Latency-Aware 360-Degree Video Analytics Framework for First Responders Situational Awareness

Jiaxi Li
University of Illinois
Urbana-Champaign
Urbana, Illinois, USA

Jingwei Liao
George Mason University
Fairfax, Virginia, USA

Bo Chen
University of Illinois
Urbana-Champaign
Urbana, Illinois, USA

Anh Nguyen
George Mason University
Fairfax, Virginia, USA

Aditi Tiwari
University of Illinois
Urbana-Champaign
Urbana, Illinois, USA

Qian Zhou
University of Illinois
Urbana-Champaign
Urbana, Illinois, USA

Zhisheng Yan
George Mason University
Fairfax, Virginia, USA

Klara Nahrstedt
University of Illinois
Urbana-Champaign
Urbana, Illinois, USA

ABSTRACT

First responders operate in hazardous working conditions with unpredictable risks. To better prepare for demands of the job, first responder trainees conduct training exercises that are being recorded and reviewed by the instructors, who check for objects indicating risks within the video recordings (e.g., firefighter with an unfastened gas mask). However, the traditional reviewing process is inefficient due to unanalyzed video recordings and limited situational awareness. For better reviewing experience, a latency-aware Viewing and Query Service (VQS) should be provided. The VQS should support object searching, which can be achieved using the video object detection algorithms. Meanwhile, the application of 360-degree cameras facilitates an unlimited field of view of the training environment. Yet, this medium represents a major challenge because low-latency high-accuracy 360-degree object detection is difficult due to higher resolution and geometric distortion. In this paper, we present the Responders-360 system architecture designed for 360-degree object detection. We propose a Dynamic Selection algorithm that optimizes computation resources while yielding accurate 360-degree object inference. The results, using a unique dataset collected from a firefighting training institute, show that the Responders-360 framework achieves 4x speedup and 25% memory usage reduction compared with the state-of-the-art methods.

CCS CONCEPTS

• **Applied computing → Computers in other domains**; • **Information systems → Multimedia information systems**; • **Computing methodologies → Machine learning**.

KEYWORDS

360 Video Analytics, Latency-Aware 360 Object Detection, Dynamic Tile Selection Algorithm

ACM Reference Format:
Jiaxi Li, Jingwei Liao, Bo Chen, Anh Nguyen, Aditi Tiwari, Qian Zhou, Zhisheng Yan, and Klara Nahrstedt. 2023. Latency-Aware 360-Degree Video Analytics Framework for First Responders Situational Awareness. In *The 33rd edition of the Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV '23), June 7–10, 2023, Vancouver, BC, Canada.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3592473.3592568

1 INTRODUCTION

First responders work in high-risk environments like natural disasters, crime scenes, and traffic accidents, where situational awareness is challenging due to out-of-view risks[15]. To mitigate these risks and enhance their preparedness, trainees participate in regular drills, where fixed cameras record their practical exercises. Instructors review the recorded videos to identify potential risks, such as trainee misbehaviors and hazards around them, using object detection techniques and a priority-based approach. For example, Table 1 presents the objects of interest and their attention priorities during firefighting activities. A high priority means the object is of particular importance and is highly related to the safety of the firefighter trainee. Guided by the objects within the video recordings, instructors teach trainees to take appropriate action to avoid risks and correct their misbehaviors.

However, reviewing video recordings can be very inefficient and inconvenient for instructors because 1) Traditional cameras only capture a limited field of view, which is not desirable for first responder working environments where risks could come omnidirectionally, and 2) Without video analyzing, instructors have to watch the entire clip to look for moments where an event of risk occurs, which takes extra time and effort.

To improve instructors' reviewing experience, a 360-Degree Viewing and Query Service (VQS) has been proposed to provide comprehensive situational awareness of the training environment

NOSSDAV '23, June 7–10, 2023, Vancouver, BC, Canada

ACM ISBN 979-8-4007-0184-9/23/06…$15.00
https://doi.org/10.1145/3592473.3592568

Object of Interest	Priority
Civilian	High
Fire	High
Smoke	High
Gas Mask	High
Firefighter	Low
Helmet	Low

Table 1: Objects of interest with attention priorities within firefighting activities. The data is based on an interview with a physical training instructor at a training institute[15].

through spherical videos[2, 3, 20]. Features including Object Searching and Object-Based Viewport Recommendation should be offered in the VQS to help the instructors locate events of interest and prevent them from missing objects that are outside the viewport. We propose to utilize 360-degree object detection techniques to automatically discover object of interest from the videos.

