Point Cloud Gathering for an Autonomous Bucket Excavator in Dynamic Surroundings

Gregor Zolynski, Christopher Schank und Karsten Berns

Robotics Research Lab, University of Kaiserslautern, 67663 Kaiserslautern

Abstract This publication shows a concept and implementation of an uncomplicated system for collecting point clouds from multiple sensors with known relative pose which is capable of maintaining a (to some extent) consistent representation of dynamic surroundings under the impact of heavy sensor noise and sensor vibration. In addition to the ever-changing terrain the presence of mobile objects (e.g. humans, construction vehicles) is taken into account.
We present a simple technique for scan point clustering avoidance as well as a quick removal of invalid data points caused by mobile objects. The system utilizes already existing sensors intended for security tasks.

1 Introduction

This document presents a system which is utilizing sensors mounted to a bucket excavator and intended for safety applications for additional gathering of 3-dimensional point cloud data. The system is capable of using multiple sensors (planar laser scanners, but other distance sensors can be incorporated) simultaneously and yields a useful representation of a semi-dynamic environment.
The remainder of this section presents the related work, a short discussion about the advantages of point clouds to other forms of data representation, and a few applications which can use the data generated by this work. The main section 2 introduces the main reasoning and concepts which have been developed during the course of the work. Section 3 shows a few interesting details of the implementation. Afterwards, section 4 presents some results.

1.1 Motivation

Bucket excavators are a very common sight on construction sites. Those versatile machines are used for landscaping tasks, e.g. digging trenches or loading piles of material into dump trucks. The highly-articulated arm allows for very precise and quick handling of soil. There are very complex tasks which can be performed by a skilled driver, but the two examples given constitute the majority of bucket excavator employments and also demand no special abilities (except the ability to actually steer the machine). Those tasks could also be executed autonomously, with the driver only assuming the role of the supervisor of multiple excavators. This is the long-term goal of a cooperation between the Robotics Research Lab of the University of Kaiserslautern and VOLVO CE in Konz.

1.2 Related Work

Automating machinery is of course no new idea. Many publications have presented approaches for realizing either completely autonomous or semi-autonomous operation of construction machines or agricultural vehicles [1,2,3]
As for point cloud gathering, some publications were presented. Many of the presented approaches stem from the area of computer graphics [4] or indoor robotics with static scenes [5,6]. There are also publications in the area of outdoor robotics which cover tasks for navigation and/or mapping [7]. Only few approaches consider dynamic surroundings, like [8].

1.3 Point Clouds vs. other Data Structures

The design decision to use only slightly filtered point clouds instead of other primitives (e.g. planes, height maps, voxel maps) for storage purposes stems from the intention to use the data for different tasks. A plane representation for example is useful for trailer detection while a voxel map or a height field can be used for terrain shape estimation. The conversion into those data types from point clouds is possible, but the reverse direction may not always be, thus the preferred storage form was chosen to be a simple one.

1.4 Applications for this Work

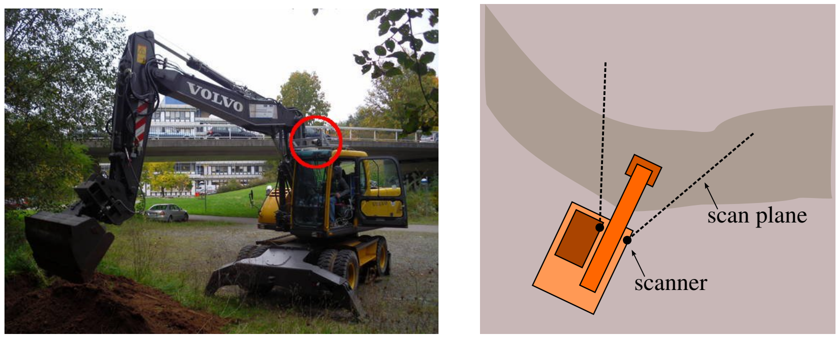
The data generated with this approach will be used for various tasks with the control scheme of the autonomous bucket excavator project [9]. A few examples are terrain shape estimation, human discovery, and vehicle detection and classification.

2 Concept

This section introduces the concepts created for the task at hand as well as some boundary conditions which have to be taken into account. The designed solution depends strongly on a few assumptions which restrict the system's versatility in favor of simplicity and runtime.

