Sonar and Video Fusion for Vehicle Trajectory Estimation in Under-ice Environments

Anthony Spears, Ayanna Howard
Electrical and Computer Engineering Department
Georgia Institute of Technology
Atlanta, GA

Michael West, Thomas Collins
Georgia Tech Research Institute
Atlanta, GA

***Abstract*— Localization and navigation in an under-ice underwater environment presents a difficult challenge. GPS, the standard means of localization for aerial and terrestrial vehicles, is not available underwater. Common underwater navigation solutions such as compass sensors and acoustic triangulation are not available under the ice or require overwhelming support infrastructure. Decreased localization error in such missions would increase the mission capabilities of the underwater vehicle. Proposed here is a method for utilizing common camera and sonar sensors already onboard many underwater vehicles to bound the error drift rates encountered in many navigation systems. A factor graph framework is used to synergistically combine relative pose estimates from these two independent and complementary sensors. A novel noise model is proposed to weight each pose estimate in the overall graph. Optimization of the factor graph over all estimates and corresponding weights can be performed to obtain a robust estimated vehicle trajectory over the entire mission duration. Such a self-contained and robust method can help aid in bounding the error rates of an under-ice navigation system.**

Keywords—under-ice, navigation, sensor fusion, camera, sonar, GTSAM, factor graph

I. Introduction

Localization and navigation in an underwater environment presents a difficult challenge. GPS, the standard means of localization for aerial and terrestrial vehicles, is not available underwater. While some open water autonomous underwater vehicles (AUVs) have developed a solution to occasionally surface and obtain a GPS baseline, this is not possible in under-ice environments. Another commonly used solution for underwater localization takes the form of acoustic beacon networks for triangulation. However, deployment of such a system is infeasible in many ice-covered environments due to the infrastructure required to deploy the required beacons through the ice. Under-ice environments located in the polar regions of Earth also introduce the additional challenge of limited compass functionality at such extreme latitudes so close to the magnetic poles. All of these challenges present a very inhospitable environment for autonomous systems, which can easily get lost as their estimated ego-location accumulates error. Under-ice AUVs cannot be easily recovered after an accumulation of error in the position estimate (as those in open ocean can by simply surfacing), and much greater care must be taken in the localization and navigation problem in under-ice environments. A system capable of providing better estimates of localization over an extended trajectory in sensor deprived under-ice environments would allow for longer mission durations, wider vehicle trajectories and greater autonomous capabilities for these AUVs.

The baseline localization method used in almost all open-water and under-ice AUVs is an inertial navigation system (INS) provided with information from an IMU (inertial measurement unit) and gyroscope that can be integrated to obtain position information. While the sole reliance on an INS is prohibitive to most missions due to the unbounded drift rates encountered, this remains the most common solution for under-ice AUVs due to the unique environmental constraints. One proposed solution to the under-ice localization problem is visual SLAM (simultaneous localization and mapping). SLAM [10] refers to a localization method commonly used in terrestrial robotics where observations of landmarks in the environment are used to concurrently create a map of the environment and localize the vehicle inside that map as the vehicle moves. However, under-ice environments are largely low-contrast and featureless, and comprise mostly of repetitive ice texture. Therefore, only a very limited set of current under-ice vehicles utilize a camera-based localization method. SLAM using a forward-looking sonar sensor (present on many underwater vehicles) has been used in open ocean environments, but typically requires large anomaly features for blob tracking throughout the vehicle trajectory. Due to the limited anomaly features present in the sub-ice topography, the use of such sonar sensors for under-ice localization is also very uncommon.

Here, we propose the novel use of both video- and sonar-based relative pose estimation methods to provide additional constraints on the estimated vehicle trajectory and to help bound the drift rates of the INS. In previous work [11] we determined that a forward-looking sonar sensor could be used with feature-based optical flow methods to track three degree of freedom ego-motion, even with relatively featureless environments such as sand or ice. Such a method extracts feature points [12] (pixels with unique and high-gradient neighborhoods) to match between successive frames which can be used to estimate a world motion model. We also determined during initial testing that video-based motion could be estimated, even from the low contrast, relatively featureless environments encountered under the ice. From these insights, we developed a method to obtain both sonar- and video-based estimation of vehicle motion between sensor frames. Sonar data can be used to obtain a three degree of freedom estimate (yaw, surge and sway), while five and a half degrees of freedom (minus translational scale) can be estimated from the video data. However, each of these sensors has unique

978-0-933957-43-5 ©2015 MTS

limitations when used independently. In addition, some individual frame-to-frame estimates over the trajectory can be inaccurate due to a low number of feature points detected in a frame, a low number of point matches between frames, or inaccurate point matching.

Independent INS, camera-based, and sonar-based systems each present unique weaknesses and strengths, and each sensor can encounter periods of inaccurate estimation. We propose the use of a factor graph framework [1][3] to fuse this information together and obtain a robust estimate over the vehicle trajectory which combines the strengths of each sensor to overcome the independent weaknesses. A factor graph framework is a model well suited to complex estimation problems where variables represent unknown random variables to be estimated and the factors between them represent probabilistic information on those variables (Figure 1).

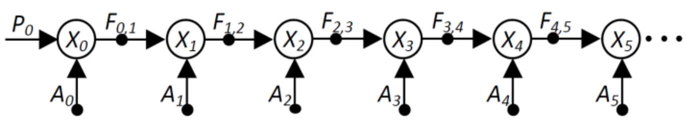

Fig. 1. Visualization of a generic factor graph framework, where variable nodes are shown as large open circles (Xn), a prior node is shown as P0, factor nodes are shown as small solid circles (Fn,m for binary and An for unary, or absolute, factors) and where n,m $\in$ {0,1,2⋯}. Movement left to right here represents passage of time.

