MOTION BLUR KERNEL ESTIMATION USING NOISY INERTIAL DATA

Ruiwen Zhen and Robert L. Stevenson

Department of Electrical Engineering, University of Notre Dame

ABSTRACT

In the case of motion blur due to unknown motion, most of the existing image deblurring algorithms rely on good initial estimate of the kernel or latent image obtained through blind deconvolution and only consider 3-dimensional camera motions. To overcome these problems, Joshi [1] presented a novel blur kernel estimation and image deblurring approach by integrating 6-dimensional inertial sensors with a camera. However, the drift in the estimated camera motion path introduced by inertial measurement noise was not well handled in Joshi's work. In this paper, we propose an alternating optimization scheme to move the drifted camera motion path to the correct position. The camera pose space in the projective motion blur model is replaced by a motion path set to compensate for path drift. Experiments are performed on synthetic and real images to show the effectiveness of our approach.

Index Terms— Blur Kernel, Motion Path, Accelerometer, Deblur, Noise

1. INTRODUCTION

Motion blur is an artifact in photography usually caused by the relative motion between a camera and an imaged scene during exposure. This process is typically modeled as the convolution of a latent sharp image with a blur kernel plus noise. A common assumption usually embedded in this model is that the motion blur kernel is spatially invariant. However, this assumption does not hold in general due to camera rotation, scene depth variation and other reasons [1].

Recently, a new geometric model, the projective motion blur model [1], has been proposed to represent the degraded image as an integration of the clear scene under a sequence of projective motions during the exposure time [2, 3, 4]. In discrete form, the blurry image g is the weighted sum of homographically transformed copies of the unblurred image f plus Gaussian noise n,

$$g = \sum_{i=1}^{M} w_i f(H_i x) + n, \tag{1}$$

where g, f, n are in column-vector form and x is a 3×1 vector indicating the homogeneous pixel coordinate. The intermediate transformed copy $f(H_i x)$ can be viewed as the image seen by the camera at one pose characterized by homography H_i and its corresponding weight w_i is the proportion of exposure time spent by the camera at that pose. M denotes the number of all possible poses. This model is able to describe spatially varying motion blurs by constraining the global camera motion path rather than directly recovering different blur kernels at different pixel locations.

Whyte et al.[3] substituted this model into existing blind deblurring algorithms to recover 3D camera rotations. Concurrently, Gupta et al.[4] described similar work to explain a different set of 3D camera motions. Both of them sampled poses in a 3D camera motion space and estimated the weights $\{w_i\}$ at sampled poses. Hu et al.[5] used backprojection to narrow down the camera pose space and adopted image edges instead of latent image as the intermediate result to speed up optimization. The work mentioned above relied on the accuracy of the initial blur kernel estimate and did not attempt to recover the motion path. In contrast, Tai et al.[2] assumed that the camera spent equal exposure time at each pose over the motion path whose parameters were regarded as known. In this case, a motion path could be obtained. Zhang et al.[6] still adopted Tai's assumption, but tried to blindly estimate the 4D parameters of all the homographies with the constraint that camera should move at a constant speed.

To recover the motion path without the above assumptions, Joshi et al.[1] combined the camera with inexpensive gyroscope and accelerometer to estimate the 6D parameters of homographies $\{H_i\}$. However, noisy inertial measurements lead to drift in the estimated motion path due to integration, especially in x and y translations (assume z is parallel to the optical axis). Joshi proposed to search the end point of the motion path within a 1mm radius of the initially computed end point and pick the optimal end point location that resulted in a deconvolved image with the highest likelihood. However, when the exposure time is longer or the accelerometer generates larger noise, it is difficult to apply Joshi's method.

In this paper, we propose an approach to estimate the camera motion path using noisy inertial measurements. A motion path set is used instead of the camera pose space in equation (1). Then an efficient optimization scheme that alternates between kernel weight estimation and latent sharp image restoration is adopted. At the beginning of each iteration, we add the image prediction step, same as [7], for progres-

[1] This geometric model has different names in [2, 3, 4], and the name in [2] is used in this paper.

sively refining the blur kernel. The proposed approach is evaluated on both synthetic and real images.

