Investigation into the Effects of Transmission-channel Fidelity Loss in RGBD Sensor Data for SLAM

Jodie Wetherall [1], Matthew Taylor [2], Darren Hurley-Smith [1]
[1] University of Greenwich, Chatham Maritime, Kent, UK
[2] NIC Instruments Ltd, Folkestone, Kent, UK
J.C.Wetherall@greenwich.ac.uk

Abstract - **Simultaneous Location and Mapping (SLAM) is computationally expensive, and requires high-fidelity sensor data. This paper investigates the effects of transmission channel fidelity loss in Red-Green-Blue-Depth (RGBD) sensor data. A mobile robotic platform developed for Explosive Ordinance Disposal (EOD) is used, with a highly constrained data and video link to a base station which computes a SLAM solution. Experiments were conducted offline, using well known data-sets with ground truth data, and their results have been compared to determine the effect of fidelity loss under various multiplexing approaches with a constrained transmission channel.**

Keywords **– SLAM; mapping; computer vision; image processing**

I. INTRODUCTION

Explosive Ordinance Disposal (EOD) Unmanned Ground Vehicle (UGV) are a class of telepresence device that allows a remote operator to investigate, disarm or potentially destroy a suspected explosive device from a safe distance. Similar UGV systems are used in Search and Rescue (SAR) and environments that are too hazardous for direct human reconnaissance, for example in response to Chemical, Biological, Radiological or Nuclear (CBRN) accidents [1].

The platform used as a target in this paper is NIC Instruments LTD's First Responder EOD UGV. This platform consists of a differential track drive chassis, with an articulated manipulator arm. The UGV is controlled remotely via radio link with a rugged control station. Live video is streamed back from the UGV via analog video radio, allowing the operator to view the UGV surroundings via five cameras attached to the UGV chassis and manipulator arm.

Situational awareness in search and investigation scenarios can be invaluable. Using sensors to produce a 3D map of the environment around the UGV can allow the operator to better navigate through the environment. The 3D environment map can also be used to plot routes through the environment allowing for greater remote automation for the UGV.

The generation of the 3D maps requires high-fidelity sensor data and considerable computing power [2]. Classically this has necessitated a client-server or client-cloud based architecture, offloading the generation of maps from local sensor data to a remote host optimized for this computation.

This paper investigates the challenges of using a client-server architecture with a low-fidelity radio link, and seeks to identify alternative approaches for computing these maps locally on the UGV using Commercially Off The Shelf (COTS) hardware, using the low-fidelity link to provide a remotely rendered view only.

II. LITERATURE REVIEW

A. Simultaneous Location And Mapping (SLAM)

SLAM is the process of generating a model of the environment around a robot or sensor, while simultaneously estimating the location of the robot or sensor relative to the environment [3, 4]. SLAM has been performed in many ways, which can be categorized generally by their focus on localisation or environment mapping.

SLAM systems focused on localising the sensor accurately, relative to the immediate environment, make use of sparse sensor data to locate the sensor.

SLAM systems focused on map output use dense sensor output to create a high-fidelity 3D map of the environment, while using this data to also compute relative location of the sensor [5, 6]. Many modern SLAM algorithms combine both approaches, usually by extracting sparse features from the sensor and using these for efficiently computing the location of the sensor. This position is then used to construct a map from dense sensor data.

1) Range SLAM

SLAM computation using only distance from the sensor to the environment is common in many robot applications, using range sensors such as scanning laser range-finders [7], LiDAR and SONAR [8].

2) Visual SLAM

Visual sensors used for SLAM range from single-camera (typically called Mono-SLAM), stereo-camera, or n-camera. A specialist sub-type of camera that is gaining popularity is the depth camera. These cameras use structured laser light and multiple calibrated cameras to read colour image and distance from the environment in-front of the camera [9].

3) RGB-Depth (RGBD) SLAM

Depth cameras are able to provide dense sensor data in the form of 3D point clouds that can be used to construct detailed 3D representations of their field of view.

Because depth cameras also provide standard video images, efficient sparse feature SLAM can be used to quickly estimate the pose of the sensor, reducing the computational load of merging the dense point-cloud into the map.

RGBD SLAM [10] is a program built upon the Robotic Operating System (ROS) framework, and uses colour and depth image streams to estimate camera pose and construct a point-cloud map.

