Tactical 3D Model Generation using Structure-From-Motion on Video from Unmanned Systems

Josh Harguess, Mark Bilinski, Kim B. Nguyen, and Darren Powell

Space and Naval Warfare Systems Center Pacific, 53560 Hull St., San Diego, CA, USA

ABSTRACT

Unmanned systems have been cited as one of the future enablers of all the services to assist the warfighter in dominating the battlespace. The potential benefits of unmanned systems are being closely investigated – from providing increased and potentially stealthy surveillance, removing the warfighter from harms way, to reducing the manpower required to complete a specific job. In many instances, data obtained from an unmanned system is used sparingly, being applied only to the mission at hand. Other potential benefits to be gained from the data are overlooked and, after completion of the mission, the data is often discarded or lost. However, this data can be further exploited to offer tremendous tactical, operational, and strategic value. To show the potential value of this otherwise lost data, we designed a system that persistently stores the data in its original format from the unmanned vehicle and then generates a new, innovative data medium for further analysis. The system streams imagery and video from an unmanned system (original data format) and then constructs a 3D model (new data medium) using structure-from-motion. The 3D generated model provides warfighters additional situational awareness, tactical and strategic advantages that the original video stream lacks. We present our results using simulated unmanned vehicle data with Google Earth™ providing the imagery as well as real-world data, including data captured from an unmanned aerial vehicle flight.

Keywords: structure from motion, unmanned systems, 3D model generation

1. INTRODUCTION

Vast amounts of data is collected by electro-optical (EO) sensors on manned and unmanned assets deployed around the world with a concentration of the collections in areas of national interest. The sensor data is natively generated and stored in a temporal format which makes post mission analysis grow exponentially as additional sensor data from additional platforms is collected, assuming the analysis is performed only by humans. Additionally, humans are not efficient at correlating numerous sensor streams from multiple viewing angles collected over time.

Structure-from-motion (SfM)[1–4] is a computer vision method that builds a spatial model from the set of EO temporal data collected by the sensors on an object, or area of interest. This concept has the benefit of building a three-dimensional (3D) model from the two-dimensional (2D) imagery, condensing hours of collected data into a concise representation. Additional collections increase the model fidelity without increasing an analyst's (or any other user's) view time and it may be related to other geo-spatial tracks, as well as other data.

SfM requires a large amount of computation time which may be reduced through distributed computation. As the density of the computation and storage increases, the application of SfM will push closer to the source of the data. This enables forward deployed applications, such as a Special Forces team flying a small unmanned aerial vehicle (UAV) around a ship prior to boarding. Additionally, as centralized data repositories are made available through an open architecture and network bandwidth optimizations, SfM will enable a more effective way to view massive amounts of information. The application of SfM is of particular interest to open architectures such as SSC Pacific's UxV to the Cloud via Widgets, as described by Nguyen et al..[5]

This work was funded by the Naval Innovative Science and Engineering Program at Space and Naval Warfare Systems Center Pacific.

Please send correspondence to Josh Harguess:
E-mail: joshua.harguess@navy.mil, Telephone: 1 619 553 0777

This paper discusses the application of SfM to three scenarios and data sets designed to mimic a forward deployed operation. The first dataset is a simulated UAV video capture of Petco Park in San Diego, CA, taken from Google Earth™. The second dataset consists of frames taken from a recent media video release of a Littoral Combat Systems ship in an open ocean environment. The third dataset was taken in-house from a GoPro® camera mounted on a quadcopter collecting video of a grouping of several objects (palette, tire, hoses, etc.). The purpose of this scenario is to simulate a remotely operated, semi-autonomous unmanned discovery of a potential improvised explosive device (IED).

and consists of video taken of a grouping of several objects (palette, tire, hoses, etc.) to simulate a scenario of a remotely operated, semi-autonomous unmanned discovery of a potential improvised explosive device (IED).

2. BACKGROUND AND METHODOLOGY

In this section, a brief background into structure-from-motion (SfM) will be followed by a high-level description of our 3D model generation pipeline using SfM and post-processing methods.

