EFFICIENT IMAGE CORRESPONDENCE MEASUREMENTS IN AIRBORNE APPLICATIONS USING INERTIAL NAVIGATION SENSORS

Matthew Woods and Aggelos Katsaggelos

Northwestern University
Department of Electrical Engineering and Computer Science, McCormick School of Engineering and Applied Sciences, Northwestern University, Evanston, Illinois 60208-3118
phone: + (1) 847-894-5953, email: mcw430@northwestern.edu, aggk@eecs.northwestern.edu

ABSTRACT

This paper presents a computationally efficient method for the measurement of a dense image correspondence vector field using supplementary data from an inertial navigation sensor. The application is suited to airborne imaging systems (such as on a UAV) where size, weight, and power restrictions limit the amount of onboard processing available. The limited processing will typically exclude the use of traditional, but expensive, optical flow algorithms such as Lucas-Kanade. Alternately, the measurements from an inertial navigation sensor lead to a closed-form solution to the correspondence field. Airborne platforms are also well suited to this application because they already possess inertial navigation sensors and global positioning systems (GPS) as part of their existing avionics package. We derive the closed form solution for the image correspondence vector field based on the inertial navigation sensor data. We then show experimentally that the inertial sensor solution outperforms traditional optical flow methods both in processing speed and accuracy.

1. INTRODUCTION

Image correspondence vector fields for frame to frame motion in a video sequence are an enabling input data item for a number of image processing algorithms including computer vision, optical flow measurement, and super-resolution enhancement [2], [10]. Traditional methods for generating dense correspondence maps, such as Lucas-Kanade and Horn-Schunk [3], continue to be a challenge in the image processing community due to both their computational complexity as well as their inherent reliance on sufficient image texture. For conditions in which the image flow is dominated by the motion of the sensor platform as opposed to that of individual objects in the scene, an alternative method is to directly calculate the frame to frame correspondence based upon data from an inertial navigation sensor.

Landscape video taken from an airborne platform, such as a UAV, is well suited to the above conditions. The landscape itself is essentially static in an earth fixed reference frame; so, all of the observed image motion is due to the linear and angular motion of the sensor platform. Additionally, airborne platforms have the characteristics 1) they already have an embedded inertial navigation sensor as part of their avionics package and 2) size, weigh, and power restrictions may be prohibitive for the high performance computing power needed to estimate motion fields in real-time using image based algorithms.

The challenge in utilizing an inertial navigation sensor is that, in order to generate the sub-pixel accuracies required by algorithms such as super-resolution enhancement, the inertial navigation sensor and the imaging sensor must be well aligned and calibrated. In general, this precision alignment will require specialized equipment which may not be practical for small platforms. Therefore, an online calibration procedure is proposed.

The remainder of the paper is organized as follows. In section 2, we first introduce the geometric models of the inertial navigation sensor and imaging sensor respectively. In section 3, we describe the process by which the inertial sensor measurements are utilized to calculate image correspondence. In section 4 we discuss the proposed method for performing a periodic online calibration of the two sensor systems. The section 5 we presents experimental results using simulated imagery and section 6 concludes the paper.

2. SENSOR MODELS

2.1 Inertial Navigation Sensor Model

Advances in both inertial navigation technology using micro electromechanical systems (MEMS) as well as global positioning system (GPS) receivers has led to low size, weight, power, and cost integrated GPS and inertial navigation sensors. The MEMS inertial sensors consist of an orthogonal triad of linear acceleration measurement devices and an orthogonal triad of angle rate measurement devices. The outputs of the inertial sensors and the GPS are optimally combined in a filter to produce an accurate navigation solution [4].

