Incorporating structure from motion uncertainty into image-based pose estimation

Ben T. Ludington[*a], Andrew P. Brown[a], Michael J. Sheffler[a], Clark N. Taylor[b], Stephen Berardi[c]
[a]Toyon Research Corporation, 6800 Cortona Drive, Goleta, CA USA 93117; [b]Air Force Research Laboratory WPAFB, OH USA 45433; [c]Northrop Grumman Corporation 21240 Burbank Boulevard, Woodland Hills, CA USA 91367

ABSTRACT

A method for generating and utilizing structure from motion (SfM) uncertainty estimates within image-based pose estimation is presented. The method is applied to a class of problems in which SfM algorithms are utilized to form a geo-registered reference model of a particular ground area using imagery gathered during flight by a small unmanned aircraft. The model is then used to form camera pose estimates in near real-time from imagery gathered later. The resulting pose estimates can be utilized by any of the other onboard systems (e.g. as a replacement for GPS data) or downstream exploitation systems, e.g., image-based object trackers. However, many of the consumers of pose estimates require an assessment of the pose accuracy. The method for generating the accuracy assessment is presented.

First, the uncertainty in the reference model is estimated. Bundle Adjustment (BA) is utilized for model generation. While the high-level approach for generating a covariance matrix of the BA parameters is straightforward, typical computing hardware is not able to support the required operations due to the scale of the optimization problem within BA. Therefore, a series of sparse matrix operations is utilized to form an exact covariance matrix for only the parameters that are needed at a particular moment. Once the uncertainty in the model has been determined, it is used to augment Perspective-n-Point pose estimation algorithms to improve the pose accuracy and to estimate the resulting pose uncertainty.

The implementation of the described method is presented along with results including results gathered from flight test data.

Keywords: Structure from motion, bundle adjustment, pose estimation, uncertainty

1. INTRODUCTION

This paper discusses an approach for generating a three dimensional scene model using structure-from-motion (SfM) approaches for an airborne electro-optic (EO) camera and then later using the model for image geo-registration. Such an approach is useful in a scenario in which a vehicle overflies a particular region during a portion of the mission and then returns to the region later in the mission or during another mission. By utilizing the model generated during the first flight over the area, the vehicle can navigate over the reference area without a significant degradation in navigational accuracy.

A sample scene model is depicted in Figure 1. A subset of the features that make up the model is shown as dots in the image, and the color of the dot represents the estimated elevation. In this case, this imagery was gathered during an orbit of the region of interest. When the vehicle returns to this modeled area, the modeled features are used for navigation and to improve the geo-registration of the gathered imagery.

In such an approach, sensor and navigation errors that are present during model generation cause errors in the resulting models. These model errors then degrade the resulting camera poses that are estimated using the model. To reduce the effects of these errors, the model errors must be estimated, and the pose estimation method must incorporate these estimates. This paper describes approaches for these two key components of the system.

*bludington@toyon.com; phone 1 703 674-0612; fax 1 703 674-0616; www.toyon.com

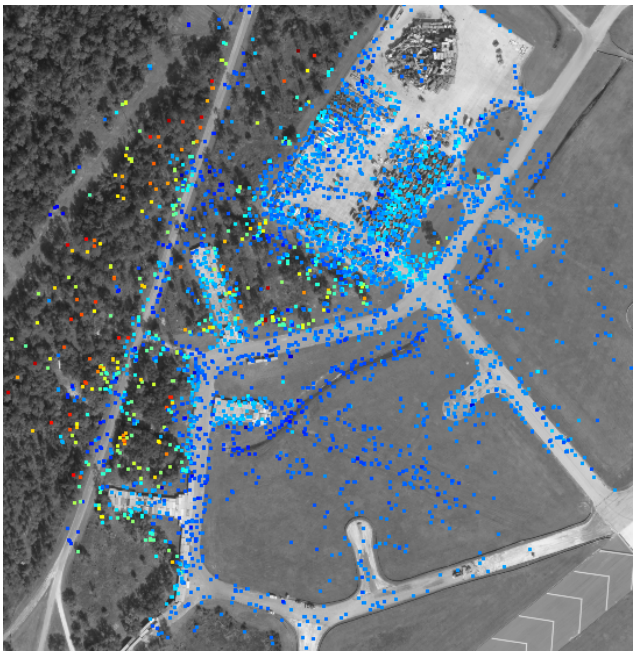

Figure 1. Sample model overlaid on USGS imagery. A subset of the modeled points are shown using the colored dots. The colors represent estimated elevation. Dark blue dots represent 14.5 m MSL, while dark red dots represent 40.5 m MSL.

