Adaptive Zoom Control Approach of Load-View Crane Camera for Worker Detection

Tanittha Sutjaritvorakul, Atabak Nejadfard, Axel Vierling and Karsten Berns

Dept of Computer Science, Technische Universität Kaiserslautern, Germany
E-mail: {tanittha,nejadfard,vierling,berns}@cs.uni-kl.de

Abstract -
Zoom camera is essential for detecting objects from the top-view. The deep learning detection algorithm can fail to handle scale invariance, especially for detectors whose input size is changed in an extremely wide range. The adaptive zoom feature can enhance the quality of the deep learning worker detection. In this paper, we introduce an automatic zoom control approach and demonstrate its efficacy in real-world top-view object detection. To avoid further data gathering and extensive re-training, the zoom adaptability method of the load-view crane camera is able to support the deep learning algorithm, specifically in the high scale variant problem. The finite state machine is employed for control strategies to adapt the zoom level to cope not only with inconsistent detection but also abrupt camera movement during lifting operation. As the result, the detector is able to detect a small size object by smooth continuous zoom control without additional training.

Keywords -
Construction safety, Worker detection, Safety monitoring, Visibility assistance, Adaptive zoom control, Automatic zoom adjustment, Zoom tracking

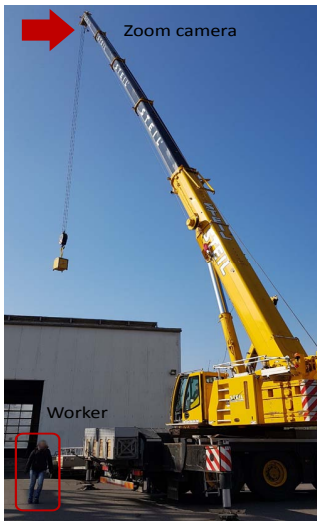

Figure 1. Crane load-view camera where red arrow points to, mounting on pendulum bracket.

1 Introduction

On the construction site, there are a great number of accidents caused by visibility. The vision-based autonomous technology can help workers and remove the human factor to reduce injury and fatality. In autonomous construction tasks, object detection is one of the main components in the perception pipeline. It provides the baseline to allow the robot or machine to further extract semantic information at a higher level such as worker safety monitoring [12] and crane lifting assistance system [4].

The primary problem with top view detection is scale variation. Despite less occlusion compared to the frontal view, the viewpoint of the object is changed and the target size becomes very small which make it more difficult for detectors to recognize, also for the human because of the small size and less information—only a head and a shoulder of a person can be seen from the top view. Furthermore, the size can be changed in different altitudes. Although there are many deep learning object detection studies, the research in detecting objects from top view or aerial images remains limited, particularly in the construction domain. The applications using top view images include surveillance, traffic, inspection, and construction. The image sensor can be installed on Unmanned Aerial Vehicle (UAV), buildings, or large machines such as a mobile crane. To address the problem, there is great effort to augment training small object size dataset for training to yield better accuracy of the data-driven methods [5].

Another possibility to tackle the scale variant issue in object detection is using *adaptive zoom*. Zoom camera is widely used in many applications such as construction site or surveillance. The zoom feature is used to retain the image quality in a wide field of view or provide a close-up view for better recognition. For example, the zoom camera can enhance the load view for the crane operator to observe the safety proximity surrounding the load during the lift operation. The surveillance zoom camera can track the movement of suspects and zoom in to their faces for accurate facial recognition [3]. Nevertheless, it is crucial to control the zoom level automatically because it requires stable zoom control to hold each zoom constraint. In general, it is challenging to adjust the zoom level smoothly, given the noisy sensor data. Moreover, zoom control has rarely been studied directly. The research is mainly focused on adaptive zoom conditions rather than how to implement an effective zoom control. In other words, the

