Robust Motion Estimation with RBG-D Cameras

Robert C. Leishman[*] and Daniel Koch[†] and Timothy W. McLain[‡]

Brigham Young University, Provo, UT 84604, USA

Estimating vehicle motion using vision sensors in real time has been greatly explored in the past few years due to speed improvements and advances in computer hardware. Six degree of freedom motion estimation using vision information is desirable due to a vision sensors low cost, low power requirements and light weight and for the quality of the solutions that can be obtained using few assumptions about the environment. However, cameras have the downside of not providing good estimates when visual features are sparse or not available. Also, there are problems with changes in lighting and when light is low or unavailable. Laser scanners have been shown to be robust in these situations. We view an RGB-D sensor as providing three complimentary modalities that are useful for providing motion estimation solutions: a monocular camera, a 3D point cloud and the combination providing RGB-D information. Obviously motion estimates produced using the combined sensor information are best. However, there are times when information from both sensors is not available. The monocular camera remains useful when depth information is absent or insufficient, like in a large room, down a long hallway or outdoors. The 3D point cloud may still be available when there is insufficient light to utilize the RGB image. The approach described in this work seeks to take advantage of all three of these sensor modalities to provide a more robust motion estimation solution.

I. Introduction

Estimating vehicle motion using vision sensors in real time has been greatly explored in the past few years due to speed improvements and advances in computer hardware. Six degree of freedom (6DOF) motion estimation using vision information is desirable due to a vision sensors low cost, low power requirements and light weight and for the quality of the solutions that can be obtained using few assumptions about the environment. Motion estimates using vision[1,2] can be obtained with monocular cameras, stereo rigs, and recently, RGB-D cameras, which provide many of the benefits of vision and laser scanners combined in a synchronized package.

We are interested in developing vision-based autonomy specifically for small quadrotor vehicles that operate primarily in indoor environments. Motion estimation is a critical element for this autonomy, especially as GPS is typically unavailable. These small aerial vehicles present challenging constraints such as stringent payload limitations and fast vehicle dynamics. Consequently, motion estimation solutions that are robust and fast are especially important to the long term autonomy and flexibility in mission roles of these autonomous aerial systems.

A monocular camera is lightweight and inexpensive. It has been shown to provide valid solutions for motion estimation,[3,4] even on quadrotors.[5] However, challenges with scale and being in environments with few features make robust motion estimation difficult with only a monocular camera.

Stereo vision provides the advantage of depth information at close ranges and can be used to provide highly accurate motion estimates.[6,7] However, additional processing time to compute depth and required synchronization for the images are obstacles which have proven difficult for implementation onboard a quadrotor.[8,9] Environments with few features also degrade or eliminate motion estimation with stereo.[10]

Camera information in general has the downside of not providing good estimates when visual features are sparse or not available, i.e. when viewing blank walls or outdoor terrain. Also, there are problems with

[*] PhD Candidate, Mechanical Engineering.

[†] Undergraduate, Mechanical Engineering.

[‡] Professor and Chair, Mechanical Engineering, Associate Fellow AIAA.

changes in lighting and when light is low or unavailable. Laser scanners or ICP algorithms using RGB-D sensors have been shown to be robust in these situations.

RGB-D has the benefits of stereo in indoor environments and has recently been used for visual motion estimation,[11,12] with successful implementations onboard aerial vehicles.[13,?] Solutions developed without aerial implementations in mind generally use an iterative closest point (ICP) algorithm[14] on the 3D point cloud for further refinement of the estimated motion.[11,12] However, an ICP algorithm is too computationally expensive for the fast motion estimation solutions that are required by an aerial platform.

Some researchers have worked to combine the benefits of vison and laser by utilizing both sensors.[15,9] These researchers seek to bring the benefits of both to provide more robust solutions. However, utilizing more than one sensor incurs additional processing, complexity and weight.

