Electronic Image Stabilization using Optical Flow with Inertial Fusion

Michael J. Smith
354 Civil Engineering Squadron
Eielson AFB, AK 99702
Email: michael.smith-02@eielson.af.mil

Alexander Boxerbaum
Department of Mechanical and
Aerospace Engineering
Case Western Reserve University
Cleveland, OH 44106
Email: alexander.boxerbaum@case.edu

Gilbert L. Peterson
Air Force Institute of Technology
Wright-Patterson AFB, OH 45433
Email: gilbert.peterson@afit.edu

Roger D. Quinn
Department of Mechanical and
Aerospace Engineering
Case Western Reserve University
Cleveland, OH 44106
Email: rdq@case.edu

***Abstract*—When a camera is affixed on a dynamic mobile robot, image stabilization is the first step towards more complex analysis on the video feed. This paper presents a novel electronic image stabilization (EIS) algorithm for highly dynamic mobile robotic platforms. The algorithm combines optical flow motion parameter estimation with angular rate data provided by a strapdown inertial measurement unit (IMU). A discrete Kalman filter in feedforward configuration is used for optimal fusion of the two data sources. Performance evaluations are conducted using a simulated video truth model (capturing the effects of image translation, rotation, blurring, and moving objects), and live test data. Live data was collected from a camera and IMU affixed to the DAGSI Whegs mobile robotic platform as it navigated through a hallway. Template matching, feature detection, optical flow, and inertial measurement techniques are compared and analyzed to determine the most suitable algorithm for this specific type of image stabilization. Pyramidal Lucas-Kanade optical flow using Shi-Tomasi good features in combination with inertial measurement is the EIS algorithm found to be superior. In the presence of moving objects, fusion of inertial measurement reduces optical flow root-mean-squared (RMS) error in motion parameter estimates by 40%.**

I. Introduction

Electronic Image Stabilization (EIS) is the process by which undesired motion is digitally removed from a video feed. These are motion estimation, followed by motion compensation. In the motion estimation stage, the physical movement of the video image pixels between image frames are defined within the parameters of a particular motion model. Once a set of parameter estimates is found frame to frame, compensation is applied to counteract the perceived motion. this creates a stable image feed.

The motivation for this work is non-GPS navigation [1] on biologically inspired robotic platforms. Platforms like the DAGSI Whegs [2], shown in Figure 1, are capable of traversing stairs, large rocks, and other difficult terrain. This versatile mobility incurs costs upon imaging sensors affixed to the platform. Large image displacement frame to frame, image blurring, and moving objects in the scene make reliable motion parameter estimates difficult to achieve. Thus a robust EIS algorithm is necessary to perform accurate navigation estimation in the presence of these effects.

Fig. 1. The DAGSI Whegs Robot and Camera/IMU Setup.

This paper presents a novel EIS algorithm designed to operate in the presence of large image displacement, image blurring, and moving objects. Using the similarity motion model, the algorithm fuses pyramidal Lucas-Kanade optical flow using Shi-Tomasi good features with inertial measurement motion estimation by way of a discrete Kalman filter. Inertial measurement motion estimation is performed by summing angular displacements between frames of a MIDG II inertial measurement unit (IMU) and multiplying the angular displacements by a constant. The two motion estimates are then optimally fused using a nine-state discrete Kalman filter.

A simulated video truth model and live test data are used to compare the performance of several EIS algorithms. The effects of image translation, rotation, blurring, and moving objects are captured. Using the novel optical flow with inertial fusion algorithm results in 40% less RMS error than the best alternative algorithm. A video showing unstabilized and stabilized performance is included.

II. Background and Related Work

Images are described in raster coordinates, (m, n), and normalized Cartesian coordinates, (x, y), as shown in Figure 2. Aspect ratio is described as

$$\text{aspect ratio} = \frac{1}{\mathsf{a}} = \frac{\mathrm{H}}{\mathrm{W}} \tag{1}$$

where a is the inverse of the aspect ratio, H is the height of the image, and W is the width [4].

Given an image pixel coordinate (m, n), the normalized Cartesian coordinate pair is

$$x = \frac{2m - \mathrm{W}}{\mathrm{S}} \quad \text{and} \quad y = \frac{2n - \mathrm{H}}{\mathrm{S}} \tag{2}$$

where $\mathrm{S} = \max(\mathrm{W}, \mathrm{H})$ [4]. The (x, y) coordinate pair is referred to in matrix form as c, where

$$\mathsf{c} = \begin{bmatrix} x \\ y \end{bmatrix}. \tag{3}$$

The homogeneous representation for an untransformed image point (x, y) is $\tilde{\mathsf{c}}$, where

$$\tilde{\mathsf{c}} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}. \tag{4}$$

The similarity transform [4], which describes image motion involving translation, rotation and scale, is defined as

$$\mathsf{c}^{+} = \begin{bmatrix} \lambda \mathbf{R} & \mathsf{T} \end{bmatrix} \tilde{\mathsf{c}}^{-} \tag{5}$$

where

$$\mathsf{T} = \begin{bmatrix} \mathsf{T}_x \\ \mathsf{T}_y \end{bmatrix} \tag{6}$$

is the pixel translation in the x and y directions, λ is the scale factor for the transform, and the rotation matrix $\mathbf{R}$ is defined as

$$\mathbf{R} = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{bmatrix}. \tag{7}$$

The α parameter is the rotation angle of the image. The sign of the angle α is positive in the counter-clockwise direction. The $-$ and $+$ superscripts refer to the pixel coordinates before and after the transformation, respectively.

