A Robotic Excavator for Autonomous Truck Loading

ANTHONY STENTZ, JOHN BARES, SANJIV SINGH AND PATRICK ROWE
The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213

Abstract. Excavators are used for the rapid removal of soil and other materials in mines, quarries, and construction sites. The automation of these machines offers promise for increasing productivity and improving safety. To date, most research in this area has focussed on selected parts of the problem. In this paper, we present a system that completely automates the truck loading task. The excavator uses two scanning laser rangefinders to recognize and localize the truck, measure the soil face, and detect obstacles. The excavator's software decides where to dig in the soil, where to dump in the truck, and how to quickly move between these points while detecting and stopping for obstacles. The system was fully implemented and was demonstrated to load trucks as fast as human operators.

Keywords: autonomous excavation, robotic excavator, integrated robotic system, laser rangefinder, software architecture, manipulator, dig planning

1. Introduction

The surface mining of metals, quarrying of rock, and construction of highways require the rapid removal and handling of massive quantities of soil, ore, and rock. Typically, explosive or mechanical techniques are used to pulverize the material, and digging machines such as excavators load the material into trucks for haulage to landfills, storage areas, or processing plants. As shown in Figure 1, an excavator sits atop a bench and loads material into trucks that queue up to its side. The operator is responsible for designating where the truck should park, digging material from the face and depositing it in the truck bed, and stopping for people and obstacles in the loading zone.

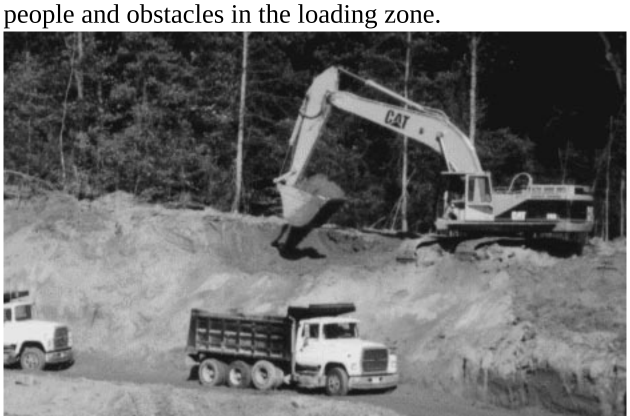

Figure 1. Excavator loading a truck with soil in a typical mass excavation work scenario.

The opportunities for automation are immense. Typically, loading a truck requires several passes, each of which takes 15 to 20 seconds. Reducing the time of each loading pass by even a second translates into an enormous gain across the entire job. The operator's performance peaks early in the work shift and degrades as the shift wears on. Scheduled idle times, such as lunch and other breaks, also diminish average production across a shift. All of these factors are areas where automation can improve productivity.

Safety is another opportunity. Excavator operators are most likely to be injured when mounting and dismounting the machine. Operators tend to focus on the task at hand and may fail to notice other site personnel or equipment entering the loading zone. Automation can improve safety by removing the operator from the machine and by providing complete sensor coverage to watch for potential hazards entering the work area.

Numerous researchers have addressed aspects of automated earthmoving (Singh, 1997). The lowest and most common level of automation has been teleoperation. Typically, the operator is removed from the scene for reasons of safety. Teleoperated excavators are used in applications that pose a danger to humans, such as the uncovering of buried ordnance (Nease and Alexander, 1993) and waste (Burks et al., 1992; Wohlford et al., 1990), or excavation around buried utilities. A higher level of autonomy is achieved by systems that share control of the excavation cycle with a human operator. Typically, these systems (Bradley et al., 1993; Bullock and Oppenheim, 1989; Huang and Bernold, 1994; Lever et al., 1994; Rocke, 1994; Sakai and Cho, 1988; Salcudean et al., 1997; Sameshima and Tozawa, 1992; Seward et al., 1992) concentrate on the process of digging. An operator chooses the starting location for the excavator's bucket and a control system takes over the process of filling the bucket using force and/or joint position feedback to accomplish the task. At the next level of autonomy are systems that automatically select where to dig. Such systems measure the topology of the terrain using ranging sensors (Feng et al., 1992; Singh, 1995; Takahashi et al., 1995) and compute dig trajectories that maximize excavated volume. At the highest level of autonomy are systems that sequence digging operations over a long period (Bullock et al., 1990; Romero-Lois et al., 1989; Singh, 1998).

The prior work addresses many subproblems important for autonomous truck loading, however in order to field a fully automated system that performs at the level of its manually operated equivalent, a much broader set of problems must be solved than just digging. Sensors are needed to sense the dig face, recognize and localize the truck, and detect obstacles in the workspace. Perception algorithms are needed to process the sensor data and provide information about the work environment to the planning algorithms. Planning and control algorithms are needed to decide how to work the dig face, deposit material in the truck, and move the bucket between the two.

We have developed a complete system for loading trucks fully autonomously with soft materials such as soil. The Autonomous Loading System (ALS) was implemented and demonstrated on a 25-ton hydraulic excavator and succeeded in loading trucks as fast as an expert human operator. The rest of the paper describes the ALS and presents results from experimental trials.

