Finding Static Shots

Luke Ding

Abstract:

This paper highlights an algorithm to automatically go through all the shots of a video, and divide the video into static and moving shots. The algorithm also trims out black picture in picture videos, as well as opening and closing credits with very high precision and acceptable recall.

Introduction:

Computer Vision research often need static shots in order to develop a ground control data set to test for things like motion and changes. Getting that dataset, however, is challenging. One could run a shot boundary detection algorithm and then manually go through and classify each shot as static or moving. One could also use more complex algorithms that determines motion, but if one only cares about high precision, this algorithm works well enough.

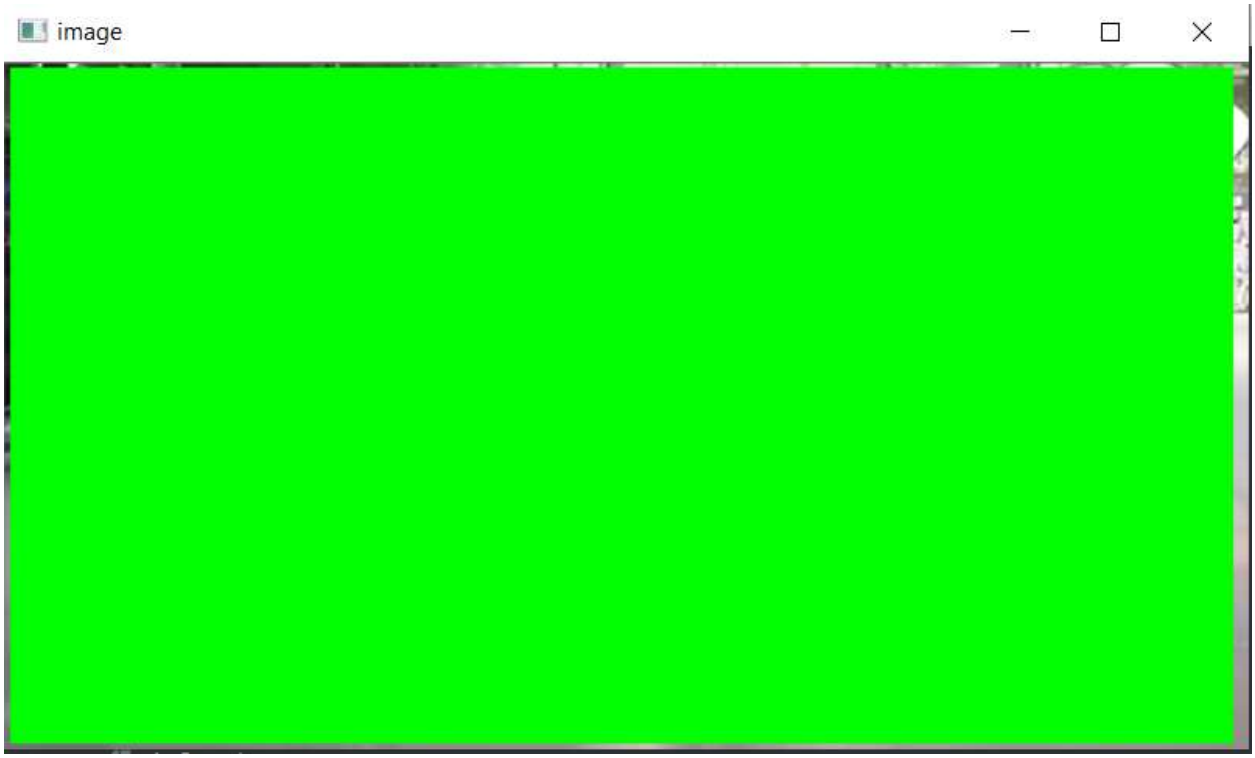
The general heuristic for the algorithm is as follows. Assume that we are dealing with a standard square video, with a subject that does not occupy the entire shot. The difference between a static shot and a moving shot should exist at the boundaries. If a shot is static, then at least one of its four borders should be constant (with slight adjustments for noise). If a shot is moving, then all four borders should be moving as the camera is moving. That is the general idea. However, as one begins actually coding the algorithm, one will discover the following problems:

(1) **Letterboxes-** Often times there are letterboxes around a video that we need to solve for, or else the border will be constant throughout the whole video. This will effect precision.
(2) **Major Border Actions in Static Shots-** In most static shots, there is at least one of the four border pieces that has some motion. This will effect recall.
(3) **Minor Border Actions in Static Shots-** In many (not necessarily most) static shots, there can occasionally be minor actions (think small light flickers, flies, etc).
(4) **Compression Noises-** Compression of videos in .mp4 format (which is the format we used) often introduces noises on a frame by frame basis. Even if a frame and its successors are exactly the same visually and meant to be moving, pixels from a frame and its successor may have slight differences.
(5) **Black picture in picture or movie credits-** Black movie credits or picture in pictures need to be removed.