The output of the inertial navigation sensor is a current position, velocity, and attitude of the air vehicle relative to an earth fixed reference frame. Typically, the reference frame is aligned to the local north, east, and down directions (NED). The position output is the geodetic latitude, longitude, and altitude of the platform (φ, θ, h). The velocity output is provided in the NED coordinate system, V^{NED}. The attitude output is represented as a 3x3 orthonormal direction-cosine-

 ISSN 2076-1465

matrix (DCM) mapping vectors in the NED coordinate system to the platform coordinate system, P; i.e. for an arbitrary vector v:

$$v^P = [T^P_{NED}]v^{NED}$$

2.2 Imaging Sensor Model

For purposes of deriving geometric relationships between the inertial sensors and the imaging sensors, it is convenient to utilize a normalized perspective projection model of the image sensor as discussed in [1] (see Figure 1). In such a model, the image plane is considered to be located at a unit distance from the focal point such that a 3D object located at space vector $R^S = [x \quad y \quad z]^T$ in a coordinate system S attached to the sensor, will be projected to a normalized pixel location

$$[x' \quad y' \quad 1]^T = [x/z \quad y/z \quad 1]^T$$

The normalized pixel location $(x',\ y')$ will not correspond directly to the true pixel location (u, v) of the projection of R^Sonto the image plane because the real camera will not perfectly match the idealized projective projection model. However, from the intrinsic calibration parameters of the camera, there is a known mapping function, $f\ (\cdot)$, such that

$$(u, v) = f(x', y')$$
$$(x', y') = f^{-1}(u, v)$$

The functional form of $f\ (\cdot)$ depends upon if the sensor is best described by a thin lens, thick lens, or other more intricate optical model. In either case, the function $f\ (\cdot)$ is assumed known via factory calibration of the sensor. The orientation of the sensor relative to the platform is represented as a 3x3 orthonormal DCM mapping vectors in the P coordinate system to the S coordinate system; i.e. for an arbitrary vector v

$$v^S = [T^S_P]v^P$$

3. CORRESPONDENCE MAPPING

3.1 Geometric Mapping

Given the geometric models of section 2, it is possible to explicitly calculate the correspondence field of the sensor between video frames. Let $R^{NED}_G(x', y', k)$ represent the 3D vector from the platform to the ground projection of normalized image pixel (x', y') on frame k. Call this ground projection point G. Let $(x' + \Delta x, y' + \Delta y)$ represent the projection of the same ground point G back onto the normalized sensor image on frame $k + 1$ (see Figure 2). Then, $(\Delta x, \Delta y)$ is the correspondence vector for pixel (x', y') between frames k and $k + 1$. In order to calculate the motion of the ground point G in the normalized image plane, it is necessary to first compute the vector $R^{NED}_G(x', y', k)$ according to

$$R^{NED}_G(x', y', k) = \frac{\|R^{NED}_G(x', y', k)\|}{\|[x' \quad y' \quad 1]^T\|}[T^P_{NED}]^T_k[T^S_P]^T\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix}, \tag{1}$$

where the subscript "k" on the NED to platform DCM indicates that it represents the orientation of the platform at a time coincident with video frame k. The "T" superscript indicates the matrix transpose. Because the DCM matrices are orthonormal, the matrix transpose is equivalent to the matrix inverse. The magnitude of the projected line from the platform to the ground point G, $\|R^{NED}_G(x', y', k)\|$, is calculated based on the altitude and position of the platform. This calculation is discussed in more detail in section 3.2. The second step is to calculate the translational motion of the platform, ΔR, based on the average velocity, that is,

$$\Delta R = \Delta t\frac{V^{NED}_{k+1} + V^{NED}_k}{2}, \tag{2}$$

where, again, the subscript "k" on the velocity indicates the velocity indicated by the inertial navigation sensor at a time coincident with the video frame k. The time period Δt is the time interval between frames k and $k + 1$. ΔR is expressed in the NED coordinate system. The third step is to adjust the position of the fixed ground point G relative to the platform using the position change, ΔR, that is,

$$R^{NED}_G(x' + \Delta x, y' + \Delta y, k + 1) = R^{NED}_G(x', y', k) - \Delta R \tag{3}$$