More robust approaches are proposed with RGB-D cameras for ground vehicle motion.[16,?] A visual odometry[17] (VO) technique is paired with an ICP algorithm, as is typically done. However, three different outcomes are possible. When error is low, the VO solution is used. When there is too much error in the VO but an approximate solution is available, the VO solution initializes an ICP solution. Finally, when insufficient features are detected on the image, the ICP algorithm generates a solution using only the point cloud data. Essentially two modalities of the RGB-D sensor are used to provide a more robust solution.

We view an RGB-D sensor as providing three modalities that are useful for providing motion estimation solutions: a monocular camera, a 3D point cloud and the combination providing RGB-D information. Obviously motion estimates produced using the combined sensor information are best. However, there are times when information from both sensors is not available. The monocular camera remains useful when depth information is absent or insufficient, like in a large room, down a long hallway or outdoors. The 3D point cloud may still be available when there is insufficient light to utilize the RGB image. The approach described in this work seeks to take advantage of all three of these sensor modalities to provide a robust motion estimation solution.

The remainder of the article is outlined as follows. We provide additional details on the robust motion estimation approach in Section II. A 3D view matching algorithm is described in Section III. The 2D version of the view matching is presented in Section IV. We detail a 2D scan matching approach in Section V. Hardware results are presented in Section VI. We finish with conclusions and recommendations for future work in Section VII.

II. Robust Motion Estimation

An RGB-D sensor essentially provides three separate modalities useful for motion estimation. The proposed algorithm completes view matching using the images and scan matching with the point cloud. View matching[?] is similar to VO, where the current image is compared to another image to find the relative 6DOF change in pose. However in visual odometry, consecutive images are compared. We compare each current image to a reference image, called a *keyframe*. Once the overlap between the current image and the keyframe becomes too small and the accuracy of the pose estimates is reduced, we declare a new keyframe and continue. We utilize view matching for both 3D-to-3D and 2D-to-2D point correspondences.

The scan matcher we utilize provides a 3DOF change in pose between a current scan and a local map of scans, where the local map is based on the keyframe image. Currently the scan matching uses only a band of the 3D point cloud as a 2D scan.

The algorithm is able to adapt and focus priority on the sensor modality that is likely to provide the best information. The percentage of the RGB image with valid depth information is measured on each current image. When there is a significant portion of the image without depth, like in a hallway, monocular view matching is used alongside 3D view matching. When a large majority of the image does not have depth available, monocular view matching replaces the 3D view matching until depth information is restored. When an insufficient number of features are detected on the image yet depth information is available, the scan matcher provides the pose estimates. This allows the motion estimation to maximize the information available and continue providing estimates where a single sensor would fail. It also reduces the complexity and weight that accompany the use of multiple sensors.

We implement the algorithm using OpenCV[a] and ROS[b] open source tools. The algorithm is capable of running at about 28Hz, which is near the frame rate of the ASUS Xtion Pro Live, however we usually run

[a] code.opencv.org

[b] www.ros.org

it at 15Hz as it provides a sufficient amount of information to obtain good pose estimates and this saves computation time for other processes.

II.A. Sensor Fusion

Currently, the output measurements from the motion estimation algorithm are fused with altimeter and IMU measurements to provide state estimates for a hexacopter aerial vehicle. This vehicle has fast dynamics and consequently requires quality, fast estimates to maintain flight and to avoid obstacles.

The sensor fusion is provided by a multiplicative extended Kalman filter (MEKF) CITE MEKF that has been designed specifically to function with a unique relative navigation approach CITE ICUAS. We update the filter using altimeter, accelerometer, motion-estimation position, and motion-estimation orientation measurements. The motion estimation measurements can come from the 3D view matching, 2D view matching and/or scan matching.

III. 3D View Matching

3D view matching is the process of estimating the 6DOF change in pose between a current image and a reference image with the associated depth information. We provide a brief summary of the algorithm here.

Sets of color and depth images are sent to the algorithm. The first image set sent is designated as the keyframe image pair and all following image pairs are compared to this set until a new keyframe image is assigned. This occurs once the camera has moved 0.25 meters or 10 degrees in yaw from the location where the keyframe image was taken.

