



<http://www.diva-portal.org>

Postprint

This is the accepted version of a paper presented at *IEEE/RSJ International Conference on Intelligent Robots and Systems SEP 25-30, 2011, San Francisco, CA*.

Citation for the original published paper:

Magnusson, M., Almqvist, H. (2011)

Consistent pile-shape quantification for autonomous wheel loaders

In: *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 4078-4083). IEEE

IEEE International Conference on Intelligent Robots and Systems

<https://doi.org/10.1109/IROS.2011.6048680>

N.B. When citing this work, cite the original published paper.

Permanent link to this version:

<http://urn.kb.se/resolve?urn=urn:nbn:se:oru:diva-22334>

Consistent Pile-Shape Quantification for Autonomous Wheel Loaders

Martin Magnusson and Håkan Almqvist
Centre for Applied Autonomous Sensor Systems (AASS)
Örebro University, Sweden

Abstract—This paper presents a study of approaches for selecting an efficient attack pose when loading piled materials with industrial construction vehicles. Automated handling of piled materials is a highly desired goal in many construction and mining applications. The main contributions of the paper are an experimental study of two novel approaches for selecting an attack pose from 3D data, compared to previously published approaches and extensions thereof. The outcome is based on quantitative validation, both with simulated data and data from a real-world scenario with nontrivial ground geometry.

I. INTRODUCTION

A. Overview

Handling heterogeneous piled materials is a core component in many construction and mining applications. A typical work cycle of a wheel loader working in these applications (see Fig. 1) consists of three repeated tasks: loading, hauling, and dumping. Hauling between the load and dump points can be handled in a number of ways, whether by GPS-waypoint following, or some more flexible approach from the rich literature on mobile-robot navigation. Dumping is relatively straightforward and can, in principle, be performed with preprogrammed motions. Efficient loading is a harder problem than the dumping sequence, and no practical solution for fully autonomous vehicles exists today. For economical and environmental reasons, it is important that the bucket is filled maximally in each load cycle and that the mechanical stress on the machine is minimised. When an automated wheel loader approaches a gravel pile, then, it should first analyse the shape of the pile (from a 3D range scan) and evaluate potential *attack poses* along the pile edge; i.e., positions and orientations at which it is efficient to approach the pile.

This paper presents a comparison of previously published approaches and also presents the results of two novel methods for selecting attack poses. The outcome is based on quantitative experimental validation, both with simulated data and data from a real-world site.

The present paper contains several contributions.

a) *Head-to-head comparison*: Prior work on automated loading has only described single methods. This paper presents a comparative evaluation of two previous methods as well as two novel methods.

b) *Real-world evaluation*: Prior publications only present experiments in simulation or in lab-like setups. This paper presents evaluations on real-world data from an asphalt-production site with nontrivial ground geometry.

c) *3D extension of Singh and Cannon [12]*: A 3D extension of the 2D method that was published by Singh and Cannon [12] has been implemented and evaluated.



Fig. 1. The experimental platform in the target environment.

d) *Pile evaluation using a three-part bucket model*: As a way to overcome the drawbacks of the two-part bucket model of Singh and Cannon [12], a method that uses a three-part bucket model to estimate the pile properties is also presented.

e) *Pile evaluation using quadric fitting*: In addition to the aforementioned methods, a novel method for attack-pose evaluation is presented, which fits quadric surfaces in order to estimate the local convexity and skewness characteristics of the pile at each candidate attack pose.

f) *Sub-column moment estimate*: The results also demonstrate the benefit of using polygon clipping to increase the resolution of previous methods.

g) *Pile detection*: Another contribution is a method for classifying unordered point clouds into “pile” and “nonpile” segments. This problem has largely been ignored by previous authors, but is important in cluttered real-world environments.

B. Loading piled materials

It is not always evident which attack pose is optimal for a pile, but a number of heuristic criteria can be defined. One such criterion is that the wheel loader’s bucket should enter the pile flat on the ground. If not, it is more difficult to fill the bucket completely, and it will be more difficult to maintain a good pile profile. The pile should also be attacked at a slightly convex point, so that more of the pile volume is in the middle of the bucket than at the edges. Another criterion is to attack the pile so as to prevent an asymmetric load (sideloading the bucket). These criteria have also been noted by, e.g., Singh and Cannon [12]. Using these criteria, the search space for finding an optimal attack pose is one-dimensional: a point along the 2D edge profile of the pile. However, the profile of the pile may be different higher up than at the bottom edge.

