Camera Motion Tracking for Deblurring and Identification

Roarke Horstmeyer, MIT Media Lab
MAS 863 Final Project, Spring 2010

Abstract

Motion blur in images is often an artifact of the camera moving and rotating during a finite exposure time. It can be modeled as a convolution between an original sharp image and a blur kernel that represents the camera's motion. In this project, I measure the camera's motion and rotation by adding a sensor containing accelerometers and gyroscopes. Estimated motion from this sensor is then used in an aided deconvolution process. Furthermore, the possibility of using this motion data for user identification is investigated.

1. Introduction

Conventional digital optical sensors like charge-coupled devices (CCD) or CMOS arrays typically require on the order of 10^3 photons per bit of delivered digital information. Thus, in low light settings, long exposure times are required to use the entire 8-10 bits per pixel and also overcome noise challenges. Low light situations often arise when there is little ambient lighting, when the aperture is significantly stopped down to increase depth of field, and in small-scale imager layouts, as with current cell phone camera designs.

A common problem tied to low-light conditions and the requirement of long exposure times is motion blur. This can arise from either the photographed object moving, the camera moving, or both. While a lot of previous work has focused on removing blur due to the camera's motion, a couple projects have examined digitally inverting object motion blur [1,2]. This project will examine a certain method of removing blur due to camera motion that has the potential to be combined with the object motion removal architecture in [1].

Specifically, this project will explore the use of accelerometers and gyroscopes to estimate the 6D position and orientation vector of the camera over its exposure time. Both of these sensors, along with digital cameras, are found in an increasing number of cell phones. This is one of the main motivations in choosing these sensors to obtain a 6D position/orientation measurement. By using accelerometers and gyros to measure how the camera moves and rotates, an estimate of the blur kernel (temporal point spread function (PSF)) is created. Camera-generated motion blur can generally be modeled as a convolution between the desired sharp image and a blur kernel representing the camera's motion. Thus the measured PSF, which is currently assumed to be shift-invariant, is used in an aided deconvolution process in an attempt to recover a sharp image.

Besides deblurring, this project begins to investigate the possibility of using camera motion and rotation estimation for user identification. A database of images and associated motions for a number of individuals is collected to test whether these movements are not completely random. Relevant indicators may include age, gender, and photography experience. These results are very preliminary.

2. Background

There are a number of previous hardware and software-based approaches for motion blur minimization. Below is a short summary of relevant designs and a discussion of the novelty of the proposed design.

Hardware-based Approaches

One of the most commonly used approaches to minimize motion blur is to keep the camera as stationary as possible, as with a tripod or a Steadicam [3]. If the user wants to be free of restricting externals, there are also several blur-minimization techniques available in consumer cameras. These designs can broadly be split into two groups: sensor-based and lens-based approaches. Some early lens-based designs [4] used physical inertial means to keep image formation as stable as possible. Other more advanced designs (e.g. Nikon VR [5], Mega OIS [6]) use two gyroscopic measurements to either pause until no motion is detected, or to cause two electromagnets to shift a compensation lens in the middle of the lens set. Sensor-based approaches (e.g. Sony Steadyshot [7], Pentax shake-reduction [8]) apply the same electromagnet-gyroscope pair idea to the digital sensor. Unlike the proposed design, these systems only measure 2 degrees of rotation and not position. Likewise, no image post-processing is performed: gyroscope measurements are directly used for optical stabilization.

Software-based Approaches

Digital image deblurring implemented post-capture is a long-standing problem in computer vision and graphics. An early and influential method of deblurring with a known blur kernel is from Richardson [9], which I will use for demonstration. However, there are many other more sophisticated techniques, a good summary of which is in [10]. There are two quite recent attempts at software based deblur that are quite similar to this effort. Fergus et al. [11] use a Bayesian statistics approach and natural image statistics to estimate a PSF kernel, while Joshi et al. [12] attempt to preserve sharp edges. One problem with attempting to deblur post-capture is an irrecoverable loss of high-frequency content, noticed by [1, 2]. For this reason, joint hardware and software approaches often yield optimal results.