On each image FAST features CITE? and BRIEF descriptors CITE? are extracted and the feature positions corrected using the distortion information of the camera. A mask is used so that features may only be found in areas with valid depth information. The image is also binned to allow an even dispersion of features across the image. The number of features is currently capped at 750 features. The 3D point location $\mathbf{p} = (X \ Y \ Z)^\top$ for the 2D image feature $\bar{\mathbf{p}} = ()x \ y)^t op$ is found by looking up the depth Z in the depth image and using the projection equations

$$X = \frac{(x - c_x)Z}{f_x} \tag{1}$$

$$Y = \frac{(y - c_y)Z}{f_y}, \tag{2}$$

where c_x, c_y, f_x, and f_y are the intrinsic camera calibration parameters for the image center and focal points.

Next, correspondence between the current image features and the keyframe features are estimated using forward and backward constrained brute-force searches in a mutual consistency check.[1] The corresponding features are passed into RANSAC,[18] which is then employed to find a pose motion estimate while eliminating outliers. We use a three point singular value decomposition (SVD) algorithm based on[19] as the motion model in RANSAC. The solution estimate provides the 6DOF rotation and translation between the keyframe and the current coordinate frames and is of the form

$$\mathbf{p}^c = \mathbf{R}^c_{key}\mathbf{p}^{key} + \mathbf{T}^c.$$

Where $\mathbf{p}^c$ and $\mathbf{p}^{key}$ are the current and keyframe 3D feature position vectors, $\mathbf{R}^c_{key}$ rotates points expressed in the keyframe coordinate frame into the current image coordinate frame, and T^c is the origin of the keyframe coordinate frame expressed in the current image coordinate frame.

Inliers are found for the sample solution by re-projecting the 3D keyframe features onto the current image using the sample solution and the intrinsic calibration parameters. To be considered a valid inlier, the pixel error distance between the locations of current image feature and the reprojected keyframe feature must be smaller than some threshold. Checking the error on the image provides improved results over solutions evaluated by error in 3D positions.

The solution estimate with the highest inlier count is returned by the RANSAC algorithm. Note that we do not use the inlier points to calculate a least-squares solution. We have found by comparing estimates to truth from a motion capture system that using the best solution typically provides more accurate results than a least squares solution.

III.A. Covariance Estimate

An estimate of the covariance of the relative pose change is required for the sensor fusion step. Currently we use a fixed covariance that we found experimentally using truth information from a motion capture system. We do scale up the covariance when the number of inliers falls below a threshold. Not only is this method quite fast, we have also been able to obtain quality estimates from the sensor fusion algorithm using this approach CITEMEKF.

IV. 2D View Matching

View matching with 2D points is the process of estimating the 6DOF transformation between the current and keyframe images without using the depth information. This process is similar to that of the 3D view matching.

FAST features and BRIEF descriptors are also extracted from each image and the feature locations are undistorted. But the mask that is used only allows features to be found on portions of the image without depth information. Unlike the 3D method, we require normalized calibrated image coordinates for the features. The same brute-force searches with a mutual consistency check are employed to find estimates of the matching features between the keyframe and current images.

RANSAC is also employed to find the best solution and eliminate outliers. The motion model employed in RANSAC is the five-point algorithm.[20] We acquired a basic open source implementation of this method[c] and modified it to work with RANSAC. The five-point algorithm uses a minimal subset of five corresponding points to estimate the essential matrix $\hat{\mathbf{E}}$. Additional details on the properties of the essential matrix can be found in CITEMULTIPLEVIEWGEOMETRY. For further refinement, we use the closest actual essential matrix $\mathbf{E}_{closest}$, by the L_2 norm, to $\hat{\mathbf{E}}$ using

$$\mathbf{E}_{closest} = \mathbf{U} \left\{ \begin{matrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{matrix} \right\} \mathbf{V}^{\top}, \tag{3}$$

where $\mathbf{U}$ and $\mathbf{V}$ are from the SVD of $\hat{\mathbf{E}}$, as detailed in Ref. 21.