Therefore a perpendicular attack pose at a convex point at the pile's bottom edge may still lead to an uneven load. For this reason, Sarata et al. [8] also consider other orientations than the pile's normal at each edge point, leading to a two-dimensional search space. The same approach is used in the present work.

II. PREPROCESSING

A. Pile representation

Given a point cloud acquired with some range sensor (e.g., an actuated SICK lidar), there are several possibilities as to how to process the points to create a pile model.

Sarata et al. have published a number of papers describing the development of an automated wheel loader [6, 7]. In their work, the pile is modelled as a set of columns, where each column stores the mean height of the points within the column — in other words, an elevation map. The column model requires a dense sampling of points; otherwise the columns must have wide bases in order to contain a sufficient number of points. Using a scan resolution of one degree, the distance between points on the scan line is 35 cm at 20 m range. Therefore, 50-cm columns were used for the results shown in this paper. With this resolution, holes in the column model would start appearing at 28.6 m range.

The advantage of using a column model is that it is easy to compute the volume of the pile from the columns, and since the columns effectively subsample the point cloud, handling the column model requires less time and memory than operating on the full point cloud.

Another alternative is to use the full point cloud, and to estimate the surface of the pile with a triangulated mesh. Triangulated meshes, however, are not trivial to create. As long as the pile model is created from a single, ordered, point cloud (as can be expected from an actuated lidar) it is easy to add meaningful triangles in the ideal case. However, depending on the surface reflectivity, there may be missing points, in which case a raster-based triangulation will fail even on an ordered point cloud. This is likely to happen, e.g., in loading applications at an asphalt-production site, where piles of milled asphalt present a particularly challenging surface. In the more general case of modelling a pile from a registered *set* of scans, more time-consuming triangulation algorithms would have to be employed. For the present work we used semi-3D Delaunay triangulation. Assuming a ground-plane model exists, the volume of the pile can easily be computed by summing the volume of the triangular columns between the triangle faces and the ground.

The main advantage of the triangle model compared to a column model, of course, is the increased resolution, which should lead to more accurate volume estimates.

Without explicit surface generation in the form of triangulation or a column model, the raw point cloud can also be used. In this case, a local surface parametrisation can be estimated “on the fly”; e.g., by fitting a quadric surface to a local neighbourhood of points.

The main advantages of this approach is that no potentially time-consuming triangulation preprocessing is required, and

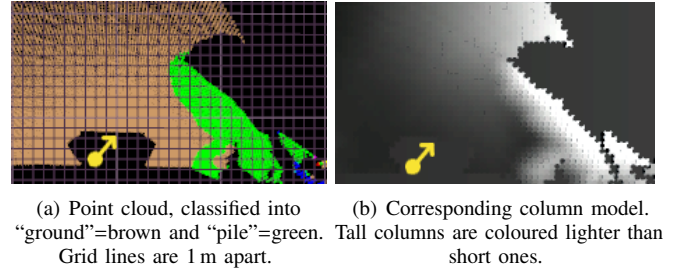


Fig. 2. Failure of using a column model to classify pile regions. In these figures, the scan data are viewed from above. The approximate viewpoint of the camera in Fig. 1 is indicated by the yellow arrow. The approach proposed in Section II-B has been used to correctly detect the pile in the lower right area of Fig. 2(a). The part of the ground in the upper area is higher than most parts of the pile, as can be seen by the column heights in Fig. 2(b), but does not have a surface slope and shape that is pile-like.

also that it is not susceptible to artifacts in the volume estimate caused by holes in the triangle mesh or missing columns (as is otherwise likely to happen with sparse or uneven data).

B. Pile detection

The first problem encountered by an autonomous wheel loader as it approaches a pile is to reliably detect which parts of the current view that belong to a pile. This problem has not been explicitly handled in previous publications.

