Optical Computing System for Fast Non-uniform Image Deblurring

Tao Yue yue-t09@mails.tsinghua.edu.cn
Jinli Suo {jlsuo, xyji, qhdai}@tsinghua.edu.cn
Xiangyang Ji
Qionghai Dai
Department of Automation, Tsinghua University, Beijing

Abstract

Removing non-uniform blurring caused by camera shake has been troublesome for its high computational cost. To accelerate the non-uniform deblurring process, this paper analyzes the efficiency bottleneck of the non-uniform deblurring algorithms and proposes to implement the time-consuming and repeatedly required module, i.e. non-uniform convolution, by optical computing. Specifically, the non-uniform convolution is simulated by an off-the-shelf projector together with a camera mounted on a programmable motion platform. Benefiting from the high speed and parallelism of optical computation, our system is able to accelerate most existing non-uniform camera shake removing algorithms extensively. We develop a prototype system which can fast compute non-uniform convolution for the blurring image of planar scene caused by 3D rotation. By incorporating it into an iterative deblurring framework, the effectiveness of proposed system is verified.

1. Introduction

Image blur caused by camera shake is a common degradation and key issue in computational photography. Perspective geometry tells that camera shake blur may be intensively varying in spatial domain. However, due to the high complexity and computational cost of non-uniform blurring models, for a long time studies on camera shake removal formulate camera shake blur with uniform convolution and propose many deblurring methods[17, 5, 18, 3, 25].

With the progress of image deblurring, researchers refocus their attentions on non-uniform blur model but suffer severely from the high computational cost. Although patch-wise based deblurring [9] can be used to solve the non-uniform deblurring problem efficiently, the approximation accuracy is limited for the intensively varying blurry cases. In spite of the recent progress in non-uniform deblurring, the low computation efficiency is still a limitation for the application of existing algorithms.

Most existing deblurring algorithms iteratively calculate blurred image from recent estimation of sharp image and camera motion, and then correct the sharp image and camera motion according to residual between calculated blurred image and captured one. For non-uniform deblurring algorithms, the blurring process is time-consuming because it need to be computed in pixel-wise and performed many times during the deblurring process. However, there is no accurate acceleration methods for this operation so far. This dilemma motivates us to explore acceleration approaches by resorting to assistance from hardware.

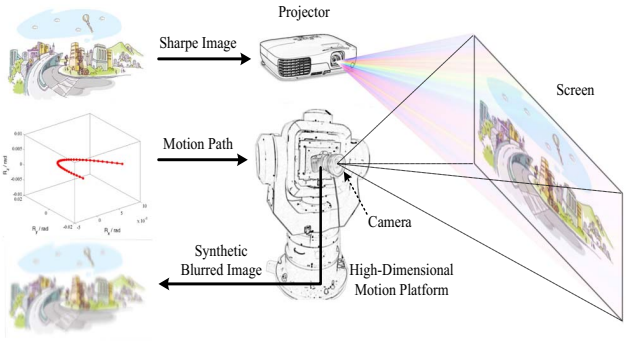


Figure 1. **The diagram of our optical computing system for fast non-uniform convolution.**

Optical computing is well studied in optics and many often-used mathematical operations can be accelerated by delicate optical systems[14, 1]. However, there is no existing optical computing system for the non-uniform convolution, which is a crucial module needs acceleration. Intuitively, the time-consuming pixel wise convolution corresponds to a spatially varying image blur process. This motivates us to build a new imaging system to physically simulate an imaging process (as shown in Fig. 1) that corresponds to the convolution exactly, and thus alleviate the computing process. In other words, we simulate the non-uniform computation directly instead of computing it pixel by pixel or approximating by patch-based methods. Specifically, we project the sharp image onto a planar screen as a synthetic scene and simulate the blurring process by imaging the screen using a shaken camera driven by a programmable motion platform. Based on this system, we

978-0-7695-4990-3/13 $26.00 © 2013 IEEE
DOI 10.1109/CVPRW.2013.138

build an optical computing framework incorporating above simulation into widely used deblurring iterations for non-uniform camera shake removal under constant depth assumption, which is widely used in the existing non-uniform deblurring algorithms [23, 24, 7, 22, 21, 12, 9, 10] except for [19] and [26].

To the best of our knowledge, this paper is the first attempt to address the efficiency issue in non-uniform deblurring by optical computing. The main contributions of this paper are: (1) build an optical computation framework for fast non-uniform deblurring; (2) accelerate the non-uniform convolution operation in non-uniform deblurring algorithms by a motion platform based system; (3) implement the prototype of optical system for non-uniform convolution and verify its effectiveness by incorporating it into non-uniform Richardson-Lucy algorithm [22].

2. Related Works

Non-uniform deblurring. Recently, studies on non-uniform blur are gaining momentum, and the existing methods are roughly categorized as pixel-wised or patch-wised.

Pixel-wise methods. Tai *et al*. [22, 21] extend Richardson-Lucy (RL) algorithm for non-uniform deblur, and estimate the 6-D projective motion from user interactions. Differently, Joshi *et al*. [12] measures the 6-D motion using inertial senors and compute the latent sharp image with sparse image prior. Blind estimation of 6-D blur kernel is often computationally cost, so Whyte *et al*. [23] simplify 6-D camera motion to 3-D rotations and solve it by extending the uniform algorithm by Fergus *et al*. [5]. Gupta *et al*. [7] propose the 2-D translation and 1-D rotation blur model to approximate the camera motion, and solve it with a RANSAC-based framework. However, non-uniformity requires to compute convolution in pixel-wise manner and pursue optimum blur kernel by exhaustive searching, so the above approaches all suffer from the high computational cost and this inspires our studies on optical computing.