The coordinate frame for the system is shown in Figure 3. Using Euler angular representation, the ϕ, θ, and ψ angular directions follow the right hand rule and coincide with the x, y, and z directions, respectively. Observe that in the video feed, positive ϕ displacement generates negative α image rotation, positive θ displacement generates negative m pixel movement, and positive ψ displacement generates positive n pixel movement.

Far field scenes are assumed, which means that perceived scene movement is attributed to camera rotation only, and that camera translation is of negligible significance.

EIS algorithms can be categorized into four main techniques: template matching, feature detection, optical flow, and inertial measurement. In template matching techniques, a small template image is copied out of the image, and this location

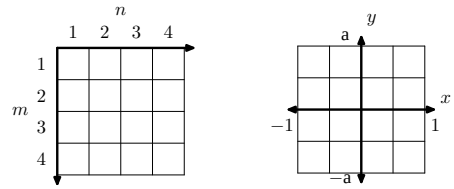

Fig. 2. Raster (left) and normalized Cartesian (right) coordinates.

is stored. On the next image, the template is translated and rotated until a suitable match is found. The template movement required for the match is the image displacement [3,5,6].

Feature detection techniques calculate frame displacement by tracking prominent and unique image features in successive frames [8,9,10,11]. In the previous frame, a feature pixel point is found, and a vector of directional gradients is formed as a descriptor. In the current frame, this descriptor vector is compared against known features until a match is found. This pair of matching features in two different images is called a correspondence. Once several correspondences have been found, the transformation matrix can be computed using a least squares approach or an algorithm such as random sample and consensus (RANSAC) [7].

Optical flow techniques determine image displacement by calculating the spatial derivatives of the image at certain pixel locations and the image time derivative. both sparse and dense forms can be used. Sparse optical flow, in the form of the Lucas-Kanade algorithm [12], is used in [13] and [14].

The final class of EIS methods is inertial measurement [15,16]. In this technique, an IMU is used to provide angular rate data, and these values are summed between image capture times. A coefficient is then multiplied by the angular displacement output by the IMU in between frames, and this results in the motion parameter values.

III. Optical Flow with Inertial Fusion

Template matching, feature detection, and optical flow methods all generate motion estimates directly from the video feed. Inertial measurement does not require a video feed, and is invariant to image anomalies like blurring and moving objects. Optimally combining inertial measurement with an image-based estimate will thus result in a better estimate than one method alone.

In developing the novel optical flow with inertial fusion algorithm, representative algorithms of the four main classes of

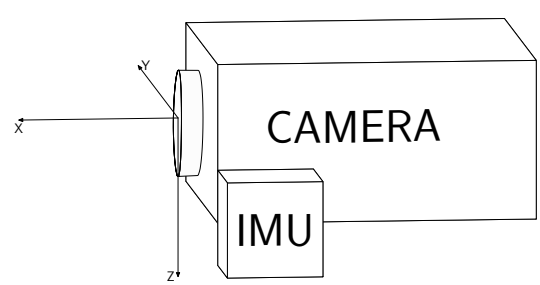

Fig. 3. Platform Coordinate Frame.

EIS are created. Template matching, feature detection, and optical flow are compared in order to determine the most suitable image based estimate for use with inertial measurement. These three algorithms are tested against a truth model to identify the technique that performs the best prior to inertial aiding. The optical flow algorithm is shown to perform best, and is then used in combination with inertial measurement in the final algorithm.

A. Evaluation of Non-Inertial Algorithms

In order to select the best non-inertial algorithm for use with inertial fusion, the non-inertial algorithms are presented and evaluated according to a video truth model.

1) Non-Inertial Algorithm Descriptions:

Template Matching Algorithm. Following the process in [3], the current image and the previous image are taken from the video feed. Six templates evenly spaced within the image, each of size 64×64 pixels, are copied out of the previous image. These templates are then decimated, and a four level template image pyramid is created. Similarly, the current image is decimated, and a four level image pyramid is created for the current image. Template searches are then performed across the levels of the pyramids. At the highest level, the search is constrained to 16 pixels, which amounts to a 128 pixel displacement for the original image. Each subsequent level then performs a more localized search, until the final match is achieved. The search is for the m and n pixel displacements which bring the correlation value as close to unity as possible.

Feature Detection Algorithm. Scale-invariant feature transform (SIFT) [18] is performed on both the previous image and the current image. Each feature has a particular pixel location and a directional gradient descriptor vector. Feature matching is performed by finding the descriptors which have the smallest inverse cosine of the dot product between them. RANSAC is then performed on these sets of correspondences, and the output is the most probable transformation matrix. From the transformation matrix, the m and n pixel displacements and the image angle α are found.

