our alignment algorithm on the Blur/Noisy image pair from (28), as shown in Figure 5-3.



(a) Blurred Image 1 (b) Noisy Image (c) Aligned result

Figure 5-3. Real Image pairs from (28)

The estimated parameters are (15.566˚, 0.8514) while the suggested result given by (28) is (15.59˚, 0.852). The difference is (0.024˚, 0.0006), though the algorithm is completely different. As the result in (28) only kept 2 digits in degree and 3 digits in scaling, the actual difference may be even smaller. In addition, this algorithm can also tolerate the color shift due to white balance or lightening condition changes by normalizing each channel to same mean and variance.

The run time is about 3 minutes by Matlab for image with size of 1024×768 on an Intel P4 3.2G desktop. Meanwhile the running time for (28) was about 40 minutes.

Chapter 6.

Conclusion

This thesis addresses two problems, modeling and correcting spatially varying image blur and alignment on blurred images.

Projective Motion Blur Model was designed to modeling spatially varying image blur caused by rigid camera motion with static scene. It treats the blur as an integration of a clear scene over exposure time with a sequence of projective transformations which describe the camera motion. This formulation is intuitive to the physical phenomena causing the blur effect and also offers a compact representation for the spatially varying motion blur. Subsequently, we propose the Projective Motion Richardson-Lucy (RL) algorithm to recover a clear image from an image subject to spatially variant blur according to our Projective Motion Blur Model. We also incorporated state-of-the-art regularization image priors to improve deblurring results. The effectiveness of Projective Motion RL algorithm was demonstrated during experiment. There are also a few future research directions based on the Projective Motion Blur Model, e.g. projective motion estimation or refinement, deblurring with multiple blurred images, video deblurring and extension to other causes of spatially varying blur (e.g. lens distortion). A more challenging problem is to perform blind-deconvolution.

A blurred image alignment algorithm was also presented in this thesis, as a preprocessing step for the future research in multiple blurred images deblurring. This algorithm solved the problem of alignment on blurred images, which is especially useful in multiple images deblurring algorithm, such as previous work “Robust Dual Motion Deblurring” (12). Our method achieved a comparable accuracy with the alignment algorithm proposed in (28). While the algorithm in (28) only works on clear/blurred image pair, our algorithm could align two blurred images. In addition, comparing with algorithm in (28) which solved the alignment problem by estimating the kernel and measuring its sparseness, our method is more direct and efficient to align blurred image by measuring kernels ratio in the Fourier Domain. The run time was also more than 10 times faster. Therefore, the our alignment method is fast, accurate and capable on dealing with two blurred images.

References

1. **A. Levin, Y. Weiss, F. Durand and W. Freeman.** Understanding and evaluating blind deconvolution algorithms. *CVPR.* 2009.

2. **W. Richardson.** Bayesian-based iterative method of image restoration. *J. Opt. Soc. Am.* 1972.

3. **L. Lucy.** An iterative technique for the rectification of observed distributions. *Astron. J.* 1974.

4. **Wiener and Norbert.** Extrapolation, interpolation and smoothing of stationary time series. *New York: Wiley*. 1949.

5. **R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis and W. T. Freeman.** Removing camera Shake from a single photograph. *ACM SIGGRAPH.* 2006.

6. **N. Dey, L. Blanc-Fraud, C. Zimmer, Z. Kam, P. Roux, J. Olivo-Marin and J. Zerubia.** A deconvolution method for confocal microscopy with total variation regularization. *In IEEE International Symposium on Biomedical Image: Nano to Macro.* 2004.

7. **A. Levin.** Blind motion deblurring using image statistics. 2006.

8. **L. Yuan, J. Sun, L. Quan and H. Shum.** Progressive interscale and intra-scale non-blind image deconvolution. *ACM Transactions on Graphics*. 2008.

9. **Q. Shan, J. Jia and A. Agarwala.** High-quality motion deblurring from a single image. *ACM Transactions on Graphics.* 2008.

10. **L. Yuan, J. Sun, L. Quan and H. Shum.** Image Deblurring with Blurred/Noisy Image Pairs. *SIGGRAPH.* 2007.

