Removing tourists from videos

Yuanchen Liu and Qianyun Ma
University of Wisconsin, Madison
Fall 2015
Website: http://www.yuanchen-liu.com/cs534_project.html

Abstract

The idea of this project is similar to Adobe's 'Monument Mode' on Adobe Max 2015 Conference. In the project, in order to distinguish moving objects (such as tourists and cars) from backgrounds or sightseeing, we implement People Detector, Foreground Detector and Optical Flow methods, and develop our own algorithm of Median Method to find and remove moving objects from sightseeing in real-time videos.

1. Introduction and Motivation

It may be difficult to take a picture only contains some landmark buildings if there are too many tourists around it. Although removing tourists can be done by Photoshop or other image processing software, it stills need user to get familiar with the software and spend some time to process the image -- it is not a "real-time" process. By the program we have designed, we can use a camera (stationary for better results!) to record a video and process the video to remove moving objects on time. User can watch the real-time result on screen, if satisfactory result appears, user can capture the image and therefore, got a pure image with only background (the landmark buildings) or sightseeing.

Our program mainly builds on Matlab, the main program handles a video input and by applying our improved median method, we can remove the moving objects and thus we get a picture only contains the background or something static in the video.

2. Related Work

Since there is very few researches or papers about handling the videos and removing moving objects, we have to find or develop an algorithm to detect and remove the moving objects by ourselves.

In 2008, Feng Liu, Yu-hen Hu and Michael L. Gleicher, the alumni of University of Wisconsin, have published a paper *Discovering panoramas in web videos* in Proceedings of the 16th ACM international conference on Multimedia. Although their main focus is on panoramas, the section of 2.2 Moving object detection offers us an idea to use the optical flow to detect the moving objects first.

3. Methods

Our project mainly build on Matlab, and there is a useful tool in Matlab called Computational Vision toolbox, which contains algorithms, functions, and apps for designing and simulating computer vision and video processing systems. There are several ways for object detection and tracking. One of the most straightforward method is the People Detector in Vision Package.

3.1 **People Detector -- From Matlab Computer Vision System Toolbox**

This method detects people in an input image using the *Histogram of Oriented Gradient* (HOG) features and a trained

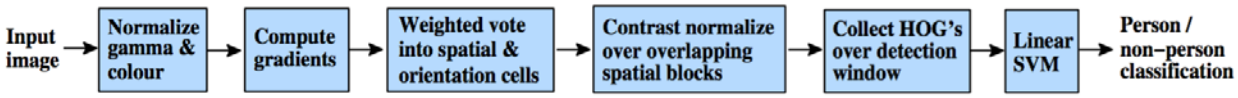


Figure 3.1.a. An overview of our feature extraction and object detection chain. The detector window is tiled with a grid of overlapping blocks in which Histogram of Oriented Gradient feature vectors are extracted. The combined vectors are fed to a linear SVM for object/non-object classification. The detection window is scanned across the image at all positions and scales, and conventional non-maximum suppression is run on the output pyramid to detect object instances, but this paper concentrates on the feature extraction process.

Support Vector Machine (SVM) classifier. The object detects unoccluded people in an upright position.

Since this method uses HOG features and *Scale Invariant Feature Transformation* SIFT approach to detect human, so the motion from camera itself can be ignored to some extent which means, the camera can be hand held, not necessarily be stationary. However, during our tests, we found that this method is quite slow: it will take about one second to process one frame of image. If the input video is 30 fps, it will take 30 seconds to process one-second video. That is slow for our purpose: “real-time” removing.

In addition to the slow-speed of processing, this method is not accurate for small objects, in figure 3.1.c, the backpack and the legs of the girl are also recognized as human, and in figure 3.1.d, the bottle in the right-hand-side of the image is recognized as an upright human.