Patch-based methods. To compute the time-consuming non-uniform convolution fast, an Efficient Filter Flow (EFF)[9] based method is proposed for acceleration. Although EFF based methods can greatly reduce the running time, it may lead to some artifacts in the cases with intensively varying blur, since the assumption of slow variance on blur kernels is violated in such cases. Our approach is largely different from and advantages over patch based approximation, since the acceleration is not obtained at the expense of accuracy.

Optical computation. Optical computation tries to perform computation with photon movement using lens, modulators, detectors and any other optic elements. Researchers make use of the high speed and parallelism of light transport in last decades and have made great progress, we refer the readers to [14, 1] for a survey of this field.

The earlier works[8, 4, 2, 11, 6, 15] in optical computing basically focused on general purpose computing, such as matrix multiplication, Fourier transformation, matrix decomposition, etc. However, with the rapid development of digital computer, the advantages of optical computing in aspect of speed are greatly weakened. However, it is still promising to design specific optical computing systems for concrete tasks, which need intensive non-general calculations without acceleration implementations. For example, O'Toole *et al*. [16] use a projector-camera system to perform light transport computing, Lefebvre*et al*. [13] and Yu *et al*. [27] apply optical computing for pattern recognition.

Non-uniform deblurring algorithms are highly time-consuming with some non-general operations. They bear the property of parallelism but cannot be implemented directly with current optical computing systems. Naturally, some elegant designs are necessary for such task specific computations, here we design and implement the system, then validate it with a series of experiments.

3. Computational Intensive Calculations in Non-uniform Deblurring

Targeting for an optical computing system for fast non-uniform deblurring, here we firstly analyze the common time-consuming calculations in existing algorithms. Many non-uniform deblurring algorithms are based on specific motion blur models, e.g. 3-D rotation model[23], 2-D translation and 1-D rotation model[7], perspective projection model(6-D)[22, 21, 12]. Based on these models, most existing algorithms optimize latent sharp image $\mathbf{L}$ and blur kernel (or motion path) $\mathbf{K}$ by minimizing following energy:

$$E(\mathbf{L},\mathbf{K}) = ||\hat{\mathbf{B}}(\mathbf{L},\mathbf{K}) - \mathbf{B}||^2 + \lambda_1 J_1(\mathbf{L}) + \lambda_2 J_2(\mathbf{K}), \tag{1}$$

where $\hat{\mathbf{B}}$ is the blurring function for generating a blurry image from latent sharp image $\mathbf{L}$ and motion $\mathbf{K}$ (in our scenario, $\mathbf{K}$ is a 3D array denoting 3D motion), $\mathbf{B}$ is the observed blurry image. For convenient to mathmatical representation, $\mathbf{L}$, $\mathbf{K}$ and $\mathbf{B}$ are denoted by column vectors in the following. $J_1(\cdot)$ and $J_2(\cdot)$ respectively regularize the estimated sharp image and kernels to reduce the ill-posedness, with λ_1, λ_2 being weighting factors.

According to the concrete form of $J_1(\cdot)$ and $J_2(\cdot)$, the objective functions can be convex or non-convex. For the former case, traditional algorithms like gradient-based method can be applied, such as steepest-descent, Conjugate Gradient(CG) or closed form solution (in frequency domain and inapplicable for non-uniform deblurring). For non-convex case, with terms favoring sparse high-frequency components in recovered images or forcing the camera motion trajectory to be sparse, the optimization is much more complicated. For non-convex optimization, the IRLS

[20] algorithm convexifies the objective into summation of weighted least square terms which can be solved by CG and is widely used for deblurring due to its effectiveness. In summary, almost all the existing non-uniform deblurring algorithms, either convex or non-convex, take gradient of energy E in each iteration. However, the non-uniformity makes this operation computationally intensive and become the efficiency bottleneck of most non-uniform algorithms.

For non-uniform deblurring, the prior terms $J_1(\cdot)$ and $J_2(\cdot)$ in Eq. 1 are still uniform, so their gradients calculation can be accelerated easily. On the contrary, the data term $||\hat{\mathbf{B}}(\mathbf{L},\mathbf{K})-\mathbf{B}||^2$ is non-uniform and its derivative manipulations $\frac{\partial E}{\partial \mathbf{L}}$ and $\frac{\partial E}{\partial \mathbf{K}}$ need to be calculated pixel by pixel, and thus is calculation demanding.

$\frac{\partial E}{\partial \mathbf{L}}$ for optimizing latent image $\mathbf{L}$. For analysis convenience, we rewrite Eq. 1 as

$$\begin{aligned} &E(\mathbf{L},\mathbf{K}) = ||\mathbf{HL}-\mathbf{B}||^2 + \lambda_1 J_1(\mathbf{L}) + \lambda_2 J_2(\mathbf{K}) \\ &\mathbf{H} = \sum_{\theta\in\mathbf{\Theta}} K^\theta \cdot \mathbf{W}^\theta, \\ &\frac{\partial E}{\partial \mathbf{L}} = 2\mathbf{H}^T\mathbf{HL} - 2\mathbf{H}^T\mathbf{B} + \lambda_1 \frac{\partial J_1}{\partial \mathbf{L}} \end{aligned} \tag{2}$$

where $\mathbf{H}$ is the blurring matrix determined by camera motion and the blur model. Specifically, $\mathbf{\Theta}$ denotes the discretized high-dimensional motion parameter space and each θ corresponds to a camera pose (described by N-tuple for a N-degree-of-freedom blur model) during exposure; K^θ denotes the weight reflecting the time elapse that camera spends at pose θ; $\mathbf{W}^\theta$ is the warping matrix mapping the reference view to the view corresponding to camera pose θ.