One relevant example of a joint approach is from Ben Ezra and Nayar [13], who utilize a high-speed camera with a high-resolution SLR to aid in deblurring. This technique could be used to validate accuracy of the current design in the future. Finally, there have been two other recent suggestions for motion estimation using accelerometer data, from Park et al. [14] and a to-be-released paper by Joshi et al. [15]. This project attempts to build on these designs by adding a measure of rotation, and attempting to identify users based on their 6D motion patterns.

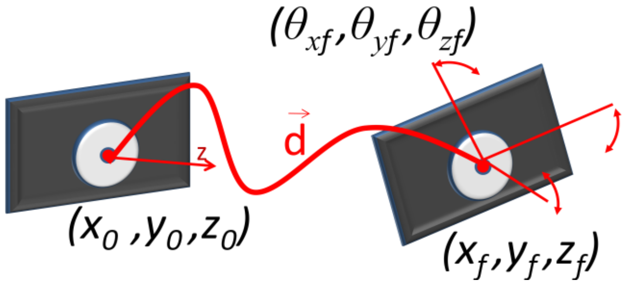

Figure 1: The motion of a camera through time can be modeled with 6 degrees of freedom. We will assume initially the camera's optical axis lies in the z-direction. The camera will follow a 3D vector in space *d*, and will rotate about the three axes each movement. Positional change can be measured with integrated accelerometer measurements, and rotational change from integrated gyroscopic measurements.

3. Motion and Rotation to Image Blur

This section will first show how position and angle can be mapped to blur values, and discuss which position and angle values are most important in this procedure. Then, since the proposed camera design can only measure acceleration and rotational rate with accelerometers and gyroscopes, I will show how these values can be transferred effectively to position and angle coordinates..

3.1. Obtaining a position and angle measurement

As mentioned previously, image motion blur can be modeled as a convolution between an original sharp image and a kernel representing the motion of the object or camera. In reality, this kernel is shift-variant – one side of an image may see a different direction and shape of blur than the other side. However, for simplicity, this project will assume a shift-invariant blur kernel. This allows the recorded image *S* to be modeled as,

$$S = I * K + N \qquad (1)$$

where *I* is the original sharp image, *K* is the blur kernel and *N* is noise. The consequences of this assumption will become clear in the next section.

Focusing only on camera motion, a well-known equation in the computer vision community models the projection of an image as a function of camera rotation *R* and translation *T* with,

$$V_t = K\left(R + \frac{1}{d}TU^T\right)K^{-1} \qquad (2)$$

$$(u_t, v_t, 1)^T = V_t(u_0, v_0, 1)^T \qquad (3)$$

where *V* is a transformation matrix for image points *(u,v)* over time, *d* is scene depth, and *U* is a vector normal to the image plane. From (2) and (3), it is clear that if *R* and *T* can be measured, the image's transformation over time can be determined, yielding the blur kernel *K*.

I will now simplify these equations. Using the geometry shown in Figure 1, where the *z* is along the optical axis, we will consider how each of the 6 position and orientation vectors independently contribute to blur in discrete time steps. Figure 2(a) shows how motion perpendicular to the optical axis, given by *x* (and an orthogonal movement vector *y*), yields a blur measurement on the sensor p_m:

$$\vec{p}_m = \begin{bmatrix}\vec{p}_{mx} \\ \vec{p}_{my}\end{bmatrix} = M\begin{bmatrix}\vec{x} \\ \vec{y}\end{bmatrix}, \qquad (4)$$

where M is the magnification of the camera and x and y are motion vectors. Likewise, from Figure 2(b), the blur p_r due to rotation about the axes perpendicular to the optical axis can be modeled with