To provide a mechanism for benchmarking RGBD SLAM against alternate approaches, set of standardised datasets have are available and used to support this paper [11].

4) Common Architectures

a) Client-server and cloud model SLAM

Consumer products generally use a client-server or client-cloud model. This architecture consists of a mobile device with sensors, typically a consumer smart phone or tablet, and a remote server, or cluster of servers, with hardware optimised to efficiently compute SLAM solutions [1].

Sensor data is recorded by the client device, transmitted to the server for computation, with the results being relayed back to the client if required.

b) Client only model

A client-only model is typically used on embedded devices which have either a time-critical requirement on the SLAM computation, or have constraints on data transmission to a remote server. An example of these are automated road vehicles, which must compute SLAM in real-time.

c) Other notable examples

An example of a communication constrained SLAM system is the Mars Science Laboratory, more commonly known as the Curiosity Rover, which has an incredibly constrained communication link to remote Earth-based servers that precludes any remote processing of sensor data [13].

Due to increased uptake of autonomous systems, client only models are now being researched more thoroughly, with an emphasis on development of tools for bench-marking and comparing new solutions and hardware [14].

B. Communications

1) Data

Data communication with a remotely operated vehicle is typically handled by a digital radio operating in either simplex, half duplex or full duplex mode.

Transmission of data to and from the NIC First Responder is provided by a low-bandwidth serial data radio operating in half-duplex mode. Low carrier frequency and low transmission rate have been chosen to maximize the transmission distance between the OCU and the UGV.

2) Video

Video streams are transmitted from the NIC First Responder using a composite video simplex radio. This makes use of COFDM encoding that reduces signal loss and artefacts generated by enclosed interior environments, or non-line of sight (non-LOS) outdoor environments, with a range of up to 2km LOS and 500m non-LOS.

This trade-off has led to video transmission with increased range, at the expense of only having acceptable fidelity for the human eye.

III. Experimental Methodology

A. Hypothesis

Considering the technical difficulties presented by the transmission of composite video data from a mobile UGV to a base station for post-processing, a great deal of vital image data may be lost in transit. It is hypothesised that local computation of SLAM and subsequent transmission of the solution will provide more accurate, higher-fidelity map output than SLAM solutions derived using transmitted composite video data.

To test this hypothesis, an experiment was devised to analyse the possible fidelity loss to RGB-D sensor data when transmitted using the existing radio system. The results are presented and compared to existing bench-marking results of RGBD-SLAM. The SLAM solution is computed offline, using bench-marking data provided by the computer vision group, and their RGBD-SLAM bench-marking tool [15].

B. Compression of Sensor Data

The depth camera used, a Microsoft Kinect, provides two outputs:

- An RGB video stream providing a colour image ahead of the camera, with approximately 45 degree field of view (FOV). The stream is formatted as a 640x480 16-bit RGB image. The RGB camera is calibrated to associate RGB pixels with depth pixels directly. Warping of the RGB image to match the depth image is handled in silicon on the connect sensor.
- A depth video stream describing the distance as pixels in the FOV of the sensor, represented by a 640x480 16-bit grey-scale image, with 1mm depth resolution.

The data-sets used in the experiment were created from the original data, downloaded as a ROS bag file container containing colour, depth, IMU and ground-truth data.

The ROS functions used to provide data from the sensor additionally provide camera calibration data for both cameras. Image frames from the sensor are not time matched, nor are they guaranteed to be sequential, and a nearest match was used to create image streams with synchronous frames.

In order to transmit these frames over the single-channel radio link, three multiplexing approaches were attempted:

- Interleaving alternate RGB and Depth frames. Matching frames were chosen based on a closest-time pattern. This process results in a 50% reduction in frame rate on both video streams.
- Top/Bottom split-screen, with RGB and Depth frames stretched vertically to half their originally height, resulting in 640x240 frame size for both video streams. Matching frames were chosen based on a closest-time pattern.
- Left/Right split-screen, with RGB and Depth frames stretched horizontally to half their originally height, resulting in 320x480 frame size for both video streams. Matching frames were chosen based on a closest-time pattern.