2.1 Utilizing Existing Hardware

The vehicle which is meant to be using the point cloud gathering system presented in this paper is a VOLVO EW180B bucket excavator (figure 1a). This vehicle has a mass of 18 tons. To prevent accidents while panning the excavator's cabin and arm two planar laser scanners (SICK LMS 151) are mounted on the chassis as shown in figure 1b, a kinematic model of the setup is shown in figure 3. The sensors have an evaluation field mechanism [10] which sets an alarm signal if the field is violated.
Having two valuable and dependable sensors already present on the vehicle and performing only one (albeit crucial) task leads to the proposition of a secondary use, which is gathering of data for the excavation task. Of course the laser scanners register only



(a) Real machine (b) Scanner mounting positions

Figure 1. VOLVO EW180B demonstrator vehicle

data from 2-dimensional planes, but since the "natural" motion of an excavator is panning, they are in constant motion. This allows for sampling 3-dimensional data without any additional mechanical sensor articulation, saving costs and making the whole system mostly maintenance-free.

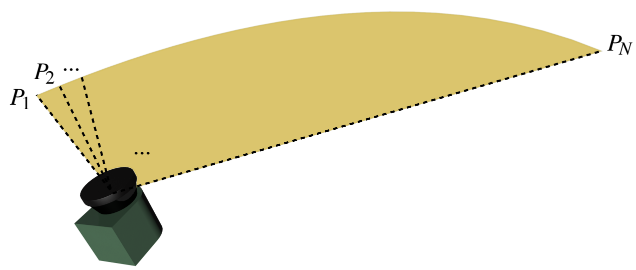


Figure 2. Every point scanned by a laser scanner is located on the scan plane

2.2 Requirements

When planning and implementing a system for data gathering the expected conditions and constraints have to be considered. Many approaches for similar tasks (see section 1.2) are designed to work in static indoor environments. The sensor noise is often negligible or can be smoothed using a filtering operation. In contrast, the application in rough construction site environments requires the explicit handling of noisy measurements originating from sensor vibration, dust, etc.
Additionally, the system has to be computationally inexpensive to deliver data quickly even on a cheap embedded computer. At first, two sensors have to be incorporated,

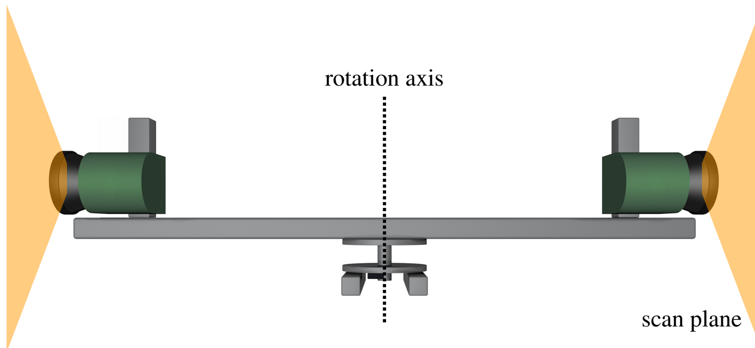


Figure 3. Schematic (front view) of one possible scanner configuration. Both scanners get information from the angular encoder located on the rotation axis.

but the system should be designed in a way that allows for easy extension to use more distance sensors.

2.3 Assumptions

Stationary vehicle The scans are performed from a stationary vehicle. The only motion is rotation (panning) about a specified axis. Registering point clouds from a moving vehicle would lead into the field of SLAM (e.g. [7]) which is not desired here. Also, the data storage is meant to represent only a local area and not the complete surroundings. If the excavator is to be displaced the current point cloud has to be erased. This assumption is no grave limitation for the usual employment of an excavator since most of the time the vehicle can remain stationary. Since the scan takes place as the excavator rotates the upper body in position current data is always gathered just before an excavation action.

Big obstacles For indoor robotics and 3-dimensional object scanning applications the goal is to create a finely-grained representation depicting the original object as closely as possible. In rough outdoor environments this is a Utopian goal. It was decided that a considerable amount of sensor noise and inaccurate measurements (originating e.g. from undetected translations or rotations of the lower body) have to be accepted. This means the sampled measurements have to severely filtered (see section 2.5). One of the drawbacks of filtering is the inability to distinguish fine and small structures in the surroundings. But on a construction site the object size can be assumed to be at least a few centimeters in each dimension.