Optimization can then be performed over all available estimates to obtain the optimal vehicle state estimate over the entire trajectory. Here we propose the use of a GTSAM (Georgia Tech Smoothing and Mapping) [1][2][3] framework to fuse frame-to-frame sonar and video relative pose estimates. We propose a novel sensor fusion method which incorporates the number of features matched between frames into the error model of the estimated vehicle trajectory in order to more heavily weight the impact of estimates with more matches over those with less matches. In this way, stronger estimates from one sensor can overpower the weaker, more inaccurate estimates from the sensor with less point matches, resulting in a more robust system. The proposed sensor fusion method shows promising results for aiding in vehicle relative pose estimation by robustly fusing estimates from both sonar and video sensors. The proposed fusion framework, algorithm and evaluation results are discussed herein.

II. Prior Work

Exploiting the complementary and independent sensor data obtained with underwater range and optical sensors through fusion has been proposed in a few previous works [4][5][6][7][8]. However, very few systems in the literature investigate fusion of underwater sonar and camera sensors for use in ego-motion estimation. Majumder et al. [4] propose a low level fusion of these two sensors prior to feature extraction for more reliable mapping and navigation. Williams and Mahon [5] propose the projection of strong sonar cluster returns into camera images to provide initial search locations of visual point-features used for SLAM on the Great Barrier Reef. While both of these low-level fusion methods exploit the strengths of both camera and sonar sensors, they require consistently reliable input from both sensors (in contrast to the higher level fusion methods such as that proposed here). Monocular camera and multibeam sonar fusion in a pose graph SLAM framework is used by Kunz and Singh [6] to automatically build fused 3D maps of underwater terrain. Underwater sensor fusion is performed by Hover et al [8] for vehicle position estimation during ship hull inspection by combining sonar sensor and monocular camera constraints using a pose graph framework. While these pose graph sensor fusion methods are similar to the work proposed here, these applications use ice draft topography [6] (requiring uniquely varying ice draft) and clustered, strong gradient pixel returns [8] as features to match and provide sonar constraints, in contrast to the point-feature-based odometry constraints proposed here. These previous pose graph systems use SLAM instead of odometry methods, which require unique landmark features to be present in both datasets that can be saved and robustly matched over the vehicle's trajectory, especially during loop closure events. While these systems can obtain very low errors assuming re-observation of unique landmarks, this assumption breaks down in the case of continuously textured but non-uniquely varying environments, such as sand-ripples and the majority of ice topography. While these previous sensor fusion systems have proven the utility of such methods in underwater environments, systems robust to the unique feature-poor, under-ice environments, such as that considered here, do not currently exist in the literature.

III. Factor Graphs and GTSAM

Under-ice environments can be very feature-poor and have low-contrast. Therefore, relative pose estimates obtained from one sensor (i.e. camera or sonar) can be prone to occasional inaccuracies. Fusion of estimates from multiple sensors can provide much more robust estimates. The proposed type of high-level sensor fusion uses a pose graph framework to optimally combine noisy but partially redundant estimates of relative pose from sonar, camera and other navigation sensors. Pose graphs are abstract representations of variables and constraints in an optimization problem [3]. The algorithm proposed here utilizes a factor graph framework inside the Georgia Tech Smoothing and Mapping Library (GTSAM) presented by Dellaert and Kaess [1][2][3]. A factor graph is a bipartite graphical model G = (Ƒ, Q, Ɛ) containing variable nodes $q_j \in$ Q (unknown random variables in the estimation problem), factor nodes fi $\in$ Ƒ (probabilistic information on those variables derived from measurements or prior knowledge) and edges $e_{ij} \in$ Ɛ (encode sparse structure between q_j and f_i). The Levenberg-Marquardt algorithm [9] is used to optimize the pose graph over the entire trajectory.

IV. Relative Pose Estimation Overview

The use of the proposed fusion algorithm assumes frame-to-frame estimates of vehicle motion have been calculated for both sonar and camera sensor data. While a detailed description of the calculation of these estimates would be too lengthy to provide here, a summary of the methodology used is presented in this section. For each sensor, raw data frames are first obtained from the sensor for use as input. Upon arrival of each new frame, strong feature points (uniquely identifiable points with a steep gradient in both the x and y directions) are extracted for both the current and previous frame. These points are matched between frames and an optimal model for sensor (and therefore vehicle) motion between frames is calculated using a robust set of inlier matches. In the camera case, a six degree of freedom (actually five and a half, as the x, y, z translation estimate is limited to a unit vector without magnitude) model is calculated using Nister's 5-point algorithm [13]. In the sonar case, a 3-point rigid motion model [14] is used to calculate a three degree of freedom (x, y, and yaw) estimate from a robust set of inlier point feature matches. The outputs of the frame-to-frame motion estimation algorithms consist of an array of six and three motion estimates (camera and sonar respectively) along with a timestamp and the number of inlier match points used for the model. The output camera array (Equation 4) contains estimates for change in roll, pitch and yaw (ϕ, θ and ψ respectively) in degrees as well as the direction (x, y, z unit vector) of the change in translation. The output sonar array (Equation 5) contains an estimate for change in yaw as well as an estimate of x and y translation in meters. The outputs from these shift estimation algorithms are provided directly as inputs to the proposed fusion algorithm presented here. These values are assumed to have been converted to the vehicle's coordinate system (Figure 2) prior to input.