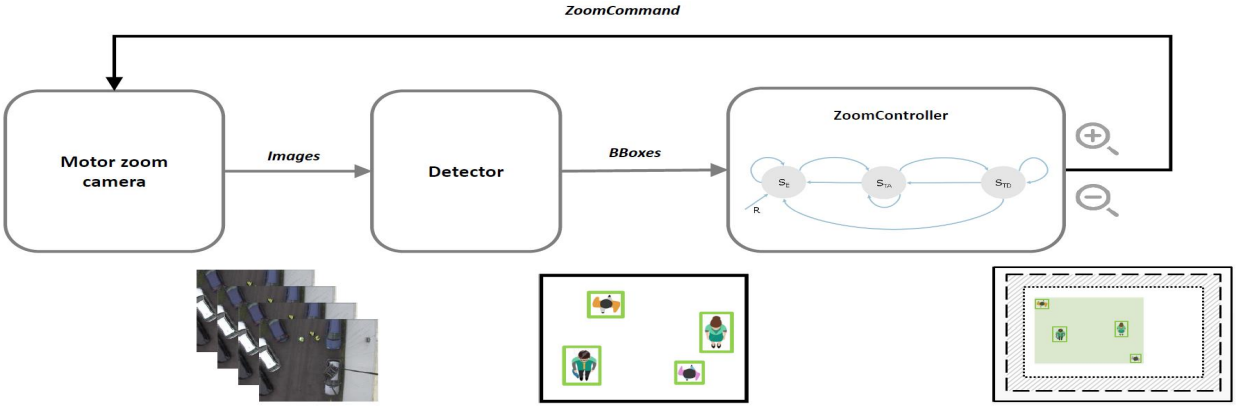

Figure 2. Dataflow of the adaptive zoom control architecture.

previous studies[6, 15] do not provide evidence on how to conduct the zoom control to reach the desired zoom level, but only zoom constraints. For instance, the authors in [6] merely mentioned that the camera is zoomed to hold a defined pixel range. However, there are many factors that should be considered such as inconsistent detection, zoom speed, which can cause zoom oscillation.

In the construction domain, Azar [1] improves construction equipment detection for an automated monitoring system for productivity. The construction equipment is detected via AprilTag [11]—a visual marker. The author proposed an automated zoom control algorithm in order to have reliable detection. The active zoom control maintains the minimum tag pixel resolution [2]. The zoom function of the crane load-view camera is essential for the crane operator. Vierling et al. [15] propose an automatic zoom load-view camera based on the working zone and load occlusion. The authors trained four CNNs with the load-view images as input. Each CNN corresponds to one zoom level. At the end, the arbiter picks the optimal zoom level which provides less occlusion and is suitable for the operator. Li et al. [6] increase the precision in tracking tower crane hook during hoisting to avoid blind-lifting. To capture the hook movement, the author used the pan–tilt–zoom (PTZ) surveillance camera to detect the hoist cable instead. The adaptive zoom is used to maintain the hook size on the monitor display.

Zoom camera is one of the essential stages in the object detection pipeline. Nonetheless, no study to date has examined the zoom mechanism on a mobile crane for object detection. To fill this literature gap, this paper identifies the adaptive zoom control method to maintain the quality of the worker detection from a load-view crane camera. The goal is to avoid data gathering and re-training deep learning algorithms. Second, the adaptive zoom gives a significant advantage for the crane operators as they have to simultaneously work on many tasks during the lifting. The proof of the adaptive zoom control is verified by using AprilTag detector. To our knowledge, no prior studies have examined zoom function on the mobile crane to improve worker detection for safety monitoring.

2 Proposed Approach

In this work, we propose an adaptive zoom control method to eliminate the re-training and data augmentation process in object detection using deep learning algorithm, and meanwhile increase the situational awareness of a crane operator.

The data-driven method mandates a large amount of training data to reach high accuracy. Detecting objects from a load-view crane camera is challenging especially in the construction area. The object appears in wide-ranging size and appearance. During lifting, the distance between the load-view camera and the ground is dynamically changing because the boom arm can be lowered or extended. Hence, the detected objects appear differently—small, medium or large[1] [9]. Additionally, the background is full of features that can be easily misclassified as a worker. On the other hand, recording data from the crane for training data-driven detector is expensive and effort demanding such as crane rental, (un)mounting sensor on the huge machine, and data annotation.

The image sensor in this work is a Motec MC5200 crane motor zoom camera mounted at the boom tip of the mobile crane with a pendulum bracket, see Fig. 1. The video output is analog which contains an interlaced display. It results a partially interlaced image after digitalization. The camera can do basic controller function by sending the output control command, *Zoom in* (O_{zi}) and *Zoom out*

[1]small (BBox< 32×32 px), medium (32×32 px < BBox< 96×96 px) and large(BBox> 96×96 px). BBox stands for bounding box.