This kind of recognition issue can be eliminated with a larger training set for SVM; however, in Dalal and Triggs paper, they used 1.7 GB of RAM for SVM training and this training process leads to a significant improvement. The huge amount of RAM usage may restrict the performance of this method.

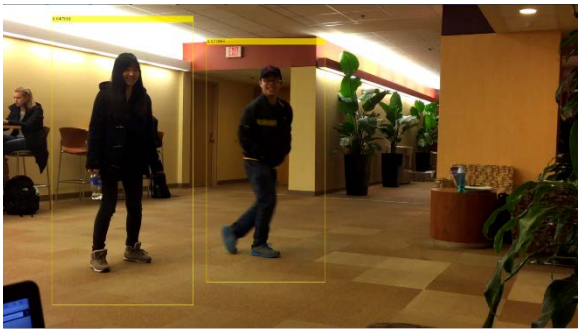


Figure 3.1.b. A good result of HOG-people detector

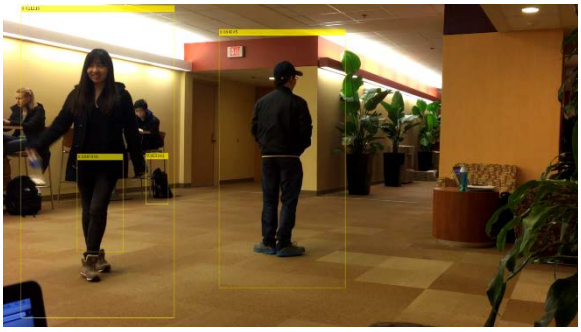


Figure 3.1.c. A poor result of HOG-people detector that recognized backpack and the legs as human

Figure 3.1.d. A poor result of HOG-people detector that recognized a bottle as a human

3.2 Foreground Detector -- From Matlab Computer Vision System Toolbox

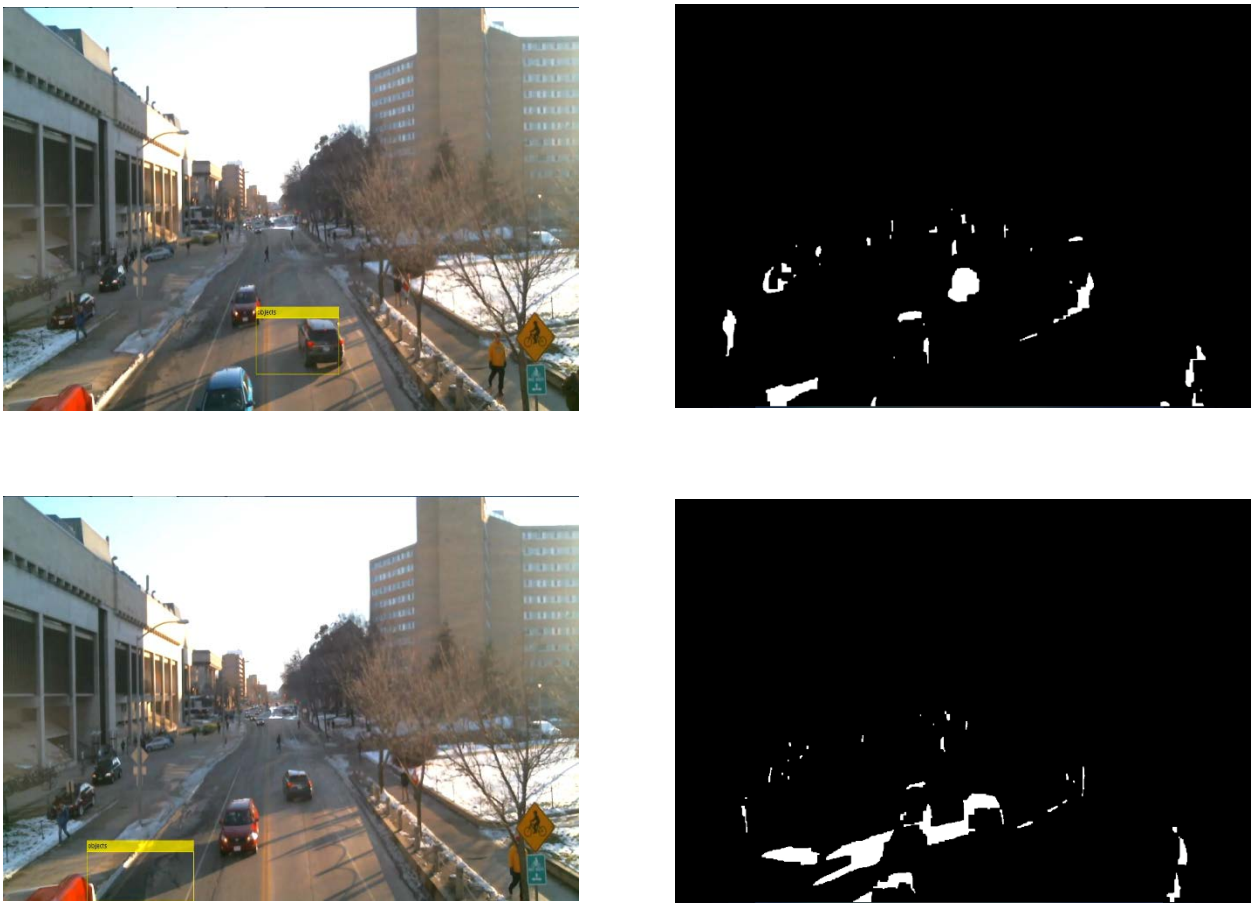
Since the poor performance of the HOG-people detector method, we altered our mind from "human detection" to "background detection". It may be hard for us to detect the human or other moving objects correctly and quickly, but we can try to detect the background instead.