$$CameraEst = [\hat{x}\ \hat{y}\ \hat{z}\ \phi\ \theta\ \psi] \tag{4}$$

$$SonarEst = [x\ y\ \psi] \tag{5}$$

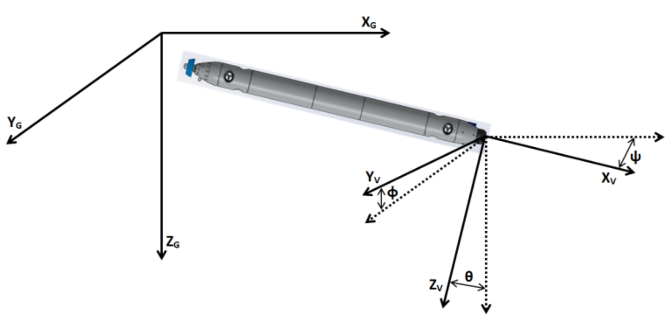

Fig. 2. Vehicle coordinate system. A global reference frame is used for referencing absolute angles and translations while a vehicle reference frame is used for referencing velocity.

V. ALGORITHM

The fusion algorithm presented here requires processed frame-to-frame relative pose estimates from both an acoustic sonar sensor and an optical camera sensor as inputs. These two sensors provide concurrent and independent vehicle motion estimate information, which is partially redundant but also complimentary. Sonar estimates provide absolute shift estimates (measured in meters and degrees) but are only capable of resolving motion in the vehicle's x, y, and yaw directions. Camera estimates provide a more complete six degree-of-freedom picture with the addition of z (depth or heave), roll, and pitch information. However the translational (x, y, and z) estimates are limited to a unit vector (due to a scale ambiguity), providing direction of the translational shift without the magnitude values. When processed concurrently, fusion of motion estimates from these two independent sensor streams can leverage the strengths of each sensor to provide a full and robust estimate of vehicle motion.

Sonar frames are obtained from the sensor for processing at a rate that can vary from the advertised minimum of 67ms to over a second in the case of the BlueView P900-45 used here (depending on the range distance). In almost all cases, this frame rate is much lower than that of the camera in the system (30 fps). The fusion algorithm proposed here utilizes all frames available, even those obtained at vastly different frame rates, without the need for prior synchronization the sensors. A pose graph is used to encompass all estimate information over the vehicle trajectory. The rate of node creation in the fusion graph is set by the sensor with the higher frame rate (camera). A variable node and two factor nodes are created in the fusion graph corresponding to each camera frame timestamp. For each set of successive camera frames, two constraints are added to the fusion graph between the nodes corresponding to the timestamps of the two camera frames. The first constraint contains the sonar motion estimate of x, y, and yaw which has been subdivided in time to match the camera frame timestamps (Equations 6-7). The second constraint provides the camera motion estimate of roll, pitch and yaw as well as an estimate of translational motion obtained by multiplying the unit vector from the camera estimate by the total magnitude translation estimate obtained by the sonar sensor (Equations 8-10). A fusion factor graph containing such variable nodes and factor constraints can be optimized (provided an initial estimate) to provide an optimal trajectory of the vehicle motion over the entire trajectory. This optimal result utilizes all estimate information provided by the two sensors to create a robust and complete estimate of the vehicle's motion.

$$\boldsymbol{sonVelocity} = \frac{1}{timeDiffSon} * [x_{son}\quad y_{son}\quad \psi_{son}] \tag{6}$$

$$\boldsymbol{sonEstimate} = timeDiffCam * \boldsymbol{sonVelocity} \tag{7}$$

$$sonVelocityMag = \begin{cases} \frac{sonVelocity_x}{\hat{x}_{cam}} & \hat{y}_{cam} < 0.1 \\ \frac{sonVelocity_y}{\hat{y}_{cam}} & \hat{x}_{cam} < 0.1 \\ \frac{sonVelocity_y}{2*\hat{y}_{cam}} + \frac{sonVelocity_y}{2*\hat{y}_{cam}} & Otherwise \end{cases} \tag{8}$$

$$\boldsymbol{camVelocity} = sonVelocityMag * [\hat{x}_{cam}\quad \hat{y}_{cam}\quad \hat{z}_{cam}] \tag{9}$$

$$\boldsymbol{camTransl} = timeDiffCam * \boldsymbol{camVelocity} \tag{10}$$

The proposed sensor fusion algorithm was designed for use in both real time applications as well as during post-processing analysis. Therefore, the algorithm design takes into account the unique input data flow. The proposed algorithm is presented in pseudo code form in Figure 3. A GTSAM nonlinear factor graph is initialized during startup to provide the framework for the fusion algorithm. An initial prior factor node is added to the factor graph to encode the prior estimate of vehicle state along with the confidence (or noise) of the prior estimate. An initial value array is also instantiated to provide a framework for the initial guess at each graph node, which is required for factor graph optimization. An initial value is added to the beginning of the initial value array corresponding to the prior factor node added to the factor graph.

```
Pseudo Code:
Create factorGraph
Create initialValueArray
Add priorFactor to factorGraph
Add prior value to initialValueArray
//Main (Sonar) Loop:
While(input data available)
{
 While(no sonar frame available)
   Buffer camera rrame
  If(Sonar frame arrives)
  {
    If(sonInliers>inlierThresh)
      Update robust sonar estimate
    timeDiffSon=frame2.time-frame1.time
    Calculate sonVelocity (Eq. 6)
    Calculate sonar noise model (Table 1)
    //Secondary (Camera) Loop:
    For(all buffered camera frames)
      {
        If(sonInliers>inlierThresh)
          Update robust sonar estimate
        timeDiffCam=frame2.time-frame1.time
        Subdivide sonar estimate (Eq. 7)
        Calculate sonVelocityMag (Eq. 8)
        Calculate camVelocity (Eq. 9)
        Calculate camera estimate (Eq. 10)
        Calculate camera noise model (Table 1)
        factorGraph.add(sonarEst)
        factorGraph.add(cameraEst)
        initialValue = avg(sonarEst, cameraEst)
        initialValueArr.add(initialValue)
      } End For
  }End If
}End While
Result = Optimize(factorGraph, initialValueArr)
```