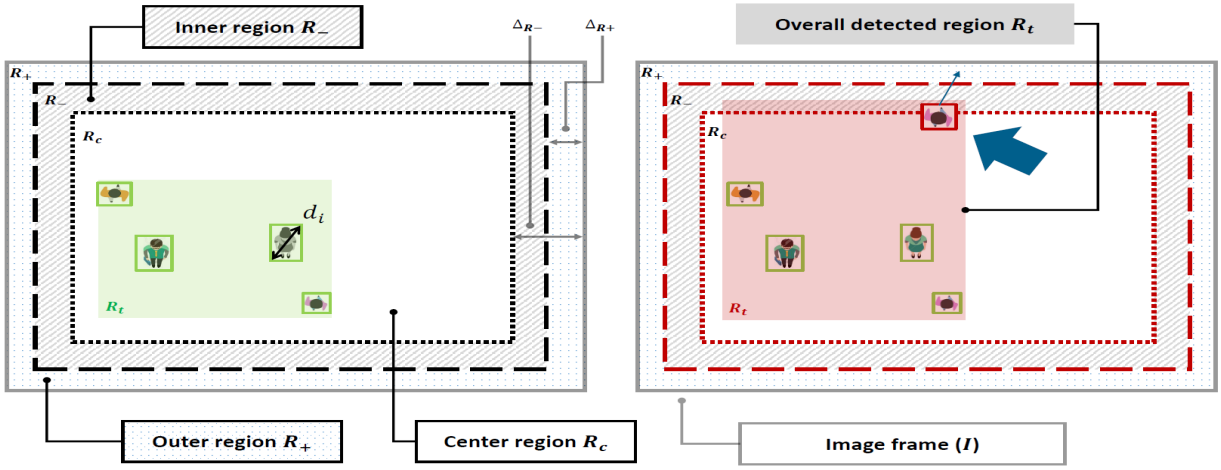

Figure 3. The sketches illustrate two scenarios of the proposed FSM, target size preservation (left) and target area preservation (right). R_+ is the outer region filled with a dot pattern. R_- is the inner region filled with an upward diagonal. R_c is the center region. R_t is the overall detected region which is shown in translucent. On the left figure, the green translucent region R_t identifies the satisfied constraints of size including the borders which show in black. On the other hand, the right figure depicts the violated case of both size and region which are identified by the red translucent rectangle and red border. d_i is a bounding box diagonal of target x_i in pixel.

(O_{zo}) which make the objects inside the image becomes larger and smaller, respectively.

2.1 System Architecture

In our system modules, there are two main components, namely perception, and control. First, the perception part is an object detector. In a real-world application, a worker detection using deep learning approach, whereas AprilTag detector, which is adopted from [11] used to evaluate our proposed zoom control method. Second, the control regulates the zoom level to satisfy the defined constraints. We applied a finite state machine (FSM) for control strategies to generate O_{zi} and O_{zo} pulse command as the output from our *ZoomController* by incrementally increase or decrease, respectively. The detail of the control logic will be further discussed in Sec. 3. The system data flow is shown in Fig. 2. The load-view crane camera feeds image frames to the detector. The detector processes the image and subsequently passes the recognized bounding boxes (BBoxes) to *ZoomController*. Finally, the controller generates the zoom command based on the observation back to the camera to adjust the zoom level.

2.2 Detectors

AprilTag Detector: The incoming sensor data, like the worker detector, can be inconsistent. It can cause difficulty to assess the zoom control logic. To decouple the control part from the perception, we then verify our control using AprilTag. The AprilTag detector is used as a reference of sensor data. This fiducial marker detector provides relatively more reliable and consistent sensor data. In other words, using the visual fiducial marker creates more measurable and controllable experiments[11]. Additionally, the output from both detectors, which is BBoxes, is comparable.

Worker Detector: In the real application, we adopted the load-view worker detector based on RetinaNet from [12]. The network of the detector is fine-tuned with the crane camera images. The data is collected from both simulation platform [13, 14] and the real construction environment. RetinaNet [8], a single-stage detector, introduced the focal loss to solve the class imbalance problem on top of Feature Pyramid Network (FPN) [7]. The featurized image pyramid addresses the problem of object detection at multiscale, while the focal loss is defined to penalize easy negative examples.

3 Zoom Controller

The proposed solution exploits the zoom function of the standard crane camera to keep the quality of the detector instead of parameter tuning and re-training the deep learning network. The principle idea of the method is to maintain the BBox size of all target instances while keeping them in the image frame as long as possible. In

particular, the zoom method preserves the targets in the image frame not to let them out of the camera field of view (FOV). The regions, $R_{\pm}, R_c$ are additionally defined to restrict targets by two image frame offsets $\Delta_{R\pm}$, shown in Fig. 3. Each offset is equally positioned in both x- and y-axis. The outer region R_+ is a prohibited zone, where any target should not be inside. R_- is an inner region while R_c is a center region.

The zoom control logic in *ZoomController* is mathematically modeled in the Mealy FSM, see Fig. 4. The nomenclature of the zoom control is described in Table. 1. The FSM is defined using a 6-tuple $(S, S_0, \Sigma, \Lambda, T, G)$ as the following.

- A finite set of all states $S = \{S_E, S_{TA}, S_{TD}\}$
- An initial or reset state $S_0 = S_E$
- A set of inputs $\Sigma = \{\overline{N}, \overline{D}, \overline{A}\}$
- A set of outputs $\Lambda = \{O_{zi}, O_{zo}\}$
- Transition function $T : S \times \Sigma \rightarrow S$
- Output function $G : S \times \Sigma \rightarrow \Lambda$
- A set of parameters $\Pi = \{D_D \pm \Delta_D, \alpha, R_{\pm}, R_c\}$.