11. **A. Rav-Acha and S. Peleg.** Two motion blurred images are better than one. *PRL, 26:311-317.* 2005 : s.n.

12. **J. Chen, L. Yuan, C. Tang and L. Quan.** Robust Dual Motion Deblurring. *IEEE Conference on Computer Vision and Pattern Recongnition(CVPR).* 2008, pp. 1-8.

13. **R. Raskar, A. Agrawal and J. Tumblin.** Coded exposure photography: motion deblurring using fluttered shutter. *ACM Trans. Graph., 25(3).* 2006.

14. **A. Agrawal and R. Raskar.** Resolving objects at higher resolution from a single motion-blurred image. *CVPR.* 2007.

15. **M. Ben-Ezra and S. Nayar.** Motion Deblurring using Hybrid Imaging. . *IEEE Trans. Pattern Analysis and Machine Intelligence.* 2004.

16. **M. Ben-Ezral and S. Nayar.** Motion-based Motion Deblurring. *IEEE Trans. Pattern Analysis and Machine Intelligence.* 2004.

17. **A. A. Sawchuk.** Space-variant image restoration by coordinate transformations. *Journal of the Optical Society of Ameraca, 64(2):138-144.* 1974.

18. **J. Bardsley, S. Jefferies, J. Nagy and R. Plemmons.** Blind iterative restoration of images with spatially-varying blur. *Optics Express.* 2006.

19. **S. Cho, Y. Matsushita and S. Lee.** Removing non-uniform motion blur from images. *Proc. IEEE Int. Conf. Computer Vision.* 2007.

20. **F. Li, J. Yu and J. Chai.** A hybrid camera for motion deblurring and distributions. *CVPR.* 2008.

21. **Q. Shan, W. Xiong and J. Jia.** Rotational Motion deblurring of a rigid object from a single image. *Proc. IEEE Int. Conf. Computer Vision*. 2007.

22. **N. Joshil, R. Szeliski and D. Kriegman.** Psf estimation using sharp edge prediction. *Proc. IEEE Conf. Computer Vision and Pattern Recongnition*. 2008.

23. **Y. Tai, H. Du, S. Lin and M. Brown.** Image/Video deblurring using a hybrid camera. *Proc. IEEE Conf. Computer Vision and Pattern Recongnition*. 2008.

24. **Y. Tai, H. Du, M. Brown and S. Lin.** Correction of spatially varying image and video blur using a hybrid camera. *IEEE Trans. PAMI.* 2009.

25. **R. Szeliski.** Image alignment and stitching : A tutorial. *Foundations and Trends in Computer Graphics and Computer Vision, 2(1) : 1-104.* 2006.

26. **A. Rav-Acha and S. Peleg.** Restoration of multiple images with motion blur in different directions. *IEEE workshop on Applications of Computer Vision.* 2000.

27. **J. Flusser and T. Suk.** Degraded image analysis: an invariant approach. *PAMI, 29(6):590-603.* 1998.

28. **L. Yuan, J. Sun, L. Quan and H. Shum.** Blurred/Non-Blurred Image Alignment using Spareseness Prior. *Proc. IEEE Int. Conf. Computer Vision*. 2007.

29. **C. Kuglin and D. Hines.** The Phase Correlation Image alignment Method. *IEEE Int. Conf. on Cybernetics and Society*. 1975.

30. **B. Reddy and B. Chatterji.** An FFT-based technique for translation, rotation, and scale-invariant image registration. *IEEE Transactions on Image Processing.* 1996.

31. **M. Irani and S. Peleg.** Improving resolution by image registration. *CVGIP: Graph. Models Image Process., vol. 53, pp. 231--239,.* 1991.

32. **A. Levin, R. Fergus, F. Durand and W. T. Freeman.** Image and depth from a conventional camera with a coded aperture. *ACM Transactions on Graphics.* 2007.

33. **S. Dai and Y. Wu.** Motion from blur. *Proc. IEEE Conf. Computer Vision and Pattern Recognition.* 2008.

34. **D. G. Lowe.** Object recognition from local scale-invariant features. *International Conference on Computer Vision (ICCV).* 1999.

35. **J. Flusser and T. Suk.** Degraded image analysis: An invariant approach. *PAMI.* 1998.