2. System Overview

The Autonomous Loading System uses two scanning laser rangefinders that are mounted on either side of the boom (see Figure 2) to sense the dig face, truck, and obstacles in the workspace. Two scanners are needed for full coverage of the workspace and to enable concurrent sensing operations. Each sensor has a sample rate of 12 kHz, and a motorized mirror sweeps the beam circularly in a vertical plane. Additionally, each scanner can pan at a rate of up to 120 degrees per second, enabling this circle to be rotated about the azimuth, as shown in Figure 3. The scanner positioned over the operator's cab is called the "left scanner", and it is responsible for sensing the workspace on the left hand side of the excavator. The "right scanner", which is located at a symmetric position on the right side of the boom, is responsible for sensing the workspace on the right hand side of the excavator.

Figure 2. Sensors mounted on excavator.

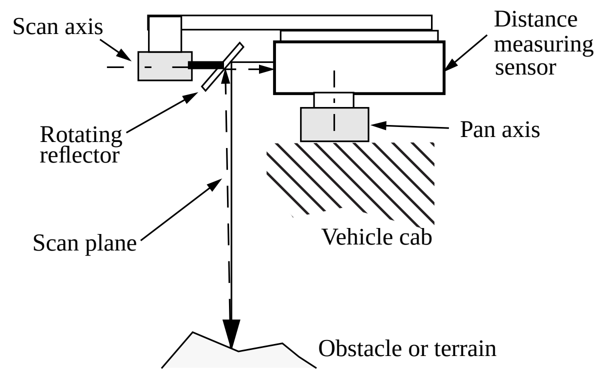

Figure 3. Two axis scanning sensor configuration.

The excavator uses its scanners in the following fashion when loading a truck (Figure 4). While the excavator digs its first bucket, the left scanner pans left from the dig face across the truck both to detect obstacles and to recognize, localize, and measure the dimensions of the truck. Using this information, a desired location in the truck to dump the material is planned, and the bucket swings toward the truck. During this swing motion, the right scanner pans left across the dig face to measure its new surface, and the next location to dig is calculated. The right scanner continues to pan toward the truck. After the soil is dumped into the truck, the right scanner pans back across the dig face to detect obstacles in the way of the implements. The excavator swings back to the next dig point. During this swing, the left scanner pans across the truck to measure the soil distribution in the truck bed, and the next desired dump location is calculated. This process repeats for each subsequent loading pass until the truck is full, with the exception that truck recognition is only necessary for the first pass for each new truck. Typically, six passes are needed to load our twenty-ton truck with our excavator testbed.

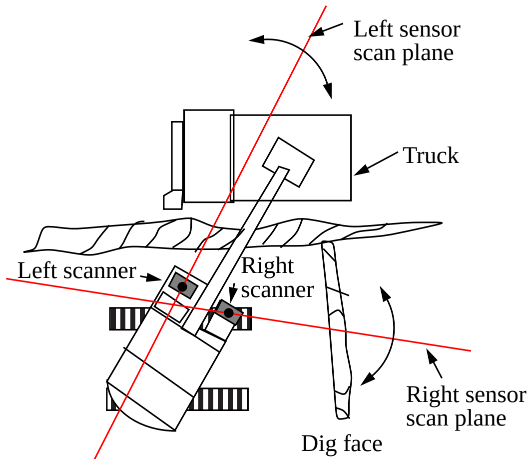

Figure 4. Top view of sensor configuration.

Information from the scanners is processed using an on-board array of four MIPs processors. The software architecture is shown in Figure 5. The boxes are software modules that can run on one of the system processors. Circles are hardware components such as sensors. Lines represent communication channels. The *sensor interfaces* receive data from the two scanners and control the panning motion of the devices. Sensor data from the interfaces are passed to *scanline processors*, where they are converted from spherical, sensor coordinates to Cartesian, world coordinates using corresponding data from the *position system*. These three-dimensional range points are then made available to whatever perception software modules require them.

One consumer of this processed sensor data is the *truck recognizer*, which recognizes the truck and measures both its dimensions and location. Two others are the *dig point planner*, which plans a sequence of dig points for eroding the dig face, and the *dump point planner*, which plans a sequence of dump points for loading soil into the truck bed. The *digging motion planner* controls the excavator during digging at the specified location. The *dumping motion planner* dumps the bucket of soil into the truck and returns to the dig face. The *sensor motion planner* controls the panning for both scanners to coordinate scanner and excavator motion, following the scenario described above. The *obstacle detector* processes sensor data from the scanner that is sweeping in advance of excavator's motion and stops the machine if an obstacle is detected in its path. The *machine controller interface* communicates commands to the low level machine joint controller, which executes the commands and sends excavator state information back to the planning modules.