The work of Sarata et al. uses the heights of the columns to locate the pile, which assumes that the pile is the only object on a flat floor, as in the publications showcasing their method [11, 2]. However, in a real environment such as the one depicted in Fig. 1, it is not a viable solution to only group areas of a certain height. Surface slope and shape must also be considered in order to discriminate between the pile (to the right in the image) and the drivable slope (to the left) in this case. See Fig. 2 for an illustration of the problem of classifying piles using height alone. The same problem is also present in cluttered work-site environments, with other machines and objects that may be large but are not piles.

A better solution is to consider also the surface orientation and roughness within a local neighbourhood of each point. The method outlined in the following paragraph is similar to the method for boulder detection in piles presented by Magnusson [3].

For each scan point \mathbf{p} , perform principal component analysis (PCA) to find the eigenvectors and eigenvalues of the distribution of surface points within a local neighbourhood radius ρ . The sorted eigenvalues $\lambda_1 \leq \lambda_2 \leq \lambda_3$ describe the shape of the surrounding surface. Points that are planar enough (with $\lambda_1 \ll \lambda_2 \leq \lambda_3$; i.e., with a distribution that is not linear or spherical) and have the right inclination (such that the angle between the corresponding eigenvector \mathbf{e}_1 and the horizontal plane is within two angle thresholds α_1 and α_2) are classified as “pile”. Planar points with an angle below α_1 are classified as “ground”. Nearby points with the same class are then clustered and segmented (using radially-bounded nearest-neighbour clustering [1]), and only the “pile” clusters are used for the succeeding pile analysis.

C. Ground-plane detection

In order for the volume estimates of the triangle- and column-based representations to be meaningful, a ground-plane estimate is also required. In the present implementation, MLESAC [13] is used to fit a plane to the points classified as “ground” by the preceding PCA pile detection. The scan is then rotated and translated so that the ground plane coincides with the xy -plane in the local coordinate frame.

III. ATTACK-POSE EVALUATION

This section contains the heart of the paper: a comparison of several ways to evaluate potential attack poses, including related work published on the topic.

Once the points that belong to the pile have been selected, the system needs to search along the bottom edge of the pile in order to find an efficient attack pose. In this search, the heuristics described in Section I-B are used; i.e., pile convexity and bucket sideload.

Potential attack poses are selected from each scan point that is at the border of the pile, and close to the ground plane. The orientation of the attack pose is taken along the normal of the border point. (In the presented results, only the normal direction is used, in order to make the presentation clearer. In the real application, alternative orientations at an angle offset up to $\pm 10^\circ$ are also considered.) A set of quantities estimating the local convexity and sideload characteristics are computed for each potential attack pose.

A. Column-model moment estimates

Sarata et al. [8, 9, 10, 11, 2] evaluate poses using a model of the moments acting on the bucket, computed from a column model. For each column i inside the trajectory that is followed by the bucket when it has entered the pile, the moment is computed from the height h_i of the column and the lateral distance w_i to the bucket’s centre. The selected pose is the one that minimises $A'_C = \sum_i h_i w_i$.

In our implementation, the columns are clipped (using Sutherland-Hodgman polygon clipping) so as to include only the part of each column that falls inside the bucket trajectory, as opposed to using discrete columns [9]. In other words,

$$A_C = \sum_i v_i w_i, \quad (1)$$

with v_i denoting the partial volume of column i that is inside the bucket trajectory.

B. Two-part bucket model

Singh and Cannon [12] use a 2D bucket model split in two, and examine the area that falls into each half in order to judge the convexity and sideload at each attack pose. Their two-part bucket model approach has been demonstrated in a 2D simulation. The implementation used for the present paper uses a straightforward extension using volumes computed from the triangulated 3D point cloud instead of areas.

With V_l and V_r denoting the pile volume within the left and right half of the bucket, the convexity measure is simply

$$B_C = V_l + V_r. \quad (2)$$

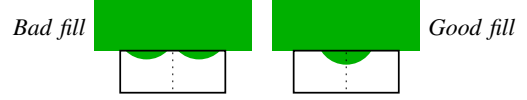


Fig. 3. Failure of B_C convexity measure. Pile profile shown in green, bucket shown as black outline. Both cases would give similar values of B_C , while C_C would correctly label the left case as concave and thus undesirable.

