Journal of Computing in Civil Engineering. Submitted September 19, 2011; accepted July 23, 2012;
posted ahead of print November 17, 2012. doi:10.1061/(ASCE)CP.1943-5487.0000242

Vision-Based Tower Crane Tracking for Understanding Construction Activity

Yang, J.[1] and Vela, P.A. [2] and Teizer, J. [3] and Shi, Z.K. [4]

ABSTRACT

Visual monitoring of construction work sites through the installation of surveillance cameras has become prevalent in the construction industry. These cameras also have practical utility for automatic observation of construction events and activities. This paper demonstrates the use of a surveillance camera for assessing tower crane activities during the course of a work day. In particular, it seeks to demonstrate that the crane jib trajectory together with known information regarding the site plans provides sufficient information to infer the activity states of the crane. The jib angle trajectory is tracked using 2D-3D rigid pose tracking algorithms. The site plan information includes a process model for the activities and site layout information. A probabilistic graph model for crane activity is designed to process the track signals and recognize crane activity as belonging to one of the two categories: concrete pouring and non-concrete material movement. Experimental results from a construction surveillance camera show that crane activities are correctly identified.

Keywords: Construction, Tower Crane, Visual Tracking, Activity Understanding

INTRODUCTION

Many construction projects involve the coordination and cooperation of multiple entities in order to achieve the final as-built structure. Failure to properly coordinate leads to losses, which can be ultimately measured in terms of cost, productivity, and possibly safety. Before

[1]Northwestern Polytechnical University, Xi'an, China 710072. E-mail: junyang@nwpu.edu.cn
[2]School of Electrical and Computer Enginnering, Georgia Tech, Atlanta, GA 30332. E-mail: pvela@gatech.edu
[3]School of Civil and Environmental Engineering, Georgia Tech, Atlanta, GA 30332. E-mail: teizer@gatech.edu
[4]Northwestern Polytechnical University, Xi'an, China 710072. E-mail: zkeshi@nwpu.edu.cn

Journal of Computing in Civil Engineering. Submitted September 19, 2011; accepted July 23, 2012;
posted ahead of print November 17, 2012. doi:10.1061/(ASCE)CP.1943-5487.0000242

coordination and cooperation of the construction site entities can be analyzed, tracking and activity analysis of the individual entities is required (Gong and Caldas 2009). The goal of this investigation is to understand tower crane operations in order to eventually connect them to construction progress on the work site, and also to ultimately understand the coordination of activities related to the tower crane. The tower crane is chosen as the track entity because of its prominent and consistent visual geometry, and its important role in construction projects. In order to advance the goal, algorithms to track both the crane jib and to connect the motion to activities are needed. Output of the automated algorithms provides information regarding the activities supported by the tower crane and their execution over time. Longer term, these activities could be connected to progress on the physical as-built structure.

The activity information sought is most closely aligned with the field of activity or productivity analysis (CII 2010). Such analysis is performed through a variety of techniques such as work sampling, direct observation techniques, checklists, and survey/interview based methods (Oglesby et al. 1989; McCullouch 1997; Thomas 1991). These assessments types are typically performed by qualified humans and can be inherently error-prone or subjective. Ultimately, the intent is to identify the activities being performed at the measurement time instants in order to build statistics regarding activity levels. The activity level is defined as the percentage of time craft workers spend on a particular activity. Productivity is measured as the direct work time rate. Computing these statistics requires the collection of activity and timing data regarding the various activities of interest. The efforts involved in manually collecting the measurement data means that continuous monitoring is infeasible. New information and sensing technologies provide a manner to generate a steady, reliable, and consistently interpreted data stream of construction processes. Videotaping is one such technology, whose potential and benefits for collecting data on job-sites has been extensively studied (Abeid and Arditi 2002; Bolivar and Mee 1997). Unfortunately, recorded video often requires manual review, which is an inefficient process. Automating the conversion of video

Journal of Computing in Civil Engineering. Submitted September 19, 2011; accepted July 23, 2012;
posted ahead of print November 17, 2012. doi:10.1061/(ASCE)CP.1943-5487.0000242

into activity data requires advanced pattern recognition and computer vision techniques tailored to the construction context (Navon and Sacks 2007).

Within the general construction context, research conducted to address automated sensing challenges includes the following. Digital color image processing algorithms have been designed to detect the idle time of a hydraulic excavator in a specific site environment (Zou and Kim 2007). Wireless cameras have been used to measure productivity for a bridge replacement project (Kim and Bai 2007). In addition, automated vision tracking techniques have been studied to provide spatial information of construction resources for specific construction operations. Research involving visual tracking on work-sites includes the tracking of personnel, of excavator end effectors, and of concrete buckets (Brilakis et al. 2011; Gong and Caldas 2009; Makhmalbaf et al. 2010; Teizer and Vela 2009; Yang et al. 2010). In (Gong and Caldas 2009), a machine learning algorithm was trained to detect the crane bucket in video footage. By analyzing the bucket location in the image using prior knowledge of the scene, they measured the productivity of a concrete pouring project. Follow-up work in (Gong and Caldas 2011) demonstrated activity decomposition results for other construction activities. Outside of vision, other sensor-based technologies show potential applications for assisting automated productivity measurement on material installations (Cheng et al. 2011; Grau et al. 2009; Grau and Caldas 2007) and construction process tracking (Pradhan and Akinci 2007; Pradhan et al. 2011).

With regards to cranes, past research has focused on improving productivity and safety. Luo et al. (Luo et al. 2011) examine the sensing requirements associated to crane safety monitoring. Everett and Slocum (Everett and Slocum 1993) introduced a video system called CRANIUM to transmit a real time picture of the loads to the operator for improved communication. Shapira et al (Shapira et al. 2008) designed a tower-crane-mounted live video system to enhance the visibility of the operator for both daytime and nighttime operation. Other researchers (Abdelhamid and Everett 1999; Ju et al. 2005; Tantisevi et al. 2008; Zavichi and Behzadan 2011) focus on optimizing the location of tower crane and materials

Journal of Computing in Civil Engineering. Submitted September 19, 2011; accepted July 23, 2012;
posted ahead of print November 17, 2012. doi:10.1061/(ASCE)CP.1943-5487.0000242

in the construction site, optimizing the crane task ordering, or improving the crane control to restrain the sway and swing.

The current objective of this paper is to demonstrate that a monocular surveillance camera capable of monitoring a tower crane and its operational environment can be used to automatically identify the activities associated to the tower crane. Activity level reports follow directly from the timing statistics of the activity information. Rather than focus on detecting a specific load carried by the tower crane as in (Gong and Caldas 2009), this investigation focuses on the connections between the crane jib signal and both the site layout and the construction process model. Visual tracking of the crane jib is chosen over load detection since a tower crane transports a variety of objects, including, but not limited to, concrete buckets, equipment, beams, slabs, columns, etc. Furthermore, given the setup of a general purpose surveillance camera, c.f. Figure 1(a), the crane bucket may not always be visible or may be in poor visibility. Lastly, while tracking via vision is a focus, the proposed activity inference approach allows for trajectory information from other means to be used in lieu of vision. In this last case, visual detection of the load category may not be available, yet the proposed approach would still apply as it relies on other available information.

This paper demonstrates that trajectory information of the crane jib in conjunction with a process model and a site layout plan is sufficient to infer the construction activities associated to a tower crane. The two main steps are the tracking of the crane and estimation of the activities. A model-based visual pose estimation algorithm will track the jib angle versus time (Section 3). The activities of the tracked tower crane will be inferred using the Viterbi algorithm on a probabilistic graph model determined from the crane process model and site layout (Section 4). Actual recorded image sequences of a tower crane over several hours are converted into activity reports and compared to manually determined ground truth (Section 5).