Semi-dynamic terrain Since an excavator is a vehicle for active terrain shaping, the world view has to be constantly updated. Scanning the environment is a continuous process which is never fully completed and which scans portions of the terrain multiple times at somewhat regular intervals. Also, not all changes to the terrain are caused by the vehicle itself, so they can not be predicted. Multiple excavators could be working in close proximity, and of course sometimes the terrain changes without direct interaction, e.g. sliding soil.

Dynamic obstacles The most commonly expected objects encountered on a typical construction site are human workers and vehicles. These kinds of objects are expected to randomly appear and disappear from the scanner measurements. It is assumed that most objects which obstruct parts of the sensor view are mobile and therefore only short-lived.

2.4 Point Cloud Storage and Decaying Strategies

To prevent memory overflow when collecting and storing data over time a strategy for data discarding has to be devised. The universal solution for this problem is a simple bounded ring buffer (or a timestamp method) where the oldest data is overwritten by new data. Incorporating only such a decaying strategy would quickly lead to inconsistent data when using the system in a semi-dynamic environment. An object which appeared briefly in the measurements should also quickly disappear when it's not in view anymore. Distance data which is temporarily covered by a mobile object should not be discarded prematurely. The strategy employed in this work is a combination of a timestamp functionality (section 2.7) ensuring the removal of very old and possibly incorrect data and a removal strategy for evidently invalid points (section 2.6) based on the assumptions explained in section 2.3.

2.5 Error Volume

Assuming that the pose of the laser scanner is known exactly and does not change without notification the data coming from a laser scanner could be directly used for point cloud gathering. The intrinsic, systematic errors of the used scanners are minuscule in comparison, so if the sensor was absolutely stationary and the surroundings were static no further processing were necessary. But when measuring the distance from a potentially moveable platform, sensor pose errors have to be taken into account.

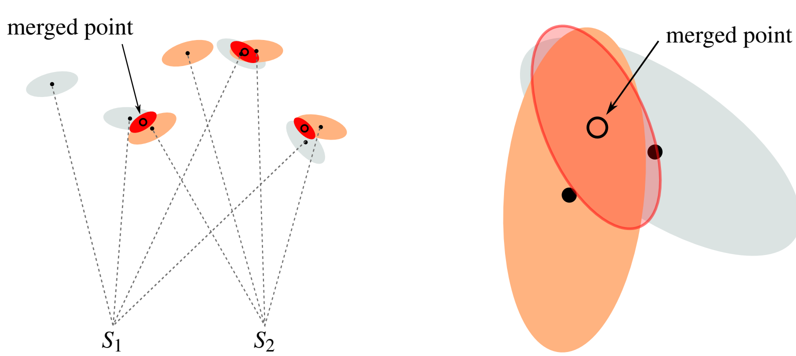


Figure 4. Error volume principle: Scan points from two scanners or from two sweeps with one scanner which are close to each other are merged into one single point.

For the solution presented in this publication, each scanner point is stored with a enclosing ellipsoid volume, called the *error volume*. This volume can be created by taking a certain quantile (e.g. 2σ) of a Gaussian distribution for each of the three deviations (distance error, left-right-error, up-down-error). Each scan point is located at the center of its volume. When new points are scanned, the overlap of their error volumes is checked against already stored data. If two volumes overlap, points are merged, and create a new point with a smaller error volume (figure 4). This represents the assumption that a point which is scanned multiple times can be localized with a higher accuracy. This approach also effects a point cloud thinning since it eliminates points in the immediate vicinity of already known points.

2.6 Deletion Volume

The method for identifying invalid points developed for this work is called the *deletion volume* method. The optical center of the scanner and all points from each scan are located on one plane, the scan plane (already shown in figure 2 at the beginning of section 2). The scan planes from two consecutive scans can be used to define a volume. This volume represents a portion of the environment the laser scanner has swept over. Since the laser rays have passed through this space and since according to the assumptions presented in section 2.3 there are no small objects in view, this volume can be assumed as being empty. This means all points in the point cloud which are inside this volume are invalid and should be deleted. A visualization of this principle is presented in figure 2.6. Unlike timestamps, this method has the advantage that points become invalid immediately during the scan sweep instead of staying in the point cloud until they deteriorate.