However, object detection on 360-degree video is a non-trivial problem due to 1) Geometric distortion[1, 7, 17, 21, 25] and 2) Larger frame size[25]. Preferably, object query results should be delivered to instructors within a short time, which requires low-latency object inference. However, the higher number of pixels inside 360-degree videos leads to longer processing time and higher GPU memory requirement. Meanwhile, for better compatibility, commercial cameras often store 360-degree videos in equirectangular projection (ERP) representation[24, 27], which projects spherical videos into rectangular-shaped videos in a 2D plane, causing distortion of the objects. Yet, distorted objects introduced by ERP representation can cause the state-of-the-art 2D object detectors[5, 6, 13, 14] to misclassify objects at the edge of the video screen, which reduces detection accuracy. Previous works have proposed a dual-projection approach that crops and projects each ERP frame into 6 *cubic tiles* to reduce distortion and GPU memory utilization. However, the extra computation time due to the projections was not well addressed.

To tackle the above challenges, we propose Responders-360, a framework for latency-aware 360-degree video object detection based on the dual-projection approach. To address the extra processing time, we design a novel Dynamic Selection optimization algorithm, which efficiently filters out unnecessary projections and inferences based on inter-frame similarity and another novel metric, *object cohesion*, that reflects the likelihood of objects appearing in a *cubic tile* at a timestamp. Evaluation results based on a testing dataset of 420 video frames collected from fire drills at a training institute show that Responders-360 achieved better performance in detection accuracy, speed, and memory utilization on 360-degree videos with different resolutions compared to the state-of-the-art methods.

The remainder of this paper is structured as follows. Section 2 discusses the system requirements and challenges of providing the VQS. Section 3 presents previous research works on 360-degree object detection. Section 4 introduces the Responders-360 framework with a description of the system components and the data flow. Section 5 illustrates the Dynamic Selection optimization algorithm in detail. The evaluation of the Responders-360 framework is provided in Section 6. Section 7 concludes the paper.

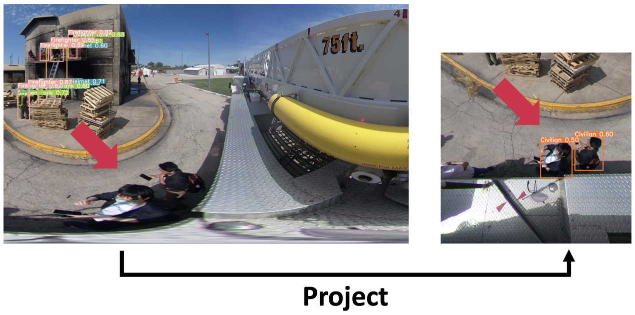

Figure 1: Examples of distortion in the ERP representation of a 360-degree frame. The distortion is reduced after projecting the ERP-based frame to cubic representation.

2 REQUIREMENTS AND CHALLENGES

2.1 Viewing and Query Service

To facilitate an efficient reviewing experience, a viewing and query service (VQS) with the following features is desirable.

Object Searching: Instructors should be able to query for a specific category of object from a video. The VQS should respond with timestamps of all occurrences of the requested object in a timely manner, so that the instructor can directly watch the intervals which contain the desirable object.

Object-Based Viewport Recommendation: The VQS should contain a video player with one or multiple viewing windows, or viewports[4, 12, 16, 18] , each presenting a portion of the 360-degree video with respect to a different viewing direction. The VQS should support automatically adjusting the viewing directions and offer the portion with higher density of objects to the instructors.

2.2 Challenges on 360-Degree Video Object Detection

Distortion: Geometric distortion appears in the pole areas of 360-degree images in the equirectangular projection (ERP) representation, which can mislead the object detector. For example, the image on the left in Figure 1 represents an example of ERP-based 360-degree frames. As indicated by the arrow, the two civilians in the pole area are distorted and are missed by the detector.

Many previous works have proposed to build an object detector that directly works on ERP representation of 360-degree images[1, 17, 19, 24], but they suffer from poor detection accuracy.

Frame Size: 360-degree videos are generally of 4K resolution and contain 4 times the amount of pixels of a normal 1080p video. Inference on high resolution videos require advanced GPUs and more memory, which may not be supported by a general training institute. In addition, fast delivery of labeled videos and query response is desired by the instructors. However, longer processing time due to more pixels can negatively influence fast delivery guarantee.