For computing $\frac{\partial E}{\partial \mathbf{L}}$, $\mathbf{H}^T\mathbf{HL}$ and $\mathbf{H}^T\mathbf{B}$ are the key manipulations. Here $\mathbf{H}$ is the sparse blurring matrix for non-uniform convolution.$\mathbf{HL}$ can be calculated by exactly simulating the blurring process, i.e. calculate all the views along the motion path and integrate them with corresponding weights. Similarly, $\mathbf{H}^T(\mathbf{HL})$ and $\mathbf{H}^T\mathbf{B}$ should be computed by doing the blurring procedure with inverse motion path, see [22] for details.

In addition, it is worth noting that the convolution of residuals which contains negative values need to be computed in many deblurring methods. However, the negative values cannot be modeled by physical imaging process. To address this problem, we normalize the input to make the dynamic range from 0 to 255, and remap them back after optical computing.

$\frac{\partial E}{\partial \mathbf{K}}$ for optimizing non-uniform blur kernel $\mathbf{K}$. We reform Eq. 1 with matrix representation as

$$\begin{aligned} &E(\mathbf{L},\mathbf{K}) = ||\mathbf{AK}-\mathbf{B}||^2 + \lambda_1 J_1(\mathbf{L}) + \lambda_2 J_2(\mathbf{K}) \\ &\mathbf{A} = [\mathbf{W}_1^\theta\mathbf{L} \quad \mathbf{W}_2^\theta\mathbf{L}\cdots\mathbf{W}_n^\theta\mathbf{L}] \\ &\frac{\partial E}{\partial \mathbf{K}} = 2\mathbf{A}^T(\mathbf{AK}-\mathbf{B}) + \lambda_2 \frac{\partial J_2}{\partial \mathbf{K}} \end{aligned} \tag{3}$$

Here $\theta_1 \cdots \theta_n$ is the camera poses traversing the motion parameter space and $\mathbf{W}^{\theta_i}$ is the warping matrix corresponding to pose θ_i. To minimize Eq. 3, matrix $\mathbf{A}$ needs to be calculated many times. Warping images is quite slow even with GPU acceleration. Therefore, to derive the non-uniform motion blur kernel, a series of image warping are performed which are highly time-consuming.

The above analysis tells that non-uniform convolution and warping image sequence are common calculations and efficiency bottlenecks respectively in computing $\frac{\partial E'}{\partial \mathbf{L}}$ and $\frac{\partial E'}{\partial \mathbf{K}}$, where E' is the first term of E. Fortunately, by a delicately designed optical computing system, both of them can be accelerated by a simple snapshot. Therefore, we replace these two operations with optical computing for acceleration, as blocked in Alg. 1.

Algorithm 1. A simple non-uniform deblurring algorithm incorporating our optical computing system

---

Input: Blurred image **B**
Output: latent sharp image **L** and camera motion **K**

Initialization: $\mathbf{L}_0 = \mathbf{B}$ and $t = 0$;
Repeat
  Predict sharp image by bilateral and shock filters
  Repeat following step **until** converge:
    Project $\mathbf{L}_t$, move the camera along $\mathbf{K}_t$ to predicte blurred image $\hat{\mathbf{B}}_t = \hat{\mathbf{B}}(\mathbf{B}_t,\mathbf{K}) = \mathbf{A}_t\mathbf{K}_t$;
    Compute residual: $\mathbf{e}_t = \mathbf{B}_t - \hat{\mathbf{B}}_t$;
    Project $\mathbf{e}_t$, move camera traversing motion parameter space, compute $\frac{\partial E'}{\partial K^\theta} = (W^\theta\mathbf{L})_t^T\mathbf{e}_t$ for each θ;
    Update motion kernel: $\mathbf{K}_{t+1} = \mathbf{K}_t + \partial E'/\partial \mathbf{K}_t$;
    Set $K_{t+1}^\theta = 0$ if $K_{t+1}^\theta < 0$, and normalize $K_{t+1}^\theta = \frac{K_{t+1}^\theta}{\sum_\theta K_{t+1}^\theta}$
  Repeat following step **until** converge:
    Project $\mathbf{L}_t$, move the camera along $\mathbf{K}_t$ to predicte blurred image $\hat{\mathbf{B}}_t = \hat{\mathbf{B}}(\mathbf{B}_t,\mathbf{K})$;
    Compute residual: $\mathbf{e}_t = \mathbf{B}_t - \hat{\mathbf{B}}_t$;
    Project $\mathbf{e}_t$, move the camera along $\mathbf{K}_t^{-1}$ to capture correction image $\frac{\partial E'}{\partial \mathbf{L}} = \hat{\mathbf{B}}(\mathbf{e}_t,\mathbf{K}_t^{-1})$;
    Update latent sharp image: $\mathbf{L}_{t+1} = \mathbf{L}_t + \partial E'/\partial \mathbf{L}_t$;
  $t = t+1$;
Until $||\hat{\mathbf{B}}_t - \mathbf{B}||^2 < thr$.