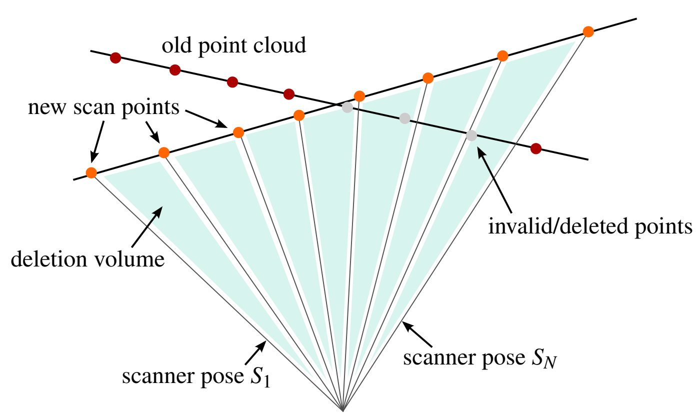


Figure 5. Principle of the deletion volume approach. Scans $S_1 \ldots S_N$ seen from above. Deletion volumes are spanned between two consecutive scans. Points located within the volumes are deleted.

2.7 Epoch System

The point merging approach shown in section 2.5 has one unfavorable effect. When the distance of new scan points during a sweep is smaller than the error volume, old points are immediately merged with new points, resulting in a dragging effect. To prevent this effect, a simple timestamp-like method has been used. Each point in the point cloud stores an *epoch* number. The difference of this epoch system to real timestamp methods is that the current epoch is not incremented at each time step. Instead, it is incremented when the scanner's rotation direction changes or if the scanner is not moving. Points are only merged when they belong to different epochs, the epoch of a point merged from two points is set to the higher epoch number. When the allocated point buffer overflows, points are removed beginning with the lowest epoch number.

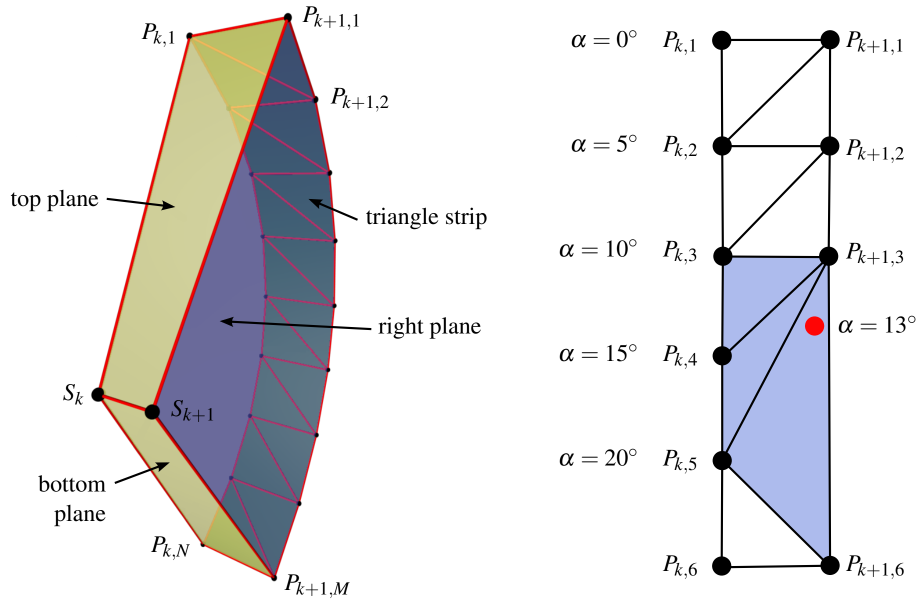
3 Implementation

The implementation of the system was performed within the MCA2-KL[1] framework. Most parts of the implementation correspond to the concepts described in section 2. The only change is to use spheres instead of ellipsoids for the error volumes. The benefit from this is the much quicker merging.
A kinematic chain was set up using Denavit-Hartenberg convention. This allows for easy inclusion of movable joints and the calculation of relative pose changes of the sensors. Right now, only one rotational encoder is used which is connected to both laser scanners. Thanks to the kinematic chain approach new scanners can be added quickly.