Optical Flow Algorithm. Shi-Tomasi good features [19] are found in the previous image, and then Lucas-Kanade optical flow is performed on the current image. This provides sets of feature correspondences between images, and RANSAC is performed. The m, n, and α displacements are then found from the transformation matrix.

2) Performance of Non-Inertial Algorithms:

A video truth model is developed in order to rate the anticipated performance of the image-based EIS algorithms on DAGSI Whegs. The truth model allows for the observation of the exact motion parameters between frames. Knowing these values, errors associated with a given EIS algorithm can be described.

The base image used for the truth model is a hallway image from the live test data, shown in Figure 4. The input parameters of the truth model generator function are the maximum values for rotation, θ, translation, T, blur angle, γ, and blur length, ξ. The function then generates random values for these parameters according to a uniform distribution from the negative maximum to the positive maximum. The function takes a large image, 1024×1280, and rotates it by a random θ. The rotated image is cropped to match the original 1024×1280 size. A window of size 512×640 is then taken from the center of the rotated image, plus the random horizontal and vertical translation values T. This allows for a simulation of m, n, and α for the duration of the video.

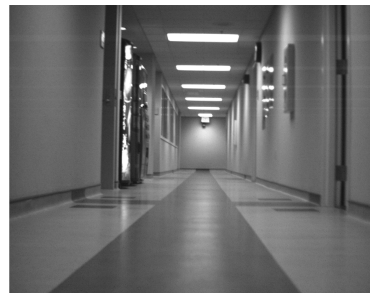

Fig. 4. Base Image for Truth Model

The effects of blur are captured by parameters γ, and ξ. The values of γ and ξ are input into a point spread function (PSF) which is applied to the image. When blurring is activated, each image is given a 50% probability of experiencing a blur.

The determination of error between the EIS parameter estimate and the true parameter value is accomplished by subtracting the estimated frame to frame displacement vector from the true frame to frame displacement vector. The residual vector is the estimation error for the duration of the test. Using the root-mean-squared (RMS) value of this error vector is effective to compare the overall performance of an EIS algorithm for a particular run.

Non-Inertial Results. Using the truth model, translation testing is performed upon each of the three non-inertial algorithms. The truth model is given a uniformly sampled 100 pixel maximum movement in the m and n directions, which approximates the actual maximum pixel movement for the live data set. The length of the video is 100 frames. The results of the test are shown in Table I. Note that template matching performs the worst, and both feature detection and optical flow perform near perfectly. Because template matching performs so poorly it is excluded from further analysis.

Next, the image undergoes a rotation of maximum $6°$ in addition to translation. Table II shows that both feature detection and optical flow perform well. In the next test, an additional blur function is added to the video. The maximum blur length is set to 100 pixels, and the maximum blur angle is set to $180°$. Table III shows that performance significantly

TABLE I. Translation Test Error.

	m RMS	n RMS	α RMS
Template Matching	27.67	63.84	N/A
Feature Detection	0.0	0.0	0.01°
Optical Flow	0.67	0.67	0.01°

TABLE II. Trans. and Rot. Test Error.

	m RMS	n RMS	α RMS
Feature Detection	4.88	4.67	0.01°
Optical Flow	4.76	4.69	0.01°

TABLE III. Trans., Rot., and Blur Test Error.

	m RMS	n RMS	α RMS
Feature Detection	34.97	36.98	1.47°
Optical Flow	4.99	5.19	0.21°

degrades for the feature detection algorithm, whereas optical flow maintains good accuracy. Finally, a moving object is injected into the video scene in the form of a 300×300 black box moving at constant speed from left to right. both algorithms face significant errors, as shown in Table IV. The hallway scene with moving object are shown in Figure 5.

Both optical flow and feature detection are effective for estimating large image displacements in translation and rotation. However, the optical flow algorithm is more robust against image blurring and is therefore selected for use in the final EIS algorithm.

B. Inertial Image Stabilization Algorithm

The optical flow method is now combined with the inertial measurement method. The algorithm follows the block diagram shown in Figure 6. Motion estimation at the current time requires both the current image frame and the last captured image frame. The good features are found using the Shi-Tomasi technique. Pyramidal Lucas-Kanade optical flow is then performed. The algorithm uses the find good features function and the pyramidal Lucas-Kanade optical flow function provided by the OpenCV library [21].

The track error vector is used with the number of feature correspondences to generate a detector value. The detector value approximates the error in the EIS method and is required for the Kalman filter to fuse the EIS and IMU results. The detector yields values close to unity when there is significant disturbance and differences between the current frame and the previous frame, and close to zero when there is great similarity

TABLE IV. Trans., Rot., Blur, and Moving Object Test Error.

	m RMS	n RMS	α RMS
Feature Detection	38.20	26.01	3.33°
Optical Flow	28.93	38.44	4.21°

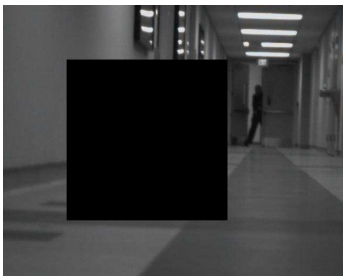

Fig. 5. Hallway With Moving Object Injection.