$$\overrightarrow{p_r} = \begin{bmatrix} \vec{p}_{r\theta x} \\ \vec{p}_{r\theta y} \end{bmatrix} = d_i \begin{bmatrix} \tan \vec{\theta}_x \\ \tan \vec{\theta}_y \end{bmatrix}, \qquad (5)$$

where d_i is the distance from the lens to the sensor, typically the camera focal length under the assumption that objects are close to infinity. It is clear that these 4 variables of motion perpendicular to the optical axis are quite important to model the motion blur kernel. However, what about motion along and rotation about z, the optical axis? The blur here, unlike above, will not be a function of movement. Instead, it can be modeled as a blur function G_z, which is also dependent on object magnification and the lens depth of field (DOF):

$$\vec{p}_z \propto G_z(\vec{z}, M, DOF) \qquad (6)$$

For the current project, we will assume the function of this blur is minimal. Likewise, rotation around the optical axis will result in a blur that is *not* shift invariant (imagine an image being spun about its center, where the center doesn't move/blur but edges do). Thus, for this project we will also assume that the camera is relatively steady about the optical axis, which proved to be valid in experiment. The final blur kernel p can be given as a function of the three blurs above,

$$\vec{p} = G_z(\vec{z}, M, DOF) * \left(\begin{bmatrix} \vec{p}_{mx} \\ \vec{p}_{my} \end{bmatrix} + \begin{bmatrix} \vec{p}_{r\theta x} \\ \vec{p}_{r\theta y} \end{bmatrix} \right). \qquad (7)$$

Next, we will see how to get position and angle measurements for these 5 required variables.

3.2. From measurements to position and angle

Without going into too much detail, transferring from acceleration and angular velocity measurements is not as simple as just performing integrations. Since the sensor is not at the exact center of mass (COM) of the camera system, the two variables are not independent. Likewise, since gravity is a constant vector that needs to be subtracted out, rotations cause this subtraction to vary in coordinate application. Specifically, the total *measured* acceleration a_m and angular velocity ω_m of the system can be described as,

$$\begin{aligned} \vec{a}_m(t) = R[\vec{a}(t) + \vec{g} + \vec{\omega}_m(t) \\ \times (\vec{\omega}_m(t) \times \vec{r}) + (\vec{\alpha} \times \vec{r})] \end{aligned} \qquad (8)$$

$$\vec{\omega}_m(t) = R * \vec{\theta}(t), \qquad (9)$$

where R is the rotational operator from above, r is distance between COM and sensor, and acceleration can be seen as a vector sum of measured acceleration, gravity, centripetal acceleration and rotational acceleration.

This equation pair can be solved through iteration, beginning by finding the rotational state R of the camera at every time step. The starting and evolution conditions for this are,

$$\theta(0) = 0, R(0) = 0 \qquad (10)$$

$$\theta(t) = R(t-1)\omega(t-1)\Delta t + \theta(t-1) \qquad (11)$$

Where $R(t)$, the updated rotation operator, is just a matrix of mapped $\theta(t)$ values. After all rotation operators $R(0)$ – $R(n)$ are determined, stationary acceleration $a(t)$ in (8) can be solved though a similar iteration process. With this set of iterations, the variable set (x,y,θ_x,θ_y) can eventually be determined through (discrete) integration:

$$x(t) = \int\int a(t)\, dt^2 \qquad (12)$$

$$\theta(t) = \int \omega(t)\, dt \qquad (13)$$

These values can in turn be used in (4) – (7) to determine sensor blur.

3.3. Required Assumptions

The above analysis is possible due to a number of simplifications and assumptions. Following in a brief list of some of the major assumptions, all of which generally hold true during experiment, but may contribute to error:

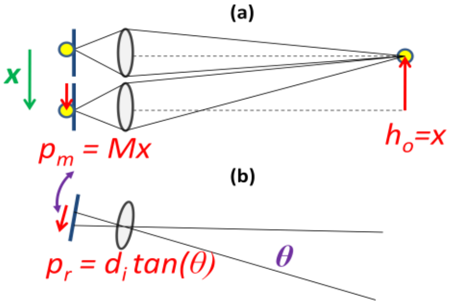

Figure 2: Simple geometrical constructions to show how blur distances can be determined from camera position and orientation. (a) Camera translation along x results in the image of a point source turning into an extended object, whose size is given by camera magnification. (b) Similar triangles can give the blur distance from rotation about the optical center as a function of angle.