In the work of Singh and Cannon, potential attack poses are generated by tracing the two front corners of the bucket model along the edge of the pile. This way of selecting potential poses is slightly different from the one used for the methods in Sections III-A, III-C, and III-D. The convexity measure B_C depends on this particular way of selecting attack poses. If the attack pose is not such that the bucket corners touch the pile edge, B_C is not a relevant measure of convexity. The sideload is measured as

$$B_S = \frac{|V_l - V_r|}{V_l + V_r} \quad (3)$$

so that $B_S = 0$ is a perfectly even distribution and $B_S = 1$ is the case when all of the volume is in one half of the bucket.

This approach is quite sensitive to the point sampling of the pile, especially at places that are only mildly convex. Because only a small part of the bucket model enters the pile when the front corners touch the edge, it can easily happen that $B_S = 1$ even when the bucket enters the pile perpendicularly. It is clear that a model of the whole bucket-fill trajectory is preferable, both for convexity and side-load estimation, but then another convexity measure is required.

C. Three-part bucket model

An alternative to the two-part bucket model of Singh and Cannon is to use a three-part model. Let V_c be the volume inside the centre third of the bucket and V_l and V_r the left and right thirds. The convexity can then be modelled as

$$C_C = \frac{V_c}{\max(V_l, V_r)} - 1, \quad (4)$$

which is a more explicit convexity measure than the area, or volume, inside the bucket used in (2). Convex areas have $C_C > 0$ and concave areas have $C_C < 0$. The two-part convexity measure B_C cannot discriminate between perfectly flat and concave points. To illustrate a further problem with the B_C convexity measure, consider Fig. 3.

The three-part sideload is measured (similarly to the two-part model) as

$$C_S = \frac{|V_l - V_r|}{V_l + V_r}. \quad (5)$$

D. Quadric fitting

Disregarding a triangle- or column-based volume estimate, the convexity and sideload can also be estimated by fitting a quadric surface to the points within the local neighbourhood of a potential attack pose. The local neighbourhood is taken to be all the points that fall into the volume traced by the bucket trajectory.

For a point \mathbf{p} on a smooth surface, the principal directions are the directions in the tangent plane with minimum and

Algorithm 1 Estimate local curvature at a surface point \mathbf{p} .

- Let $\mathcal{P} = \{\mathbf{p}^i\}_{i=1}^n$ be a set of points in the neighbourhood of \mathbf{p} , expressed in the global coordinate frame.

- 1) Fit a plane to the points in \mathcal{X} using total least squares. Use the plane's normal as an estimate of the surface normal \mathbf{n} at \mathbf{p} .
- 2) Construct a rotated principal frame \mathbf{X}_r (7).
- 3) Map the points of \mathcal{P} into \mathbf{X}_r (6):

$$\mathbf{p}_r^i = \begin{bmatrix} x_r^i & y_r^i & z_r^i \end{bmatrix}^T = \mathbf{R}_r(\mathbf{p}^i - \mathbf{p}). \quad (10)$$

- 4) Fit the rotated principal quadric

$$z_r = ax_r^2 + bx_r y_r + cy_r^2 + dx_r + ey_r \quad (11)$$

to the mapped points.

- The horizontal curvature corresponds to a . The sideload can be estimated by the horizontal slope term d .
-

maximum curvatures. The principal coordinate frame $\mathbf{X}_p = [\mathbf{x}_p \ \mathbf{y}_p \ \mathbf{z}_p]^T$ is an orthonormal frame [5] that has \mathbf{x}_p and \mathbf{y}_p aligned with the principal directions, and \mathbf{z}_p aligned with the surface normal \mathbf{n} . In the principal frame, the principal quadric is a second-order description of the surface [4].