The inputs of FSM are obtained by preprocessing the raw data X_t. The component of a detected target BBox consists of top left box point x_{tl}, y_{tl}, w, and h in an image coordinate. In addition to the set of inputs Σ, we have a set of parameters Π. The values of Π are chosen by trial and error experiments. The moving average (MA) is applied to the input data as a noise filter.

The set of states $S = \{S_E, S_{TA}, S_{TD}\}$ is designed to associate to three following scenarios, namely target loss, target area preservation and target size preservation, respectively. Fig. 3 depicts the last two scenarios. The pseudocode of the method is described in Alg. 1. We manipulate the zoom control by zoom level and perception. Zoom level Z ideally represents how much the camera lens has move based on the zoom pulse command as there is no original zoom control access. The following presents the definition of state machine in Fig. 4 including the transition T and output G function.

- State *Explore* S_E - This state corresponds to target loss case (T, G:`SearchTarget`). When there is no target or the detector is unable to recognize the target, the camera should explore or search for the target(s) by zooming in or out. The scenarios is depicted in Fig. 3 on the left. The state machine is initiated or reset to this state. The zoom level Z of the camera must be set first at the zoom out max ($Z_0 = 0$). Then the camera starts to *search* for targets until the target appears in the image frame or the detector is able to recognize it, which implies $\overline{N} > 0$. The searching procedure is carried out by zooming in (O_{zi}: $Z_{t+1} = Z_t + 1$) until the camera reaches maximum zoom in ($Z_t = Z_{ci,max}$) then it starts to zoom out (O_{zo}: $Z_{t+1} = Z_t - 1$). The procedure repeats until a reliable target is found. In this case, the next state goes to S_{TA} to further observe the overall target area.

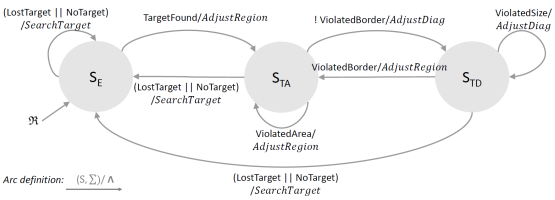

Figure 4. Mealy finite state machine diagram of the *ZoomController* logic. $\mathfrak{R}$ is a reset signal.

- State *TrackArea* S_{TA} - This state corresponds to target area preservation case which the overall target area is assured in the center area (T, G:`AdjustRegion`). Any target steps into the region R_+, the camera should adjust the zoom level to keep the target inside at least in the inner region R_- or the center region R_c as long as it is not beyond the camera FOV limit, see Fig. 3 on the right. In other words, if $\overline{A}$ intersects R_+, the camera zooms out until $\overline{A}$ intersects R_- or $\overline{A}$ does not anymore overlap with R_+. When the area criterion is satisfied, the next state goes to S_{TD}.
- State *TrackDiagonal* S_{TD} - This state corresponds to target size preservation case (T, G:`AdjustDiag`). The camera should adapt the zoom level to keep the average diagonal value of overall detected objects $\overline{D}$ to the ideal diagonal D_D which is suitable to the selected deep learning detector. In particular, this condition $(D_D - \Delta_D) \leq \overline{D} \leq (D_D + \Delta_D)$ should be satisfied. When the $\overline{D}$ is lower than the desired diagonal range, the camera zooms in to observe the targets closer, and vice versa. Unless the overall detected area $\overline{A}$ complies, the next state goes back to S_{TA} because the area criterion has higher priority than the diagonal one.

3.1 Zoom Controller Verification

This section presents the verification of the zoom controller using AprilTag as a reference target to evaluate the controller function. The AprilTag family is 36h11 with the size of 16×16 cm. The test was set up in a hallway. Both camera and the tag were placed on the same ground plane. The maximum distance from the camera to the corridor end was 20.3 meters. During the experiment, only the tag