36. **R. Hartley and A. Zisserman** Multiple View Geometry in Computer Vision. *Cambridge University Press, ISBN 0521623049.* 2000.

37. **E. De. Castro and C. Morandi.** Registration of Translated and Rotated Images Using Finite Fourier Transforms. *IEEE Transactions on pattern analysis and machine intelligence*. 1987.

38. **M. A. Fischler and R. C. Bolles,** Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM - CACM.* 1981.

39. **O. Whyte, J. Sivic, A. Zisserman and J. Ponce.** Non-uniform Deblurring for Shaken Image. *CVPR.* 2010.

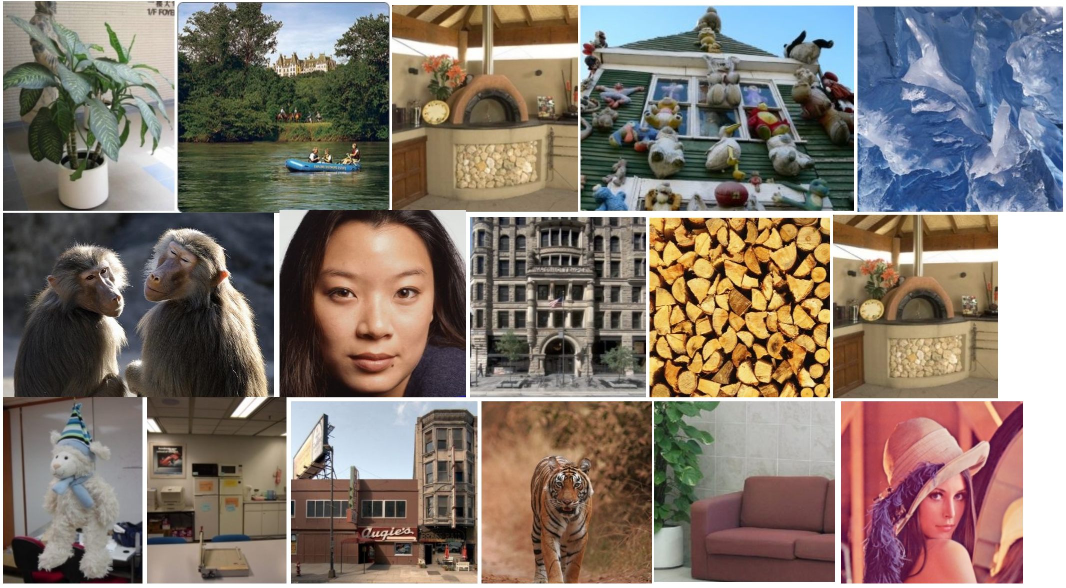
Publications

1. **Y. W. Tai, P. Tan, L. Gao and M. S. Brown,** Richardson-Lucy Deblurring for scenes under Projective Motion Path, *IEEE Transactions on Pattern Analysis and Machine Intelligence, major revision.*

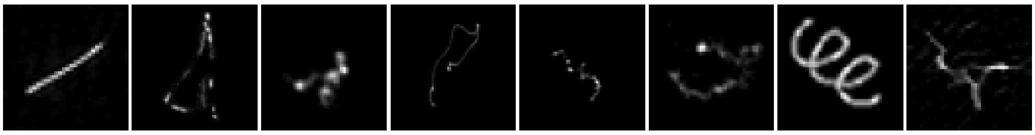
2. **Y. W. Tai, P. Tan, L. Gao and M. S. Brown,** Richardson-Lucy Deblurring for scenes under Projective Motion Path, *Technical report, KAIST, 2009.*

Appendix A: Experiments on Blurred Image Alignment

Testing Image:



Blur Kernel:



Experiment on accuracy analysis:

1) Transform the original image with a random rotation degree and scale factor. (r,s)
2) Randomly choose two blur kernels to generate two blurred images from both the original image and the transformed image. Noise is then added to blurred image.
3) Measure the error between the alignment result and the ground truth.

Example of Synthetic Blurred images and alignment result.

