Richardson-Lucy Deblurring for Scenes Under A Projective Motion Path

Yu-Wing Tai   Ping Tan   Michael S. Brown

***Abstract*— This paper addresses how to model and correct image blur that arises when a camera undergoes ego motion while observing a distant scene. In particular, we discuss how the blurred image can be modeled as an integration of the clear scene under a sequence of planar projective transformations (i.e. homographies) that describe the camera's path. This projective motion path blur model is more effective at modeling the spatially varying motion blur exhibited by ego motion than conventional methods based on space-invariant blur kernels. To correct the blurred image, we describe how to modify the Richardson-Lucy (RL) algorithm to incorporate this new blur model. In addition, we show that our projective motion RL algorithm can incorporate state-of-the-art regularization priors to improve the deblurred results. The projective motion path blur model along with the modified RL algorithm is detailed together with experimental results demonstrating its overall effectiveness. Statistical analysis on the algorithm's convergence properties and robustness to noise is also provided.**

I. Introduction

Motion blur from camera ego motion is an artifact in photography caused by the relative motion between the camera and an imaged scene during exposure. Assuming a static and distant scene, and ignoring the effects of defocus and lens aberration, each point in the blurred image can be modeled as the convolution of the unblurred image by a point spread function (PSF) that describes the relative motion trajectory at that point's position. The aim of image deblurring is to reverse this convolution process to recover the clear image of the scene from the captured blurry image as shown in Figure 1.

A common assumption in existing motion deblurring algorithms is that the motion PSF is spatially invariant. This implies that all pixels are convolved with the same motion blur kernel. However, as recently discussed by Levin *et al.* [17] this global PSF assumption is typically invalid. In their experiments, images taken with camera shake exhibited notable amounts of rotation that causes spatially varying motion blur within the image. Figure 2 shows a photograph that illustrates this effect. As a result, Levin *et al.* [17] advocated the need for a better motion blur model as well as image priors to impose on the deblurred results. In this paper, we address the former issue by introducing a new and compact motion blur model that is able to describe spatially varying motion blur caused by a camera undergoing ego motion.

We refer to our blur model as the *projective motion blur model* as it represents the degraded image as an integration of the clear scene under a sequence of planar projective transforms. Figure 3 shows a diagram of this representation. One key benefit of this model is that it is better suited to represent camera ego motion than the conventional kernel-based PSF parameterization that would require the image to be segmented into uniform blur regions, or worse, a separate blur kernel per pixel. However, because our approach is not based on convolution with an explicit PSF, it has no apparent frequency domain equivalent. One of the key contributions of this paper is to show how our blur model can be used to extend the conventional pixel-domain *Richardson-Lucy* (RL) deblurring algorithm. We refer to this modified RL algorithm as the *projective motion Richardson-Lucy* algorithm. Similar to the conventional RL deblurring, regularization based on various priors can be incorporated in our algorithm.

Our paper is focused on developing the projective motion blur model and the associated RL algorithm. We assume that the motion path of the camera is known and that the camera's motion satisfies our projective motion blur model. Recent methods useful in estimating the projective motion path are discussed in Section VI. As with other camera shake deblurring approaches, we assume that the scene is distant and and void of moving objects.

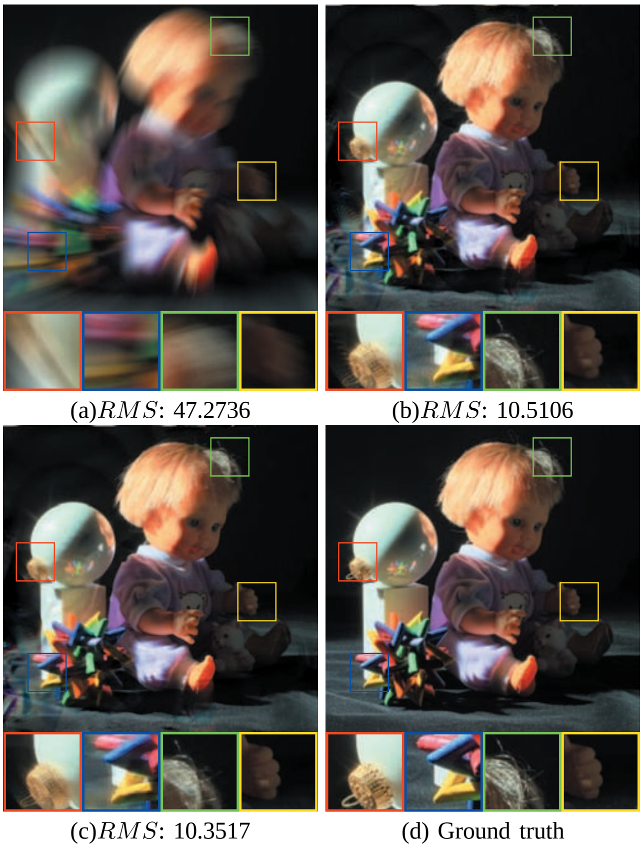


Fig. 1. (a) An image degraded by spatially varying motion blur due to camera ego motion. (b) The result from our basic algorithm. (c) Our result with added regularization. (d) Ground truth image. The RMS errors are also shown below each image.

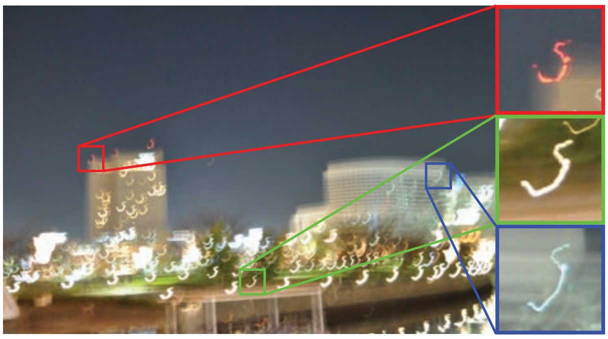


Fig. 2. This example demonstrates the spatially varying nature of camera shake (a similar example was shown in [31]). The motion paths of the saturated point light sources (shown zoomed-in) represent the PSF at various locations in the image. It is clear that the PSFs are not uniform in appearance.

The remainder of this paper is organized as follows: Section II discusses related work; Section III details our motion blur model; Section IV derives the projective motion Richardson-Lucy deconvolution algorithm; Section V describes how to incorporate regularization into our modified RL deconvolution algorithm with implementation details; Section VI discusses potential methods that can be used to estimate the projective motion path; Section VII provides analysis of the convergence properties of our algorithm, its sensitivity to noise, and comparisons against other approaches. A discussion and summary of this work is presented in Section VIII.

II. Related Work

Existing work targeting image blur due to camera ego motion has assumed a global PSF for the entire image. When the blur PSF is known, or can be estimated, well-known debluring algorithms such as Richardson-Lucy [22], [20] and Wiener filter [32] can be applied to deblur the image. Due to poor kernel estimation, or convolution with PSFs that result in unrecoverable frequencies, these conventional deblurring algorithms can introduce undesirable artifacts in the deblurred result such as “ringing” and amplification of image noise.

Consequently, research addressing image deblurring, including camera shake and other types of blur, typically target either blur kernel estimation or ways to regularize the final result, or both. For example, Dey *et al.* [10] and Chan and Wong [6] utilized total variation regularization to help ameliorate ringing and noise artifacts. Fergus *et al.* [11] demonstrated how to use a variational Bayesian approach combined with gradient-domain statistics to estimate a more accurate PSF. Raskar *et al.* [21], [1] coded the exposure to make the PSF more suitable for deconvolution. Jia [13] demonstrated how to use an object’s alpha matte to better compute the PSF. Levin *et al.* [16] introduced a gradient sparsity prior to regularize results for images exhibiting defocus blur. This prior is also applicable to motion blurred images. Yuan *et al.* [33] proposed a multiscale approach to progressively recover blurred details while Shan *et al.* [24] introduced regularization based on high order partial derivatives to reduce image artifacts.

These previous approaches all work under the uniform PSF assumption. As mentioned in Section I, camera ego motion causes a spatially varying motion blur that cannot be accurately modeled with a uniform PSF. Prior work has recognized the need to handle non-uniform motion blur for camera ego motion, moving objects, and defocus blur. For example, early work by Sawchuk [23] addressed motion blur from a moving camera by first using a log-polar transform to transform the image such that the blur could be expressed as a spatially invariant PSF. The range of motion that could be addressed was limited to rotation and translation. When addressing moving objects, the input image can be segmented into multiple regions each with a constant PSF as demonstrated by Levin [15], Bardsley *et al.* [2], Cho *et al.* [7] and Li *et al.* [18]. Such segmented regions, however, should be small to make the constant PSF assumption valid for the spatially varying motion blur in camera shake motion. For example, Tai *et al.* [27], [28] extended the hybrid camera framework used by Ben-Ezra and Nayar [4] to estimate a PSF per pixel using an auxiliary video camera. This need for a per-pixel PSF revealed the futility of relying on the conventional kernel based PSF model for spatially varying blur due to ego motion.

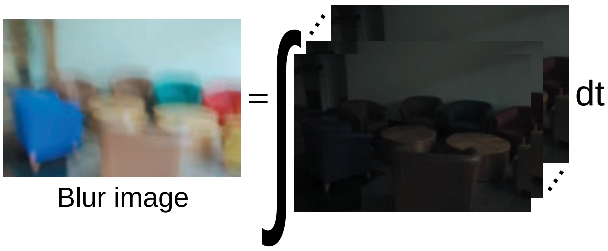


Fig. 3. Our input image is considered the integration of an image scene under projective motion.

The impetus of this work is to introduce a better blur model for camera ego motion. While this paper was under review, the utility of this projective motion path model has already been demonstrated by Tai *et al.* [29] for deblurring moving objects and Li *et al.* [19] for generating sharp panoramas from motion blurred image sequences. In addition, Whyte *et al.* [31] simultaneously proposed a similar blur formulation and correction algorithm. While their work focuses on rotational motion about the camera’s optical center, the formulation of their motion blur model is identical to ours.

III. The Projective Motion Blur Model

In this section, we describe our blur model based on a planar projective motion path. This model will be used in the following section to derive the deblurring algorithm.

In photography, the pixel intensity of an image is determined by the amount of light received by the imaging sensor over the exposure time:

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

where $I(x)$ is the image recorded after exposure; $\delta I(x,t)$ and its discrete equivalent $\Delta I(x,t_i)$ are the image captured by the sensor within an infinitesimal time interval dt at time instance t; $[0,T]$ is the total exposure time and x is a 3×1 vector indicating the homogenous pixel coordinate. In our model, we assume N (the discrete sampling rate over exposure time) is large enough so that the difference between continuous integration and discrete integration is negligible.

