Activity Segmentation and 3D-Visualization of Pusher-Loaded Earthmoving Operations from Position Data

Sanghyung Ahn[1], Phillip S. Dunston[2], Amr Kandil[3] and Julio C. Martinez[4]

Abstract: ***By logging position data from GPS-equipped construction machines, we re-create daily activities as 3D animations to analyze performance and facilitate look-ahead scheduling. The 3D animation enables going back to any point in time and space to observe the activities as they took place. By segmenting data into a set of activities, it is possible to obtain actual measures of performance such as cycle times, production, speed profiles and idle times. The measures of performance can then be compared to those expected (e.g., theoretical speed profiles vs. observed profiles), and instances of significant difference can be flagged for further investigation. Idle times and queues that exceed prescribed thresholds can also be identified. In general, many of the traditional real-time performance analyses can be performed after the fact. Situations of interest can be identified automatically and the events in this manner enhances effective performance improvement in construction. The proposed research is explained and demonstrated using a real push-loaded earthmoving operation.***

Keywords: Activity Segmentation, Data Visualization, Earthmoving, Global Positioning System (GPS), Animation

I. INTRODUCTION

Discrete-event simulation can represent the operation of a dynamic and uncertain process, such as construction operations, as a chronological sequence of events in a computer model. Experiments with a simulation model allow users to better understand system behavior, evaluate impacts of various constraints and factors, identify bottlenecks, allocate resource, manage risks, and ultimately improve system performance. Simulation applications can be found in planning and optimizing various types of construction operation, such as high-risk building, earthmoving, tunneling, sewer-line construction, and paving operations [1,2].

However, simulation has been primarily used for long-term planning (e.g., master scheduling or optimizing site layout) during pre-construction stage, where a simulation model is designed to predict long-term, steady state system behavior. The modeling process usually assumes that the target system is stationary and that models will operate under a given set of system design parameters, for example activity precedence relationships and duration distributions [3]. In contract, the goal of look-ahead scheduling is no longer to average out the randomness in the system's behavior but rather to account for and react to system changes on a real-time or near-real-time basis. It requires that a simulation model must be able to capture site condition changes constantly and be updated accordingly so that the changes and their impacts can be evaluated in a timely manner [4]. This also presents several unique challenges when applying simulation as a look-ahead scheduling tool. First, data regarding the most recent project performance are the basis for look-ahead scheduling, yet the time and cost required to manually collect and analyze these data within a short period of time can be prohibitive. Second, since rescheduling is necessary whenever there is a significant change in project status or future outlook, such frequent manual updating can be daunting task in the light of inevitable and constant changes in the project environment along the project timeline. These frequent manual data collection and model updating procedures will inevitably make traditional simulation studies expensive and cumbersome for industry use [4].

Therefore, to make simulation useful for look-ahead scheduling, the current set of simulation modeling methods, such as STROBOSCOPE [5], must be enhanced to a more automated and low-maintenance modeling process. Song and Eldin [4] address that real-time tracking of operation data is the catalyst necessary to transform the current simulation modeling process for look-ahead scheduling. With the recent advancement of tracking sensors (e.g., location, motion, and image sensors), a large set of data can be captured and streamed off construction sites for remote and automated data processing and analysis. These data contain the most current performance data that can be analyzed for operational changes and uncertainties for look-ahead scheduling. This information can also be used to drive a real-time and more automated model updating process, which can significantly reduce engineers' burden in maintaining a simulation model.

Moreover, the collected data can be used to visualize

[1] Research Fellow, School of Civil Engineering, The University of Queensland, Brisbane St Lucia, QLD 4072, Australia, sanghyung.ahn@uq.edu.au (*Corresponding Author)
[2] Professor, School of Civil Engineering, Purdue University, 550 Stadium Mall Drive, West Lafayette, IN 47907, USA, dunston@purdue.edu
[3] Associate Professor, School of Civil Engineering, Purdue University, 550 Stadium Mall Drive, West Lafayette, IN 47907, USA, akandil@purdue.edu
[4] (Posthumously, deceased on June 4th, 2013) Professor, School of Civil Engineering, Purdue University, 550 Stadium Mall Drive, West Lafayette, IN 47907, USA

operations in 3D space as animation. It is necessary to depict the movement, transformations, and interactions between these visualization elements to analyze operations. To depict smooth motion, visual elements must be shown at the right position and orientation several times per second. Many sensors, such as Global Positioning System (GPS), are capable of recording and transmitting data with a frequency of a few minutes of even seconds for accurate visualization of operations.

