Determining the Necessary Frame Rate of Video Data for Object Tracking under Accuracy Constraints

Anup Mohan
Intel Corporation
California, USA
anup.mohan@intel.com

Ahmed S. Kaseb
Cairo University
Giza, Egypt
akaseb@eng.cu.edu.eg

Kent W. Gauen, Yung-Hsiang Lu,
Amy R. Reibman, Thomas J. Hacker
Purdue University, Indiana, USA
{gauenk, yunglu,
reibman, tjhacker}@purdue.edu

Abstract

Network cameras, a type of surveillance cameras, generate real-time, versatile, and high quality video content that can be used for applications such as public safety and surveillance. Analyzing high frame rate video streams imposes heavy computing needs and significant loads to the network. High frame rates may not be essential for meeting the accuracy requirements of the analyses. For example, high frame rates may not be required to track cars inside a garage compared with cars on a highway. In this paper, we study object tracking and propose a method to automatically determine the necessary frame rate for videos in network cameras for object tracking and adapt to run-time conditions. We demonstrate that the frame rates can be reduced up to 80% based on accuracy constraints.

1. Introduction

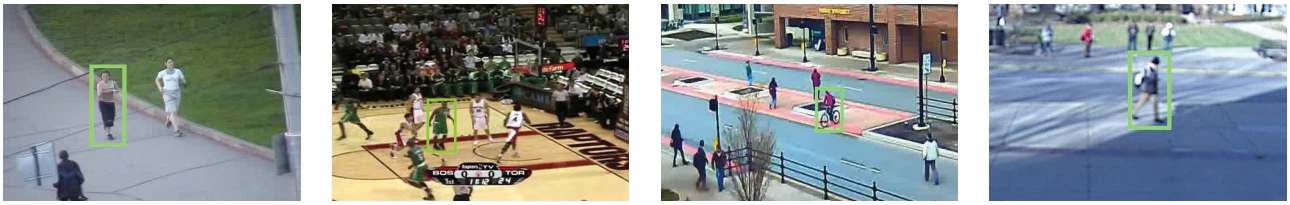
The majority of multimedia content on the Internet is comprised of images and videos. There are more than 1 billion smartphones and 245 million surveillance cameras [7] around the world forming the *Internet of Video Things*. Network cameras, a type of surveillance cameras, are of particular interest as they generate continuous real-time video data with rich and versatile content. Video data from network cameras are used for many applications such as improving public safety [2] and surveillance [8].

Some network cameras are capable of generating high quality videos with high resolutions (10MP) and frame rates (30 frames per second, FPS). Analyzing high quality video imposes heavy computing needs and transmitting this data adds significant loads to the network [1]. Surveillance cameras can generate data in the range of Exabytes [1] and saving this information can be expensive. Cloud computing with on-demand pricing is preferred to meet the computational and storage requirements. Offloading all the video data to the cloud is not feasible as a metropolitan area network can support only tens of thousands of video streams [9]. The commonly used dynamic resource provisioning methods are not efficient to analyze these high quality video streams because these methods are dependent on resource utilizations (e.g., CPU and memory) [6]. The utilizations are high while analyzing high quality video data.

The frame rate is an important aspect of video quality, and high frame rates may not always be essential to meet the accuracy requirements of the applications. For example, if the application is to track cars, then a higher frame rate will be necessary for cars on highways due to fast motion, compared with cars inside parking garages. This motivates the need for content-aware resource provisioning systems for analyzing video streams, where the resource management decisions are based on the content, and the quality of the data can be controlled by the system. An essential aspect of such systems is to determine the necessary video quality based on the content. Determining the necessary quality for millions of network cameras is not easy, as the content generated by each network camera is different. This paper focuses on determining the necessary frame rate. Frame rate, measured in FPS, impacts the amount of data being transmitted and analyzed, and therefore impacts the bandwidth usage and the costs for analyzing and storing the video data.

This paper proposes a method to automatically determine the necessary frame rate for object tracking on video streams from network cameras based on the content, and adapt based on the run-time conditions. The necessary frame rate is determined under accuracy constraints. We select object tracking, as it is an essential component for many applications, and use a state of the art tracking method based on hierarchical convolutional features [5]. The video data for our experiments are obtained from two network cameras and the visual tracker dataset [10]. We use the *precision rate*, which determines the error in distance between the tracked object and the ground truth, to measure the accuracy. Based on the accuracy constraint (e.g., precision rate $> 60\%$) our method can determine the necessary frame rate. We use the displacement of the object being tracked in the video to determine the necessary frame rate. The main contributions of this paper are as follows.