2.1 Structure-from-motion (SfM)

The main component of 3D model generation for our system is SfM. We utilize the open-source and freely available SfM application known as VisualSFM.[6,7] VisualSFM is an end-user, "off-the-shelf" implementation of SfM that is easy to configure and used for most 3D model generation applications from imagery. While the usual interface with VisualSFM is through their graphical user interface (GUI), we will be utilizing their command line implementation which allows us to make the SfM application accessible through a user interface integrated with the rest of our system.[5]

There are two types of 3D model generation available within VisualSFM; sparse and dense reconstruction. Sparse reconstruction begins by computing SIFT[7] features of each of the images, using a GPU-enabled implementation[8] to increase speed performance. Next, VisualSFM performs SIFT image feature matching between the images by comparing the SIFT descriptors between the images and accepting a match if the distance between two descriptors is less than a specified threshold. Bundle adjustment is then used to refine the 3D coordinates of the point clouds as well as the parameters and positions of the "cameras" in the scene used to solve for the sparse reconstruction.[9]

Dense reconstruction in VisualSFM consists of utilizing multi-view stereo algorithms to take the outputs of the sparse reconstruction and combine the point clouds into a single, more dense point cloud. In the results section of this paper, the "sparse" result presented is simply the first (but sometimes only) sparse model generated by VisualSFM. The method for dense reconstruction utilized by VisualSFM is that of Furukawa et al..[10,11] In addition to the multi-view stereo step, a "bundle adjustment" step is again performed as in the sparse reconstruction.

Figure 2b shows an example result of the sparse reconstruction step in VisualSFM while figure 2c shows the dense reconstruction from the same dataset. The result in the dense reconstruction case is less than desirable, which leads to the discussion in the next section of post-processing steps to improve and "clean" the 3D model.

2.2 Post-processing

In several of our experiments, VisualSFM would produce an acceptable 3D model during the sparse reconstruction phase with a very minimal number of 3D points. However, the resulting dense reconstruction model would contain many spurious and erroneous 3D points due to confusion in the SIFT matching and bundle adjustment steps. Therefore, post-processing of the dense reconstruction may be required to yield a higher fidelity 3D model. We utilize the Point Cloud Library (PCL)[12] and Meshlab[13] to implement the post-processing steps outlined below.

The sparse reconstruction more accurately represents the underlying geometric structure of the true 3D model, since it has a more strict threshold for the SIFT matching and bundle adjustment steps. Therefore, the idea is to use the sparse model as a reference for cleaning the noisier dense reconstruction. First, we remove statistical outliers from the sparse reconstruction in the following manner. For each point, the mean distance to it's k nearest neighbors is calculated, assumed to follow a Gaussian distribution, and then removed from the dataset if outside the selected standard deviation,[14] producing a filtered version of the sparse model. Next, the

goal is to remove points from the dense reconstruction that are far from this filtered sparse model. For every point in the dense reconstruction, we find it's nearest point in the filtered sparse reconstruction and record that Euclidean distance. Then, any point whose distance exceeds the specified threshold is removed from the dense model, resulting in a clean dense reconstruction.

3. RESULTS

This section will present the findings of the VisualSFM application along with the post-processing results on the three datasets: Petco Park, LCS, and Palette.

3.1 Petco Park: Simulated Data from Google Earth™

The Petco Park dataset was taken from a flight simulator programmed to fly in a pattern over Petco Park in San Diego, CA, using Google Earth™ as the visual feedback. This *simulated* dataset depicts a set of imagery of what might be collected if a UAV platform was to actually fly over Petco Park. This scenario provides us an idea of capabilities of 3D model generation in a forward deployed reconnaissance mission using SfM.

Example images of the Petco Park dataset are shown in Figure 1a. The sparse and dense reconstructions are shown in Figures 1b and 1c, respectively. To get a feel for how this 3D data might be used, three other views of the dense reconstruction are shown in Figures 1d, 1e, and 1f. For instance, in Figure 1f, you can zoom in to see the relative heights of buildings with respect to Petco Park, so if you knew any of the buildings' heights, or the heights of the doorways or other features in the model, then you could compute the absolute of height of Pecto Park and surrounding buildings. Note that since the dense reconstruction model is sufficient, no other post-processing is performed on the point clouds for the Petco Park data.