The final step is to map the NED vector back into the sensor coordinate system, that is,

$$\begin{pmatrix} x' + \Delta x \\ y' + \Delta y \\ 1 \end{pmatrix} = \frac{1}{\alpha}\left([T^S_P][T^P_{NED}]_{k+1}R^{NED}_G(x' + \Delta x, y' + \Delta y, k + 1)\right), \tag{4}$$

where, α is a normalizing scale factor such that the third elememt on the left-hand side of equation (4) is equal to unity. Equation (4) is rearranged into a form suitale for computer implementation by substituting in equation (3) and subtracting the vector $[x' \quad y' \quad 0]^T$ from both sides. This yields,

$$\begin{pmatrix} \Delta x \\ \Delta y \\ 1 \end{pmatrix} = \frac{1}{\alpha}\left([T^S_P][T^P_{NED}]_{k+1}(R^{NED}_G(x', y', k) - \Delta R)\right) - [x' \quad y' \quad 0]^T \tag{5}$$

Again, α is a normalizing scale factor. Equation (5) represents the final closed-form solution to the image correspondence vector field. For every pixel location (x', y') on frame k, it computes the correspondence vector $(\Delta x, \Delta y)$ based on the known inputs $[T^S_P]$, $[T^P_{NED}]_{k+1}$, $R^{NED}_G(x', y', k)$, and ΔR.

3.2 Range to Ground

The range to the ground point, $\|R^{NED}_G(x', y', k)\|$, is obtained through the use of a digital terrain elevation database

(DTED). DTED is a table indexed by latitude and longitude that provides the elevation of the ground above sea level. Global coverage DTED based on the Shuttle Radar Topography Mission (SRTM) is publically available in 3 arc-second (level 1) or 1 arc-second (level 2) resolution from the Earth Resources Observation and Science Center. GPS position is all that is needed to accurately calculate ground range. See the geometry in Figure 2.

4. ON-LINE CALIBRATION

Assuming that intrinsic errors in both the inertial navigation sensor and the imaging sensor are minimized through factory calibration, the remaining error sources of interest are 1) misalignment in the installed orientation of the image sensor and 2) mis-synchronization between the image and inertial sensor data. These errors may be written as small angle modifications to the DCMs, that is,

$$\begin{aligned}[T_{NED}^{P}] &= (I - \omega_x \tau)\left[\hat{T}_{NED}^{P}\right] \\ [T_P^S] &= \left[\hat{T}_P^S\right](I - \partial_x)\end{aligned} \tag{6}$$

I represents the 3x3 identity matrix. ω is the 3-element angular velocity vector of the platform relative to NED (as returned by the inertial navigation sensor). τ represents the temporal mis-synchronization between the inertial and image data and ∂ is a three element vector representing the roll, pitch, and yaw misalignment of the platform to image sensor DCM. The subscript "x" applied to the vectors ω and ∂ in equation (6) is an operator that converts them into the 3x3 skew-symmetric cross-product matrix; i.e.

$$\partial_x = \begin{pmatrix} 0 & -\partial_3 & \partial_2 \\ \partial_3 & 0 & -\partial_1 \\ -\partial_2 & \partial_1 & 0 \end{pmatrix} \tag{7}$$

With that definition, the 3x3 matrices:

$$(I - \partial_x) \text{ and } (I - \omega_x \tau)$$

are small angle approximations to the DCMs generated by rotations about the x,y, and z axes given by the elements of ∂ and $\omega\tau$ respectively.

In equation (6), the DCMs with the ^ symbol indicate the uncorrected matrices and the DCMs without the ^ represent the post-correction. The goal is to estimate τ and ∂ such as to improve the correspondence calculation. The technique for doing so is to apply a traditional image based optical flow algorithm to a small set of pixels on each video frame and find the values of τ and ∂ that minimize the discrepancy between the correspondence vectors as measured via optical flow versus the correspondence predicted by the inertial sensor. Only a small number of data points are required to solve for the four unknowns in τ and ∂. Therefore, it is not necessary to generate a dense correspondence map using a computationally expensive, optical flow algorithm such a Lucas-Kanade. Instead, optical flow algorithm only need to return a relatively few correspondence vectors per frame. Once the error parameters, estimate τ and ∂, are resolved, equation (5) is used to find the dense flow field.