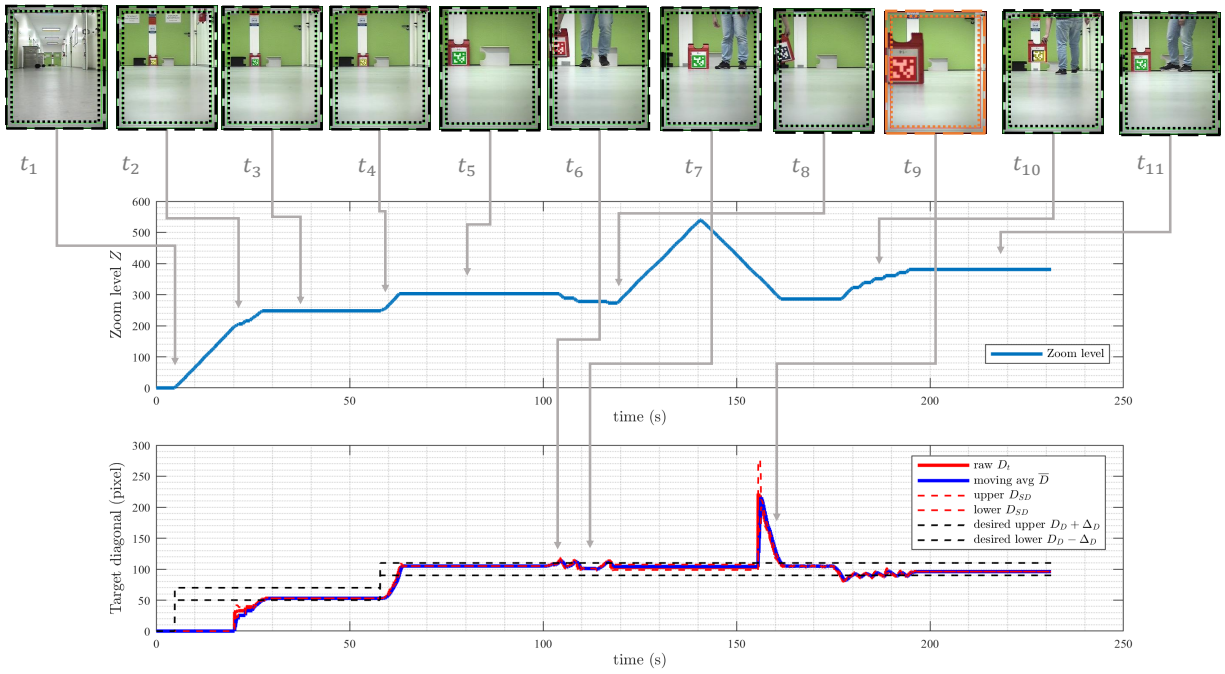

Figure 5. *ZoomController* verification using AprilTag. The dashed line locates in between the region R_+ and the region R_-, while the dotted line divides between the region R_- and the region R_c. The translucency on the tag identifies the size violation. The green tag at $t_{3,5,7,11}$ means $\overline{D}$ are in target range, $(D_D - \Delta_D) < \overline{D} < (D_D + \Delta_D)$. The yellow tag at $t_{2,4,10}$ means it is below the range, $\overline{D} < (D_D - \Delta_D)$. The red tag at $t_{6,9}$ means it is over the range, $\overline{D} > (D_D + \Delta_D)$. The red border at t_9 identifies the area violation i.e., $\overline{A}$ overlaps with R_+.

was moved farther away or nearer to the camera. The D_D is primarily set to 60 with its offset Δ_D of 10 pixels. The graph in Fig. 5 shows how the zoom level Z adapted to the target. In each image frame, the dashed line locates in between the region R_+ and the region R_-, while the dotted line divides between the region R_- and the region R_c.

At t_1, both $\overline{D}$ and Z_t are zero because no tag was found. Therefore, the FSM started to search for the target by O_{zi}. Despite the tag was found at t_2, the FSM continued to zoom in because $\overline{D}$ remained lower than the floor of D_D. At t_3, Z_t started to be steady as it met the diagonal criterion. At t_4, D_D was later manually increased, thus the zoom control started to zoom in and $\overline{D}$ was then back again in range at t_5.

From t_6 to t_7, Z_t slightly depreciated because *ZoomController* tried to maintain the size by O_{zo} as the tag was moved toward the camera which caused $\overline{D}$ became larger and accordingly exceeded $D_D + \Delta_D$.

Between t_8 and t_9, the tag was removed out of the camera FOV. For this reason, *ZoomController* went to the explored state S_E. At t_9, $\overline{D}$ suddenly soared up during the searching because the tag immediately appeared with violated $\overline{D}$ and $\overline{A}$ where the tag was colored in translucent red and the borders visualized in red, respectively.

At t_{10}, The zoom level Z_t gradually rose because the tag is moved away from the camera. $\overline{D}$ became lower than the desired range where the tag was colored in translucent yellow. *ZoomController* simultaneously tried to handle until the tag is back in the D_D range at t_{11}.