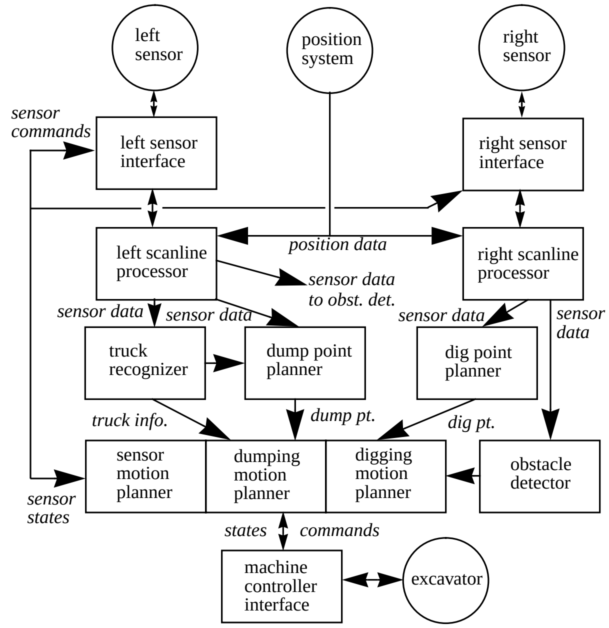

Figure 5. ALS software architecture.

3. Hardware Subsystem

The ALS hardware subsystem consists of the servo-controlled excavator, on-board computing system, perception sensors, and associated electronics. In this paper we focus on the perceptual sensors which provide the data from which the truck is identified, the dig location is chosen, obstacles are avoided, and ultimately the mass excavation process is achieved.

With the target application of earthmoving, we focussed on developing a laser based scanning system that would be able to penetrate a reasonable amount of dust and smoke in the air. The laser itself would need to be able to accurately measure range from a variety of target materials (e.g., metals, wood, dirt, rock, snow, ice, and water), colors and textures. We also needed a system that would be robust to dust and dirt accumulating on the protective "exit" window (glass or plastic which protects the laser and optics from weather and dirt, though permits the beam to pass).

Over the past decade, a variety of laser based scanners have been produced. With the exceptions of the Dornier (Shulz, 1997) and Schwartz (Schwartz) scanners, most have either been research devices or limited to indoor usage. None that we know of addresses the problems of dust penetration or a partially occluded (i.e., dirty) exit window.

We have developed two different time-of-flight scanning ladar systems that are impervious to ambient dust conditions. The first uses a "last-pulse" technique that observes the waveform of the returned light and rejects early returns that can arise from internal reflections off of a dirty exit window, or from a dust cloud obscuring the target (see Figure 6). In general, the next-to-last pulse returns are due to dust in the scene and are indicative of what a normal "first pulse" rangefinder would see. For instance, in Figure 6, a first pulse rangefinder would detect the dirty exit window and would be unable to "see" the target. Even if the window were clean, the first pulse unit would still "see" the dust cloud instead of the target. Since reflections off the exit window are rejected with the last pulse technique, the unit can be environmentally sealed using an inexpensive transparent cover that does not have to be optically perfect or clean. Another advantage is that the laser system can also report when multiple returns occur, giving a warning that dust is present. This is important because overall ranging reliability and accuracy is decreased in dusty conditions, so an autonomous machine might need to adopt a slower, more conservative motion strategy.

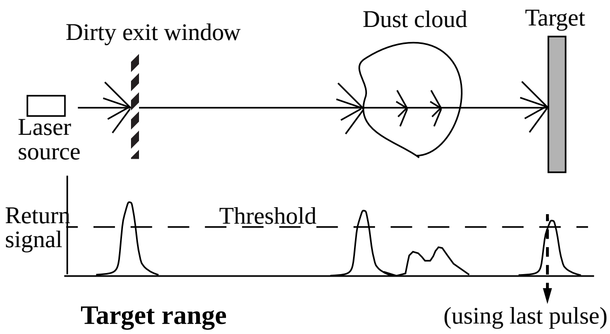

Figure 6. Last pulse detection concept.

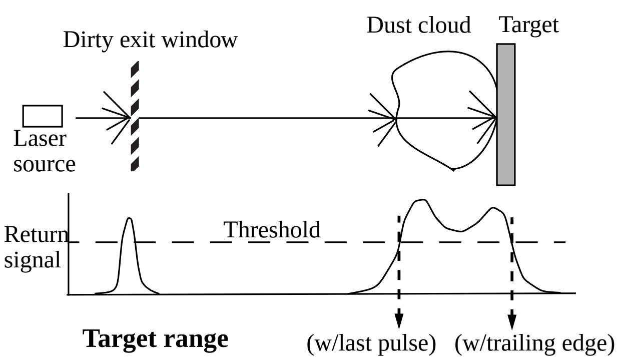

Figure 7. Trailing edge detection of target when target is obscured in dust cloud.