3 RELATED WORKS

Modern 2D Object Detection Frameworks: Modern 2D object detectors are roughly divided into two categories, including 1) one-stage detectors, e.g., YOLO[13] and SDD[9], and 2) two-stage

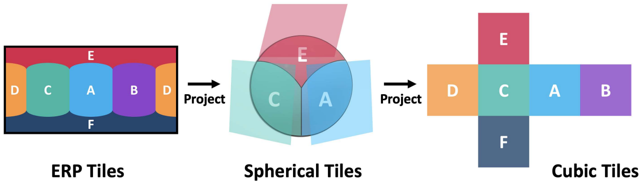

Figure 2: Dual-Projection Process.

detectors, e.g., R-CNN[6], Faster R-CNN[14] and FPN[8]. Considering the system burden on latency in our case, the faster one-stage detectors are more suitable to build the VQS for first responder training. However, due to severe geometric distortion on the frame edge of ERP-based 360-degree videos, detection accuracy degrades when 2D object detectors are directly applied to 360-degree videos[1, 17, 19, 23, 24, 26].

Dual-Projection: A general approach to remove distortion in ERP frames is through a dual-projection process[11, 20]. Figure 2 provides an overall description of this process. The basic idea is to segment each input ERP frame into 6 *ERP tiles*, transform each *ERP tile* to the corresponding title in the spherical representation (*spherical tile*) under the first projection, and transform each *spherical tile* to a *cubic tile* using the cube mapping method. Other than detecting on the whole input frame, the detection in the dual-projection approach is on a tile basis and is performed on the 6 *cubic tiles*, which involve less distortion. For example, the image on the right of Figure 1 represents a *cubic tile* projected from a ERP frame. The dual-projection process reduces the distortion on the pole area, which enables the object detector to discover the object of interest (civilian).

Although the dual-projection process can potentially reduce distortions in ERP videos, it does not consider the extra computation time introduced by the projections. Experimental results from a laptop with a 4-cores CPU show that the dual-projection process requires about 2 seconds on average for a frame of 1080P resolution, which is more than 30% of the time required by performing object detection for a frame. As a result, projecting every input frame would increase system burden on processing speed.

360-Degree Object Detection Frameworks: Other works proposed object detection frameworks specially designed for 360-degree videos. For example, based on Faster R-CNN, Su et al.[17] introduced the first spherical convolutional neural network (S-CNN) for object detection, which extracts features from spherical 360-degree images. Yu et al.[26] and Wang et al.[19] proposed their own 360-degree object detectors based on R-CNN and S-CNN respectively, which directly process 360-degree images in ERP format. However, most of the 360-degree object detection frameworks originate from two-stage detectors, which requires longer processing time and hence are not suitable in our case. In addition, some of the detectors, such as [19, 22, 26], still suffered from geometric distortion.

4 SYSTEM OVERVIEW

Our Responders-360 object detection framework is comprised of the following three stages, as indicated in Figure 3,

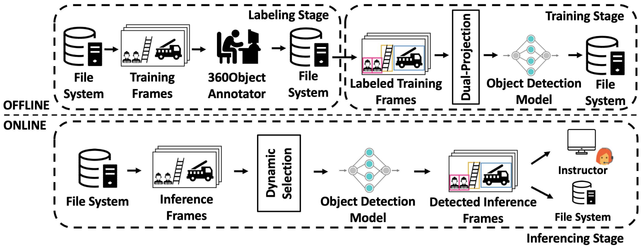

Figure 3: Overview of the Responders-360 Object Detection Framework.

Offline Labeling Stage: The objects in the training frames are manually labeled using our 360ObjectAnnotator [10], which provides precise and distortion-free annotations for 360-degree videos. The labeled frames are saved in the file system for future training.

Offline Training Stage: Each labeled frame is projected into 6 *cubic tiles* in the normal 2D perspective through the dual-projection process. A base 2D object detector is trained from the *cubic tiles*. The trained model parameters are saved in the file system for future inference.

Online Inference Stage: When an object query arrives, the VQS checks whether object detection for the video has been completed, and sends the video with objects to the instructor if completed. Otherwise, the inference stage is initiated. In our framework, a dynamic selection optimization step is introduced to determine if detection is necessary for a *cubic tile*. Repetitive and useless computation of dual-projection and detection steps is skipped to improve the system performance. When the inference is completed, the video with the detected objects is sent to the instructor as well as saved into the file system for answering later queries.

Next, we discuss the Dynamic Selection algorithm of the Responders-360 framework.

5 DYNAMIC SELECTION

As discussed in Section 3, one of the major issues left by the dual-projection approach is the extra processing time due to transforming *ERP tiles* to *cubic tiles*. In our Responders-360 framework, we propose a novel Dynamic Selection algorithm which efficiently filters out unnecessary projection and detection.

5.1 Algorithm Overview

The Dynamic Selection algorithm is summarized in Algorithm 1, which involves the following steps.