Thus, this research proposes an adaptive real-time sensor-based analysis and 3D visualization-simulation framework for earthmoving construction applications for the purpose of look-ahead of scheduling. This paper describes this new analysis framework and demonstrates its feasibility through a prototype system.

II. Research Background

The limitation of current simulation approaches and the need for a real-time simulation environment have been documented in several publications [3]. The construction industry has recognized the opportunity for real-time project monitoring and control using the voluminous data streaming off job sites that are made available because of new sensing and communication technologies [6]. However, there is very limited work in integrating real time data collection and simulation for more efficient look-ahead scheduling in the field. For progress monitoring and measuring project, algorithms were developed to utilize real-time project data for measuring project progress, productivity, and actual consumption of materials [7]. The applications of real-time simulation concepts in construction are rather more limited. Chung, Mohamed, and AbouRizk [2] collected project data from a tunneling project manually on a biweekly basis and used the data to improve simulation input models using Bayesian updating techniques. Their study showed that repetitive long-term projects provide opportunities to fine-tune simulation input parameters based on actual project progress. Lu [8] developed a real-time decision-support system for planning concrete plant operations. This system tracks activity durations in real time and later uses the data to update simulation input models.

In summary, the prior studies focused primarily on leveraging real-time data for modeling activity durations. Application of real-time data also brings new opportunities to transform other components of simulation modeling, such as input modeling, model validation and updating. In addition, since simulation-based look-ahead scheduling as a relatively new research area, there is a need to define a systematic approach to transform and integrate various modeling components. The following section defines system components of the proposed adaptive sensor-based analysis and 3D visualization-simulation framework and explains at the conceptual level how it can transform the existing simulation process for look-ahead scheduling.

III. Framework For Sensor-based Analysis And 3d visualization-simulation

The proposed adaptive real-time 3D visualization-simulation framework contains four components, including real-time data acquisition, process knowledgebase, adaptive modeling, and discrete-event simulation services. These system components integrate real-time data and process knowledge to facilitate constant model updating and refinement to reflect changes in the construction field for look-ahead scheduling.

A. Real-Time Data Acquisition

Real-time simulation is a data-driven modeling process. To accurately represent and forecast the system performance, a significant amount of data is required for determining current project status and fine-tuning the model's operation logic structure (e.g., activity definition and sequence) and input models (e.g., activity duration distributions). The proposed data-collection component constantly collects data that can adequately describe current project status and the future outlook of the field operations, such as productivity performance, current project progress, and resource allocation for the near future. These data can be collected from various sources, such as data-collection sensors, information systems and databases, as well as project personnel.

Operation and planning data must be collected constantly to reflect the changes in project performance in the job site environment. However, collecting data in real time can be challenging. The time and cost required to manually collect and process these data are prohibitive. Nevertheless, with the recent advent of sensing and communication technology along with already widely used information systems and database applications, the lack of real-time data is becoming less and less of a factor in the development and implementation of real-time simulation.

A broad range of embedded, wide-area, and satellite-based sensors (e.g., speed, location, motion, and image sensors) are economically available for wireless, automatic, or remote data gathering. For example, equipment-intensive heavy construction projects have been leveraging the Global Positioning System (GPS) to track the real-time locations of construction equipment such as earth-moving machines [7,9] and concrete-hauling trucks. GPS is also used in the prototype system for this research.

B. Process Knowledgebase

Once data are collected and made available from a highly sensed construction site, the next logical step is to analyze and interpret the time-sequenced data for simulation modeling. Many sensors, such as GPS, are capable of recording and transmitting data with a frequency of a few minutes or even seconds. This high-frequency data collection process not only improves data accuracy but also generates a significant amount of data that challenge the ways in which meaningful information can be extracted in a timely fashion. While complete automation of this information extraction process is not realistic, there is an opportunity to streamline and semi-automate the process by incorporating domain knowledge of real-world operation, such as activity sequencing and input modeling, into the real-time simulation framework, so direct user involvement can be minimized.