3.2 LCS: Littoral Combat Ship (LCS) Video

The LCS dataset is taken from a publicly available video[15] (Copyright held by the United States Navy) and depicts an LCS ship performing several maneuvers on open ocean water. Frames are extracted from this video to form the imagery dataset and three examples of this dataset are shown in Figure 2a. These frames are then fed into VisualSFM and the post-processing pipeline. Examples of the sparse and dense reconstructions, along with the clean point cloud after post-processing, are shown in Figures 2b, 2c, and 2d, respectively.

Due to the confusion in SIFT feature matching across the images because of the ocean state, there is a lot of noise and error in the dense reconstruction. In other words, since the state of the ocean is changing rapidly between frames, the dense reconstruction algorithm incorrectly includes SIFT matches between water patches because of their visual similarity, even though they are clearly not related between frames to our human eyes. This is caused by the threshold parameter that can be difficult to set in imagery such as this. Therefore, the post-processing steps outlined in Section 2.2 are used to clean the point cloud generated by the dense reconstruction. The results can be seen in other views of the cleaned reconstruction in Figures 3a, 3b, and 3c.

3.3 Palette: GoPro® Video from a Quadcopter

To form the Palette dataset was taken in-house from a GoPro® mounted on a quadcopter. The quadcopter was programmed to fly around the pile of various objects, such as a palette, tire, and hoses, and record video during the flight. Then, the imagery was extracted for our 3D model generation pipeline. Examples of the imagery are show in Figure 4a. The resulting sparse and dense reconstructions, along with the clean point cloud after post-processing, are shown in Figures 4b, 4c, and 4d, respectively.

In these results, the dense reconstruction does not exhibit the same noise artifacts as seen in the LCS dense reconstruction, however the Palette dense reconstruction does include 3D points in the final model that are not helpful for visualizing the objects of interest. Therefore, the post-processing steps were again applied and additional views of these clean reconstructions can be seen in Figures 5a, 5b, and 5c. As you can see from these figures, a single 3D model can encapsulate and summarize a stream of video into a single product which may be very useful for object identification, which could mean the difference between correctly and incorrectly identifying an IED in a forward deployed environment.

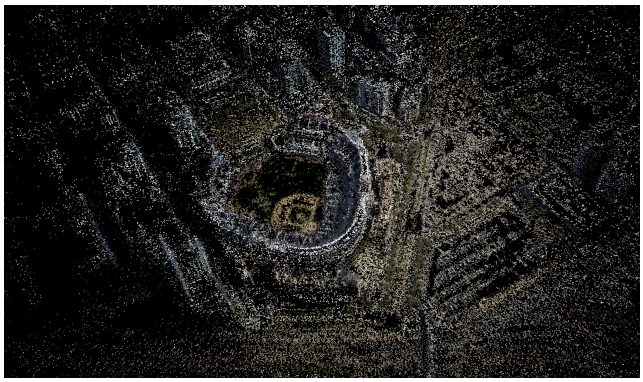
4. CONCLUSION AND FUTURE WORK

While the rate of EO sensor integration into manned and unmanned platforms continues to grow, the number of analysts and operators needed to monitor and analyze the imagery created from the platforms remains static, at best. Additionally, the sensors themselves are growing in complexity by increasing the resolution, frame rates, and functionality, resulting in an even larger and more complex dataset than the operators and analysts currently encounter. Therefore, the need for automatically summarizing large amounts of imagery and other data into meaningful products that users are quickly and easily able to consume is growing dramatically. In this paper, we have presented one such method known as structure-from-motion (SfM). SfM takes in the collected imagery and produces a 3D model of an object or area of interest for tactical planning, object identification, and many other applications. We have presented the results of on three datasets; one simulated and two of real-world imagery. The results demonstrate the effectiveness of the SfM approach at summarizing a large, difficult to digest, amount of information into a product that may expose information that the 2D imagery was not able to display. Future work in this area is to better understand the computational complexity more clearly at each of the steps and work towards increasing the speed to improve the integration into real-time and near real-time systems.