Fig. 3. Pseudo code for the proposed algorithm

After initialization, the main loop begins with data inflow from camera and sonar frame-to-frame shift estimates. Due to the higher frame rate, camera shift estimates are buffered until a corresponding sonar estimate arrives. Upon arrival of the next sonar shift estimate, the time difference between frames is recorded as *timeDiffSon* for use in later calculations. For both sonar and video frames, 200 inlier point-feature matches (*inlierThresh*) is considered sufficient to provide a robust estimate of relative pose. If the number of match inliers between sonar frames exceeds 200, indicating a confident estimate, this estimate is saved as the most up to date robust information obtained. This robust estimate can be used in future calculations in cases where more current sonar data is unreliable or nonexistent. Prior to arrival of the first robust sonar estimate of sufficient confidence (at least 200 matching inlier points), the algorithm estimates zero translational shift and relies only on the camera inputs for rotational information. This robust estimate is also used as input to an accumulator which keeps a running sum of frame-to-frame estimates to be used for nodes in the initial value array. In order to synchronize the sensor inputs across different frame rates and arrival times, velocity estimates (x, y and yaw) are calculated from the sonar shift estimates and the time difference between frames (*timeDiffSon*), as shown in Equation 6. These velocity values normalize the shift estimates over time and are used in later calculations.

The confidence (noise or variance) value corresponding to the frame-to-frame sonar estimate is then calculated, as it is required when the factor is added to the factor graph. This confidence value is calculated using the number of inlier matches between the sonar frames (Equation 11 and Table 1). If there are at least 200 matching inlier points between sonar frames, the confidence is a constant for both translation (0.0001) and rotation (0.00001). In cases with less than 200 inlier matches, the sonar noise value model is linear with a slope equal to -1/200 and a maximum at two orders of magnitude above the high confidence values. In the case of zero inlier matches, this results in a noise value of 0.01 for translation and 0.001 for rotation. All minimum and maximum variance values can be seen in Table 2. The model for sonar noise versus number of inlier matches is shown in Figure 4 (translational) and Figure 5 (rotational). It is important to note that while all maximum and minimum values of variance are presented here as explicit, quantitative values, these values were chosen based on relative considerations and the absolute values can be adapted to fit any similar sensor error models. The important considerations for the explicit minimum and maximum noise values selected here include an order of magnitude factor between translation (meters) and rotation (degrees), and an order of magnitude between the minimum and maximum in each case. Some consideration was taken in this case to adapt these absolute min/max values in order to enable extraction of marginal covariance values at each node in the factor graph. If all absolute minimum and maximum variance values presented here are increased or decreased by an order of magnitude, the optimization algorithm will converge but marginal covariance values cannot be extracted using the available GTSAM function (assumed to be caused by an overload of the numerical type used, resulting in a non-numerical result).

$$noiseMult = 1 - \frac{numInlierMatches}{inlierThreshold} \tag{11}$$

TABLE I. NOISE MODEL EQUATIONS

Sensor	Transl/ Rot	Noise (Variance) Model Equations		
		Inliers = 0	***Inliers < 200***	***Inliers ≥ 200***
Sonar	**Transl**	0.01	$0.01 * sonNoiseMult$	0.0001
	Rot	0.001	$0.001 * sonNoiseMult$	0.00001
Camera	**Transl**	Eq. (11)	Eq. (12)	Eq. (13)
	Rot	0.001	$0.001 * camNoiseMult$	0.00001

TABLE II. MINIMUM/MAXIMUM NOISE VARIANCES

Noise	Sonar	Camera (Sonar=MAX)	Camera (Sonar=MIN)	Sonar	Camera
Max	0.01	0.1	0.01	0.001	0.001
Min	1E-4	0.001	$1*10^{-4}$	$1*10^{-5}$	$1*10^{-5}$

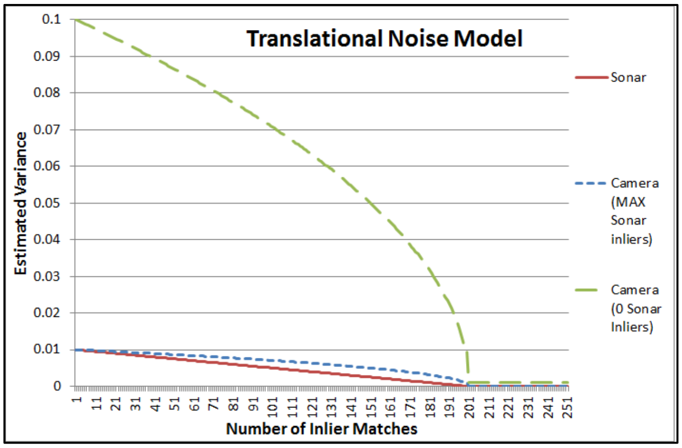

Fig. 4. Translational noise model showing variance values used for the noise model on a factor versus the number of inlier point matches used to calculate the relative pose model. The maximum variance occurs with zero matches while the minimum variance occurs with over 200 matches.

