An Underwater Slam Using Sonar, Visual, Inertial, and Depth Sensor

2019 IROS

Paper Reading

Allison Wong

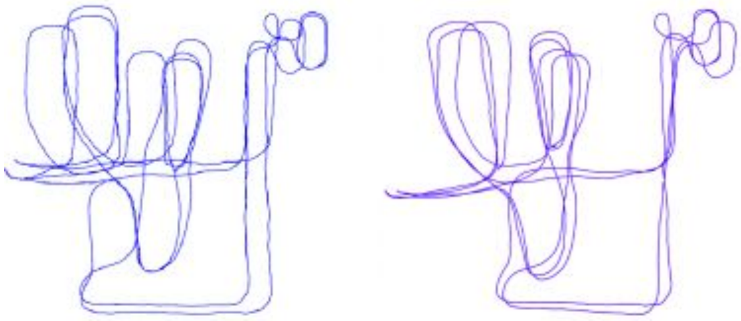
8/25/20

Problem

- Underwater navigation algorithms are based on DVL, USBL, and sonar
    - Expensive
    - Less accessible
- Visual Odometry (VO)/Visual Inertial Odometry (VIO)
    - Cameras are significantly cheaper and easily accessible
    - Abundance of work in indoor/outdoor environments
    - Visibility dependent on water turbidity, light attenuation, and color attenuation
    - Lose tracking
- **Extend current available open sourced VIO packages by infusing a sonar for underwater applications, and investigate the limitations and capabilities**

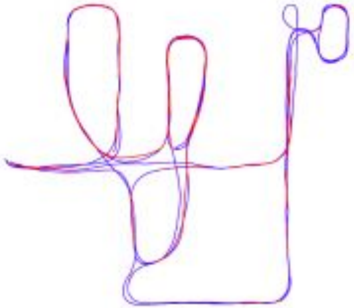
Current solutions

- Vision based state estimation techniques
  - Stereo or monocular cameras that are indirect (feature-based) or direct methods
- Focus on VIO and feature based method state estimation systems
- ORB-SLAM
  - Loop closure - bag of words (BoW)
  - g2o for nonlinear optimization
  - Initializes underwater
- OKVIS
  - No loop closure and batch optimization
  - Initializes underwater
- VINS-Mono
  - Loop closure – discrete bag of words 2 (dBoW2)
  - Ceres solver for nonlinear optimization
  - Struggles to initialize underwater



(a) Trajectory of OKVIS.

(b) Trajectory of VINS-Mono without loop closure.



(c) Trajectory of VINS-Mono with relocalization and loop closure. Red lines indicate loop detection.