0-7695-6354-6/18/$31.00 ©2018 IEEE
DOI 10.1109/MIPR.2018.00081



(a) Jogging (b) Basketball (c) Cyclist (d) Pedestrian

Figure 1: Sample videos used in the experiments; The jogging and basketball sequences are part of visual tracker dataset; The cyclist and pedestrian sequences are obtained from two network cameras;

- Proposing a method to automatically determine the necessary frame rate for object tracking on videos from network cameras and adaptively adjust to the run-time conditions.
- Studying the effects of the displacement of the object on tracking accuracy and using the displacement to determine the necessary frame rate.
- Demonstrating that necessary frame rates are dependent on the video content and can be reduced up to 80% under accuracy constraints.

Korshunov et al. [3, 4] describe methods to determine the critical video quality for face detection, tracking, and recognition applications with a goal of reducing the utilization of the network bandwidth. Our paper is different from their work as we dynamically adjust the frame rate based on the content. Korshunov et al. modify the compression mechanism in cameras to reduce the frame rates and resolutions. It is not possible to change the compression of network cameras as they are owned by different organizations. Hence the frame rates are controlled at the receiving end (e.g., cloud).

2. Factors Affecting the Necessary Frame Rate

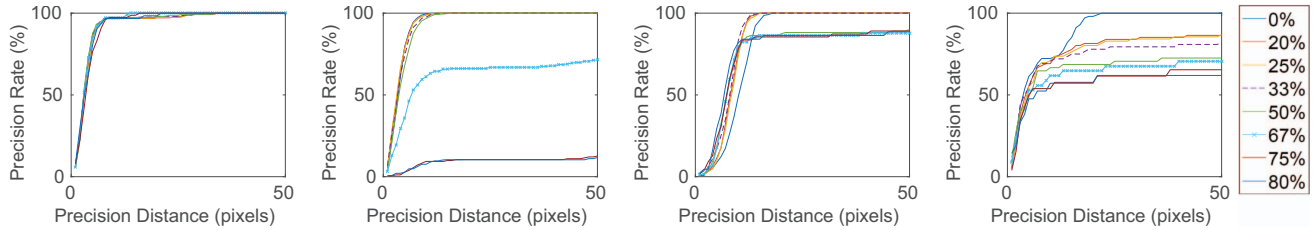
The major factor affecting the necessary frame rate for object tracking is the video content. There are different characteristics of the video content that determine the necessary frame rate such as the speed of motion of the object, direction of motion, and occlusion by other objects. To simplify our problem, we focus only on the speed of motion of the object as it is directly related to the frame rate. When the frame rate is decreased by dropping frames, the perceived speed of motion among the sequential frames can increase. As the distance the object moves increases, the tracker may lose track of the object. We study single object tracking, as multiple objects have different characteristics and determining the necessary frame rate becomes challenging. Usually, a good object detector precedes the tracker. For our experiments, we manually initialize the tracker with the position of the object to be tracked.

We use the hierarchical convolutional features based method of Ma et al. [5] as it is reported [10] to have better accuracy and more robust than the other popular tracking algorithms. Videos from the visual tracking dataset [10] and video sequences obtained from network cameras are used for the study. Figures 1(a) and (b) show sample images from the visual tracking dataset. Figures 1(c) and (d) show sample network camera images. In the visual tracker dataset, the camera follows the object being tracked as opposed to the stationary network cameras.

3. Necessary Frame Rates for Object Tracking on Videos from Network Cameras

Since the information collected by the network cameras are different, the accuracy of the application will highly depend on the content of the video data. Therefore the necessary frame rates will also rely on the content. We propose running a *test phase* where the necessary frame rate for each network camera is determined by saving the video data for a specified duration. During the execution phase, the videos are analyzed at the necessary frame rate and are adjusted at run-time based on the content.