When there is no relative motion between the camera and the scene, assuming the effects of sensor noise is small, $\Delta I(x,t_1) \cong \Delta I(x,t_2) \cong \cdots \cong \Delta I(x,t_N)$ and $I(x) \cong N\Delta I(x,t_0) \triangleq I_0(x)$ is a clear image. When there is relative motion, $I(x)$ is

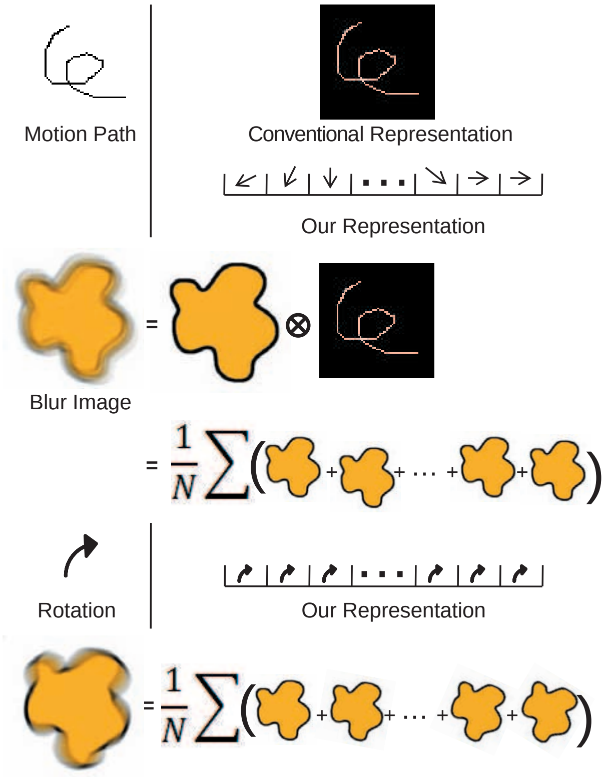


Fig. 4. This figure compares our blur model and the conventional model. Given the motion path (PSF), a conventional model uses an rectangular kernel (analogous to an image patch) to represent the PSF. In comparison, our model uses a sequence of transformation matrices. For rotational motion, our representation encodes the rotation via a sequence of homographies naturally, while the conventional approach would need to store pixelwise PSFs.

the summation of multiple *unaligned* images $\Delta I(x,t_i)$. For a static distant scene, the relative motion causes a planar projective transform in the image plane, i.e. $\Delta I(x,t_i) = \Delta I(h_i x, t_{i-1})$. Here, h_i is a homography[1] defined by a 3×3 non-singular matrix up to a scalar. Suppose all h_i are known, we can then express $\Delta I(x,t_i)$ by $I_0(x)$ using the following formulation:

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

where $H_i = \prod_{j=1}^{i} h_j$ is also a homography. Hence, we obtain our projective motion blur model as:

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

where $B(y)$ is the motion blurred image, and $I_0(x)$ is the clear image we want to estimate. According to our model, the blur image is the average of multiple clear images, each of which is a planar projective transformed of the clear image $I_0(x)$.

Figure 4 illustrates the relationship between our blur model and the conventional representation. The conventional spatially invariant PSF representation is a special case of our model for which every h_i is a translation. Note that for in-plane translational motion, the conventional kernel-based model provides a more compact representation of the motion blur than our model. However, in the cases of other motions, e.g. in-plane/out-of-plane rotation, our projective motion model is more compact and intuitive.

IV. Projective Motion Richardson-Lucy

In this section, we describe how to modify the Richardson-Lucy algorithm to incorporate our blur model. To do so, we first give a brief review of the Richardson-Lucy algorithm [22], [20] and then derive our algorithm in a similar manner. For simplicity, the term I instead of I_0 is used to represent the clear image to be estimated.

A. Richardson-Lucy Deconvolution Algorithm

The derivation in this section of the Richardson-Lucy Deconvolution algorithm [22], [20] is based on the paper from Shepp and Vardi [26]. In particular, the derivation from Shepp and Vardi shows that the RL algorithm can be considered a maximum likelihood solution using the Poisson distribution to model the likelihood probability $P(B,k|I)$:

$$\begin{aligned} P(B,k|I) &= \prod_{x \in I} \frac{g(x)^{B(x)} e^{-g(x)}}{B(x)!}, & (4) \\ g(x) &= \sum_{y \in k} I(y) k(x-y), & (5) \end{aligned}$$

where B is the observed motion blurred image, k is the motion PSF, i.e. $\sum_{y \in k} k(y) = 1$, I is the clear image we want to estimate, and $g(x)$ is a convolution process for a pixel located at x. Equation (4) assumes that the likelihood probability is conditionally independent for each x. Since $\sum_{y \in k} k(y) = 1$ and $\sum_{x \in B} B(x) = \sum_{x \in I} I(x)$, the overall intensity is preserved.

In [26], Shepp and Vardi show that Equation (4) is a concave function by showing the matrix of second derivatives of Equation (4) is negative semi-definite. In order to optimize Equation (4), it follows from the Theorem 2.19(e) of [30] that the sufficient conditions for I to be a maximizer of Equation (4) are the Kuhn-Tucker conditions where all x satisfy:

$$I(x) \frac{\partial}{\partial I(x)} \ln(\prod_{x \in I} \frac{g(x)^{B(x)} e^{-g(x)}}{B(x)!}) = 0, \tag{6}$$

and

$$\frac{\partial}{\partial I(x)} \ln(\prod_{x \in I} \frac{g(x)^{B(x)} e^{-g(x)}}{B(x)!}) \leq 0, \quad \text{if } I(x) = 0. \tag{7}$$

To obtain the iterative update rule for the RL algorithm, we use the first condition in Equation (6)[2], for all $x \in I$:

$$\begin{aligned} I(x) \frac{\partial}{\partial I(x)} \ln(\prod_{x \in I} \frac{g(x)^{B(x)} e^{-g(x)}}{B(x)!}) &= 0, \\ I(x) \sum_{x \in I} \frac{\partial}{\partial I(x)} (B(x) ln(g(x)) - g(x) - ln(B(x)!)) &= 0, \\ I(x) \sum_{x \in I} \frac{B(x)}{g(x)} \frac{\partial}{\partial I(x)} g(x) - I(x) \sum_{x \in I} \frac{\partial}{\partial I(x)} g(x) &= 0, \\ I(x) \sum_{y \in k} \frac{B(y)}{g(y)} k(y-x) - I(x) \sum_{y \in k} k(y-x) &= 0. \end{aligned}$$

[1] We use a homography for its ability to model all planar transformations. More restrictive transformations, e.g. rotation, translation, etc, can be used instead when prior knowledge of the camera's motion path is known.

[2] The second condition in Equation (7) is used to relate the RL algorithm with the EM algorithm [9] for the convergence proof. For further details, we refer readers to [26].

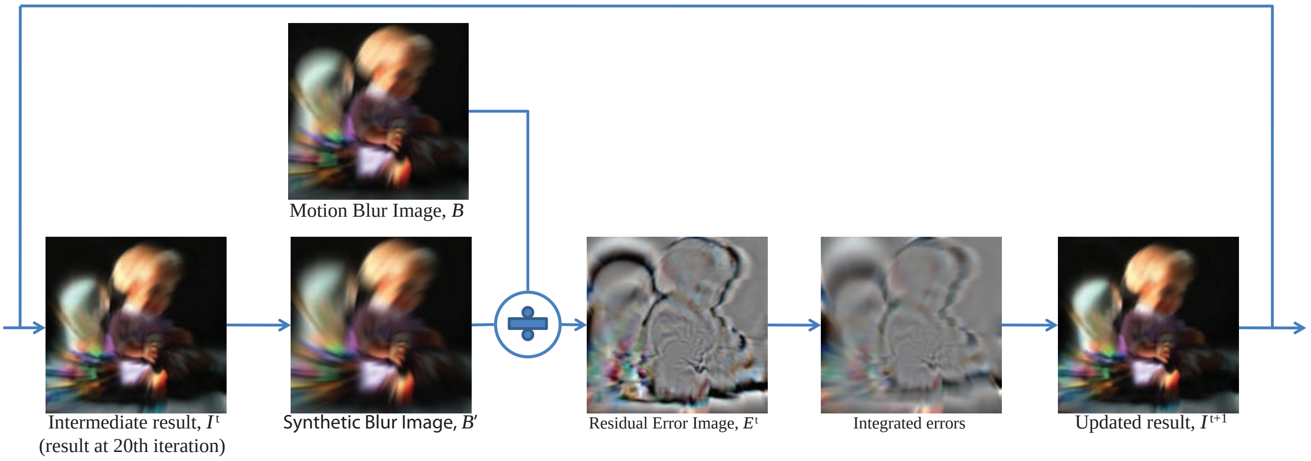

Motion Blur Image, B
Intermediate result, I^t
(result at 20th iteration)
Synthetic Blur Image, B'
Residual Error Image, E^t
Integrated errors
Updated result, I^{t+1}

Fig. 5. Overview of the projective motion RL algorithm. Given the current estimation I^t, we compute a synthetic blur image B' according to the given motion in terms of H_i. The residual error image $E^t = B/B'$ is computed by pixel-wise division. The residual errors are then integrated according to H_i^{-1} to produce an updated estimation I^{t+1}. The I^{t+1} is then used as the initial guess for the next iteration. This process is iterated until convergence or after a fixed number of iterations. In our implementation, the number of iterations is fixed to 500.

Since $\sum_{y \in k} k(y) = 1$, we have $\sum_{y \in k} k(y-x) = 1$. After adding the iteration index, t, we get:

$$I^{t+1}(x) \quad = \quad I^t(x) \sum_{y \in k} \frac{B(y)}{\sum_{z \in k} I^t(z) k(y-z)} k(y-x). \qquad (8)$$

Utilizing the convolution operation for the whole image, we obtain the RL algorihm:

$$I^{t+1} \quad = \quad I^t \times \tilde{k} \otimes \frac{B}{k \otimes I^t}, \qquad (9)$$

where $\tilde{k}$ is the transpose of k that flips the shape of k upside-down and left-to-right, $\otimes$ is the convolution operation and $\times$ is a pixel-wise multiplication operation. To understand Equation (9), we can consider that $B'^t = k \otimes I^t$ is the prediction of a blurred image according to the current estimation of clear image I^t and the given point spread function k. Thus, B/B'^t is the residual errors (by pixel-wise division) between the real blurred image B and the predicted blurred image B'^t. The correlation operation ($\tilde{k}\otimes$) integrates the residual errors distributed according to $\tilde{k}$. The update rule in Equation (9) essentially computes a clear image I^∞ that would generate the blurred image B given a known point spread function k. Typically, the algorithm start with an initial guess of $I^0 = B$.

B. Projective Motion Richardson-Lucy algorithm

With the basic Richardson-Lucy algorithm, we can derive our projective motion Richardson-Lucy algorithm. From the projective motion blur model defined in Equation (3), we can integrate the motion path at each pixel location y and define a spatially varying motion PSF k_y:

$$B(y) = \frac{1}{N} \sum_{i=1}^{N} I(H_i x) = \sum_{x \in k_y} I(x) k_y(x), \qquad (10)$$

and

$$k_y(x) = \begin{cases} \frac{1}{N}, & if \quad x = H_i^{-1} y \\ 0, & otherwise \end{cases}, \qquad (11)$$

where $\sum_{x \in k_y} k_y(x) = \sum_{i=1}^{N} \frac{1}{N} = 1$. Because $x = H_i^{-1} y$ does not correspond to a discrete integer pixel coordinate, bicubic interpolation is used to estimate the pixel values for the points x.

Substituting and replacing Equation (5) with Equation (10) for the RL algorithm, we get:

$$I^{t+1}(x) \quad = \quad I^t(x) \sum_{y \in k_x} \frac{B(y)}{\sum_{z \in k_y} I^t(z) k_y(z)} \tilde{k}_x(y), \qquad (12)$$

which is the general form for spatially varying motion deblurring using the RL algorithm. The motion path in $\tilde{k}_x$ is the reverse direction of motion path in k_x.

