A Video Optimization Framework for Tracking Teachers in the Classroom

Lele Ma
College of William and Mary
lma03@email.wm.edu

Yantao Li
Southwest University
yantaoli@swu.edu.cn

Gang Zhou
College of William and Mary
gzhou@cs.wm.edu

ABSTRACT

Classroom video capturing is an extremely powerful pedagogical tool to support student education. Due to teachers always moving around the blackboard and facing in different directions, how to make the video with the best view is becoming a hard mission. In this paper, we present a video optimization framework, *OptVideo*, that optimizes videos from different directions of the classroom into one single and complete video with the best view. More specifically, *OptVideo* can split videos into multiple video clips, select video clips with the best view from videos captured on different directions, and combine them into a complete video with time order, by exploiting the face detection and motion tracking methods. We design an easy-to-use interface for users to interact with the post-processing of the video, which provides video split and combination recommendations according to the analysis of videos. The interface also allows users to select the best view of the videos and publish the final video by themselves. We evaluate the performance of *OptVideo* in terms of the accuracy of the face detection and motion tracking, and the processing speed. The evaluation results show that *OptVideo* achieves the correct ratio with the face detection algorithm up to 100% and the process speed up to 250.11 ms/frame on a desktop.

KEYWORDS

Video optimization, face detection, motion tracking, GUI

ACM Reference Format:
Lele Ma, Yantao Li, and Gang Zhou. 2018. A Video Optimization Framework for Tracking Teachers in the Classroom. In *Proceedings of the Second ACSIC Symposium on Frontiers in Computing (SOFC'18).* ACM, New York, NY, USA, 5 pages. https://doi.org/10.475/123_4

1 INTRODUCTION

For student education, video capturing in classrooms is an extremely powerful pedagogical tool. However, the current technical hurdles that must be overcome are often enough to discourage teachers from using it [1]. Traditional methods of video capturing are hard to use, which require lots of human efforts on video capturing or post-processing, and cost a lot on equipments or human efforts. Automatically video capturing in a classroom and automatical post-processing can save lots of human efforts as well as the budgets.

SOFC'18, June 1-2, 2018, Dallas, Texas USA

ACM ISBN 123-4567-24-567/08/06.
https://doi.org/10.475/123_4

There are some products available in the market, such as Beyond AutoTracker [2] and Swivl [3]. Beyond AutoTracker is a HD automated tracking camera which can automatically track a person without a camera operator. However, it can track any moving object who is probably not the targeted object and it cannot capture the face of the object who is facing back to the camera. Swivl is a robot that allows users to capture and share video using their own tablet or smartphone. But it requires the users to wear markers when the video is being captured, which is pretty obtrusive. Another thing is that the tracking speed via the wireless signal is relatively slow, and the angle of the robot rotating and tracking is also limited, which means if the object walks too fast or walks far away around the classroom, it probably loses track of the object. In addition, there are still other challenges that are not addressed by the current state-of-the-art products.

The first challenge is how to detect and track the teachers from different points of view in the classroom. The teachers can walk around the classroom and change the directions of their faces freely during the class. A single stable camera cannot capture the face and movement of the teachers effectively. Even a hired photographer cannot capture the teacher's face effectively by controlling the direction of a stable camera. To address this challenge, we setup multiple stable cameras in different positions in a classroom. As for the teacher detection and tracking, it leverages video analysis technique of face detection and motion tracking to detect and track the teacher in each videos.

The second challenge is how to automatically post-process the captured videos. The captured videos could include many poor or out-of-focus pieces of video clips, which take an adverse effect to the video. To improve the quality of the videos, many human efforts will be taken and much resource will be cost. To address the second one, we propose to post-process the video automatically and publish the videos without much human efforts. In order to minimize the human interaction and do all the stuff automatically, based on the first step of detection and tracking, we split each video into multiple pieces of video clips. For each time interval, we compare the clips from different cameras and choose the best clip. Then, we combine all the best clips into a single video that is ready to published. Additionally, if the users do not feel satisfied with the best clips the system has chosen, the system also provides easy-to-use graphic user interface which allows users to replace any of the clip before combining them to the final video.