There is, however, a limitation to last-pulse rangefinding. When the target is within the dust cloud, the receiver electronics can have difficulty separating the dust and target returns (see Figure 7). We have built a second dust penetrating scanner system that identifies that target by locating the "trailing edge" of the last return signal as is shown in Figure 7. Like the last pulse system, this device is also robust to occlusions on the exit window making it ideal for construction and mining environments. Though the trailing edge detection technique forgoes some range accuracy, we believe it is a superior approach for environments where the dust may frequently surround the target.

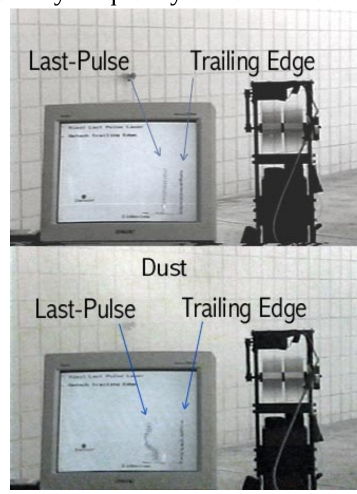

Figure 8. Last pulse vs. trailing edge detection when target is within dust cloud.

The television monitor pictured in Figure 8 shows range points plotted from a single scanline for both the last pulse and trailing edge scanners. Range increases from the left to the right of the monitor. The top monitor screen shows scans of the rear of a dump truck. The bottom screen shows scans of the same truck but shrouded in a heavy dust cloud. Note that the last pulse device is unable to separate the dust cloud from the truck and reports the front of the cloud. The trailing edge device correctly reports range to the truck regardless of the presence of dust.

It is important to note that both dust penetrating techniques are physically limited by very heavy dust levels that attenuate the return target signal below the point of detectability.

4. Software Subsystem

The software subsystem consists of several software modules that process sensor data, recognize the truck, select digging and dumping locations, move the excavator's joints, and guard against collision. In this section, the algorithms employed by key software modules in the software architecture are described.

4.1. Truck Recognition

In order to properly load a truck, an excavator operator must verify that it is a loadable vehicle, determine its location, and determine its dimensions. This information is essential for calculating a loading strategy and for planning the sequence of joint motions that implements this strategy.

In some scenarios, such as surface mining, the loaders are serviced by a mine-owned fleet of haulage trucks. An automated system could acquire this information by equipping each truck with a global positioning system (GPS) sensor and an identification transponder. However, in other scenarios such as highway construction, the loaders are serviced by a variety of independently-owned, on-highway trucks of varying dimensions, so equipping each and every truck with such sensors could be infeasible. For such scenarios, an automated system could acquire the necessary information using rangefinder data.

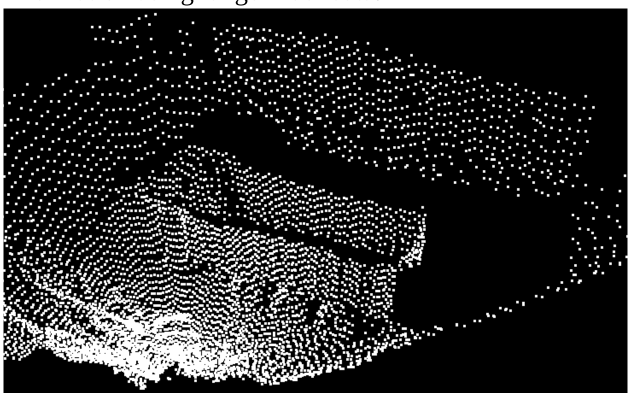

Figure 9. Raw range data of a truck.

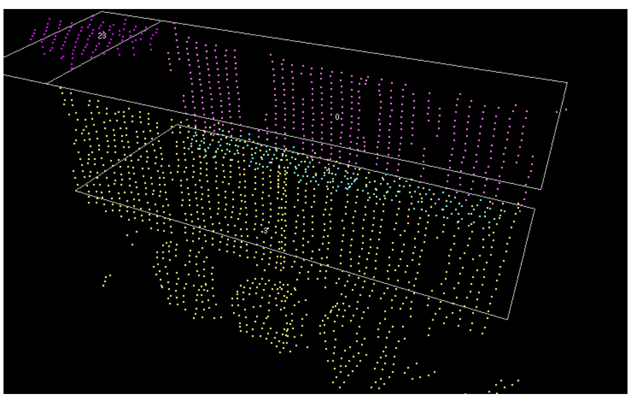

Figure 10. Truck model fit to segmented data.

The truck recognizer uses sensor data to automatically recognize, localize, and dimension haulage trucks. As the excavator digs its first bucket of soil, the left scanner pans across the truck, which is assumed to be parked to the excavator's side. The raw sensor data are shown in Figure 9. Each rotation of the mirror returns one vertical scanline of data, created by intersecting a vertical plane with the truck. Each scanline is processed into line segments which are grouped with coplanar line segments from other scanlines to form planar regions.