REFERENCES

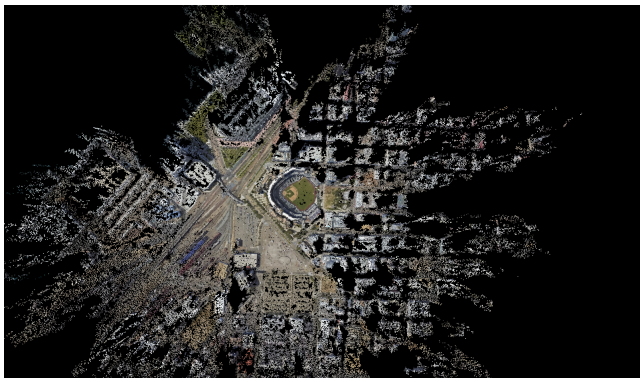
1. Huang, T. S. and Netravali, A. N., "Motion and structure from feature correspondences: A review," *Proceedings of the IEEE* **82**(2), 252–268, IEEE (1994).
2. Beardsley, P. A., Zisserman, A., and Murray, D. W., "Sequential updating of projective and affine structure from motion," *International journal of computer vision* **23**(3), 235–259, Springer (1997).
3. Tomasi, C. and Kanade, T., "Shape and motion from image streams under orthography: a factorization method," *International Journal of Computer Vision* **9**(2), 137–154, Springer (1992).
4. Agarwal, S., Snavely, N., Simon, I., Seitz, S., and Szeliski, R., "Building rome in a day," in [*Computer Vision, 2009 IEEE 12th International Conference on*], 72–79, IEEE (2009).
5. Nguyen, K. B., Powell, D. N., Yetman, C., August, M., Alderson, S. L., and Raney, C. J., "Cloud-based distributed control of unmanned systems," in [*SPIE Defense+ Security*], International Society for Optics and Photonics (2015).
6. Wu, C., "Towards linear-time incremental structure from motion," in [*3D Vision-3DV 2013, 2013 International Conference on*], 127–134, IEEE (2013).
7. Wu, C., "Visualsfm: A visual structure from motion system," (2011). `http://ccwu.me/vsfm/`.
8. Wu, C., "Siftgpu: A gpu implementation of scale invariant feature transform (sift)(2007)," (2007). `http://cs.unc.edu/~ccwu/siftgpu`.
9. Wu, C., Agarwal, S., Curless, B., and Seitz, S. M., "Multicore bundle adjustment," in [*Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*], 3057–3064, IEEE (2011).
10. Furukawa, Y. and Ponce, J., "Accurate, dense, and robust multiview stereopsis," *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **32**(8), 1362–1376 (2010).
11. Furukawa, Y., Curless, B., Seitz, S. M., and Szeliski, R., "Towards internet-scale multi-view stereo," in [*Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*], 1434–1441, IEEE (2010).
12. Rusu, R. B. and Cousins, S., "3d is here: Point cloud library (pcl)," in [*Robotics and Automation (ICRA), 2011 IEEE International Conference on*], 1–4, IEEE (2011).
13. Cignoni, P., Callieri, M., Corsini, M., Dellepiane, M., Ganovelli, F., and Ranzuglia, G., "Meshlab: an open-source mesh processing tool," in [*Eurographics Italian Chapter Conference*], 129–136, The Eurographics Association (2008).
14. PCL, "Removing outliers using a statisticaloutlierremoval filter." `http://pointclouds.org/documentation/tutorials/statistical_outlier.php#stastical-outlier-removal`.
15. Youtube, "Us navy - uss independence (lcs 2) maneuvering capabilities demonstration," (July 2013). `https://www.youtube.com/watch?v=S2l1aRcPOyk`.

