Integrating Feedback from Image Based 3D Reconstruction into Models of Excavation Processes for efficient Scheduling

Maximilian Bügler[1], Henning Metzmacher[2], André Borrmann[1], Christoph van Treeck[2]

[1]Chair of Computational Modeling and Simulation, Technische Universität München, Germany
[2]Chair of Energy Efficiency and Sustainable Building, RWTH Aachen University, Germany

max.buegler@tum.de, metzmacher@e3d.rwth-aachen.de

Abstract. The progress of construction sites is in many cases unexpectedly delayed by external influences, for instance weather, changes in the client's demands or involvement of residents. This may cause changes in the construction cycle. The detailed initial plans become invalid and it is required to immediately develop new plans taking the new circumstances into account. This will allow construction companies to avoid unnecessary allocation of resources, idle time and costs. In this paper methods to determine the volume of excavation pits based on point clouds obtained by image based 3D reconstruction are investigated. Furthermore it explains how to introduce the results into a model of the construction project which can be used to create new schedules. Additionally the possible benefits of re-planning project schedules after observing a task to be faster or slower than expected are described.

1. Introduction

Unexpected delays in construction sites create huge costs each year. In order to be able to react correctly it is required to acquire information about the progress as efficiently as possible. The excavation of pits is a major factor for many construction sites since most tasks of the project require a pit to be finished. Previously reported methods of recording the progress using laser scanning technology have some major pitfalls since the equipment required for laser scanning is very expensive and requires trained personnel. While there are many ways to collect data on construction sites (Horenburg et al., 2012), this paper investigates the usability of point clouds created using computational photogrammetry for estimating the volume of a partially excavated pit in order to make predictions about the time still required to finish it. Those predictions can then be used to create a new optimal schedule for the project, adapting to possible delays in the excavation process. Patch-based Multi-view Stereo (PMVS) (Furukawa and Ponce, 2010) is used for dense point cloud generation from a set of photographic images. The interest region is extracted by generating a k-nearest neighbor graph and through subsequent processing of the resulting graph components. A convex hull is generated from the final point cloud segment which is used to estimate the volume of the excavation site. In order to calculate optimal schedules an Integer Linear Programming formulation of the problem is used (Mingozzi et al., 1998) in combination with a discrete event simulation model.

2. Process Model

The simulation model is implemented as a discrete event simulation (DES). The tasks, resources and respective dependencies are importet from a XML file. Resources include workers, machines, the areas available to place equipment and expendable supplies. The simulation takes provided performance factors of machines and allocation of resources into account. After a simulation has

been performed, a Gantt Chart, visualizing the order of task execution, is produced. The resulting schedule can then be optimized with respect to execution time and costs (Wu et al., 2010). Furthermore data recorded during construction can be included to update the schedule.

In order to do this, a Mixed Integer Programming (MIP) approach is used. This approach is based on maximizing (or minimizing) a given objective function (See Equation 1), while the variables are subject to a number of constraints (See Equation 2) (Dantzig, 1951). To convert the problem into a form that can be solved using a MIP solver, it is necessary to define the objective function and all constraints required for a complete representation.

$$\begin{aligned} &\text{maximize} && \mathbf{c}^{\mathrm{T}}\mathbf{x} && (1)\\ &\text{subject to} && A\mathbf{x} \leq \mathbf{b} && (2)\\ &\text{and} && \mathbf{x} \geq \mathbf{0} && (3) \end{aligned}$$

where $\mathbf{x}$ is a vector of variables to be optimized, $\mathbf{c}$ is the cost vector, A contains the left hand sides of the constraints, and $\mathbf{b}$ contains the right hand sides.

The variables $\mathbf{x}$ are defined as the starting times for the individual processes and each precedence constraint is converted into a linear constraint of the following form:

$$t_A + d_A \leq t_B \tag{4}$$

where t_A is the start time and d_A is the duration of task A.