2. APPRAOCH

2.1. Blur Model

In equation (1), the homography H_i is usually given as [4]:

$$H_i = I\left(R_i + \frac{1}{d}T_i\begin{bmatrix}0 & 0 & 1\end{bmatrix}\right)I^{-1}, \tag{2}$$

where R_i and T_i are the rotation matrix and translation vector at camera pose indexed by i, I is the camera intrinsic matrix and d is the scene depth assumed constant. Since the noise of x- and y-axis accelerometer data (XY-plane is aligned with the camera's sensor's coordinate frame) generate more drift than that of other inertial measurements [1], we only consider estimating the blur kernel caused by camera in-plane translation based on measured accelerations (The relatively accurate inertial measurements can be also added without changing the main approach). In this case, R_i becomes an identity matrix and $T_i = \begin{bmatrix}T_i(x) & T_i(y) & 0\end{bmatrix}^T$, so H_i can be simplified to

$$H_i = I\left(\begin{bmatrix}1 & 0 & T_i(x)/d\\ 0 & 1 & T_i(y)/d\\ 0 & 0 & 1\end{bmatrix}\right)I^{-1}, \tag{3}$$

where $T_i(x)$ and $T_i(y)$ are x and y components of T_i. In order to make the optimization problem solvable, a sparse matrix K_i is derived from homography H_i using bilinear interpolation [3] and each row of K_i is the blur kernel placed at each pixel location, then

$$g = \sum_{i=1}^{M} w_i K_i f + n = Cf + n. \tag{4}$$

With measured accelerometer data, K_i and w_i correspond to the acceleration sample and sample interval respectively (similar to [2]). The linear kernel matrix C sums up a sequence of weighted K_i, representing a camera motion path.

However, using the kernel matrix C calculated from noisy measurements may influence the deblurred result. To recover the correct position of camera motion path, this work represents the observed blurry image as the weighted sum of images blurred by different camera motion paths instead of camera poses, i.e.,

$$g = \sum_{j=1}^{N} W_j C_j f + n = \sum_{j=1}^{N} W_j \left(\sum_{i=1}^{M} w_{ji} K_{ji}\right) f + n, \tag{5}$$

where C_j is a kernel matrix corresponding to a motion path P_j, W_j is the weight for C_j and N is the number of paths. Then the new kernel matrix becomes

$$C = \sum_{j=1}^{N} W_j C_j \tag{6}$$

and a new motion path could be built as:

$$P = \sum_{j=1}^{N} W_j P_j. \tag{7}$$

P_j and P can be viewed as a sequence of translation vectors. Though C and P are not one-to-one, both adapt in the same direction during the optimization process to approach the "ground-truth", which can better account for the blurry observation and the inertial measurements.

2.2. Image Prediction

At the beginning of each iteration, we compute gradient maps of the estimated latent image $\hat{f}$ along x and y directions as in [7]. This prediction step helps prevent the deconvolution artifacts produced in the image estimation step from hindering kernel estimation. As iteration proceeds, the parameters for filtering and thresholding are gradually changed by multiplying a factor α_1 to expose more image details and refine the motion blur kernel.

2.3. Kernel Estimation

In this step, the motion path set $\{P_j\}$ and kernel set $\{C_j\}$ are built and the corresponding weights $\{W_j\}$ are estimated based on image gradient maps computed in the prediction step. Initially the motion path set $\{P_j\}$ is derived from measured accelerations. With the sampled noisy accelerometer data $a_1(m), a_2(m), \ldots, a_M(m), m = x, y$, the camera translations $T_1(m), T_2(m), \ldots, T_M(m)$ can be calculated through twice rectangle integration. The measurement noise is usually modeled as white noise with standard deviation $\sigma_d(m)$. Then the corresponding translation drift also follows a Gaussian distribution with deviation [8]:

$$dev(T_i(m)) = \sqrt{\frac{\sigma_d(m)^2}{6f_s^4}(i-1)i(2i-1)}, \tag{8}$$

where f_s is the accelerometer sampling frequency and i is the pose index. In theory, the end point of the correct path should lie in the square $[T_M(m) - dev(T_M(m)), T_M(m) + dev(T_M(m))], m = x, y$ with a large confidence. Therefore, an end point set $\{(U_j(x), U_j(y)), j = 1, 2, \ldots, N\}$ is built within the predicted square by choosing proper length sampling unit and angle sampling unit (as shown in Fig.1). Each end point corresponds to a motion path P_j with accelerations matching the measurements in the least-squares sense, i.e.,

$$\begin{aligned} P_j &= \{\hat{T}_i(m), i = 1, 2, \ldots, M, m = x, y\} \\ &= \underset{\{\hat{T}_i(m)\}}{\operatorname{argmin}} \sum_{m=x,y}\sum_{i=1}^{M}\left(\frac{d^2\hat{T}_i(m)}{dt^2} - a_i(m)\right)^2 \\ &\quad + \beta \sum_{m=x,y} (\hat{T}_M(m) - U_j(m))^2, \end{aligned} \tag{9}$$

where $\hat{T}_0(m) = 0, m = x, y$ and β is a regularization coefficient. This multivariable optimization problem can be solved by a Quasi-Newton method. In the following iterations, $\{a_i(m)\}$ and $\{T_i(m)\}$ come from the estimated path P of the previous iteration and the end point deviation is gradually decreased by multiplying α_2.