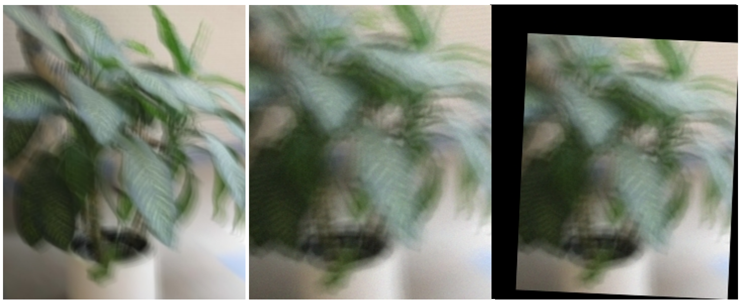
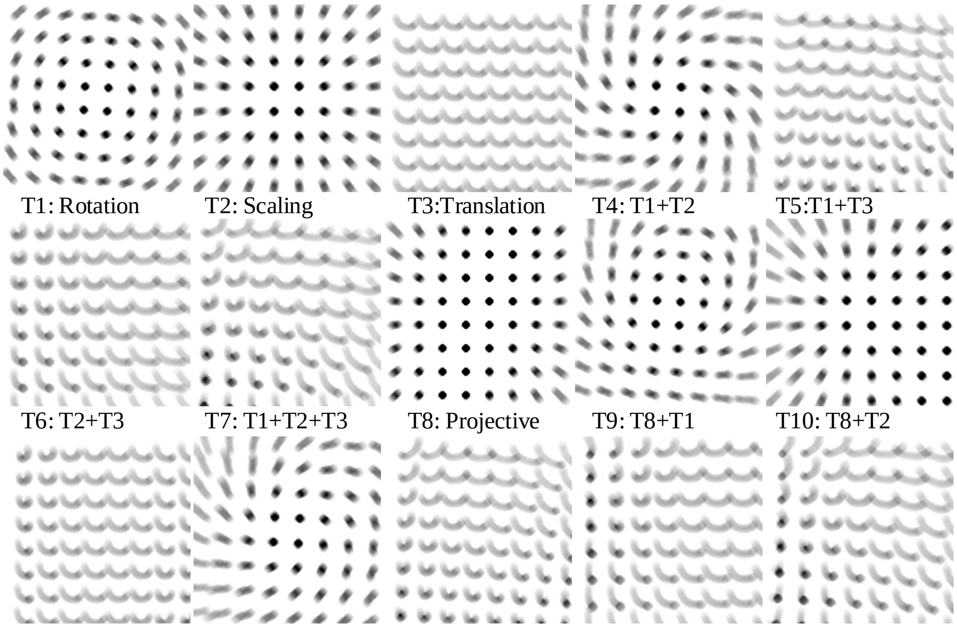


	Image1	Image2
Kernel	No.6	No.5
Scaling	0	1.1459
Rotation	0	2.3377°
Result		1.1469,2.327°
Error		0.001, 0.011°

Appendix B: Test Results - Projective Motion RL

Test Image:

Test Motion :

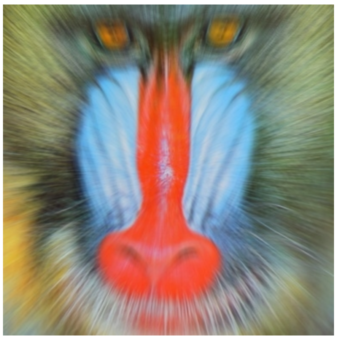


T1: Rotation T2: Scaling T3:Translation T4: T1+T2 T5:T1+T3

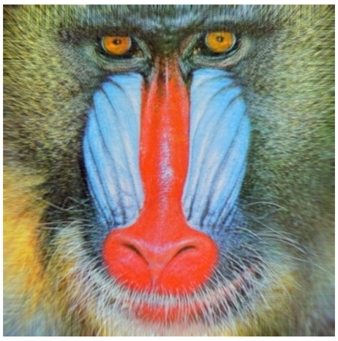
T6: T2+T3 T7: T1+T2+T3 T8: Projective T9: T8+T1 T10: T8+T2