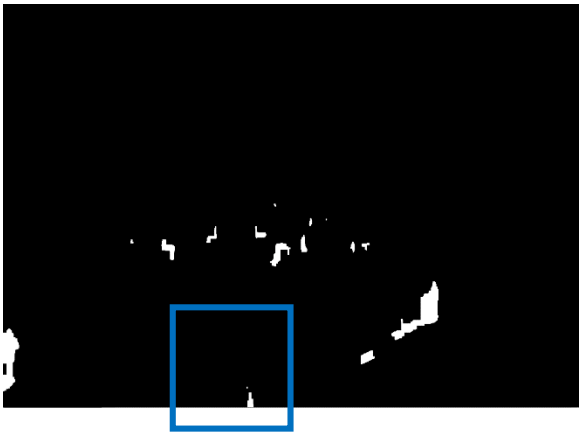
This method compares a color or grayscale video frame to a background model to determine whether individual pixels are part of the background or the foreground. It then computes a foreground mask using Gaussian mixture models (GMM). By using background subtraction, we can detect foreground objects in an image taken from a stationary camera.

Better than the HOG-people detector method, this foreground detector is still relatively slow. This method can process (i.e. detect moving objects in foreground) 5 – 10 frames per second and it may not a good method choice for our purpose – "real time" removing.

Since Foreground Detector uses the first several frames of the video as the training set of the background model, if someone or something does not move, they are going to be recognized as background. Despite the fact that they move after the first several frames, they cannot be recognized as foreground objects correctly.

Figure 3.2. The input video on the left-hand-side and the foreground mask on the right hand-side. It works well at the very first frames, but look at picture 4, 5, 6, the red vehicle stops for seconds, and it is recognized as background.





3.3 **Optical Flow**

This method estimates object velocities from one image or video frame to another. We use the Horn-Schunck method for optical flow in our project. Then we use vision.BlobAnalysis as a morphological filter to fill holes and connect nearby moving regions.

The Optical Flow method is much faster than People Detector and Foreground Detector, so it can detect motion in real time.