When the precedence constraint does not require task A to be finished, but only to be started, when task B starts, the constraint can be simplified to:

$$t_A \leq t_B \tag{5}$$

To define constraints for resource usage and blockage of geometric areas, the forbidden sets approach is used (Schaeffter, 1997). Each combination of tasks that cannot be executed simultaneously, due to resource constraints, is defined by a set of constraints. The following constraints prevent tasks B and C to be executed in parallel.

$$t_B + d_B - Mx_i \leq t_C \tag{6}$$
$$t_C + d_C + Mx_i - M \leq t_B \tag{7}$$

where x_i is a binary variable switching the precedence case and M is a very large number (Griva et al., 2009).

The first constraint forces B to be executed prior to C, given $x_i = 0$, while $x_i = 1$ disables this constraint by subtracting the very large number M from the left hand side. The opposite holds for

the second constraint, which forces B to be executed after C given $x_i = 1$.

In order to perform time optimization on this problem an artificial task is introduced, which lists all other tasks as precedence constraints, so this task is executed after all tasks have finished. The objective function therefore only consists of the starting time of this final task, that is minimized.

A problem that arises when solving the resulting MIP, is the potentially large number of possible forbidden sets. Consider 10 tasks, of which each requires one resource of the same type, while 5 of those resources are available. The number of forbidden sets would be $\binom{10}{6} = 210$, which is the number of subsets of size 6. Since each forbidden set introduces a number of binary variables into the problem, the time required to solve it in a deterministic manner increases exponentially (Florian et al., 1980). Therefore the best projection heuristic is used (Brady and Catanzaro, 2008) as well as the heuristic by Driebeck and Tomlin (Tomlin, 1971) included within the GNU Linear Programming Toolkit (GLPK, 2013) to obtain a solution. Additionally forbidden sets of a size larger than a certain threshold are neglected in order to reduce the number of binary variables. Afterwards the result is simulated using the discrete event simulator to verify the feasibility of the resulting schedule. If a larger forbidden set turns out to render the schedule unfeasible, the respective constraints are added for this set and the procedure is repeated. This will always result in a feasible schedule.

3. Data Acquisition

Recent advances in the field of computational photogrammetry allow for the detailed reconstruction of 3D-models from photographic images. Through a combination of image feature point extraction using, for example, the SIFT algorithm (Lowe, 2004), bundle adjustment algorithms and information about photo-consistency and visibility consistency as used in the PMVS algorithm (Furukawa and Ponce, 2010) it is possible to generate dense, colored 3D-point clouds. Point clouds can provide valuable information about a scene such as volumetric data about an object. This paper presents a way to extract and measure the volume of an excavation site and subsequently integrate the data into the process model described in Section 2. To prove the feasibility of the approach the developed methods were tested on a real-world excavation pit. The image datasets were acquired using a conventional HD camcorder extracting a frame every second. The point cloud was generated using VisualSFM and PMVS (Wu, 2007, 2011; Wu et al., 2011). No fixed calibration or EXIF data was provided to VisualSFM as the software can estimate intrinsic camera parameters on the fly. In the following, an algorithm for the extraction of the excavation site geometry is given.

The initial dataset is shown in Figure 1a. An undirected k-nearest neighbor graph is generated from the point cloud based on the euclidean distance (Equation 8) between each pair of points.

$$d_{p,q} = \sqrt{(x_p - x_q)^2 + (y_p - y_q)^2 + (z_p - z_q)^2} \tag{8}$$

Depending on the dataset this generates a high amount of isolated connected components. In order to exclude outliers and points that are unlikely to be significant from the dataset, components with an order below a manually defined threshold t_c are discarded. Hence, a component is discarded if Equation 9 holds. The resulting point cloud can be seen in Figure 1b.

$$|V| < t_c \tag{9}$$

In the next step, each graph component c_i is searched for the point with the lowest y-coordinate. The component is discarded if its lowest point is above the average y-coordinate μ_y of the entire dataset times a factor a_y. This is illustrated in Equation 10.