(a) Petco Park: example images

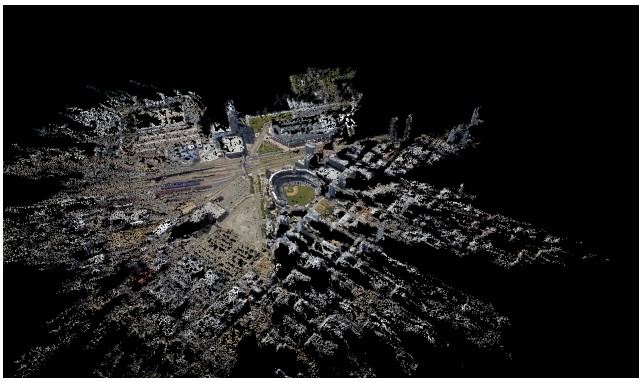


(b) Petco Park: sparse

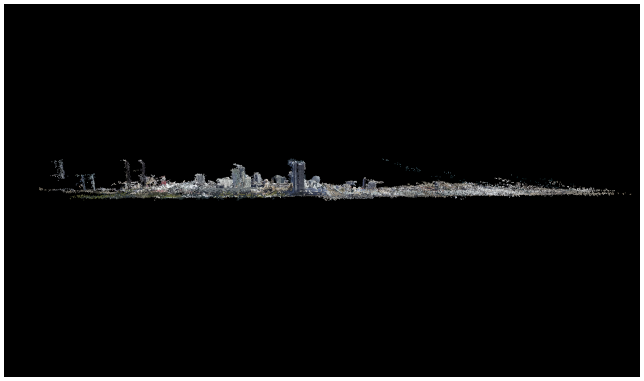
(c) Petco Park: dense



(d) Petco Park: dense top view



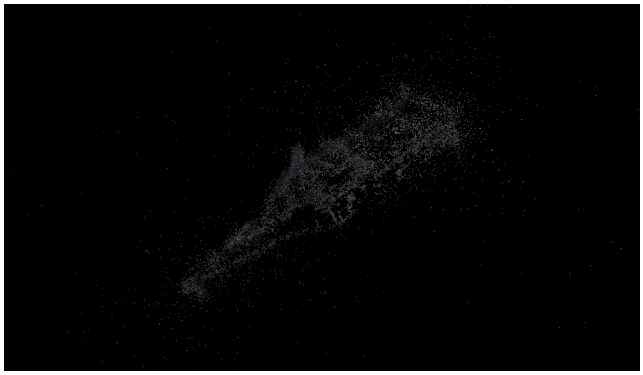
(e) Petco Park: dense 45° view



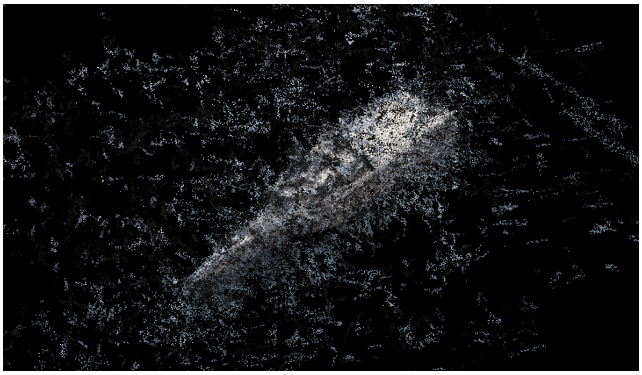
(f) Petco Park: dense side view

Figure 1: Petco Park: (a) Example images from the dataset and example views of the (b) sparse and (c) dense resulting point clouds, and (d-f) three example views of the dense point cloud