A common approach for quadric fitting (adapted from McIvor and Valkenburg [4]) is listed in Algorithm 1. It makes use of a *rotated principal frame* \mathbf{X}_r . For pile convexity estimation, it is mostly the horizontal curvature that is interesting. The horizontal curvature may not correspond to the principal curvature. Therefore, a rotated principal frame, defined such that one axis is aligned with the surface normal and one is parallel to the ground, should be used instead. Consider the “world” coordinate frame $\mathbf{X}_w = [\mathbf{x}_w \ \mathbf{y}_w \ \mathbf{z}_w]^T$ with \mathbf{x}_w and \mathbf{y}_w in the ground plane, and \mathbf{z}_w pointing to the sky. The transformation for a point \mathbf{p}_w^i from the world coordinate frame to the rotated principal frame for point \mathbf{p}_w is

$$\mathbf{p}_r^i = \mathbf{R}_r(\mathbf{p}_w^i - \mathbf{p}_w). \quad (6)$$

The rotated principal frame can be defined using

$$\mathbf{R}_r = \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{r}_3 \end{bmatrix}^T, \quad (7)$$

with $\mathbf{r}_3 = \mathbf{n}$, $\mathbf{r}_2 = (\mathbf{z}_w \times \mathbf{n}) / (||\mathbf{z}_w \times \mathbf{n}||)$, $\mathbf{r}_1 = \mathbf{r}_3 \times \mathbf{r}_2$.

The rotated principal frame is related to the principal frame \mathbf{X}_p by a rotation around \mathbf{n} .

For fitting the parameters of the rotated principal quadric (step 4 of Algorithm 1), the BFGS quasi-Newton algorithm is used in our implementation. Empirical tests have shown that, for this problem, the BFGS algorithm is faster than using the true Hessian of the minimisation problem. Furthermore, line search was found to be faster than a trust-region approach for limiting the step size in each iteration.

The quantities used for evaluating attack poses are taken from (11), with convexity and sideload

$$D_C = -a, \quad (8)$$

$$D_S = |d|. \quad (9)$$

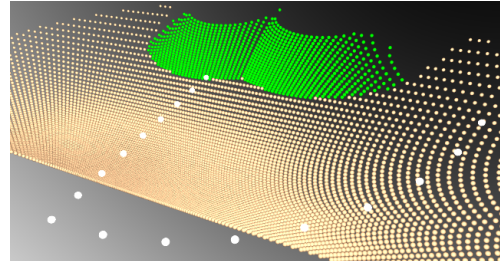


Fig. 4. The simulated data set. Colours denote the ground-truth classification of points into “pile” (green/dark) and “nonpile” (brown/light) points. White points show simulated sensor positions for Section IV-B.

IV. EXPERIMENTS

This section presents experimental evaluations of the methods introduced above. The scan data used for these evaluations is publicly available at <http://aass.oru.se/~mmn/#data>.

A. Validity

First, evaluations of the algorithms described in Section III using a simulated pile scan will be presented. In this case, laser scan points were generated by ray tracing in a scene containing pile-like geometric primitives (cones and planes). The simulated data set is shown in Fig. 4. The pile in this case consists of two cones, simulating a pile with a small hollow running down the centre.

A small test demonstrating how well the different quantities describe the convexity and sideload of the simulated pile is shown in Fig. 5. The convexity measures B_C , C_C , and D_C all show non-convex values for the central region. The effect is seen most clearly for C_C and D_C , and by the fact that $B_C=0$. However, the plots also demonstrate the high noise levels in A_C (mainly due to the tessellation of the pile model) even though the central region generally has higher values than the surroundings, as it should.

The sideload measures C_S and D_S mostly agree on small sideload values along the rounded convex areas of the cones. B_S is substantially more noisy. Because of the low convexity, only a very small volume is inside the bucket model at the points evaluated by this method, which causes the ratio between the two halves to fluctuate. Still, in this case, all the methods would succeed in choosing to attack the pile at a convex area, perpendicular to the pile edge.

B. Consistency

A good shape measure should give consistent values independently of the viewpoint at which the pile is observed or the sensor configuration. In a real scenario where the machine is constantly scanning and evaluating the pile while moving towards it, it is important that the top-candidate attack pose does not jump along the pile edge.

To measure consistency, we computed the value of the quantities for several sets of scans, using one set of edge points. All scans in each set were registered to the same coordinate frame. The per-point standard deviation was used to measure the consistency for each quantity.