4 Experiment

For the worker detector, we simulated the camera position on the crane by mounting the camera from the rooftop of a building. The camera looked down to the parking lot which has an even surface. The approximate distance from the camera to the ground was 22 meters which is equivalent to the height of a 6-story building. The $D_D \pm \Delta_D$ was initially set to 60 ± 10 pixels. During the experiment, four workers walked into the camera field of view. In spite of worker safety requirements, the workers did not wear any protective gear or PPE. Most of the traditional detector methods exploited the PPE appearance to ease the detector which allowed the detector to see the target better [12]. However, there are many incidents of non-compliant workers violating the rules [10]. Thus, it is better to set the construction environment as close as possible.

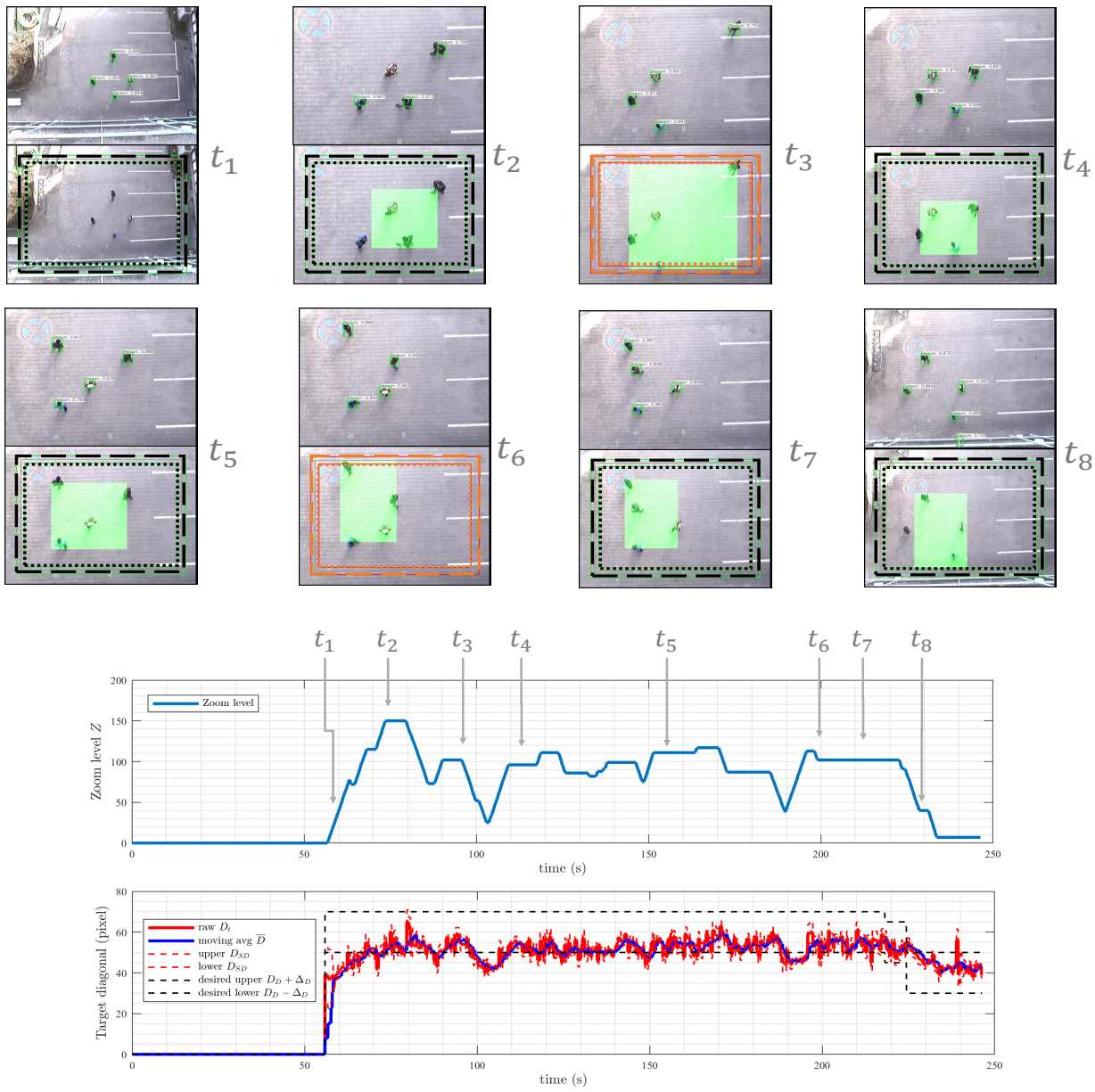

Figure 6. *ZoomController* experiment with the worker detection from load-view crane camera. The upper image row of each timepoint shows the raw data from the detector while the lower image shows the visualized result from the *ZoomController*. The red border identified the area violation which means there is an intersection area between $\overline{A}$ and R_+.

4.1 Result

The graph in Fig. 6 shows how the zoom level Z adapted to the detected workers. The figure consists of two main parts, which are snapshot image frames and the zoom control result. In the image section, the 1st and 3rd row of images show the detection result while the 2nd and 4th row display the same image frame with the result for zoom control logic.