The parameter estimation for τ and ∂ may be linearized to the form

$$\alpha\left[\begin{pmatrix}\Delta x \\ \Delta y\end{pmatrix}_{IP} - \begin{pmatrix}\Delta x \\ \Delta y\end{pmatrix}_{IS}\right] = \begin{pmatrix}1 & 0 & 0 \\ 0 & 1 & 0\end{pmatrix}[A \quad A\omega]\begin{pmatrix}\delta \\ \tau\end{pmatrix} \tag{8}$$

where IP and IS refers to the correspondence vectors determined by the image processing and inertial system respectively. The matrix A is given by

$$A = [\hat{T}_P^S]\left[\frac{\|R_G^{NED}(x', y', k)\|}{\|[x' \quad y' \quad 1]^T\|}\left((y_{k+1}^P)_x - [\hat{T}_k^{k+1}](y_k^P)_x\right) - V_{k+1}^P \Delta t\right]$$

where,

$$y_{k+1}^P = [\hat{T}_k^{k+1}]\,[\hat{T}_P^S]^T \begin{pmatrix}x' \\ y' \\ 1\end{pmatrix}$$

$$y_k^P = [\hat{T}_P^S]^T \begin{pmatrix}x' \\ y' \\ 1\end{pmatrix}$$

$$\left[\hat{T}_k^{k+1}\right] = [\hat{T}_{NED}^P]_{k+1}[\hat{T}_{NED}^P]_k^T$$

$$V_{k+1}^P = [\hat{T}_{NED}^P]_{k+1}\left(\frac{V_{k+1}^{NED} + V_k^{NED}}{2}\right)$$

By equation (8), each data point generates two equations. Therefore, a minimum of two data points is required for a solution to the four error parameters τ and ∂. For a robust solution, many more data points are necessary yielding an overconstrained linear relationship which may be solved using a standard least squares approach. Or, if *apriori* probability distributions are available for the measurements and unknown parameters, through a maximum likelihood or maximum *a posteriori* estimation method.

5. RESULTS

The ideal means of testing the above algorithms is to collect data from an aircraft of UAV equipped with an imaging sensor and inertial navigation sensor. For the purpose of this paper, video data is simulated using a large aerial photograph [6] and remapping the imagery to the point of view of a sensor on a simulated aircraft. The mapping requires considering the large image as a ground fixed texture and then using the geometry of **Figure 1** to project each pixel of the virtual camera to the ground. Bi-linear interpolation is used to handle the non-integer relationship between the pixels of the virtual camera and the ground map. Figure 3 illustrates the ground image (top) and the simulated virtual camera image (bottom). The simulated aircraft is flying at 1000 m altitude with a 10 degree right bank angle. The virtual camera has a 60 degree field-of-view such that the ground projection

of the field-of-view is shown by the yellow box in Figure 3 (top).

Although the image plane of the virtual camera is square, the ground projection of the field-of-view is asymmetric due to the 10 degree roll angle of the simulated aircraft. In order to make a test video feed, it is necessary to generate a six degree-of-freedom trajectory for the simulated aircraft and repeat the virtual camera projection for a time series of frames.

To test the improvement in execution speed of the closed form solution to the correspondence vs. an optical flow technique, the closed-form option is coded up in Matlab and compared to a public code for performing the Lucas-Kanade algorithm written by Sohaib Khan [7]. Using the simulated data, and not introducing any errors on the inertial sensors or alignment errors as discussed in section 4, the closed-form solution, equation (5), represents the truth reference for the correspondence field. The error in the Lucas-Kanade method is expressed as a histogram in Figure 4. For the purpose of these figures, the correspondence error per pixel is defined as

$$\epsilon = \sqrt{(\Delta x_{LK} - \Delta x_{TRUE})^2 + (\Delta x_{LK} - \Delta x_{TRUE})^2}$$

where the subscripts "LK" and "$TRUE$" correspond to the Lucas-Kanade estimate and the truth reference respectively.