3.1 Efficient Error Volume Checking

The deletion volume introduced in section 2.6 was implemented using ray-triangle intersection tests. Given two consecutive scanner positions S_k and S_{k+1} and their respective scan points $P_{k,1} \dots P_{k,N}$ and $P_{k+1,1} \dots P_{k+1,M}$ a volume can be constructed from 4 planes and a triangle strip (see figure 6a). Note that in many cases the number of scan points of two consecutive scans can differ ($N \neq M$), as invalid scan points (e.g. with too high uncertainty) are discarded before deletion volume processing.
To check whether a point present in the already gathered point cloud is within the deletion volume, a few calculations have to be performed. The first principal check is calculated using the Hessian normal form of the four planes. This is a simple signed distance check operation. If all four distances are positive the currently handled point is possibly within the error volume. This steps eliminates a huge amount of candidates since the error volume is usually rather narrow.
The second and last check is to see if the point is located in front of or behind the triangle strip. This operation is potentially very costly since it can involve many ray-triangle intersection tests. To be absolutely sure a point is contained in the error volume it has

[1] http://rrlib.cs.uni-kl.de/mca2-kl/



(a) The deletion volume is defined by 4 planes and one triangle strip. The width of the volume is exaggerated.

(b) The triangle strip seen from the sensor's point of view. The red dot is the currently handled point. The blue triangles are the only candidates for intersection checks.

Figure 6. Deletion volume implementation details.

to be located between the sensor center and every of the (usual) 200–300 triangles. The method employed here is a series of simplifications reducing the calculations to a minimum.

If the surroundings haven't changed between two sensor sweeps, most of the scan points will be similar to the old data. Therefore it makes no sense to delete old points and to replace them with new, almost identical, points. The deletion volume is shrunk (the triangle strip is moved towards the sensor) to prevent unnecessary deletion. The point merging prevents the generation of too high densities.

When checking whether a point is between the scanner and the triangle strip, it has to be between the sensor and every triangle. Conversely, if the point is behind any of the triangles it is outside the error volume. Therefore the line-triangle-intersection is done from the point outwards using the method presented in [11].

Since the scan points are spaced out in an equiangular fashion on the scan plane the unfolded triangle strip yields equidistant points (the position is proportional to the scan point's index). By calculating the angle of a point the candidate triangles for intersections can be found very quickly. These triangles must have corner points with a greater angle as well as corner points with a smaller angle than the tested point (see figure 6b). The algorithm is presented below:

while *new scan* **do**
  calculate triangle strip $T = \{t_1, t_2, ..., t_N\}$;
  calculate boundary planes P_{left}, P_{right}, P_{top}, P_{bottom};
  foreach $p \in$ *point cloud* **do**
    if p *within* P_{left}, P_{right}, P_{top}, P_{bottom} **then**
      find intersection candidate triangles $t \subseteq T$;
      construct ray r from p in direction $(p - Origin)$;
      if r *intersects any* $t_k \in t$ **then**
        discard p;
      end
    end
  end
end

4 Tests and Results

The presented system was tested in a lab environment since unfortunately the real vehicle was not available at the time. The test setup is shown in figure 7. The scanners are mounted on a pannable mounting which can be actuated by hand.
Figure 8 shows a cross-section of the resulting point cloud. After the first sweep, there is a box covering a portion of the wall. Then, the box is removed and a second sweep is performed. As expected, the points belonging to the wall are included into the point cloud. Also, the points belonging to the box are removed since the scanner has seen through them. The box is added again and a third sweep is performed. New scan points showing the box are added to the point cloud. The wall points behind the box are still present. This is because potentially mobile objects are assumed to be visible only for a brief period of time. Points covered for a short amount of time stay in the memory. In the last image the box has been moved and again the system removes invalid points while keeping invisible, but possibly still valid points.