$$min_y(c_i) > \mu_y \times a_y \tag{10}$$

Setting $a_y = 0.5$ seems to yield good results. Figure 1c shows the resulting point cloud. This step is implemented under two assumptions. At first, it is assumed that the ground of the excavation site represents the lowest part of the point cloud. Therefore, it is desired to exclude graph components with their lowest points much higher than this. Second, it is based on the assumption that the excavation site forms multiple large clusters that connect the ground to the walls. Thus, the walls of the excavation site are not cut off. Subsequently, the resulting extracted components are again filtered for outliers starting from their highest point. Tests showed that this is necessary in order to deal with high objects within the excavation site. The remaining components are processed as follows: Search for the point p_h with the highest y-coordinate (y_h). For a number of iterations n count the vertices where the conditions of Equation 11 and 12 are true.

$$y > y_h + y_{interval} \times i \tag{11}$$

$$y < y_h + y_{interval} \times (i + 1) \tag{12}$$

Here y is the y-coordinate of the respective point, $y_{interval}$ is a defined iteration interval and i is the number of the current iteration. If the number of vertices in that range is below a defined threshold t_h then the all points from the interval are discarded. This step is done to exclude points that are connected to the excavation site components but have points that higher than the assumed edge. This is often the case for cranes and similar machines within the site.

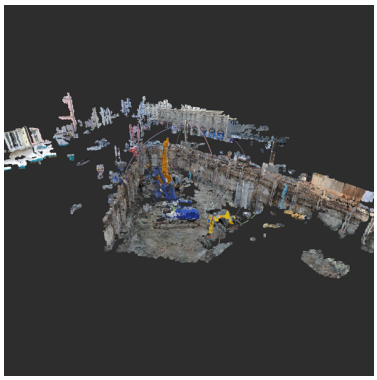
In order to estimate the volume of the extracted point cloud, first a remeshing algorithm to compute a convex hull from the points is employed. This is done using the quick hull algorithm (Barber et al., 1996) which is included in the Qhull C libary. Applying this algorithm results in a closed, triangulated mesh (See Figure 1f). Subsequently, an algorithm is applied to re-orient the face normals of the mesh coherently using an implementation in the Meshlab software. To compute the volume estimate, an technique described by Mirtich (Mirtich, 1996) which is included in the VCG computer graphics library is used. The algorithm is also accessible through the Meshlab interface.

The procedure is applied to both the point cloud reconstructed from the image collection (See Figure 1d - 1f) and a reference point cloud acquired from a laser scan (See Figure 1g - 1i). The resulting measurements are shown in Table 1. The estimate for the volume and the surface area of the reconstructed point cloud deviates from the laser scanned dataset by approximately 10%. The results shown in Table 1 are scaled arbitrarily as for evaluation purposes only the relative accuracy is of interest.

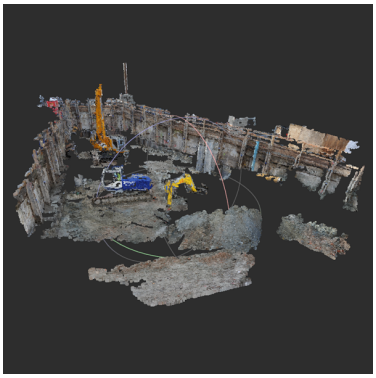
Dataset	Mesh Volume	Mesh Surface
Camera	132.953125	197.169189
Laser	146.417969	210.682938

Table 1: Geometric measures of the camera and the laser scanned pointcloud.

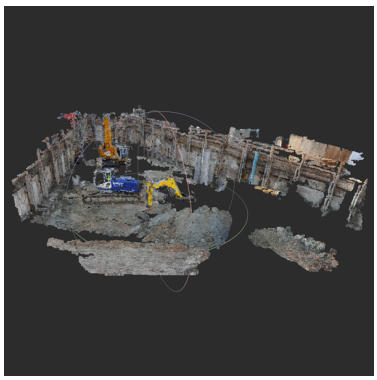
The accuracy of the reconstructed point cloud is highly dependent on the distance of the camera to the construction site but also on the resolution of the camera itself. Depending on the image resolution and camera distance, PMVS is able to achieve accuracy to the millimeter, however, the necessary image coverage might be infeasible. Hence, there is a trade-off between the complexity of the input data and the accuracy of the final measurement.



