Advanced Technologies for Hypersonic, Propulsive, Energetic and Reusable Platforms
REU-HYPER 2019
28 May – 2 August 2019, Orlando, United States

REU-HYPER 2019

ASSESSMENT OF 3D OPTICAL RECONSTRUCTION TECHNIQUES FOR COMBUSTION APPLICATIONS: POINT CLOUD RECONSTRUCTION

Malcolm Gardner[1]

[1]University of Central Florida

ABSTRACT

As it stands today there are a handful of methods capable of reconstructing turbulent hypersonic flames in three dimensions. Struggling to be both reliable and cost-effective, the area for discussing and analyzing hypersonic flames has been dampened over the years. With this thought, examinations into various reconstruction techniques have resulted in an algorithm that can create dense 3D point cloud reconstructions from 2D images. Utilizing the feature-tracking algorithm Kanade-Lucas-Tomasi (KLT), multiple views of a single object were calculated and positioned together on a single plot for a near solid surface reconstruction. These results show the robust capabilities of open source code development that can meet the specifications of a project, the needs of an average laboratories' budget and compete with the industry standards.

1. INTRODUCTION

As new advancements further the studies of hypersonic flight, having the ability to reliably and efficiently measure turbulent flames is becoming a more relevant issue. In order for planes to fly at speeds exceeding the sound barrier we need to understand the the behavior of turbulent flames at higher atmospheric pressure. Early research for this involves methods that are taking advantage of point cloud and tomographic reconstruction to measure the velocity of these flames. However, they are often barred by the shear expense and complexity of an optical rig or that of a designed software. Additionally, the techniques of previous works were primarily focused in two dimensions which is inefficient when dealing with asymmetrical flames. Having since shifted into time-resolved three dimensional methods, these relatively new techniques lack the versatility to be used by a large number of people due to the cost of equipment and cross-disciplinary knowledge requirements. Thus, the development of flame measurement techniques has been hindered as a whole. This study seeks to expand upon the viability of developing effective, "in-house" algorithms for 3D reconstructions, despite knowledge gaps in coding or image processing.

2. MATERIALS AND METHODS

This section details the methodology and reasoning of the 3D reconstruction algorithm techniques deployed.

2.1 Calibration

Before work on the algorithm could begin it was necessary to create a consistent manner for capturing and processing images. Limiting the number of variables in any experiment is the key to obtaining quality results and would prove to be a necessity here. Having a calibrated camera was paramount to the success of the algorithm's accuracy. Utilizing MATLAB's built in camera calibration app, pictures of a checkered board at different angles distances was used in estimating the camera parameters of an IPhone 8's rear camera. Camera parameters, including focal length, radial distortion and reprojection error, were all calculated with an accuracy of less than a pixel. By inputting the real world measurement of the checkered squares and tracking the sharp changes in intensity values, the camera calibration app estimated the features of the camera with a high degree of accuracy. Having these camera parameters are essential to estimating the external camera positions, calculating the position of features in other images without having to manually specify and in removing the slight edge distortions a camera produces.

2.2 Image Processing and Acquisition

An equally important preparation was capturing a quality image. When possible images were taken in places with flat, monochromatic backgrounds in an attempt to reduce the amount of extraneous features being tracked by the algorithm. However, this was unavoidable in some cases,

requiring that the background be reduced or removed through image processing. This was achieved by subtracting the intensity values of the background from the image. In addition to removing background noise, images were also cropped in the algorithm with an adjustable region of interest. Furthermore, the images were converted to grayscale to better suit the algorithm's form of point tracking. Relying on the sharp changes in intensity to track points, it is more suitable to calculate this in a single range of BW values as opposed to three in RGB values.

2.3 Feature Extraction

For each input image there were a cluster of unique features that could be detected and stored. These features are what the algorithm uses to "see" the object or scene, so by taking pictures at different angles it can generate a sense of the actual world geometry in later step. Moreover, the method of detection involved the Kanade-Lucas-Tomasi (KLT) feature-tracking algorithm which detects these features based off of sharp changes in intensity. This was a fitting choice for a flame since it would produce stark contrast against a dark background.