To create the test sensor data for each of the above sets, the following procedure was taken:

1. Select matching frames from the benchmark data set based on their closest-time neighbour
2. Multiplex using interleaving, top and bottom and side by side
3. Save the video into a video file
4. Play back the multiplexed video from an embedded computer with composite output
5. Transmit the video over a COFDM radio link
6. Capture the video onto a second computer via the USB composite video capture device
7. Record the captured video into a new video file
8. De-multiplex using the reverse of the multiplexed method
9. Place the image data back into a copy of the original datasets.

Table 1 provides the observations made as a result of the initial testing of each approach.

TABLE I. COMPARISON OF VIDEO MULTIPLEXING APPROACHES

Multiplexing Process	Resulting Effective Resolution	Resulting Effective Frame Rate
Interleaved	100%	50%
Top/Bottom	50%	100%
Left/Right	50%	100%

IV. RESULTS

There is a perceptible drop in image-fidelity after transmission and reconstruction. This loss of fidelity presented itself through a greater contrast between the light and dark tones, and blurring of details. Contrast shift resulted in depth images that gave a poor representation of the original depth data. The meta-data associating greyscale tones with depth information became warped during transmission, leading to unreliable topographical output.

Tearing was caused by re-timing the digital image stream from 25fps to 30fps. Once converted back to 25fps, frames were found to contain sections with inter- leaved lines from the next or preceding frame, as shown in Figure 1. This tearing rendered the interleaved frame multiplexed streams un-recoverable, and this multiplexing approach was discontinued.

Figure 1. Example of image distortion of multiplexed frames after transmission

As a result of the frame rate change, the interleaved transmission produced an unusable video captured stream. Whilst the obvious solution is to use matching frame rates at all points in the system, the experiment was attempting to assess the suitability of such an approach using the existing technology of the NIC UGV.

A. *Image Analysis*

Frames that had been passed through the transmission stream were compared using ImageMagick [16] which provides Peak Signal to Noise Ratio (PSNR), Root Mean Squared Error (RMSE) and Max Absolute Error (MAE) metrics.

Table 2 outlines the results of this comparison. Top/Bottom multiplexing showed the highest error in the RGB channel, with similar errors in the depth image in both multiplexing processes.

TABLE II. IMAGE COMPARISON RESULTS

MUX Channel	PSNR	RMSE	MAE
Side/Side RGB	8.87025	23602.6 (0.360153)	17660.1 (0.269475)
Side/Side Depth	16.1778	10176.2 (0.155279)	8276.41 (0.12629)
Top/Bottom RGB	12.6852	15213 (0.232135)	11774 (0.17966)
Top/Bottom Depth	16.2149	10132.8 (0.154616)	8761.3 (0.133689)

B. *SLAM Benchmark*

While the RGBD-SLAM benchmark tool was able to process the data, in the multiplexed data-sets, the error was considerably higher. Pose estimation was poor, as shown in Figure 2.

Top/Bottom MUX was found to provide a close approximation of the original image depth data. An average error rate of 12% was observed, when comparing the original data to the Top/Bottom (T/B) MUX counting all additional and subtracted depth data.

Side/Side (S/S) MUX degrades the image-data significantly. As shown in the third graph of Figure 2, there is little to no comparison between the results of transmission and reconstruction, and the original data. This would make for a very unreliable SLAM output, to the point where the data is functionally useless for mapping purposes.

With the reduction in effective resolution taken into account, both T/B and S/S MUX result in a significant loss of data. T/B, including resolution loss, loses an average of 24% of the accuracy of the original, as the resolution is reduced by 50%, and an average of 12% of the reconstructed data is corrupted by