In order to produce comparable values of the quantities, which have different magnitudes, the statistics in Table I are

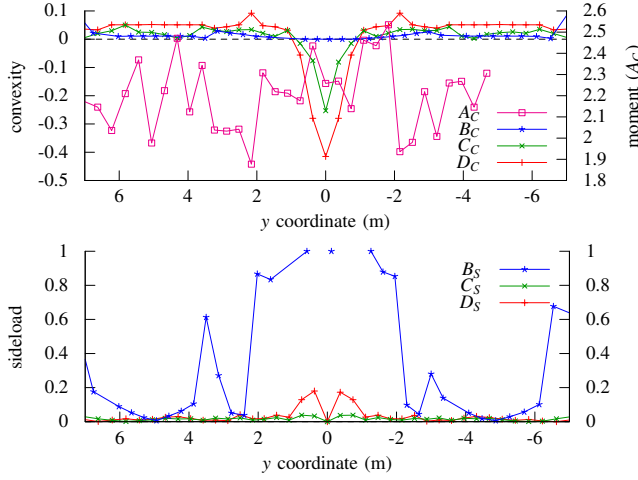


Fig. 5. Plotting one set of evaluation quantities for the simulated pile.

TABLE I

CONSISTENCY EVALUATION, SHOWING AVERAGE NORMALISED STANDARD DEVIATIONS.

	Simulated				Real
	distance	view angle	sweep speed	sensor noise	
A_C	0.789	0.956	0.724	0.960	0.35
A'_C	0.954	0.981	0.861	0.982	0.43
B_C	0.150	0.272	0.100	0.281	0.29
B_S	1.250	1.217	1.025	1.154	1.24
C_C	0.137	0.707	0.156	0.946	0.52
C_S	0.163	0.879	0.280	0.950	0.64
D_C	0.088	0.290	0.094	0.341	0.21
D_S	0.176	0.550	0.173	0.616	0.26

computed after normalisation. Each quantity is normalised by subtracting the total mean value and dividing by the total standard deviation (computed over all edge points and all scans). In the table, the normalised *average* standard deviation is presented; i.e., the mean of the normalised standard deviations (over the n scans) of all edge points.

1) *Distance*: Eight scans of the simulated pile were created, with sensor positions at distances between 6 m and 20 m from the pile edge with intervals of 2 m (always at a height of 3 m above the ground).

As can be seen in the first column of Table I, D_C has the smallest variance of the convexity measures (\bullet_C). The variance of A_C is substantially larger, and looking at the A'_C values, it can be seen that the corresponding values computed with discrete columns [9] are even worse. The variance of B_S is also quite large, which can, again, be explained by the fact that B_C and B_S only integrate over a small portion of the pile edge.

2) *View angle*: In order to test the stability with respect to viewing angle, another eight scans were simulated, each at 20 m distance from the hollow at the centre of the simulated pile, but following a circular arc from 0° to 70° . The simulated scanner positions in these tests and the distance tests of Section IV-B.1 are shown with white spheres in Fig. 4.

In this case, the values shown in Table I are computed only for the right half of the pile (from 0 m to -5.5 m). Otherwise the influence of occlusions would cause the numbers to

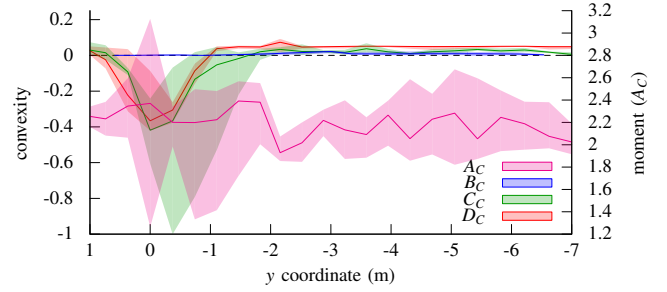


Fig. 6. Consistency of convexity measures for different view angles. Ideally, the width of each band of values should be zero.

be higher. Occlusions are especially problematic for the triangulation-based quantities (B_\bullet and C_\bullet).

Even using only this half, the values are spread much wider for C_C and C_S than for D_C and D_S . In Fig. 6 another weakness of the A_C measure can be observed. For one of the scans, the global minimum of A_C is at $y = 0$. This point is located at the centre of the pile's hollow and is the worst attack pose of all. The reason is that poses at which little volume is covered by the bucket trajectory also result in a lower A_C moment, whether or not it is a good attack pose.