2.4 Feature Matching

In order to begin building the extracted features must be matched and related between the different input images. If different images contain a point that has the same description, then that point can be marked and evaluated as a matched feature. With that, images were taken within close proximity of each other as to increase overlap significantly and make the matching and initialization of features more optimal.

2.5 Geometric Verification

Once the features are matched and potential points are stored, they must then be verified. Matches come with errors, so to make sure the matches are correct the camera parameters along with the estimated essential matrix are utilized. By calculating the epipolar lines, the distance between two matched features, we can choose the degree of acceptable error for each point and eliminate a large number of outliers that could affect calculations going forward. The particular algorithm used is called RANSAC which can be tailored to improve computational time.

2.6 Reconstruction Initialization

With verified matched features a point cloud of two input images with the most geometrically verified points can be initialized. By fixing the camera positions and points the image registration, triangulation and bundle adjustments can work around this initial framework this process create. This will allow for the points from additional input images to build on top on it and in improve accuracy.

2.7 Image Registration

The algorithm uses an iterative process to create points since it can only compare two input images at a time. Registering new images after each step allows the algorithm to slowly build the point cloud using both 2D features and the previously initialized 3D points while reducing the overall stored memory. Once the matched points are verified they will be added to the point cloud and triangulated to their relative locations.

2.8 Triangulation

The final parts of the algorithm help visualize the data as a whole. By matching 3D points across multiple views the coordinates of the points are refined with the addition of new images to the make the point cloud denser and more precise. The camera parameters are used to estimate the relative location and size of each point and because the features in every image are unique, color are can be applied as well. This was achieved by storing the RGB value associated with a matched feature from the original image. Hence, plotting these together yields a dense reconstruction of a 3D scene.

2.9 Bundle Adjustment

The bundle adjustment phase is used to put the finishing touch on the point cloud. By accounting for inaccuracies produced when the external camera positions are estimated and points are triangulated, it can refine the 3D points. Like the triangulation phase, this process is iterative and improves accuracy by reducing the amount of errors that can compile with the addition of each new input image.

3. RESULTS AND DISCUSSION

The following figures show the progression of the algorithm's development thus far.

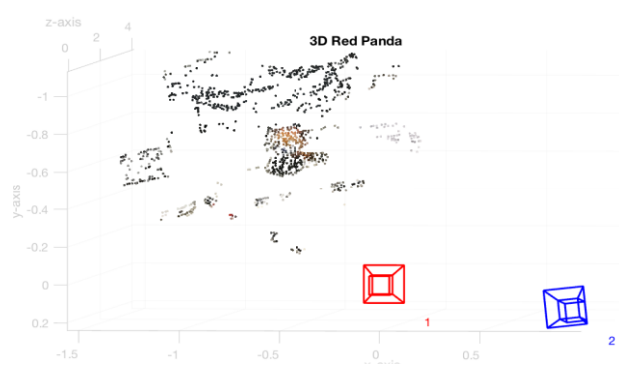

Fig. 1. The first point cloud created with 2 images, without a calibrating camera. The same input images were used in Fig. (2).

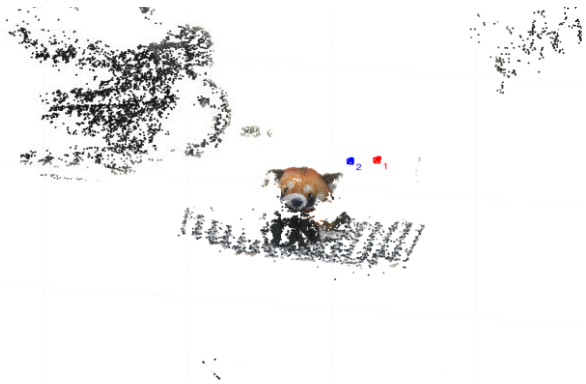

Fig. 2. Point cloud created with 2 images, with a calibrated camera. The same input images were used in Fig. (1).