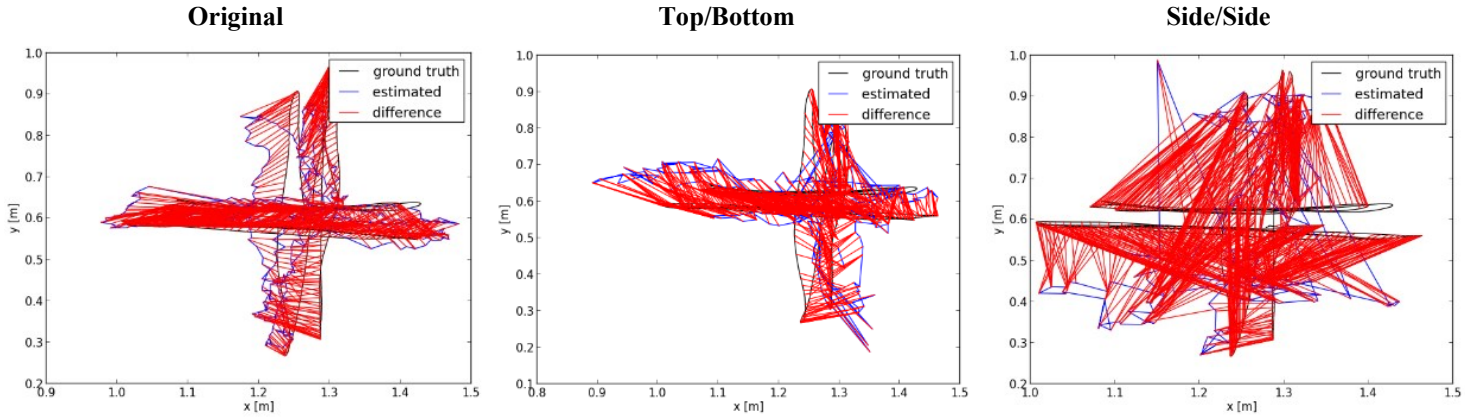

Figure 2. Example of the comparison of pose estimation between original and multiplexed data using the freiburg1_xyz dataset

the transfer process. S/S MUX completely corrupts the image, providing no usable depth data.

Figure 3 shows the results of the two multiplexing approaches and against the original dataset, comparing the outcomes of RGBD-SLAM. In both T/B and S/S, there is a large deviation from the original dataset for complex environments, with freiburg1_room showing between 0.92m and 1.02m of deviation from the mean.

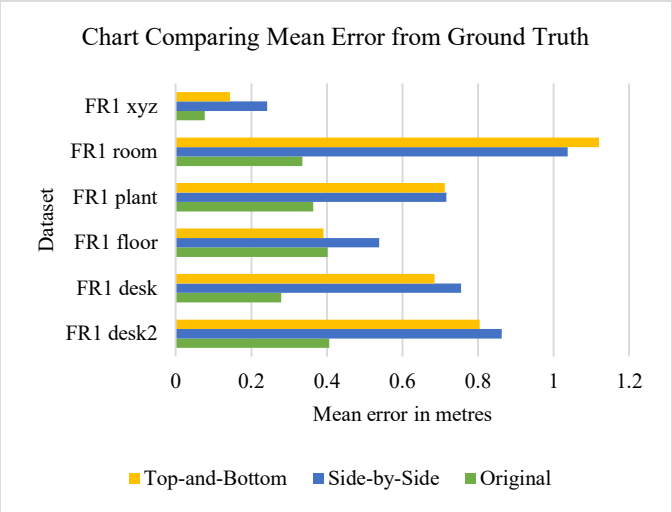

Figure 3. Comparison of absolute trajectory error across several datasets

T/B, across all samples, has a range of between 0.16m and 1.02m mean error. S/S has a mean error range of between 0.21m and 0.92m. Locally computed original data has an error rate of between 0.03m and 0.39m, significantly lower than tested alternatives.

It can be surmised that the effects of mobility in mapping an area are more pronounced for S/S, while T/B has more trouble with large areas. Both approaches are inferior to the locally computed original solution, when attempting to construct reliable, high-fidelity RGBD-SLAM output.

V. Conclusion

Transmission of RGB and Depth data from a remote mobile robot to a local base station, for the purpose of SLAM computation, requires a high-fidelity, high bandwidth connection between the robot-base station pair. In many common SAR and EOD operations, computation of a 3D map of the operational area would be beneficial to the operation.

The platform targeted in this paper has data and video connections optimised for transmission range and image fidelity suitable for a human operator. Using this same transmission system to transmit RGB-D camera data back to the base station for SLAM computation has shown to result in poor performance, due to loss of fidelity in the image stream. In both cases, the loss of fidelity was too great for the data to be useful in generating a SLAM solution with over 1m of error in room size spaces.

Analysis of two approaches to RGBD-SLAM data transmission found that T/B produced the highest errors per frame for the RGB channel. T/B and S/S were found to have similar depth-data errors. Both T/B and S/S were found to suffer serious degradation of output quality when profiled using the RGBD-SLAM benchmark tool. This resulted loss of depth-related data made the computation of a reliable SLAM solution too challenging.