C. *Adaptive Modeling*

To call something adaptive is to say it has the ability to become more suitable or fit for a specific use or situation. The key feature of the proposed real-time simulation is its capability to adapt a pre-defined simulation model to constant changes of the project environment. This adaptive modeling capability is achieved by updating both the model operation logic and the input models when they are no longer able to accurately represent current and anticipated future project performance. Our prototype system does not address this component yet, but such a capacity is envisioned.

D. *Simulation Services*

Simulation services required by real-time simulation include model verification, validation, simulation execution, and output data collection. Many simulation software tools available offer simulation functions such as graphical modeling, simulation algorithms, and output data analysis, and they allow end users to construct models and conduct simulation experiments to verify and validate the simulation model. This checking process is normally conducted with intensive user intervention. The proposed real-time 3D animation-simulation system provides support for verifying and validating a simulation model through the efficient means of 3D animation.

IV. ACTIVITY SEGMENTATION

The prototype system uses GPS (5Hz) tracking which can record equipment's location, speed, direction, and time stamp. However, the states of equipment are hidden behind the collected observation data. The proposed framework uses Hidden Markov Model (HMM) to extract activity information from GPS observation data. HMM consists of two simultaneous stochastic processes. The first underlying stochastic process constitutes a Markov chain, but unlike with ordinary Markov models, the states cannot be observed. The second stochastic process produces a sequences of observations. Each state has a probability distribution for the observations to appear. Thus, based on the sequence of observations the most probable respective state sequence can be deduced. Due to the stochastic and dualistic nature of HMMs, they are very often used in modeling human actions and performance.

A. *Hidden Markov Models*

An HMM with one dimensional discrete observation probability distributions can be defined as follow. A Hidden Markov Model λ consists of the following elements [10].

1) A set of possible hidden states, $S = \{S_1, S_2, \dots, S_N\}$. The state at time t is denoted as q_t and the state sequence within $1 \leq t \leq T$ as $Q = \{q_1, q_2, \dots, q_T\}$.

2) Observation symbols v_k, where $1 \leq k \leq M$. Observation at t is denoted as o_t and the observation sequence within $1 \leq t \leq T$ as $O = \{o_1, o_2, \dots, o_T\}$.

3) State transition probabilities $A = \{a_{ij}\}$, representing the probability of transition from state S_i to S_j, where $a_{ij} = P(q_{t+1} = S_j | q_t = S_i), 1 \leq i, j \leq N$.

4) Observation probabilities $B = \{b_j(k)\}$, representing the probability of observation symbol v_k in state S_j, where $b_j(k) = P(o_t = v_k | q_t = S_j)$ and $1 \leq j \leq N, 1 \leq k \leq M$.

5) The prior probabilities of states $\pi = \{\pi_i\}$, where $\pi_i = P(q_1 = S_i), 1 \leq i \leq N$.

For convenience of A HMM is usually presented $\lambda = (A, B, \pi)$.

B. *Viterbi-algorithm*

The Viterbi-algorithm [11] is a dynamic programming algorithm for decoding the most likely sequence of hidden states given a sequence of observed events. To be able to find the best matching sequence, an incremental quantity $V_t(i)$ has to be defined (for states $1 \leq i \leq N$):

$$V_t(i) = \vee_i\big(V_t(i) a_{i,j}\big) b_j(k)\,,\ o_{t+1} = v_k \quad (1)$$

The Viterbi algorithm has four steps, initialization, recursion, termination and backtracking of state sequence. At initialization, the values of the quantity $V_t(j)$ and the state sequence back tracking array φ are initialized (for states $1 \leq j \leq N$):

$$V_1(j) = b_j(k)\pi_j,\ \ o_1 = v_k;\ \varphi_1(j) = 0 \quad (2)$$

Then the values are updated recursively using the sequence of observations $o_2, \dots, o_T$, and maximizing the probability of the possible state transitions (from state $1 \leq i \leq N$ to state $1 \leq j \leq N$).