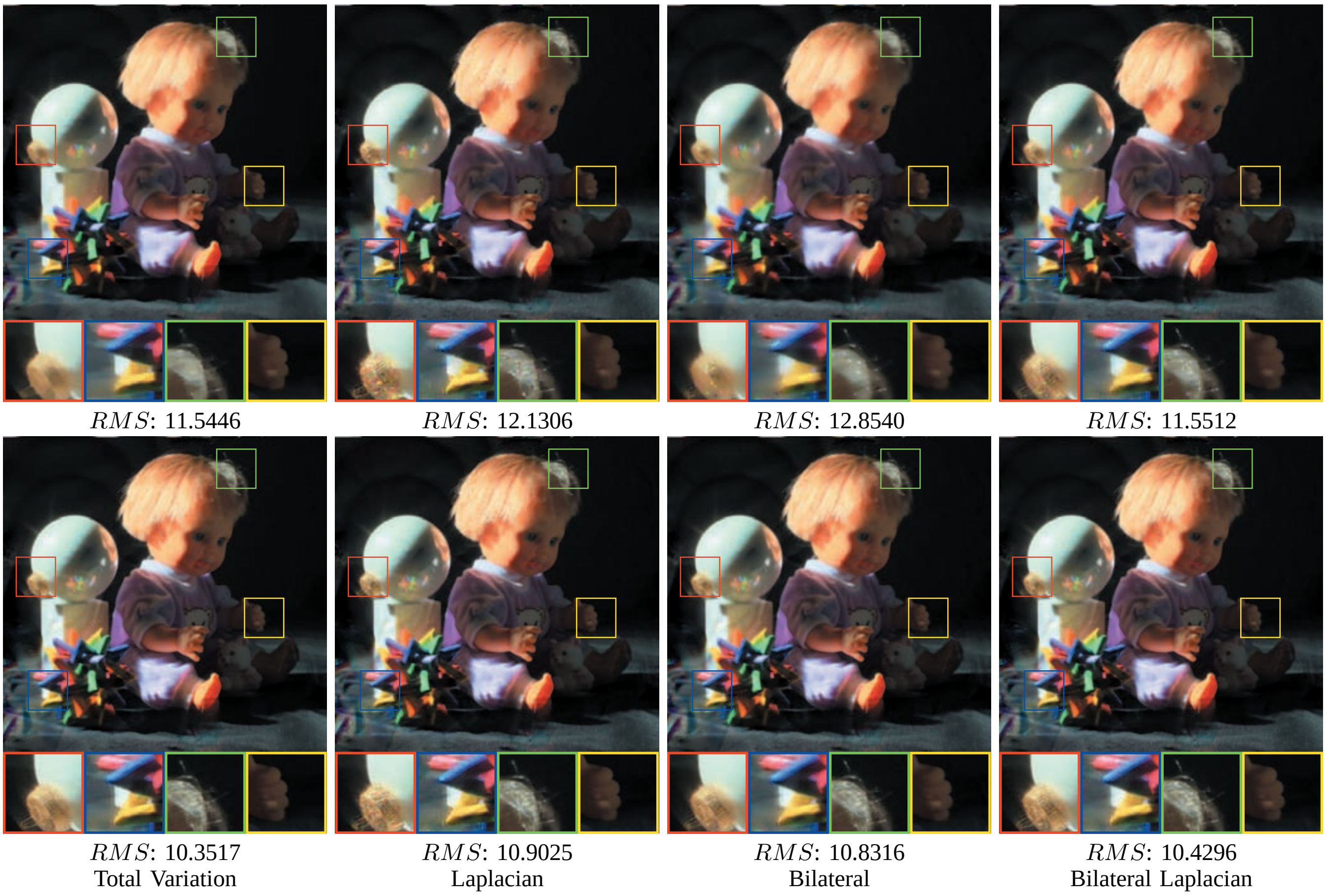
Since the motion path in k_x, according to Equation (10), can be described by a sequence of homographies, $H_1 \ldots H_N$, we can also group the motion path of $\tilde{k}_x$. Grouping the motion path of $\tilde{k}_x$ forms a new sequence of homographies which is the original homography sequence but with each matrix inverted and applied in reverse order, i.e. $H_N^{-1} \ldots H_1^{-1}$. For each point along the motion path in $H_1 \ldots H_N$, $H_N^{-1} \ldots H_1^{-1}$ reverse the transformation and integrate the errors along the motion path. Thus, we obtain the iterative update rule for the projective motion blur model:

$$I^{t+1} = I^t \times \frac{1}{N} \sum_{i=1}^{N} E^t(H_i^{-1} x), \qquad (13)$$

where $E^t(x) = \frac{B(x)}{\frac{1}{N} \sum_{i=1}^{N} I^t(H_i x)}$ is the residual error between the real blurred image B and the predicted blurred image $B'^t = \frac{1}{N} \sum_{i=1}^{N} I^t(H_i x)$. Note that although we computed the per-pixel motion PSF during the derivation, Equation (13) processes the image as a whole and does not need to reconstruct the per-pixel motion PSF explicitly. This is similar to the global convolution and correlation process in the conventional RL algorithm. In essence, our approach replaces the convolution and correlation operators in the conventional RL algorithm with a sequence of forward projective motions and their inverses via the homographies. Figure 5 diagrams our projective motion RL algorithm.

V. Adding Regularization

The result derived in the previous section is a maximum likelihood estimation with respect to Poisson noise models and imposes no regularization on the solution. Recent deblurring works (e.g. [10], [16], [33]) have shown that imposing certain image priors can significantly reduce deconvolution artifacts. In this section,



RMS: 11.5446 RMS: 12.1306 RMS: 12.8540 RMS: 11.5512

RMS: 10.3517 Total Variation

RMS: 10.9025 Laplacian

RMS: 10.8316 Bilateral

RMS: 10.4296 Bilateral Laplacian

Fig. 6. Comparisons on different regularization: Total Variation (TV) regularization, Laplacian regularizatio gdn, Bilateral regularization and Bilateral Laplacian regularization. The results on the top row are produced with fixed $\lambda = 0.5$; The results on the bottom row are produced with our suggested implementation scheme which λ decreases progressively as iterations increase. The number of iterations and parameters setting (except λ) are the same for all testing case. Our implementation scheme is effective as the deblurring quality improves in terms of both visual appearance and RMS errors. The input blur image and ground truth image can be found in Figure 1(a) and (d) respectively.

we first derive the general form for including a regularization term in the Richardson-Lucy algorithm. We then describe how regularization based on total variation [10], Laplacian [16], and bilateral regularization in [33] can be incorporated into our algorithm. The goal of this section is not to compare and evaluate the performance of different regularization in the deconvolution process. Instead, we want to show the capability of our projective motion RL to include regularization terms that have been used with the conventional RL. In addition, we discuss implementation details that allow the regularization terms to be more effective and avoid the tuning of the parameter, λ, that controls the relative weight of regularization.

Regularization in Richardson-Lucy algorithm

We start with the derivation from the Bayes's equation. Suppose that $P(I)$ is the prior probability of the restored image I. Then, the objective function we want to optimize becomes:

$$\begin{aligned} & \arg\max_I P(I|B,k) \\ = & \arg\max_I P(B,k|I)P(I) \\ = & \arg\min_I -L(B,k|I) - L(I) \\ = & \arg\min_I \sum_{x\in I}(g(x) - B(x)\ln(g(x)) + \lambda R(I(x)), \end{aligned} \tag{14}$$

where $L(\cdot) = \ln(P(\cdot))$, $R(I) = -\frac{1}{\lambda}L(I)$ is the regularization term and λ is the relative weight between the data term and the regularization term in Equation (14). Computing the first derivative of Equation (14) with respect to $I(x)$ and setting it equal to zero, we get:

$$\sum_{y\in k} \frac{B(y)}{\sum_{z\in k} I^t(z)k(y-z)} k(y-x) = 1 - \lambda\nabla R(I(x)), \tag{15}$$

where $\nabla R(I(x))$ is the first derivative of $R(I(x))$ with respect to $I(x)$. In order for the RL algorithm to converge, we need $\frac{I^{t+1}}{I^t} = 1$ as $t \to \infty$. Hence, the regularized RL algorithm becomes:

$$I^{t+1}(x) = \frac{I^t(x)}{1 - \lambda\nabla R(I^t(x))} \sum_{y\in k} \frac{B(y)k(y-x)}{\sum_{z\in k} I^t(z)k(y-z)}, \tag{16}$$

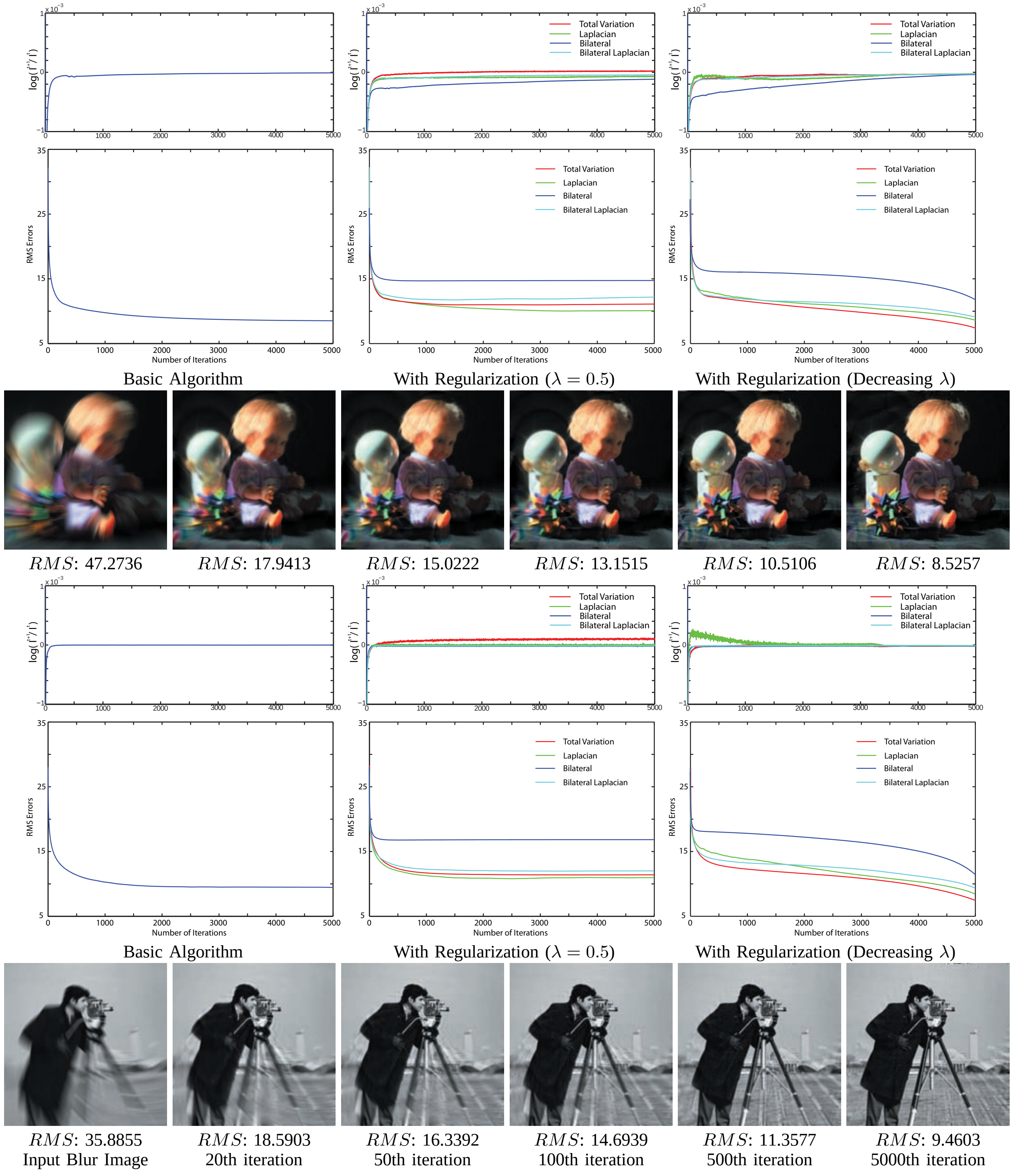
and

$$I^{t+1} = \frac{I^t}{1 - \lambda\nabla R(I^t)} \times \tilde{k} \otimes \frac{B}{k \otimes I^t}. \tag{17}$$

Similarly, for our projective motion RL algorithm, we can derive from Equation (16) to get a regularized version of the update rule:

$$I^{t+1}(x) = \frac{I^t(x)}{1 - \lambda\nabla R(I^t)} \times \frac{1}{N}\sum_{i=1}^{N} E^t(H_i^{-1}x). \tag{18}$$

In the following, we describe several regularization approaches commonly used with motion deblurring.