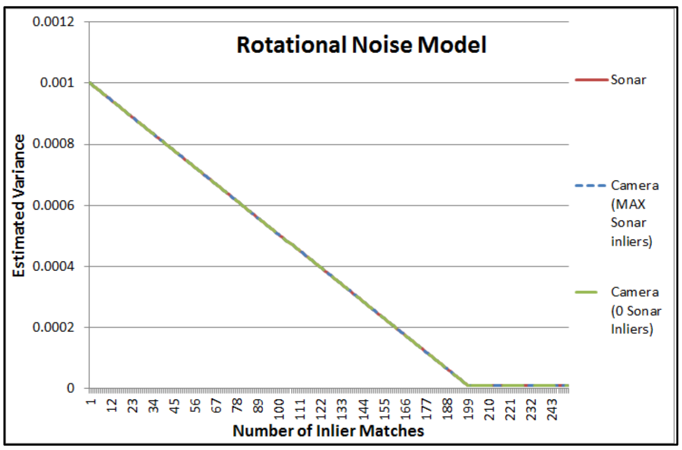

Fig. 5. Rotational noise model showing variance values used for the noise model on a factor versus the number of inlier point matches used to calculate the relative pose model. The maximum variance occurs with zero matches while the minimum variance occurs with over 200 matches.

Once all calculations are complete following the arrival of a new sonar frame-to-frame estimate, the previously buffered camera estimates obtained between the last sonar arrival and the current arrival are considered. A second nested loop is used to calculate and add each camera factor along with a corresponding sonar factor to the factor graph. For each camera estimate, the time difference between frames is calculated (*timeDiffCam*). In a similar manner to the sonar processing method, the robust camera estimate is updated with the current shift if the number of inlier matches exceeds the threshold of 200. This previous robust estimate is used for future calculations in cases where the current shift estimate is unreliable or non-existent, yielding a more robust system. An estimate for total magnitude translational camera velocity is calculated from both the robust sonar (x and y) estimate magnitudes and the camera (x, y and z) estimate unit vector due to the lack of magnitude information available from the camera estimates (Equation 8). This calculation assumes that the robust sonar translational direction and camera translational direction are in agreement in order to extrapolate magnitude information from the sonar estimate to the camera estimate. If the x and y components of the estimated camera translation unit vector are above 0.1, an average of the two is used to calculate the estimated velocity magnitude. However, if one of these component values is below this threshold, the other is used alone to avoid unstable division by small numbers. The translational velocity magnitude is first obtained by dividing the robust sonar velocities by the corresponding unit vector components (Equation 8). The total estimated camera translational shift magnitude can then be calculated from this velocity magnitude estimate by multiplying by the camera translation unit vector (Equation 9) followed by the time between camera frames (*timeDiffCam*) obtained previously (Equation 10).

$$\sqrt{sonNoiseTransl} \tag{12}$$

$$\sqrt{camNoiseMult * sonNoiseTransl} \tag{13}$$

$$\sqrt{0.0001 * sonNoiseTransl} \tag{14}$$

Values for rotational and translational noise (variance) are also obtained for each camera factor. Because the previous robust sonar estimate is used in the calculation of the translational camera estimates, the noise of this sonar estimate must be incorporated into the overall translational noise estimate of the camera factor. A geometric mean (Equation 13) is used to combine the two translational noise estimates (*sonNoiseTransl* and *camNoiseTransl*) into a resultant fused translational sonar-camera noise estimate represented by *camSonNoiseTransl*. As rotational camera noise does not depend on the sonar estimates, such an averaging technique is not required for this calculation. The strictly camera-based portion of the noise estimate (*camNoiseTransl* and *camNoiseRot*) is calculated in a similar manner to that of the sonar noise estimates described previously. These variance values are calculated using the number of inlier matches between the camera frames with the same approach described previously for sonar frames (Equation 11 and Table 1). If there are at least 200 matching inlier points, the confidence is a constant for both translation (0.0001) and rotation (0.00001). For cases where there are less than 200 inlier matches, the confidence value model is non-linear with a maximum of two orders of magnitude more than the high confidence values. In the case of zero inlier matches, this results in a value of 0.01 for translation noise and 0.001 for rotation noise. The resultant camera-only noise estimate (*camNoiseTransl*) is then combined with the most recent sonar-only robust noise estimate (*sonNoiseTransl*) using a geometric mean (Equation 13), as described above. A geometric mean is used to add robustness to the common case where the two estimates differ by multiple orders of magnitude. In these cases, a geometric mean method normalizes the weight of each estimate to have a more equivalent effect on the result, in contrast with the use of an arithmetic mean approach. In the case of a strong sonar translational noise value (0.0001), the range of values for the *camSonNoiseTransl* variable spans from 0.01 to 0.0001. At the other extreme, a minimum sonar translation variance (0.01) using a sonar estimate with zero inliers yields a span of resultant camera noise an order of magnitude higher (0.1 to 0.001). An order of magnitude difference between sonar and camera translational variances is used to ensure equal noise values following the geometric sum in the case of full feature match inlier sets (high confidence) with both sensors. The model for camera noise versus number of inlier matches is shown in Figure 4 (translational) and Figure 5 (rotational) for both the case of a full sonar inlier set (> 200) and zero sonar inliers.