In this paper, we propose a video optimization framework, called *OptVideo*, which detects the teachers' faces and tracks the movements by multiple videos from different points of view in a classroom without human interactions. *OptVideo* optimizes videos from different directions of the classroom into one single complete video with the best view by automatically selecting video clips with the

best view in different directions and combining them into a complete video. We also present an easy-to-use interface for users to interact with the post-processing of the videos, which provides video split and combination recommendations according to the analysis of videos by filtering out all the video pieces containing the out-of-focus teachers or other scenarios we are not interested. The interface can also be operated manually by users to select the best view of the videos and publish the final video by themselves. We evaluate the performance of *OptVideo* in terms of the accuracy of the face detection and motion tracking, and the processing speed. The evaluation results show that *OptVideo* achieves the correct ratio with the face detection algorithm up to 100% and the process speed up to 250.11 ms/frame on a desktop.

The main contribution of this work can be summarized as:

• We design a video optimization framework, *OptVideo*, that optimizes videos from different directions of the classroom into one complete single video with the best view. By exploiting the face detection and motion tracking methods, *OptVideo* can automatically select video clips with the best view from videos on different directions and combine them into a complete video.

• We propose an easy-to-use interface for users to interact with the post-processing of the video, which provides video split and combination recommendations according to the analysis of videos. Moreover, it provides an interface for users to select the best view of the videos and publish the final video by themselves.

• We evaluate the performance of *OptVideo* in terms of the accuracy of the face detection and motion tracking, and the processing speed. The evaluation results show that *OptVideo* achieves the correct ratio with the face detection algorithm up to 100% and the process speed up to 250.11 ms/frame on a desktop.

The rest of the paper is organized as follows. Some basic background about video analysis techniques in computer vision is introduced in Sec. 2. We detail the architecture of *OptVideo* in Sec. 3. In Sec. 4, we evaluate the performance of *OptVideo* in terms of the accuracy of the face detection and motion tracking, and the processing speed. We review the related work in Sec. 5 and discuss the future work in Sec. 6. We finally conclude this work in Sec. 7.

2 BACKGROUND

In this section, we introduce the background of the object detection based on HAAR features, object tracking via Lucas-Kanade algorithm, and OpenCV Library, respectively.

2.1 Object detection based on HAAR features

Haar-like features are digital image features used in object recognition [4]. One of the most popular algorithms based on HAAR features is the Viola-Jones object detection [5]. It is the first object detection algorithm to provide competitive object detection rates in real-time [6]. In the detection phase of the Viola-Jones object detection framework, a window of the target size is moved over the input image, and for each subsection of the image the Haar-like feature is calculated. This difference is then compared to a learned threshold that separates non-objects from objects. Because such a Haar-like feature is only a weak learner or classifier (its detection quality is slightly better than random guessing), a large number of Haar-like features are necessary to describe an object with sufficient accuracy. In the Viola-Jones object detection framework, the Haar-like features are therefore organized in something called a classifier cascade to form a strong learner or classifier [4].

The key advantage of a Haar-like feature over most other features is its calculation speed. A Haar-like feature of any size can be calculated in constant time (approximately 60 microprocessor instructions for a 2-rectangle feature) [4]. HAAR features could be used to detect a various kinds of objects like human eyes and faces, human bodies or cat faces, cat bodies, etc. In this paper, we will use HAAR features of human frontal faces to detect faces in a video.

2.2 Object tracking via Lucas-Kanade algorithm

The Lucas-Kanade method is a widely used method for optical flow estimation in computer vision [7] [8]. The optical flow is the pattern of apparent motion of image objects between two consecutive frames caused by the movement of object or camera [9]. It is 2D vector field where each vector is a displacement vector showing the movement of points from first frame to second. From user point of view, the idea is simple, given some points, the algorithm could compute the optical flow vectors of those points so that we can track the given points via those vectors. In this paper, we combine Lucas-Kanade optical flow tracking and face detection algorithm.

2.3 OpenCV Library

OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library [10]. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. Being a BSD-licensed product, OpenCV makes it easy for businesses to utilize and modify the code. The library includes a comprehensive set of computer vision and machine learning algorithms, including the algorithms we mentioned above. So that in this paper , we use these algorithms in OpenCV to detect faces, track moving objects, and finally find the teacher's face and motion in a classroom video.