PROPOSED TECHNIQUE

The proposed activity estimation and reporting technique consists of two steps. The first

Journal of Computing in Civil Engineering. Submitted September 19, 2011; accepted July 23, 2012;
posted ahead of print November 17, 2012. doi:10.1061/(ASCE)CP.1943-5487.0000242

step estimates the crane jib rotation angle for each frame by converting a given surveillance image into a simplified image through a signal processing pipeline, then comparing the image to rendered images generated from a tower crane model. The best fit rendered image determines the rotation angle. From the trajectory information, the second step uses the Viterbi algorithm to infer the activity states from a probabilistic instantiation of the crane process model. This section discusses the overall methodology, while the subsequent sections detail further the approach.

The experimental data was obtained from a surveillance camera mounted on the Georgia Tech campus that monitors the construction of a building. The camera views the construction site from the roof of a nearby building. The perspective allows for the structure and it surrounding area to be seen. Access to the roof is possible, allowing for measurement of the crane and the camera parameters for modeling purposes. A robotic total station (RTS) was used to measure 3D points necessary for camera calibration (Hartley and Zisserman 2003) and tower measurement. The video sequences were captured at three seconds per frame.

The known information, aside from the modeling data above, includes knowledge regarding the site layout and its planned use. Inferring the crane activities requires knowledge of the functions associated to different regions of the site space. Many construction sites have site layout plans that describe the intended use of the construction work space, which includes a plan view description of the extents of the as-built structure, expected roadways, laydown yards, and permissible crane flyby zones. Figure 1(a) shows the building construction site divided into three major zones: driveway, parking lot, and working zone. The site's logistics plan, Figure 1(b), indicates that the concrete mixer is allocated two spots along the driveway, where it serves the crane. The coverage of the crane jib is a disc centered at the tower mast. Empirically, the crane loads materials from the storage area to unload in the work zone. Ideally, if the storage area is organized with sub-areas for distinct material types, the material type lifted by the crane could be inferred by area. Based on our observations of the available surveillance video, that is not strictly the case. However, out of all the materials

Journal of Computing in Civil Engineering. Submitted September 19, 2011; accepted July 23, 2012;
posted ahead of print November 17, 2012. doi:10.1061/(ASCE)CP.1943-5487.0000242

lifted, the concrete bucket is unique. The concrete mixers are located at specific spots of the work site, thus distinguishing the concrete pouring activity from other materials lifting tasks.

Automated estimation of the jib rotation angle will be performed through the use of a 3D crane model in combination with the camera calibration parameters. From the model and the camera parameters, 3D pose estimation techniques will provide the jib rotation angle versus time. 3D pose estimation algorithms first generate renderings of the 3D crane as perceived by a camera with known parameters. These rendered images are then compared to the actual observed image. The estimated geometry is correct when the rendered image and the captured image are in agreement. At this point, automated activity analysis is performed by converting the continuous trajectory information into discrete activity observations over time. The activity observations provide coarse observations over time such as stopped, moving, in region of interest, etc., which are then analyzed to infer the true activity states over time. Analysis is performed using the Viterbi algorithm given a probabilistic graph model for the activities associated to a process model.

The automated process is summarized as follows:

1. Camera calibration and tower crane model generation.
2. Conversion of process model to probabilistic graph model.
3. Translation of site layout information to computational rules.
4. Image processing to generate a binary representation.
    - Sky extraction through thresholding.
    - Skyline removal through background modeling.
5. Crane jib angle estimation through rendering.
    - Local search for crane jib angle by optimizing a pose-fitting energy.
    - Kalman filtering to clean noise and estimate velocity.

Journal of Computing in Civil Engineering. Submitted September 19, 2011; accepted July 23, 2012;
posted ahead of print November 17, 2012. doi:10.1061/(ASCE)CP.1943-5487.0000242

6. Crane trajectory converstion to crane state observations.
7. Crane activity state sequence inference from crane state observations.
8. Automated computation of activity statistics.

The following section describes the algorithms further.

CRANE JIB TRACKING

Tracking of the crane jib using vision will be performed through a generative model, and will provide the jib angle signal versus time. The generative model simulates an expected rendering of the sensed image given the crane angle. The rendered image is not a true-life rendering of the work-site as seen in Figure 1(a), but is instead a rough approximation that depicts only the crane model. The simplified rendering is compared to a processed version of the actual sensed image, as per Figure 2. The angle θ_k^* that maximizes agreement between the two is the current crane jib angle.

A crude crane model was determined by surveying the installed crane using a robotic total station (RTS). The final model is depicted in Figure 3. Using the known camera calibration configuration, a rendering of the crane jib generates a predicted image of the scene given a specified jib rotation angle. Figure 4(d) depicts several such simulated renderings given distinct jib rotation angles. The renderings only depict the crane jib, and exclude both the tower and the tower mast. These binary images must be compared to the actual captured camera image. Given that the image is far richer than the simulated rendering, pre-processing algorithms convert the captured image into a binary image. The process for doing so incorporates three steps, all of which are described in the following paragraphs: (a) cropping of the image to consist of the crane jib operational region, (b) a sky elimination step which identifies the sky regions, and (c) a background subtraction step which isolates the crane arm from other static elements of the image.

The first and simplest step isolates the image region that the crane arm could realistically occupy. The region is identified by rotating the model through the full 360° arc and noting

Journal of Computing in Civil Engineering. Submitted September 19, 2011; accepted July 23, 2012;
posted ahead of print November 17, 2012. doi:10.1061/(ASCE)CP.1943-5487.0000242

the crane jib image regions of all of the projections. In this case, that corresponds roughly to the top portion of the image (100 lines of 480 lines). This step is depicted in Figure 4(a). The second step converts the image to grayscale and applies Otsu's thresholding algorithm (Otsu 1979) to the image in order to isolate and remove the sky regions of the image. Results for various sky conditions are depicted in Figure 4(b). The remaining visual elements in the image $I_{skyline}$ are the tower crane and buildings.

The buildings and other such permanent visual elements form the skyline. These elements can be identified and removed using a background model. The algorithm utilized is the single Gaussian background modeling algorithm (Wren et al. 1997), which models the background as an image whose intensities obey a Gaussian distribution. Associated to each pixel are mean and covariance values. The expected image is generated by the mean pixel values. When a new image is captured and has the sky regions removed, it is then compared against the mean Gaussian model. Pixels are considered to be outliers if they have low likelihood of belonging to the Gaussian model, e.g., if they lie too many standard deviations away from the mean. Here, the threshold was two standard deviations. The third row, Figure 4(c) depicts the classified statistical outliers to the Gaussian image model. What mostly remains in the output image I_{crane} is the crane jib.

The rendered binary image is compared to the processed surveillance camera image to see how well the two images match. Let $I_{model}(\theta)$ represent the silhouette generated from the 3D model with jib angle θ, and let $I_{crane}(t_k)$ represent the current processed camera image. The overlap energy between these two binary images is defined as:

$$E(\theta) = \sum_p I_{model}(\theta; p) \cdot I_{crane}(t_k; p),$$

which sums over all image pixels, p. When the estimated jib angle agrees with the actual jib angle, the two silhouette images are in agreement and the energy is maximal. Searching through all possible rotation angles can be exhaustive given the need to render the images.

Journal of Computing in Civil Engineering. Submitted September 19, 2011; accepted July 23, 2012;
posted ahead of print November 17, 2012. doi:10.1061/(ASCE)CP.1943-5487.0000242

Since the target application is tracking, we can assume that the angle from the previous frame is known and the current angle is sought. As the crane has a finite angular rate of change, the set of angles reachable from one frame to the next is limited. Thus a window-based search is applied to find the angle that maximizes the matching energy $E(\theta)$,

$$\theta^{\star} = \arg\max_{\theta \in [\theta^{-} - r_{\theta}, \theta^{-} + r_{\theta}]} E(\theta),$$

where θ^{-} is the jib angle from the previous frame and r_{θ} is the search radius. The angular search range $[\theta - r_{\theta}, \theta + r_{\theta}]$ is discretely approximated with the constant step size $\Delta\theta$ (here, $\Delta\theta = 1°$). The angle measurement is then used in a constant velocity Kalman filter to estimate the velocity and to smooth out any noise in the measurements. By choosing a second order model, with acceleration as the uncertainty, the Kalman filter will estimate velocity from position measurements.