1) $\omega(0) = 0$ (no initial angular velocity)
2) $v(0) = 0$ (no initial velocity)
3) A single object plane
4) Gravity vector is initially aligned with *y*
5) Movement small enough to allow successful deconvolution

The most limiting of these assumptions is actually (5), since a large motion is desired so that is easily detected, but it cannot be large enough to degrade high spatial frequencies.

4. Sensor Construction and Noise

4.1. Camera Setup

From (7), it is clear that 5 variables will be needed to determine the blur kernel *p*: 3 position measurements and 2 rotation measurements. For this task, a three-directional accelerometer (ADXL-335) and a two-directional gyroscope (IDG500) were placed as close one another as possible on the bottom side of the camera (Figure 3). The ADXL-335 is spec'd to give a 300mV/g sensitivity, with a 150µg/Hz$^{1/2}$ RMS noise density (leading to ~1mg RMS noise). Likewise the IDG500 is supposed to yield a 2mV/deg/s sensitivity. Experiment measured associated sensitivity values to be much larger, however, as will be discussed in the next section.

For this experiment, a Canon Digital SLR Rebel XSI with an f/1.8 Canon lens was used. To trigger the sensors, the camera's flash hotshoe was used. When the camera is not taking a picture, the hotshoe is at a roughly constant 800mV, and during exposure the hotshoe drops to 0V. The typical function of this operation is to provide an attached or external flash with information on exposure time. It proved a useful method of triggering data readout. Sensors were powered with 3.3V and shared the hotshoe's ground. The accelerometer – gyroscope pair and hotshoe output are all fed to the analog inputs of an Arduino microcontroller, also physically attached to the camera (Figure 4). The sensor outputs were read at roughly 50Hz, yielding approximately 25-50 measurements per average exposure. The microcontroller's 6 analog inputs are then digitized and send via USB to a laptop, which also interfaces with the camera to store images. Both Processing and Matlab were used to grab and process sensor and image data.

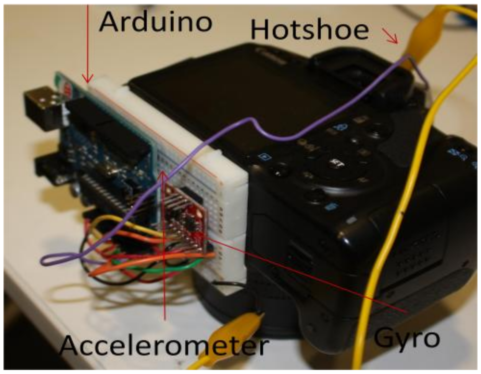

Figure 3: An image of the camera sensor setup. The sensor was attached to the flat surface on the bottom of a Canon SLR. Accelerometers are aligned such that all are orthogonal and one is parallel to the optical axis. Gyros are aligned to measure toop-bottom rotation and front-back rotation, *not* rotation about the optical axis. Distance from the sensor to the camera COM is roughly 3cm.

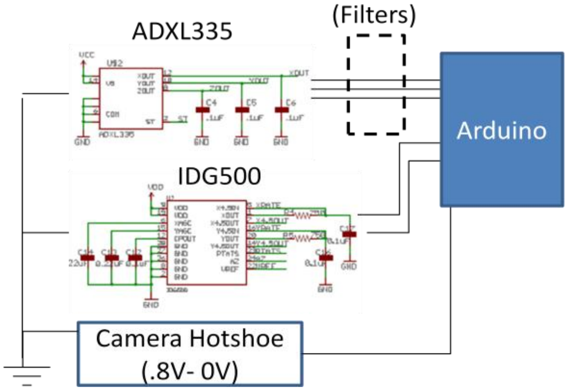

Figure 4: Circuit schematic for the sensor attached to the camera. 5 analog outputs and a trigger from the camera hotshoe are used as inputs to an Arduino microcontroller, which digitizes the signal. Low-pass filters were used to roll-off high frequency noise, limiting BW to 50 Hz. Filters were tested to reduce accelerometer noise before reaching the microcontroller.