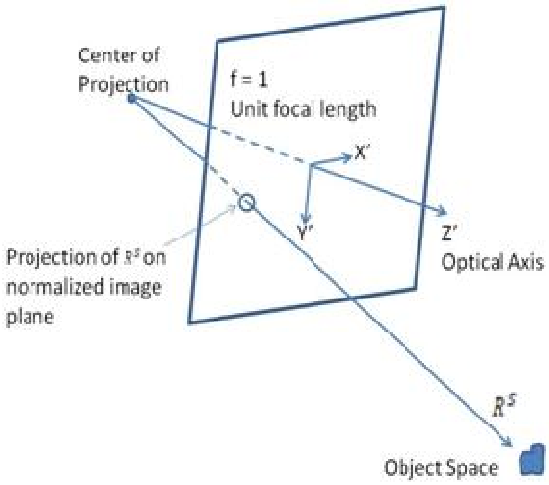

Figure 1: Normalized Perspective Projection Model

The histogram of ϵ in Figure 4 is generated for the correspondence map between the image in Figure 3 and the next image in a 30 Hz video sequence. The total truth motion between the two frames was on the order of 0.5 pixels. The 95 percentile error for the Lucas-Kanade method, based on the histogram in Figure 4, was 0.4 pixels. The run-time of the Lucas-Kanade code was 77 times that of the code that performed the closed-form solution on every pixel.

The closed-form inertial based solution, in reality, is not error free. The dominant errors affecting the performance are the velocity error, the absolute attitude error, and the attitude error drift between consecutive video frames. References [8] and [9] investigate GPS velocity error. Both of these references predict errors that are less than 1 m/s RMS. Reference [4] develops an integrated GPS/Inertial filter using a low-cost Crossbow AHRS-DMU-HDX inertial measurement unit (IMU) which has an empirically measured attitude error of 0.04 degrees in the roll and pitch attitude axes and 0.36 degrees in yaw. The preceding numbers are one-sigma values.

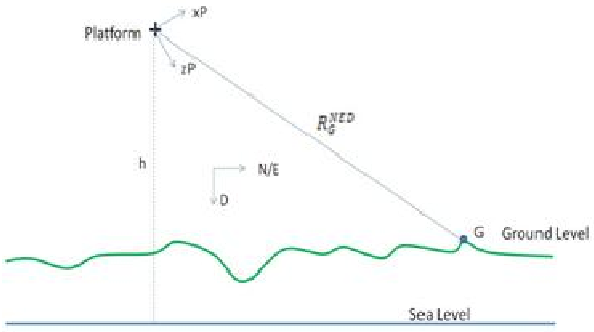

Figure 2 – Range Projection to Ground.

Figure 3: Simulated Aircraft Imagery

Additionally, the Crossbow IMU has an angular readout noise of $8.5e^{-2}$ degrees / second (0.0028 degrees over a 1/30 second frame). Placing these quantities into a 100 run Monte-

Carlo simulation shows a 95 percentile error of 0.04 pixels for the closed-form inertial based solution (see Figure 5).

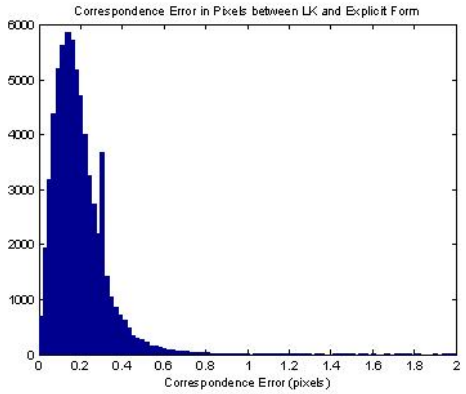

Figure 4: Lucas-Kanade Correspondence Map Error Histogram

The Monte-Carlo is executed by making a random draw of the error sources for each run and using them to perturb equation (5). The error form of equation (5) is created by making the following replacements

$$[T_{NED}^{P}]_k \leftarrow [D1][T_{NED}^{P}]_k$$
$$[T_{NED}^{P}]_{k+1} \leftarrow [D2][D1][T_{NED}^{P}]_{k+1}$$
$$\Delta R \leftarrow \Delta R + D_v \Delta t$$