However, this method is too sensitive that brightness or shadows can influence the detection and this method could detect even tiny background motion (i.e. leaves in the wind) or light change.

It is also not accurate for motion at similar color range, for instance, on game day, almost everyone wear red; thus it is hard to figure it out. In Figure 3.3.b the black coat's color is similar to the background (i.e. the back of the people on the right-hand-side).

It is hard to find out a proper morphological filter value: if the morphological filter value is too large, some slight changes in the background will be magnified as some moving objects, while if the filter value is too small, a single moving object will be recognized as many different objects.

Figure 3.3.a. A good result of Optical Flow method that can detect moving hand perfectly.

Figure 3.3.b. A poor result of Optical Flow method. On the right-hand-side of the image, the black color of the moving arm is close to the background, so the detection cannot work perfectly.

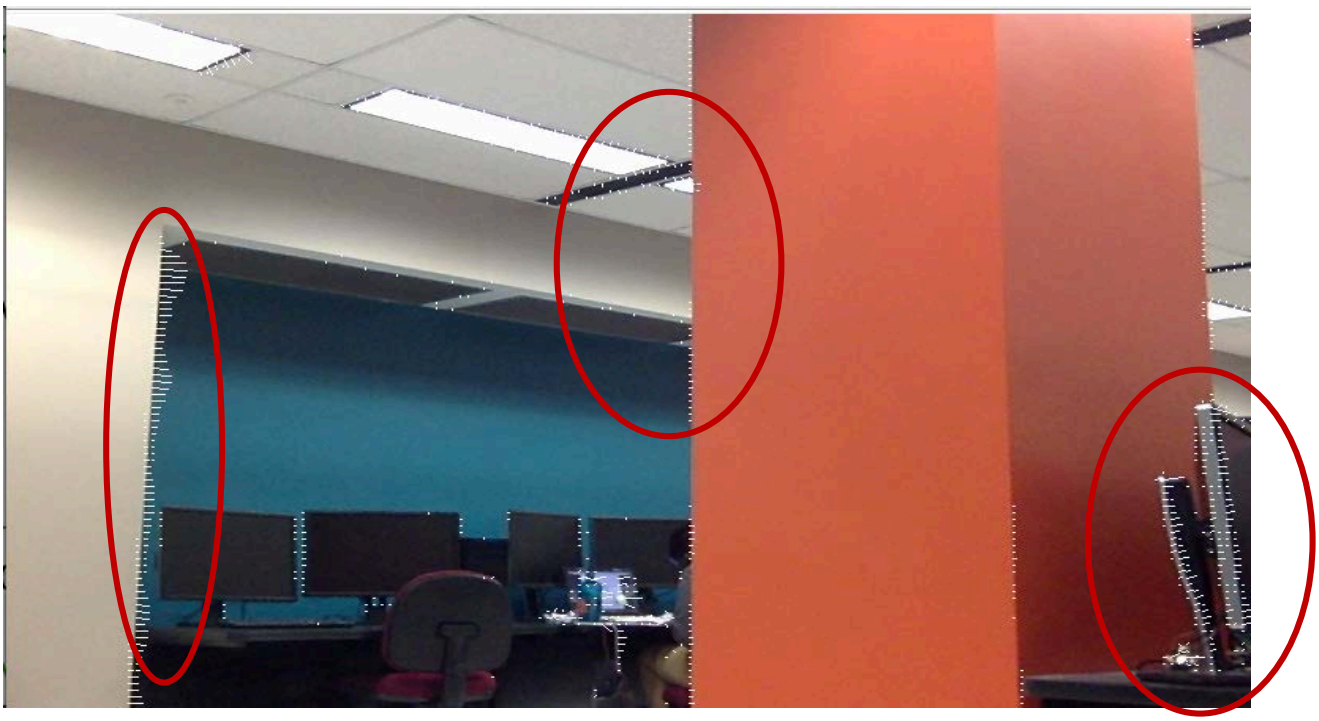


Figure 3.3.c. A poor result of Optical Flow method. Brightness and shadow will influence the detection. The parts of the static background has been detected as moving objects.