T11: T8+T3 T12: T8+T4 T13: T8+T5 T14: T8+T6 T15: T8+T7

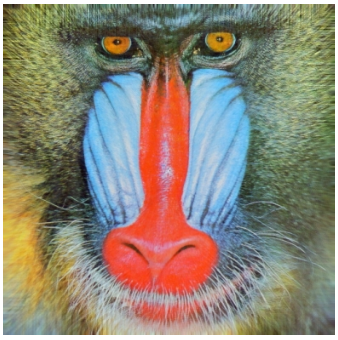
Deblur Result:



Synthetic Blurred image (T2)

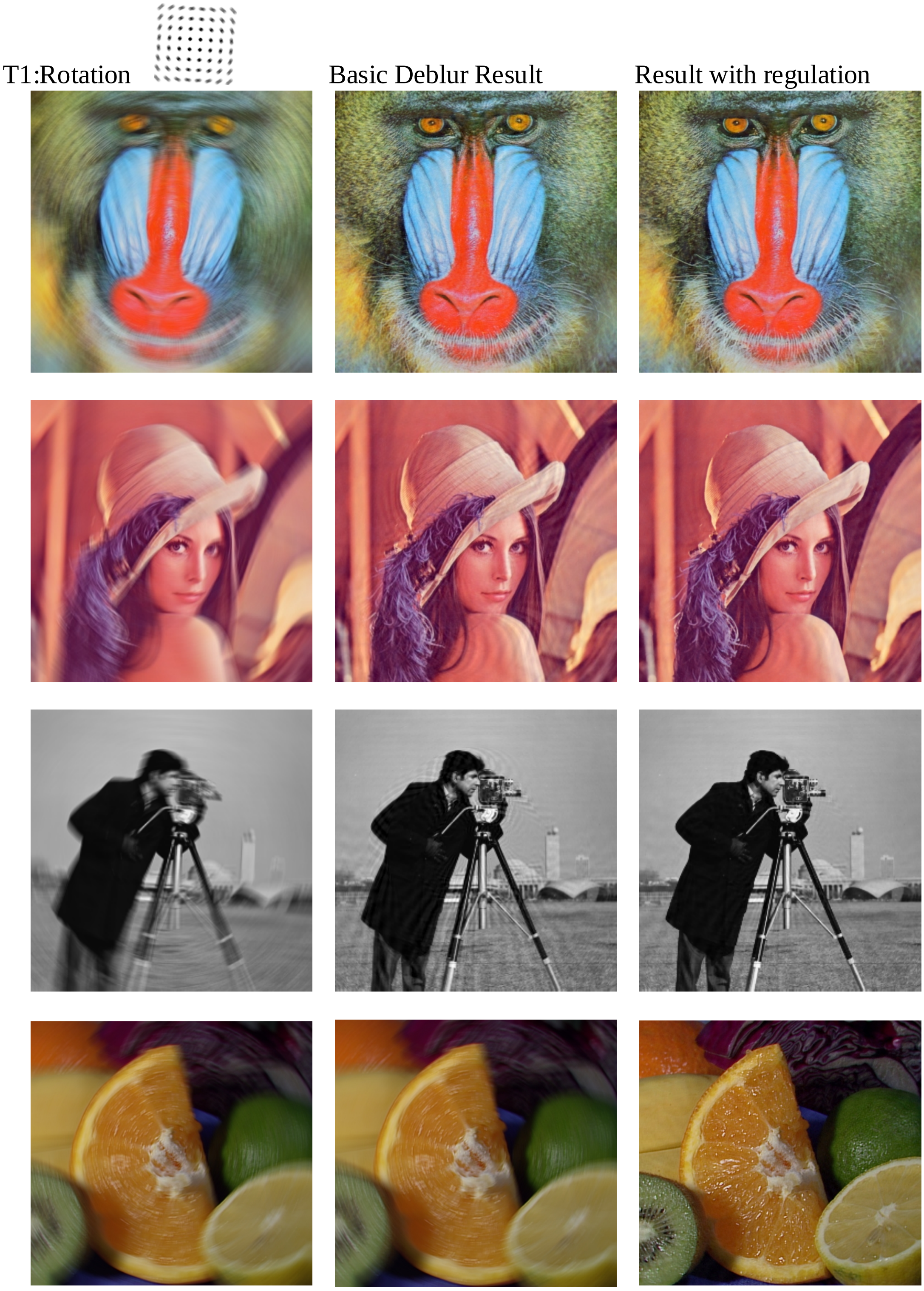


Deblurred result By basic Projective Motion RL algorithm