Basic Algorithm

With Regularization ($\lambda = 0.5$)

With Regularization (Decreasing λ)

RMS: 47.2736 RMS: 17.9413 RMS: 15.0222 RMS: 13.1515 RMS: 10.5106 RMS: 8.5257

Basic Algorithm

With Regularization ($\lambda = 0.5$)

With Regularization (Decreasing λ)

RMS: 35.8855 Input Blur Image

RMS: 18.5903 20th iteration

RMS: 16.3392 50th iteration

RMS: 14.6939 100th iteration

RMS: 11.3577 500th iteration

RMS: 9.4603 5000th iteration

Fig. 7. Convergence rates of our projective motion RL algorithm. The first and the fourth rows show the plot of I^{t+1}/I^t, the second and the fifth rows show the plot of RMS errors against number of iterations and the input motion blur image for our basic algorithm, regularized algorithm with fixed weight ($\lambda = 0.5$), and regularized algorithm with decreasing weight (linearly from 1 to 0) as the number of iterations increase. Note that the motion blur are different for the two test cases. The third and the sixth row show the input motion blur image and intermediate results at the 20th, 50th, 100th, 500th and 5000th iterations for the basic algorithm.

Total variation regularization Toal variation (TV) regularization has been demonstrated by Chan and Wong [6] and Dey *et al.* [10]. The purpose of introducing this regularization is to suppress image noise amplified during the deconvolution process by minimizing the magnitude of gradients in the deblurred image:

$$R_{TV}(I) = \int \sqrt{||\nabla I(x)||^2} dx \tag{19}$$

where $\nabla I(x)$ is the first order derivative of $I(x)$ (in x and y direction). Substituting this regularization term into Equation (18), we get:

$$I^{t+1}(x) \;=\; \frac{I^t(x)}{1 - \lambda \nabla R_{TV}(I)} \times \frac{1}{N} \sum_{i=1}^{N} E^t(H_i^{-1} x) \tag{20}$$

where $\nabla R_{TV}(I) = -\nabla \frac{\nabla I^t}{|\nabla I^t|}$. As reported in [10], $\lambda = 0.002$[3] is used in their experiments.

Laplacian regularization The Laplacian regularization, sometimes called the sparsity regularization, asserts that for natural images the histogram of the gradient magnitudes should follow a heavy-tailed distribution that can be modeled by a Laplacian distribution. The Laplacian regularization, suggested by [16], takes the following form:

$$R_L(I) = \exp(-\frac{1}{\eta}|\nabla I|^d), \tag{21}$$

where d is a parameter controlling the shape of distribution, and the term $\eta = 0.005$ (according to [16]) is the variance of the image noise. In [16], d is set to 0.8. In our implementation and the source code provided[4] we follow the same parameter settings. The effect of Laplacian regularization is also to suppress noise and to reduce small ringing artifacts.

Adding this regularization into our projective motion RL algorithm, we obtain another set of update rules:

$$I^{t+1}(x) \;=\; \frac{I^t(x)}{1 - \lambda \nabla R_L(I)} \times \frac{1}{N} \sum_{i=1}^{N} E^t(H_i^{-1} x) \tag{22}$$

where

$$\nabla R_L(I) = -\frac{1}{\eta} \exp(-\frac{1}{\eta}|\nabla I|^d)|\nabla I|^{d-1}\nabla^2 I. \tag{23}$$

A typical value for λ (according to the implementation of [16]) is between 0.001 to 0.004. We note that in the implementation of [16] a slightly different formulation is used:

$$\nabla R_L(I) = -\frac{1}{|\nabla I|^{d-2}} \nabla^2 I. \tag{24}$$

The effects of the two regularization schemes in Equation (23) and Equation (24), however, are similar with larger regularization weight given to smaller gradient and vice versa. In our implementation, we use the method in Equation (23).

Bilateral regularization In order to suppress visual artifacts while preserving sharp edges, Yuan *et al.* [33] proposed an edge-preserving bilateral regularization cost:

$$R_B(I) = \sum_x \sum_{y \in N(x)} g_1(||x-y||^2)(1 - g_2(||I(x)-I(y)||^2)), \tag{25}$$

where $N(x)$ is local neighborhood of x, $g_1(\cdot)$, $g_2(\cdot)$ are two Gaussian functions with zero mean and variance of σ_s^2, σ_r^2 respectively. In [33], the deblurring process is performed in a multi-scale fashion with the spatial variance of Gaussian blur kernel from one level to another set to be $\sigma_s^2 = 0.5$. The term σ_r^2 is the range variance and it is set to be $0.01 \times |\max(I) - \min(I)|$. The size of local neighborhood $N(x)$ is determined by σ_s.

In [33], visual artifacts are progressively suppressed by both inter-scale and intra-scale regularization. The bilateral regularization corresponds to the inter-scale regularization. Our iterative update rule with the bilateral regularization term are derived as [33]:

$$I^{t+1}(x) \;=\; \frac{I^t(x)}{1 - \lambda \nabla R_B(I)} \times \frac{1}{N} \sum_{i=1}^{N} E^t(H_i^{-1} x), \tag{26}$$

where

$$\nabla R_B(I) \;=\; \sum_{y \in N(x)} (I_y^d - I_y^d D_y), \tag{27}$$

$$I_y^d(x) \;=\; g_1(||x-y||^2) g_2(||I(x) - I(y)||^2) \frac{(I(x) - I(y))}{\sigma_r}. \tag{28}$$

The term D_y is a displacement operator which shifts the entire image I_y^d by the displacement vector $(y - x)$, and λ (as reported in [33]) is set to be a decay function whose weight decreases as the number of iterations increases. The effect of this bilateral regularization is similar to the effect of the Laplacian regularization[5], however, we found that the bilateral regularization tends to produce a smoother result not only because its use of a Gaussian distribution for $g_2(\cdot)$, but also because the larger local neighborhood size $N(x)$. In our implementation, we also include a regularization where $g_2(\cdot)$ follows a Laplacian distribution. We call this a "Bilateral Laplacian" regularization. In our implementation, the parameters of $g_2(\cdot)$ is set to be the same as that used in [16].

Implementation Details One major challenge in many deblurring approaches is the need for parameter tuning. In our case, the most important parameter is the term λ which adjusts the relative weight between the data term and the regularization term. If λ is too large, details in the deblurred results can be overly smoothed. On the other hand, if λ is too small, deconvolution artifacts cannot be suppressed.

In our experiments, we found that having a large initial λ that is progressively decreased as the number of iterations are increased produces the best results. This implementation scheme has been used in [24], [33]. In regards to its effectiveness, [17] analysis of this approach noted that this scheme allows the energy function to converge to a local minima, but this local minima is closer to the ground truth image than its global minima. In our implementation, we divided the iterations into five sets each containing 100 iterations. The λ are set to be $1.0, 0.5, 0.25, 0.125, 0.0$ in each set. We ran our deblurring algorithm starting from the largest λ to the smallest λ. One interesting observation we found is that under this implementation scheme the total variation regularization produces the best results in terms of root-mean-square (RMS) error[6]. One explanation is that the Laplacian and Bilateral regularization tend to protect false edges that arise from deconvolution artifacts.

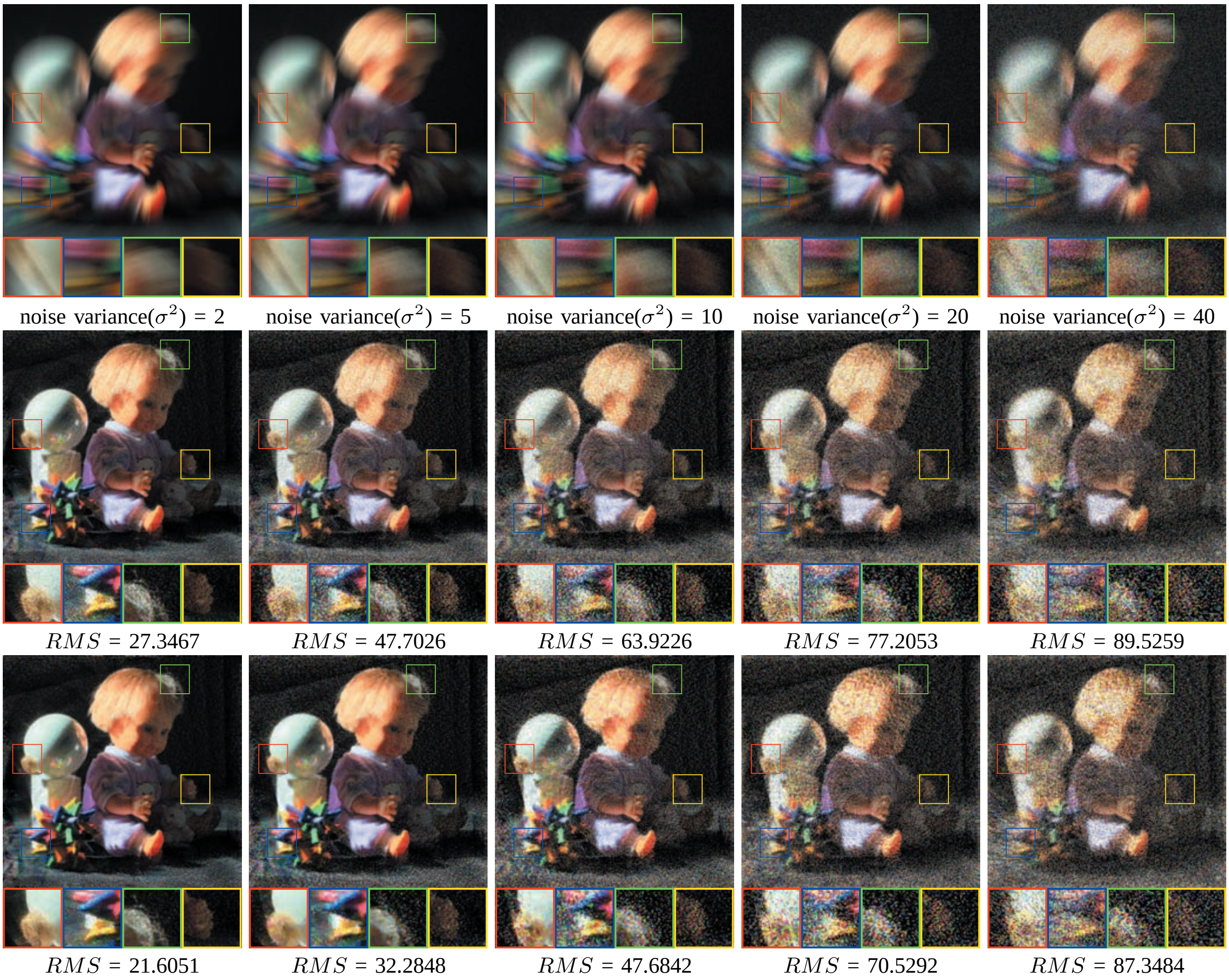
The visual appearance of our deblurred results with different regularization are similar and all of the previously discussed

[3] This is a weight for image intensity between 0 and 1. For image intensity between 0 and 255, a normalized weight should be equal to $0.002 \times 255 = 0.51$.