ACTIVITY INFERENCE

Once the jib angle trajectory is known, the activities of the crane can be inferred. As a lifting machine which moves materials around the construction site, a tower crane's activity can be clearly defined as loading, lifting and unloading materials. A static crane occurs during loading, unloading, or transitions between the two, while a moving crane corresponds to lifting. In the scenarios being considered, the primary activity of the crane is to assist with concrete pours, with material hoist cycles occuring intermittently. Crane activity will be categorized into concrete pouring and non-concrete pouring. Since the material being lifted is not actively classified, the site layout plans will be exploited to infer the tower crane's activities. Based on the site plan 1(b), a mapping between the crane jib rotation angle and activity zones can be extracted to obtain Figure 5(a). When overlaid with the site plan, the work zone corresponds to where the structure will be, and the mixing zones correspond to where the concrete mixers will be.

Based on the activity categories and the crane action modes (lifting, loading, unloading),

Journal of Computing in Civil Engineering. Submitted September 19, 2011; accepted July 23, 2012;
posted ahead of print November 17, 2012. doi:10.1061/(ASCE)CP.1943-5487.0000242

a natural model for describing the crane activity is the process model which is a graph-based model for describing a process. Following the description of (Gong and Caldas 2009), the process model breaks down the overall concrete pour process into a collection of activity states, which define the graph nodes, with acceptable transitions between the states defining the directed graph edges. Transitions between activity nodes happen when the proper observable condition is met, that is typically associated with initiating the activity state. As shown in Figure 5(b), the process model for concrete pouring has five states which normally happen in a fixed order: concrete loading at the mixer, moving from the mixer to work zone, unloading concrete at the work zone, moving from the work zone back to the mixer, and detaching the concrete bucket at the mixing zone. An additional two states capture the non-concrete pouring activities. Transitions between states are determined by motion of the crane jib. To be robust to the measurement noise, two thresholds are defined. One is for the instant angular speed, the other is for the state transition. When the instant angular speed exceeds the speed threshold, the crane jib is considered to be moving. To generate a transition, the transition event has to be continuously detected for a specific amount of time (the time threshold). The thresholds were obtained by taking a training video with crane movements and manually identifying proper values (they were 2°/s and 3 frames respectively).

The activity observations are the events that trigger transitions between activity states, which are essentially the stopped and moving observations. Furthermore, stopping within a region with a known functionality triggers a separate observation than the generic stop observation. Figure 6 depicts the output observations given the trajectory information. The blue continuous curve (whose axis is to the left) depicts the actual crane jib angle trajectory over time, while the green discrete state curve illustrates the activity observations (and whose axis is to the right).

The true activity states of the system over time must be inferred from the activity observations. The Viterbi algorithm provides an optimal estimate of the activity states

Journal of Computing in Civil Engineering. Submitted September 19, 2011; accepted July 23, 2012;
posted ahead of print November 17, 2012. doi:10.1061/(ASCE)CP.1943-5487.0000242

over time given a probabilistic model of the activity observations and of the transitions between activity states. Thus, the process model from Figure 5 must be augmented to have transition probabilities associated with each directed edge, plus self-transition probabilities. Furthermore, additional edges are required indicating the link between the states and the observations. Figure 7(a) depicts the graph of the observation probabilities conditioned on the activity states, denoted by $P(x \,|\, y)$ where x is the activity state variable and y is the output observation variable. Figure 7(b) depicts the graph of current activity state probabilities given the previous activity state, denoted by $P(x_i \,|\, x_{i-1})$ where the subscripts denote the observation time. In both cases, missing links imply zero probability of occurance. Both of the probabilities are used to define the cost of selecting the current activity state given the current observation and the previous activity state, which the Viterbi algorithm uses to find the optimal sequence of state transitions given the measured observations.

Based on the defined probabilities, the optimization problem is defined:

$$\{x_1^*, \cdots, x_N^*\} = \underset{x_1,\cdots,x_N}{\arg\max} \sum_{i=1}^{N} \left(-\log P(x_i \,|\, y_i) - \log P(x_i \,|\, x_{i-1})\right), \tag{1}$$

which returns the optimal selection of activity states in time $\{x_1, \cdots, x_N\}$ given N total activity observations $\{y_1, \cdots, y_N\}$ and a known initial state x_0. Where out-of-sequence events may occur, it is common to assign a low probability of occurence in order to robustly manage them. For example, since the threshold for stopped versus moving may not always be accurate for the given scenario, there is a low probability of measuring a stop condition in accurately, thus the small probability connecting *Return Bucket* to the *Stop/Slow* observation in Figure 7(a). Assigning non-zero probabilities allows the inference to contemplate multiple options in order to pick the most likely. Because of the way that the Viterbi algorithm processes a probabilistic graph model, multiple options are considered and non-unique paths can be resolved. For example, once the crane has completed a pour cycle, the operator can choose to service other workers or can choose to continue servicing the concrete pouring team.

Journal of Computing in Civil Engineering. Submitted September 19, 2011; accepted July 23, 2012;
posted ahead of print November 17, 2012. doi:10.1061/(ASCE)CP.1943-5487.0000242

Given the observations over time, the Viterbi algorithm correctly identifies the action taken by the crane operator at a particular instance of time when the best sequence of activities to represnt the observations can be established. Figure 8 depicts an example case of inference from a jib angle track signal and its observation signal. Based on the optimization equation (1), the sequence of activity states in Figure 8(b) was inferred to be generated from the sequence of activity observations in Figure 8(a). The accuracy of the inferencing algorithm is discussed in the next section.

EXPERIMENTS

This section covers the resulting output associated to the proposed methodology for videos captured during crane operation. In particular, two image sequences generate activity timing reports for observed concrete pouring cycles, while an additional sequences is applied to another cyclic material delivery task. Sample frames from each sequence type are depicted in Figure 9. The video sequences were also manually processed in order to validate that the automated activity timing results were consistent with a human annotator. All results are discussed.

Concrete Pour Cycles. The two concrete pour video sequences contain material hoist activities that are not concrete pour cycles. In the first sequence, concrete pouring is interrupted then resumed. The algorithm correctly infers the proper begin and end points of each cycle as can be seen in the activity versus time plot of Figure 10(a). The concrete pour timing chart is given in Table 1. The table consolidates the return movement and the bucket drop as simply one consolidated time labeled *Return*. There are two sets of cycles C1 to C3 and C4 to C8. Of note, cycle C1 is the longest. Inspection of the video shows that the concrete mixer servicing the bucket left the work-site and a new mixer began to service the concrete pour cycles. The remaining cycles were serviced by the same concrete source and show more uniform timing. The second concrete pouring sequence contained only three complete pour cycles, followed by several material hoist tasks. The activity versus time

Journal of Computing in Civil Engineering. Submitted September 19, 2011; accepted July 23, 2012;
posted ahead of print November 17, 2012. doi:10.1061/(ASCE)CP.1943-5487.0000242

plot is given in Figure 10(b), while the activity timing table is found in Table 2. Outside of the outlier pour cycle from the first sequence, the times spent on the concrete pouring activities within each cycle are consistent, which is expected when the crane is operated by an experienced operator. Time spent on concrete loading is shorter than time on concrete pouring, and time of moving the bucket to the working zone is longer than time of moving the bucket back to the mixer.

The manual annotations of the crane activity state matches well that of the algorithm for the concrete pour cycles, but not so well for the other material hoist tasks. The difference lies in the approach for going from the video to the activity label. When viewing the video frame-by-frame, there can be some uncertainty as to the crane movement, plus the human may also apply different thresholds for deciding when there is motion than the algorithm. For example, the crane hook may move while the crane jib remains static, which is then followed by a crane jib motion. Human annotators tend to trigger motion when the crane hook begins moving, which means that the human tends to have higher movement times versus stopped times (as can be seen in many of the Move versus Pour numbers). The automated algorithm generates consistent observations as to when the crane is in motion or not. For example, in reference to Figure 6, the first sequence has a point where the crane slowly drifts a few degrees (between time 900 and 1000). The algorithm does not consider the drift to be separate hoist activity transitions, while the manual annotator does as can be seen for the same time period in Figure 10(a). A review of the video shows that the algorithm interpretation is sensible as there was not sufficient activity to trigger activity state changes. In general, much of the minor variation in activity transitions between the automated and human annotations are a result of differences in identifying when the crane is moving versus when it is stopped. Large errors are often due to special events in worker activity that cause incorrect inferences. Overall, however, a review of the numbers shows that the automated annotation is usually within 10% of the human annotation for a cycle (with averages of 296 versus 279 seconds for Sequence 1 and 229 versus 211 seconds for Sequence 2).