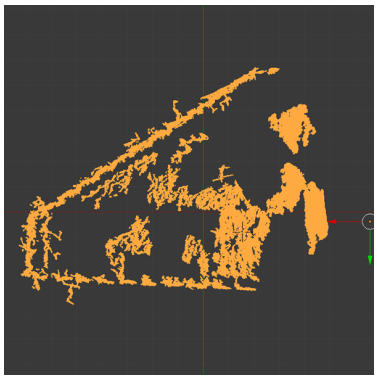
(a) The initial point cloud dataset.



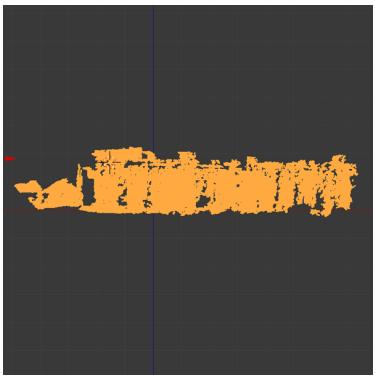
(b) The excavation site after discarding surrounding components.



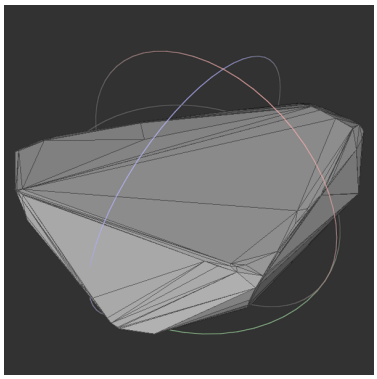
(c) The dataset after discarding top vertices.



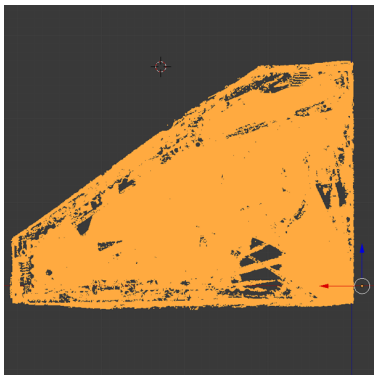
(a) Camera reconstruction top view.



(b) Camera reconstruction side view.

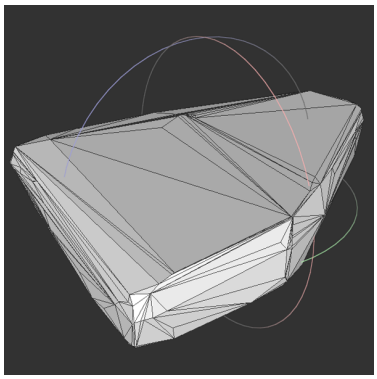


(c) Camera reconstruction convex hull.



(d) Laser scan top view.

(e) Laser scan side view.



(f) Laser scan convex hull.

4. Data Integration

In order to include data recorded from the construction site, a new constraint is introduced for each process that was executed and expected durations of processes which are currently in execution are updated. Given a task A was executed at time t_1 a new constraint is introduced (13), which constitutes the value for the associated variable t_A. When a task's progress was observed and does not match the expectations, so is either going to finish later or earlier, the durations in the constraints involving this task need to be updated. This is in most cases at least one instance of constraints (4) and (5) and multiple instances of constraints (6) and (7).

$$t_A = t_1 \tag{13}$$

To illustrate the necessity of re-planning a schedule, as soon as more information is available, an illustrative example project is provided. This project features four tasks, three resources and two dependencies and is illustrated in Figure 1. It features two connected excavation areas A and B on a slope and each area is fitted with a wall (C and D) on the back to hold back the soil. Tasks A and B both need an excavator and tasks C and D need a team of workers to build the walls. While there are 2 excavators, there is only one team of workers, preventing both walls to be built simultaneously. Figure 2 shows the created optimal schedule without any additional information used. Now the first task A is assumed to be delayed by an amount d. Since the schedule demands the dependency successor C to be executed next, which blocks the final task, the entire schedule is prolonged by a time span d. Therefore the delay of the first task is propagated through the entire schedule. This is illustrated in Figure 3. When the schedule is re-planned after the delay was recorded, Figure 4 shows that it would now be more efficient to swap the last two tasks. This results in the shorter overall delay of d_2.