[4] http://yuwing.kaist.ac.kr/projects/projectivedeblur/index.htm

[5] Laplacian and bilateral regularization are essentially the same if $N(x)$ is the first order neighborhood and $g_2(\cdot)$ is a Laplacian function.

[6] The RMS error measured for image intensity between 0 to 255.



noise variance(σ^2) = 2 noise variance(σ^2) = 5 noise variance(σ^2) = 10 noise variance(σ^2) = 20 noise variance(σ^2) = 40

RMS = 27.3467 RMS = 47.7026 RMS = 63.9226 RMS = 77.2053 RMS = 89.5259

RMS = 21.6051 RMS = 32.2848 RMS = 47.6842 RMS = 70.5292 RMS = 87.3484

Fig. 8. Evaluation on the robustness of our algorithm by adding different amount of noise in blurred images. Top row: noisy blurred image, the amplitude of noise is determined by the noise variance (σ^2). Second row: Deblurring results with our basic projective motion RL algorithm. Third row: Deblurring results with total variation regularization. In the presence of image noise, our deblurring algorithm amplified the image noise in deblurred results. The effects of regularization hence become significant in suppressing amplified image noise.

regularization schemes can successfully improve the visual quality of the deblurred image. One important thing to note is that in the last set of iterations, we do not impose any regularization but we start with a very good initialization that is produced from the regularized deblurring algorithm. We found that even without regularization, having a good initialization can effectively reduce ringing and noise amplification.

VI. Motion Estimation

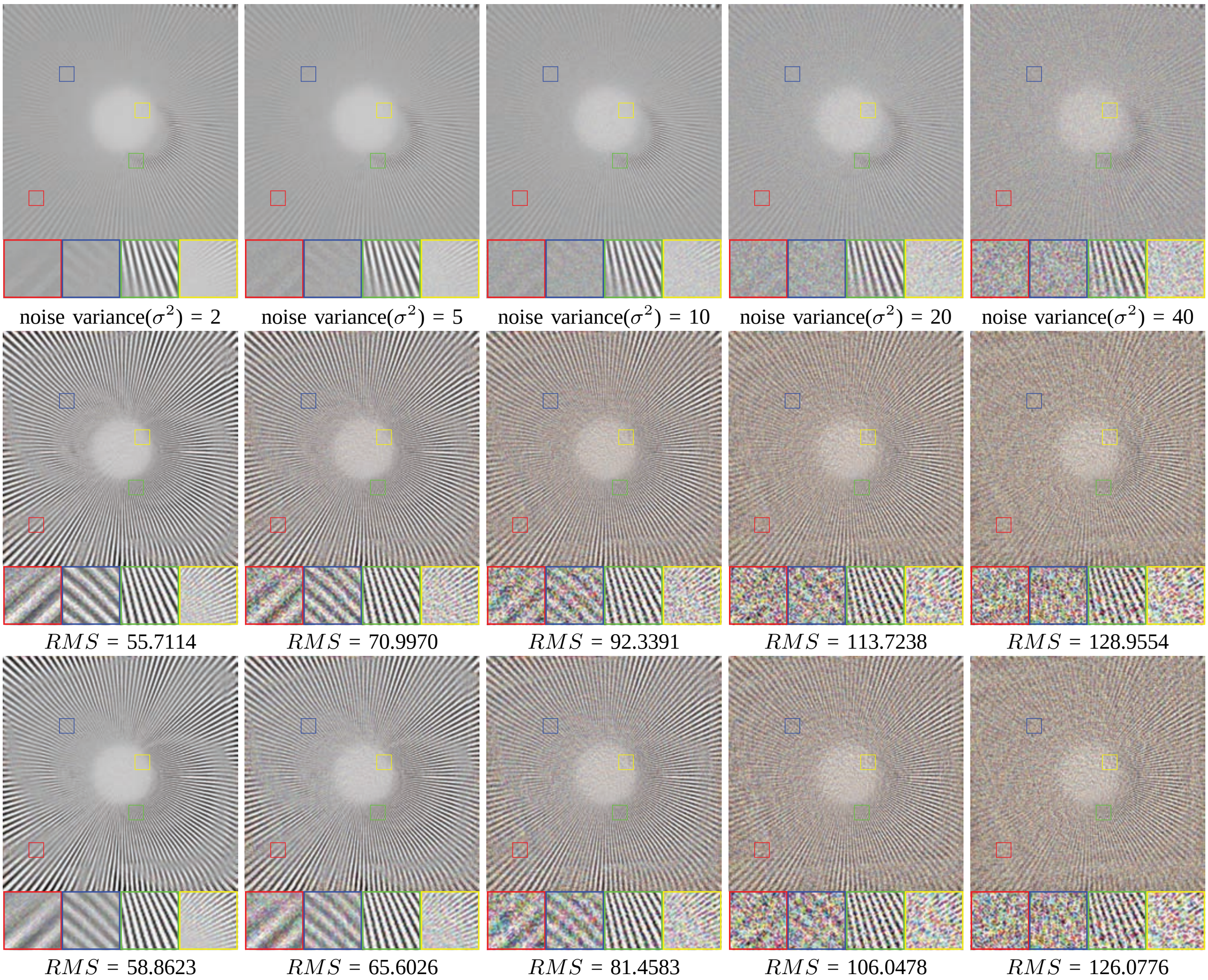
Although the focus of this paper is the derivation of our projective motion path blur model and its associated deblurring algorithm, for completeness we describe some promising methods to compute the projective motion within the exposure time. General algorithms for motion estimation is part of future research.

Auxiliary Hardware or Imaging Modifications A direct method to estimate camera motion in terms of homographies during the exposure is to use a hybrid camera [4], [28] that captures an auxiliary high-frame-rate low-resolution video. From the auxiliary video, motion trajectory at each pixel (i.e. optical flow) can be computed as done in [28] and sequence of homographies can be fitted to each frame. Another promising hardware coupling is the use of accelerometers and inertial sensors for estimating the cameras motion as demonstrated in [12] and [14]. Recently, Tai *et al.* [29] showed how a coded exposure could be used to capture an image containing a motion blurred moving object. Through the analysis of motion discontinuities protected by the coded exposure, homographies describing the motion path could be computed.

Uniform Motion Assumption If the motion blur is caused by a constant motion path over the exposure time, then the path can be described as $h_1 = h_2 = \cdots = h_N$. According to the definition of $H_i = \prod_{j=1}^{i} h_j$, we can get

$$h_i = \sqrt[N]{H_N} \qquad 1 \leq i \leq N. \tag{29}$$

Thus, we can obtain h_i by computing the N-th matrix root [5] of H_N, and the estimation of the series of h_i is reduced to the estimation of H_N. The easiest technique to estimate this uniform motion is described by Shan *et al.* [25] which relies on the user


noise variance(σ^2) = 2

noise variance(σ^2) = 5

noise variance(σ^2) = 10

noise variance(σ^2) = 20

noise variance(σ^2) = 40

RMS = 55.7114

RMS = 70.9970

RMS = 92.3391

RMS = 113.7238

RMS = 128.9554

RMS = 58.8623

RMS = 65.6026

RMS = 81.4583

RMS = 106.0478

RMS = 126.0776

Fig. 9. Evaluation on the robustness of our algorithm by adding different amount of noise in blurred images. Top row: noisy blurred image, the amplitude of noise is determined by the noise variance (σ^2). Second row: Deblurring results with our basic projective motion RL algorithm. Third row: Deblurring results with total variation regularization. In the presence of image noise, our deblurring algorithm amplified the image noise in deblurred results. The effects of regularization hence become significant in suppressing amplified image noise.

to supply image correspondences to establish the transformation. Another highly promising technique is that proposed by Dai and Wu [8] that uses the blurred objects alpha matte to estimate H_N. In our experiments, we use a former approach in [25]. Recently, Li *et al.* [19] proposed a method to estimate a global homography between successful video images to perform mosaicing. The frames could also be deblurred using a motion path model similar to that proposed in this paper.

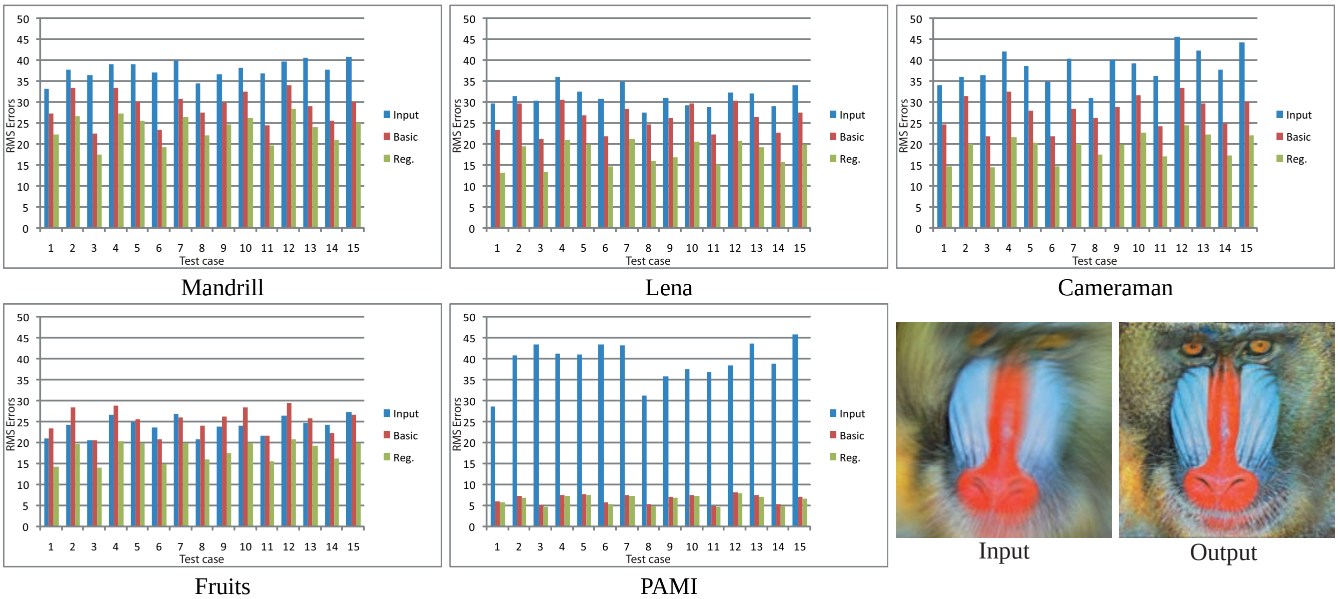
Recently, Whyte *et al.* [31] use a variational Bayesian approach to estimate the weight of quantized rotation parameters without the need for auxiliary hardware or a uniform assumption. Their estimation of the motion PSF, however, is not sufficiently accurate leading to deconvolution artifacts in the deblurred result. This work concurrently showed a method to estimate the spatially varying PSF using blurred and noisy image pairs.