Using an interpretation tree approach (Grimson, 1990), the simple model for a truck bed, shown in Figure 10, is matched to the segmented data region by region. Depth-first search is used to hypothesize model-to-scene region matches. At each level in the tree, constraints are used to prune the search and to check for consistency with previously hypothesized matches. The interpretation that matches most of the model regions and survives the verification stage is selected as the correct one. In order for the truck recognizer to recognize a class of truck models rather than just a single model, the model in Figure 10 uses parameter ranges rather than single parameter values. Ranges are used on the sizes of the planar regions in the model, the locations of their centroids relative to each other, and the angles between the planes. These parameter ranges are checked for consistency at every level in the interpretation process to prune the search. This specification allows the truck recognizer to identify trucks of varying sizes and truck bed shapes.

For each complete interpretation (i.e. an attempt to match all model regions to scene regions), the truck recognizer performs a verification. The verification consists of finer-grained consistency checking of truck parameters, and the identification of the four "corner points" in the sensor data that define the opening of the truck bed. For the selected interpretation, the corner points are used to calculate the position and orientation of the truck bed. This information is passed to other modules in the system for producing a dumping strategy. Figure 10 shows the model matched to the planar regions segmented from the raw sensor data.

4.2. Coarse-to-Fine Dig Point Planning

Automated earthmoving operations such as leveling a mound of soil are distinguished from typical planning problems in two important ways. First, soil is diffuse and therefore a unique description of the world requires a very large number of variables. Second, the interaction between the robot and the world is very complex and only approximate models that are also computationally tractable are available. The large state space and complex robot-world interaction imply that only locally optimal planners (i.e. per dig) can be created. In order to deal with the practical issues of excavating large volumes of earth in applications, we have developed a multi-resolution planning and execution scheme. At the highest level is a coarse planning scheme that uses the geometry of the site and the goal configuration of the terrain to plan a sequence of "dig-regions." In turn, each dig region is searched for the "best" dig that can be executed in that region. Finally, the selected dig is executed by a force based closed loop control scheme (Rocke, 1994). Treatment of the problem at three levels meets different objectives. The coarse planner ensures even performance over a large number of digs. The refined planner chooses digs that meet geometric constraints (reachability and collisions) and which locally optimize a cost function (e.g. volume, energy, time). At the lowest level is a control scheme that is robust to errors in sensing the geometry of the terrain. Figure 11 shows the process of coarse to fine planning for the excavator.

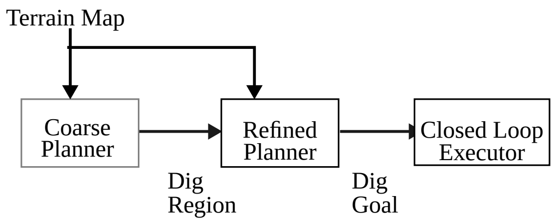

Figure 11. Coarse to fine planning strategy.

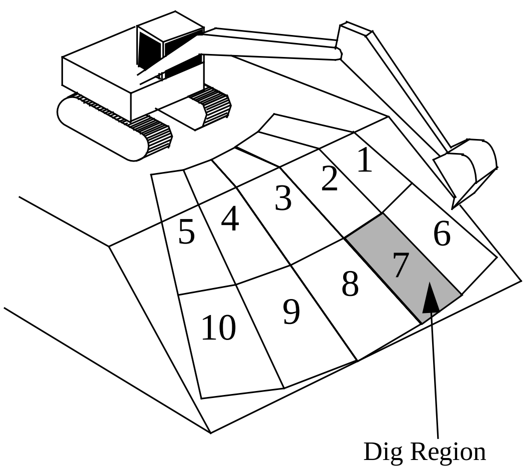

Figure 12. Coarse plan for an excavator.

The coarse planner takes as input processed sensor data which it places in a terrain map (a 2-D grid of height values). The output is a sequence of dig regions, each of which is in turn sent to a refined planner. Figure 12 shows a strategy for removing material that was recommended by an expert excavator operator. Each box indicates a region, and the number within the box indicates the order in which the region is provided to the refined planner. In this strategy, material is removed from left to right, and from the top of the face to the bottom. There are several reasons for choosing this strategy. In most cases, the truck is parked on the operator's left hand side so that the operator has an unobstructed view of it. By digging from left to right, the implements do not need to be raised as high to clear material when swinging to the truck. In digging from top to bottom, less force is required from the implements because it is not necessary then to work against the weight of the material up above. In addition, clearing material away from the top minimizes the range shadows cast on the face of the terrain given a scanning range sensor that is mounted on the cab.

The refined planner operates on an abstract representation of an atomic action (i.e. a single dig). Rather than searching for a bucket trajectory, the refined planner searches through compact task parameters within the bounds specified by the coarse planner. In order to select the best digging action, the refined planner evaluates candidates through the use of a forward model that simulates the result of choosing an action (in our case the starting location of the bucket). An evaluation function scores the trajectory resulting from each action, and the action that meets all constraints and optimizes the cost function is chosen. This process is shown in Figure 13.