4.2. Accelerometer Random Walk Noise

A common issue with using accelerometers for position measurement is the amplification of noise. Low and mid-end accelerometers are typically somewhat noisy to begin with, with contributions from white noise, temperature bias and measurement drift. When this noise value is integrated twice over, it can lead to significant changes in positional estimation, known as random walk. Specifically, white noise can lead measured acceleration variance to increase over the number of measurements (i.e., acceleration drift). This drift begins proportional to the square root of the number of measurements, which becomes squared each integration step.

The measured noise characteristics for the sensor used in this experiment are shown in Figure 5. Average standard

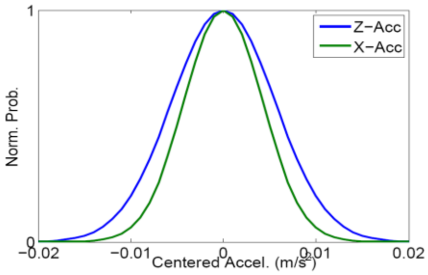

Figure 5: Plot of noise performance for two centered accelerometer readings. White noise is assumed to dominate and follow a Gaussian distribution. Measurements for x and z (here z is aligned with gravity) are different because of different measured acceleration ranges.

deviation was on the order of 0.007 m/s^2, which integrates to quite a small positional error (on the order of hundreds of microns). However, since the average acceleration measurement is these experiments was on the order of 0.05-0.1 m/s^2, the total average SNR is really quite low – typically on the order of 15-20 dB. Thus, random walk error proved to be a large source of error in this experiment.

4.3. Accelerometer Bandpass Filtering Schemes

To minimize the integration of accelerometer noise into position error, several different filtering schemes were attempted. Previous work suggests that conventional, Butterworth and Kalman filters (for use in digital post-processing) all work well at reducing random walk [16]. Support for these filters rests in their ability to reduce high-frequency content, which is assumed to be noise-related. However, they generally reduce sensor bandwidth. In the current configuration, the RMS noise N_{RMS} is

$$N_{RMS} = N_{Density}\sqrt{1.6BW}, \qquad (14)$$

where the noise density is given as 150μg/Hz$^{1/2}$ and BW is the bandwidth.

An initial low-pass filter is implemented in the current circuit with R=32kOhm and C=.1μF, limiting the bandwidth to roughly 50Hz. Different additional Butterworth filters were also attempted to quickly roll off high frequency noise, both in analog and in digital processing of the analog signal. The Sallen-Key circuit shown in Figure 6 was first attempted as a two-pole Butterworth to minimize noise readout from the analog signal. Several different parameters were varied, including order and frequency cutoff. However, it was generally found to reduce the signal's dynamic range too much (Figure 7). A Butterworth filter with a larger cutoff was then implemented digitally, which smoothed out the signal less. Kalman filters were also attempted without success. In the end, 4th order Butterworth filters with a normed cutoff frequency of 0.6-0.7 times the Nyquist frequency were found to optimally reduce stationary noise without over-smoothing data.

In general, applying these types of filters may not be optimal for this specific application. Often, accelerometers and gyros are used on moving machines, which tend to vibrate. Low-pass Butterworth's and Kalman's are found to work well at reducing random walk from vibration [16], but these small vibrations (~.1mm) are important in the measurement of camera shake, as blur kernels are on the order of .5-1mm. In summary, accelerometer white noise propagates to have a large drift effect on the small measurements that are required to determine the correct blur kernel, making deblurring difficult to achieve. Optimal filtering schemes are still being investigated.