---

4. Optical Computing System

We design an optical computing system for accelerating the above two computationally intensive calculations: a motion platform based projector-camera module system.

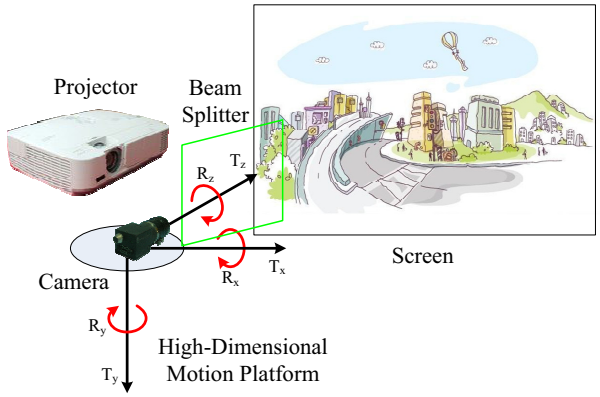


Figure 2. **Diagram of our high-dimensional motion platform based projector-camera system.**

4.1. High-Dimensional Motion Platform Based Projector-Camera System

Different from the common used projector-camera system, our camera is mounted on a motion platform. Thus the camera is able to move along a given trajectory or fixed at a certain pose following users' requirement. The diagram of the basic system is shown in Fig. 2(a).

Assuming the light transport function from projector to screen be $\mathcal{T}_p(\cdot)$, an input image $\mathbf{L}$ is projected onto the screen to form $\mathcal{T}_p(\mathbf{L})$. Similarly, assuming the light transport function from screen to the camera is $\mathcal{T}_c(\cdot)$, we obtain an output image from the camera as $\mathcal{T}_c(\mathcal{T}_p(\mathbf{L}))$. As the platform moves, the light transport function of camera $\mathcal{T}_c(\cdot)$ changes correspondingly, and generate the captured image $\mathcal{T}_c^{\theta}(\mathcal{T}_p(\mathbf{L}))$ at camera pose θ. Under assumption of a pin hole camera model, $\mathcal{T}_p(\cdot)$ and $\mathcal{T}_c^{\theta}(\cdot)$ are both linear transformations which can be represented by transport matrices. For convenience, we denote them as $\mathcal{T}_p$ and $\mathcal{T}_c^{\theta}$ respectively.

If the motion platform moves during exposure, the resulting captured image can be represented by

$$\mathbf{B} = \int_{t\in\tau} \mathcal{T}_c^{\theta_t}\mathcal{T}_p\mathbf{L}dt \tag{4}$$

where $\mathbf{B}$ is the captured blurry image, θ_t is the camera position at time t and $\{\theta_t : t \in \tau\}$ compose the whole camera motion trajectory, with τ being the exposure time range.

To facilitate the analysis, we define an origin position of the motion platform as θ_0, and then the view $\mathcal{T}_c^{\theta_t}\mathcal{T}_p\mathbf{L}$ captured at position θ_t can be decomposed as $\mathcal{T}_c^{\Delta\theta_t}(\mathcal{T}_c^{\theta_0}\mathcal{T}_p\mathbf{L})$ with $\Delta\theta_t = \theta_t - \theta_0$, and $\mathcal{T}_c^{\Delta\theta}$ is just the warping matrix $\mathbf{W}^{\theta}$ in Eq. 2. In other words, the latent sharp version of the captured blurry image is the view captured at the origin position, and the blurry image is the integration of a sequence of views captured along a relative motion trajectory $\{\Delta\theta_t\}$. Mathematically, if the whole transport matrix of our optical system at origin position $\mathcal{T}_c^{\theta_0}\mathcal{T}_p$ is an identity matrix, i.e. the image captured at origin position θ_0 is exactly the same as the input image of projector, our high-dimensional motion platform projector-camera system can simulate the physical blurring process by directly moving the camera along the relative motion path $\{\Delta\theta_t\}$.

Theoretically, $\mathcal{T}_c^{\theta_0}\mathcal{T}_p = I$ indicates that the projector-camera should be exactly the inversion of each other in optics, and optical computing systems need precise calibration to ensure the accuracy. However, it is often hard to find the mutually inverse projector/camera pair practically due to several reasons: (i) most commonly used projectors and cameras are of totally different optical parameters; (ii) the precise alignment between optical centers of camera and projector lenses is difficult; (iii) considering the non-linear transformations(e.g. gamma correction), the transport matrix based linear model may not be accurate enough in the real cases. Therefore, in implementation we add a calibration process—both geometric and photometric—to cope with the errors beyond the adopted linear model. Leaving the calibration process details in the latter experiment section, we first give the optical computation process of time-consuming module using our calibrated system.

Calculating warping image sequence. The compuataiton of warping image sequcence is quite simple and straightforward for our optical computing system. Practically, for a specific view pose θ_i, the snapshot by projecting $\mathbf{L}$ and capturing with pose θ_i is exactly the desired warping image $\mathbf{W}^{\theta_i}\mathbf{L}$.

Calculating spatially varying convolution. For our optical computing system, the blurring manipulation caused by camera shake can also be computed simply by reproducing the real procedure of generating a blurry image. For a given high-dimensional camera trajectory, we can capture the desired blurry image exactly by driving the motion platform along the given trajectory in a properly set exposure time with velocity correponding to the elapse of poses along the path.