Proposed method

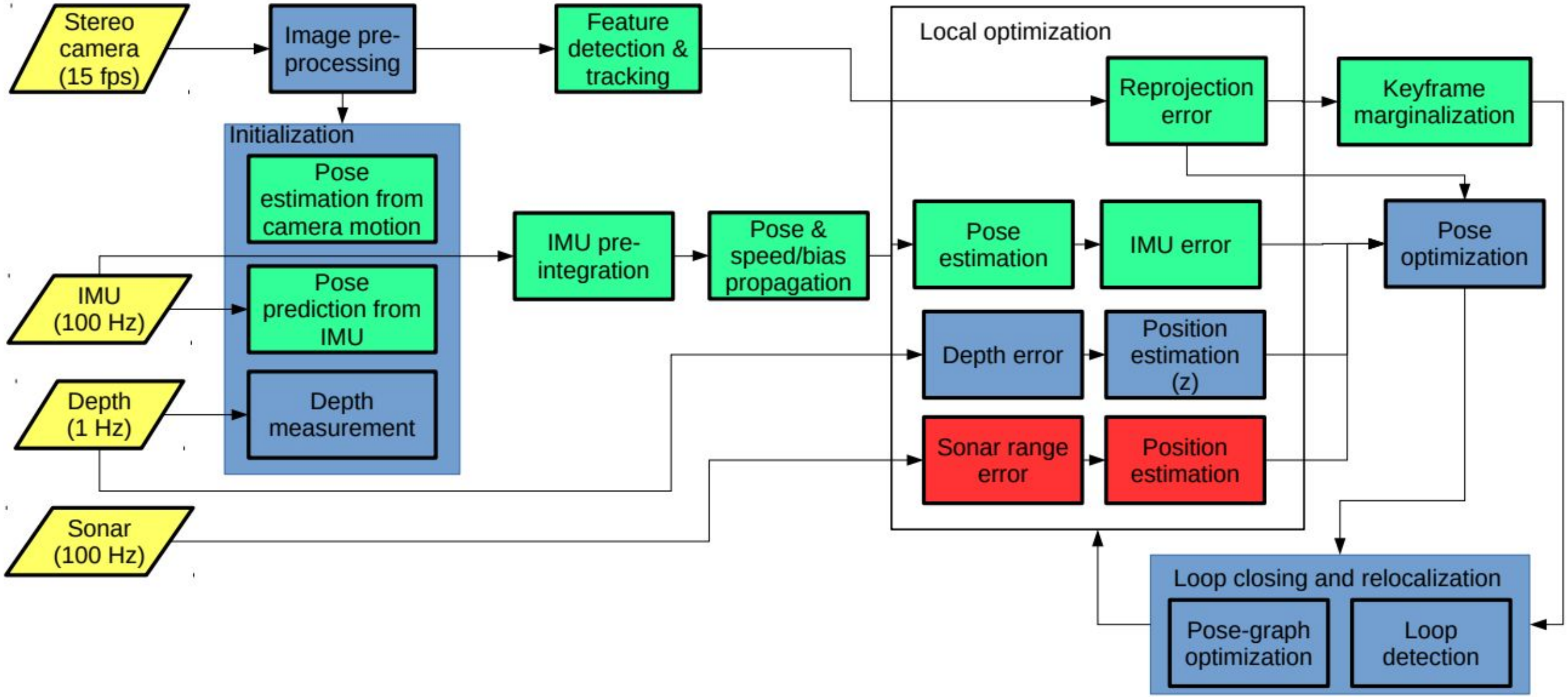
- **SVIn2:** SLAM system that fuses acoustic (mechanical scanning profiling sonar), visual (stereo camera), inertial (IMU), and depth (pressure sensor) data that is adaptable and applicable for any underwater vehicle using OKVIS's framework
  - Image enhancement technique targeted for the underwater domain
  - Depth measurements in the optimization process
  - Loop closure capabilities
  - Robust initialization

Contribution

- Developed a robust state estimator using an open sourced VIO (OKVIS) that is capable of being used underwater

Software Architecture

Legend
Yellow: sensor input
Green: OKVIS components
Red: previous work
Blue: new contribution



Initialization: Two Step Scale Refinement

- First refinement: depth data used to refine initial scale factor from stereo camera

$$ {}_W p_{zD} = s_1 * {}_W p_{zC} + {}_W \mathbf{R}_{zC} \, {}_C \mathbf{p}_D $$

Transformation between camera *C* and depth *D*

- Second refinement: Above eq. aligned w/ IMU pre-integral values

$$ {}_W \mathbf{p}_I = s_2 * {}_W \mathbf{p}_{r1C} + {}_W \mathbf{R}_C \, {}_C \mathbf{p}_I $$

Transformation between camera *C* and IMU *I*

- Substituting above into state prediction into IMU pre-integration terms, least squares problem:

$$ \min_{\boldsymbol{\chi}_S} \sum_{i \in K} \left\| \hat{\boldsymbol{z}}_{S_i}^{i+1} - \mathbf{H}_{S_i}^{i+1} \boldsymbol{\chi}_S \right\|^2 $$

$$ \hat{\boldsymbol{z}}_{S_i}^{i+1} = \begin{bmatrix} \hat{\boldsymbol{\alpha}}_{I_i}^{i+1} - {}_I\mathbf{R}_W^i \, {}_W\mathbf{R}_C^{i+1} \, {}_C\mathbf{p}_I^{i+1} + {}_I\mathbf{R}_C^i \, {}_C\mathbf{p}_I^i \\ \hat{\boldsymbol{\beta}}_{I_i}^{i+1} \end{bmatrix} $$

and

$$ \mathbf{H}_{S_i}^{i+1} = \begin{bmatrix} -\mathbf{I}\Delta t_i & \mathbf{0} & -\frac{1}{2} {}_I\mathbf{R}_W^i \Delta {t_i}^2 & {}_I\mathbf{R}_W^i ({}_W\mathbf{p}_{r1C}^{\,i+1} - {}_W\mathbf{p}_{r1C}^{\,i}) \\ -\mathbf{I} & {}_I\mathbf{R}_W^i \, {}_W\mathbf{R}_I^{i+1} & -{}_I\mathbf{R}_W^i \Delta t_i & \mathbf{0} \end{bmatrix} $$