3 SYSTEM DESIGN

The goal of the system design is to capture the teacher from different point of views and reduce the human efforts in post-processing the videos. First, we get multiple videos by placing multiple cameras around the classroom in order to capture the teacher from different points of view. Then, after we captured the videos, we analyse the video via the face detection and motion tracking. Based on the detected faces and motions, *OptVideo* filters all the video pieces that contain the teachers or other scenarios that we are interested in. At last, after finishing the video processing successfully, the Graphical User Interface of *OptVideo* allows the user to do a final check on the post-processing result. The architecture of *OptVideo* is shown in Fig. 1, which is composed of modules of face detection, motion tracking, filters, and graphical user interface.

3.1 Face detection and motion tracking

Among all the video pieces in the same time interval, we find the best video pieces that contain the best view of the teacher. That is,

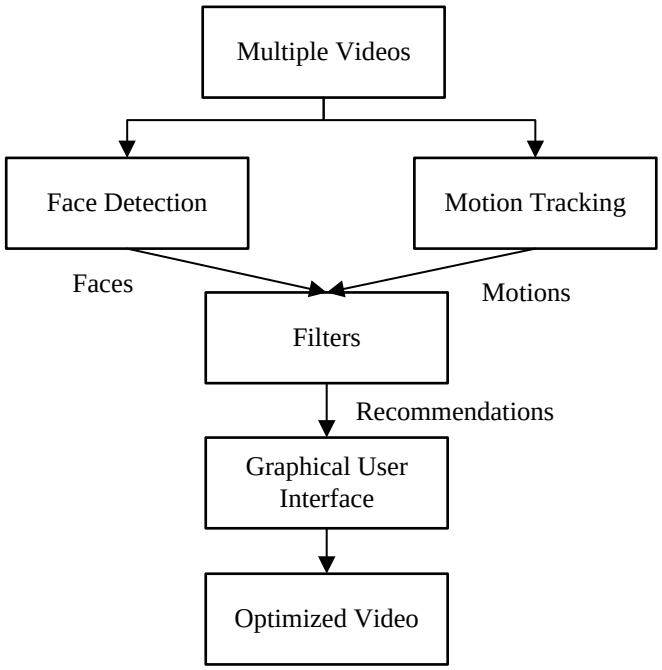


Figure 1: Architecture of *OptVideo*

the video includes the teacher's front face. We exploit two different technologies to collect the data for analysis: face detection and motion tracking. Face detection can find the faces in each frame of the video with certain accuracy. It will give us all the possible rectangle areas which matches the face features. The face features are based on HAAR features as we mentioned in Sec. 2.1. It is trained from various positive face images and also trained from various negative images which contain no faces. All the trained pictures are under certain fixed sizes and resolution. So that, when we use the trained features, it is critical to get a reasonable size of the video frame. Because only the faces that have similar size with the training data will be detected as a face. Therefore, all the video frames should be transformed to proper size before they are passed to the video analysis algorithm. Motion tracking can track a given pixels and get the data of the movements, such as move distance, or trajectory length, etc. The work is based on Lucas-Kanade Optical Flow algorithm as mentioned in Sec. 2.2, which can track a given pixels based on its surroundings.

The system can find all the moving points in a video frame and track them. Then, once we have faces detected, we re-initialize the tracking points again and track all the points. The moving points may be in or out of the faces, but all the moving data and face data can be collected in order to do further filtering.

3.2 Filters

Since both the face detection and motion tracking have certain errors, we exploit the Filter to detect which face is teacher's and which motion is teacher's movement by combining and comparing the two sets of data.

Before we conduct filtering, we have two assumptions: 1) if it has a very high possibility that a human object has been moving for the longest time during class among all the objects in a video, then the human object is the teacher; 2) For certain amount of time (such as 30 seconds or 1 minute), if an object is always moving, it has a high probability that the object is a teacher. If an object is moving less than that time threshold, it has a lower possibility that it is a teacher.