We use Sampson's error to find inliers in the RANSAC routine for each sample solution. The error for a keyframe feature $\mathbf{x}$ and a current image feature $\mathbf{x}'$ and the current sample solution $\mathbf{E}$, based on (3), is

$$err((\mathbf{x}, \mathbf{x}'), \mathbf{E}) = \frac{\mathbf{x}'^{\top}\mathbf{E}\mathbf{x}}{\sqrt{(\mathbf{x}'^{\top}\mathbf{E})_0^2 + (\mathbf{x}'^{\top}\mathbf{E})_1^2 + (\mathbf{E}\mathbf{x})_0^2 + (\mathbf{E}\mathbf{x})_1^2}}. \tag{4}$$

Where the subscripts $_0$ and $_1$ on a vector indicate the first and second elements of that vector.

Using $\mathbf{E}_{closest}$, we estimate the projection matrix $\hat{\mathbf{P}}$, which contains the rotation and translation between the keyframe and current image. Because of the lack of depth information, the translation is known only up to an unknown scale factor.

The scale is determined using a ratio of estimates of 3D features. With each image pair, keyframe and current image, we can estimate 3D locations for the features that are inliers from the estimated solution. Following Ref. 1, the 3D feature locations from at least two 3D features X_{k-1} and X_k, from two consecutive image pairs, pair i and pair j, are compared in a ratio

$$r = \frac{\|X_{k-1,i} - X_{k-1,j}\|}{\|X_{k,i} - X_{k,j}\|}. \tag{5}$$

We compare multiple 3D points and compute the mean for a more robust ratio. We use the mean as we are only including inlier points. The translation in P is multiplied by the ratio r for the appropriate scale.

IV.A. Covariance Estimate

We also employ a covariance estimated by comparing truth from motion capture data to the motion estimates produced by the approach. This estimate is scaled up when an insufficient number of inliers are found in the solution.

[c] `nghiaho.com/?p=1675`

V. Scan Matching

While scan-matching algorithms were originally developed for use with laser scan matchers **citations? or even say this?**, we have been able to successfully implement such an algorithm using depth information from the RGB-D data. A laser scan is simulated by taking a thin slice of the depth information and feeding to the algorithm as a laser scan. Our current implementation of provides 2D estimates (x translation, y translation, and yaw) of the pose of the rotorcraft.

The scan matcher we chose to implement is the point-to-line metric ICP (Iterative Closest/Corresponding Point) matcher described in Ref. 22. We are implementing it using an **modified** open-source ROS wrapper[d] that uses the code provided by the author of Ref. 22[e].**how much detail do we want on this?** A general ICP scan matcher estimates the change in pose of the vehicle, $\mathbf{x}$, between two scans by comparing an incoming scan, $\mathbf{y}_t$, to a reference or keyframe scan, $\mathbf{y}_{t-1}$. A reference surface, S_{ref}, is created from the reference scan by connecting the scan points with a poly-line. A set of correspondences $\{(\mathbf{p}_k, \mathbf{q}_k)\}$ is then calculated, where $\mathbf{p}_k$ is a point from the scan $\mathbf{y}_t$, and where $\mathbf{q}_k$ is the projection of $\mathbf{p}_k$ onto S_{ref} after $\mathbf{p}_k$ has been rotated and translated by $\mathbf{x}$, denoted by

$$\mathbf{q}_k = \Pi\left(S_{\text{ref}}, \mathbf{T}_{\mathbf{x}}\mathbf{p}_k\right)$$

where $\mathbf{T}_{\mathbf{x}}\mathbf{p}_k$ denotes the rotation and translation of $\mathbf{p}_k$ by $\mathbf{x}$. An incremental solution for $\mathbf{x}$ is then found to minimize the error function

$$J\left(S_{\text{ref}}, \{\mathbf{p}_k\}, \mathbf{x}\right) = \sum_k \|\mathbf{T}_{\mathbf{x}}\mathbf{p}_k - \mathbf{q}_k\|^2$$

talk about the difference between PLICP and regular ICP?