Tightly Coupled Non-Linear Optimization with Sonar Visual Inertial Depth (SVIND) measurements

Cost function

$$J(\mathbf{x}) = \sum_{i=1}^{2}\sum_{k=1}^{K}\sum_{j\in\mathcal{J}(i,k)} \mathbf{e}_r^{i,j,k^T}\mathbf{P}_r^k\mathbf{e}_r^{i,j,k} + \sum_{k=1}^{K-1}\mathbf{e}_s^{k^T}\mathbf{P}_s^k\mathbf{e}_s^k + \sum_{k=1}^{K-1}\mathbf{e}_t^{k^T}\mathbf{P}_t^k\mathbf{e}_t^k + \sum_{k=1}^{K-1} e_u^{k^T} P_u^k e_u^k \quad (3)$$

All errors are added to the Ceres Solver

e_r: reprojection error

e_s: IMU error

e_t : sonar error

e_u : depth error

P_r^k: visual landmarks information matrix

P_s^k: IMU information matrix

P_t^k: Sonar range information matrix

P_u^k : Depth measurement information matrix

Loop Closing and Relocalization

Loop Closure

- DBoW2
  - keypoints detected during local tracking are used to build the BoW database
  - No new features will be detected in the loop closure step
- Outlier rejection using RANSAC

Loop detection

- Additional optimization step taken with matched landmarks with loop candidate to calculate sonar error and reprojection error

$$J(\mathbf{x}) = \sum_{i=1}^{2}\sum_{k=1}^{K}\sum_{j\in Loop(i,k)} \mathbf{e}_r^{i,j,k^T}\mathbf{P}_r^k\mathbf{e}_r^{i,j,k} + \sum_{k=1}^{K-1}\mathbf{e}_t^{k^T}\mathbf{P}_t^k\mathbf{e}_t^k$$

- After loop detection, 6DoF pose graph optimization is executed to optimize relative constraints between poses to correct drift
- Two poses (T_i and T_j) or current keyframe in current window *i* and keyframe *j*

Error term between keyframes *i* and *j* in tangent space

$$\mathbf{e}^{i,j}_{\mathbf{x_p},\mathbf{x_q}} = \Delta\mathbf{T}_{ij}\hat{\mathbf{T}}_i\hat{\mathbf{T}}_j^{-1}$$

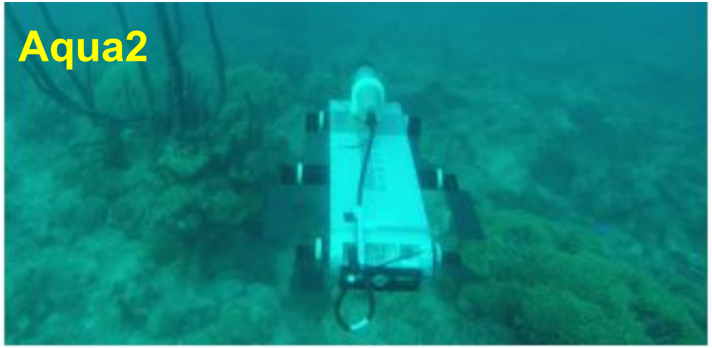
Cost Function

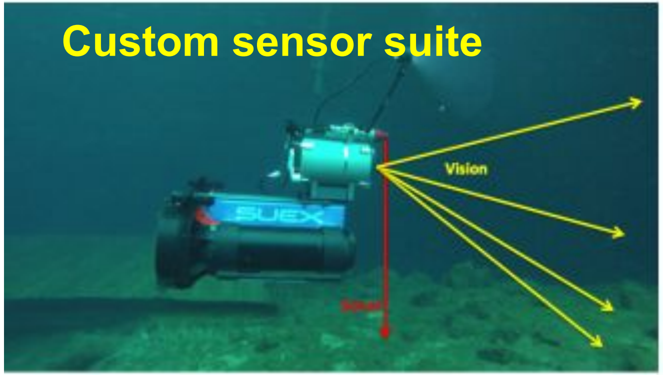
$$\begin{aligned} J(\mathbf{x_p},\mathbf{x_q}) &= \sum_{i,j} {\mathbf{e}^{i,j}_{\mathbf{x_p},\mathbf{x_q}}}^T \mathbf{P}^{i,j}_{\mathbf{x_p},\mathbf{x_q}} \mathbf{e}^{i,j}_{\mathbf{x_p},\mathbf{x_q}} \\ &+ \sum_{(i,j)\in Loop} \rho({\mathbf{e}^{i,j}_{\mathbf{x_p},\mathbf{x_q}}}^T \mathbf{P}^{i,j}_{\mathbf{x_p},\mathbf{x_q}} \mathbf{e}^{i,j}_{\mathbf{x_p},\mathbf{x_q}}) \end{aligned}$$