The remainder of this paper is organized as follows. First, an overview of the model generation and pose estimation approaches are given. Then, brief introductions are provided for error estimation within scene modeling and pose estimation. This is followed by discussion of an approach that is used to perform the error calculation on typical computing hardware along with an overview of the typical resulting performance.

2. APPROACH OVERVIEW

The approach is broken into two distinct steps as shown in Figure 2. First, imagery and telemetry are collected during the first flight over a region of interest. *A priori* knowledge, such Digital Terrain Elevation Data (DTED) or other digital elevation maps (DEM), are also ingested. The available information is used to form a reference model of the region using bundle adjustment (BA)[1]. The model, which consists of the three-dimensional positions of the features, is stored in memory. When the vehicle returns to the region, features are recalled from the model and matched. The modeled positions of the matched features are utilized to determine the position and orientation of the camera using the perspective-n-point (PnP) algorithm[1,2].

Model Generation

Within the model generation algorithm, perspective-corrected image features are extracted and tracked between frames. In general, several thousand features are extracted from each frame and matched with time-adjacent frames. The collection of feature tracks is then combined with the DEM to auto-calibrate the sensor internal (*e.g.* focal length, pixel spacing, radial lens distortion) and external (sensor angles and position offsets relative to the inertial measurement unit (IMU) coordinate frame) parameters[3]. Once these calibration parameters have been found, they are held constant for the remainder of the processing assuming the physical parameters remain fixed.