- A) Segment each input frame into 6 *ERP tiles*. In our algorithm, j is used to index *ERP tiles* and *cubic tiles*.
- B) Transform *ERP tiles* of the first frame ($i = 1$) to *cubic tiles*, perform detection, and collect tile-based results in $R[1][1...6]$
- C) For the rest frames ($i > 1$), compute the inter-frame similarity (*sim*) for each *ERP tile* between its corresponding *ERP tile* in the previous timestamp.
- D) If *sim* is higher than the threshold *ts*, the previous timestamp's tile-based inference result ($R[i-1][j]$)[1] is adopted for the current tile, otherwise Step E is conducted.

[1] We are using i to interchangeably refer to frame number and timestamp.

Algorithm 1 Dynamic Selection

```
Input:  $I[1...N]$: The input ERP frames
            ▷ $N$: Number of frames in the video
        $ts$: The inter-frame similarity threshold
        $tc$: The object cohesion threshold
Output:  $R[1...N]$: The inference results for input ERP frames
Initialize:  $ET[1...N][1...6]$: The ERP tiles
             $RT[1...N][1...6]$: The tile-based inference results
for $i = 1, i++, i \leq N$ do ▷ Iterating through the video frames
    $ET[i][1...6] \leftarrow Segment(I[i])$                ▷ Step A

for $j = 1, j++, j \leq 6$ do            ▷ Iterating through the 6 tiles
    $cubicTile_{1j} \leftarrow DualProjection(ET[1][j])$     ▷ Step B
    $RT[1][j] \leftarrow Inference(cubicTile_{1j})$

for $i = 2, i++, i \leq N$ do
    for $j = 1, j++, j \leq 6$ do
        $sim \leftarrow ComputeSIM(ET[i-1][j], ET[i][j])$   ▷ Step C
        if $sim \geq ts$ then
            $RT[i][j] \leftarrow RT[i-1][j]$                ▷ Step D
        else if $sim < ts$ then
            $oc \leftarrow ComputeOC(R[i-1][j])$           ▷ Step E
            if $oc < tc$ then
                $R[i][j] \leftarrow R[i-1][j]$
            else if $oc \geq tc$ then
                $cubicTile_{ij} \leftarrow DualProjection(ET[i][j])$
                $R[i][j] \leftarrow Inference(cubicTile_{ij})$

for $i = 1, i++, i \leq N$ do                               ▷ Step F
    $R[i] \leftarrow MergeResults(RT[i][1...6])$
```

- E) The object cohesion (oc), discussed in detail in Section 5.3, is computed based on the previous timestamp's tile-based inference result ($R[i-1][j]$). If oc is lower than the threshold tc, the tile-based inference result in the previous timestamp will be adopted for the current tile. Otherwise, *ERP tiles* are transformed to *cubic tiles* and the tile-based results are collected based on detection results on the *cubic tiles*.
- F) Merge the title-based results for each input frame when the inference is completed.

The complexity of processing each image is primarily determined by the function *ComputeSIM*, which takes a complexity of $O(S)$ where S is the size (width times height) of the image. Therefore, the complexity of the Algorithm 1 can be simplified as $O(NS)$, where N is the number of frames in the video.

The Dynamic Selection algorithm is motivated by the idea that, not every pixel in the 360-degree frames is useful and meaningful for the purpose of object detection. Hence, we made the assumption that, for *cubic tile* j at timestamp i, or $cubicTile_{ij}$, detection should be performed only when the following two conditions are satisfied.

- Condition (1) $cubicTile_{ij}$ is structurally different from the corresponding *cubic tile* in the previous frame ($cubicTile_{i-1j}$)
- Condition (2) $cubicTile_{i-1j}$ contains a high object cohesion

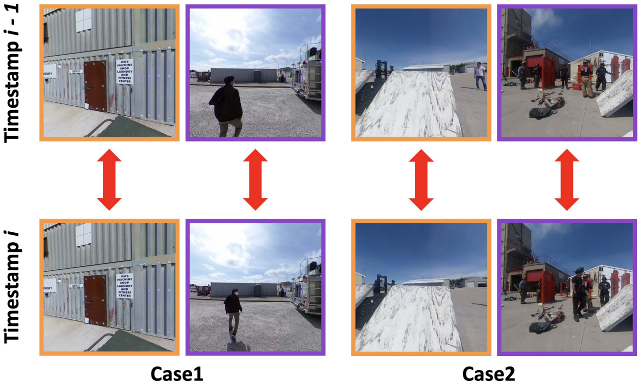

Figure 4: Case 1 provides an example of two pairs of *cubic tiles* with different inter-frame similarity. Case 2 provides an example of two pairs of *cubic tiles* with similar inter-frame similarity. In both cases, a new detection in timestamp i is needed only for the purple *cubic tile.*