At t_1, Z_t swiftly increased because the average overall diagonal size $\overline{D}$ was below the floor of D_D. At t_2, Z_t is stable because the size preservation was complied.

At t_3, the border violation occurred. Two of the workers walked near toward the prohibited region R_+. One

Algorithm 1: Adaptive zoom control algorithm

Input: Σ, I
Output: $\Lambda = \{O_{zi}, O_{zo}\}$
Parameters: $\Pi = \{D_D \pm \Delta_D, \alpha, R_{\pm}, R_c\}$
Initialization;
$Z_0 := 0$;
$S := S_0$;
while *true* **do**
  $X_t := \texttt{DetectTarget}(I)$;
  $N := n\{X_t\}$;
  $\overline{N}_0 := (\overline{N} == 0)$;
  $R_t := BBox(min\ P_X, min\ P_Y, w, h)$;
  $d_i := \sqrt{w_i^2 + h_i^2}$;
  $D_t := \frac{\sum_{i=1}^{N} d_i}{N}$;
  $\overline{A} := \frac{1}{\alpha-1} \sum_{i=1}^{\alpha} R_i := \frac{R_1+R_2+\cdots+R_{\alpha-i}}{\alpha}$;
  $\overline{D} := \frac{1}{\alpha-1} \sum_{i=1}^{\alpha} d_i := \frac{d_1+d_2+\cdots+d_{\alpha-i}}{\alpha}$;
  if $\overline{N}_0$ **then**
    $S := S_E$;
    $\texttt{SearchTarget}(S, \overline{N})$;
  else
    if $\overline{A}$ *is violated* **then**
      $S := S_{TA}$;
      $\texttt{AdjustRegion}(S, R_{\pm}, R_c, R_t, \overline{N})$;
    else
      if $\overline{D}$ *is violated* **then**
        $S := S_{TD}$;
        $\texttt{AdjustDiag}(S, D_D \pm \Delta_D, \overline{D}, \overline{N})$;
      end
    end
  end
end

Table 1. Nomenclature for zoom controller.

Symbol	Definition	Value
Input		
$\overline{A}$	Moving average R_t	-
d_i	Bounding box diagonal of target x_i (pixel), see Fig. 3	-
D_{SD}	Standard deviation of D_t (pixel)	-
D_t	Instant average diagonal of all targets (pixel)	-
$\overline{D}$	Moving average of D_t (pixel)	-
I	Image frame	-
N	Instant target number	-
$\overline{N}$	Moving average of N	-
P_X, P_Y	A set of instant target BBox coordinates in x- and y-axis	-
R_t	Instant overall detected region, see Fig. 3	-
X_t	A set of detected targets at time t	-
Output		
O_{zi}, O_{zo}	Input zoom control command in and out	-
Z_t	Instant zoom level	$[0, Z_{ci,max}]$
Parameters		
D_D	Desired BBox diagonal (pixel)	60
$R_{\pm}$	Outer and inner region, see Fig. 3	-
R_c	Center region, see Fig. 3	-
α	Moving average window size	5
Δ_D	Range of the desired BBox diagonal D_D (pixel)	10
$\Delta_{R+,T}, \Delta_{R-,T}$	Image frame offset of R_+ and R_- for tag detector	(5,35)
$\Delta_{R+,W}, \Delta_{R-,W}$	Image frame offset of R_+ and R_- for worker detector	(45,75)

worker was toward the top right corner and the other was toward the bottom of the image frame. Consequently, Z_t decreased in between t_3 and t_4 until the workers stayed inside the inner region R_-. After the area adjustment, the zoom level went up because of the size violation before t_4 where *ZoomController* reached the stable state S_E. At t_5 including the nearby period, the zoom level was gently changed and kept $\overline{D}$ in the desired diagonal range because of the size change. Likewise, the border violation again happened as the border showed in red at t_6. One worker walked toward the top of the frame. As the result, the camera zoomed out and later Z_t became consistent at t_7. At t_8, the camera again plunged because the parameter D_D was configured to a smaller value.

In summary, our results demonstrated that adaptive zoom control can improve the quality of data-driven worker detection. To evaluate the zoom control logic, we replaced the worker detector with AprilTag detector which provides a reference target. The verification performs well, giving the correct result as is defined in the FSM. The zoom level was adjusted without jitters. The camera first could not detect the target because of the too-small size object. With adaptive zoom control, the target was able to be recognized from distance. When comparing AprilTag results to the worker detector, it must be pointed out that there is a lot of noise from the sensor data. In other words, the RetinaNet could not constantly detect all four workers despite the MA filter. However, the *ZoomController* functions in the same manner without any zoom oscillation. The controller is able to handle the side effect of the detector such as miss detection, anomaly, and outlier detection. Furthermore, running deep learning detector on less powerful hardware including the video transmission introduced a delay of approximately two seconds in our experiment. Our proposed method can be adjusted to this lagging via the parameters Π.