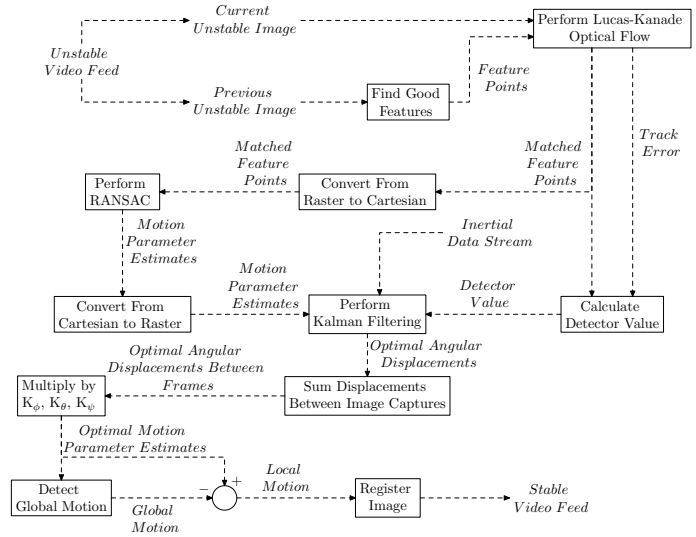

Fig. 6. Optical Flow with Inertial Fusion Block Diagram.

between frames. The equation used to generate the detector value is

$$\mathrm{D(T,C)} = (\frac{\mathrm{T}}{1000})(1 - \frac{\mathrm{C}}{500}) \tag{8}$$

where T is the average track error for the frame, and C is the number of correspondences. The track error ratio, $\frac{\mathrm{T}}{1000}$, and the correspondence ratio, $(1 - \frac{\mathrm{C}}{500})$, are multiplied to calculate the detector value. The number 1000 is used in the track error ratio because the maximum track error value during testing was approximately 1000. The value of 500 was selected for the correspondence ratio because the maximum number of correspondences allowed in the OpenCV pyramidal Lucas-Kanade optical flow algorithm is 500. Multiplying the two ratios, the final detector value gives insight into the reliability of the motion parameter estimates found by optical flow.

Note that even though the detector value is correlated to actual performance of the stabilizer, a high detector value does not necessarily mean features could not be accurately matched. It states only that there is significant pixel intensity difference between the image.

The vectors of matched points in the image, output by the Lucas-Kanade algorithm in raster coordinates, are converted into Cartesian coordinates in order for RANSAC determination of the transformation matrix. The motion parameter values are then extracted from the transformation matrix. The x, y, and α displacements are then fused with gyro sensor data from the inertial device to generate the optimal motion estimate. This fusion is accomplished by way of Kalman filtering.

Kalman Filter Development. The nine state discrete Kalman filter uses a perturbation model to estimate the IMU output errors. The IMU output errors are characterized and modeled as a first-order Gauss-Markov process.

The state vector for the system is shown in Equation (9). There are three angular displacement states, ϕ, θ, and ψ, three angular rate states, p, q, and r, and three drift bias states b_ϕ, b_θ, and b_ψ. The translational states have negligible significance

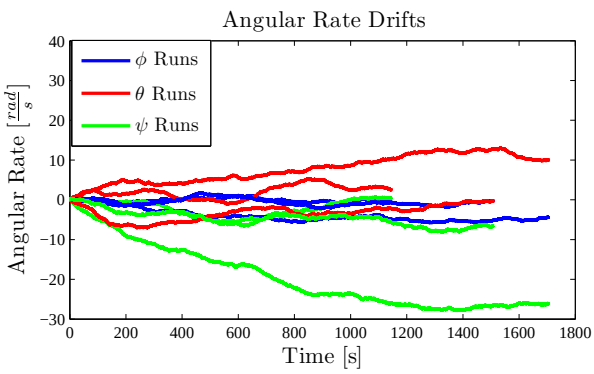

Fig. 7. Angular Rate Drift.

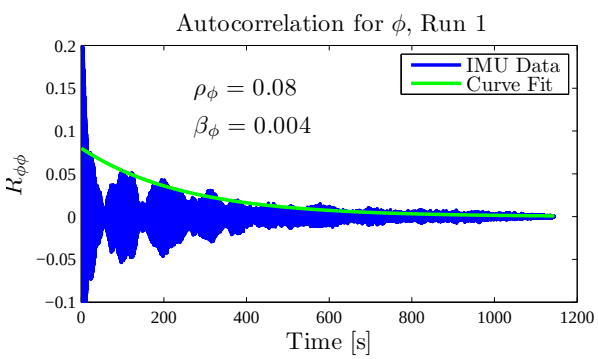

Fig. 8. Example Autocorrelation and Parameter Determination.

because of the far field assumption and are ignored.

$$\mathsf{x} = \begin{bmatrix} \phi & \theta & \psi & \mathsf{p} & \mathsf{q} & \mathsf{r} & \mathsf{b}_\phi & \mathsf{b}_\theta & \mathsf{b}_\psi \end{bmatrix}^{\mathrm{T}} \tag{9}$$