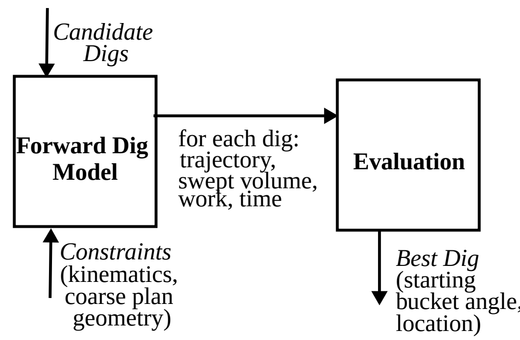

Figure 13. Operation of refined planner.

4.3. *Template Based Dump Planning*

The truck must be loaded evenly and completely. Because of uncertainty in soil settlement, the dumping strategy may need to be revised for each successive bucket load. The dump point planner applies a template-based approach to robustly find the low regions of soil distribution in the truck bed.

Sensor data are gathered after each bucket of soil has been dumped in the truck as the excavator is swinging back to the dig face. Like the dig point planner, the sensor data are placed in a 2-D terrain map. The dump point planner also requires information about the location of the truck, provided by the truck recognizer module, so it can filter out any irrelevant sensor data that are outside of the truck bed. The terrain map is then smoothed using a simple Gaussian filter to eliminate any sensor noise. The current grid cell resolution of the truck bed terrain map is 15 cm, with a typical map containing on the order of 500 cells.

Occlusion of the deposited soil by the truck bed walls is a serious problem. Rather than assuming that nothing is in the unseen regions of the truck bed, the dump point planner fills in any unknown grid cells with the average elevation of the known grid cells. This results in some slight inaccuracies in the perceived soil distribution at first, but they diminish as more soil is placed in the truck bed.

Finally, a specific terrain shape template is convolved over the entire truck bed terrain map to produce a score for each grid cell. This small 5x5 or 7x7 grid cell template looks for a certain profile of the material in the truck bed, such as a slope or a hole. Simple templates of constant elevations can be used to find the lowest elevation in the truck bed terrain map as well. The convolution operator produces a score which represents how well the template matched the particular region in the truck bed, and the location of the cell with the best score is returned as the desired dump location.

4.4. Script Based Motion Planning

The motion planning software coordinates the motions of the excavator's joints for each loading pass, beginning immediately after digging a bucket of soil and ending when the bucket has returned to the next dig point. The main objectives of the motion planner are to plan motions which place the soil at the desired dump location, avoid all known obstacles in the workspace such as the truck, and execute each loading cycle as quickly as possible.

Joint 1: Swing	Command
1) When digging finishes, wait	$\theta_1 = \mathbf{5^\circ}$
2) If $\theta_2 > \mathbf{14^\circ}$, swing to truck	$\theta_1 = \mathbf{101^\circ}$
3) If $\theta_4 > \mathbf{10^\circ}$, swing to dig	$\theta_1 = \mathbf{0^\circ}$
4) If $\theta_1 = \mathbf{0^\circ}$, stop and execute dig	

Joint 2: Boom	Command
1) When digging finishes, raise	$\theta_2 = \mathbf{18^\circ}$
2) If $\theta_1 < \mathbf{60^\circ}$, lower to dig	$\theta_2 = \mathbf{6^\circ}$

Joint 3: Stick	Command
1) When digging finishes, wait	$\theta_3 = \mathbf{-100^\circ}$
2) If $\theta_1 > \mathbf{31^\circ}$, move to spill point	$\theta_3 = \mathbf{-76^\circ}$
3) If $\theta_4 > \mathbf{-30^\circ}$, move to dump point	$\theta_3 = \mathbf{-92^\circ}$
4) If $\theta_1 < \mathbf{65^\circ}$, move to dig	$\theta_3 = \mathbf{-75^\circ}$

Joint 4: Bucket	Command
1) When digging finishes, curl	$\theta_4 = \mathbf{-90^\circ}$
2) If $\theta_1 > \mathbf{60^\circ}$ and $\theta_3 > \mathbf{-89^\circ}$, open	$\theta_4 = \mathbf{30^\circ}$
3) If $\theta_1 < \mathbf{60^\circ}$, move to dig	$\theta_4 = \mathbf{7^\circ}$

Figure 14. Truck loading script for an excavator.

Because of power constraints and joint coupling effects of the excavator's hydraulic system, as well as the difficulty in accurately modeling the dynamics of such a machine, more traditional optimal trajectory generation schemes do not work well. Instead, recognizing the fact that the excavator's motions are highly repetitive and very similar from loading cycle to loading cycle, and that it operates in a relatively small portion of its total workspace, a script based approach to motion planning was adopted (Rowe and Stentz, 1997). A *script* is a set of rules which define the general motions of the excavator's joints for a certain task, in this case loading trucks. These rules contain a number of variables, known as *script parameters*, which get instantiated on every different loading pass.

The rules of script were designed with the input of an expert human excavator operator and implicitly constrain what the excavator is and is not allowed to do. For example, if it was advised that moving two particular joints simultaneously was a bad idea, then the rules of the script make that motion impossible. The left hand side of the rules are functions of the excavator's state, and the right hand side of the rules are the commands which the planner sends to the excavator's low level joint controllers. Thus, when the left hand side of a particular rule evaluates to true, its corresponding command gets sent to the excavator. The rules get re-evaluated at a fixed rate, 10 Hz for example, during the execution of the excavator's motion.