$$V_t(j) = \vee_{i \leq i \leq N}\big(V_{t-1}(i) a_{i,j}\big) b_j(k)\,,\ o_t = v_k \quad (3)$$

$$\varphi_t(j) = arg\, \vee_{i \leq i \leq N}\big(V_{t-1}(i) a_{i,j}\big) \quad (4)$$

V. THE PROTOTYPE SYSTEM

Earthwork operations involve the excavation, transportation and placement or disposal of materials. Successful execution and control of these projects rely on an efficient look-ahead scheduling approach that can capture dynamic project data and incorporate them into the scheduling of upcoming work. The proposed framework was implemented to earthwork operations with three push-loaded tractor scrapers and one pusher as a case study.

A. *Push-loaded Earthmoving Operations*

Tractor-pulled scrapers are designed to load, haul, and dump loose material. The greatest advantage of tractor-scraper combinations is their versatility. The key to a pusher-scraper spread's economy is that both the pusher and the scraper share in the work of obtaining the load. To the extent that they can self-load, they are not dependent on other equipment. If one machine in the spread experiences a temporary breakdown, it will not shut down the job, as would be the case for a machine that is used exclusively for loading. If the loader breaks down, the entire job must stop until repairs can be made. For off-highway situations having hauls of less than a mile, a scraper's ability to both load and haul gives it an advantage. Additionally, the ability of these machines to deposit their loads in layers of uniform thickness facilitates compaction operations.

B. Prototype System Implementation

The prototype system integrates GPS tracking technology with simulation for the purpose of look-ahead scheduling. The pusher-loaded earthmoving operations are associated with the location, speed, or travel direction of a piece of equipment. GPS uses satellites that transmit precise signals to allow a GPS receiver installed in a vehicle to determine its location, speed, travel direction, and the time.

To extract meaningful data for simulation modeling—e,g., scraper loading time and hauling time—3D animation technique is used to assist data integration. The collected GPS data can be reviewed by project personnel to prepare training data for activity segmentation. The user interface of a 3D data review tool in Figure I (a) shows all the data points captured. Additionally, users can re-create operations in 3D animation to identify hidden states of operations for the training data set (Figure I (b); (c)). The list of hidden states is prepared with a STROBOSCOPE activity cycle diagram in Figure II. A set of hidden states is {*Load* (S1), *Haul-Dump-Return-1* (S2), *Stop* (S3), *Haul-Dump-Return-2* (S4), *WaitToLoad & WaitToEnter* (S5), and *EnterToLoad* (S6)}. Haul, dump, and return activities are considered as one single state to simplify the activity segmentation process. By replaying the first cycle of the scrapers in 3D animation, users can manually identify the state of each data point. The prototype system classifies the collected observation data following the rule in Table I. Then, the set of {hidden state #, observation #} generated from the first cycle of scrapers is prepared to use as training data for HMM to extract activity information. Figure III shows an example activity segmentation result from cycle #1 to cycle #5 of scraper #1. The sequence of observations graph plots the prepared observations data based upon Table I. Another stair plot at the bottom of Figure III validates the feasibility of proposed framework to identify the activity information. The sequences of hidden states graph in Figure III shows that there is no error between cycle #1 and cycle #5. In cycle #5, there are some discrepancies between the actual and calculated hidden states. Some of the actual state #2 data points, *Haul-Dump-Return-1*, are identified as state #4, *Haul-Dump-Return-2*, which is not significant. Both state #2 and state #4 can be considered as one single activity when there is no stop point between two activities, which is the case in cycle #5.