For object tracking applications, we measure the speed of motion using the displacement of the object defined as the distance the object moves between frames, measured in pixels. Assuming that the object moves in one direction (e.g., forward), and the camera is stationary, the displacement of the object increases as the frame rate is reduced. When the object displacement increases, the tracker may miss the object because the tracker may not have estimated the object to have a higher displacement. The tracker has a search window within which the tracker estimates the position of the object. The search window is dependent on the size of the object and the frame size and is automatically calculated by the tracker. When frames are dropped, if the object moves out of the search window, the tracker will miss the object. Therefore, during the test phase, object tracking is performed on the saved video, and the maximum displacement of the object is calculated. If the maximum displacement is below a threshold value (*low displacement threshold*), the frames are dropped from the saved video to reduce the frame rate, and the maximum displacement is calculated. This process is repeated for different frame rates, and the lowest frame rate for which the maximum



(a) Jogging (b) Basketball (c) Cyclist (d) Pedestrian

Figure 2: The precision rates for different precision distances on 4 specific video sequences. The first two plots are for the visual tracker videos and the next two are for the network camera videos. The different curves are obtained by dropping frames up to 80%. For Jogging and cyclist sequences, the frame rate can be dropped up to 80% for precision rates greater than 80%. For basketball and pedestrian sequences, the frame rate can be dropped up to 50% and 25% respectively for precision rates greater than 80%. The data suggests that the necessary frame rates depend on the video content.

displacement is less than the threshold is selected. Considering the maximum displacement ensures that the tracker doesn't miss the object when the frame rate is reduced. If the maximum displacement of the saved video at the initial frame rate is above a threshold (*high displacement threshold*), then the frame rate is not reduced. Frame rates can be adaptively modified using the same method. During the execution phase, if the maximum displacement goes above the *high displacement threshold* or below the *low displacement threshold*, the frame rate can be increased or decreased respectively. The thresholds are determined based on the search window size of the tracking method as explained in Section 4. The video used in the test phase needs to be a good representative of the real-time scenario.

4. Determining Frame Rate under Accuracy Constraints

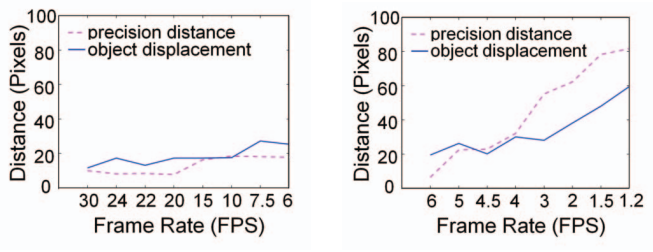
This section proposes a method to determine the necessary frame rates automatically under accuracy constraints and evaluates it. We control the frame rate by dropping some frames at the receiving end (e.g., in the cloud instance). Even though some cameras allow configuring the frame rate, we drop frames at the receiving end to be consistent across all the cameras. Since the frame rate supported by network cameras are different, we use the *frame drop rate* to represent frame rates consistently. For example, a frame drop rate of 0% is equivalent to the original frame rate (e.g., 30FPS). A frame drop rate of 50% is achieved by dropping alternate frames and reduces the frame rate by half (e.g., 15FPS). Note that a frame drop rate of 0% may represent two different frame rates based on the network camera. This is acceptable as we determine the necessary frame rate of network cameras independently.

The accuracy of the tracking algorithm is determined using *precision distance* and *precision rate* [5]. The precision distance is defined as the Euclidean distance between the center point of the tracked object and that of the ground truth. The precision rate is defined as the percentage of frames in the sequence for which the precision distance is below a selected threshold. Low precision distances and high precision rates are desired.

4.1. Frame Rate and Precision Rate

Figure 2 shows the precision rate as a function of the precision distance. A commonly used threshold for precision distance is 20 pixels [10]. Figures 2 (a) and (b) show the precision rate results for the *jogging* and *basketball* video sequences respectively. For the jogging sequence, the frame rate can be reduced by 80% without decreasing the precision rate. Since the camera follows the person being tracked in the jogging sequence, the displacement does not increase with reducing frame rate and the frame rates can be dropped aggressively. For the basketball sequence, the frame rate can be reduced up to 50% without decreasing the precision rate. In the basketball sequence, the player moves at high speed with sudden changes in directions and is often occluded by the other players. Therefore, the tracking error increases as the frame rate reduces.

Figures 2 (c) and (d) shows the precision rate results for the *cyclist* and *pedestrian* network camera sequences respectively. The network camera sequence tracking a cyclist has a frame rate of 30 FPS (0%). The frame rate can be reduced by 33% (20 FPS) without decreasing the precision rate. The tracking error increases when the frame rate is reduced by more than 33% because the displacement of the cyclist increases. The frame rate of the cyclist sequence can be reduced by 80% (6 FPS) for precision rates above 80%. The pedestrian sequence has a frame rate of 6 FPS (0%). Accuracy decreases for this sequence as the frame rate is reduced because the displacement at the original frame rate is high. The frame rate of the pedestrian sequence can be reduced by 80% (1.2 FPS) for precision rates above 60%.