A major source of limitation is the lack of camera control information access such as zoom level. In the proposed method, we use an approximate estimation of zoom level Z by incrementing the Z counter up and down when zooming in/out is executed. Additionally, the crane control information panel of the operator e.g., hook length measurement and boom angle are definitely would be beneficial to refine the detector. For instance, the zoom controller can be performed based on hook cable length in addition to the proposed criteria.

5 Conclusion and Outlook

In this paper, we investigated the adaptive zoom control of the load-view crane camera for worker detection. This is an important finding in the understanding of how to handle the zoom control to reach the zoom criteria. We exploited the zoom mechanism which exists in typical mobile crane cameras. The proposed method adopts the

Mealy FSM to observe and determine the zoom command which suitable for the given situations, namely target loss, target area preservation, and target size preservation. The state definition is characterized by the three scenarios. The evaluation is first verified by using the reliable and controllable detector, AprilTag. Our proposed zoom control method is able to smoothly adapt to the problem of deep learning object detection, which is inconsistent detection and detecting small size objects.

Further studies should investigate sensor fusion with more access to crane information for zoom control improvement. An additional monocular camera can be installed nearby the zoom camera. The monocular camera provides overview information to the zoom camera. Although the zoom camera status is in maximum zoom in, the overview camera can notify *ZoomController* of the zoom camera if there is a new incoming target then the zoom camera can zoom out. Moreover, the position of workers including the velocity in world coordinate can be estimated by camera projection and object tracking. Hence, the risk of each worker can be assessed for the operator safety assistance system. For instance, if the worker walks away from the crane, the risk of the worker getting hit by the crane is low.

References

[1] E Rezazadeh Azar. Active control of a pan-tilt-zoom camera for vision-based monitoring of equipment in construction and surface mining jobsites. In *ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction*, volume 33, page 1. Vilnius Gediminas Technical University, Department of Construction Economics . . . , 2016.

[2] Ehsan Rezazadeh Azar. Construction equipment identification using marker-based recognition and an active zoom camera. *Journal of Computing in Civil Engineering*, 30(3):04015033, 2015.

[3] Yinghao Cai and Gérard Medioni. Persistent people tracking and face capture using a ptz camera. *Machine Vision and Applications*, 27(3):397–413, 2016.

[4] Yihai Fang, Yong K Cho, and Jingdao Chen. A framework for real-time pro-active safety assistance for mobile crane lifting operations. *Automation in Construction*, 72:367–379, 2016.

[5] Mate Kisantal, Zbigniew Wojna, Jakub Murawski, Jacek Naruniec, and Kyunghyun Cho. Augmentation for small object detection. *arXiv preprint arXiv:1902.07296*, 2019.

[6] Yanming Li, Shuangyuan Wang, and Bingchu Li. Improved visual hook capturing and tracking for precision hoisting of tower crane. *Advances in Mechanical Engineering*, 5:426810, 2013.

[7] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[8] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[9] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[10] OHS. Survey Finds High Rate of PPE Non–Compliance, November 2008. [Online; accessed 18-06-2019].

[11] Edwin Olson. Apriltag: A robust and flexible visual fiducial system. In *2011 IEEE International Conference on Robotics and Automation*, pages 3400–3407. IEEE, 2011.

[12] Tanittha Sutjaritvorakul, Axel Vierling, and Karsten Berns. Data-driven worker detection from load-view crane camera. In *Proceedings of the 37th International Symposium on Automation and Robotics in Construction (ISARC)*, pages 864–871. International Association for Automation and Robotics in Construction (IAARC), 2020.

[13] Tanittha Sutjaritvorakul, Axel Vierling, and Karsten Berns. Simulated environment for developing crane safety assistance technology. In *Commercial Vehicle Technology 2020. Proceedings of the 6th Commercial Vehicle Technology Symposium – CVT 2020*, Kaiserslautern, Germany, 2020.

[14] Tanittha Sutjaritvorakul, Axel Vierling, Jakub Pawlak, and Karsten Berns. Simulation platform for crane visibility safety assistance. In *Advances in Service and Industrial Robotics*, volume 84 of *Mechanisms and Machine Science*, pages 22–29. Springer International Publishing, 2020.

[15] Axel Vierling, Tanittha Sutjaritvorakul, and Karsten Berns. Crane safety system with monocular and controlled zoom cameras. In *ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction*, volume 35, pages 1–7. IAARC Publications, 2018.