5.2 Inter-Frame Similarity

Object detection on *cubic tiles* for every frame is redundant, because *cubic tiles* among different timestamps may be structurally similar and thus yield similar detection results. For example, as indicated by Case 1 in Figure 4, in between timestamp $i-1$ and i, the purple *cubic tile* has structurally changed while the orange *cubic tile* remains the same. Therefore, the dual-projection process and object inference for the purple *cubic tile* of timestamp i are necessary to discover the new position of the civilian, but would be unnecessary for the orange *cubic tile*. It would be sufficient for the orange tile of timestamp i to take the inference result of the previous timestamp.

Existing image similarity metrics can be utilized to check if neighboring *cubic tiles* have structurally changed. However, directly calculating inter-frame similarities on *cubic tiles* is time consuming due to the required dual-projection process to transform *ERP tiles* to *cubic tiles*. Instead, we propose inter-frame *ERP tile* similarity (sim), which does not require computation of the dual-projection process, could potentially forecast the inter-frame *cubic tile* similarity.

Meanwhile, approaches are taken to reduce the numerical deviation led by the distortion. In our algorithm, a correction matrix is applied to the input ERP frame to assign less weight on highly distorted portions of the frame when calculating the inter-frame similarity. The frame-based correction step is summarized in the following formula where I referred to the input frame and I' referred to the corrected frame.

$$I' = M \cdot I$$

The correction process is fast because it involves a single time of matrix multiplication. In addition, the proposed correction matrix M can be pre-computed because the correction weights are the same for all frames in one video, which has a constant resolution.

For a specific pixel on the input frame, the correction weight it received is summarized in the following formula.

$$x' = x \cdot \frac{pixel\ sample\ rate}{center\ sample\ rate} = x \cdot \frac{2\pi r cos(\theta)}{2\pi r} = x \cdot cos(\theta)$$

x refers to a pixel in the input frame and x' refers to x after correction. r represents the equatorial perimeter of the sphere. $cos(\theta)$

represents the corrected weight, which is negatively correlated to the degree of distortion that is reflected by θ, the latitude of the pixel in the spherical representation. When spherical images are projected to the ERP representation, distortion is created as lines of latitude with different circumferences are stretched and projected into horizontal lines of the same length in the 2D plane. Therefore, the higher the latitude, the larger the distortion.

Based on the corrected frame, the following formula is proposed to calculate the inter-frame *cubic tile* similarity for *cubic tile* j in between neighboring timestamps $i-1$ and i,

$$sim(cubicTile_{i-1j}, cubicTile_{ij}) = NRMSE(ET'_{i-1j}, ET'_{ij})$$

ET'_{i-1j} and ET'_{ij} refers to the corrected *ERP tile* j in between neighboring timestamps $i-1$ and i, respectively. The Normalized Root Mean Square Error (NRMSE) is selected to measure the similarity on corrected *ERP tiles* because of the fast calculation speed.

If sim is higher than ts, an empirically determined threshold, the neighboring *cubic tiles* are determined to be the same. Otherwise, object cohesion (oc) of the previous *cubic tile* is calculated.

5.3 Object Cohesion

The fact that neighboring *cubic tiles* are structurally different does not solely mean that a new detection is needed. For example, in Case 2 of Figure 4, the inter-frame similarity calculation has output similar values for the orange *cubic tiles* and the purple *cubic tiles* in between timestamp $i-1$ and i. However, a new detection in timestamp i should only be performed on the purple *cubic tile* to discover the new position of the firefighter, and should not be performed on the orange *cubic tile* because it now involves no object of interest.

To account for this scenario, a new metric, *object cohesion* (oc), is introduced in our algorithm, which reflects if any objects will remain in the *cubic tile* in the next timestamp. High oc means that objects are more likely to appear in the same *cubic tile* at the next timestamp, which leads to a higher priority for a new detection.

The calculation of oc is based on the Principle of Temporal Locality, which is assumed to be existing in first responder training videos. Temporal locality means that if an object appears in a *cubic tile*, it has a tendency to stay in the same *cubic tile* for the next few timestamps. The following factors are taken into consideration when measuring the tendency.

- **Distance to the *cubic tile* center** The closer the object is to the center of the *cubic tile*, the more likely it will stay in the same *cubic tile* in the next timestamp.
- **Object size** The larger the object, the more likely it will move to other *cubic tiles* in the next timestamp.
- **Confidence level** Higher confidence level means that the detection result of an object is more likely to be correct.