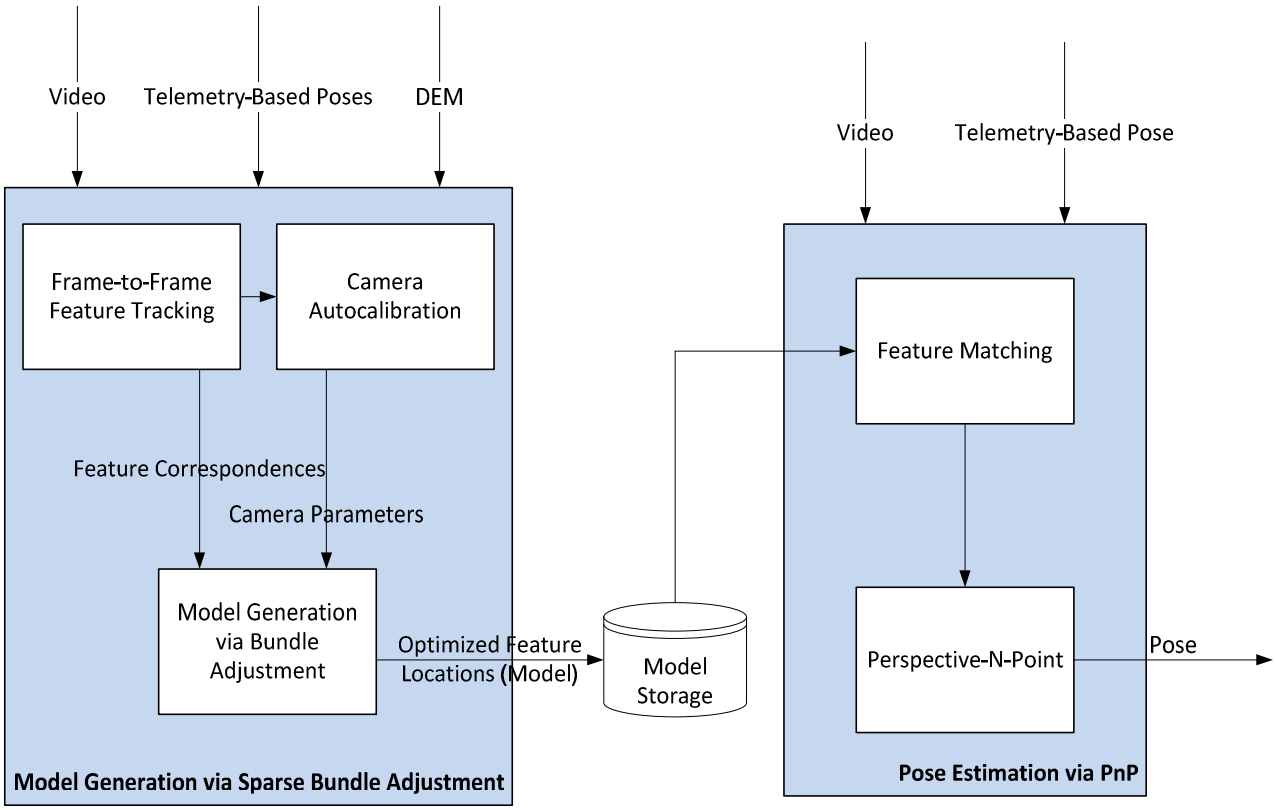

Figure 2. Overview of model generation and pose estimation modules.

After feature track outlier rejection and auto-calibration, BA is utilized to estimate the positions, attitudes and headings of the camera during data collection as well as the three-dimensional positions of the feature points. At the core of BA is a nonlinear optimization that finds camera pose and feature position parameters to minimize the total reprojection error. Assuming a Euclidean camera model, the mapping between world points to camera points can be written as

$$\lambda x = K[R\ t]X \tag{1}$$

where X is a 4×1 vector representing a world point projecting on an image point, x, which is represented by a 3×1 vector; λ is an arbitrary scale factor; K is a 3×1 matrix of the camera's intrinsic calibration parameters; R is an orthogonal rotation matrix, and t is a 3×1 vector which, along with R, encodes the position of the camera. During BA processing, an optimization is executed to estimate R, t, and X to minimize the difference between x and $\hat{x}$ in a least squares sense, where $\hat{x}$ is formed from the estimated parameters and x are the feature locations measured within the images. In other words, given an assumed function, $\hat{x} = f(p)$, which maps a set of parameters p (which are used to construct estimates of R, t, and X), BA finds the best p that minimizes the squared distance cost function, $(x - \hat{x})^{\mathrm{T}}(x - \hat{x})$.

BA typically utilizes a Levenberg-Marquardt (LM) algorithm, which is an iterative nonlinear optimization method that combines the best features of steepest-descent and the Gauss-Newton methods[4, 5]. The method assumes that $f(p)$ is locally linearizable such that

$$f(p + \delta p) \approx f(p) + J\delta p \tag{2}$$

Assuming a starting estimate, p_0, which may be generated from telemetry and DEM intersections, the LM algorithm solves the following equation at each iteration to update the estimate of p toward the optimal value.

$$(J^{\mathrm{T}}J + \mu I)\delta p = J^{\mathrm{T}}(x - \hat{x}) \tag{3}$$

where μ is the damping term and is used to balance the speed of the optimization with the accuracy of solution. At each iteration, the damping is changed until a value of δp is found that decreases the cost function.

In the case of interest, p is made up of the pose parameters and the positions of the features. Since the pose parameters for each camera are independent from one another and the measurements of features are independent from one another, the Jacobian, J becomes sparse. Through this sparseness, the normal equations may be solved for large numbers of features and views using common computing equipment.

Pose Estimation

Once the model is generated, it may be used for pose estimation when the camera returns over the modeled region. First, features in the camera frame are matched with features in the model. After matching, the feature locations are utilized within perspective-N-point (PnP) processing to generate a pose. Several PnP methods have gained popularity[1,2], and they typically use an optimization to minimize the reprojection error assuming a particular model. In our case, we again utilize a Euclidean camera model. Therefore, the optimal R and t are found by minimizing

$$\min_{R,t}(x-\hat{x})^{\intercal}(x-\hat{x})\ s.t.\lambda\hat{x}=K[R\ t]X \tag{4}$$

In general, the popular PnP methods assume that all points are known equally well and that all measurements are equally accurate. In our case, the errors in X not only vary, but are also correlated. The next section discusses how the covariance of this error is calculated and how it is utilized within an augmented PnP process to produce a better representation of the cost function.