(a) (b)

Figure 3: Relationship between frame rates, maximum displacement, and precision distance for; (a) cyclist sequence; and (b) pedestrian sequence. When the maximum displacement is higher than 20 pixels, reducing the frame rates result in higher precision distances.

4.2. Object Displacement and Precision Distance

Since the network cameras are stationary, the displacement of the objects may increase as the frame rate is reduced. The cameras used for generating videos in the visual tracker dataset follow the objects being tracked and the effect of displacement is less critical for these videos. Hence, we only consider the network camera video sequences in this section. Figure 3 shows the maximum object displacement and precision distance for the network camera sequences at different frame rates.

For both the video sequences, the object does not change directions as it moves. The search window size determined by the tracker for the cyclist and pedestrian sequences are 55×36 pixels and 35×28 pixels respectively. We will only consider the smallest dimension to make sure the object doesn't move out of the search window. In this case, the smallest dimension is the search window width. The displacement for the cyclist sequence does not increase beyond the search window width of 36 pixels as the frame rate is reduced, and the precision distance doesn't increase significantly thereby resulting in precision rates greater than 80%. It can be noted that the displacement of the cyclist increases beyond 20 pixels when the frame rate is reduced by more than 33%, resulting in a small increase in precision distance. Referring to Figure 2, the precision rate reduces for frame drop rates greater than 33%. The displacement for the pedestrian sequence is greater than the search window width (28 pixels) for frame rates lesser than 4 FPS (33%), causing a significant increase in the precision distance. Referring to Figure 2, the precision rates reduce below 80% for frame drop rates greater than 33%.

Section 3 requires selecting a *low displacement threshold* and *high displacement threshold* based on the content. The results in this section suggest that the search window width with an offset can be used as the *high displacement threshold* as the tracker will miss the object having a displacement greater than the search window width. The *low displacement threshold* may be calculated from the search window width by applying a safe margin such as 50%. For example, in case of the cyclist sequence, the low and high displacement thresholds are 18 and 26 (offset of 10 pixels) pixels respectively. Note that the safe margin of 50% and offset of 10 pixels are estimated values and the sensitivity of these parameters should be carefully studied.

5. Conclusion

This paper determines the necessary frame rate for object tracking under accuracy constraints based on the content. It is observed that high frame rates may not always be necessary. We reduce the frame rate by 80% for precision rates higher than 60%. A method to automatically select the frame rate for network cameras and adapt based on the content is proposed, and more investigation needs to be done for implementing this method. We would like to extend this study by considering multiple object tracking and by determining the necessary video resolution.

The authors would like to thank the organizations that provide the camera data (complete list at https://www.cam2project.net/ack/). This project is supported in part by National Science Foundation ACI-1535108. Any opinions, findings, conclusions, and recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the sponsors.

References

[1] Cisco visual networking index: Forecast and methodology, 2015–2020. *CISCO White paper*, 2016.

[2] Y. Koh et al. Improve safety using public network cameras. In *IEEE Symposium on Technologies for Homeland Security*, pages 1–5, 2016.

[3] P. Korshunov and W. T. Ooi. Critical video quality for distributed automated video surveillance. In *ACM international conference on Multimedia*, pages 151–160, 2005.

[4] P. Korshunov and W. T. Ooi. Video quality for face detection, recognition, and tracking. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 7(3):14, 2011.

[5] C. Ma et al. Hierarchical convolutional features for visual tracking. In *IEEE International Conference on Computer Vision*, pages 3074–3082, 2015.

[6] A. Mohan. *Cloud Resource Management for Big Visual Data Analysis from Globally Distributed Network Cameras*. PhD thesis, Purdue University, 2017.

[7] A. Mohan et al. Internet of video things in 2030: A world with many cameras. In *IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–4, 2017.

[8] Regazzoni et al. Video analytics for surveillance: Theory and practice. *IEEE Signal Processing Magazine*, 27(5):16–17, 2010.

[9] M. Satyanarayanan et al. Edge analytics in the internet of things. *IEEE Pervasive Computing*, 14(2):24–31, 2015.

[10] Y. Wu et al. Online object tracking: A benchmark. In *IEEE conference on computer vision and pattern recognition*, pages 2411–2418, 2013.