(a) LCS: example images



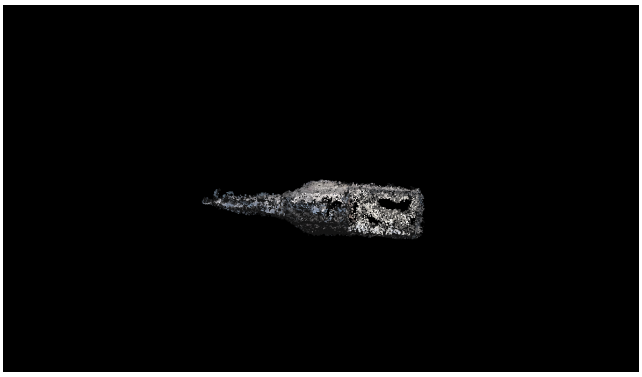
(b) LCS: sparse



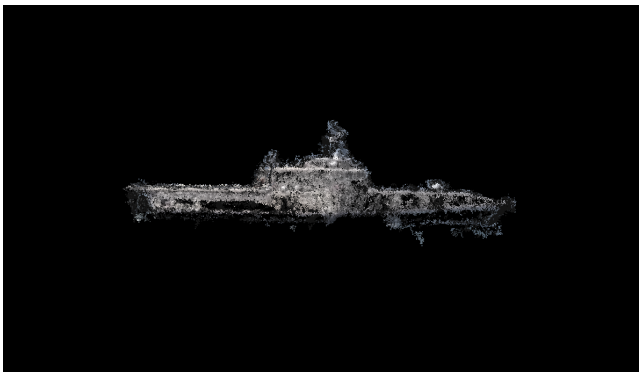
(c) LCS: dense

(d) LCS: clean

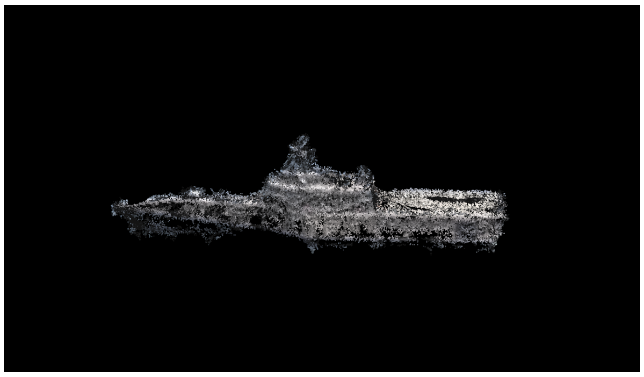
Figure 2: LCS: (a) Example images from the dataset and example views of the (b) sparse, (c) dense, and (d) clean resulting point clouds.



(a) LCS: top view



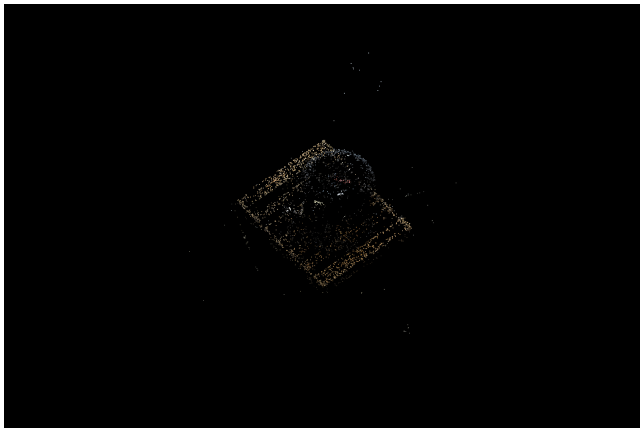
(b) LCS: 45° view



(c) LCS: side view

Figure 3: LCS: Three example views of the cleaned point cloud

(a) Palette: example images



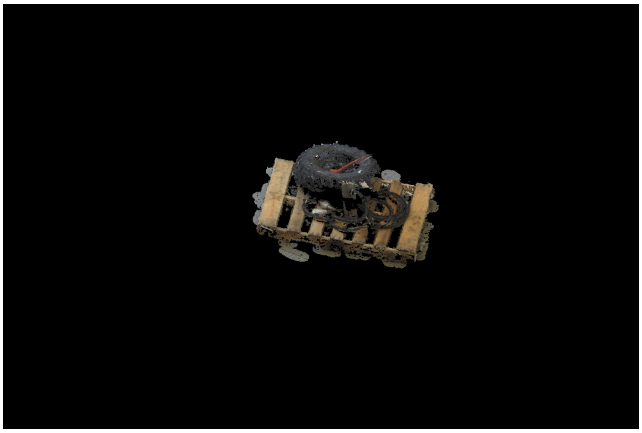
(b) Palette: sparse

(c) Palette: dense

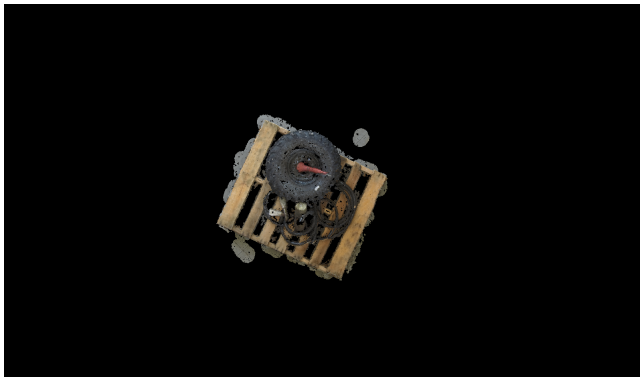
(d) Palette: clean

Figure 4: Palette: (a) Example images from the dataset and example views of the (b) sparse, (c) dense, and (d) clean resulting point clouds.

(a) Palette: side view



(b) Palette: 45° view



(c) Palette: top view

Figure 5: Palette: Three example views of the cleaned point cloud