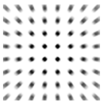


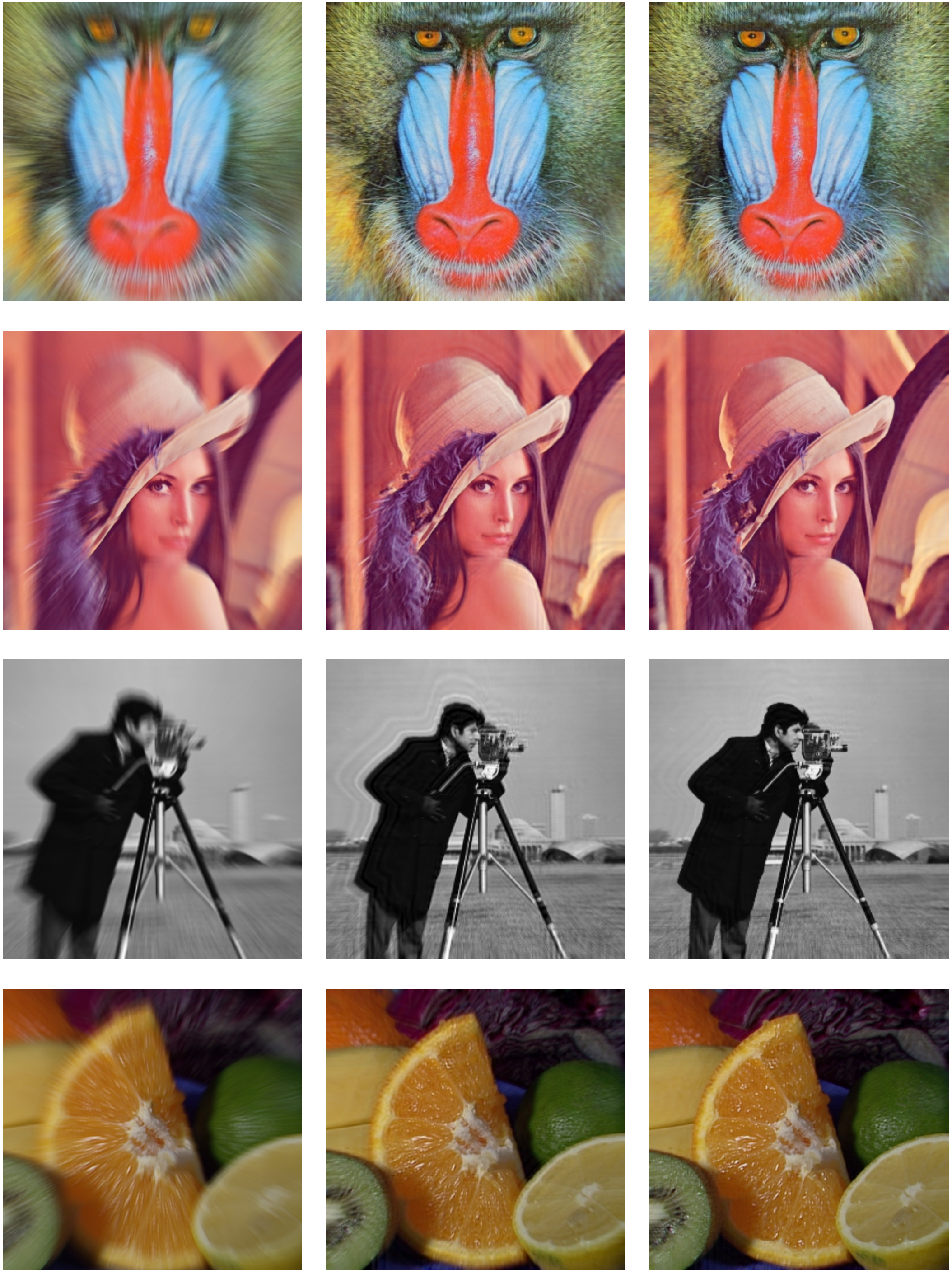
Deblurred result with regulation

More results for Motion T1,T2,T3,T7,T15 are shown on following pages.