3) *Scanner sweep speeds*: For this test, scans with simulated scanner sweep speeds between 0.1 rad/s and 0.7 rad/s were generated. The effect of a lower sweep speed is higher vertical scan resolution. All scans use the same 1-degree horizontal resolution. The results can be seen in the third column of Table I. Using a higher sweep speed has a similar effect to scanning from a larger distance (although when scanning from further away the effective horizontal resolution is also affected). Indeed, the standard deviations for this test are similar to the ones for the distance test.

4) *Scanner noise*: The stability to sensor noise was also evaluated. For this test, each scan point was perturbed by Gaussian noise with standard deviation from zero up to 10 cm along the direction of the corresponding ray.

As can be seen in the fourth column of Table I, the variance is generally larger for this test than for the other, as is to be expected for this amount of noise. The general trend that the convexity and sideload measures from quadric fitting, D_C and D_S , are more stable than the other can still be observed. The only exception is that B_C has lower variance in this case, because it is computed from a smaller part of the pile.

C. Real-world data

The methods have also been evaluated on real-world pile data. Only the results for one gravel pile (see Fig. 7) are included in the paper, because of space constraints.

It is not trivial to obtain ground-truth data for the optimal attack pose of any real-world pile. We have discussed optimal attack procedures with human operators but it is rarely the case that there exists a single optimal pose that we could measure the distance from. Not being able to compute a meaningful distance to a ground-truth optimal attack pose, the presented real-world results instead show the same kind of stability analysis as for the simulated data. The evaluation methods were applied to five scans from different viewpoints

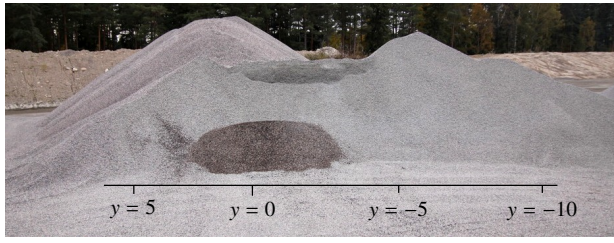


Fig. 7. Photo of the pile analysed in Fig. 8.

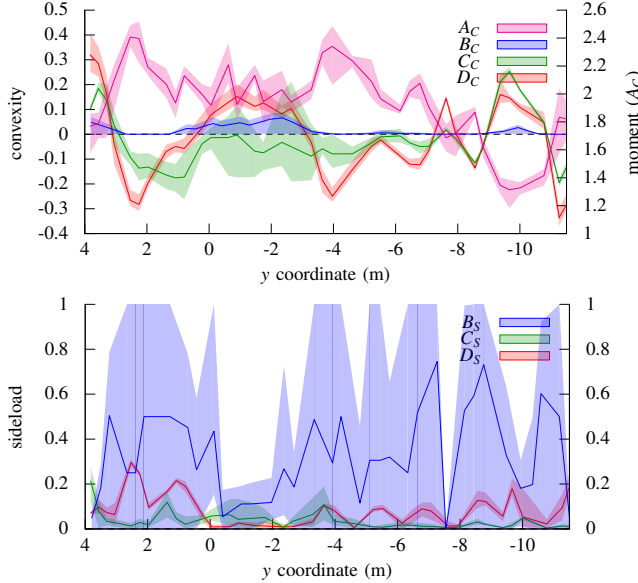


Fig. 8. Outcomes for the real-world data. The three convex areas on this pile are at $y = -10$ m, -3 m to 1 m, and 4 m.

registered to the same coordinate frame. Fig. 8 shows the values of the different pile-shape quantification methods for the real-world pile. The sensitivity results are summarised in the rightmost column of Table I.

The variances are slightly larger in this case than for the corresponding tests with the simulated data. What's more important to note is how the convexity measures straddle zero in some cases (see the upper plots in Fig. 8). This could have the adverse effect that an attack pose that is selected when the vehicle starts driving toward the pile is later regarded as unacceptable, causing disruptions for the vehicle's path planner. This problem is most pronounced around the bulge between 0 and -3 m (the dark patch in Fig. 7), for all values except D_C , and also at a point at -6 m for C_C . The same problem can also be seen in the sideload plots in Fig. 8. There are many points along the pile edge where the C_S and B_S values are small in one scan but large in other scans.