Based on the above assumptions, we use the six matrices to determine where the teacher is and how to cut the videos into pieces:

- Trajectory: when tracking each pixels in the video, we record each object's trajectory and compute their trajectory distances. It will be used as an important evaluation data which helps us to recognize the teacher.
- Movement Threshold: in order to avoid noise in the video, any movements that move under a certain speed is ignored.
- Trajectory Threshold: in order to avoid movement noise in the video, the pixels whose movement have relatively high speed but no further than a threshold are considered as stable pixels.
- Face Size Threshold: Given the exact resolution of the camera and the possible distance change between the teacher and camera, the biggest face size as well as the smallest face size in terms of pixels can be set. It is possible that all the faces that are out of the defined scope are ignored.
- Movement Counting: Count how many times one object has been moving during a class. This can be useful under the assumption that a teacher could be the one that mostly possible to be moving during the class.
- Moving Time Weights: The moving time weights can be computed based on how long time the action of moving continues in a sliding time window. If the object is moving all the time during a 30-second or 60-second time window, it has high weight, which means it has a high possibility to be a teacher.

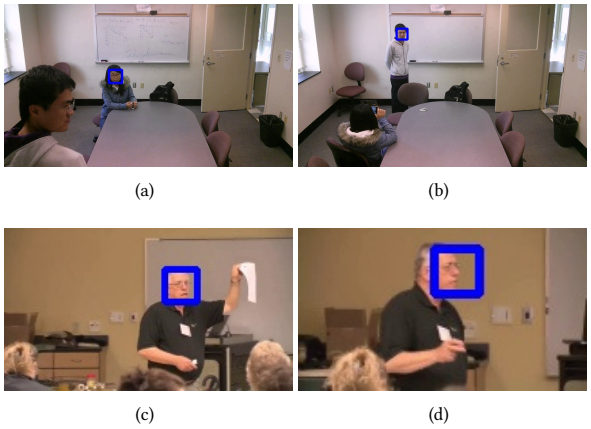
3.3 Graphical user interface

After finishing the video processing successfully, the Graphical User Interface of the system will show all the video pieces from different point of views and recommend the best view to users. Each recommendation will be the video frame that contains the teacher it detected automatically. The only thing the user is supposed to check is to see whether the recommendation frame has a teacher or not. If the frame has the teacher being detected correctly, then the whole video pieces will be correct.

On the other hand, for each time interval of the video pieces, the system allows users to choose which one is the best view by their own, regardless what the recommendation is. So if the system has a wrong recommendation, the user could choose the right video clip.

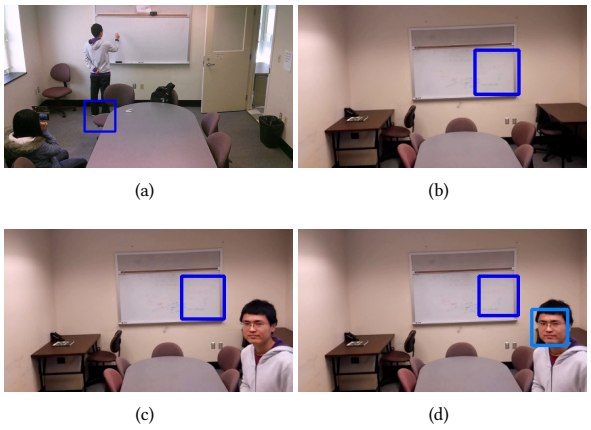
4 EVALUATION

In this section, we evaluate the performance of *OptVideo* in terms of the accuracy of the face detection and motion tracking, and the processing speed.



(a) (b) (c) (d)

Figure 2: True Positive Faces Detected



(a) (b) (c) (d)

Figure 3: False Positive Results

4.1 Accuracy of the face detection and motion tracking

For face detection algorithm, most of time it can detect all the faces in each frame of the video, regardless who the teacher is. The true positive is shown in Fig. 2. Some of the test videos are download from [11]. We can see that it can detect the front face for all the human faces in a video, which is just what we want.