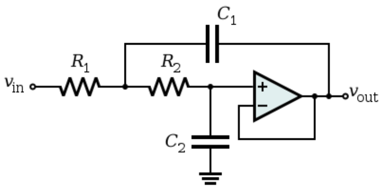

Figure 6: Unity gain Sallen-Key layout used as a low-pass filter - a second-order Butterworth filter. This filter, as well as other similar low-pass configurations, were experimentally tested to de-noise accelerometer output. Placement in the circuit is indicated by the dotted filter rectangle in Figure 4. Although it reduced noise, these filters smoothed the signal too much (decreased the bandwidth) for the current application.

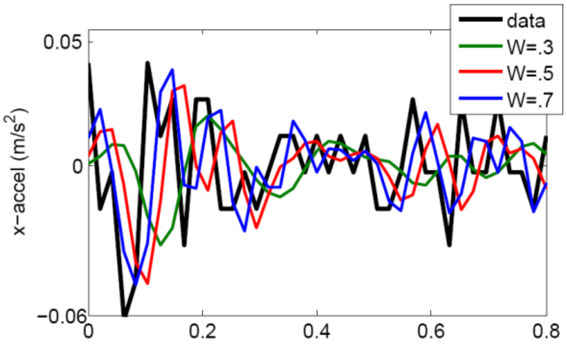

Figure 7: A typical accelerometer output reading for a .8sec. exposure is shown as the black curve, with 50Hz sampling. To reduce noise, various low-pass filtering schemes were attempted. In green, red, and blue, are curves associated with applying a 4th order Butterworth filter with a cutoff at .3, .5 and .7 Nyquist. Note that as the cutoff is reduced, so is high-frequency noise, but the useful signal also becomes washed out (dynamic range is reduced). A low SNR in this specific application causes accelerometer noise reduction to be a tough problem to solve.

4.4. Gyroscope Output

Gyroscope noise is not as significant an issue as accelerometer noise. This is partially due to a more accurate MEMs-based sensor reading, and partially due to the single integration required to determine angular orientation. Still, filtering schemes do exist for reducing random walk [17]. However, it was determined through simple experiment that the standard deviation for the gyroscopes is on the order of 0.5 degree per second, which did not lead to significant random walk over the roughly 0.5 second exposure time.

5. Experimental Results

Following are some preliminary experimental results demonstrating how the sensor's 5D output can be used for image deblurring and user identification. Before data could be taken, several calibration steps were performed to map digital outputs to actual movement values. The camera was laid in 6 different directions *+/-(x,y,z)* to find acceleration with respect to gravity, and was turned repeatedly at a specific rate to estimate angle mapping.

5.1. Image Deblurring

An image with 0.8 second exposure taken with a somewhat shaky hand is presented in Figure 8. Position and angle measurements were combined from the sensor arrangement described in Section 4 to predict a blur function given by (7). The detected blur kernel is shown in Figure 9. To check approximate movement values, a blind deconvolution was used to estimate the blur kernel size. Furthermore, a point source of light included in the scene as a rough indication of the measurement's correct shape (Figure 8, upper right). A more robust verification procedure is certainly desired for future work, and the setup in [8] may be implemented for these purposes.

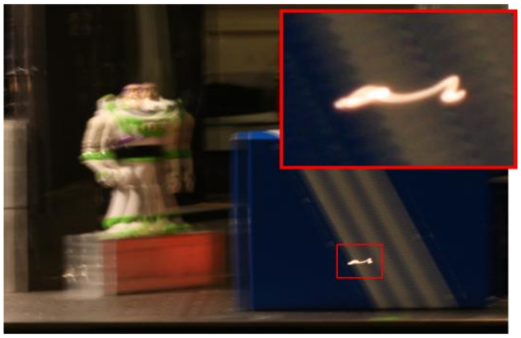

Figure 8: An example blurry image taken with the setup described in Section 5, with an exposure of .5s. A point source was placed on the blue box to give an indication of blur shape. Note that motion is primarily in the horizontal direction, due both to movement and rotation, with some jittering.