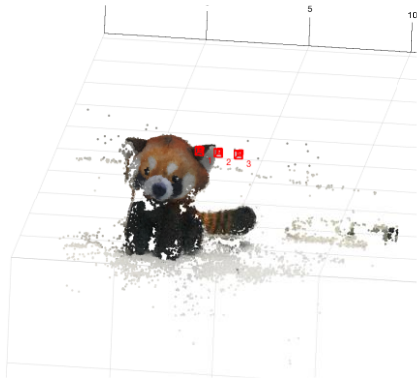

Fig. 3. The first point cloud created from multiple input images.

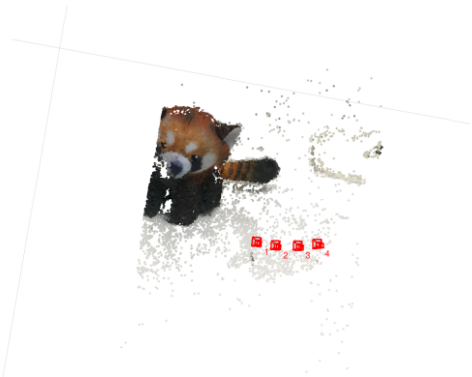

Fig. 4. **A** point cloud created from the first set of 8 total images. Reference point cloud for Fig. (6).

Fig. 5. **A** point cloud created from the second set of 8 total images. Reference point cloud for Fig. (6).

Fig. 6. Scene consisting of points from Fig (4) and Fig (5) on a single plot

The first 6 figures show the results of the algorithm in the early stages of development. Fig. 1. and Fig, 2. highlight the importance of having a calibrated camera, showing significant increase in points cloud density. While Fig. 3.-6. compare the algorithm's accuracy using multiple input images.

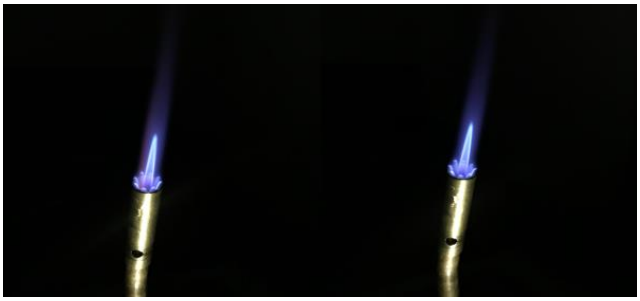

Fig. 7. Input images of images from Fig. 8.

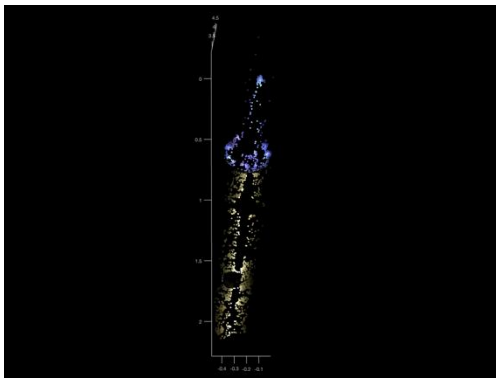

Fig. 8. Sparse reconstruction of images from Fig. 7.

Fig. 9. Input images for Fig. 10.

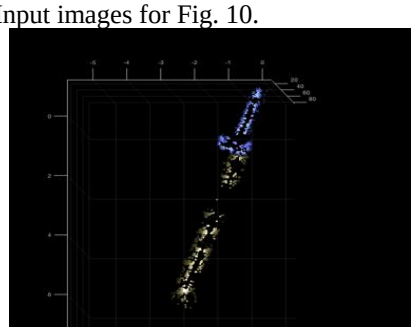

Fig. 10. Sparse reconstruction of images from Fig. 9.

Fig. 11. Input images for Fig. 12.

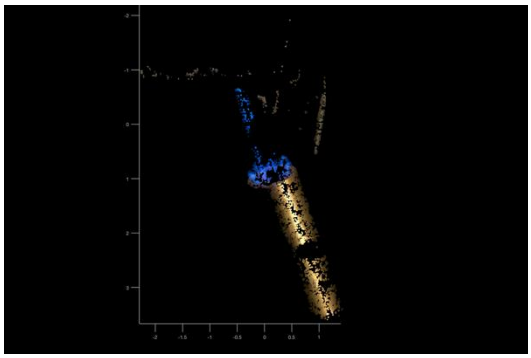

Fig. 12. Sparse reconstruction of images from Fig .11.