However, face detection algorithms are easy to be affected by multiple factors. Some factors are related to the video itself, such as the light of the video, and the resolution of the video. Some factors can be related to the algorithm we use, such as how the HAAR features are trained, and how the parameters of the algoritms are set. Due to a lot of factors, the face detection can result in uncertain results occasionally, as shown in Fig. 3. It is easy to see that the HAAR feature detects some stable area that is certainly not related to human face at all. However, the common features are that they are stable, so that by combining the motion tracking algorithm, it can be easy to be filtered out.

Table 1: Correct Ratio on Video Clip1: rm139back.avi. Resolution: 1920×1080, length: 206 seconds.

Cascade (Trained Features)	Parameter Setting 1 (Scale=$\frac{1}{3}$)	Parameter Setting 2 (Scale=$\frac{1}{4}$)
HAAR_default	0.58	0.21
HAAR_alt	0.99	0.98

Table 2: Correct Ratio on Video Clip2: rm139clip.avi. Resolution: 1280×720, length: 45 seconds.

Cascade (Trained Features)	Parameter Setting 1 (Scale=$\frac{1}{3}$)	Parameter Setting 2 (Scale=$\frac{1}{4}$)
HAAR_default	1.00	0.96
HAAR_alt	0.15	0.08

Table 3: Correct Ratio on Video Clip3: classclip.avi. Resolution: 640×360, length: 59 seconds.

Cascade (Trained Features)	Parameter Setting 1 (Scale=$\frac{1}{3}$)	Parameter Setting 2 (Scale=$\frac{1}{4}$)
HAAR_default	0.99	1.00
HAAR_alt	1.00	0.98

Table 4: Process Speed on Different Platforms

Picture Size	Desktop (ms/frame)	Laptop (ms/frame)
Original (1920×1080)	121.31	250.11
Scaled by 1/4	33.86	102.73

Tables 1, 2, and 3 show the correct ratio of the face detection algorithm. It is apparent that the accuracy is high related to certain videos and the trained features as well as the parameters we set.

4.2 Processing speed

In order to evaluate the speed of the processing, we set up two environments to run the same workloads, which are set as:

Desktop: 3.00GHz Intel Core TM 2 Duo, 4G Memory

Laptop: 2.10GHz Intel(R) Core TM 2 Duo, 2G Memory

The results are shown in Table 4. We can see that on a desktop, the speed is very near to the playback speed of the video, which means it has high potential to do online analysis. Note that the normal playback speed is 33.33 ms/frame.

5 RELATED WORK

There are some works focusing on video capturing, and we discuss the related works in classroom video capturing devices and the object detection and tracking, respectively.

5.1 Classroom video capturing devices

Beyond AutoTracker [2] is a HD automated tracking camera which can automatically track a person without a camera operator. This product also utilizes image recognition and motion detection to find and track the subject. It can also auto pan tilts and zooms as it

tracks the subject. The person being tracked doesn't need to wear a lanyard or device of any type. The camera can lock the moving person in the center of the image and track even if the subject turns around or stands still. However, it could track any moving object who is probably not the teacher, which means when we have some students that are moving around, the machine also tracks the moving student. On the other hand, the camera cannot move which means when the teacher is facing back to the camera, it cannot capture the teacher's face. Last but not least, the price of the product is now around $4,995, which is relatively high cost for being widely deployed in schools.

Another tool that can be used for classroom video capturing is Swivl[3]. Swivl is a robot that allows users to capture and share video using their own tablet or smartphone. Swivl Robot can wirelessly follow the movement of the marker and capturing high quality audio at the same time by the marker. However, the user must wear the marker when the video is being captured, which is pretty obtrusive. The tracking speed via the wireless signal is relatively slow, and the angle of the robot can rotate and track is also limited, which means if the teacher walks too fast or walks far away around the classroom, it probably lose the tracking of the teacher. Moreover, the camera also cannot move around to capture the teacher from different points of view. The last thing is the cost. Although its price is around $399, a relatively cheaper price than the above Beyond AutoTracker, it requires the teacher to install a smartphone or tablet to the robot. So overall, the combined cost and gain are still not so competitive.

5.2 Object detection and tracking

One popular method to detect object is the Haar cascade classifier which is originally designed for face detection, which is introduced and evaluated in [12]. The Haar cascade classifier can achieve a high speed for low resolution videos, which makes it a proper choice for online video analysis.