3. UNCERTAINTY ESTIMATION

Like the processing flow described above, the uncertainty estimation takes place in two steps. First, the uncertainty inherent in the inputs to the model generation algorithm is utilized to determine the resulting uncertainty in the model. Then, the estimated uncertainty in the feature positions within the model is used within pose estimation to form a more accurate estimate and to generate an estimate of the resulting pose. The model uncertainty is captured through two components. The first component captures the uncertainty in the auto-calibration. This term is calculated by evaluating the sensitivity of each parameter to slight variations in the inputs such as the telemetry generated by an onboard GPS/INS. Therefore, the telemetry errors are modeled to ensure the magnitude and correlation is adequately captured when calculating this error term. The covariance due to this term is fixed for all cameras and can be mapped to a feature position covariance through the Jacobians. In general, it is uncorrelated from the other terms. Therefore, it is left out of the following equations for simplicity of notation. The second component captures the uncertainty within BA. Assuming the optimization within BA has reached a global minimum, the resulting uncertainty in the parameters is

$$\Sigma_p = (J^{\intercal}\Sigma_x^{-1}J)^{\dagger} \tag{5}$$

where Σ_p is the covariance of the parameters and Σ_x is the covariance of the measurements. For convenience and to facilitate ease of processing, the parameter list is typically constructed of two groups: group a is made up of the camera pose elements, while group b is made up of the feature positions. Therefore,

$$J = \left[\frac{\partial f}{\partial a}\ \frac{\partial f}{\partial b}\right] = [A\ B] \tag{6}$$

and

$$\Sigma_p = \begin{bmatrix}\Sigma_a & \Sigma_{ab}\\ \Sigma_{ba} & \Sigma_b\end{bmatrix} \tag{7}$$

Expressions for the blocks of Σ_p are[1]

$$\begin{aligned}\Sigma_a &= (U - WV^{-1}W^{\intercal})^{\dagger}\\ \Sigma_b &= Y^{\intercal}\Sigma_a Y + V^{-1}\\ \Sigma_{ab} &= -\Sigma_a Y\end{aligned} \tag{8}$$

Where

$$\begin{aligned} U &= A^{\intercal}\Sigma_x^{-1}A \\ V &= B^{\intercal}\Sigma_x^{-1}B \\ W &= A^{\intercal}\Sigma_x^{-1}B \\ Y &= WV^{-1} \end{aligned} \tag{9}$$

Therefore, the measurement covariance can be mapped to a parameter covariance matrix through the Jacobians. To determine the uncertainty in the features currently viewed by a camera, only the portion of Σ_b corresponding to features in the view are needed. This portion of the feature position covariance matrix will be denoted as $\Sigma_{b'}$ and is discussed in the next section.

Once $\Sigma_{b'}$ is available, it may be utilized to modify the cost function within PnP such that the resulting optimization matches features that have low positional uncertainty better than the features that may have higher uncertainties. The cost function becomes

$$\min_{R,t}(x-\hat{x})^{\intercal}(B'\Sigma_{b'}B'^{\intercal})^{-1}(x-\hat{x})\ s.t.\ \lambda\hat{x} = K[R\ t]X \tag{10}$$

where B' maps the uncertainty in position errors to uncertainty in measurements for the observed features.

The method for finding the pose parameters and the uncertainty in the parameters is discussed in the next section.

4. UNCERTAINTY ESTIMATION IMPLEMENTATION METHODS

Many of the modeling problems of interest utilize wide field-of-view video frames captured while orbiting a large region. Therefore it is not uncommon to obtain measurements for several hundreds of thousands of features. In such cases, performing the operations of Equation (8) may not be feasible on typical processing hardware. Under these circumstances, the sparseness of the two blocks of the Jacobian (Equation (6)) is utilized to enable computation of the blocks of the various matrices needed within Equation (8).

