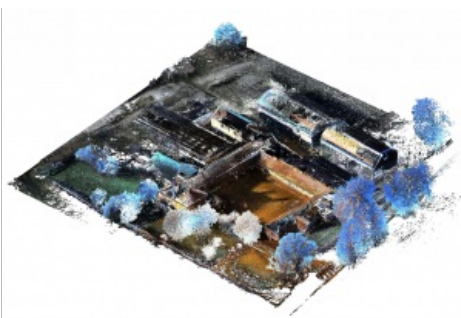
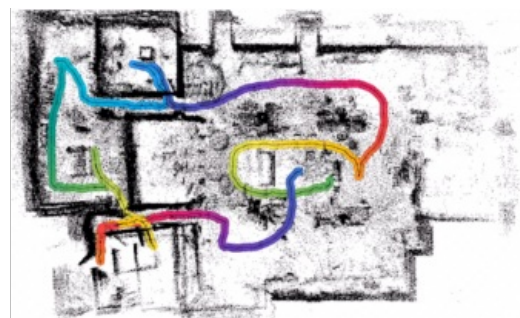


What's the Difference Between SLAM Algorithms for Handheld Lidar?



To get the full story when researching handheld Lidar scanners, you also need to compare the SLAM algorithms. These three criteria will help you: processing speed, sensor integration and colourization.

While researching handheld Lidar scanners, you will probably scour spec sheets, cross-reference features, and read about software workflows. But that



will get you only part of the way to understanding the market – you will also need to compare the simultaneous localization and mapping (SLAM) algorithms that power these devices and enable them to create a point cloud as you walk.

The problem is that SLAM algorithms can be difficult to understand if you're not a software engineer. Good news: there are a few criteria you can use to judge SLAM algorithms. These criteria will help you understand what sets one SLAM algorithm apart from one another, giving you the information you need to narrow the field.

Those are processing speed, sensor integration, and colorization. Let's dig in.

Processing Speed

A SLAM algorithm makes an astounding number of calculations each second. It tracks your movement through a space using data fed by a number of sensors, and then fuses that sensor data to align the millions of measurements captured by the Lidar. It's a tough challenge, and each SLAM algorithm will approach it in a different way, which means that each algorithm will process at a different speed.

Real-time processing: These SLAM algorithms process your scan as you capture so that you can download a usable point cloud as soon as you are done capturing the environment. The tradeoff is that this approach sacrifices some reliability, making the algorithm more likely to make tracking or processing errors in challenging environments. This method is best for when you absolutely need immediate turnaround on your data.

Post-capture processing: Many SLAM algorithms use this method, which requires that you begin processing your scan after completing the capture. It takes longer than real-time processing (obviously), but it also ensures greater reliability in your scan results. Depending on your timeline, the extra processing may not prove to be a big issue – you might have downtime built into your schedule, for instance while driving to a site, overnight between days of work, while eating meals. If you can use these otherwise inactive periods to process your data, you can ensure the highest quality data without any extra active time.

Both: Some SLAM implementations let you select whether to process your data in real time or after the capture, enabling you to adapt your approach to the needs of each individual situation.

Which is right for you? The best SLAM algorithm will offer a balance of processing speed and reliability that fits your company's workflows.



Colorized point clouds with high point density, low point noise, full environmental coverage, and even detail, generated by a PX-80 handheld scanner.

Sensors

SLAM implementations can differ according to the hardware they use, and that's true especially of the sensors that capture the data the algorithm needs to make its calculations.

An IMU and a Lidar sensor: This SLAM approach is very effective in open areas that contain distinct 3D geometry, like chairs, pipes, railings, trees, etc. However, SLAM like this can have trouble in areas dominated by planar shapes like walls. Hallways, warehouses, and other areas are more likely to cause errors and result in bad data.

One or more cameras: This approach to SLAM, which is more often used in robotics, is better at tracking the operator as they move through the feature-poor environments listed above. There are still limitations, though. Camera-based SLAM requires good lighting conditions, so it can have trouble in areas of high-contrast light, dim light, or outdoors. It also returns errors in environments with repeating patterns. Though it can track planar shapes, it will often have trouble with blank walls.

Lidar sensor and a camera: This kind of SLAM is better at tracking your movement through a variety of environments, indoors and out, regardless of the dominant geometric features or lighting conditions. Though it's not error-proof, it offers the best of both methods outlined above.

Which is right for you? The SLAM algorithms you should test are the ones that are most likely to work well in the environments you will be scanning regularly.

Colorization

Many SLAM algorithms offer a way to colorize your point cloud. The result is much easier to read than a raw point cloud – it looks like a colored 3D photograph, making it much easier to discern environmental elements like columns, rebar, doors, trees, cars, and so on.

Post-scan colorization: Many SLAM algorithms will colorize your point cloud data after you have completed the scan. Scanners using this method often require you to attach a secondary camera to the device, and perform extra processing steps in your workflow to achieve the desired results.

Automatic colorization: Some approaches to SLAM will colorize your scan automatically as you capture. However, not all handheld scanners approach this function in the same way.



A colorized point cloud.

Some handheld scanners include a camera and enable you to use it for either tracking your movement or colorization, but not both during the same scan. Others enable you to do both.

Some scanners use SLAM that offers automatic colorization, but include a camera with a smaller field of view than the Lidar sensor. That means even if you ensure that you capture the entire environment with the Lidar, there will be portions you miss with the camera – resulting in a mix of colorized data and uncolorized data. Other scanners also include cameras with a wider FOV to ensure that all Lidar points are colorized.

Which is right for you? Think about whether a raw, uncolorized point cloud is ideal for your workflows, or if you could benefit from colorization. Then, decide whether you would like that colorization to be automatic, and what implementation covers your needs.

Conclusion

Even without getting too technical on SLAM algorithms, you can see that not all SLAM algorithms are the same. Each SLAM algorithm results from a series of choices and compromises, and that means no SLAM algorithm is going to offer the best performance in every environment.

After reading the list above, you should be in a good position to determine which factors are most important to your workflow and make an informed choice about what scanners to bring in for a test.

The original version of this article was published by [Paracosm](#), the developer of the [PX-80 handheld scanner](#).