Journal of Computing in Civil Engineering. Submitted September 19, 2011; accepted July 23, 2012;
posted ahead of print November 17, 2012. doi:10.1061/(ASCE)CP.1943-5487.0000242

Material Hoist Cycles. The third video consists of a material delivery event that was combined with material hoist cycles to install the material at its installation location. A truck with a large flatbed, c.f. Figure 9(b), was unloaded via crane leading to many material hoist cycles. Borrowing from the same probabilistic graph model, however with adjustments to the site plan to incorporate a flatbed loading zone, and also to the cycle transitions, activity inference is performed as for the concrete pours. The final results can be found in Figure 11. Note that initially there is a material installation, followed by attention to another group of workers, followed by a return to the material resources to install which continues for the duration of the day. Statistics of the cycle timings can be found in Table 3. While the majority of the cycle estimates are the same as for the manual annotation, there is one obvious difference occuring roughly between frames 1600-2500 (cycle C12). At that point, the crane operator returned the hook to the loading area, which was followed by a pause in the materials delivery as a new flatbed is brought in and prepared for additional material delivery. The inferencing algorithm cannot distinguish total inactivity from loading activity when the hook is in the material zone, thus the algorithm estimates the crane hook as being in the loading state, whereas a human can clearly see that there is no direct loading activity going on. There is indirect activity as the new materials are delivered and unstrapped. These cases can be remedied by tracking the hook and also detecting the existence of personnel or material resources, much as in (Gong and Caldas 2009). In this case, the algorithm identified the outlier cycle based on the prevailing statistics. If the outlier load time for C12 is removed, then the average load time drops to 264 seconds and the average cycle time drops to 470 seconds, both of which are closer to the human annotated times (within 13% and 3.5% respectively).

Implementation Discussion. In order to run the algorithm several parameters were necessary, many of which were manually obtained from a single pass through a video sequence not related to the ones processed. By comparing the results visually against the video, the

Journal of Computing in Civil Engineering. Submitted September 19, 2011; accepted July 23, 2012;
posted ahead of print November 17, 2012. doi:10.1061/(ASCE)CP.1943-5487.0000242

velocity and timing parameters could be readily obtained. The Kalman filter parameters are obtained from the measurement fluctuations associated to a static jib. Lastly, the probabilities in the graph models were likewise obtained by running the algorithm through a separaet video sequence. While machine learning may be used as in (Gong and Caldas 2009), the amount of time needed to generate a training dataset is far larger than the few minute spent adjusting the parameters. Furthermore, the probabilities associated with the graph model generate some robustness to these parameters. The process model is quite similar from site to site, thus the model and its probabilities can be recycled with slight adjsutments per construction site. The concrete pour process and many material hoisting processes are very similar in terms of the process model graph structure, thus we anticipate that a general probabilistic graph model can be identified and loaded to perform the activty inferencing. Lastly, we believe that the tower crane models will eventually part of a public domain library of construction machines that can be loaded automatically and registered automatically using the 2D-3D pose estimation algorithm, thus freeing the practitioner from manaully measuring the structure geometry.

CONCLUSION

This paper illustrated the use of computer vision and discrete state inference algorithms for construction project analysis. A visual tracking algorithm for the tower crane coupled with a finite state machine for activity states enabled construction activity understanding. Tower crane activity was categorized into concrete pour cycles and non-concrete pouring movement, with automated work sampling timing statistics of the concrete pour cycles. Extension of the work for material hoist cycles was also demonstrated. Experimental results show that the visual tracking algorithm is able to track the tower crane, while a probabilistic instantiation of the finite-state machine with observations of the transitions distinguishes the crane activities. Errors in the activity timing are within 10-15%. Weaknesses of the algorithm revolved around poor inferencing due to unmodeled events. These errors can be remedied through additional visual processing.

Journal of Computing in Civil Engineering. Submitted September 19, 2011; accepted July 23, 2012;
posted ahead of print November 17, 2012. doi:10.1061/(ASCE)CP.1943-5487.0000242

Future work seeks to consider additional activities common to the construction process. Furthermore, the vision-based jib tracking can be improved. In this vein, future work seeks to actively detect, track, and classify the load to more accurately assess the activity state.

REFERENCES

Abdelhamid, T. and Everett, J. (1999). "Time series analysis for construction productivity experiments." Journal of Construction Engineering and Management, 125, 87–95.

Abeid, J. and Arditi, D. (2002). "Linking time-lapse digital photography and dynamic scheduling of construction operations." Journal of Computing in Civil Engineering, 16, 269–279.

Bolivar, A. and Mee, A. S. (1997). "Activity analysis using computer-processed time lapse video." 1405–1411.

Brilakis, I., Park, M. W., and G., J. (2011). "Automated vision tracking of project related entities." Advanced Engineering Informatics, In Press.

Cheng, T., Venugopal, M., Teizer, J., and P.A., V. (2011). "Performance evaluation of ultra wideband technology for construction resource location tracking in harsh environments." Automation in Construction, In Press.

CII (2010). "Guide to activity analysis." Report No. Implementation Resource 252-2a.

Everett, J. and Slocum, A. (1993). "Cranium: device for improving crane productivity and safety." Journal of Construction Engineering and Management, 119(1), 23–39.

Gong, J. and Caldas, C. (2011). "Learning and classifying motions of construction workers and equipment using bag of video feature words and bayesian learning methods." ASCE International Workshope on Computing in Civil Engineering. 274–281.

Gong, J. and Caldas, C. H. (2009). "A computer vision based video interpretation model for automated productivity analysis of construction operations." Journal of Computing in Civil Engineering, 24(3), 252–263.

Grau, D. and Caldas, C. (2007). "A framework for automated localization of on-site construction components." ASCE/CIP Construction Research Congress, Reston, VA.

Journal of Computing in Civil Engineering. Submitted September 19, 2011; accepted July 23, 2012;
posted ahead of print November 17, 2012. doi:10.1061/(ASCE)CP.1943-5487.0000242

Grau, T., Caldas, C., Haas, C., Goodrum, P., and Gong, J. (2009). "Assessing the impact of automated materials tracking technologies on construction craft productivity." Automation in Construction, 18, 903–911.

Hartley, R. and Zisserman, A. (2003). Multiple view geometry in computer vision. Cambridge Univ Pr.

Ju, F., Choo, Y., et al. (2005). "Dynamic analysis of tower cranes." Journal of Engineering Mechanics, 131, 88.

Kim, S. H. and Bai, Y. (2007). "Development of a real-time productivity measurement system for bridge replacement." Proc. of Mid-Continet Transpiration Symposium, Ames, Iowa.

Luo, X., Leite, F., and O'Brien, W. (2011). "Requirements for autonomous crane safety monitoring." ASCE International Workshop on Computing in Civil Engineering, Miami, FL. 331–338.

Makhmalbaf, A., Park, M., Yang, J., Brilakis, I., and Vela, P. (2010). "2D vision tracking methods' performance comparison for 3D tracking of construction resources." Construction Research Congress 2010: Innovation for Reshaping Construction Practice: Proceedings of the 2010 Construction Research Congress, May 8-10, 2010, Banff, Alberta, Canada. 459–469.

McCullouch, B. (1997). "Automating field data collection in construction organizations." Proc. 5th Construction Congress: Managing Engineered Construction in Expanding Global Markets, Reston, VA. ASCE, 957–963.