VII. Experiment Results

In this section, we empirically examine our algorithm's convergence properties by plotting the RMS error against the number of iterations. Robustness is analyzed by comparing the results with different amounts of additive noise. We also evaluate the quality of our projective motion RL with and without regularization by creating a set synthetic test cases. Our results are also compared against spatially invariant deblurring method, e.g. [16], with synthetic test cases that resemble real motion blurred images caused by camera shake. Finally, we show results on real images for which the projective motion paths were estimated using methods described in Section VI.

A. Convergence Analysis

While the conventional RL algorithm guarantees convergence, in this section, we empirically examine the convergence of our projective motion RL algorithm. At each iteration, we compute the log ratio $\log(I^{t+1}/I^t)$ and the RMS error of the current result against the ground truth image. We run our algorithm for a total of 5,000 iterations for each case. The convergence rates of our basic projective motion RL, regularized algorithm with fixed regularization weight ($\lambda = 0.5$, for intensity range between 0



Mandrill

Lena

Cameraman

Fruits

PAMI

Input

Output

Fig. 10. RMS pixel errors for different examples and test cases in our synthetic trials. We compare the RMS of the input blur image (blue), the deblurred image using basic projective motion RL (red) and the deblurred image using projective motion RL with (Total Variation) regularization (green). **All tested images and results are available in the supplemental materials. An example (input and output) from these results are shown here.**

and 255) and regularized algorithm with decreasing regularization weight (linearly from 1 to 0) are compared. Plotted is the RMS error with respect to the ground truth image. Figure 7 shows the graphs plotting RMS errors against the number of iterations.

Typically, our method converges within 300 to 400 iterations for both the basic algorithm and regularized algorithm with fixed regularization weight. As the number of iterations increases, the difference of RMS errors between successive iterations decreases, however, after 500 iterations the visual improvement in the deblurred result is unnoticeable as shown in some intermediate results in Figure 7. We found that the algorithm with regularization produced RMS errors that are higher than the basic algorithm. The main reason is that the test cases demonstrated here are noise-free, and the regularization tends to smooth out high frequency details resulting in higher RMS error. However, as we will show in the next subsection, when noise is present in the image, incorporating regularization becomes effective. Using the decreasing regularization weight scheme we still observe a decrease in RMS errors even at 5000 iterations. This approach also achieves RMS errors lower than both the basic algorithm and regularized algorithm using a fixed regularization weight. This convergence study confirms the effectiveness of this scheme observed by [24], [33].

B. *Noise Analysis*

To test for the robustness of our algorithm, we added different amounts of zero-mean Gaussian noise to the synthetic blur images. To simulate sensor noise, the added image noise does not undergo the convolution process, and is independent of the motion blur effect.

Figure 8 and Figure 9 show our deblurring results with different amounts of Gaussian noise[7] added. We show results of our algorithm with and without regularization. For this experiment, we use the Poisson noise model with total variation regularization. The test pattern is a resolution chart. As expected, our deblurring results without regularization amplifies image noise like other deblurring algorithms. The quality of our deblurring algorithm degrades as the amount of noise increases. In addition, larger motions tend to produce noisier results. In such cases, the added noise in the image around the center of rotation becomes less apparent than those in image regions with larger motion blur (i.e. the image boundaries). In the presence of image noise, the regularization term becomes important to improve the visual quality of the results. Better results can be achieved by increasing the regularization weight as the noise level increased. However, for fairness of comparisons, we use the same regularization weight as discussed in Section V. The difference between the regularized and un-regularized results are significant both in terms of visual quality and RMS errors. However, when the amount of image noise added is very large, e.g. $\sigma^2 \geq 20$, the regularization term cannot suppress image noise effectively.

C. *Qualitative and Quantitative Analysis*

Figure 1 has already shown a synthetic example of spatially varying motion blur with known h_i. To further evaluate our algorithm quantitatively, we have created a test set consisting of fifteen test cases and five test images: *Mandrill*, *Lena*, *Cameraman*, *Fruits* and *PAMI*. The *Mandrill* example contains significant high frequency details in the hair regions. The *Lena* example contains both high frequency details, smooth regions and also step edges. The *Cameraman* image is a monotone image but with noise added independently to each RGB channel. The *Fruits* example also contains high frequency details and smooth regions. Lastly, the *PAMI* example is a text-based binary image (i.e. black and white). For each test case, we add additional Gaussian noise ($\sigma^2 = 2$) to simulate camera sensor noise. The parameters setting are the same for all test cases and the values we used are the same as in the implementation of [16].

[7] The noise variance σ^2 of Gaussian noise added are with respect to intensity range between 0 and 255.

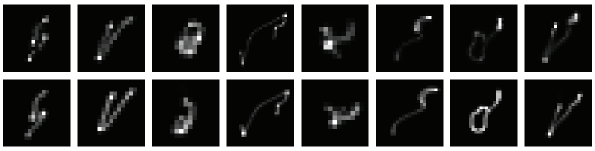


Fig. 11. Top row: Ground truth PSF from [17]. Bottom row: Our approximated PSF using projective motion paths.

We show the RMS of the input motion blurred image, the RMS of deblurred image using our basic projective motion RL algorithm and the RMS of the deblurred image with regularization (total variation) are provided. Figure 10 shows our results. Our projective motion RL is effective, especially when regularization is used with our suggested implementation scheme. These test cases also demonstrate that our approach is effective in recovering spatially varying motion blur that satisfied our projective motion blur assumption. We note that in some test cases, e.g. examples from the *Fruits* test case, the RMS errors of the deblurred results are larger than the RMS errors of input motion blurred images. This is due to the effects of amplified noise. After noise suppression with regularization, the RMS errors of the deblurred results are smaller than the RMS errors of input motion blurred images.

D. Comparisons with spatially invariant method

To evaluate our projective motion blur model for the motion blur effects caused by camera shake, another test set was created to simulate blur from camera shake motion. Our rests are compared against results obtained using a spatially invariant blur model based on the deblurring algorithm in [16].

This test set consists of 8 test cases and 5 images: *Doll*, *Einstein*, *Lotus*, *Tower* and *Palace*. For each test case, we first use homographies to approximate the shape and intensity variation of "real" motion blur kernels. The real kernels are from the ground truth kernels in the data provided by Levin *et al.* [17]. Figure 11 shows their ground truth kernels and our approximated kernels using homographies to describe the projective motion path. Since Levin *et al.* locked the Z-axis rotation handle of tripod when they captured the images, their data does not contain any rotation. However, real camera shake usually contains small amount of rotation. We added 1 degree of rotation in total to the overall motion path to simulate the effects of in-plane rotation of camera shake as shown in Figure 12. Although the motion blur of this test set contains rotation, the effect of rotation is almost invisible from the motion blurred images itself since the translational motion dominates the effects of motion blur. However, when we apply spatially invariant motion deblurring algorithm [16] to these images, the effects of rotation is obvious. Similar to the previous test set, we also added Gaussian noise ($\sigma^2 = 2$) to simulate camera sensor noise.

Figure 13 shows the RMS errors of our results (with TV regularization) compared with results from [16] using nine different PSFs sampled at different locations of the image. We use the source code provided by [16] with parameter $\lambda = 0.001$ to obtain the results for comparison. Our results consistently produce smaller RMS errors and less visual artifacts when compared to the results from [16]. These test cases also show the insufficiency of conventional spatially invariant motion to model camera shake. The results of [16] obtain good results around local regions where the PSF is sampled, however, visual artifacts become increasingly noticeable as we move farther away from the sampled point of the PSF used for deconvolution.

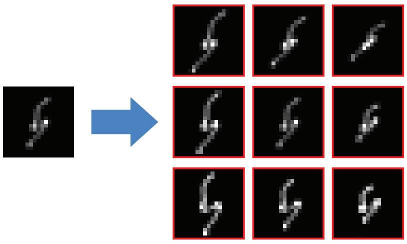


Fig. 12. Our simulated spatially varying motion blur PSF for camera shake. From the approximated kernels in Figure 11, we included 1 degree of rotation (in total) in the motion path. This is done by first estimating the PSF using homographies (Figure 11) and then introducing a rotation in the motion path of a single degree distributed over the whole range of motion. To demonstrate the effect of this rotation, we reconstructed the PSFs at nine different positions in the image. These reconstructed PSFs will be used by a spatially invariant motion deblurring algorithm [16] for comparison. With only 1 degree of rotation we can see a significant change to the camera shake PSFs in different regions of the image. Rotational motion for real camera shake (e.g. as shown in Figure 2) would be larger than shown here.

E. Real Examples

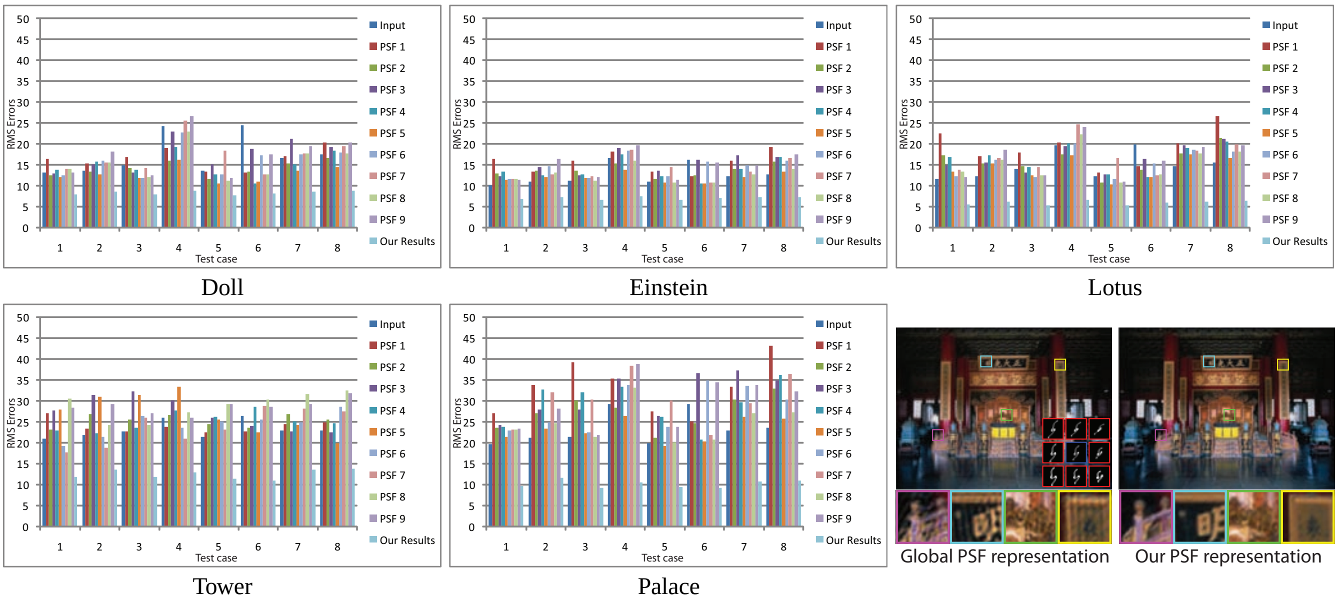
Figure 14 shows an example of global motion blur obtained from our previous work using a hybrid-camera system [28]. To obtain the motion path homographies, we use the motion vectors between in the low-resolution high-framerate camera as a set of point correspondences and fit a global homography per low-resolution frames. We also show the effectiveness of our regularization compared with previous results. Our approach achieves comparable result with [28], however we do not use the low-resolution images for regularization as done in [28].