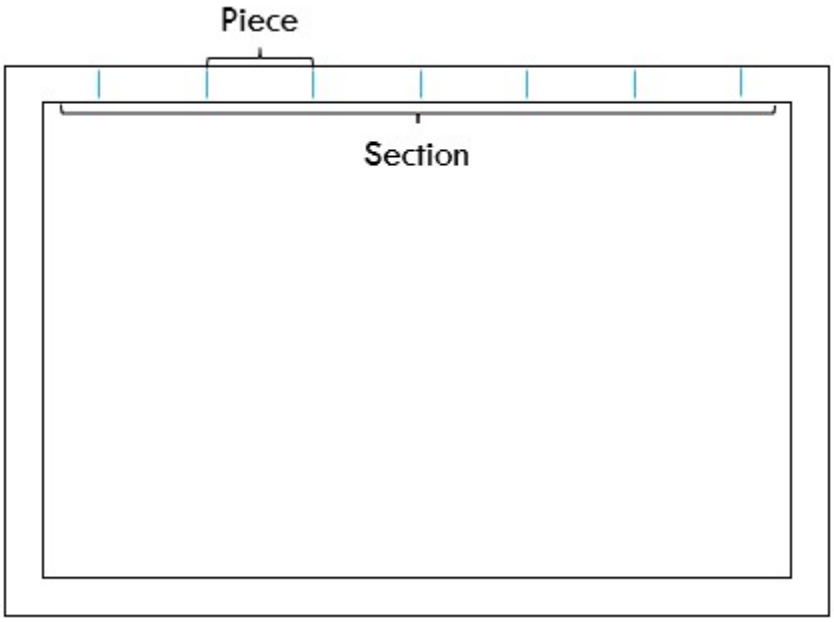
Method:

The methodology for the algorithm will be described below. References to the key problems are are included to reference how they are addressed by the algorithm. The algorithm is described step by step below. A reference of the terminology is also below.

1) A manual tool is used to crop the region of interest for videos. This is done to prevent letterboxes from providing false static shots to the algorithm. This solves problem (1). Example image below.



2) The borders are divided into four sections. Top, bottom, left, and right. Each section is further divided into 8 pieces.
3) Frames are analyzed piece by piece, frame to frame. If 0.15% of the pixels in a frame differ by more than the previous one, then the piece is classified as a non-static piece. The 0.15% frame difference and pixel difference of more than one are to control for compression noise (4).
4) If more than 1 piece is classified as a non static then border the border section is classified as non-static. This adjusts for problem (3), minor actions in border frames.
5) If more than 3 border sections of the frame are non-static, that frame is non static. This adjusts for problem (2).
6) The algorithm then produces a sequence of at least 20 frames.
7) The algorithm then cleans up picture in picture and border credits.



Experiments:

The algorithm was tested against 24 videos, which contained a variety videos from Bollywood movies to tutorials. The algorithm performed well, with no errors that were unknown for. The only frames that were identified as errors were opening credits as well as black picture in picture. A second pass of the algorithm determined whether a frame was a credit, and the patch has been applied.

There are two data sets. The first is a csv file called test05_26_2019.csv. This contains the static frames for 24 videos. The pickle files are also included for fast play and load. Frames like Break, which have opening credits, have those credits included. The test06_01_2019.csv file contains the patch, and was tested on a few videos that did have PIP or black credits. There, the credits are taken out (e.g. Loner and Break). I have also loaded the pickle files and original video files for ease of view.

The instructions to play clips by clips are included in the instructions tab.

Next Steps:

There are two next steps. The first is to make sure the algorithm runs properly on Hoang's machine, as I've had difficulty getting it to run on older Python2 versions.

The second is that Hoang is writing an automatic cropper. The automatic cropper should be much faster at trimming the letterboxes, and improve accuracy as well.

Running Instructions:

To run the program, make sure you have Python 3.X installed. As well as at least the following packages:

```
import cv2
import numpy as np
import csv
import glob
import math
from collections import defaultdict
from matplotlib import pyplot as plt
import pickle
```

The algorithm has had trouble running on older versions of Python. The following table shows the versions.

Package	Version
Click	7.0
cycler	0.10.0
kiwisolver	1.0.1
matplotlib	3.0.3
numpy	1.16.2
opencv-python	4.0.0.21
pandas	0.24.2
pip	10.0.1
pyparsing	2.3.1
python-dateutil	2.8.0
pytz	2019.1
scenedetect	0.5
setuptools	39.1.0
six	1.12.0

To run the static shot detection, do the following steps.

1) Place all videos that you wish the analyze into the same directory as the main and the import file.
2) Change like 87 to save the output into the csv file name you want
3) Run the algorithm. The algorithm will first have you manually crop the boundaries. Start at the top left corner and scroll right to the bottom right. Once you are happy with the bounding ROI box, hit ‘q’ to move onto the next one. Hit ‘r’ to skip 10 frames if that frame is not clear enough. Once you cycle through all your videos in the file, the algorithm will begin analysis.
4) The algorithm will save to a CSV file like below:

```
Break.mp4|(114,0)|(609,1279)|213|(0,109)|(193,254)|(1326,1357)|(1379,1406)|(2589,2662)|(3210,3237)|(3878,3905)|(4642,4702)|
```

The order of information is your filename, ROI top left, ROI bottom right, # static frames, and the frame start and end in tuples. It will also save the static frames in a pickle file for easy access.

To run the playback, do the following steps:

Comment out everything from lines 26-87. Uncomment out lines 90-91. Replace with the correct pickle file and mp4 file, and hit play. Hit ‘q’ to quit. Hit any other key to skip to next frame. It will take some time if the frames are far apart as the video is actually waiting until the next block.

WARNING: SOME VIDEOS HAVE NO STATIC SHOTS. IF YOU TRY TO PLAY THEIR PICKLE FILE AN ERROR WILL OCCUR. THAT IS NATURAL.