With the built path set $\{P_j\}$ and corresponding kernel set $\{C_j\}$ (equation (4)), the motion blur kernel C is estimated by optimizing the weight vector $W = \begin{bmatrix} W_1 & W_2 & \dots & W_N \end{bmatrix}^T$

$$\begin{aligned} &\underset{W}{\operatorname{argmin}} \sum_{\partial_*} \Omega_* \| \sum_{j=1}^{N} W_j C_j \partial_* \hat{f} - \partial_* g \|^2 + \gamma \|W\|^2 \\ &\text{subject to } W \geq \mathbf{0} \text{ and } \sum_{j=1}^{N} W_j = 1, \end{aligned} \tag{10}$$

where $\partial_* \in \{\partial_x, \partial_y, \partial_{xx}, \partial_{xy}, \partial_{yy}\}$ denotes the partial derivative operator in two directions of first and second orders, $\Omega_* \in \{\Omega_1, \Omega_2\}$ is the weight for partial derivatives of different orders [9], and γ is the parameter for weight sparse prior term. Since the motion path set is adopted instead of the camera pose space, no path prior or smoothness prior is needed. The sparse prior is added so that only the paths close to the "ground-truth" path get positive weights. This optimization problem is solved by an interior point method and C, P in the current iteration can be obtained through equations (6), (7).

2.4. Image Estimation

Since the blur kernel matrix C has been estimated, the image deconvolution problem is formulated as a maximum a posteriori (MAP) estimation with sparse image prior [10],

$$\hat{f} = \underset{f}{\operatorname{argmin}} \|g - Cf\|^2 + \lambda \sum_k \|G_k f\|^{0.8}, \tag{11}$$

where $\{G_k\}$ are first and second derivative filters. The minimization is performed using iterative re-weighted least square (IRLS) algorithm.

2.5. Analysis

The proposed approach is summarized in Algorithm 1. In the kernel estimation step, the kernels which are closer to the correct kernel in path sense get larger weights than others. Therefore the latent image after image estimation step will have less deconvolution artifacts introduced by the wrong kernel and have more correctly deblurred image details. In the next iteration, more useful image information can be used for refining the kernel. In addition, the paths that are closer to the "ground-truth" path also get larger weights, so the new path presented in equation (7) moves towards the "ground-truth", which improves the correctness of the motion path set built in the next iteration. If another choice, picking the path with the highest weight as the new path, is used instead, the result may diverge since some little change in the estimated weights will change the path moving direction. At the end of the optimization, both the path set and the corresponding kernel set almost contract to one and thus C and P are finally one-to-one.

Algorithm 1 Blur Kernel Estimation Using Inertial Data

procedure KERNEL_ESTIMATE(g, $\{a_i\}$)
  Let $\hat{f} = g$
  for iter $\leftarrow$ 1, N_iters **do**
    Compute gradient maps of $\hat{f}$
    $\{a_i\} \rightarrow \{T_i\}$, build $\{P_j\}$, $\{C_j\}$ using Eq.(9), (4)
    Update W using Eq.(10) and gradient maps
    Generate C using Eq.(6), update P, $\{a_i\}$
    Estimate latent image $\hat{f}$ using Eq.(11) and C
  end for
end procedure