Characterization of IMU Error. The IMU errors affecting the angular outputs are approximated as a first-order Gauss-Markov process. This is described as a noise source whose autocorrelation function is of the form

$$\mathrm{R}(\tau) = \rho_e^{-\beta|\tau|} \tag{10}$$

and of differential form

$$\dot{\mathsf{n}}(t) = \frac{-1}{\tau}\mathsf{n}(t) + \mathsf{w}_\mathsf{n}(t) \tag{11}$$

where $\mathsf{n}(t)$ is the current value of the noise, τ is the time constant for the noise process, and $\mathsf{w}_\mathsf{n}(t)$ is zero-mean additive white Gaussian noise of strength σ^2. The time constant τ and β value are related by

$$\tau = \frac{1}{\beta} \tag{12}$$

To determine the values of ρ and β in Equation (10) for the IMU used, the device was left motionless on a table and three sets of data were collected, each of duration of at least 20 minutes. Each angular rate was affected by a random drifting bias. These drifts are shown in Figure 7, which were arrived at by cumulative summing of the angular rate vector provided by the IMU.

TABLE V. Autocorrelation Parameter Values.

Run 1	*Run 2*	*Run 3*	**Average**	
0.08	0.12	0.17	**0.123**	ρ_ϕ
4.0×10^3	3.0×10^3	3.0×10^3	$\mathbf{3.33 \times 10^{-3}}$	β_ϕ
0.025	0.025	0.035	**0.0283**	ρ_θ
2.5×10^{-3}	1.5×10^{-3}	1.5×10^{-3}	$\mathbf{1.83 \times 10^{-3}}$	β_θ
0.15	0.045	0.035	**0.04**	ρ_ψ
2.5×10^{-3}	1.3×10^{-3}	1.0×10^{-3}	$\mathbf{1.115 \times 10^{-3}}$	β_ψ

Autocorrelations of the unsummed angular rate vector are calculated, and ρ and β values determined by exponential curve fitting. An example plot with its curve fit is shown in Figure 8. The complete collection of Gauss-Markov parameter values is found in Table V. IMU sensor noise is characterized by using the averages of these values.

Dynamic Model Development. The general form for the dynamic perturbation model of the platform is

$$\delta\dot{\mathsf{x}}_k = \mathbf{F}\delta\mathsf{x}_k + \mathbf{B}\mathsf{u}_k + \mathbf{G}\mathsf{w}_k \tag{13}$$

where $\delta\mathsf{x}_k$ is the perturbed state vector, u_k is the deterministic input vector, and w_k is a vector of zero-mean additive white Gaussian noise. $\mathbf{F}$ is the system dynamics matrix, $\mathbf{B}$ is the control matrix, and $\mathbf{G}$ is the noise matrix. There is no control mechanism for the system, so $\mathbf{B} = 0$. The $\mathbf{G}$ matrix is the identity matrix. The $\mathbf{F}$ matrix is

$$\mathbf{F} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -\beta_\phi & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\beta_\theta & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\beta_\psi \end{bmatrix}. \tag{14}$$

The state uncertainty matrix, $\mathbf{Q}_k$, is comprised of the variances of the different states. The variances of the three bias states, b_ϕ, b_θ, and b_ψ, are found from using Equation (15) for each run and averaging the results. The units for the variance of the bias states are degrees squared. The three angular rate state variances, p, q, and r, are found from the variance of the output of the motionless IMU and averaging the results from the three runs. The units for the variance of the angular rates are degrees squared per second squared. The variance of the angular displacement states, ϕ, θ, and ψ, are determined by integrating the angular rates and averaging the variances of the results from the three runs. The units of the angular displacement variances are degrees squared. The final values for $\mathbf{Q}_k$ are found in Equation (16).

$$\sigma^2 = 2\beta\rho \tag{15}$$

$$\mathbf{Q}_k = \begin{bmatrix} 4.3E{-9} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 4.3E{-9} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 9.2E{-9} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.6E{-5} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.6E{-5} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 3.6E{-5} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 8.2E{-4} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.0E{-4} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 9.2E{-5} \end{bmatrix} \tag{16}$$

Measurement Model Development. The measurement model is defined as

$$\delta \mathsf{z}_k = \mathbf{H}\delta \mathsf{x}_k + \mathsf{v}_k \tag{17}$$

where $\mathbf{H}$ is the observation matrix, z_k is the measurement vector, and v_k is the measurement noise. The observation matrix $\mathbf{H}$ defined as

$$\mathbf{H} = \begin{bmatrix} -55 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1898 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1898 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{18}$$

These values were found by performing parametric sweeps and visually matching the output plots to a section of video of known displacement. Note that for a perturbation measurement model, the $\mathbf{H}$ matrix is the negative of the true observation matrix.