Evaluation Method

Validate proposed approach against:

1. State of the art system on the EuRoC micro aerial vehicle public dataset (proposed loop closure and initialization works on land, disabling sonar and depth measurements in system)
2. Custom sensor suite dataset containing varying underwater conditions
3. Aqua2 underwater vehicle dataset (visual, inertial, and depth only)
4. Compared against other methods
    1. VINS-Mono
    2. OKVIS

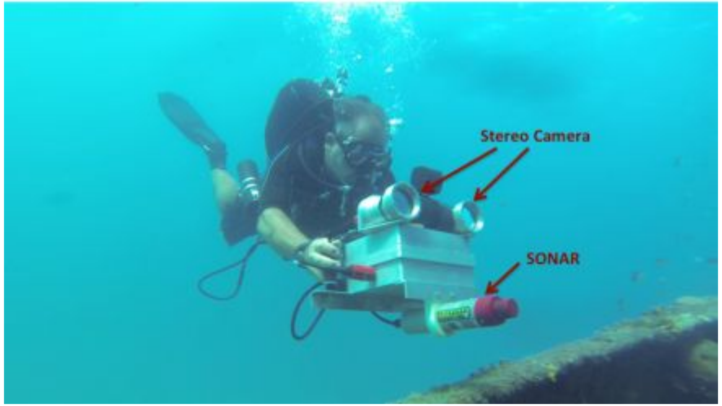




Custom Sensor Suite Overview

Sensor Suite

- 2 x IDS UI-3251LE cameras
- 1 x IMAGENEX 831L Sonar,
- 1 x Microstrain 3DM-GX4-15 IMU,
- 1 x Bluerobotics Bar30 pressure sensor
- 1 x Intel NUC