In order to reduce the drift in the global pose estimate, the first scan is declared as a keyframe scan, and then all subsequent scans are compared to this keyframe scan as opposed to comparing each scan to the one immediately preceding it. Once a predetermined threshold of rotation or linear translation has been reached and there are no longer enough correspondences between the two scans to calculate an accurate estimate, the current scan is declared as the new keyframe scan, and the algorithm continues.

Because the ICP scan matching algorithm is sensitive to large errors in the initial guess of the change in pose between scans, its convergence can be improved by using data from other sensors to provide a better initial guess22**check citation**. In our implementation, data from the IMU is used to accomplish this.

V.A. Covariance Estimate

An accurate estimate for the covariance of the pose estimate provided by the scan matching algorithm can be computed in closed form, as described in Ref. 23. The covariance of the estimate, $\hat{\mathbf{x}}$, of $\mathbf{x}$ can be approximated as[23]

$$\text{cov}\left(\hat{\mathbf{x}}\right) \approx \left(\frac{\partial^2 J}{\partial \mathbf{x}^2}\right)^{-1} \frac{\partial^2 J}{\partial \mathbf{z}\partial \mathbf{x}} \text{cov}\left(\mathbf{z}\right) \frac{\partial^2 J}{\partial \mathbf{z}\partial \mathbf{x}}^T \left(\frac{\partial^2 J}{\partial \mathbf{x}^2}\right)^{-1}$$

where $\mathbf{z}$ is the sensor measurement, which in this case is the depth information from the camera. This calculation was implemented using the author's code[f].

VI. Results

Results for each algorithm comparing its performance to truth are first presented. These results are all taken from rotorcraft flights within a room equipped with a Motion Analysis motion capture system. Afterwards, we provide results without truth data that demonstrate the robustness of the motion estimation approach which are completed using the camera in hand.

In the flight results we utilize a Mikrokopter hexacopter vehicle that carries a Intel i7 processor computer on which all the processing is computed onboard. The computer is running Ubuntu 12.04 Linux and all the applications are implemented in C++ and connected together using ROS[g]. RGB-D imagery and point clouds are provided by an ASUS Xtion Pro Live at 15 Hz. Truth data is recorded at 100 Hz with sub-millimeter and sub-degree precision.

[d] www.ros.org/wiki/laser_scan_matcher

[e] http://andrea.caltech.edu/research/robot-perception/plicp/

[f] http://andrea.caltech.edu/research/robot-perception/icpcov/

[g] www.ros.org

VI.A. 3D View Matching Results

We demonstrate two different types of results for the 3D view matching algorithm. The first are relative results, meaning that we express the truth information in the keyframe coordinate frames to get an idea of how accurate the relative transformations are between the current and keyframe images. The second set of results show the global performance of the visual odometry as is typically done.

VI.A.1. Relative Results

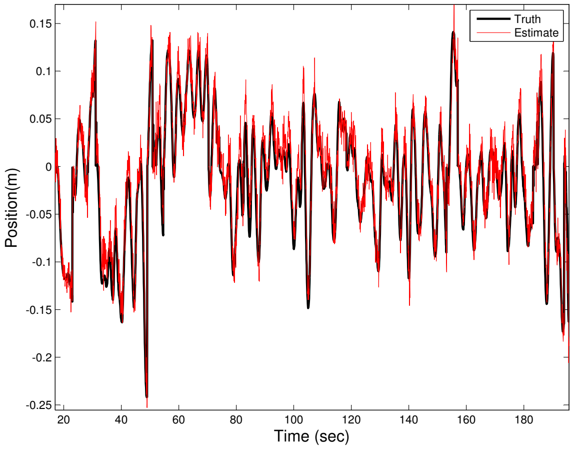

Figure 1. Relative camera $\vec{z}$ (out of plane) position comparison between truth and estimates. The discontinuities in the plots are due to new nodes being created, causing the truth and the estimates to jump to the new relative position. We express the global truth from the motion capture in the relative node coordinate frame for the comparison of these results. Results for the camera $\vec{x}$ and $\vec{y}$ positions are similar.