Upon calculation of the relative pose estimate and noise model for each set of sonar and camera factors, these factors are added to the initialized nonlinear factor graph as *BetweenFactor<Pose3>* nodes. A single sonar constraint and a single camera constraint are added for each pass through the secondary loop (corresponding to the arrival of a camera frame), while one sonar frame is processed during each pass through the overarching main loop. The populated factor graph representation used in the proposed algorithm can be visualized as in . Here the six degree of freedom vehicle pose is represented by X_n, the initial vehicle pose constraint is represented by P_0, $C_{n,m}$ represents a camera pose constraint between nodes n and m, and $S_{n,m}$ represents a sonar pose constraint between nodes n and m (n,m ∈ {0,1,2⋯t}). As in the proposed algorithm, each pose node and camera factor is added to Figure 6 correspond to the arrival of a camera frame. It can be seen that a corresponding sonar node is also added for each camera frame which is subdivided across multiple video frames to provide a synchronous system despite the differing frame rates of the sensors.

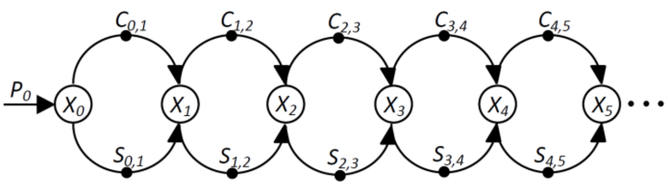

Fig. 6. A visualization of the pose graph framework used in the proposed algorithm. Here the six degree of freedom vehicle pose is represented by Xn, the initial vehicle pose constraint is represented by P0, Cn,m represents a camera pose constraint between nodes n and m, and Sn,m represents a sonar pose constraint between nodes n and m (n,m ∈ {0,1,2⋯t}).

As the algorithm runs through the main loop, accumulator sums of both sonar and camera estimates are maintained for use in calculation of an array of initial values used to seed the optimization algorithm at each node point. In the case of a unique optimization solution, these initial values can be set arbitrarily. For more complex cases in which multiple optimization minimums exist, these values must be estimated close to the optimal solution for the Levenberg-Marquardt optimization algorithm to converge to this desired solution. In the application presented here, these initial values are calculated as an arithmetic average between sonar and camera inputs at each node. In this way, the optimization algorithm begins with a solution that lies equidistant from each independent sensor solution regardless of the confidences on the estimates. During optimization, the confidence value on each estimate is then utilized to pull the result toward one sensor estimate or the other over the entire trajectory.

Once a nonlinear factor graph is created and populated with a prior factor node, as well as all between-node factors (or constraints), the graph and initial value array can be presented as input to the Levenberg-Marquardt optimization algorithm. The resultant array of value node assignments represents the vehicle six degree-of-freedom, globally optimal trajectory, taking into account all frame-to-frame estimates from each sensor and the confidence values on those estimates. This optimal result is smoothed over the entire trajectory using the iterative smoothing methodology [1][2] which smooths in both the forward and backward direction (uses all nodes available) to reduce the effect of short-duration sharp inconstancies or uncertainties at the input. This output result can be used to both estimate the current vehicle state (at the present node) and to estimate the vehicle's entire past trajectory for use in post-processing.

VI. Results

In order to evaluate the utility of the proposed algorithm, we first discuss the effect of this sensor fusion on the variance (noise or confidence) of the optimized assigned variable values. In order to obtain insightful quantitative values, we limit our scope in this discussion to the simplified cases of minimum (zero inlier matches) and maximum (> 200 matches) variance noise models with constant relative pose estimates. Limiting the discussion in this way allows for the simplification of the fusion model, since no smoothing is required and the result is a simple, linearly compounding odometry case where only the current factor estimate and noise is used in calculations. However, this can be extrapolated over all noise models in the range between the minimum and maximum variance values with the addition of a smoothing factor which results from the use of all factors in the graph during the optimization process to eliminate large jumps. The results from these middle cases lie on a continuous spectrum between the minimum and maximum variance values and share the same benefits discussed below, as well as the additional "smoothing" benefit. The fusion equations for this simplified model below (Equations 15-16) give an idea of how over-constrained inputs are weighted in the factor graph framework proposed here. In these cases σ2 represents variance and X represents vehicle state.

$$\sigma_{fused}^2 = \frac{\sigma_{son}^2 * \sigma_{cam}^2}{\sigma_{son}^2 + \sigma_{cam}^2} \quad (15)$$

$$\boldsymbol{X}_{fused} = \boldsymbol{X}_{son} \frac{\sigma_{cam}^2}{\sigma_{son}^2 + \sigma_{cam}^2} + \boldsymbol{X}_{cam} \frac{\sigma_{son}^2}{\sigma_{son}^2 + \sigma_{cam}^2} \quad (16)$$

In cases where both sensors provide reliable (>200 inlier matches, minimum variances) and equal relative pose estimates, the confidence of the estimates doubles (variance halves). The benefit of sensor fusion in this case is to more strongly (factor of two) weight the corresponding fused output due to the additional information encoded, as well as to provide the end user with twice the assurance of the assigned value over independent sensor methods. Qualitatively, a combination of two independent but equal measurements of the same variable provides valuable information on the accuracy of the measurement (reduces variance) and allows for increased dependence on the result. In all combinations of sonar and camera variances with equal pose estimates, fusion using the proposed factor graph framework simply results in increased levels of confidence (reduced variance). The first case described here represents a decreased variance of 50% over each independent sensor estimate. The other three cases considered here also see decreased variances for both sensors, the amounts of which can be seen in Table 3.