Practically, the camera trajectory is not often available, and the non-uniform blur kernel is commonly represented by a density function of a high-dimensional cube. However, due to the physical constraint, the non-zero element should be continuous and sparse. Therefore, it can be represented by a high-dimensional connected-graph, and each traversal path of this connected-graph can be used as the camera trajectory here.

5. Implementation and Experiment Results

This section demonstrates our prototype optical computing system, and then shows its performance in performing predefined non-uniform blurring and restoring blurry images after incorporated into a deconvolution framework.

The whole system is shown in Fig. 3. For simplicity, we use a 3-axis rotating platform and adopt 3-D rotational blur

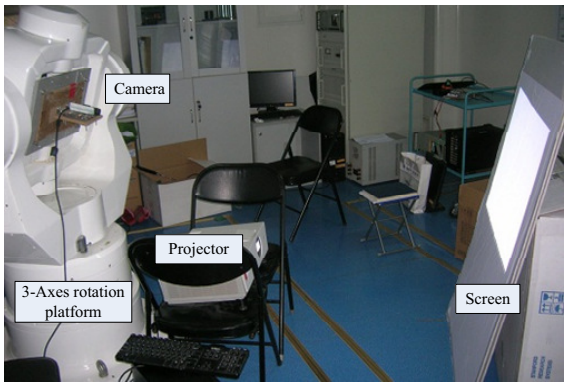


Figure 3. **The prototype of proposed high-dimensional motion platform based projector-camera system.**

model correspondingly. This rotating platform is able to burden about 40kg load (camera in our scenario) and moves to an orientation with any pitch, yaw and roll angle. The maximum moving velocity is $200°/s$ and of rotating acceleration is $200°/s^2$, which are sufficient for our prototype system to validate the proposed approach. The camera mounted on the platform is a Point Grey FL2-08S2C. Both the platform and camera are driven by a control software through RS-232 COM and IEEE 1394 respectively. We adopt software synchronization to ensure exact matching between the camera's exposure time and the given platform's motion trajectory.

5.1. System calibration

The hybrid system should be calibrated in terms of geometric distortion, intensity response curve and other distortions. Ideally, we should calibrate the projector and camera independently. For simplicity and efficiency, we treat the projector-camera system as a black box and the experiments show that our simple calibration method is accurate enough to prevent the computing process suffering from distortions.

Geometric calibration. Since the projector and camera cannot be exactly complementary (with same optical path), we need a careful calibration to ensure exact correspondence between the input image and captured image. One straightforward method is to calibrate all the parameters of the projector-camera system with a series of calibration patterns. However, this method is complex and may suffer from the unmodeled distortions. Fortunately, we only need the geometric correspondence between the input and output image, and can just use a coordinate map to represent the correspondence instead of an explicit geometric model. In implementation, we provide a chessboard pattern to the projector and capture the projected result with the camera. By corners matching, the correspondence between input and output images is determined. Since the coordinate correspondence vary smoothly, we can largely reduce computation time by interpolating corresponding map from several corners points. Fig. 4 gives an example of our geometric calibration, with Fig. 4(a) being the original calibration pattern, Fig. 4(b) being the captured version by our projector-camera system with the motion platform fixed at the origin position, and Fig. 4(c)(d) respectively presenting the mapping vectors at several landmarks for interpolation and the calibrated pattern.

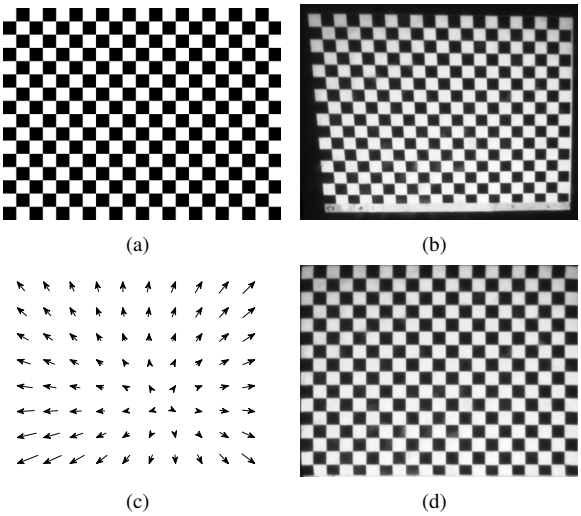


Figure 4. **Result of geometric calibration.** (a) original chessboard pattern. (b) captured pattern. (c) warping vectors from landmarks for interpolation. (d) pattern after geometric calibration.

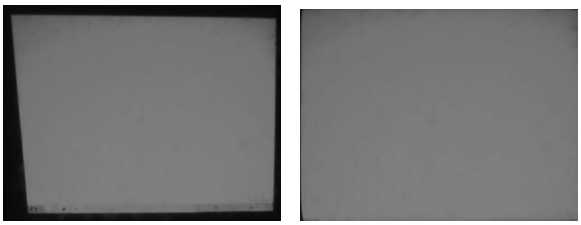


Figure 5. **Result of dark corner correction.** (a) and (b) are respectively the ratio image before and after calibration.

Dark corner correction. As known, both projector and camera suffer from the dark corner effects and the effect is more significant in such hybrid systems. To offset the dark corner, we project an constant-intensity grey image (intensity is set 128 to prevent saturation) to the screen, and compute the ratio image between the geometrically calibrated output image and the original input one. Fig. 5 (a) and (b) respectively show the captured grey image and the geometrically calibrated ratio image, which can be used to remove the dark corner effect.