Cameras

- 15 fps
- 1600x1200 pixels

Sonar

- 360
- 0.9 angular resolution

IMU

- 100 Hz frequency

Depth sensor

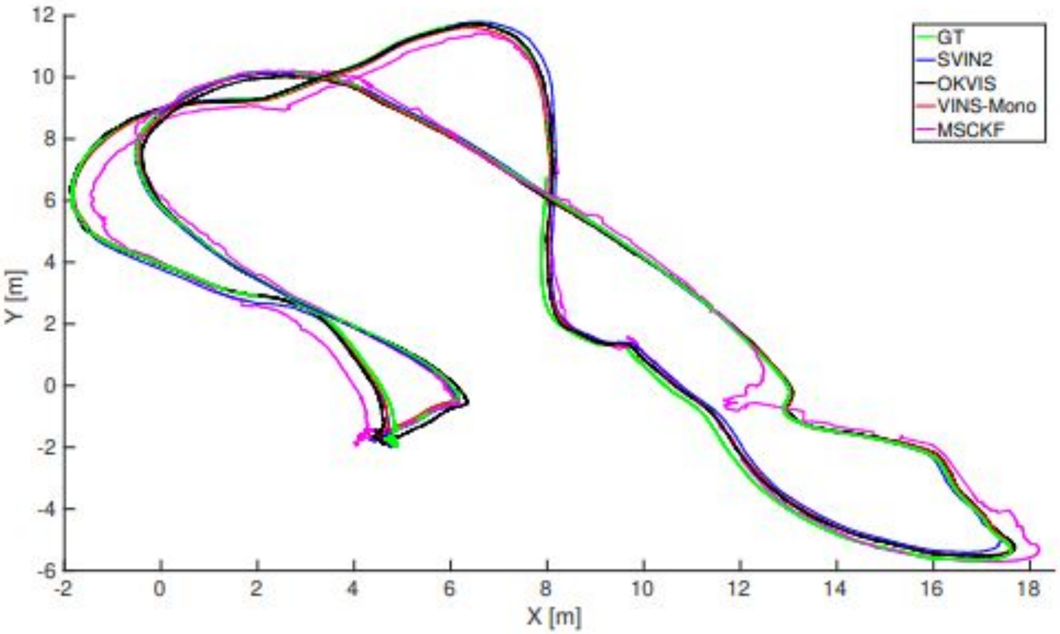
- 15 Hz frequency

Not on AUV, but hardware is compatible with the AquaAUV

A) Validation on EuRoC Standard Dataset

THE BEST ABSOLUTE TRAJECTORY ERROR (RMSE) IN METERS FOR EACH MACHINE HALL EUROC SEQUENCE.

	SVIn2	OKVIS(stereo)	VINS-Mono	MSCKF
MH 01	0.13	0.15	0.07	0.21
MH 02	0.08	0.14	0.08	0.24
MH 03	0.07	0.12	0.05	0.24
MH 04	0.13	0.18	0.15	0.46
MH 05	0.15	0.24	0.11	0.54



- Ran estimator on dataset 5 times, best run was chosen (least RMSE)
- Reduced RMSE compared to OKVIS
- VINS-Mono performed the best

B) Underwater Datasets

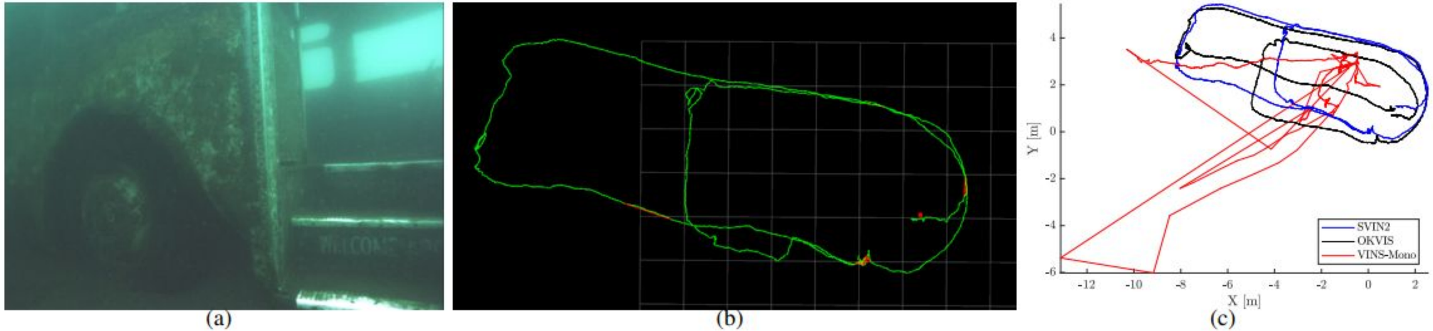


Fig. 6. (a) Submerged bus, Fantasy Lake, NC, USA; trajectories from SVIn2 with all sensors enabled shown in rviz (b) and aligned trajectories from SVIn2 with Sonar and depth disabled, OKVIS, and VINS-Mono (c) are displayed.