The z_k vector in the measurement model is comprised of the frame to frame displacement estimates provided by EIS. EIS is a particularly unique kind of measurement, because it offers relative measurements and not absolute. The displacement is given from the x_{true} at the last time of frame capture, which is estimated by $\hat{\mathsf{x}}$ in the Kalman filter. The perturbation model must account for this. The new $\delta \mathsf{z}_k$ is found to be

$$\delta \mathsf{z}_{k_{\text{cf}}} = \mathsf{z}_{k_{\text{cf}}} - \mathbf{H}\left[\mathsf{x}_{k_{\text{cf}}} - \hat{\mathsf{x}}_{k_{\text{pf}}}\right]. \tag{19}$$

where k_{cf} is the current frame time and k_{pf} is the previous frame time. The full perturbation measurement model is then

$$\delta \mathsf{z}_{k_{\text{cf}}} = \mathbf{H}\delta \mathsf{x}_{k_{\text{cf}}} - \mathbf{H}\hat{\mathsf{x}}_{k_{\text{pf}}} + \mathsf{v}_{k_{\text{cf}}}. \tag{20}$$

To make the appropriate changes to the Kalman filter equations in Figure 9, replace every occurrence of $\hat{\mathsf{x}}_k$ with $\delta\hat{\mathsf{x}}_k$, and every occurrence of z_k with $\delta \mathsf{z}_k$, and proceed as normal.

Measurement Variance Determination. The $\mathbf{R}_k$ matrix values define how much of EIS measurement is incorporated, or how little. When the detector value is near zero, all of the EIS measurement is desired. When the detector value is close to unity, all of the IMU measurement is desired.

The data from one of the three stationary test sets is used to determine the precise relationship between R and the amount of EIS measurement to incorporate. The IMU was left motionless on a table, thus the true values of the angular rates are zero for all time. EIS measurements are simulated by a zero vector. The state estimate $\hat{\mathsf{x}}_k$ of the filter should then be close to zero if EIS is fully incorporated, and deviate from zero if none of the EIS measurement is incorporated. This corresponds to the total error. The state estimate will have minimum error if all of the EIS estimate is fully incorporated, and maximum error if none of the EIS estimate is incorporated.

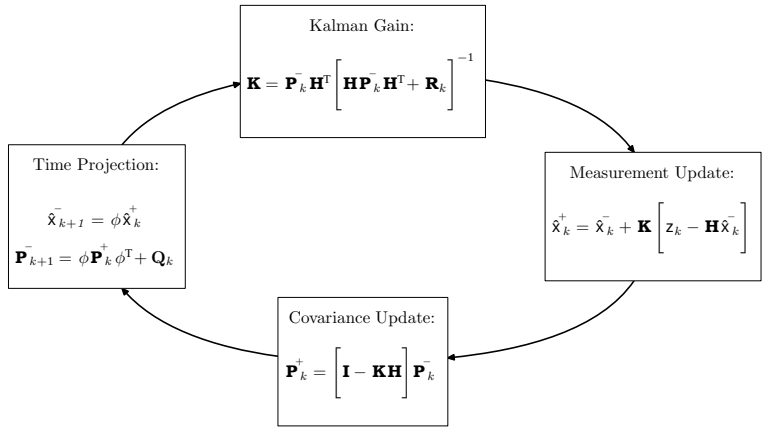

Fig. 9. Kalman Filter Equations [20].

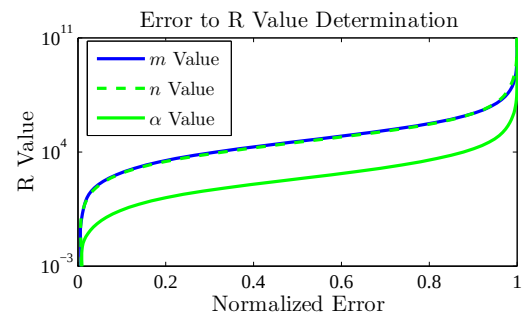

Fig. 10. R **Value Error Plot.**

RMS errors were collected for values of R ranging from 1×10^{-5} to 1×10^{15}. Maximum RMS error occurs when no EIS information is incorporated into the measurement, and minimum RMS error when EIS is incorporated into the measurement. Figure 10 shows the normalized error to R value relation. These complex curves are difficult to match with a simple equation. However if the view is constrained to the linear portions of the curves, as shown in in Figure 10, the curves resemble exponential form. Thus the R(D) equations should be of some exponential form

$$\eta e^{\gamma \mathrm{D}}. \tag{21}$$

The final values for the η and γ coefficients are determined by using parametric sweeps. These sweeps were conducted during the moving object hallway test. The coefficients are chosen so as to minimize the total RMS error for the run. These values are thus optimized for the specific environment of building hallways. Depending on the scene environment of the platform, these values may be altered to provide better results. The final R(D) equations are

$$\begin{aligned} \mathrm{R}_m &= (1 \times 10^{-3}) \times e^{8\mathrm{D}} \\ \mathrm{R}_n &= 0.1 \times e^{6\mathrm{D}} \\ \mathrm{R}_\alpha &= (1 \times 10^{-3}) \times e^{18\mathrm{D}}. \end{aligned} \tag{22}$$

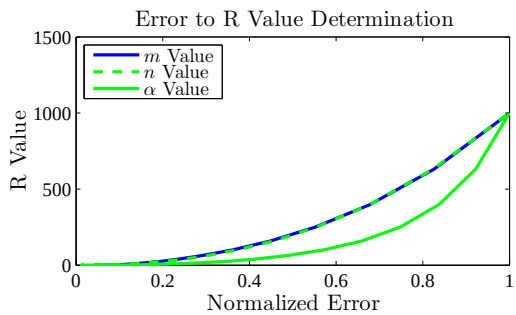

Fig. 11. Zoom of R Value Error Plot.