3.4 Median Method

In addition to implement some ideas or methods from existed paper or projects, we try to develop an algorithm by ourselves. Since it is hard to detect moving objects or static background in real-time, we try to find a method that can process the frame as a whole not focus on the certain

Assuming the objects we want to remove is moving at detectable speed, as a result, generally, in certain period of time, for each pixel in the frame, the value appears most times should represent the static view or background.

In mathematics, we can find the value that represents the background by calculating the median value of each pixels. So the major point of our algorithm is how to find the median frame as fast as possible for the purpose of "real-time" processing.

Overview of the algorithm

- Select first several frames to be used as training set for background
- Save the frames as a 3-D matrix
- For each pixel vector, calculate its median value
- Save the median value in a "median image" as a preliminary result of background
- For each frame after the training set, calculate the weighted median with "median image"
- Update the "median image"

This Median Method is fast and accurate, since we only consider the major information of the frame, not focus on the moving objects or static background anymore. By this median method, we can correctly remove some slight changes or some tiny movements of objects far away from cameras.

<u>Problems</u>
For large objects with relatively low speed, median method cannot remove these fast and correctly.

When the video is long, it takes a large amount of time to compute all previous frames and get the result.

The main issue of this method is that the camera must be stationary. Any slight motion of the hand or the camera will result in a poor result picture. We have improved our method (in Section 5) so that we can handle the motion by the camera itself.