An overview of this processing is shown in Figure 3 along with a depiction of the sparseness of the various matrices. First, the elements along the block diagonal of U and V^{-1} are found. Since V is block diagonal with 3×3 blocks along the diagonal, a series of 3×3 matrices are inverted instead of inverting the entire V matrix. By utilizing the association between cameras and features through the measurements, the blocks of W can also be computed easily. Once the blocks of U, V^{-1}, and W have been found, Y can be found by multiplying the blocks of W by the corresponding blocks of V^{-1}. Again, by utilizing the association between the cameras and the features through the measurements, the blocks of $U - WV^{-1}W^{\intercal}$ are computed efficiently. This matrix is $6m \times 6m$, where m is the number of cameras within BA (typically150-200 cameras are utilized for the problems of interest). Once the blocks of this matrix are found, they are used to populate the matrix, and the pseudoinverse is taken. The approximate size of this matrix is 1000×1000 for our typical problems. The pseudoinverse operations typically require 10 seconds or less using commonly available desktop computing hardware. The previously discussed sparse matrix operations can typically be performed in a second or less on common hardware. After Σ_a and the blocks of V^{-1} and Y are found, they are stored along with the model to be used for pose estimation.

When PnP is performed, features are selected from the model and matched with features in the current sensor field of view. Sparse matrix operations are performed to form

$$\Sigma_{b'} = \Gamma(Y^{\intercal}\Sigma_a Y + V^{-1})\Gamma^{\intercal} \tag{11}$$

where Γ is a selection matrix composed of zeros and one. Once $\Sigma_{b'}$ has been found, it is used to modify the PnP cost function as described in Equation (10). To implement Equation (10), non-weighted PnP functions[1,2] may be utilized to form a linearization point. Once this point is found, the residual reprojection error for each feature is calculated and used to update the estimate of R and t. The update is calculated by finding the Jacobian that captures the change in reprojection error due to change in pose, which is denoted A, and then taking a weighted pseudoinverse:

$$\delta = (A^{\mathsf{T}}(B'\Sigma_{b'}B'^{\mathsf{T}})^{-1}A)^{-1}A^{\mathsf{T}}(B'\Sigma_{b'}B'^{\mathsf{T}})^{-1}(\hat{x} - x) \tag{12}$$

where δ is the update to the parameters, $\hat{x}$ is calculated using the R and t of the linearization point, x is the measurement vector, and A is the Jacobian that captures the change in measurements due to changes in pose parameters. Equation (12) may be solved iteratively if the Jacobian varies significantly or if the non-weighted solution is too far from the optimal, weighted solution.

The resulting covariance in the parameter vector is then

$$\Sigma_{a'} = (A^{\mathsf{T}}(B'\Sigma_{b'}B'^{\mathsf{T}})^{-1}A)^{\dagger} \tag{13}$$