In more common cases where the two sensors provide differing pose estimates, the benefits of the sensor fusion method proposed here are much clearer. In the case where both sensors provide reliable but differing pose estimates, an average will result with a 50% decrease in variance over both individual sensor inputs. This results in a more robust estimate equidistant between the two equally likely pose estimates given. In the case where both sensors provide unreliable pose estimates (zero inlier matches, maximum variances), the output will be a weighted average, inherently favoring the sonar estimate as it is used for both translational pose estimate calculations. Lastly, in the most applicable cases for the sensor fusion method proposed here, if one sensor presents a reliable pose estimate and the other presents an unreliable estimate, the factor graph will heavily weight the reliable estimate while essentially ignoring the unreliable estimate. In the specific case of a reliable camera estimate and unreliable sonar estimate, the algorithm weights the camera data at 0.909 and the sonar data at 0.0909. In the specific case of a reliable sonar estimate and an unreliable camera estimate, the algorithm weights the camera data at 0.001 and the sonar data at 0.99. The reliable sonar data is weighted more heavily here over the previous reliable camera case due to the inherent use of sonar translation estimates in both sensor calculations. If the reliable estimates are assumed to be accurate and the unreliable estimates are assumed to be incorrect, these weights translate directly to decreases in error (99% over a camera-only system in the reliable sonar case, 91% over a sonar-only system in the reliable camera case).

In addition to the quantitative sensor fusion benefits described above based on reduced variance and weighted averaging, two important qualitative benefits should be noted. Sonar pose estimates contain information on x, y, and yaw motion but none on z, roll or pitch. Fusion of a complimentary camera sensor provides this z, roll, and pitch information that is not available using only a sonar sensor. Camera pose estimates are limited to a unit vector (no magnitude scale) x, y, z translational shift in addition to full roll, pitch, and yaw estimates. While magnitude of the translational shift cannot be recovered using only a camera sensor, fusion with a sonar sensor provides an estimate of this information. A summary of the quantitative and qualitative benefits of sensor fusion can be found in Table 3 below.

TABLE III. SENSOR FUSION ADVANTAGES

Est. Values	**Variances (Son,Cam)**	**Transl. Fusion Variance**	**Factor Weights (Son, Cam)**	**Fusion Benefit (Son, Cam)**
All	All	N/A	N/A	Added translation magnitude to camera Addition of z, roll and pitch to sonar
Equal	MAX, MAX	0.009091	0.9091, 0.09091	Lower variance (9.1%, 90.9%)
Equal	MIN, MIN	0.00005	0.5,0.5	Decreased variance (50%, 50%)
Equal	MAX, MIN	0.000909	0.09091, 0.9091	Decreased variance (90.9%, 9.1%)
Equal	MIN, MAX	0.000099	0.99, 0.0099	Decreased variance (1%, 99.01%)
Not equal	MAX, MAX	0.009091	0.9091, 0.09091	Weighted average Decreased variance (9.1%, 90.91%)
Not equal	MIN, MIN	0.00005	0.5,0.5	Robust average Decreased variance (50%, 50%)
Not equal	MAX, MIN	0.000909	0.09091, 0.9091	Heavily weight low noise value Ignore high noise val Decreased variance (90.9%, 9.1%)
Not equal	MIN, MAX	0.000099	0.99, 0.0099	Heavily weight low noise value Ignore high noise val Decreased variance (1%, 99.01%)

Multiple examples using realistic (but synthetic) input shift estimates are now considered to better present the utility of the proposed sensor fusion algorithm. In the first case, consistently equal and reliable pose estimates from both sensors results in the translational output variance plot seen in Figure 7. It can be seen here that while the variance of each independent sensor as well as the variance of the fused output compound linearly with time, the output variance consistently remains at half the value of either sensor alone. In the second, more interesting case, synthetic sonar and camera pose estimates are input with alternating periods of unreliable data. Initially both sensor estimates are reliable; then the camera sensor becomes unreliable while the sonar sensor remains reliable; then the sonar sensor becomes unreliable while the camera sensor becomes reliable; etc. The variance plot shown in Figure 8 shows the utility of the proposed sensor fusion method. While the variance of individual sensors increases dramatically upon reaching a period of unreliability, the fusion variance output remains consistently low and is not subject to such drastic effects from individual sensor unreliability periods. This stems from the fact that when one sensor's pose estimate is unreliable, sensor fusion provides another independent source of information to rely on, and a more robust, low noise result can still be obtained.

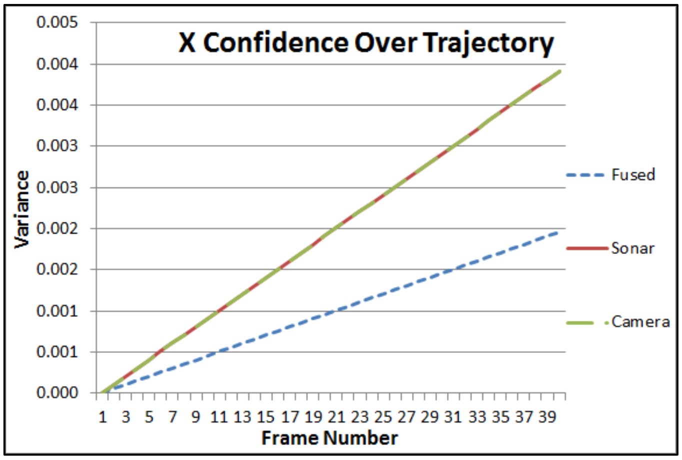

Fig. 7. Variance plot of equally confident sonar and camera estimates in the proposed fusion algorithm. Independent sonar and camera results can be seen to be twice as noisy as the fused result.