The value of oc is calculated as the sum of individual tendency values of each object in the *cubic tile*. Specifically, the following formula was proposed in our work to calculate oc for *cubic tile* j in timestamp i.

$$oc(cubicTile_{ij}) = \sum_{object\ o} tendency(o) = \sum_{object\ o} \frac{c_o}{s_o \cdot d_o}$$

For object o in $cubicTile_{ij}$, d_o refers to the distance from o to the *cubic tile* center, calculated as the distance from the bounding box center to the *cubic tile* center. s_0 measures the size of o, calculated as the ratio of the bounding box size to the *cubic tile* size. c_o refers to the detection confidence level of o.

For Case 2 of Figure 4, the purple *cubic tile* in timestamp $i-1$ has a higher oc because several objects of interest (firefighters) are clustered in the *cubic tile* center. Yet, for the orange *cubic tile*, only one object of interest (civilian) exists in the margin of the *cubic tile*, which leads to a low oc. Hence, compared to the orange *cubic tile*, the purple *cubic tile* should have a higher priority for a new detection in timestamp i.

If oc of a *cubic tile* at timestamp $i-1$ is lower than tc, an empirically determined threshold, it is determined that no object will appear in the same *cubic tile* at the next timestamp. As a result, *cubic tile* at timestamp i will not receive a new detection.

6 EXPERIMENTAL VALIDATION

6.1 Models

The following models were selected for comparison, including

- **YV3**: Detecting using the YOLOv3 model on ERP frames
- **DP**: Detecting using the YOLOv3 model on cubic tiles generated by the Dual-Projection process
- **DS**: Our approach, detecting using the YOLOv3 model on cubic tiles generated by the Dynamic Selection algorithm

YOLOv3[13] is selected as the base 2D object detection model for its fast speed. To validate the effectiveness of the proposed Dynamic Selection algorithm, we first compare it to the traditional dual-projection-based approach (DP) over processing speed and detection accuracy. To validate the improvement over GPU memory usage and accuracy led by the projection, we also introduce one non-projection-based approach (YV3), and compare it to projection-based approaches (DP and DS).

6.2 Dataset

The performance of each approach is evaluated on a unique dataset of 25 360-degree videos collected at a firefighting training institute. Among all the video recordings, 19 (76%) consist of the training set while the rest 6 (24%) are the testing set. The aggregate number of all frames from the 6 video recordings in the testing set is 420. Videos in both sets are manually labeled though the 360ObjectAnnotator[10].

6.3 Hardware Environments

Object detection is performed on machines with different hardware environments. In addition to a high-performance server, statistics from a machine of general hardware configuration is also desirable because it can stimulate the real working performance of the Responders-360 framework in the training institutes, where machines with expensive graphics may not be supported. The validation hardware environments include 1) Machine 1: a desktop with a 6-core CPU and a NVIDIA GeForce RTX 3080 Ti graphics (12GB memory), and 2) Machine 2: a laptop with a 4-core CPU and without discrete graphics.

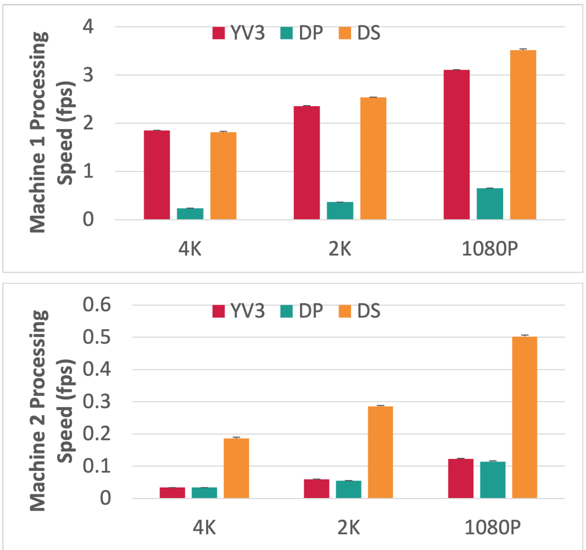

Figure 5: Processing speed of the 3 approaches on two machines over different resolutions.

6.4 Results

The preliminary results on processing speed, GPU memory utilization, and detection accuracy for 3 approaches are reported. Processing speed is measured in the number of frames detected per second (fps). The validation process is iterated for 10 times and the average processing speed is reported. GPU memory utilization is measured in GB. Detection accuracy is measured using the F1 score. The validation is performed on videos of different resolutions.