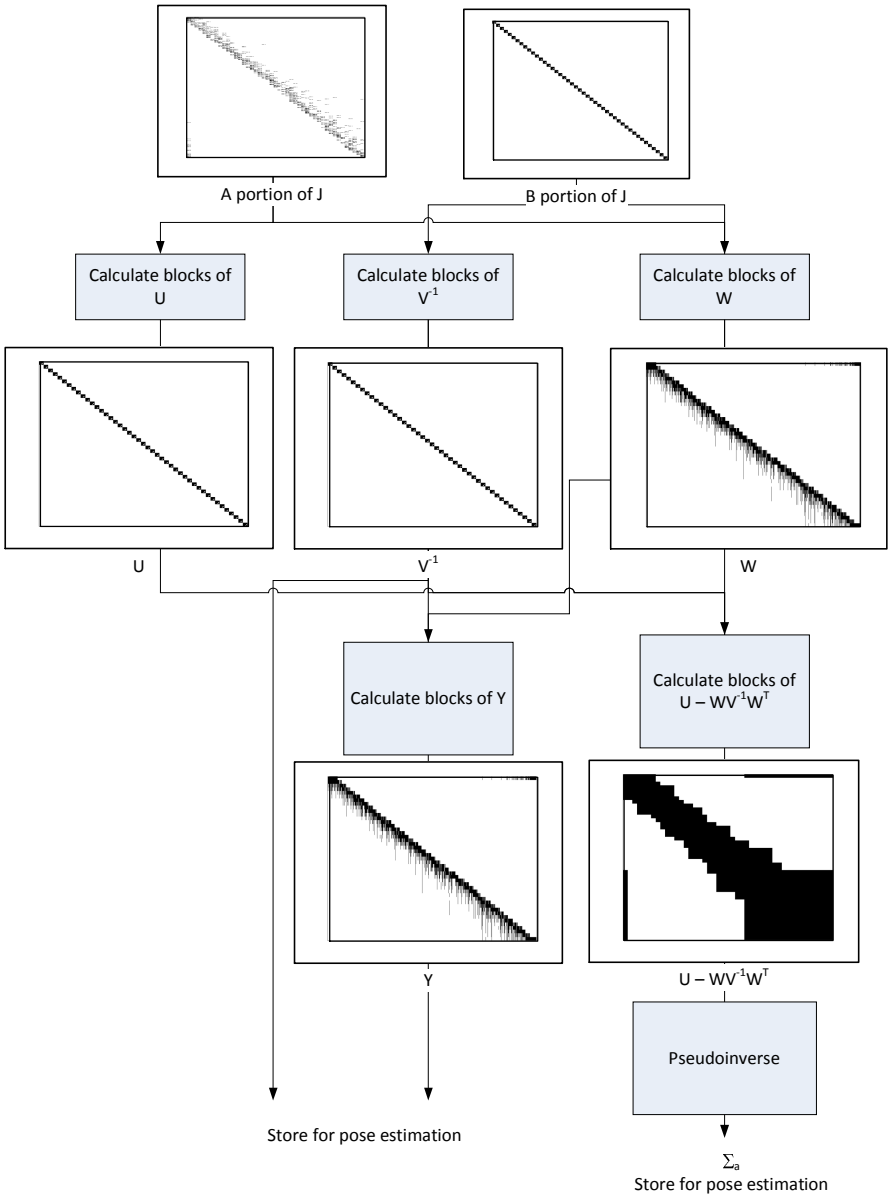

Figure 3. Sparse matrix processing utilized to find the building blocks of the feature position covariance matrix. The sparseness of the various matrices is depicted by showing the portions of the matrices that are zeros with white while showing the nonzero portions with black.

5. RESULTS

The processing described in this paper was implemented and used to process data gathered during the flight of a small unmanned aerial vehicle over terrain similar to the image shown in Figure 1. To measure the performance of the algorithm, the imagery gathered when the vehicle returned to the modeled region was projected onto a DEM using the position and orientation generated from PnP as shown in Figure 4. The covariance of the camera positions and orientations were also generated. The accuracy and statistical consistency of the PnP-based geo-projection results were then evaluated using analyst-measured geo-projection errors for visual landmarks which were easily recognizable and accurately localizable by the analyst. The error and the covariance are shown in Figure 5, which shows statistical consistency between the error and estimated uncertainty.