where the random disturbance matrices $D1$, $D2$ and the disturbance vector D_v are given by,

$$[D1] = I - \frac{\pi}{180}[0.04r_1 \quad 0.04r_2 \quad 0.36r_3]_x$$
$$[D2] = I - \frac{\pi}{180}[0.0028r_4 \quad 0.0028r_5 \quad 0.00286r_6]_x$$
$$D_v = (1\frac{m}{s})[r_7 \quad r_8 \quad r_9]^T$$

where the subscript "x" operator was defined in equation (7). The values r_1 through r_9 are independent draws from a zero-mean, unity variance normal distribution. The numerical values above are based upon the specific inertial sensor and GPS error parameters discussed previously.

6. CONCLUSION

This paper has introduced a computationally efficient means of calculating a dense correspondence vector field for a video sequence in airborne applications where an inertial navigation sensor is available. The method bypasses computationally expensive image processing methods of estimating the vector field and, instead, uses a closed form solution to the geometric mapping from the inertial sensor measurements to the image. Furthermore, the paper outlines an approach to online estimation of the synchronization and misalignment between the inertial and image sensors. Accuracy of these parameters is required for making sub-pixel measurements of the correspondence vector field. Simulation based results show that a typical low-cost GPS/inertial sensor system is able to measure the correspondence an order of magnitude faster than a typical image-based optical flow code while achieving an order of magnitude improvement in accuracy.

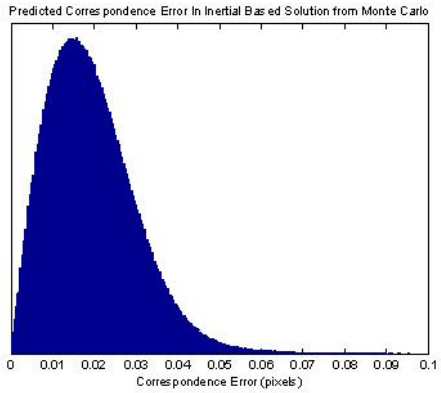

Figure 5: Monte-Carlo Based, Inertial Based Correspondence Error Histogram

REFERENCES

[1] A. Forsyth and Ponce, *Computer Vision: A Modern Approach*. Prentice Hall, Upper Saddle River, New Jersey, 2003.

[2] R. Schultz and L. Stevenson, "Extraction of High-Resolution Frames from Video Sequences," *IEEE Transactions on Image Processing*, vol. 5, no. 6, pp. 996-1011, June 1996.

[3] Siong, Mokri, Hussain, Ibrahim, Mustafa, "*Motion Detection Using Lucas Kanade Algorithm and Application Enhancement*," 2009 International Conference on Electrical Engineering and Informatics, Selangor, Malaysia, pp. 537-542.

[4] Hide, Moore, and Smith, "Adaptive Kalman Filtering Algorithms for Integrating GPS and Low Cost INS," *Position, Location, and Navigation Symposium*, 2004, pp. 227-233

[5] Performance Specification Digital Terrain Elevation Data (DTED), MIL-PRF-89020B, May 2000

[6]http://www.spatialenergy.com/images/SE_Bourtange_24x36.pdf

[7]http://www.cs.ucf.edu/vision/public_html/source.html#Optical%20Flow

[8] Herbert, Keith, Ryan, Lachapelle, and Cannon, "*DGPS Kinematic Carrier Phase Signal Simulation Analysis for Precise Aircraft Velocity Determination*", Proceedings of the ION Annual Meeting, Albuquerque, NM, June 3 – July 2 1997

[9] Serrano, Kim, and B. Langley, "A GPS Velocity Sensor: How Accurate Can it Be? – A First Look", ION NTM 2004, San Diego, CA, 26-28 January 2004

[10] A.K. Katsaggelos, R. Molina, and J. Mateos, "Super Resolution of Images and Video", Morgan and Claypool Publishers, 2007