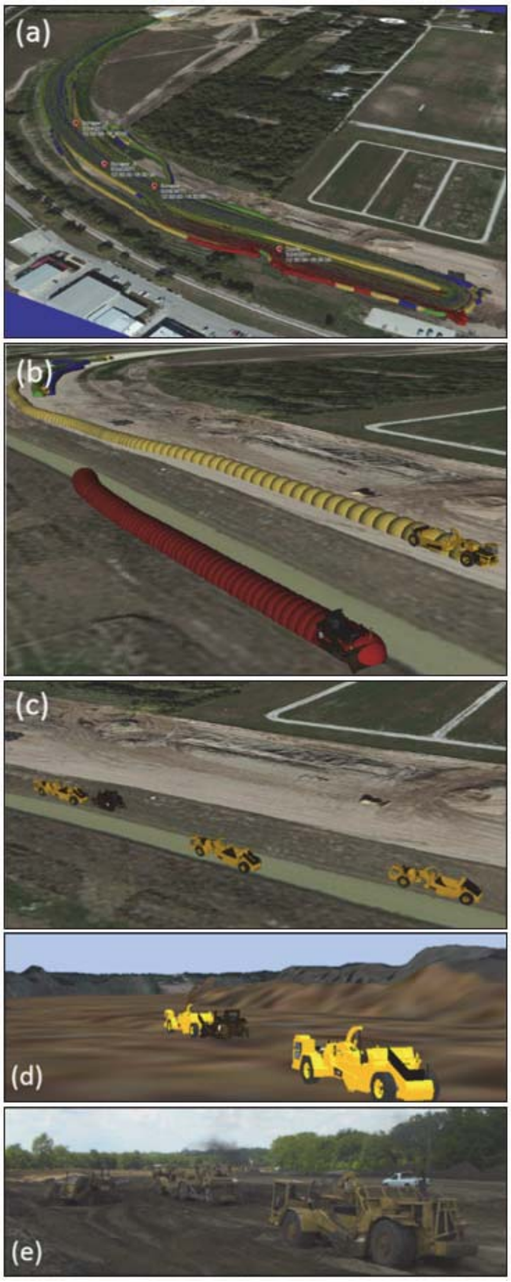
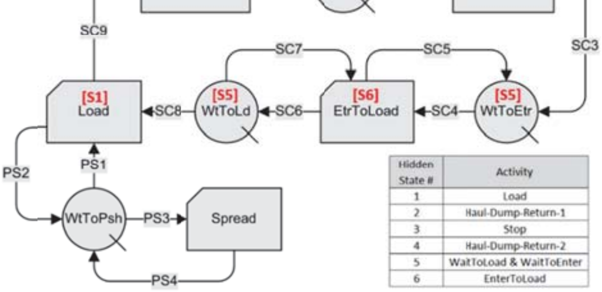


FIGURE I. (a) GPS observation data review tool; (b) re-creation of operations in 3D animation with breadcrumbs; (c) & (d) re-creation of operations in 3D animation without breadcrumb; (e) time synced recording of the same operation



Hidden State #	Activity
1	Load
2	Haul-Dump-Return-1
3	Stop
4	Haul-Dump-Return-2
5	WaitToLoad & WaitToEnter
6	EnterToLoad

FIGURE II. STROBOSCOPE activity cycle diagram and hidden states

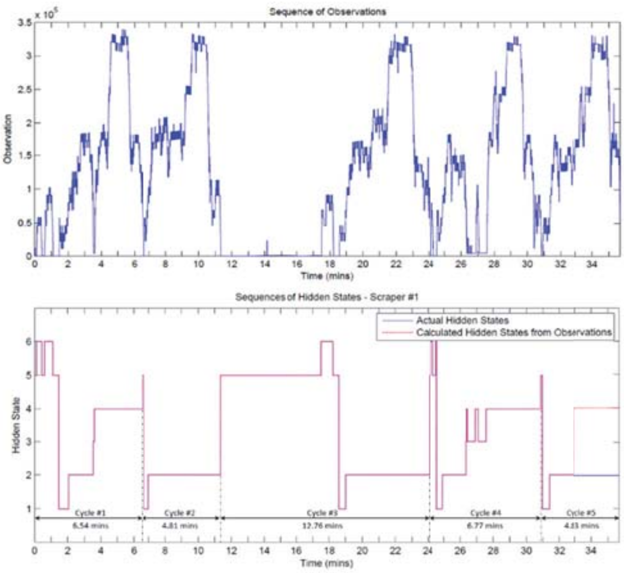


FIGURE III. Observation sequence produced by the prototype and the most likely sequence of hidden states decoded using Viterbi-algorithm

V. CONCLUSION

Real-time simulation challenges the way in which a simulation study is conducted, but at the same time, real-time data give rise to the potential of having a more streamlined and efficient modeling and experiment process for short-term scheduling. As shown in the proposed framework, real-time data inspire the possible automation of the data collection, model updating, verification, and validation processes. When fully implemented, these automated processes can help users focus on the essentials of project scheduling and control instead of on the requirements for the simulation modeling itself. Users can be freed from time-consuming data-collection activities and can direct their efforts toward building process knowledge.