Intensity response calibration. Because the response curve of neither projector nor camera is ideally linear due

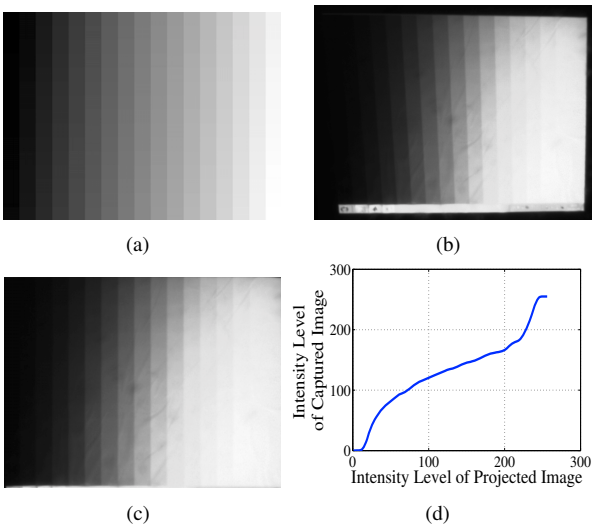


Figure 6. **Intensity response calibration.** (a) original intensity step pattern. (b) captured pattern. (c) pattern after geometric calibration and dark corner correction. (d) intensity response curve.

to some internal transformations like gamma correction, the intensity response of the whole system varies nonlinearly with intensity and need to be calibrated.

There are several contributing factors (e.g. response curve of projector, reflection properties of screen, response curve of camera etc.) for the nonlinearity, we prefer a black-box model here. We first project an intensity gradation map onto the screen, correct the geometric distortion and dark corner effect of the output image, and then compute the response of each intensity level by averaging all the pixels at this level (remove the marginal ones). Fig. 6 gives an example of intensity response calibration, (a) gives the original intensity step pattern, (b) shows the captured image and (c) is the calibrated version of (b) with geometric and dark corner correction, (d) demonstrates the calibrated intensity response curve derived by our intensity step map.

In addition, considering that the color channels of common RGB camera and projector are coupled with each other and decoupling them will increase the complexity of our system, we just calibrate grey scale images in our prototype system and process each channel separately.

5.2. Experiment results on prototype validation

Accuracy of PSF. To verify the accuracy of our high-dimensional motion platform based projector-camera system, we project a point grid onto the screen and capture the deterioration result (i.e., the PSF) by a randomly generated camera shake, by setting the camera exposure synchronized with the motion procedure.

Comparing the synthetic blurred result (with camera intrinsics and trajectory known) in Fig. 7(a) and the image captured by our high-dimensional motion platform projector-camera system in Fig. 7 (b), we can see extremely high similarity. Apparently, our system gives promising approximation and thus is of sufficient accuracy for performing the blurring manipulation optically.

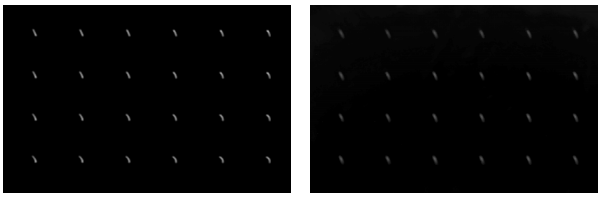


Figure 7. **Testing on PSF accuracy using a point grid pattern.** (a) synthetic result according to projective geometry. (b) blurred point grid pattern by our prototype optical computing system.

Accuracy of non-uniform blur simulation. A sharp image is projected to the screen, as shown in Fig. 9(a), with whose ground truth blurry version under a given camera shake shown in Fig. 9(b). Then the blurry image optically computed by our system are captured under the given motion trajectory, as shown in Fig. 9(d). For clearer comparison, we display in Fig. 9(c) the the absolute residual map between (b) and (d). The small residue validates the accuracy of our optical computing system. The residual errors are mainly caused by two factors: the dead/saturation area of response curve (shown in Fig. 6(d)) and the distortion not modeled in synthesizing ground truth blurry image, such as radial distortion, tangential distortion etc.

Fast non-blind deblurring with our system. We incorporate our system into the framework of non-blind non-uniform motion deblurring for acceleration. For simplicity, we adopt non-uniform version of RL[17, 22] deblurring algorithm in this experiment. Introducing our high-dimensional motion platform projector-camera system, we can replace the pixel-wise non-uniform convolution in each iteration with two snapshots, and thus the running time can be largely reduced. Neglecting the mechanical limitation of the system, it only takes a little longer than 1/15 second for one iteration with using a 30fps digital camera. In comparison, the pixel-based methods are order of magnitude slower than our system even with GPU acceleration, and patch-based methods implemented on GPU are also much slower than ours, especially in case of large image size.

Fig. 10(a)(e) show the blurry image and true sharp image respectively. The estimated sharp images and residual maps at iteration 1, 10, 20 are shown sequentially in Fig. 10(b-d) and (f-h). The increasing sharpness and decreasing residue both validate that our system can be incorporated into deblurring framework easily to raise the efficiency without introducing large artifacts.