Speed (see Figure 5). On Machine 1, we observe that our DS approach has optimized the Dual-Projection process with more than 4x speedup under all resolutions. The processing speed of the DS approach is comparable to that of the YV3 approach. For the 1080P resolution, our DS approach is able to offer 360-degree object detection in 3 fps, which can provide a smooth reviewing experience for the instructors. On Machine 2, we notice a significant speedup of our DS approach over the YV3 approach. It is because, without a GPU, neural network-based object inference requires more time and becomes the bottleneck compared to the projection process. Hence, processing time of the YV3 approach has increased relative to the other two approaches. However, our DS approach still greatly improves the detection speed and is able to provide a relatively acceptable throughput for systems without a GPU. The performance is stable in both machines because we observe consistency in processing speed around the average value and low standard deviations.

GPU Memory Utilization (see Figure 6). On Machine 1, projection-based approaches (DP and DS) realize over 25% reduction in GPU memory utilization compared to the non-projection-based approach (YV3) under all resolutions, which makes high resolution object detection accessible to systems with general hardware configuration (e.g., graphics with memory no more than 4GB). The DP approach and the DS approach share a similar GPU memory usage because the same projection procedures have been taken to transform *ERP tiles* to *cubic tiles*. Hence, the *cubic tiles* being detected are of the same size for those two approaches.

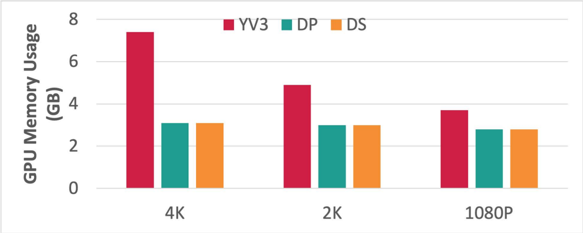

Figure 6: GPU memory usage of the 3 approaches on machine 1 over different resolutions.

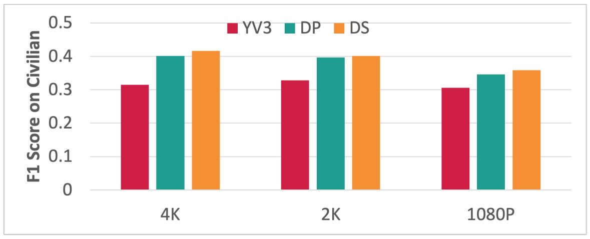

Figure 7: F1 score on civilian of the 3 approaches over different resolutions.

Accuracy (see Figure 7). Due to the limited number of some object categories in the testing set, F1 scores on selected object categories are reported in the preliminary result. We notice that, by reducing the geometric distortion, projection-based approaches (DP and DS) perform better at detecting civilians. More importantly, we also observe a similar detection accuracy between the DP approach and the DS approach, which indicates that the Dynamic Selection algorithm correctly filters out redundant *cubic tiles*.

7 CONCLUSION

In this work, with the aim of improving first responder training, we present the Responders-360 system framework for latency-aware 360-degree object detection. We design and implement a Dynamic Selection algorithm that efficiently optimizes computation resources. Preliminary results demonstrate that our approach (DS) is able to significantly speed up the detection process while retaining the improved performance in GPU memory usage and detection accuracy.

ACKNOWLEDGMENTS

We would like to thank reviewers for their comments and suggestions. We also highly appreciate the support from firefighters and researchers at the Illinois Fire Service Institute (IFSI), especially Dr. Farzaneh Masoud and Dr. Dac Nguyen, for helping us understand firefighter training and collecting the dataset. This work was funded by National Science Foundation grants NSF IIS 2140645 and NSF IIS 2140620.

REFERENCES

[1] Benjamin Coors, Alexandru Paul Condurache, and Andreas Geiger. 2018. Spherenet: Learning spherical representations for detection and classification in omnidirectional images. In *Proceedings of the European conference on computer vision (ECCV)*. 518–533.

[2] Mallesham Dasari, Arani Bhattacharya, Santiago Vargas, Pranjal Sahu, Aruna Balasubramanian, and Samir R. Das. 2020. Streaming 360-Degree Videos Using Super-Resolution. In *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications*. 1977–1986.

[3] Ching-Ling Fan, Jean Lee, Wen-Chih Lo, Chun-Ying Huang, Kuan-Ta Chen, and Cheng-Hsin Hsu. 2017. Fixation Prediction for 360° Video Streaming in Head-Mounted Virtual Reality. In *Proceedings of the 27th Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV'17)*. 67–72.