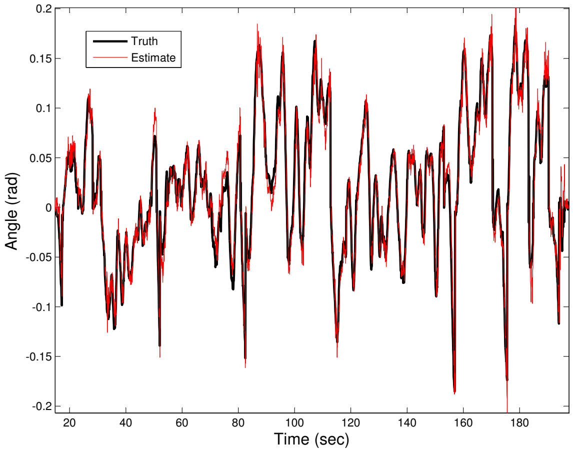

Figure 2. Comparison between truth and estimates for the rotation about the camera $\vec{x}$ axis (camera pitch angle, which is equivalent to the vehicle yaw angle because of the change in axes). Again, the discontinuities in the plots are due to new nodes being created, causing the truth and the estimates to jump to the new relative position. We express the global truth from the motion capture in the relative node coordinate frame for the comparison of these results. Results for the camera roll and yaw positions are similar.

Figure 1 presents the camera z axis portion of the relative transformations between the current and keyframe images. The discontinuities are due to changing keyframes. Figure 2 presents the results for the rotation about the camera $\vec{x}$ axis, the camera pitch angle. This angle corresponds to the vehicle yaw angle because of the change in axes. The algorithm actually outputs the change in rotation as a unit quaternion, which we have converted to Euler angles for ease of comparison. Table 1 provides the RMS error in the relative transformations over the flight. These values are representative of results that we routinely achieve in the motion capture environment.

Table 1. Presented are the RMS errors of view matching estimates produced during a flight. The motion capture data is used as truth for the computation and is expressed in the relative coordinate frame. The estimates were produced in real time during the flight.

RMS Error in Motion Estimates	
Transformation	**RMS Error**
camera $\vec{x}$ position	0.033 (m)
camera $\vec{y}$ position	0.041 (m)
camera $\vec{z}$ position	0.041 (m)
camera roll angle	0.020 (rad)
camera pitch angle	0.017 (rad)
camera yaw angle	0.013 (rad)

VI.A.2. Global Results

The global results are also produced from a hexacopter flight within the motion capture environment. However, we express the relative transformations in the motion capture coordinate frame (global) and compare these estimates to the truth. These results demonstrate the global drift that occurs due to the small errors in the relative transformations shown above.

VI.B. 2D View Matching Results

VI.C. 2D Scan Matching Results

VI.D. Combination Results

VII. Conclusions and Future Work

Acknowledgments

NOTE: Mention the Center! This work was supported through the DoD SMART Scholarship program.

References

[1] Scaramuzza, B. D. and Fraundorfer, F., "Visual Odometry Part I: The First 30 Years and Fundamentals," *Robotics & Automation Magazine*, , No. December, 2011, pp. 80—-92.

[2] Scaramuzza, D. and Fraundorfer, F., "Visual odometry Part 2: Matching, Robustness, Optimization, and Applications," *Robotics & Automation Magazine*, , No. June, 2012, pp. 78—-90.

[3] Davison, A. J., Reid, I. D., Molton, N. D., and Stasse, O., "MonoSLAM: real-time single camera SLAM." *IEEE transactions on pattern analysis and machine intelligence*, Vol. 29, No. 6, June 2007, pp. 1052–67.

[4] Klein, G. and Murray, D., "Parallel tracking and mapping for small AR workspaces," *Proceedings of the 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, ISMAR '07, IEEE Computer Society, Washington, DC, USA, 2007, pp. 1–10.

[5] Weiss, S., Achtelik, M., and Lynen, S., "Real-time onboard visual-inertial state estimation and self-calibration of mavs in unknown environments," *Proc. IEEE Int Robotics and Automation (ICRA) Conf*, 2012.