TABLE VI. Hallway Test Error.

	m RMS	n RMS	α RMS
Feature Detection	12.52	9.03	0.71°
Optical Flow	3.04	2.91	0.48°
Inertial Measurement	23.76	23.43	1.00°
Optical Flow with Inertial Fusion	2.58	5.62	0.79°

Camera to IMU Bias Determination. A bias that separates the image frame capture time from the IMU capture time exists between the camera clock and the IMU clock. The bias must be known in order to correctly fuse the data. This bias was determined to be 0.13 seconds by matching up the known displacement of the video feed to the calculated displacement output by the IMU.

Referring to the block diagram in Figure 6, fusion of the optical flow estimates and inertial data is achieved by the Kalman filter. To determine the optimal frame to frame displacements, sum the angular displacements in between frame times and multiply the result by the appropriate coefficient specified in the $\mathbf{H}$ matrix.

The motion parameter estimates of the video is then low pass filtered, and the local motion extracted. Compensation on the image is then accomplished by transforming the image by the negative values of the local motion estimate. The result is a stabilized video feed that maintains gross motion.

1) Inertial Algorithm Results:

The first test uses the video from the DAGSI Whegs as it navigated through the hallway. Because of the lack of a truth model for the IMU values, a manual determination of frame movement is accomplished. After this process is conducted on the video sequence, the resulting motion estimates are used to create a new stabilized video. The parameters are then fine tuned visually, correcting for any movement the video appeared to undergo. After several iterations of this process, a stable video is achieved, and the true frame to frame motion parameter values well approximated. The results comparing the algorithm to this manual truth model are shown in Table VI. Optical flow, and optical flow with inertial fusion, are comparable in performance.

The second test involves a simulated moving object in the hallway video. A 300×300 pixel black box is sent left to right across the scene. The results from the different algorithms are shown in Table VII. Using optical flow with inertial fusion

TABLE VII. Hallway with Moving Object Test Error.

	m RMS	n RMS	α RMS
Feature Detection	18.01	31.03	2.92°
Optical Flow	16.16	21.08	2.20°
Inertial Measurement	23.76	23.43	1.00°
Optical Flow with Inertial Fusion	11.82	14.47	0.78°

results in a 27% reduction in RMS error in the m direction, a 31% reduction in RMS error in the n direction, and a 63% reduction in rotation RMS error, compared to the optical flow algorithm. Averaging these values, optical flow with inertial fusion is capable of 40% lower RMS error than the best alternative in the presence of moving objects.

Algorithm Speeds. Algorithm speeds were also compared. The mean required time for optical flow is 1.1 seconds for each estimation loop. Feature detection requires 4.6 seconds. Template matching requires 2.5 seconds per loop. Inertial measurement required 0.6 seconds. The simulations were conducted on a personal laptop running a 2.1 MHz processor. With faster hardware and speed optimized code the algorithms will run much faster, however these loop time values provide insight into relative time requirements between the methods.

T-Significance Testing. To obtain further insight into the fusion of inertial data, a T-significance test is conducted between the error plots of the optical flow algorithm and the optical flow with inertial fusion algorithm. The resulting p-values are

$$\begin{aligned} m &: \text{p} - \text{value} = 0.801 \\ n &: \text{p} - \text{value} = 0.762 \\ \alpha &: \text{p} - \text{value} = 0.303 \end{aligned} \tag{23}$$

For the m and n directions, there is not much statistical difference between the performance of the optical flow algorithm and the optical flow with inertial fusion algorithm. This can be directly interpreted from the plots in Figure 12. Large spikes occur at the same frame transitions for both algorithms. This occurs because both optical flow and inertial measurement incur large estimation errors for these specific frame transitions. The substitution of a bad IMU measurement is made for a bad optical flow measurement. Use of a higher grade IMU will improve the IMU measurements, resulting in greater statistical difference between the optical flow algorithm and the optical flow with inertial fusion algorithm. However, even though both algorithms experience peak errors at the same frame transitions, these errors are significantly reduced.

IV. Conclusions

For robust image stabilization, it is necessary to combine the effectiveness of optical flow and inertial measurement. Optical flow provides accurate estimation using the video feed even in the presence of large image translation and image blurring. Inertial measurement provides invariance against moving objects. The novel algorithm presented in this paper, optical flow with inertial fusion, combines these two methods, and is capable of 40% reduction in RMS error compared to optical flow alone in the presence of moving objects.