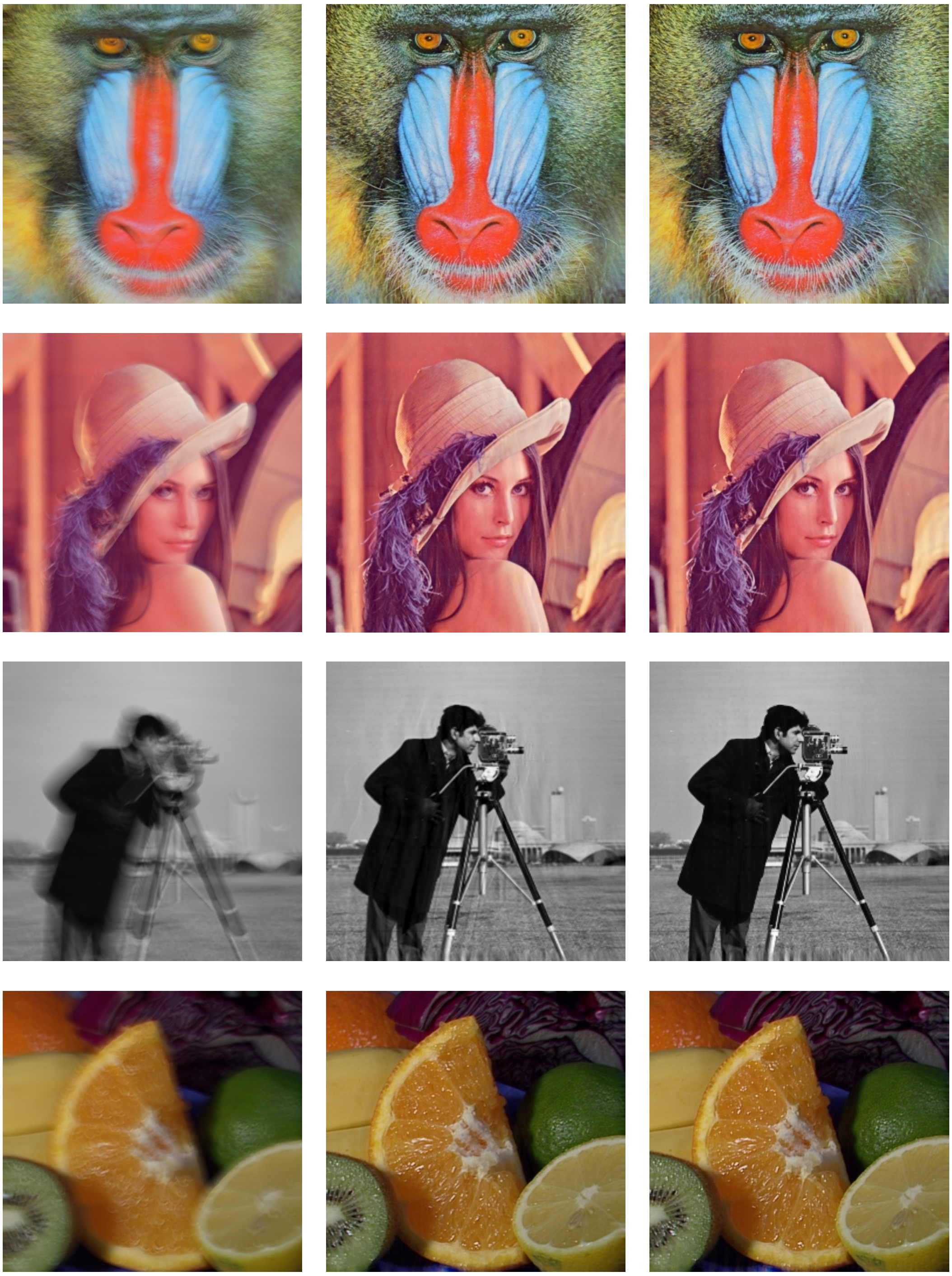

T1:Rotation
Basic Deblur Result
Result with regulation

T2: Zooming