A deconvolved result with a simple Richardson-Lucy procedure is also shown in Figure 9(b). In order to create the blur kernel for input, interpolation was used between the 40 available data points. In general, deconvolution is successful for simple blur shapes. More complicated shapes, or shake that lasts longer than roughly 1 sec. exposure, is difficult to detect accurately. An example of this difficulty is in Figure 10, a .6 second exposure, which transverses a more complicated path. Note that the sensor has neither the spatial or temporal resolution to detect the loop towards the left of the blur.

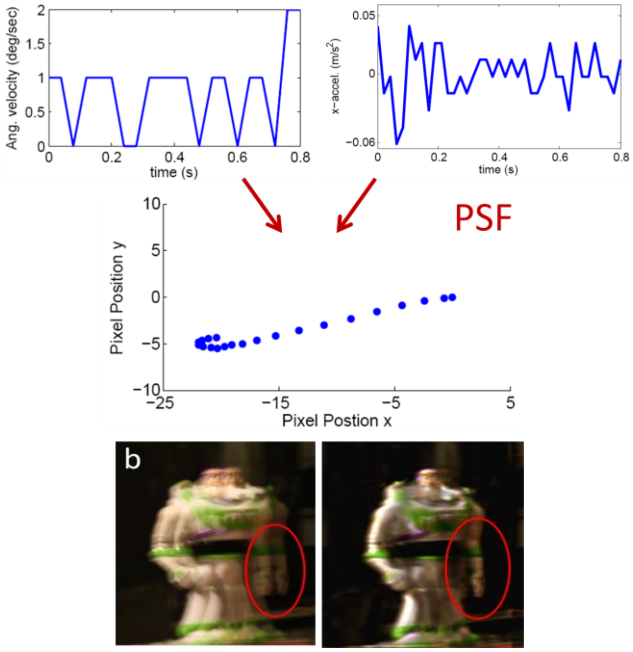

Figure 9: Data from accelerometers (top right) and gyroscopes are (top left) combined to create an estimated blur function (middle). This blur function is then used as a naïve input to the Richardson-Lucy deconvolution algorithm, which is able to transform the original image to a sharp image estimate.

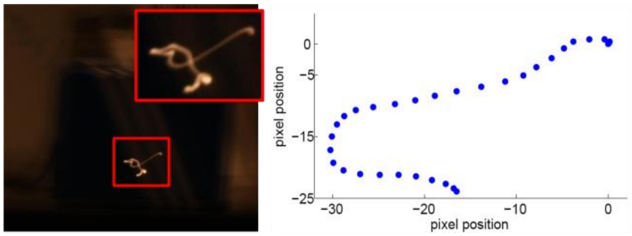

Figure 10: An example of a situation where the accelerometer fails to detect a correct blur shape. The loop that the camera goes through in its finite exposure is not correctly identified by the position sensor due to lack of resolution, which becomes a significant portion of the deconvolution operation. Deconvolution in this case will result in a poor reconstruction.

5.2. Camera User Identification

Although shake and rotation resolution is slightly too noisy for a robust deblurring system, this system may have high enough resolution to detect shake differences between different users. Initial work has begun into testing the assumption that users have unique blur signatures that appear in a high percentage of their photographs. Figure 11 contains some preliminary data sets from 4 different subjects. While an encompassing analysis has yet to be performed on individual's particular shake patterns, there does appear to be certain trends. For example, Subject 2 tends to move down to the left, Subject 3 from left to right, and Subject 4 from top-left to bottom right. In the future, a learning-based segmentation approach (e.g., Boosting) will be applied to a verified set of uniform subject data and then tested for accuracy.