[6] Konolige, K., Agrawal, M., and Sola, J., "Large scale visual odometry for rough terrain," *International Symposium on Robotics Research, November*, 2007.

[7] Paz, L. M., Pinies, P., Tardos, J. D., and Neira, J., "Large-Scale {6-DOF} {SLAM} With Stereo-in-Hand," *IEEE_J_RO*, Vol. 24, No. 5, 2008, pp. 946–957.

[8] García Carrillo, L. R., Dzul López, A. E., Lozano, R., and Pégard, C., "Combining Stereo Vision and Inertial Navigation System for a Quad-Rotor UAV," *Journal of Intelligent & Robotic Systems*, Vol. 65, No. 1-4, Aug. 2011, pp. 1–15.

[9] Tomic, T., Schmid, K., Lutz, P., Kassecker, M., Mair, E., Grixa, I., Ruess, F., Suppa, M., and Burshka, D., "Toward a Fully Autonomous UAV: Research Platform for Indoor and Outdoor Urban Search and Rescue," *Robotics & Automation Magazine*, , No. SEPTEMBER, 2012.

[10] Fraundorfer, F., Heng, L., Honegger, D., Lee, G. H., Tanskanen, P., and Pollefeys, M., "Vision-Based Autonomous Mapping and Exploration Using a Quadrotor MAV," *IEEE Conference on Intelligent Robots and Systems*, 2012.

[11] Endres, F., Hess, J., Engelhard, N., Sturm, J., Cremers, D., and Burgard, W., "An evaluation of the RGB-D SLAM system," *2012 IEEE International Conference on Robotics and Automation*, IEEE, May 2012, pp. 1691–1696.

[12] Lui, W. L. D., Tang, T. J. J., Drummond, T., and Li, W. H., "Robust egomotion estimation using ICP in inverse depth coordinates," *2012 IEEE International Conference on Robotics and Automation*, May 2012, pp. 1671–1678.

[13] Huang, A. S., Bachrach, A., Henry, P., Krainin, M., Maturana, D., Fox, D., and Roy, N., "Visual Odometry and Mapping for Autonomous Flight Using an RGB-D Camera," *Int. Symposium on Robotics Research (ISRR)*, Flagstaff, Arizona, USA, 2011.

[14] Segal, A., Haehnel, D., and Thrun, S., "Generalized-{ICP}," *Robotics: Science and Systems*, 2009.

[15] Bachrach, A., Prentice, S., Roy, N., He, R., Winter, A. D., and Hemann, G., "RANGE: Robust autonomous navigation in GPS-denied environments," *Journal of Field Robotics*, Vol. 28, No. 5, Sept. 2011, pp. 644–666.

[16] Paton, M. and Kosecka, J., "Adaptive RGB-D Localization," *2012 Ninth Conference on Computer and Robot Vision*, May 2012, pp. 24–31.

[17] Nister, D., Naroditsky, O., and Bergen, J., "Visual odometry," *Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition CVPR 2004*, Vol. 1, 2004.

[18] Fischler, M. A. and Bolles, R. C., "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, Vol. 24, No. 6, 1981, pp. 381–395.

[19] Arun, K. S., Huang, T. S., and Blostein, S. D., "Least-squares fitting of two 3-D point sets," *Pattern Analysis and . . .* , Vol. 9, No. 5, May 1987, pp. 698–700.

[20] Hongdong, L. and Hartley, R., "Five-Point Motion Estimation Made Easy," *International Conference on Pattern Recognition*, 2006.

[21] Hartley, R. and Zisserman, A., *Multiple View Geometry in computer vision*, Cambridge University Press, 2nd ed., 2003.

[22] Censi, A., "An ICP variant using a point-to-line metric," *2008 IEEE International Conference on Robotics and Automation*, May 2008, pp. 19–25.

[23] Censi, A. and La, R., "An accurate closed-form estimate of ICP s covariance," *2007 IEEE International Conference on Robotics and Automation*, , No. April, 2007, pp. 10–14.