3. EXPERIMENT RESULTS

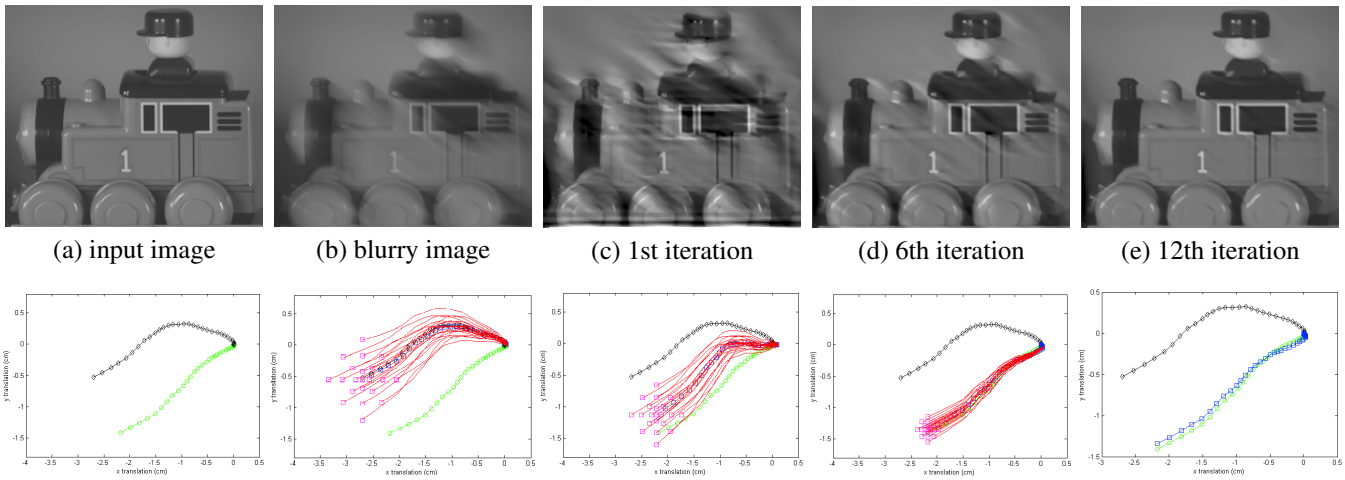
This section provides experiment results on synthetic data and real data. The imaging system used for capturing the real data has been introduced in [11]. In experiments, the multiplying factors α_1 and α_2 are set to be 0.85 and 0.9 respectively and the regularization parameters β, γ and λ are chosen as 1000, 10 and 0.001. Measurement noise deviations $\sigma_d(x)$, $\sigma_d(y)$ are estimated to be about $0.015m/s^2$ and $0.020m/s^2$.

In the synthetic example, real accelerometer data is used as ground-truth to generate the gray-scale blurry observation (Fig.1(b)). To simulate the measurement noise, we add white noise with measured deviations $\sigma_d(m)$ to the ground-truth accelerations. The initial support size and spatial standard deviation of the bilateral filter in the prediction step are set to 20 and 50 respectively. From Fig.1(g)-(i), it can be seen that the estimated path (blue) gradually approaches the ground-truth path (green). After 12 iterations, the latent image is well recovered (Fig.1(e)) and the paths in the motion path set almost contract to one path (Fig.1(i)).

In the real example, the blurry image (Fig.2(a)) and the noisy accelerations (Fig.2(e)) are captured at the same time by our imaging system. The support size and spatial standard deviation are differently selected to be 10 and 20 due to less deconvolution artifacts. The recovered motion path (blue) and latent image are shown respectively in Fig.2(h) and Fig.2(d).

4. CONCLUSION

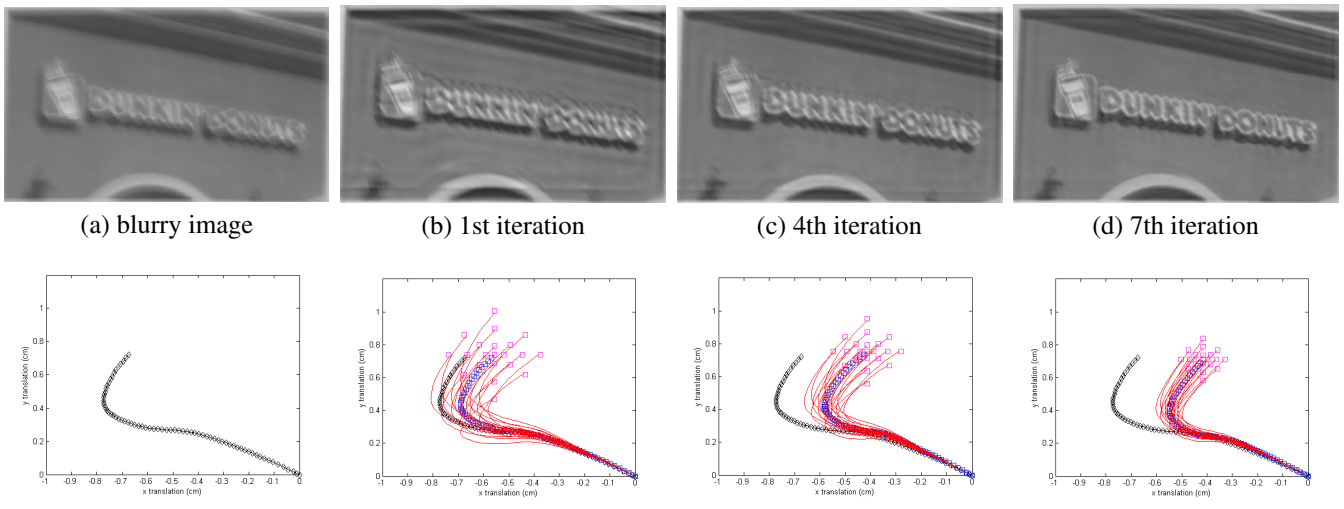
In this paper, we propose to replace the camera poses of previous approaches with a motion path set in the projective motion blur model to estimate the final motion path using an alternating optimization approach. Experiment results show that our approach can compensate the path drift brought by measurement noise and produce favorable deblurred images.