Figure 14 shows the script rules for the truck loading task. The numbers in boldface are one example set of script parameters, which will be described in more detail below. The θ's are the excavator's state, in this case the angular positions of the joints. The commands are desired angular joint positions. Notice that each joint has its own separate script. Therefore, only one rule per joint may be active at a time.

The script parameters are computed before each loading pass starts using the information about the truck's location and the desired dig and dump points. There are two types of script parameters, those which appear in the left hand side of the script rules and affect which commands are sent by the planner, and the joint commands themselves which appear on the right hand side of the rules.

The command script parameters in the right hand side of the rules are primarily computed by geometric and kinematic means. For example, consider the command of 18° from step 1 of the boom joint's script in Figure 14. That is the boom angle which is required for the excavator's bucket to safely clear the top of the truck, and is a kinematic function of the height and location of the truck relative to the excavator. Similarly, the stick joint commands are computed using knowledge about the radial distance of the truck from the excavator, and the swing joint's commands are found from the desired dig and dump points.

The script parameters in the left hand side of the rules are found through a combination of simple excavator dynamic models and heuristics. These simple dynamic models capture first order effects of the excavator's closed loop behavior when given desired angular position commands. These models provide information about the velocities, accelerations, and command latencies for each joint, which are used to intelligently coordinate the different joint motions, resulting in faster loading times. As an example, consider the case when the excavator has finished digging, and the bucket is raising up out of the ground. The excavator does not need to wait until the bucket has raised to its full clearance height before swinging to the truck. Instead, it can begin swinging at some earlier point as the bucket is still raising, but it must have the knowledge provided by the dynamic models about how much time it will take to swing to the truck and to raise the bucket so it can safely couple the two motions to avoid a collision.

4.5. Obstacle Detection

A major requirement for automated loading is detecting and stopping for people and other obstacles which pose a threat for collision. Obstacle detection software has been developed which uses sensor data to perceive objects in the excavator's workspace, and simple dynamic models to predict where the excavator's linkage will be for a short time in the future as the excavator swings back and forth between the dig face and the truck. The predicted excavator linkage locations are compared to the sensor data, and if there is an intersection, the excavator is immediately commanded to stop. It is crucial that the sensors scan far enough ahead of the excavator's motion, and the prediction is far enough in the future, for the excavator to have enough time and space to come to a complete stop and avoid hitting the obstacle. This look-ahead distance is a function of the swing joint's maximum velocity and was found through experimentation to be between 40˚ to 50˚ in front of the excavator's swing joint.

The prediction of the excavator's location is done using the simplified models of the excavator's closed loop dynamic behavior. Not only is the obstacle detection algorithm predicting what the excavator itself will do, it must also simulate what the motion planner will do using the predicted excavator state. It performs this prediction at the same rate of the script rule base update, 10 Hz for instance. The final result is a list of predicted excavator linkage states for some amount of time. The look-ahead time was found empirically to be between 2 - 3 seconds.

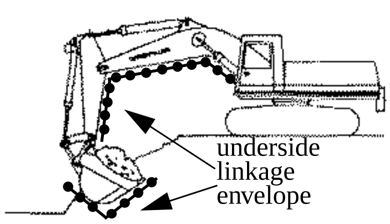

Figure 15. Depiction of the points that are calculated on the underside of the linkage.

For each predicted linkage state, the coordinates of points on the envelope underneath the linkage are computed. This is done using the forward kinematics of the excavator and simple linear models of the shapes of the linkages. This is shown in Figure 15. Each point on the underside of the linkage for each predicted linkage state is then compared to the 2-D elevation map of sensor data. If any point on the underside of the linkage is lower than the elevation of the grid cell that coincides with it, then a predicted collision is reported and the excavator is commanded to stop.

Figure 16. Typical dig for truck loading.

5. Results

Figure 16 shows the excavator after digging a bucket of soil, and Figure 17 shows the truck after it has been loaded with six buckets of soil. To date, we have autonomously loaded our truck hundreds of times. The typical loading times are 15 to 20 seconds per pass, with six passes needed to load the truck. This rate is very close to the loading times logged by an expert operator manually loading trucks in the same configuration using the same excavator.

Figure 17. Truck is loaded after six passes.

6. Conclusion

We have demonstrated an autonomous loading system for excavators which is capable of loading trucks with soft material at the speed of expert human operators. The system uses two scanning laser rangefinders to recognize the truck, measure the soil on the dig face and in the truck, and to detect obstacles in the workspace. The system modifies both its digging and dumping plans based on settlement of soil as detected by its sensors. Expert operator knowledge is encoded into templates called scripts which are adjusted using simple kinematic and dynamics rules to generate very fast machine motions. We believe ours to be the first fully autonomous system to load trucks for mass excavation.