[4] Xianglong Feng, Viswanathan Swaminathan, and Sheng Wei. 2019. Viewport Prediction for Live 360-Degree Mobile Video Streaming Using User-Content Hybrid Motion Tracking. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 3 (06 2019), 1–22. https://doi.org/10.1145/3328914

[5] Ross Girshick. 2015. Fast R-CNN. In *Proceedings of the IEEE international conference on computer vision*. 1440–1448.

[6] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 580–587.

[7] Zhirui Hu, Peiyan Dong, Zhepeng Wang, Youzuo Lin, Yanzhi Wang, and Weiwen Jiang. 2022. Quantum neural network compression. In *Proceedings of the 41st IEEE/ACM International Conference on Computer-Aided Design*. 1–9.

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

[9] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. 2016. SSD: Single Shot MultiBox Detector. In *European conference on computer vision*. Springer, 21–37.

[10] Anh Nguyen. 2022. 360ObjectAnnotator: a Tool to Annotate Object Bounding Box for 360 Videos. https://github.com/phananh1010/360-object-detection-annotation.

[11] Jounsup Park. 2021. Real-time object detection in 360-degree videos. In *Real-Time Image Processing and Deep Learning 2021*. 99–114.

[12] Feng Qian, Bo Han, Qingyang Xiao, and Vijay Gopalakrishnan. 2018. Flare: Practical Viewport-Adaptive 360-Degree Video Streaming for Mobile Devices. In *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking (MobiCom '18)*. 99–114.

[13] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. 2016. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 779–788.

[14] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems* 28 (2015).

[15] Ayush Sarkar, Anh Nguyen, Zhisheng Yan, and Klara Nahrstedt. 2022. A 360-Degree Video Analytics Service for In-Classroom Firefighter Training. In *2022 Workshop on Cyber Physical Systems for Emergency Response (CPS-ER)*. IEEE, 13–18.

[16] Rabia Shafi, Wan Shuai, and Muhammad Usman Younus. 2020. 360-Degree Video Streaming: A Survey of the State of the Art. *Symmetry* 12, 9 (2020).

[17] Yu-Chuan Su and Kristen Grauman. 2017. Learning spherical convolution for fast features from 360 imagery. *Advances in Neural Information Processing Systems* 30 (2017).

[18] Afshin Taghavi, Aliehsan Samiei, and Ravi Prakash. 2020. Viewport prediction for 360° videos: a clustering approach. 34–39.

[19] Kuan-Hsun Wang and Shang-Hong Lai. 2019. Object detection in curved space for 360-degree camera. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 3642–3646.

[20] Mai Xu, Chen Li, Shanyi Zhang, and Patrick Le Callet. 2020. State-of-the-art in 360 video/image processing: Perception, assessment and compression. *IEEE Journal of Selected Topics in Signal Processing* 14, 1 (2020), 5–26.

[21] Zhisheng Yan and Jun Yi. 2022. Dissecting Latency in 360° Video Camera Sensing Systems. *Sensors* 22, 16 (2022), 6001.

[22] Junhuan Yang, Yi Sheng, Sizhe Zhang, Ruixuan Wang, Kenneth Foreman, Mikell Paige, Xun Jiao, Weiwen Jiang, and Lei Yang. 2022. Automated Architecture Search for Brain-inspired Hyperdimensional Computing. *arXiv preprint arXiv:2202.05827* (2022).

[23] Junhuan Yang, Yi Sheng, Yuzhou Zhang, Weiwen Jiang, and Lei Yang. 2023. On-Device Unsupervised Image Segmentation. *arXiv preprint arXiv:2303.12753* (2023).

[24] Wenyan Yang, Yanlin Qian, Joni-Kristian Kämäräinen, Francesco Cricri, and Lixin Fan. 2018. Object detection in equirectangular panorama. In *2018 24th international conference on pattern recognition (icpr)*. IEEE, 2190–2195.

[25] Abid Yaqoob, Ting Bi, and Gabriel-Miro Muntean. 2020. A survey on adaptive 360 video streaming: Solutions, challenges and opportunities. *IEEE Communications Surveys & Tutorials* 22, 4 (2020), 2801–2838.

[26] Dawen Yu and Shunping Ji. 2019. Grid based spherical cnn for object detection from panoramic images. *Sensors* 19, 11 (2019), 2622.

[27] Alireza Zare, Kashyap Kammachi Sreedhar, Vinod Kumar Malamal Vadakital, Alireza Aminlou, Miska M. Hannuksela, and Moncef Gabbouj. 2016. HEVC-compliant viewport-adaptive streaming of stereoscopic panoramic video. In *2016 Picture Coding Symposium (PCS)*. 1–5.