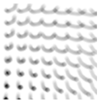


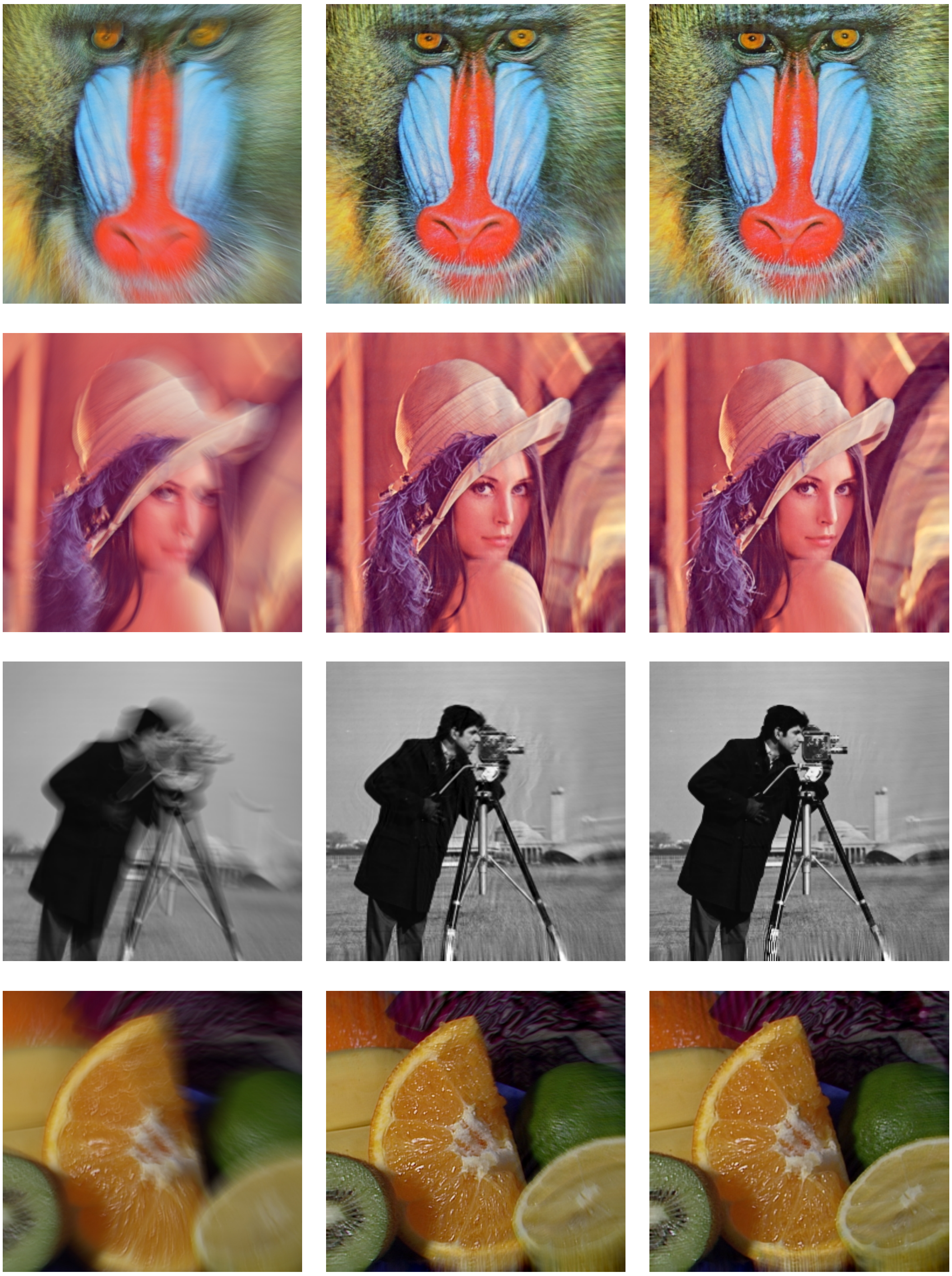


T3: Translation



T7: T1+T2+T3





T15: T7+T8