Fig. 7-12 show the initial algorithm's capabilities to reconstruct a flame after developing it primarily to reconstruct a solid object.

Fig. 13. Input images for Fig. 14.

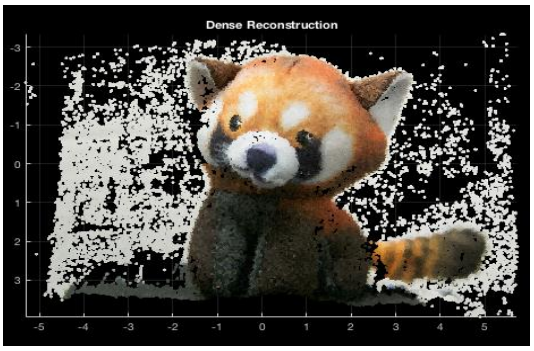

Fig. 14. Dense reconstruction of images from Fig. 13.

Fig. 15. Input images for Fig. 16.

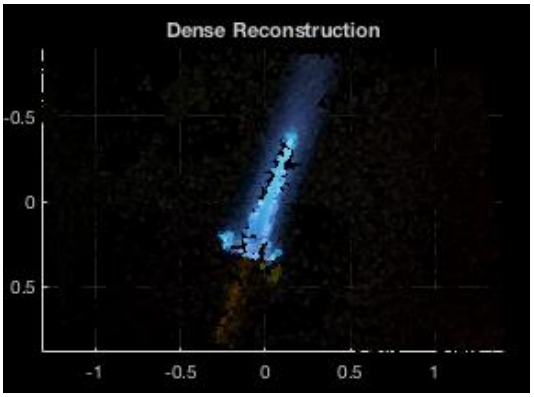

Fig. 16. Dense reconstruction of images from Fig. 15.

Fig. 13-16 display the algorithm's aptitudes thus far. Utilizing a quality camera along with additional code development these are the most visibly dense reconstructions.

4. CONCLUSION

In short, a 3D point cloud algorithm was created with little to no coding experience or extensive resources. This algorithm was capable of creating sparse and dense point clouds. Additionally, it was able to create these reconstructions using 2 or more input images. More importantly it indicated the potential point cloud has in 3D flame reconstruction and the creation of tailored algorithms that do not require expensive equipment or far-reaching knowledge. However, the knowledge that was required was a major obstacle to overcome due the shear amount of theories and methods combined with time constraints. So for an algorithm to be improved upon it is recommended for one to gain general knowledge and understanding in computer vision to further the study.

ACKNOWLEDGEMENTS

Dr. Jonathan Reyes

REFERENCES

[1] Bianco, Simone, Gianluigi Ciocca, and Davide Marelli. "Evaluating the performance of structure from motion pipelines." Journal of Imaging 4.8 (2018): 98.

[2] Shi, J., and C. Tomasi, "Good Features to Track," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 1994, pp. 593–600.

[3] Scarano, F. *Tomographic PIV: principles and practice*, 2012.

[4] Reyes, Jonathan. *Single sensor fiber-based high-speed tomographic particle image velocimetry*, 2019.

[5] Z. Zhang. A flexible new technique for camera calibration. IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(11):1330-1334, 2000.

[6] R. Hartley, A. Zisserman, "Multiple View Geometry in Computer Vision," Cambridge University Press, 2003.

[7] Brostow, Gabriel J., et al. "Segmentation and recognition using structure from motion point clouds." *European conference on computer vision*. Springer, Berlin, Heidelberg, 2008.

[8] Qi Hu et al 2018 J. Phys.: Conf. Ser. 1087 062031.

[9] Shervais, Katherine. "Structure from Motion Introductory Guide". See also URL https://www.unavco.org/education/resources/educational-resources/lesson/field-geodesy/module-materials/sfm-intro-guide.pdf>. *UNAVCO* 22 October 2015