Future work will focus on locally processed SLAM, using CUDA enabled embedded-processors. Combining this with an efficient transmission method for processed 3D map data, combined with task allocation [17], will enable autonomous drone-side self-navigation.

VI. References

[1] R. R. Murphy, S. Tadokoro, D. Nardi, A. Jacoff, P. Fiorini, H. Choset, and A. M. Erkmen, "Search and rescue robotics," in *Springer Handbook of Robotics*. Springer, 2008, pp. 1151–1173.

[2] N. Bergstrom, C. Raabe, K. Saito, E. Saad, and J. Vian, "Sensitivity study for feature-based monocular 3D slam," in *Aerospace Conference, 2015 IEEE*. IEEE, 2015, pp. 1–15.

[3] R. Smith, M. Self, and P. Cheeseman, "Estimating uncertain spatial relationships in robotics," *arXiv preprint arXiv:1304.3111*, 2013.

[4] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g 2 o: A general framework for graph optimization," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 3607–3613.

[5] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon, "Kinectfusion: Real-time dense surface mapping and tracking," in *Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium on*. IEEE, 2011, pp. 127–136.

[6] C.-K. Yang, C.-C. Hsu, and Y.-T. Wang, "Computationally efficient algorithm for simultaneous localization and mapping (slam)," in *Networking, Sensing and Control (ICNSC), 2013 10th IEEE International Conference on*. IEEE, 2013, pp. 328–332.

[7] H. Kretzschmar, C. Stachniss, and G. Grisetti, "Efficient information-theoretic graph pruning for graph-based slam with laser range finders," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*. IEEE, 2011, pp. 865–871.

[8] M. F. Fallon, J. Folkesson, H. McClelland, and J. J. Leonard, "Relocating underwater features autonomously using sonar-based slam," *Oceanic Engineering, IEEE Journal of*, vol. 38, no. 3, pp. 500–513, 2013.

[9] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison *et al.*, "Kinectfusion: real-time 3D reconstruction and interaction using a moving depth camera," in *Proceedings of the 24th annual ACM symposium on User interface software and technology*. ACM, 2011, pp. 559–568.

[10] F. Endres, J. Hess, N. Engelhard, J. Sturm, D. Cremers, and W. Burgard, "An evaluation of the rgb-d slam system," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE, 2012, pp. 1691–1696.

[11] J. Sturm and N. Engelhard and F. Endres and W. Burgard and D. Cremers, "A Benchmark for the Evaluation of RGB-D SLAM Systems", Proc. of the International Conference on Intelligent Robot Systems (IROS), 2012

[12] N. Engelhard, F. Endres, J. Hess, J. Sturm, and W. Burgard, "Real-time 3D visual slam with a hand-held rgb-d camera," in *Proc. of the RGB-D Workshop on 3D Perception in Robotics at the European Robotics Forum, Vasteras, Sweden*, vol. 180, 2011.

[13] P. J. Sellers, J. B. Garvin, A. L. Kinney, M. J. Amato, N. E. White *et al.*, "A vision for the exploration of mars: Robotic precursors followed by humans to mars orbit in 2033," *Concepts and Approaches for Mars Exploration, Houston, Texas*, 2012.

[14] L. Nardi, B. Bodin, M. Z. Zia, J. Mawer, A. Nisbet, P. H. Kelly, A. J. Davison, M. Luján, M. F. O'Boyle, G. Riley *et al.*, "Introducing slambench, a performance and accuracy benchmarking methodology for slam," *arXiv preprint arXiv: 1410.2167*, 2014.

[15] S. Scherer, A. Zell *et al.*, "Efficient onboard rgbd-slam for autonomous mavs," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*. IEEE, 2013, pp. 1062–1068.

[16] ImageMagick Studio LLC. ImageMagick [Online], http://www.imagemagick.org/, Accessed: 12/04/2015

[17] D. Smith, J. Wetherall, S. Woodhead, A. Adekunle, "A cluster-based approach to consensus based distributed task allocation", in *PDP 2014, 22nd Euromicro International Conference on*, IEEE, pp. 428-431