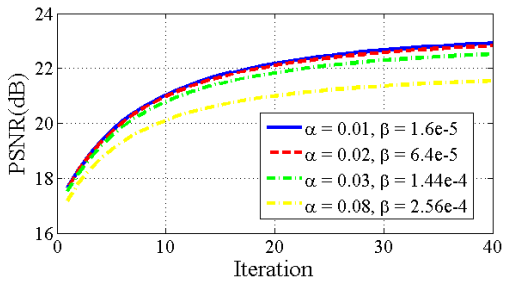


Figure 8. **Effect of noise.** We synthetically added noise to computed blurred images to simulate noise contaminated optical computing process and gives the PSNR of the result after 40 iterations.

6. Analysis and Discussions

The paper demonstrates an optical computing system which significantly accelerates the conventional non-uniform motion deblurring. The framework is fast, of high accuracy and also with flexible extensions worth studying in the future.

Effect of noise. The proposed optical computing system will introduce noise inevitably. To test the influences of imaging noise during the blur computing process, we add some shot noise, which follows Poisson-Gaussian distribution with variation $\sigma^2 = \alpha\mathbf{I} + \beta$, to image $\mathbf{I}$ in the corresponding step of RL algorithm and test the final deblurring performance. Fig. 8 shows that RL algorithm converges consistently without increased iterations at higher noise levels, and the PSNR of deblurred images slightly decreases as the noise level increases. Therefore, we recommend low ISO settings of the camera to reduce the noise effect. In addition, the prior constraints on the latent sharp image, which are widely used in deblurring algorithm are also helpful to suppress the noise.

Benefits and limitations. Benefited from the fast light transport and the parallelism of imaging system, optical computing systems can do specific operations very fast. Specifically to our system, each CCD unit acts as an individual computing unit and each snapshot can achieve parallel processing of Mega- or even Giga- pixels, thus the proposed optical computing system provides a feasible and promising solution to fast non-uniform motion delburring.

So far our prototype is mainly limited in three aspects: high cost of the motion platform, limited frame rate of camera and assumption of depth independence to the blur patterns. (i) In our implementation, we adopt a high end rotating platform with large angular velocity, accelerations, payload and very high precision (0.001°), we can also use a high end 6-D platform for arbitrary motion. However, both above platforms are too costly for consumer imaging systems. Considering the small weight and size of consumer cameras, the moment of inertia of the camera used in our system can be much smaller than the platform limitation. At a rough estimation, around 1/40 the upper bound of the adopted platform is sufficient. Therefore,we can choose a low load motion platform, so that the motion velocity can be improved to shorten the running time further. (ii) As for camera's frame rate, in our experiment the Point Grey Flea2 08S2C camera can achieve 30fps with resolution of 1024×768 pixels, so that our system can finish at most 30 times of blurring manipulations or frame-wise inner product manipulations within 1s. The computation can be further accelerated by using a higher frame-rate camera and projector. (iii) With a planar screen, our system cannot simulate depth dependent blurring, hence depth specific decomposition and more snapshots are necessary in such cases.

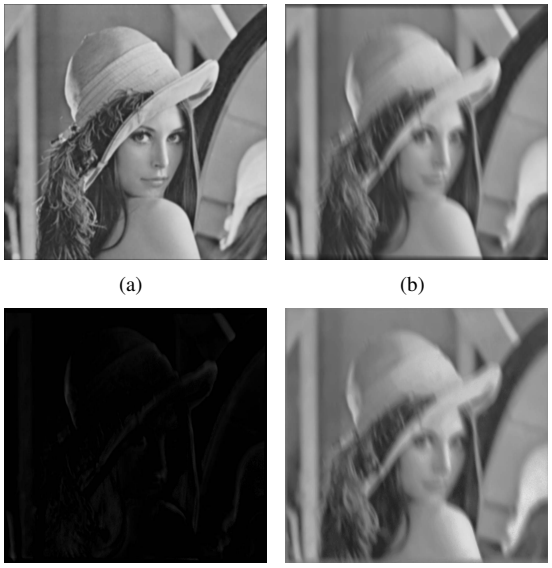


Figure 9. **Result of spatially varying blurring.** (a) a sharp image. (b) synthetic blurry image from projective geometry. (d) result of our optical computing system (c) residual between (b) and (d).

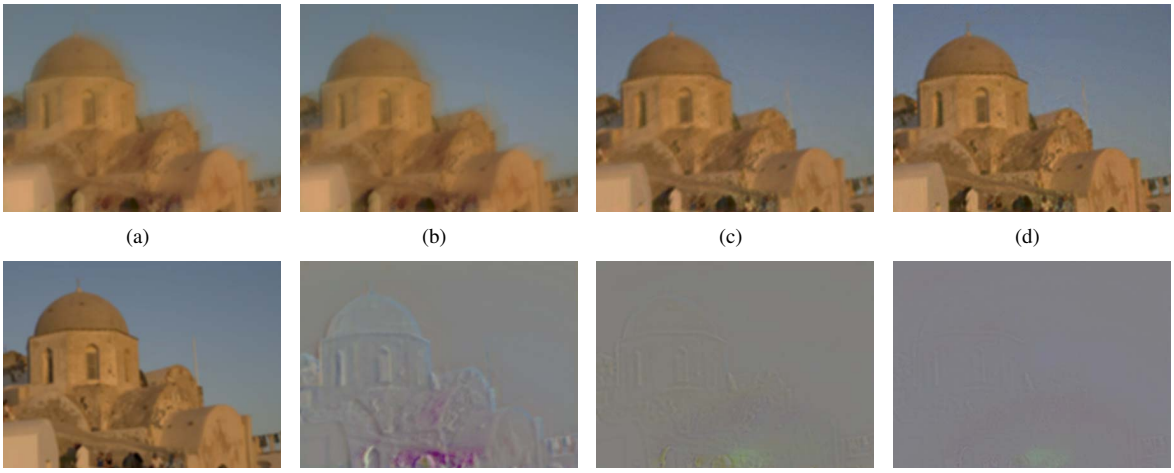
Promising extensions. The optical computation framework can be applied not only to camera shake removal, but also the out-of-focus blur. Instead of using a motion platform, a camera with programmable focal length can simulate the defocus blur at any focal settings.