Acknowledgments

This paper summarizes the work of the Autonomous Loading System team. This team consists of Stephannie Behrens, Scott Boehmke, Howard Cannon, Lonnie Devier, Jim Frazier, Tim Hegadorn, Herman Herman, Alonzo Kelly, Murali Krishna, Keith Lay, Chris Leger, Oscar Luengo, Bob McCall, Ryan Miller, Richard Moore, Jorgen Pedersen, Chris Ravotta, Les Rosenberg, Wenfan Shi, and Hitesh Soneji in addition to the authors.

References

Bradley, D.A., Seward, D.W., Mann, J.E., and Goodwin, M.R. 1993. Artificial intelligence in the control and operation of construction plant-the autonomous robot excavator. *Automation in Construction.* 2(3).

Bullock, D.M., and Oppenheim, I.J. 1989. A Laboratory Study of Force-Cognitive Excavation. In Proc. *Sixth International Symposium on Automation and Robotics in Construction*. June.

Bullock, D.M., Apte, S., and Oppenheim, I.J. 1990. Force and Geometry Constraints in Robot Excavation. In Proc. *Space 90: Engineering, Construction and Operations in Space*. ASCE, Albuquerque.

Burks, B.L., Killough, S.M., and Thompson, D.H. 1992. Remote excavation using the telerobotic small emplacement excavator. *1992 Winter Meeting. International Conference on Fifty Years of Controlled Nuclear Chain Reaction: Past, Present and Future*, American Nuclear Society, November. pp. 559-560.

Feng, P., Yang, Y., Qi, Z., and Sun, S. 1992. Research on Control Method of Planning Level for Excavation Robot. In *Proc. 9th International Symposium on Automation and Robotics in Construction*. Tokyo.

Grimson, W.E.L. 1990. *Object Recognition by Computer: The Role of Geometric Constraints*, The MIT Press.

Huang, X.D., and Bernold, L.E. 1994. Control Model for Robotic Backhoe Excavation and Obstacle Handling. In *Proc. Robotics for Challenging Environments*. Albuquerque.

Lever, P., Wang, F., and Chen, D. 1994. Intelligent Excavator Control for a Lunar Mining System. In *Proc. ASCE Conference on Robotics for Challenging Environments*. Albuquerque.

Nease, A.D., and Alexander, E.F. 1993. Air Force construction automation/robotics. In *Proc. 10th International Symposium on Automation and Robotics in Construction (ISARC).* Houston.

Rocke, D.J. 1994. Control system for automatically controlling a work implement of an earthmoving machine to capture material, US Patent 5528843.

Romero-Lois, H., Hendrickson, C., and Oppenheim, I. 1989. A Strategic Planner for Robot Excavation. In *Proc. Sixth International Symposium on Automation and Robotics in Construction*.

Rowe, P., and Stentz, A. 1997. Parameterized Scripts for Motion Planning. In *Proc. International Conference on Intelligent Robots and Systems (IROS)*. Grenoble.

Sakai, T., and Cho, K. 1988. Operation System for Hydraulic Excavator for Deep Trench Works. In *Proc. 5th International Symposium on Robotics in Construction*. Tokyo.

Salcudean, S.E., Tafazoli, S., Lawrence, P.D., and Chau, I. 1997. Impedance control of a teleoperated mini excavator, In *Proc. of the 8th IEEE International Conference on Advanced Robotics* (ICAR), Monterey, California.

Sameshima, M., and Tozawa, S. 1992. Development of Auto Digging Controller for Construction Machine by Fuzzy Logic Control. In *Proc. of Conference Japanese Society of Mechanical Engineers*.

Schwartz Electro-optics Inc., Autosense Scanner, *http://www.seord.com/comsen/autosen.htm*

Seward, D., Bradley, D., Mann, J., and Goodwin M. 1992. Controlling an Intelligent Excavator for Autonomous Digging in Difficult Ground, In *Proc. 9th International Symposium on Automation and Construction*, Tokyo.

Shulz, K. 1997. Imaging Ladar Camera for Washing Robots. In *Proc. of the 14th International Symposium on Automation and Robotics in Construction*, Pittsburgh.

Singh, S. 1995. *Synthesis of Tactical Plans for Robotic Excavation*, Ph.D Thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA.

Singh, S. 1997. State of the Art in Automation of Earthmoving, *ASCE Journal of Aerospace Engineering, Vol 10, Number 4*.

Singh, S. 1998. Multi-Resolution Planning for Earthmoving, to appear in ICRA.

Takahashi, H., Kamata, H., Masuyama, T., and Sarata, S. 1995. Autonomous shoveling of rocks by using image vision system on LHD. In *Proc., International Symposium on Mine Mechanization and Automation*, Golden, Colorado.

Wohlford, W.P., Bode, B.D., and Griswold, F.D. 1990. New capability for remotely controlled excavation. In *Proc. 1990 Winter Meeting of the American Nuclear Society*, American Nuclear Society, pp. 628-629.