REFERENCES

[1] Halpin, J., and Riggs, L. S. (1992), Planning and Analysis of Construction Operations, John Wiley & Sons, New York.
[2] AbouRizk, S. M. (2010). "Role of simulation in construction engineering and management." Journal of Construction Engineering and Management, 136(10), 1140-1153.
[3] Banks, J. (1998), *Handbook of Simulation,* John Wiley & Sons, New York.
[4] Song, L., and Eldin, N., (2012). "Adaptive Real-time Tracking and Simulation of Heavy Construction Operations for Look-ahead Scheduling." *Automation in Construction,* 27, 32–39.
[5] Martinez, J. (1996). "State and Resource Based Simulation of Construction Processes." Doctoral Dissertation, U. of Michigan, Ann Arbor, MI, USA.
[6] FIATECH. (2011). Capital Project Technology Road Mapping Initiative, FIATECH, Austin, Texas.
[7] Navon, R., Shpatnitsky, Y. (2005). "A model for automated monitoring of road condition." *Construction Management and Economics*, 23(9), 941-951.
[8] Lu, M., Dai, F., and Chen, W., (2007). "Real-time decision support for planning concrete plant operations enabled by integrating vehicle tracking technology, simulation, and optimization algorithm." *Canadian Journal of Civil Engineering,* 34, 912–922.
[9] Pradhananga, N., and Teizer, J. (2013). "Automatic spatio-temporal analysis of construction site equipment operations using GPS data." *Automation in Construction,* 29, 107–122.
[10] Rabiner, L. R., and Juang, B. H. (1986). "An introduction to hidden Markov models." *IEEE ASSP Magazine,* 29(1), 4-16.
[11] Viterbi, A. (1967). "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm." *IEEE Transactions on Information Theory,* 13(2), 260-269.

TABLE I. GPS observation data classification for HMM

Scraper Speed	Distance	Pusher Speed	Observation # (k)
$0 \le$ Speed < 1	Distance between Scraper & Pusher $\le$ (Scraper Length + Pusher Length)	$0 \le$ Speed < 1	1
		$1 \le$ Speed < 2	2
		⋮	⋮
		max(P)-1 $\le$ Speed $<$ max(P)	max(P)+1
	Distance between Scraper & Pusher $>$ (Scraper Length + Pusher Length)	$0 \le$ Speed < 1	max(P)+2
		$1 \le$ Speed < 2	max(P)+3
		⋮	⋮
		max(P)-1 $\le$ Speed $<$ max(P)	⋮
$1 \le$ Speed < 2	Distance between Scraper & Pusher $\le$ (Scraper Length + Pusher Length)	$0 \le$ Speed < 1	⋮
		$1 \le$ Speed < 2	⋮
		⋮	⋮
		max(P)-1 $\le$ Speed $<$ max(P)	⋮
	Distance between Scraper & Pusher $>$ (Scraper Length + Pusher Length)	$0 \le$ Speed < 1	⋮
		$1 \le$ Speed < 2	⋮
		⋮	⋮
		max(P)-1 $\le$ Speed $<$ max(P)	⋮
⋮	⋮	⋮	⋮
max(S)-1 $\le$ Speed $<$ max(S)	Distance between Scraper & Pusher $\le$ (Scraper Length + Pusher Length)	$0 \le$ Speed < 1	⋮
		$1 \le$ Speed < 2	⋮
		⋮	⋮
		max(P)-1 $\le$ Speed $<$ max(P)	⋮
	Distance between Scraper & Pusher $>$ (Scraper Length + Pusher Length)	$0 \le$ Speed < 1	⋮
		$1 \le$ Speed < 2	⋮
		⋮	⋮
		max(P)-1 $\le$ Speed $<$ max(P)	M

*Note: max(P) = (integer)maximum pusher speed
max(S) = (integer)maximum scraper speed
number of discrete observations = M ($1 \le k \le M$)