Navon, R. and Sacks, R. (2007). "Assessing research issues in automated project performance control (APPC)." 16(4), 474–484.

Oglesby, C., Parker, H. W., and Howell, G. (1989). Productivity Improvement in Construction. McGraw-Hill Book Company, New York, NY.

Otsu, N. (1979). "A threshold selection method from graylevel histograms." IEEE Trans. Syst., Madn. & Cybern., 9(1), 62–66.

Journal of Computing in Civil Engineering. Submitted September 19, 2011; accepted July 23, 2012;
posted ahead of print November 17, 2012. doi:10.1061/(ASCE)CP.1943-5487.0000242

Pradhan, A. and Akinci, B. (2007). "A planning based approach for fusing data from multiple sources for productivity monitoring." ASCE/CIB Construction Research Congress, Reston, VA.

Pradhan, A., Akinci, B., and Haas, C. (2011). "Formalisms for query capture and data source identification to support data fusion for construction productivity monitoring." Automation in Construction, 20, 389–398.

Shapira, A., Rosenfeld, Y., and Mizrahi, I. (2008). "Vision system for tower cranes." Journal of Construction Engineering and Management, 134, 320–332.

Tantisevi, K., Akinci, B., et al. (2008). "Simulation-based identification of possible locations for mobile cranes on construction sites." Journal of Computing in Civil Engineering, 22, 21–30.

Teizer, J. and Vela, P. (2009). "Personnel tracking on construction sites using video cameras." Advanced Engineering Informatics, 23(4), 452–462.

Thomas, H. (1991). "Labor productivity and work sampling: the bottom line." Journal of Construction Engineering and Management, 117(3).

Wren, C., Azarbayejani, A., Darrell, T., and Pentland, A. (1997). "Pfinder: Real-time tracking of the human body." IEEE Transactions on Pattern Analysis and Machine Intelligence, 19(7), 780–785.

Yang, J., Arif, O., Vela, P. A., Teizer, J., and Shi, Z. (2010). "Tracking multiple workers on construction sites using video cameras." Advanced Engineering Informatics, 24(4), 428–434.

Zavichi, A. and Behzadan, A. (2011). "A real time decision support system for enhanced crane operations in construction and manufacturing." ASCE International Workshop on Computing in Civil Engineering, Miami, FL. 586–593.

Zou, J. and Kim, H. (2007). "Using hue, saturation, and value color space for hydraulic excavator idle time analysis." Journal of Computing in Civil Engineering, 21, 238–246.

Journal of Computing in Civil Engineering. Submitted September 19, 2011; accepted July 23, 2012;
posted ahead of print November 17, 2012. doi:10.1061/(ASCE)CP.1943-5487.0000242

List of Tables

Journal of Computing in Civil Engineering. Submitted September 19, 2011; accepted July 23, 2012;
posted ahead of print November 17, 2012. doi:10.1061/(ASCE)CP.1943-5487.0000242

TABLE 1. Automated (left) and manual (right) tabulation of concrete pour procedure for Sequence 1 (units are seconds).

No.		Load	Move	Pour	Return	Cycle		Load	Move	Pour	Return	Cycle
C1		111	36	267	411	825		93	75	228	342	738
C2		54	48	78	36	216		54	60	60	57	231
C3		60	51	78	99	288		39	63	63	78	243
C4		63	42	78	39	222		48	48	63	57	216
C5		42	45	57	45	189		36	48	42	57	183
C6		42	57	57	48	204		42	54	57	48	201
C7		45	48	84	39	216		45	60	69	57	231
C8		45	42	42	75	204		30	51	30	78	189
X	Acc. Time	462	369	741	792	2364		387	459	612	774	2232
X	Avg. Time	58	46	93	99	296		48	57	77	97	279

Journal of Computing in Civil Engineering. Submitted September 19, 2011; accepted July 23, 2012;
posted ahead of print November 17, 2012. doi:10.1061/(ASCE)CP.1943-5487.0000242

TABLE 2. Automated (left) and manual (right) tabulation of concrete pour procedure for Sequence 2 (units are seconds).

No.		Load	Move	Pour	Return	Cycle		Load	Move	Pour	Return	Cycle
C1		48	42	81	39	210		45	54	57	51	207
C2		36	42	39	36	153		36	48	27	51	162
C3		54	51	123	96	324		39	78	96	51	264
X	Acc. Time	138	135	243	171	687		120	180	180	153	633
X	Avg. Time	46	45	81	57	229		40	60	60	51	211

Journal of Computing in Civil Engineering. Submitted September 19, 2011; accepted July 23, 2012;
posted ahead of print November 17, 2012. doi:10.1061/(ASCE)CP.1943-5487.0000242

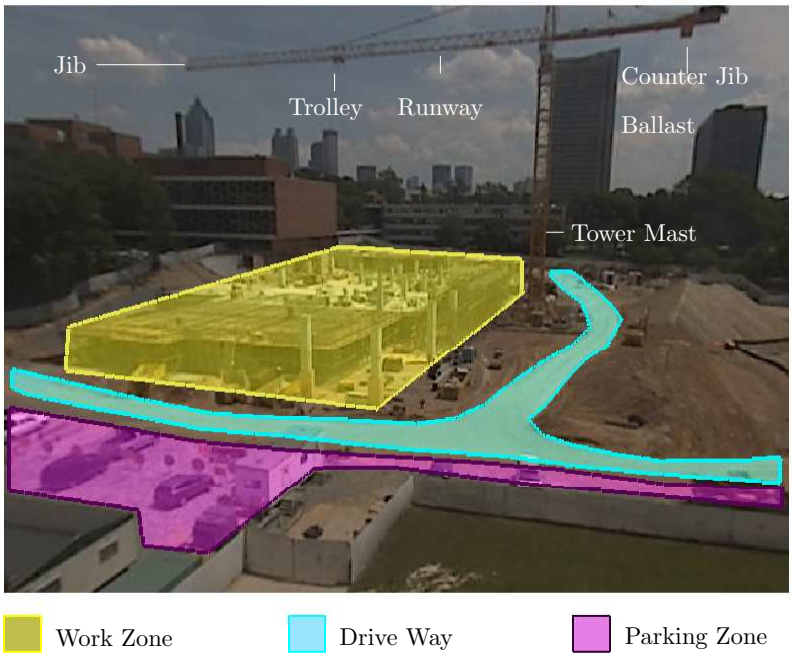
TABLE 3. Automated (left) and manual (right) tabulation of material delivery cycles for Sequence 3 (units are seconds). Outliers are with strike-out, while bold times are averages with the outliers removed.

No.		Load	Move	Deliver	Return	Cycle		Load	Move	Deliver	Return	Cycle
C1		168	39	81	27	315		165	57	66	33	321
C2		228	48	69	33	378		213	66	57	48	384
C3		222	42	105	42	411		189	69	99	24	381
C4		69	39	93	30	231		81	45	90	33	249
C5		162	36	132	27	357		111	96	120	30	357
C6		186	24	93	45	348		168	48	90	30	336
C7		141	60	45	30	276		141	51	63	30	285
C8		141	36	93	51	321		141	51	81	48	321
C9		465	81	123	57	726		333	66	123	45	567
C10		306	114	78	39	537		258	138	117	39	552
C11		240	48	141	45	474		219	75	135	36	465
C12		~~2754~~	48	150	42	~~2994~~		132	75	120	63	390
C13		210	87	87	39	423		207	54	126	45	432
C14		249	45	144	57	495		216	78	141	51	486
C15		390	111	279	36	816		378	114	273	57	822
C16		792	72	81	0	945		786	63	90	0	939
X	Acc. Time	6723	930	1794	600	10047		3738	1146	1791	612	7287
X	Avg. Time	**264**	58	112	38	**470**		234	72	112	38	455