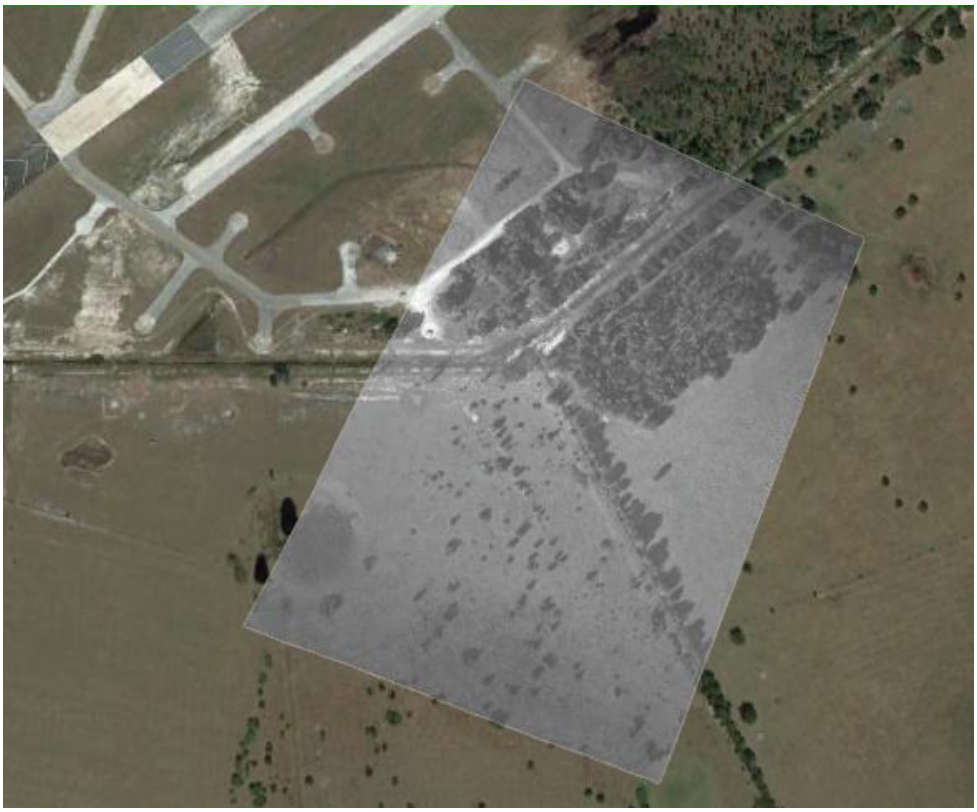

Figure 4. Geo-projected image on reference imagery for comparison. The output of PnP was utilized when constructing the projection.

The pose calculated during PnP processing was also compared to an onboard sub-tactical-grade GPS/INS. The difference in orientation is shown in Figure 6 and Figure 7. Again, the statistical consistency is apparent form the plots. Furthermore, the resulting position errors are similar to the accuracies normally seen when using GPS signals. However, during PnP processing, no GPS information is utilized. Therefore, the approach can act as either a replacement or supplement to GPS measurements.

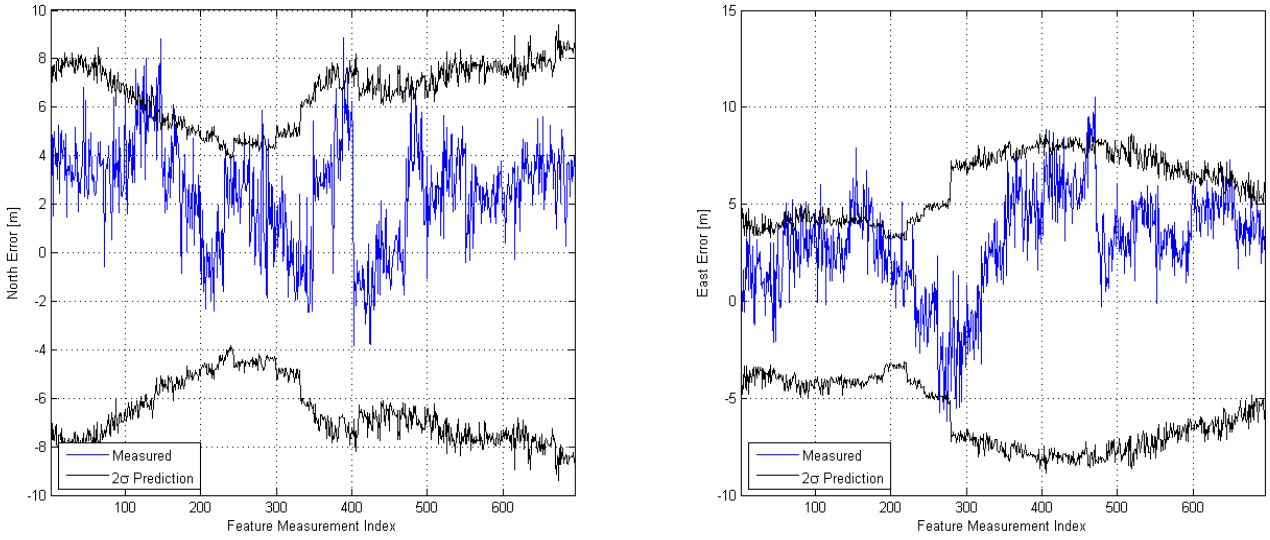

Figure 5. Comparison between geoprojected feature position and feature position in reference images. The method described above was used to calculate the position and orientation of the cameras used for projection. The difference in position is shown in blue, and the $\pm 2\sigma$ from the associated covariance operations are shown in black.

6. CONCLUSIONS

The paper presented an approach for utilizing imagery gathered during a flight over a specified region to construct a model of the reference terrain. The model was later used to estimate poses of a camera when the vehicle returned to the model region. The approach has been augmented to include estimating the covariance of the model components, the utilization of the covariance within PnP processing, and the construction of a pose covariance matrix from PnP. As presented in Figure 5 through Figure 7, the estimated uncertainty agrees well with the measured errors, and the resulting position errors approach the levels normally obtained using GPS.