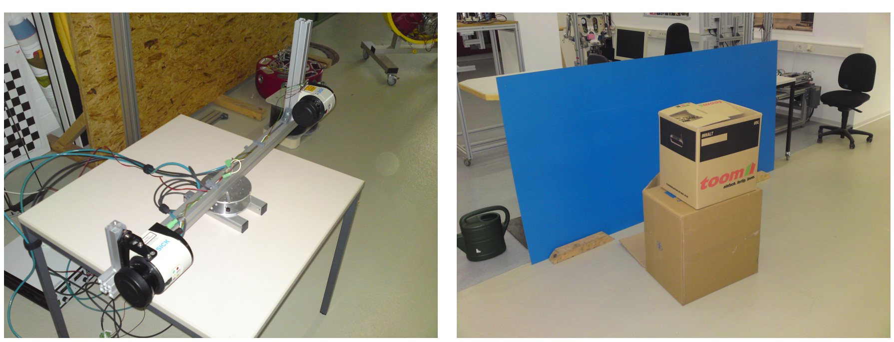
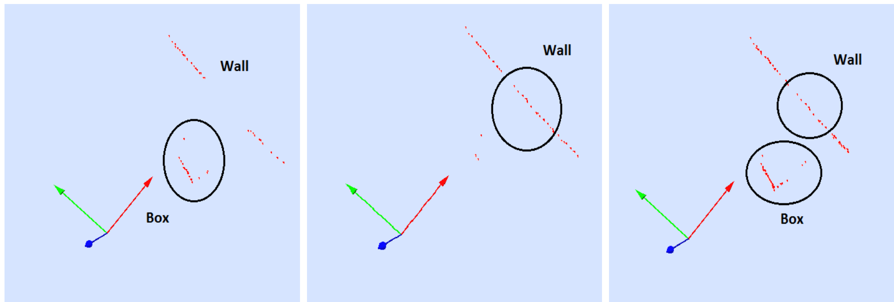


Figure 7. Test setup showing a two sensor arrangement sharing one rotational encoder. Note that the alignment of the sensors differs from the model in figure 3. The implemented kinematic chain compensates this. The scanned objects are a wall and a cardboard box.



(a) Initial scan (b) Box removed (c) Box re-added

Figure 8. Tests performed with the system. The tests were performed with 3-dimensional scans but for clarity only a cross-section (view from above) is shown

References

1. Rowe, P.: Adaptive motion planning for autonomous mass excavation. PhD thesis, Carnegie Mellon University (1999)
2. Koyachi, N., Sarata, S.: Autonomous wheel loader yamazumi-4. In: Proceedings of Robotics 2010, Clermont-Ferrand, France (September 3-4 2010)
3. Ha, Q., Santos, M., Nguyen, Q., Rye, D., Hugh Durrant-Whyte, H.: Robotic excavation in construction automation. Robotics Automation Magazine, IEEE **9**(1) (March 2002) 20–28
4. Huhle, B., Schairer, T., Schilling, A., Straßer, W.: 6dof registration of 2d laser scans. In: 3DIMPVT. (2011) 148–155
5. Wettach, J., Berns, K.: 3D reconstruction for exploration of indoor environments. In Berns, K., Luksch, T., eds.: Autonome Mobile Systeme 2007. Number 1 in Informatik aktuell, Kaiserslautern, Germany, Springer-Verlag (October 18-19 2007) 57–63
6. Stoyanov, T., Lilienthal, A.J.: Maximum likelihood point cloud acquisition from a mobile platform. In: International Conference on Advanced Robotics, 2009. ICAR 2009. (june 2009) 1–6
7. Cole, D.M., Newman, P.M.: Using laser range data for 3d slam in outdoor environments. In: Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006. (may 2006) 1556–1563
8. Dale, K., Cheslack-Postava, E., Humphreys, G., Luebke, D.P.: Scanning and reconstruction for dynamic surfaces. Technical report, University of Virginia (2006)
9. Schmidt, D., Proetzsch, M., Berns, K.: Simulation and control of an autonomous bucket excavator for landscaping tasks. In: IEEE International Conference on Robotics and Automation (ICRA), Anchorage, Alaska, USA (May 3-8 2010) 5108–5113
10. AG, S.: Operating Instructions for the SICK LMS100 Product Family. SICK AG Waldkirch. (2008)
11. Möller, T., Trumbore, B.: Fast, minimum storage ray-triangle intersection. Journal of Graphics, GPU, and Game Tools **2**(1) (1997) 21–28