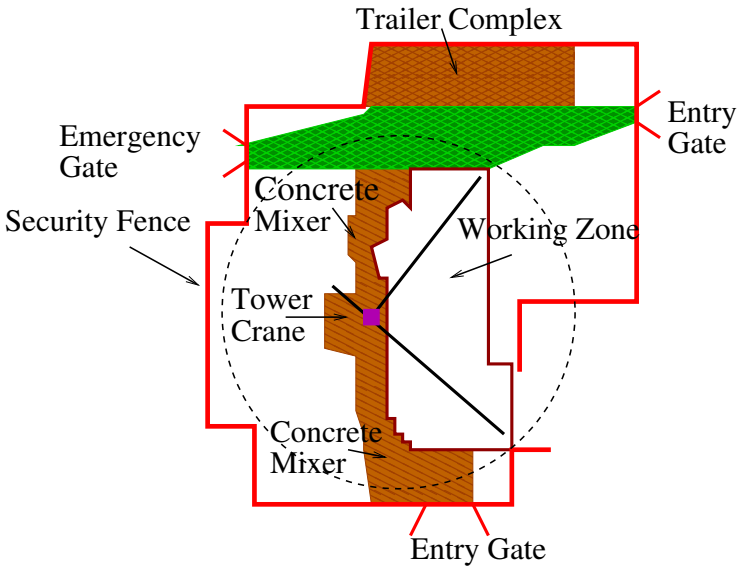
Journal of Computing in Civil Engineering. Submitted September 19, 2011; accepted July 23, 2012;
posted ahead of print November 17, 2012. doi:10.1061/(ASCE)CP.1943-5487.0000242

List of Figures

Journal of Computing in Civil Engineering. Submitted September 19, 2011; accepted July 23, 2012;
posted ahead of print November 17, 2012. doi:10.1061/(ASCE)CP.1943-5487.0000242



(a) Imaged crane structure and site layout.



(b) Site logistics plans.

FIG. 1. Depictions of surveillance image and site plan.

Journal of Computing in Civil Engineering. Submitted September 19, 2011; accepted July 23, 2012;
posted ahead of print November 17, 2012. doi:10.1061/(ASCE)CP.1943-5487.0000242

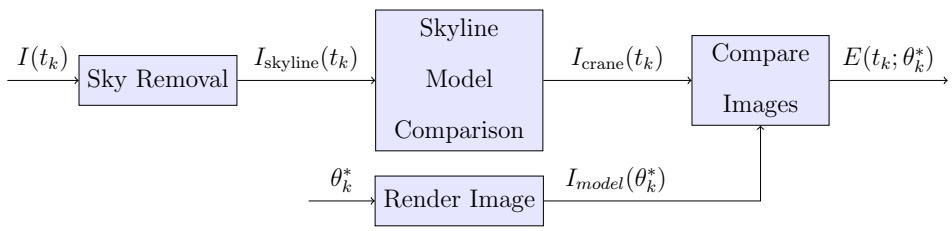


FIG. 2. Comparison pipeline to identify crane rotation angle.

Journal of Computing in Civil Engineering. Submitted September 19, 2011; accepted July 23, 2012;
posted ahead of print November 17, 2012. doi:10.1061/(ASCE)CP.1943-5487.0000242

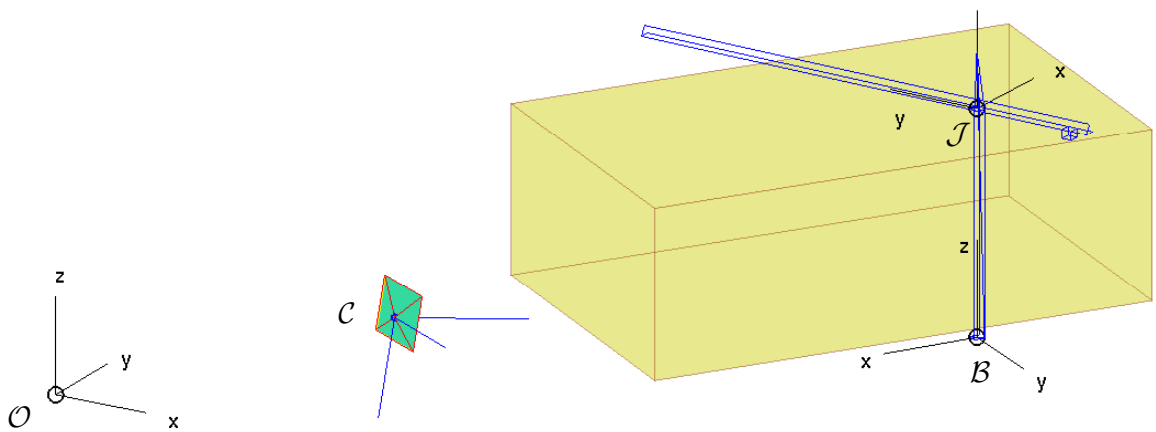
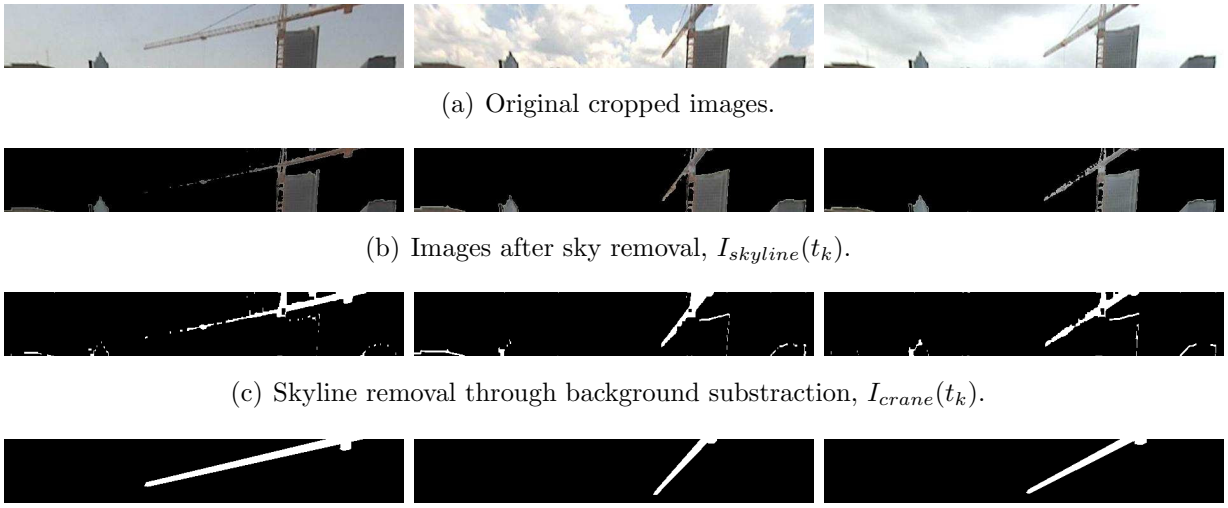


FIG. 3. Depiction of the crane wireframe model and the surveillance configuration.

Journal of Computing in Civil Engineering. Submitted September 19, 2011; accepted July 23, 2012;
posted ahead of print November 17, 2012. doi:10.1061/(ASCE)CP.1943-5487.0000242



(a) Original cropped images.

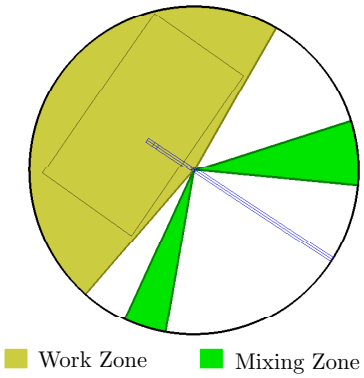
(b) Images after sky removal, $I_{skyline}(t_k)$.

(c) Skyline removal through background substraction, $I_{crane}(t_k)$.

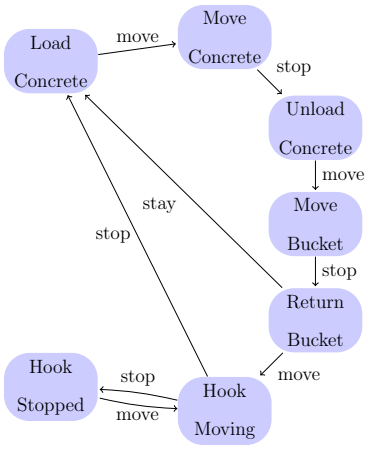
(d) Rendered/simulated version after angle optimization, $I_{render}(\theta_k^*)$.