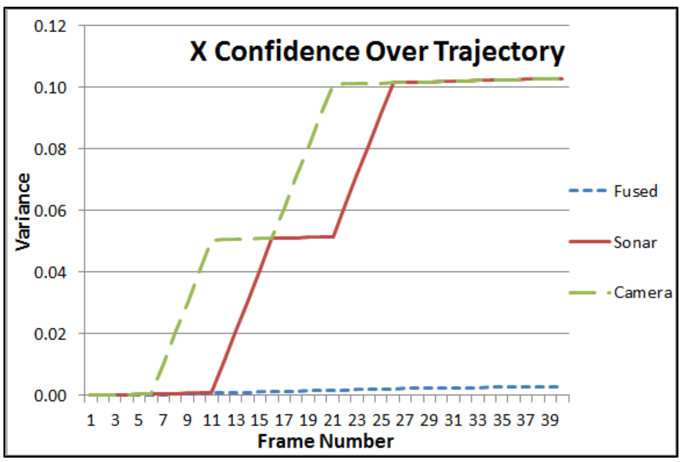

Fig. 8. Variances of the estimates from camera, sonar as well as the fusion estimate, showing the decreased variance (decreased noise and increased confidence) of the GTSAM fused over the individual results. It can be seen that when one sensor (sonar between frames 11 and 15, 21 and 25, camera between frames 6 and 10, 16 and 20) drops out, the combined estimate does not suffer.

VII. Conclusion

Navigation under the ice is a difficult challenge. The authors propose a sensor-fusion method for utilizing camera and sonar sensors already onboard many underwater vehicles to help bound drift rates encountered with inertial navigation systems. Using relative pose estimates between successive sensor frames, a combination of sonar- and camera-based information can provide a robust and full estimate of the vehicle state over the trajectory. A factor graph framework is used here to incorporate all available estimates in order to obtain an optimized trajectory result. Fusion of these two independent and complementary sensors combines the strengths of each sensor to overcome independent weaknesses, yielding a more robust and powerful system over independent sensor systems. The proposed fusion method is detailed herein and results are presented from testing over a variety of synthetic datasets. Such a sensor fusion method can be used to reduce the error in under-ice navigation systems and increase mission capabilities.

Acknowledgment

The authors would like to thank the Georgia Tech Institute for Robotics and Intelligent Machines for providing some of the funding for the work presented here. The authors would also like to acknowledge Dr. Frank Dellaert for his guidance on the GTSAM library.

References

[1] F. Dellaert and M. Kaess, "Square Root SAM: Simultaneous localization and mapping via square root information smoothing," The International Journal of Robotics Research 25, no. 12, pp. 1181-1203, 2006.

[2] M. Kaess, A. Ranganathan, and F. Dellaert. "iSAM: Incremental smoothing and mapping." Robotics, IEEE Transactions on 24, no. 6, pp. 1365-1378, 2008.

[3] F. Dellaert, "Factor graphs and GTSAM: A hands-on introduction," 2012.

[4] S. Majumder, S. Scheding, and H. F. Durrant-Whyte, “Multisensor data fusion for underwater navigation,” Robotics and Autonomous Systems, vol. 35, no. 2, pp. 97–108, 2001.

[5] S. Williams and I. Mahon, “Simultaneous localisation and mapping on the great barrier reef,” in Robotics and Automation, 2004. International Conference on, vol. 2, pp. 1771–1776, IEEE, 2004.

[6] C. Kunz and H. Singh, “Map building fusing acoustic and visual information using autonomous underwater vehicles,” Journal of Field Robotics, vol. 30, no. 5, pp. 763–783, 2013.

[7] S. Negahdaripour, H. Sekkati, and H. Pirsiavash, “Opti-acoustic stereo imaging: On system calibration and 3-D target reconstruction,” Image Processing, IEEE Transactionson, vol. 18, no. 6, pp. 1203–1214, 2009.

[8] F. S. Hover, R. M. Eustice, A. Kim, B. Englot, H. Johannsson, M. Kaess, and J. J. Leonard, “Advanced perception, navigation and planning for autonomous in-water ship hull inspection,” The International Journal of Robotics Research, vol. 31, no. 12, pp. 1445–1464, 2012.

[9] J. Moré, "The Levenberg-Marquardt algorithm: implementation and theory," In Numerical analysis, pp. 105-116, Springer Berlin Heidelberg, 1978.

[10] H. Durrant-Whyte and T. Bailey, “Simultaneous localization and mapping: part I,” Robotics & Automation Magazine, IEEE, vol. 13, no. 2, pp. 99–110, 2006.

[11] A. Spears, M. West, T. Collins, and A. Howard, “Determining Underwater Vehicle Movement from Sonar Data in Relatively Featureless Seafloor Tracking Missions,” in WACV 2014, pp. 1–8, IEEE, 2014.

[12] J. Shi and C. Tomasi, “Good features to track,” in Computer Vision and Pattern Recognition, 1994. Proceedings CVPR’94., 1994 IEEE Computer Society Conference on, pp. 593–600, IEEE, 1994.

[13] D. Nistér, "An efficient solution to the five-point relative pose problem," Pattern Analysis and Machine Intelligence, IEEE Transactions on 26, no. 6, pp. 756-770, 2004.

[14] O. Sorkine, “Least-squares rigid motion using SVD,” Technical notes, vol. 120, p. 3,2009.