(a) input image (b) blurry image (c) 1st iteration (d) 6th iteration (e) 12th iteration

(f) simulated noisy curve (g) 1st iteration (h) 6th iteration (i) 12th iteration (j) final curve close-up

Fig. 1. Experiment results with simulated image and curve. (a) input image; (b) blurry image; (c)-(e) estimated latent image after 1st, 6th, 12th iterations; (f) ground-truth curve (green) and simulated noisy curve (black) unit:cm; (g)-(i) estimated curve (blue), built end point set (magenta) and built curve set (red) after 1st, 6th, 12th iterations; (j) close-up of the ground-truth curve (green), simulated noisy curve (black) and final estimated curve (blue)



(a) blurry image (b) 1st iteration (c) 4th iteration (d) 7th iteration

(e) measured noisy curve (f) 1st iteration (g) 4th iteration (h) 7th iteration

Fig. 2. Experiment results with real image and measured accelerations. (a) blurry image; (b)-(d) estimated latent image after 1st, 4th, 7th iterations; (e) curve calculated from measured accelerometer data (black) unit:cm; (f)-(h) estimated curve (blue), built end point set (magenta) and built curve set (red) after 1st, 4th, 7th iterations

5. REFERENCES

[1] Neel Joshi, Sing Bing Kang, C Lawrence Zitnick, and Richard Szeliski, "Image deblurring using inertial measurement sensors," *ACM Transactions on Graphics (TOG)*, vol. 29, no. 4, pp. 30, 2010.

[2] Yu-Wing Tai, Ping Tan, and Michael S Brown, "Richardson-lucy deblurring for scenes under a projective motion path," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 33, no. 8, pp. 1603–1618, 2011.

[3] Oliver Whyte, Josef Sivic, Andrew Zisserman, and Jean Ponce, "Non-uniform deblurring for shaken images," *International journal of computer vision*, vol. 98, no. 2, pp. 168–186, 2012.

[4] Ankit Gupta, Neel Joshi, C Lawrence Zitnick, Michael Cohen, and Brian Curless, "Single image deblurring using motion density functions," in *Computer Vision–ECCV 2010*, pp. 171–184. Springer, 2010.

[5] Zhe Hu and Ming-Hsuan Yang, "Fast non-uniform deblurring using constrained camera pose subspace.," in *BMVC*, 2012, pp. 1–11.

[6] Xin Zhang and Fuchun Sun, "Blind nonuniform deblur under projection motion path," *Journal of Electronic Imaging*, vol. 22, no. 3, pp. 033034–033034, 2013.

[7] Sunghyun Cho and Seungyong Lee, "Fast motion deblurring," in *ACM Transactions on Graphics (TOG)*. ACM, 2009, vol. 28, p. 145.

[8] YK Thong, MS Woolfson, JA Crowe, BR Hayes-Gill, and DA Jones, "Numerical double integration of acceleration measurements in noise," *Measurement*, vol. 36, no. 1, pp. 73–92, 2004.

[9] Qi Shan, Jiaya Jia, and Aseem Agarwala, "High-quality motion deblurring from a single image," in *ACM Transactions on Graphics (TOG)*. ACM, 2008, vol. 27, p. 73.

[10] Anat Levin, Rob Fergus, Frédo Durand, and William T Freeman, "Image and depth from a conventional camera with a coded aperture," *ACM Transactions on Graphics (TOG)*, vol. 26, no. 3, pp. 70, 2007.

[11] Ruiwen Zhen and Robert L Stevenson, "Joint deblurring and demosaicking of cfa image data with motion blur," in *Proc. SPIE 9029, Visual Information Processing and Communication V*. International Society for Optics and Photonics, 2014.