4. Experimental Results

Figure 3.4.a. The initial frame of the input video

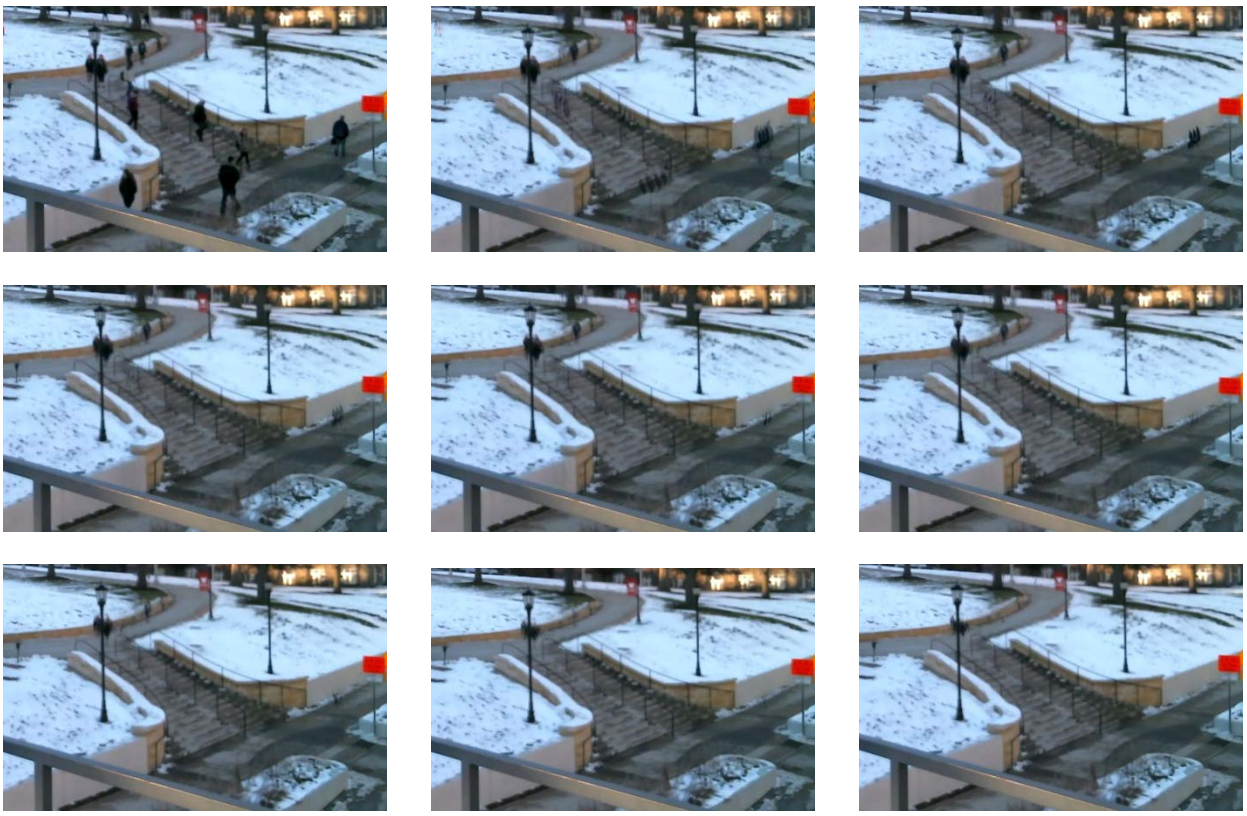


Figure 3.4.b. The process of removing people from background.

Figure 3.4.c. The successful result after removing people.

5. Improvements

For now, the main restrictions of this project is that the camera must be stationary which means that the program cannot handle the motion from the camera itself. For the improvement, we try to use the knowledge of panoramas and apply the RANSAC algorithm and SIFT technique to map every frame to the very first frame so that we can handle some small motions from the camera itself which means the camera can be held in hand.

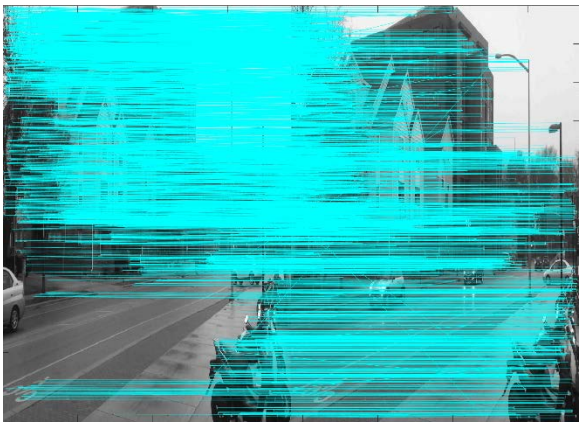


Figure 5.a. the SIFT techniques that map the current frame to the previous frame which can smooth the blur or motion by the camera itself.

Figure 5.b. One of the frame of input video that there exist many moving objects need to be removed.

Figure 5.c. The result of an input video recorded by hand-held camera, although there is some blur in the right-hand-side of the image, it removed the moving objects (passing cars and people) successfully.

6. References

[1] F. Liu, Y. Hu, and M. L. Gleicher, “Discovering panoramas in web videos,” in Proceeding of the 16th ACM international conference on Multimedia, p. 329–338, 2008.

[2] Z. Zivkovic, “Improved Adaptive Gaussian Mixture Model for Background Subtraction,” in Proceedings of the 17th International Conference on Pattern Recognition, p. 28-31,2004.

[3] N.Dalal and B. Triggs, “Histograms of Oriented Gradients for Human Detection,” in Proceedings of IEEE Conference on Computer Vision and Pattern Recognition , p. 886-893, 2005.

7. Extra

Language: Matlab

Line of code: 400+

External Code Sources: Use Matlab library functions in Computer Vision System Toolbox. Source Code can be found at mathworks.com

Contributions:

Code by Yuanchen and Qianyun

Details: General ideas and tests are made by Yuanchen and Qianyun. Yuanchen take responsibility in Optical Flow, Median method and improvement. Qianyun take responsibility in people detector and foreground detector.

Project report by Qianyun and Yuanchen

All images and videos are recorded by ourselves.