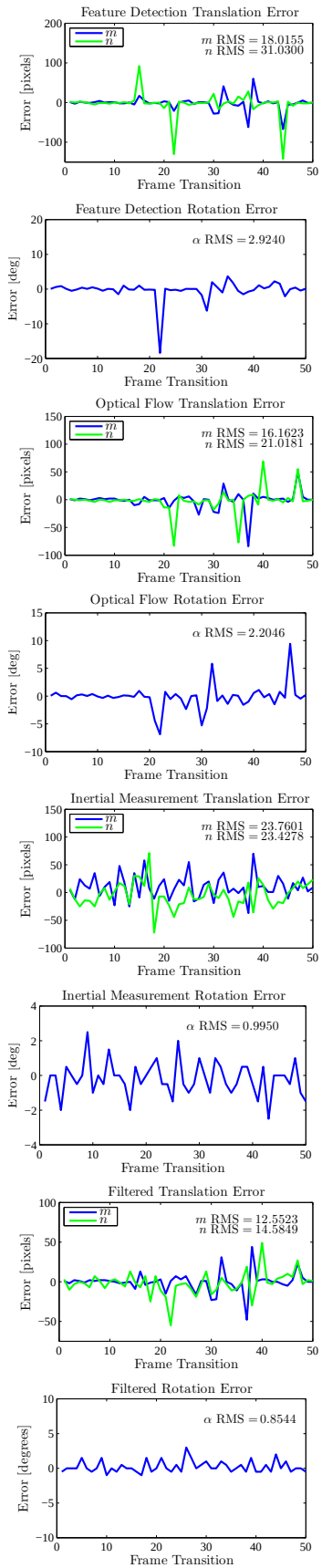

Fig. 12. Hallway with Moving Object Test Error.

The main contribution of this paper is the novel EIS algorithm, optical flow with inertial fusion. It uses Shi-Tomasi good features and pyramidal Lucas-Kanade optical flow fused with inertial data by way of a nine state discrete Kalman filter. The second contribution of this work is an EIS algorithm capable of effective stabilization on the DAGSI Whegs robotic platform, for which an image stabilizer has not been developed. The third contribution of this work is a numerical analysis performed on the four main classes of EIS.

References

[1] M. J. Veth, *Fusion of Imaging and Inertial Sensors for Navigation*. Ph.D. Dissertation. Air Force Institute of Technology, 2006.

[2] A. S. Boxerbaum, J. Oro, G. Peterson, and R. Quinn, "The latest generation whegs robot features a passive-compliant body joint," *International Conference on Intelligent Robots and Systems*, September 2008.

[3] M. Hirooka, "Hierarchical distributed template matching," *Proceedings of SPIE, the international society for optical engineering*," vol. n3029, p. 176, 1997.

[4] R. Szeliski, *Image Alignment and Stitching: A Tutorial*. Microsoft Corporation, 2006.

[5] C. Guestrin, F. Cozman, and E. Krotkov, "Fast software image stabilization with color registration," *IEEE/RSJ International Conference on Intelligent Robots and Systems*," vol. 1, pp. 19-24, 1998.

[6] W. Zhu, K. Li, X. Gao, J. Gao, J. Li, and X. Zhang, "A real-time scheme of video stabilization for mine tunnel inspection robot," *IEEE International Conference on Robotics and Biomimetics*, pp. 702-705, 2007.

[7] M. A. Fischler, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, p. 381, June 1981.

[8] B. S. Morse, D. Gerhardt, C. Engh, M. A. Goodrich, N. Rasmussen, D. Thornton, and D. Eggett, "Application and evaluation of spatiotemporal enhancement of live aerial video using temporally local mosaics," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1-8, 2008.

[9] C. Morimoto, and R. Chellappa, "Fast electronic digital image stabilization," *Proceedings of the 13th International Conference on Pattern Recognition*, vol. 3, pp. 284-288, 1996.

[10] H. R. Pourreza, M. Rahmati, and F. Behazin, "An electronic digital image stabilizer based on stationary wavelet transform (SWT)," *International Conference on Image Processing*, vol. 2, pp. 383-386, 2003.

[11] M. A. Alharbi, *Fast Video Stabilization Algorithms*. M.S. Thesis. Air Force Institute of Technology, 2006.

[12] B. D. Lucas and T. Kanade, "An iterative image registration technique with application to stereo vision," *Proceedings of Imaging Understanding Workshop*, pp. 121-130, 1981.

[13] A. Amanatiadis, I. Andreadis, A. Gasteratos, and N. Kyriakoulis, "A rotational and translational image stabilization system for remotely operated robots," *IEEE International Workshop on Imaging Systems and Techniques*, pp. 593-600, 1994.

[14] M. Ramachandran and R. Chellappa, "Stabilization and mosaicing of airborne videos," *IEEE International Conference on Image Processing*, pp. 345-348, 2006.

[15] R. Kurazume and S. Hirose, "Development of image stabilization system for remote operation of walking robots," *IEEE International Conference on Robotics and Automation*, vol. 2, pp. 1856-1861, 2000.

[16] Z. Yong-xiang, "A method of resolving gyro zero drift in electronic stabilization system," *International Conference on Computer and Automation Engineering*, March 2009.

[17] M. J. Smith, *Electronic Image Stabilization for Mobile Robotic Vision Systems*. M.S. Thesis. Air Force Institute of Technology, 2009.

[18] D. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91-110, 2004.

[19] J. Shi and C. Tomasi, "Good features to track," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 593-600, 1994.

[20] R. Brown and P. Hwang, *Introduction to Random Signals and Applied Kalman Filtering*. John Wiley and Songs, 1997.

[21] G. Bradski and A. Kaehler, *Learning OpenCV: Computer Vision with the OpenCV Library*. Sebastopol, CA: O'Reilly Media, Inc., 2008.