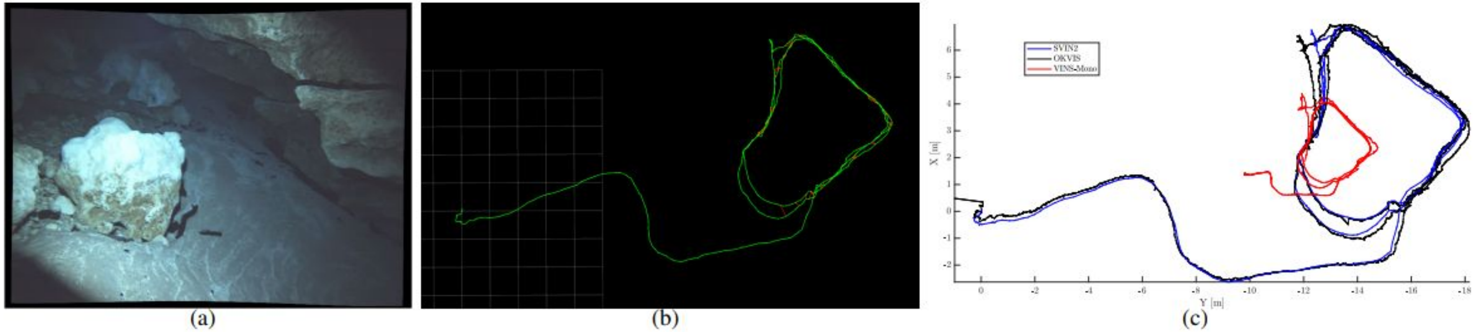
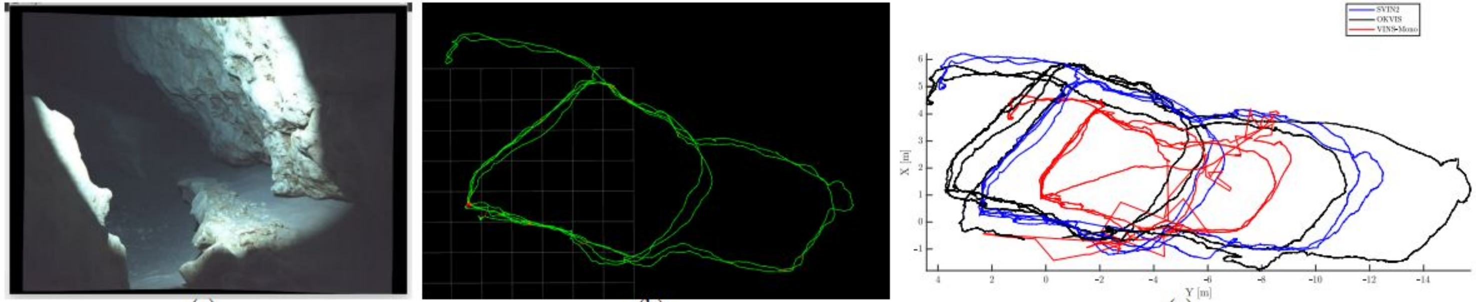


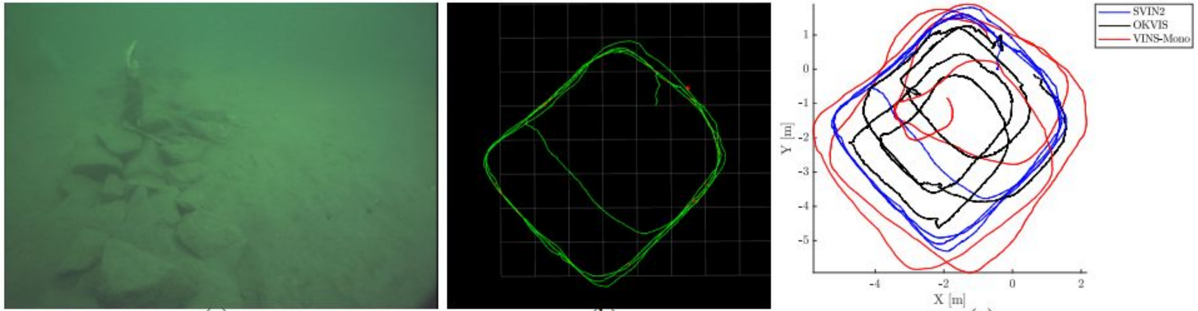
Fig. 7. (a) Cave environment, Ballroom, Ginnie Springs, FL, USA, with a unique loop; trajectories from SVIn2 with all sensors enabled shown in rviz (b) and aligned trajectories from SVIn2 with Sonar and depth disabled, OKVIS, and VINS-Mono (c) are displayed.

B) Underwater Datasets



(a) (b) (c)

Fig. 8. (a) Cave environment, Ballroom, Ginnie Springs, FL, USA, with two loops in different areas; trajectories from SVIn2 with all sensors enabled shown in rviz (b) and aligned trajectories from SVIn2 with Sonar and depth disabled, OKVIS, and VINS-Mono (c) are displayed.



(a) (b) (c)

Fig. 9. (a) Aqua2 in a fake cemetery, Lake Jocassee, SC, USA; trajectories from SVIn2 with visual, inertial, and depth sensor (no sonar data has been used) shown in rviz (b) and aligned trajectories from SVIn2 with Sonar and depth disabled, OKVIS, and VINS-Mono (c) are displayed.

Thoughts

- Would be nice if there is a 3D state estimator comparison of all algorithms
- Comparison against “traditional” underwater state estimation system
- Assuming planar features