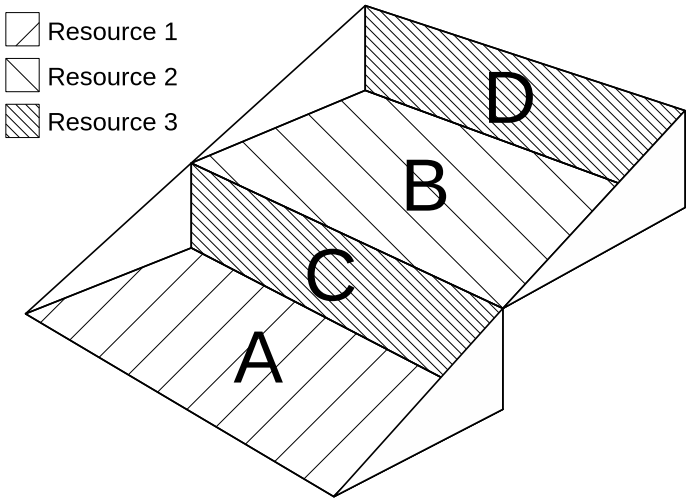


Figure 3: Example case

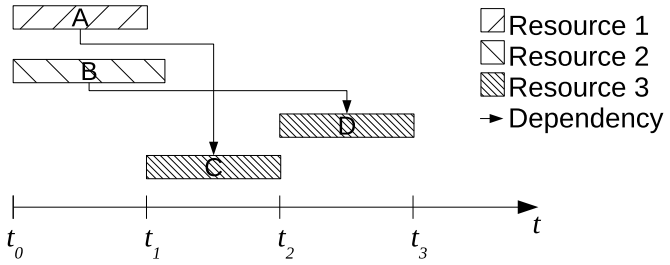


Figure 4: Initial schedule

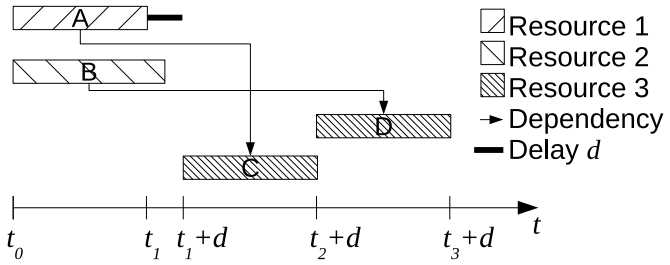


Figure 5: Initial schedule with delayed task

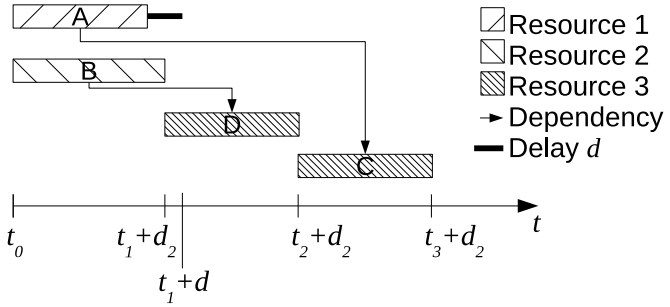


Figure 6: Updated schedule with delayed task after re-planning

5. Conclusions

Introducing feedback from construction sites into process simulations can help to reduce the effects of delayed tasks on the overall execution time. This paper provided an illustrative example of a project where current progress data helps to finish the overall project in less time, compared to continuing the initial schedule. In order to obtain data on the progress of excavation tasks that were presented an approach employing photogrammetric 3D reconstruction and subsequent processing of a point cloud. It was shown how a point cloud can be segmented into graph components using a nearest-neighbor search and proposed various domain dependent assumptions for the extraction of the excavation site. Moreover, the generation of a convex hull was proposed in order to efficiently compute a volume estimate which in turn was used as a metric for the progress ratio of the excavation task.