7. ACKNOWLEDGEMENTS

Much of this work was performed under the Air Force Maintain Accurate Geo-Registration via Image-Nav Compensation (MAGIC) program. The authors thank the program and the Air Force Research Laboratory and Northrop-Grumman program managers, J.C. Ha and Han Park, for ongoing support in developing these algorithms.

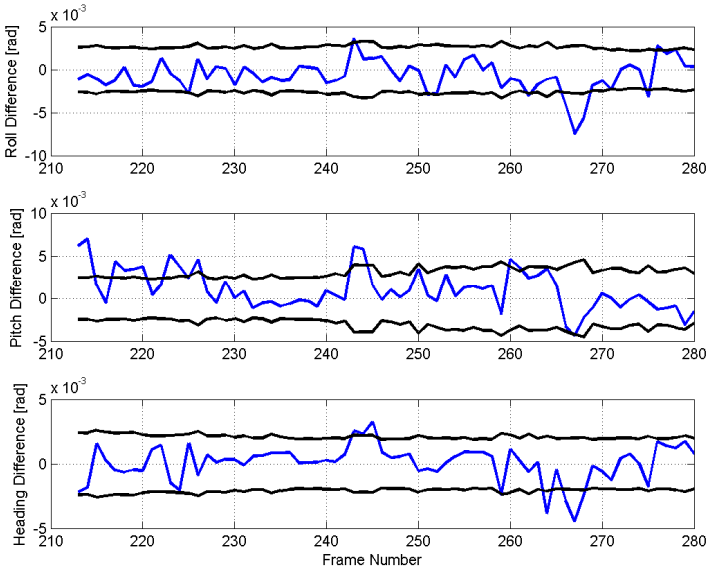

Figure 6. Comparison between PnP orientation and GPS/INS orientation. The difference is shown in blue. The estimated $\pm 2\sigma$ from the covariance processing is shown in black. The errors of the GPS/INS were not included in this analysis.

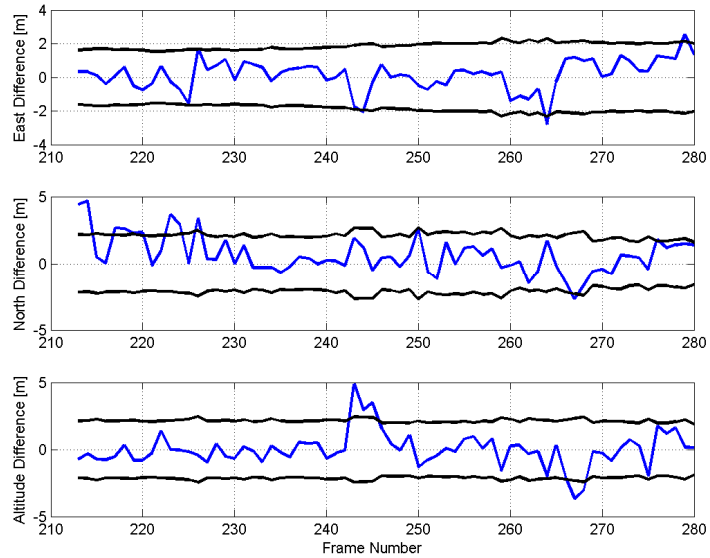

Figure 7. Comparison between PnP orientation and GPS/INS position. The difference is shown in blue. The estimated $\pm 2\sigma$ from the covariance processing is shown in black.

REFERENCES

[1] Hartly, R. and Zisserman, A., [Multiple View Geometry in Computer Vision], Cambridge University Press New York (2000).

[2] Lepetit, V., Moreno-Norguer, F. and Fua, P., "EPnP: An accurate O(n) solution to the PnP problem," International Journal of Computer Vision 81(2), 156-166 (2009).

[3] Brown, A. P., Sheffler, M. J. and Dunn, K. E., "Persistent electro-optical/infrared wide-area sensor exploitation," Proc. SPIE Conference on Evolutionary and Bio-Inspired Computation: Theory and Application IV (2012).

[4] Lourakis, M. I. A. and Argyros, A. A., "SBA: A Software Package for Generic Sparse Bundle Adjustment," ACM Transactions on Mathematical Software 36(1) 2:1-2:30 (2009).

[5] Zach, Christopher, SSBA-4.0 software repository, https://github.com/chzach/SSBA.