FIG. 4. Output images of the processing pipeline for three sample images

Journal of Computing in Civil Engineering. Submitted September 19, 2011; accepted July 23, 2012;
posted ahead of print November 17, 2012. doi:10.1061/(ASCE)CP.1943-5487.0000242



(a) Plan view of crane activity zones.



(b) Crane activity process model

FIG. 5. Understanding crane activities by incorporating information regarding site logistics plans.

Journal of Computing in Civil Engineering. Submitted September 19, 2011; accepted July 23, 2012;
posted ahead of print November 17, 2012. doi:10.1061/(ASCE)CP.1943-5487.0000242

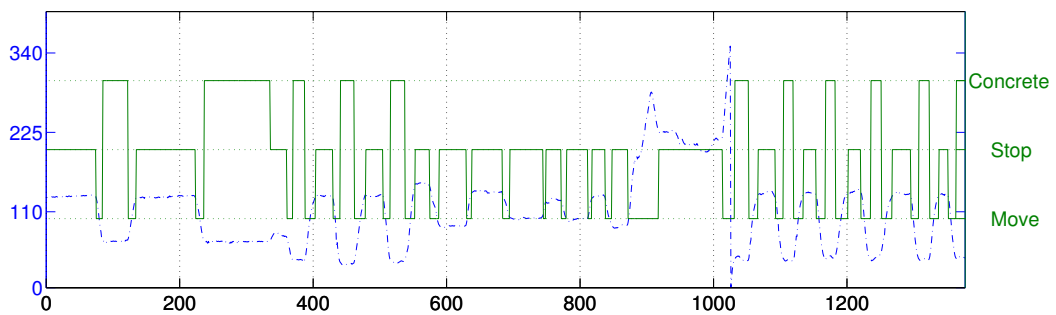
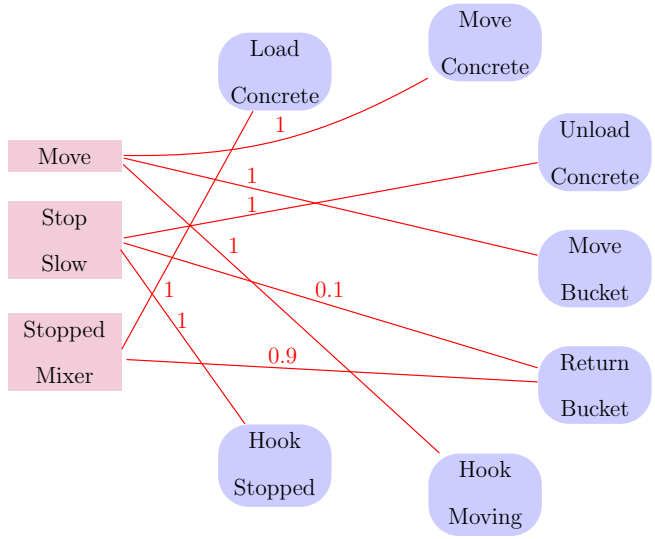
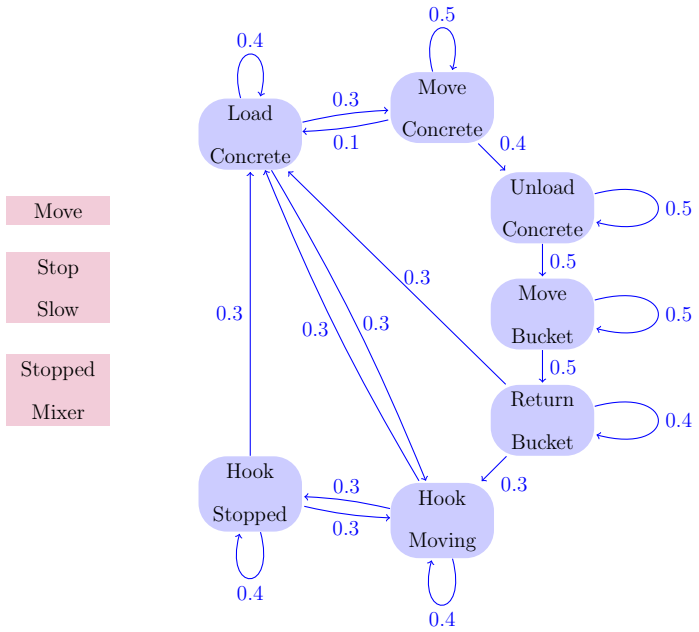


FIG. 6. Crane angle vs time (blue, dash-dotted, left axis) and crane state observation vs. time (green, solid, right axis)

Journal of Computing in Civil Engineering. Submitted September 19, 2011; accepted July 23, 2012;
posted ahead of print November 17, 2012. doi:10.1061/(ASCE)CP.1943-5487.0000242



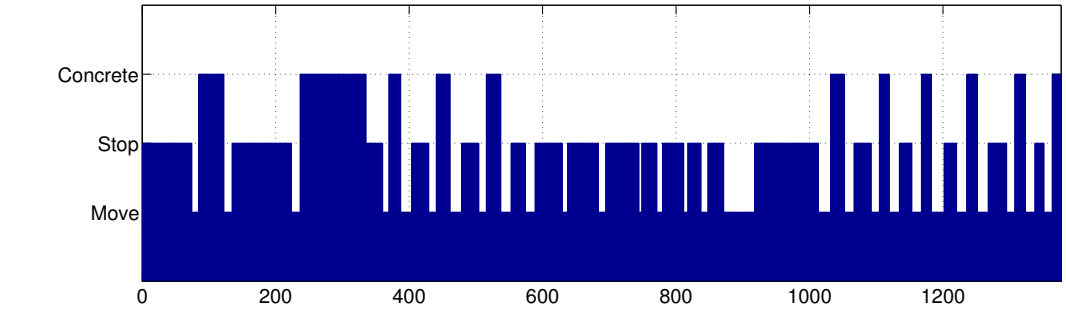
(a) Observation probabilities.



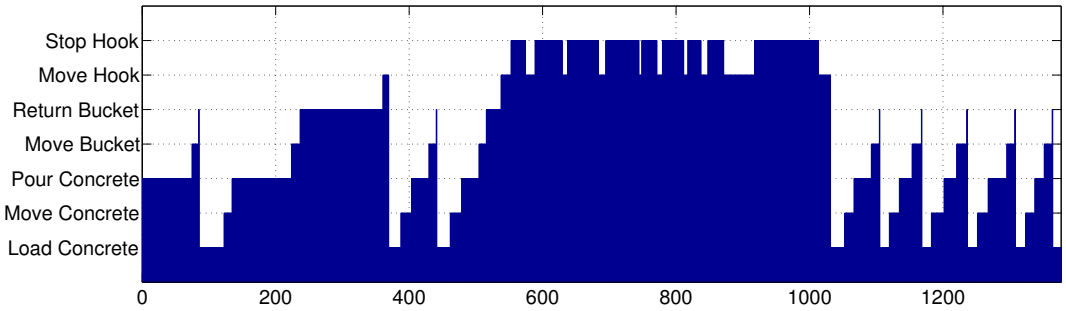
(b) Transition probabilities

FIG. 7. Observation and transition probability graphs.