Figure 15 and Figure 16 show more real examples with zoomed-in motion[8] and rotational motion respectively. These input images was obtained with a long exposure time with low ISO setting. The motion blurred matrix H_N is obtained by fitting the transformation matrix with user markup as shown in Figure 15(a) and Figure 16(a) respectively. Each h_i is computed by assuming the motion is a uniform. We show the deblurred results from our algorithm without and with regularization in (b) and (c). The ground truth image is shown in (d) for comparisons. We note that our real examples contain more visual artifacts than the synthetic examples. This is due to estimation errors in h_i. The effects of image noise in our real examples are not as significant as in our synthetic test case due to the long exposure time. We also note that the motions in our real examples are not as large as in our synthetic example.

VIII. Discussion and Conclusion

This paper has introduced two contributions for addressing image blur due to camera ego motion. The first is a formulation of the motion blur as an integration of the scene that has undergone a motion path described by a sequence of homographies. The advantage of this motion blur model is that it can better parameterize camera ego motion than the conventional approaches that rely on a uniform blur kernel. In addition, this "kernel

[8] The approach in [25] only discusses how to estimate rotational motion, but the estimation for zoom motion can be derived using a similar method.



Doll

Einstein

Lotus

Tower

Palace

Global PSF representation

Our PSF representation

Fig. 13. We compare our results with the spatially invariant motion deblurring algorithm in [16]. The RMS pixel errors for different examples and test cases are shown. For the results of [16], we sampled the PSF at nine different locations in the images and therefore obtain nine different PSFs for deblurring. Our results are compared to the results from [16] using all nine PSFs. Our approach consistently produces smaller RMS errors in all examples for all test cases.

free" formulation more closely models the physical phenomena causing the blur. Our second contribution is an extension to the RL deblurring algorithm to incorporate our motion blur model in a correction algorithm. We have outlined the basic algorithm as well as details on incorporating state-of-the-art regularization. Experimental results has demonstrated the effectiveness of our approach on a variety of examples.

The following discusses issues and limitations of this blur model, as well as a discussion on convergence, running time, and future work.

A. Conventional PSF representation versus projective model blur model

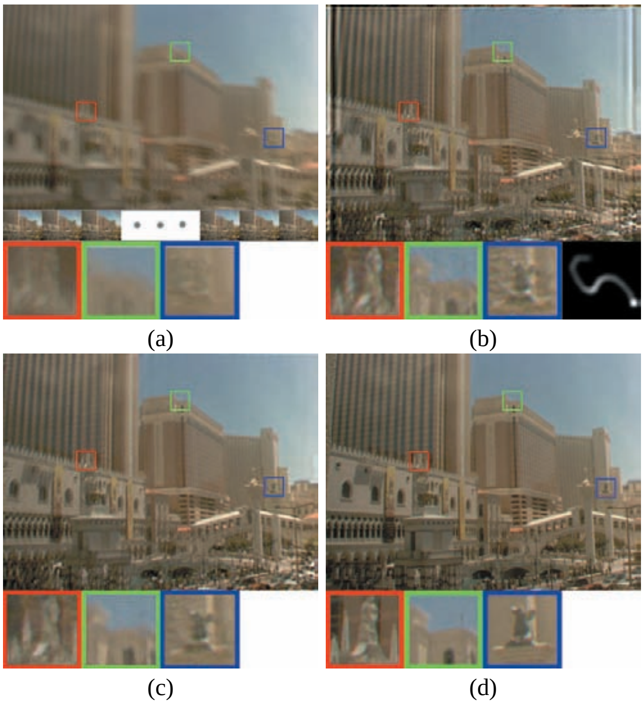
While this paper advocates a new blur model for camera ego motion to replace the conventional kernel-based approach, we note that the conventional representation has several advantages. First, the kernel-based PSF provides an easy to understand and intuitive represents of "point spread" about a point. Second, the kernel-based model can include other global blurring effects (e.g. defocus) and motion blur in a unified representation, while our representation only targets motion blur from camera motion. Third, by assuming the motion blur is globally the same, image deconvolution can be done in the frequency domain, while our projective motion RL algorithm can only be done in the spatial domain. As demonstrated several times in this paper, the conventional representation, however, is not suitable for dealing with spatially varying motion blur. For such cases, our projective motion RL formulation becomes advantageous.

B. Limitations

Our projective motion RL algorithm has several limitations similar to other deblurring algorithms. A fundamental limitation to our algorithm is that the high frequency details that have been lost during the motion blur process cannot be recovered. Our algorithm can only recover the hidden details that remain inside the motion blur images. Another limitation is that our approach does not deal with moving or deformable objects or scenes with occlusion and/or disocclusion or with significant depth variation. Figure 17, for instance, shows an object with out-of-plane rotational motion and large variation in relative depth captured with a coded exposure camera. Here, we assume the motion to be strictly horizontal, and estimate the motion PSF using only local observations around the (b) mouth, (c) ear, and (d) chair. Our deblurring results accurately recovered scene details only for the local regions used for PSF estimation; other regions are distorted by the incorrect PSF. Better results could potentially be obtained by first segmenting the image into different regions that each satisfies the projective motion model, and then applying our motion deblurring algorithm on each region separately. The problem with occlusions, disocclusions and depth variations is common to existing deblurring techniques, since the exhibited image blur cannot be adequately represented by a PSF in such cases. Other limitations of our approach include the problem of pixel color saturations and severe image noise as demonstrated in our experimental.

C. Running Time Analysis

Our current implementation takes about 15 minutes to run 500 iterations on an image size 512×512 with an Intel(R)CPU L2400@1.66Ghz and 512MB of RAM. The running time of our algorithm depends on several factors, including image size $|I|$, number of discrete sampling N (number of homographies) and the number of iterations T. Hence, the running time of our algorithm is $O(|I|NT)$. Comparing our running time to approaches which reconstruct pre-pixel PSF for deblurring (e.g. [28]), their running time is $O(|I|MT)$ where M is window size of the square blur kernel used to represent the PSF. Our approach has a significant advantage with $N << M$. In our experiments, we found that $N = 50$ is sufficient to model very large variations over the



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

Fig. 14. Image deblurring using globally invariant kernels. (a) Input from a hybrid camera (courtesy of [28]) where the high-frame-rate low-resolution images are also shown; (b) Result generated by [3](Standard RL algorithm from Matlab); (c) Result from our projective motion RL with regularization. (d) The ground truth sharp image. Close-up views and the estimated global blur kernels are also shown.

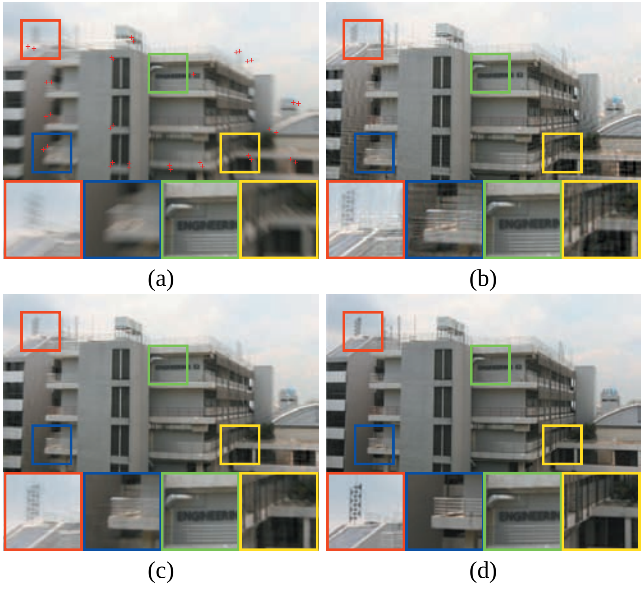
projective motion path. However, a conventional PSF kernel size M can be as large as $31 \times 31 = 961$ for small to mid-range motion. We also note that the majority of our running time is spent in the bicubic interpolation process necessary when applying the homographies. Significant speed-ups could undoubtedly be obtained with better implementation exploiting a GPU to perform the image warping.

D. Future Directions

There are several future directions of this work. One is to explore other existing algorithms and hardware (e.g. coded exposure and coded aperture) that can use our projective motion path blur formulation. Another important direction is to consider how our framework can be used to perform blind-deconvolution where the camera's motion path is unknown. Promising preliminary work by Whyte et al. [31] has already been performed in the case of rotation about the camera's optical center both in a blind estimation scenario and via image pairs.

Acknowledgments

We thank the Associate Editor and all reviewers for their thorough and constructive suggestions which are instrumental in improving the quality of the final paper. Yu-Wing Tai was supported by Korea Content Agency (KOCCA, No. N04100176) and KAIST seed grant (No. G04090064). Ping Tan was supported by Singapore grant R-263-000-555-112 and R-263-000-477-112. Michael S. Brown was supported by NUS RFC (No. R-252-000-423-112). A special thanks to Long Gao who did the initial implementation of the algorithm in Matlab.



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

Fig. 15. (a) Blurred input image and user markups for motion estimation; (b) Our result from using the basic projective motion RL algorithm; (c) Our result including regularization; (d) Ground truth image.

References

[1] A. Agrawal and R. Raskar. Resolving objects at higher resolution from a single motion-blurred image. In *CVPR*, 2007.

[2] J. Bardsley, S. Jefferies, J. Nagy, and R. Plemmons. Blind iterative restoration of images with spatially-varying blur. In *Optics Express*, pages 1767–1782, 2006.

[3] M. Ben-Ezra and S. Nayar. Motion Deblurring using Hybrid Imaging. In *CVPR*, volume I, pages 657–664, Jun 2003.

[4] M. Ben-Ezra and S. Nayar. Motion-based Motion Deblurring. *IEEE Trans. PAMI*, 26(6):689–698, Jun 2004.

[5] D. A. Bini, N. J. Higham, and B. Meini. Algorithms for the matrix p'th root. *Numerical Algorithms*, 39(4):349–378, 2005.

[6] T. F. Chan and C.-K. Wong. Total variation blind deconvolution. *IEEE Trans. Image Processing*, 7(3), 1998.

[7] S. Cho, Y. Matsushita, and S. Lee. Removing non-uniform motion blur from images. In *ICCV*, 2007.