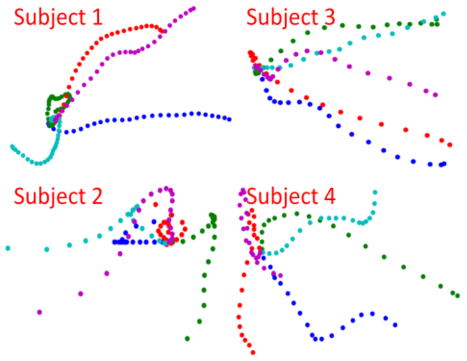

Figure 11: An example of a situation where the accelerometer fails to detect a correct blur shape. The loop that the camera goes through in its finite exposure is not correctly identified by the position sensor due to lack of resolution, which becomes a significant portion of the deconvolution operation. Deconvolution in this case will result in a poor reconstruction.

6. Future Work

A good deal of room is left to improve upon the current sensor setup. To begin, a more thorough and clever filtering scheme could be implemented, to maximize bandwidth while minimizing high-frequency noise. Since 6 different sensors are used, a Kalman filter approach that is often used with IMU's and combines channels to estimate true signal might work best. Furthermore, high-frequency image content can be preserved with additional hardware, like a coded shutter from [1]. Such an addition could significantly improve large motion deblurring results. Finally, more data from individual test subjects is being collected to determine if there is any robust indicators given from a camera's blur shape.

References

[1] R. Raskar, A. Agrawal, J. Tumblin, Coded Exposure Photography: Motion deblurring using a fluttered shutter, SIGGRAPH 2006.

[2] A. Levin, P. Sand, T. Cho F. Durand and W.T. Freeman, Motion-Invariant Photography, SIGGRAPH 2008.

[3] Steadicam Camera Stabilitzation Systems, From Tiffen Corp., http://www.steadicam.com/

[4] Canon, EF Lens Work III, The Eyes of EOS, Canon Inc, 1993

[5] Nikon Image Stabilization Overview, http://imaging.nikon.com/products/imaging/technology/vr/index.htm

[6] Panasonic Mega-OIS Image Stabilizer, http://www2.panasonic.com/webapp/wcs/stores/servlet/MegaOISExplained?storeId=15001

[7] Sony Cybershot Cameras with Steadyshot Feature, www.SonyStyle.com/CybershotCameras

[8] Pentax Shake Reduction Feature, http://www.pentaximaging.com/files/scms_docs/SHAKE_REDUCTION_FACT_SHEET.pdf

[9] W.H. Richardson, Bayesian-based iterative method of image restoration, JOSA(62) 55-59, 1972

[10] D. Kunder and D. Hatzinakos, Blind-image deconvolution, IEEE Signal Processing Magazine(13) 3, 1996

[11] R.. Fergus, B. Singh, A. Hertzmann , S.T. Rowles and W.T. Freeman, Removing camera shake from a single photograph, ACM Trans. Graph. 25, 2007

[12] N. Joshi, R. Szeliski and D.J. Kriegman, PSF estimation using sharp edge protection, IEEE CVPR, 2008

[13] M. Ben-Ezra and S.K. Nayar, Motion-based motion deblurring, IEEE PAMI 26(6) 2004

[14] S.Y Park, E.S. Park, and H.I. Kim, Image deblurring using vibration information from 3-axis accelerometer. Journal of the Institute of Electronics Engineers of Korea (45) 3, 2008

[15] N. Joshi, S.B. Kang, C.L. Zitnick and R. Szeliski, Image Deblurring using Inertial Measurement Sensors, To appear in ACM Trans. Graphics SIGGRAPH, 2010

[16] G. Pang and H. Liu, Evaluation of a Low-cost MEMS Accelerometer for Distance Measurement, J. Intel. And Robotic Systems (30), 2001

[17] M. Algrain, Kalman filtering approach for reducing gyroscope noise effects in stabilized platforms, Proc. SPIE 1697, 1992