There are plenty of other algorithms that can detect the human body or moving body in a video, one of which is the algorithm based on the "Histograms of Oriented Gradients" [13]. It can detect the human body and can be used to detect pedestrians on the road, but the speed of it is much slower than the HAAR based algorithms.

6 FUTURE WORK

Due to the limited time, the prototype system only implements the core face detection and motion tracking part of the system. The filters as well as the graphical user interface are left for further development. Also, for further development of the whole system in order to publish it as a product, there are plenty of improvement work to do, such as the accuracy improvement by introducing feature training, or platform improvement by introducing various versions of the program across mobile and desktop platforms.

6.1 Implement the filters: decide where the teacher is

The first thing needs to be done is to get trajectory of any moving objects in the video as well as the other matrices we defined above. So we can filter the results from face detection and motion detection stages.

6.2 Evaluate the usability of the user interface

We need to evaluate how easy it is to use the system. It can be evaluated based on how many video clips it produces and how accurate the results are. Intuitively, the more clips it can produce, the more attention will be needed from the users to explore and determine which video clips should be used in the final video. Additionally, the more accurate the video clips are cut, the less efforts the users will take.

6.3 Computing platform

Finally, this system can be ported to support both on the desktop to do post-processing and also on the mobile devices to do online video analysis. Once online video analysis is deployed, more features will be available for the system, such as auto-focus on the teacher when capturing the video.

ACKNOWLEDGEMENT

This work was supported in part by U.S. National Science Foundation under grants CNS-1253506 (CAREER).

7 CONCLUSION

The face detection and motion tracking method can effectively detect and tracking the teacher in a certain accuracy. However, none of them can be used stand alone to detect which object is the real teacher. We need to combine those two methods to collect data and define our own matrices to detect where the teacher is and capture the teacher's face in multiple videos. The filter of the system is the key point that is related to the accuracy of the automatic processing of the videos. The easy way to use graphical interface enables users to post-process the video with as less efforts as possible. We evaluate the performance of *OptVideo* in terms of the accuracy of the face detection and motion tracking, and the processing speed. The evaluation results show that *OptVideo* achieves the correct ratio with the face detection algorithm up to 100% and the process speed up to 250.11 ms/frame on a desktop.

REFERENCES

[1] K. Romeo. Classroom video capture needs. [Online]. Available: https://web.stanford.edu/group/ats/cgi-bin/hivetalkin/?p=2487
[2] Autotracker. [Online]. Available: http://1beyond.com/store/autotracker-camera
[3] Swivl. [Online]. Available: https://www.swivl.com/
[4] Wikipedia. Haar-like features. [Online]. Available: https://en.wikipedia.org/wiki/Haar-like features
[5] P. Viola and M. Jones. 2004. Robust real-time object detection. *Int. J. Computer Vision*, vol. 57, no. 2, pp. 137-154, 2004.
[6] Wikipedia. Violajones object detection framework. [Online]. Available: https://en.wikipedia.org/wiki/Viola–Jones object detection framework
[7] B. D. Lucas, and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Pro. the 7th Int. Joint Conf. Artificial intelligence*, Vancouver, BC, Canada, pp. 24-28, Aug., 1981.
[8] Wikipedia. Lucaskanade method. [Online]. Available: https://en.wikipedia.org/wiki/Lucas-Kanade method
[9] OpenCV. Optical flow. [Online]. Available: http://docs.opencv.org/3.1.0/d7/d8b/tutorial py lucas kanade.html#gsc.tab=0
[10] Opencv. [Online]. Available: http://opencv.org/about.html
[11] National agrability, tools and techniques for small acre gardening. [Online]. Available: http://docs.opencv.org/3.1.0/d7/d8b/tutorial_pylucas-kanade.html#gsc.tab=0
[12] R. Padilla, C. Costa Filho, and M. Costa. 2012. Evaluation of haar cascade classifiers designed for face detection. *J. WASET*, vol. 64, pp.362-365, 2012.
[13] N. Dalal and B. Triggs. 2005. Histograms of oriented gradients for human detection. In *Pro. IEEE Con. Computer Vision and Pattern Recognition*, San Diego, CA, USA, pp. 886-893, 2005.