D. Computation time

Average computation times per pose evaluation are as follows: Sarata (A_C): 1.8 ms, Singh/Cannon (B_C and B_S): 1.5 ms, three-part model (C_C and C_S): 5.5 ms, quadric fitting (D_C and D_S): 1.6 ms. These times do not include triangulation, generation of the column model, or pile detection. The three-

part bucket model is about three times slower than the other three methods. However, all methods can be performed in just a few milliseconds per point.

V. CONCLUSIONS AND FUTURE WORK

We conclude that using quadric fitting to estimate the convexity (D_C) and sideload (D_S) is a good solution in an automated loading scenario. Compared to previously published methods for the same task, it is more stable to differences in viewpoint, sensor resolution, and sensor noise. The accuracy in that it correctly identifies good attack poses is as good as or better than the other methods studied in this paper. The execution time is also fast. It does not require preprocessing in the form of triangulation or the creation of an elevation map, which makes the speed benefit even larger.

This paper has discussed methods for quantifying convexity and sideload. However, it is still an open question which combination of these surface characteristics leads to the most efficient bucket fill. Future work should study the effect of different weighting schemes of these two characteristics, also weighting poses by the estimated cost of driving there. Future work will also investigate operation over longer sequences and in varying soil conditions.

ACKNOWLEDGEMENT

This work is funded by a grant from the Swedish Knowledge Foundation (dnr 2008/0513). The authors would also like to acknowledge the assistance of Torbjörn Martinsson and Jonatan Blom in discussing pile-attack approaches.

REFERENCES

- [1] K. Klasing, D. Wollherr, and M. Buss. A clustering method for efficient segmentation of 3D laser data. In *ICRA*, pages 4043–4048, Pasadena, USA, May 2008.
- [2] N. Koyachi and S. Sarata. Unmanned loading operation by autonomous wheel loader. In *Proceedings of the ICROS-SICE International Joint Conference*, pages 2221–2225, Aug. 2009.
- [3] M. Magnusson. *The Three-Dimensional Normal-Distributions Transform — an Efficient Representation for Registration, Surface Analysis, and Loop Detection*. PhD thesis, Örebro University, Sweden, Dec. 2009. Örebro Studies in Technology 36.
- [4] A. M. McIvor and R. J. Valkenburg. A comparison of local surface geometry estimation methods. *Machine Vision and Applications*, 10: 17–26, 1997.
- [5] S. Petitjean. A survey of methods for recovering quadrics in triangle meshes. *ACM Computing Surveys*, 34(2):211–262, 2002.
- [6] S. Sarata. Model-based task planning for loading operation in mining. In *IROS*, pages 439–445, Maui, USA, Oct. 2001.
- [7] S. Sarata, H. Osumu, Y. Kaway, and F. Tomita. Trajectory arrangement based on resistance force and shape of pile at scooping motion. In *ICRA*, pages 3488–3493, New Orleans, USA, Apr. 2004.
- [8] S. Sarata, Y. Weerakamhaeng, and T. Tsubouchi. Approach path generation to scooping position for wheel loader. In *ICRA*, pages 1809–1814, Barcelona, Spain, Apr. 2005.
- [9] S. Sarata, Y. Weerakamhaeng, and T. Tsubouchi. Planning of scooping position and approach path for loading operation by wheel loader. In *ISARC*, Ferrara, Italy, Sept. 2005.
- [10] S. Sarata, N. Koyachi, T. Tsubouchi, H. Osumi, M. Kurisu, and K. Sugawara. Development of autonomous system for loading operation by wheel loader. In *ISARC*, pages 466–471, 2006.
- [11] S. Sarata, N. Koyachi, and K. Sugawara. Field test of autonomous loading operation by wheel loader. In *IROS*, pages 2661–2666, Nice, France, Sept. 2008.
- [12] S. Singh and H. Cannon. Multi-resolution planning for earthmoving. In *ICRA*, Leuven, Belgium, May 1998.
- [13] P. H. S. Torr and A. Zisserman. MLESAC: a new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding*, 78(1):138–156, Apr. 2000.