6. Future Work

The progress information obtained through the volume estimation can be used to draw conclusions on the consistency of the soil or the efficiency of the workers and equipment. The information could be used to make estimations for delays of other excavation tasks on the site. This would allow for further improvements on the efficiency of the schedules. Several possible extensions for the point cloud segmentation as well as for the volume calculation can be suggested. Since every point generated using the PMVS algorithm contains a normal vector, normals could be employed to achieve a more robust extraction of the excavation site. Moreover, feature sets for identifying pits in a 3D point can could be identified in order to train automatic classifiers for the segmentation task. For the volume computation, a more accurate hull can be computed using Poisson Surface Reconstruction (Kazhdan et al., 2006) or a marching cube approach.

References

C. B. Barber, D. P. Dobkin, and H. Huhdanpaa. The quickhull algorithm for convex hulls. *ACM Transactions On Mathematical Software*, 22(4):469–483, 1996.

B. Brady and B. Catanzaro. Pseudo-boolean heuristics for 0-1 integer linear programming. In *Proceedings of the Design, Automation and Test in Europe Conference and Exhibition*, pages 132–135, 2008.

G. B. Dantzig. Maximization of a linear function of variables subject to linear inequalities. *New York*, 1951.

M. Florian, J. K. Lenstra, and A. R. Kan. Deterministic production planning: Algorithms and complexity. *Management science*, 26(7):669–679, 1980.

Y. Furukawa and J. Ponce. Accurate, dense, and robust multi-view stereopsis. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 32(8):1362–1376, 2010.

GLPK. *GNU Linear Programming Kit - Reference Manual*. Free Software Foundation, 2013. URL `https://www.gnu.org/software/glpk/`.

I. Griva, S. G. Nash, and A. Sofer. *Linear and nonlinear optimization*. Society for Industrial Mathematics, 2009.

T. Horenburg, J. Wimmer, W. Günthner, A. Bormann, and G. Dori. Towards collection, processing and use of actual data for the process simulation during construction. In *Proceedings in International Workshop: Intelligent Computing in Engineering 2012*, 2012.

M. Kazhdan, M. Bolitho, and H. Hoppe. Poisson surface reconstruction. In *Proceedings of the fourth Eurographics symposium on Geometry processing*, SGP '06, pages 61–70, Aire-la-Ville, Switzerland, Switzerland, 2006. Eurographics Association. ISBN 3-905673-36-3. URL `http://dl.acm.org/citation.cfm?id=1281957.1281965`.

D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60 (2):91–110, Nov. 2004. ISSN 0920-5691. doi: 10.1023/B:VISI.0000029664.99615.94. URL `http://dx.doi.org/10.1023/B:VISI.0000029664.99615.94`.

A. Mingozzi, V. Maniezzo, S. Ricciardelli, and L. Bianco. An exact algorithm for the resource-constrained project scheduling problem based on a new mathematical formulation. *Management Science*, 44(5):714–729, 1998.

B. Mirtich. Fast and accurate computation of polyhedral mass properties. *Journal of Graphics Tools*, 1:31–50, 1996.

M. W. Schaeffter. Scheduling with forbidden sets. *Discrete Applied Mathematics*, 72(12): 155 – 166, 1997. ISSN 0166-218X. doi: 10.1016/S0166-218X(96)00042-X. URL http://www.sciencedirect.com/science/article/pii/S0166218X9600042X.

J. Tomlin. An improved branch-and-bound method for integer programming. *Operations Research*, 19(4):1070–1075, 1971.

C. Wu. Siftgpu: A gpu implementation of scale invaraint feature transform (sift). 2007. URL http://cs.unc.edu/ ccwu/siftgpu.

C. Wu. Visualsfm: A visual structure from motion system. 2011. URL http://homes.cs.washington.edu/ ccwu/vsfm/.

C. Wu, S. Agarwal, B. Curless, and S. M. Seitz. Multicore bundle adjustment. *CVPR 2011*, 2011.

I.-C. Wu, A. Borrmann, U. Beißert, M. König, and E. Rank. Bridge construction schedule generation with pattern-based construction methods and constraint-based simulation. *Advanced Engineering Informatics*, 24(4):379–388, 2010.