Acknowledgement

This work was supported by the Project of NSFC No. 61120106003, NO. 60932007 and No. 61171119. The authors are indebted the anonymous reviewers, as well as Gordon Wetzstein for their constructive suggestions.

References

[1] P. Ambs. A short history of optical computing: rise, decline, and evolution. In *International Conference on Correlation Optics*, 2009.



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

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

Figure 10. **Result of fast non-blind deblurring with our optical computing system.** (a) blurry image (e) sharp image. (b)(c)(d) estimated sharp image at 1st, 10th and 20th iteration. (f)(g)(h) the residual error map of (b)(c)(d) with respect to (a).

[2] R. A. Athale and W. C. Collins. Optical matrix-matrix multiplier based on outer product decomposition. *Appl. Opt.*, 21(12):2089–2090, Jun 1982.

[3] S. Cho and S. Lee. Fast motion deblurring. In *ACM SIGGRAPH Asia*, volume 28, 2009.

[4] A. Dias. Incoherent optical matrix-matrix multiplier. *NASA. Langley Research Center Opt. Inform. Process. for Aerospace Appl.*, pages 71–83, 1981.

[5] R. Fergus, B. Singh, A. Hertzmann, S. Roweis, and W. Freeman. Removing camera shake from a single photograph. In *ACM SIGGRAPH*, volume 25, Jul. 2006.

[6] P. Guilfoyle and R. Stone. Digital optical computer ii. In *Optical Enhancements to Computing Technology*. International Society for Optics and Photonics, 1991.

[7] A. Gupta, N. Joshi, C. L. Zitnick, M. F. Cohen, and B. Curless. Single image deblurring using motion density functions. In *ECCV*, 2010.

[8] R. A. Heinz, J. O. Artman, and S. H. Lee. Matrix multiplication by optical methods. *Appl. Opt.*, 9(9):2161–2168, Sep 1970.

[9] M. Hirsch, S. Sra, B. Scholkopf, and S. Harmeling. Efficient filter flow for space-variant multiframe blind deconvolution. In *CVPR*, pages 607–614, 2010.

[10] Z. Hu and M.-H. Yang. Fast non-uniform deblurring using constrained camera pose subspace. In *BMVC*, 2012.

[11] H. Huang, L. Liu, and Z. Wang. Parallel multiple matrix multiplication using an orthogonal shadow-casting and imaging system. *Opt. Lett.*, 15(19):1085–1087, Oct 1990.

[12] N. Joshi, S. Kang, C. Zitnick, and R. Szeliski. Image deblurring using inertial measurement sensors. In *ACM SIGGRAPH*, volume 29, 2010.

[13] D. Lefebvre, H. Arsenault, and S. Roy. Nonlinear filter for pattern recognition invariant to illumination and to out-of-plane rotations. *Applied optics*, 42(23):4658–4662, 2003.

[14] E. Leith. The evolution of information optics. *IEEE Journal of Selected Topics in Quantum Electronics*, 6(6):1297–1304, 2000.

[15] A. Lewis, Y. Albeck, Z. Lange, J. Benchowski, and G. Weizman. Optical computation with negative light intensity with a plastic bacteriorhodopsin film. *Science*, 275(5305):1462–1464, 1997.

[16] M. O'Toole and K. Kutulakos. Optical computing for fast light transport analysis. In *ACM SIGGRAPH*, volume 29, 2010.

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

[18] Q. Shan, J. Jia, and A. Agarwala. High-quality motion deblurring from a single image. In *ACM SIGGRAPH*, volume 27, 2008.

[19] M. Sorel and J. Flusser. Space-variant restoration of images degraded by camera motion blur. *IEEE Transactions on Image Processing*, 17(2):105–116, Feb. 2008.

[20] C. Stewart. Robust parameter estimation in computer vision. *Siam Review*, 41(3):513–537, 1999.

[21] Y. Tai, N. Kong, S. Lin, and S. Shin. Coded exposure imaging for projective motion deblurring. In *CVPR*, pages 2408–2415, Jun. 2010.

[22] Y. Tai, P. Tan, and M. Brown. Richardson-Lucy deblurring for scenes under projective motion path. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(99):1603–1618, Aug. 2011.

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

[24] O. Whyte, J. Sivic, A. Zisserman, and J. Ponce. Non-uniform deblurring for shaken images. *International Journal of Computer Vision*, 98(2):168–186, 2012.

[25] L. Xu and J. Jia. Two-Phase Kernel Estimation for Robust Motion Deblurring. In *ECCV*, 2010.

[26] L. Xu and J. Jia. Depth-aware motion deblurring. In *ICCP*, 2012.

[27] F. Yu, S. Jutamulia, T. Lin, and D. Gregory. Adaptive real-time pattern recognition using a liquid crystal TV based joint transform correlator. *Appl. Opt.*, 26:1370–1372, 1987.