[8] S. Dai and Y. Wu. Motion from blur. In *CVPR*, 2008.

[9] A. D. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from imcomplete data via the em algorithm. *J. R. Stat. Soc*, 39:1–38, 1977.

[10] 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 Imaging: Nano to Macro*, 2004.

[11] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman. Removing camera shake from a single photograph. *ACM Trans. Graph.*, 25(3), 2006.

[12] G. M. Hong, A. Rahmati, Y. Wang, and L. Zhong. Sensecoding: Accelerometer-assisted motion estimation for efficient video encoding. In *ACM Mutlimedia'08*, 2008.

[13] J. Jia. Single image motion deblurring using transparency. In *CVPR*, 2007.

[14] N. Joshi, S. B. Kang, L. Zitnick, and R. Szeliski. Image deblurring with inertial measurement sensors. *ACM Trans. Graph.*, 29(3), 2010.

[15] A. Levin. Blind motion deblurring using image statistics. In *NIPS*, pages 841–848, 2006.

[16] A. Levin, R. Fergus, F. Durand, and W. T. Freeman. Image and depth from a conventional camera with a coded aperture. *ACM Trans. Gr.*, 2007.

[17] A. Levin, Y. Weiss, F. Durand, and W. Freeman. Understanding and evaluating blind deconvolution algorithms. In *CVPR*, 2009.

[18] F. Li, J. Yu, and J. Chai. A hybrid camera for motion deblurring and depth map super-resolution. In *CVPR*, 2008.

[19] Y. Li, S. B. Kang, N. Joshi, S. Seitz, and D. Huttenlocher. Generating sharp panoramas from motion-blurred videos. In *CVPR*, 2010.

[20] L. Lucy. An iterative technique for the rectification of observed distributions. *Astron. J.*, 79, 1974.

[21] R. Raskar, A. Agrawal, and J. Tumblin. Coded exposure photography:

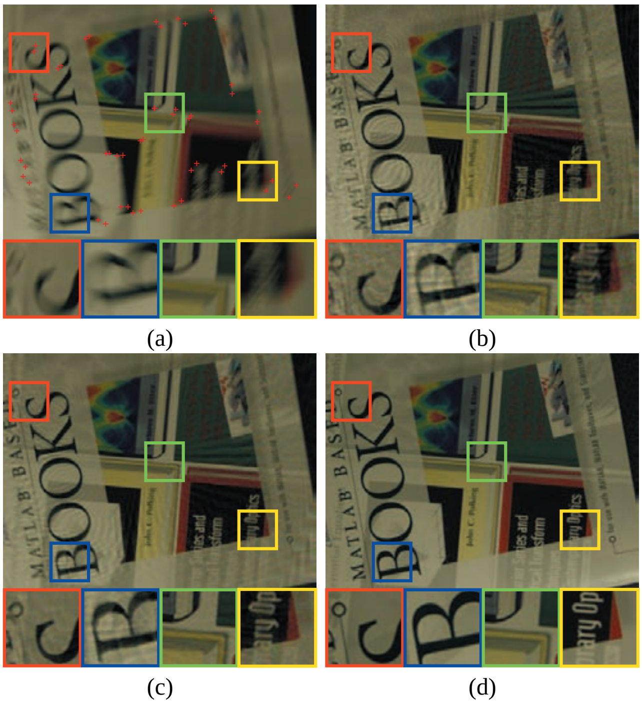


Fig. 16. (a) Blurred input image and user markups for motion estimation; (b) Our result from using the basic projective motion RL algorithm; (c) Our result including regularization; (d) Ground truth image.

motion deblurring using fluttered shutter. *ACM Trans. Graph.*, 25(3), 2006.

[22] W. Richardson. Bayesian-based iterative method of image restoration. *J. Opt. Soc. Am.*, 62(1), 1972.

[23] A. A. Sawchuk. Space-variant image restoration by coordinate transformations. *Journal of the Optical Society of America*, 64(2):138–144, 1974.

[24] Q. Shan, J. Jia, and A. Agarwala. High-quality motion deblurring from a single image. *ACM Trans. Gr.*, 2008.

[25] Q. Shan, W. Xiong, and J. Jia. Rotational motion deblurring of a rigid object from a single image. In *ICCV*, 2007.

[26] L. A. Shepp and Y. Vardi. Maximum likelihood reconstruction for emission tomography. *IEEE Trans. on Medical Imaging*, 1(2):113–122, 1982.

[27] Y. Tai, H. Du, M. Brown, and S. Lin. Image/video deblurring using a hybrid camera. In *CVPR*, 2008.

[28] Y. Tai, H. Du, M. Brown, and S. Lin. Correction of spatially varying image and video blur using a hybrid camera. *IEEE Trans. PAMI*, 32(6):1012–1028, 2010.

[29] Y. Tai, N. Kong, S. Lin, and S. Shin. Coded exposure imaging for projective motion deblurring. In *CVPR*, 2010.

[30] Y. Vardi. *Nonlinear Programming*. Englewood Cliffs, NJ: Prentice-Hall, 1969.

[31] O. Whyte, J. Sivic, A. Zisserman, and J. Ponce. Non-uniform deblurring for shaken images. In *CVPR*, 2010.

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

[33] L. Yuan, J. Sun, L. Quan, and H.-Y. Shum. Progressive inter-scale and intra-scale non-blind image deconvolution. *ACM Trans. Gr.*, 2008.

Appendix

Under Gaussian noise distribution If we assume that the image statistics follows a Gaussian distribution instead of a Poisson distribution, we can model the likelihood probability

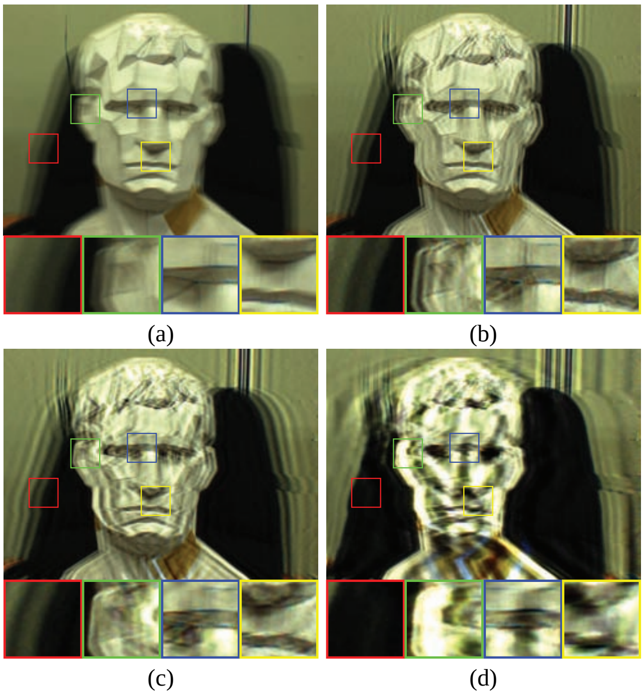


Fig. 17. Case of non-projective motion (Images from [29]). (a) Input. (b-d) Deblurring results with different motion PSFs estimated from different areas of the face.

$P(B, k|I)$ as follow:

$$\begin{aligned} & \arg\max_I P(B, k|I), \\ = \; & \arg\max_I \prod_{x \in I} \exp(-\frac{||g(x) - B(x)||^2}{\sigma_g^2}), \\ = \; & \arg\min_I \sum_{x \in I} ||g(x) - B(x)||^2, \end{aligned} \tag{30}$$

where $g(x) = \sum_{y \in k} I(y)k(x - y)$ is defined in Equation (5), and σ_g^2 is the variance of a Gaussian distribution. To solve Equation (30), we can derive an additive update rules based on the gradient-descent method as follow:

$$\begin{aligned} I^{t+1}(x) &= I^t(x) + \sum_{y \in k} \left(B(y) - \sum_{z \in k} I^t(z)k(y - z) \right) k(y - x), \\ I^{t+1} &= I^t + \tilde{k} \otimes (B - k \otimes I^t). \end{aligned} \tag{31}$$

Similarly, for our projective motion blur model, we can obtain:

$$I^{t+1} = I^t + \frac{1}{N} \sum_{i=1}^{N} E'^t(H_i^{-1} x), \tag{32}$$

where $E'^t(x) = B(x) - \frac{1}{N} \sum_{i=1}^{N} I^t(H_i x)$ is the residual error obtained by pixel-wise subtraction.

If we introduce regularization as a prior model $P(I)$, we now have to maximize a posterior probability $P(I|B, k)$. After some mathematical rearrangement, we can obtain the following energy function for minimization:

$$\begin{aligned} & \arg\max_I P(I|B, k), \\ = \; & \arg\max_I P(B, k|I)P(I), \\ = \; & \arg\min_I \sum_{x \in I} ||g(x) - B(x)||^2 + \lambda R(I), \end{aligned} \tag{33}$$

where $||g(x) - B(x)||^2$ is the data term, and $R(I) = -\frac{\sigma_g^2}{\lambda}\log(P(I))$ is the regularization term.

Computing the derivative of Equation (33), we can obtain another set of iterative update rule based on gradient-decent method:

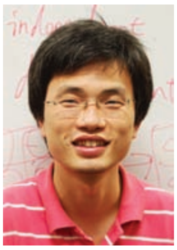
$$I^{t+1} \quad = \quad I^t + \tilde{k} \otimes (B - k \otimes I^t) + \lambda \nabla R(I^t), \tag{34}$$

and

$$I^{t+1} = I^t + \frac{1}{N}\sum_{i=1}^{N} E'^t(H_i^{-1}x) + \lambda \nabla R(I^t), \tag{35}$$

for the conventional motion blur model and our projective motion blur model respectively. Note that the regularization terms studied in Section V for the Poisson noise model can also be used for the Gaussian noise model, as well as the same implementation scheme for the adjustment of λ.

Yu-Wing Tai received the BEng (first class honors) and MPhil degrees in compute science from the Hong Kong University of Science and Technology (HKUST) in 2003 and 2005 respectively, and the PhD degree from the National University of Singapore (NUS) in June 2009. He joined the Korea Advanced Institute of Science and Technology (KAIST) as an assistant professor in Fall 2009. He regularly serves on the program committees for the major Computer Vision conferences (ICCV, CVPR, and ECCV). His research interests include computer vision and image/video processing. He is a member of the IEEE and ACM.



Ping Tan received the B.S. degree in Applied Mathematics from the Shanghai Jiao Tong University, China, in 2000. He received the Ph.D. degree in Computer Science and Engineering from the Hong Kong University of Science and Technology in 2007. He joined the Department of Electrical and Computer Engineering at the National University of Singapore as an assistant professor in October 2007. His research interests include image-based modeling, photometric 3D modeling and image editing. He has served on the program committees of ICCV, CVPR, ECCV. He is a member of the IEEE and ACM.

Michael S. Brown obtained his BS and PhD in Computer Science from the University of Kentucky in 1995 and 2001 respectively. He was a visiting PhD student at the University of North Carolina at Chapel Hill from 1998-2000. Dr. Brown is currently an Associate Professor in the School of Computing at the National University of Singapore. Dr. Brown regularly serves on the program committees for the major Computer Vision conferences (ICCV, CVPR, and ECCV). His research interests include Computer Vision, Image Processing and Computer Graphics. He is a member of the IEEE.