Journal of Computing in Civil Engineering. Submitted September 19, 2011; accepted July 23, 2012;
posted ahead of print November 17, 2012. doi:10.1061/(ASCE)CP.1943-5487.0000242



(a) Activity observations vs. time

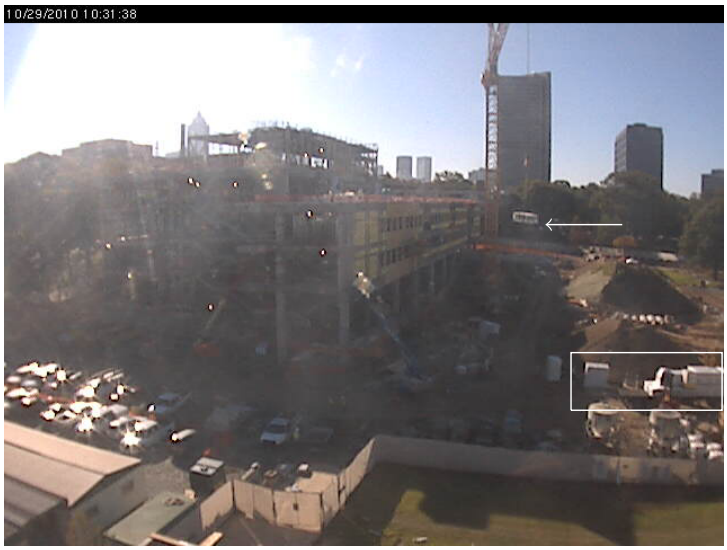


(b) Activity states vs. time

FIG. 8. Viterbi algorithm takes activity observations over time and infers the associated activity states versus time.

Journal of Computing in Civil Engineering. Submitted September 19, 2011; accepted July 23, 2012;
posted ahead of print November 17, 2012. doi:10.1061/(ASCE)CP.1943-5487.0000242

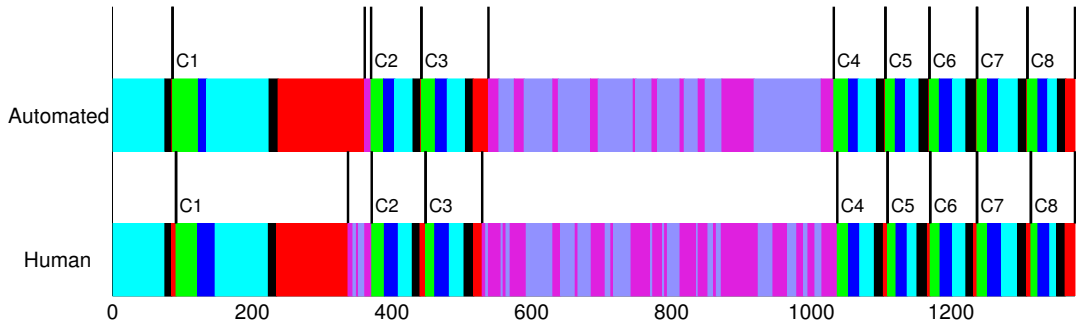
(a) Concrete pouring sequence.



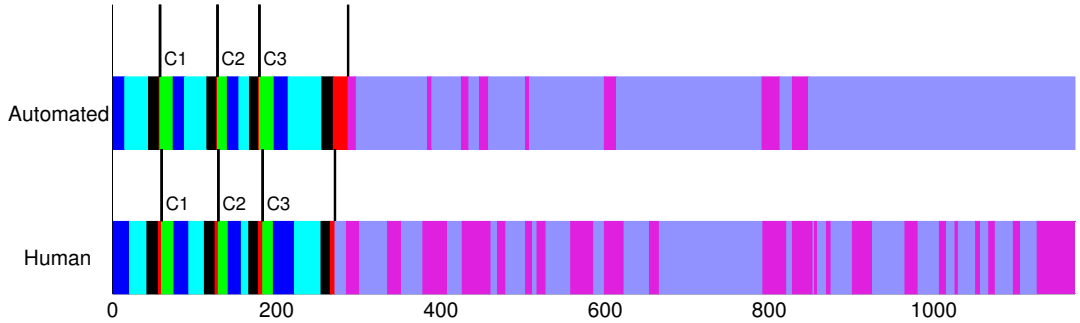
(b) Material hoisting sequence.

FIG. 9. Sample images from each experiment type, with demarcated principal loading areas (white box) and the load being hoisted (arrow).

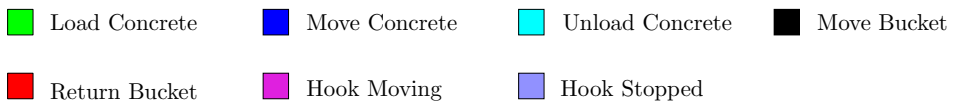
Journal of Computing in Civil Engineering. Submitted September 19, 2011; accepted July 23, 2012;
posted ahead of print November 17, 2012. doi:10.1061/(ASCE)CP.1943-5487.0000242



(a) Crane activity states for sequence 1



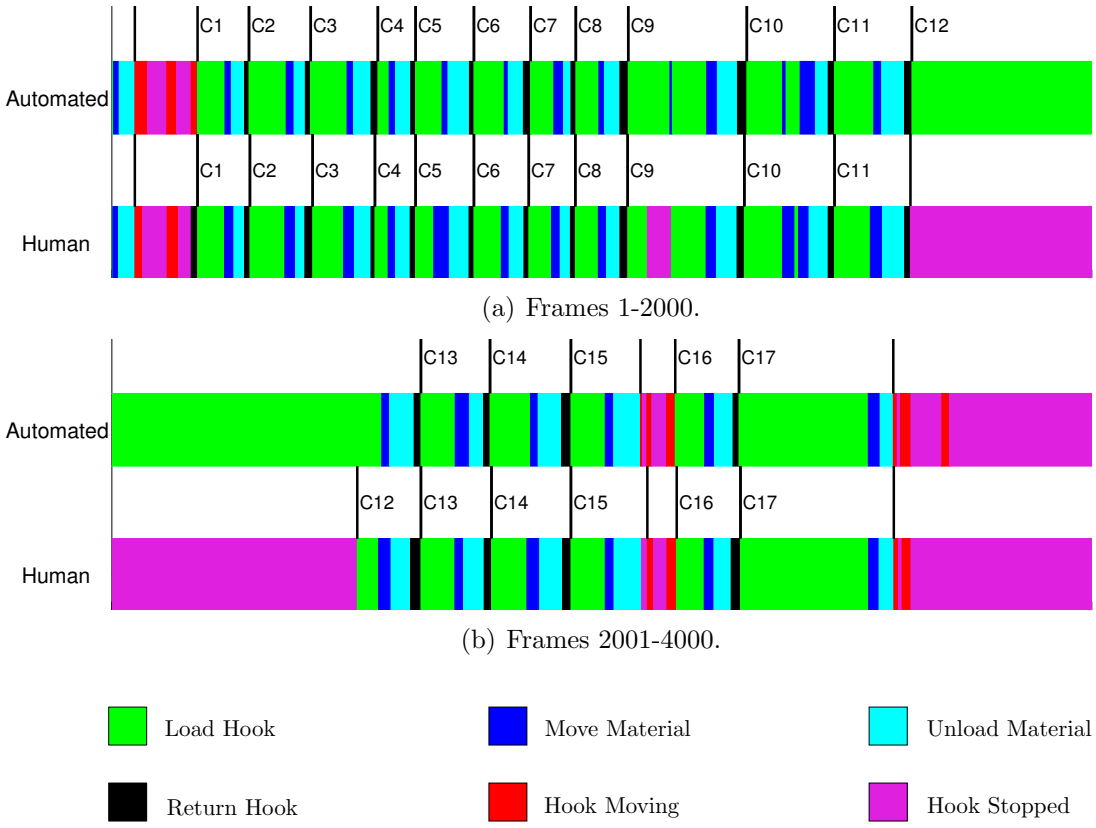
(b) Crane activity states for sequence 2



(c) Legend

FIG. 10. Activity state vs time with top row being the computed activity states and the bottom being the ground-truth.

Journal of Computing in Civil Engineering. Submitted September 19, 2011; accepted July 23, 2012;
posted ahead of print November 17, 2012. doi:10.1061/(ASCE)CP.1943-5487.0000242



(a) Frames 1-2000.

(b) Frames 2001-4000.

(c) Legend

FIG. 11. Activity state vs time for material hoist cycles. The top row is the computed activity states and the bottom is the ground-truth.