**CS512 Final Project Report**

**Su Feng A20338748** (Group with Jingcheng Deng)

**Smoking Detection Algorithm Manual**

**Testing environment:**

MacOS Mojave

Python 3.6.1

**External libraries used:**

Guided filter: Used to refine transmission image.

<http://kaiminghe.com/eccv10/index.html>

<https://github.com/joyeecheung/dark-channel-prior-dehazing/blob/master/src/guidedfilter.py>

python contrib: used to manage MHI interfaces.

<https://pypi.org/project/opencv-contrib-python/>

**Related files:**

smoke.py – main python file to run the smoke detection algorithm

guidedfilter.py – external library used to perform refinement on dark channel transmission results.

**Parameters:**

Can be adjusted by changing values in the beginning of the python file, initial values can be found on parameter tuning section in the report.

Image size: resize input if size exceeded

MHI\_DURATION: MHI history traversed per frame.

DEFAULT\_THRESHOLD: minimum absolute difference for MHI.

Colorth: RGB difference threshold for chromatic analysis.

minI: Minimum intensity threshold for chromatic analysis.

**Example Workflow:**

> Python3 smoke.py

> Please enter video path: videos/smoke\_3.mp4

> Please enter detect mode(video/realtime/debug): debug

> Do you want to apply guided filter for better result but slow the process?(Y/N):N

> Video Path:videos/smoke\_3.mp4 Mode: debug Guided\_Filter: N

> Enter debug mode!

> Do you want to exit? Y/N:

video: process input video and produce output.mp4 file

realtime: process input video and show result through cv2.imshow

debug: enter debug mode, will output intermediate result.

Filter: Whether apply guided filter

**CS512 Final Project Report**

**Su Feng A20338748** (Group with Jingcheng Deng)

**Smoking Detection Algorithm**

**Abstract**

Video based smoking detection is an important yet challenging topic. This paper [1] proposed a video-based smoke detection method by first extract smoke candidate regions using chromatic analysis and then further identify smoke moving patterns using MHI. Since smoke tend to have a blending pattern, the proposed method also includes dark channel priori that can be used to extract the background greyish components and restore the input image from potential background noise factors like atmospheric light. The paper claimed that the combination of these three methods can produce accurate smoke detection results yet efficient enough to be applied in real time video analysis.

**Main Challenges**

The paper is a 4 pages short workshop like paper with minimal information included about the system. As the paper using multiple methods introduced by other papers, we need to go through multiple referenced papers to have decent understanding in order to implement them. The paper claimed some thresholds for identifying smokes which we found out not ideal for some of the test cases which adds extra work in terms of parameter tuning.

**Contributions**

In this project, we implemented the smoke detection algorithm described in the paper in python using OpenCV. We made decisions on some implementation details that are not mentioned clearly in the paper. We tested the algorithm on inputs with various smoke senses with different smoke intensity and surrounding environment. I mainly focus on the chromatic and MHI analysis part. My group partner mainly focused on the dark channel prior part and we did the testing together. The table below shows detailed workload division per group members:

|  |  |  |
| --- | --- | --- |
| Work | Su Feng | Jingcheng Deng |
| Implementation Total | 50% | 50% |
| Chromatic Analysis | 100% |  |
| Dark Prior |  | 100% |
| MHI | 100% |  |
| Video helper functions |  | 100% |
| Misc. helper functions | 50% | 50% |
| Combination & main | 50% | 50% |
| Testing Total | 50% | 50% |
| Chromatic Analysis | 50% | 50% |
| Dark Prior | 50% | 50% |
| MHI | 50% | 50% |
| General | 50% | 50% |
| Test result / figures / ppts | 50% | 50% |

**Background**

In this section we briefly introduce some background and related works used by the paper to implement the algorithm. For this report I mainly concentrate on the works related to my contributions.

1. **Chromatic analysis:**

Since commonly smoke has greyish color with low differences between three color channels. Chromatic analysis is a simple way to extract greyish pixels to be used as potential smoke regions. A simple chromatic analysis algorithm can be written as:

The algorithm basically computes the per pixel max and min values for the RGB channels. We use the last line of the algorithm to determine whether an individual pixel is a smoke candidate pixel. The pixel is treated as smoke candidate if the maximum distance between its RGB channels is below threshold T1 and its intensity I is in range (T2, T3). We will discuss the value for these thresholds in parameter tuning section.

Below shows an example result of Chromatic analysis using a simple smoke input:



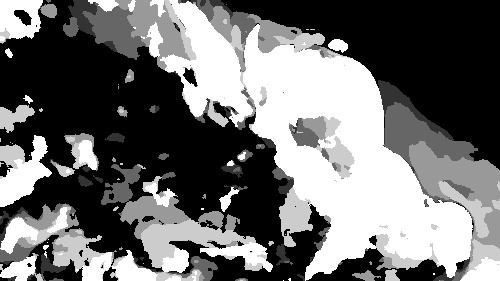
1. **MHI:**

MHI (Motion History image) is a static image template helps in understanding the motion location and path as it progresses. The temporal motion information is collapsed into a single image template where intensity is a function of recency of motion. For each pixel in the MHI image, the intensity is a function of motion history at that location where brighter values correspond to a more recent motion. A brief MHI algorithm is shown below:

We can see that depend on the function of , we can limit the number of frames captured by the MHI. In our implementation we set a threshold to adjust this number. Also, we can use threshold to adjust how much motion we want to catch. Below is an example of MHI analyzing 5 frames using different threshold values:

Original: (single frame)





We can see that lower tend to capture more movements from temporal frames and higher will only keep more distinct movements. We can use this feature to eliminate potential noises that are very easy to confuse simple static smoke detection algorithms like slow moving cloud or dense fog.

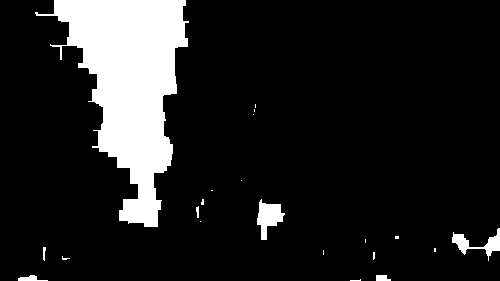
1. **Dark channel prior:**

[Briefly introduced as mainly implemented by my group partner]

Dark channel prior is an image prior first introduced by He, al. [2] used to perform haze removal on outdoor images. The technique is based on observation that most of natural images often have one RGB channel with very low intensity. In most haze region, the intensity of dark pixels in a channel are mainly because of air light, so we can calculate per pixel transmission value and use the value to restore the pixel into the condition before haze or fog. The idea to use dark prior to find smoke region is because smoke can be seen as a dense haze or fog. Using a threshold over computed transmission image template can help us find smoke regions with higher precision than chromatic analysis. Furthermore, guided filter can be use on top of the transmission result to further refine it (depend on the implementation, guided filter might be expensive to apply). Figures below shows the original image, dark channel image, transmission image and refined transmission image:



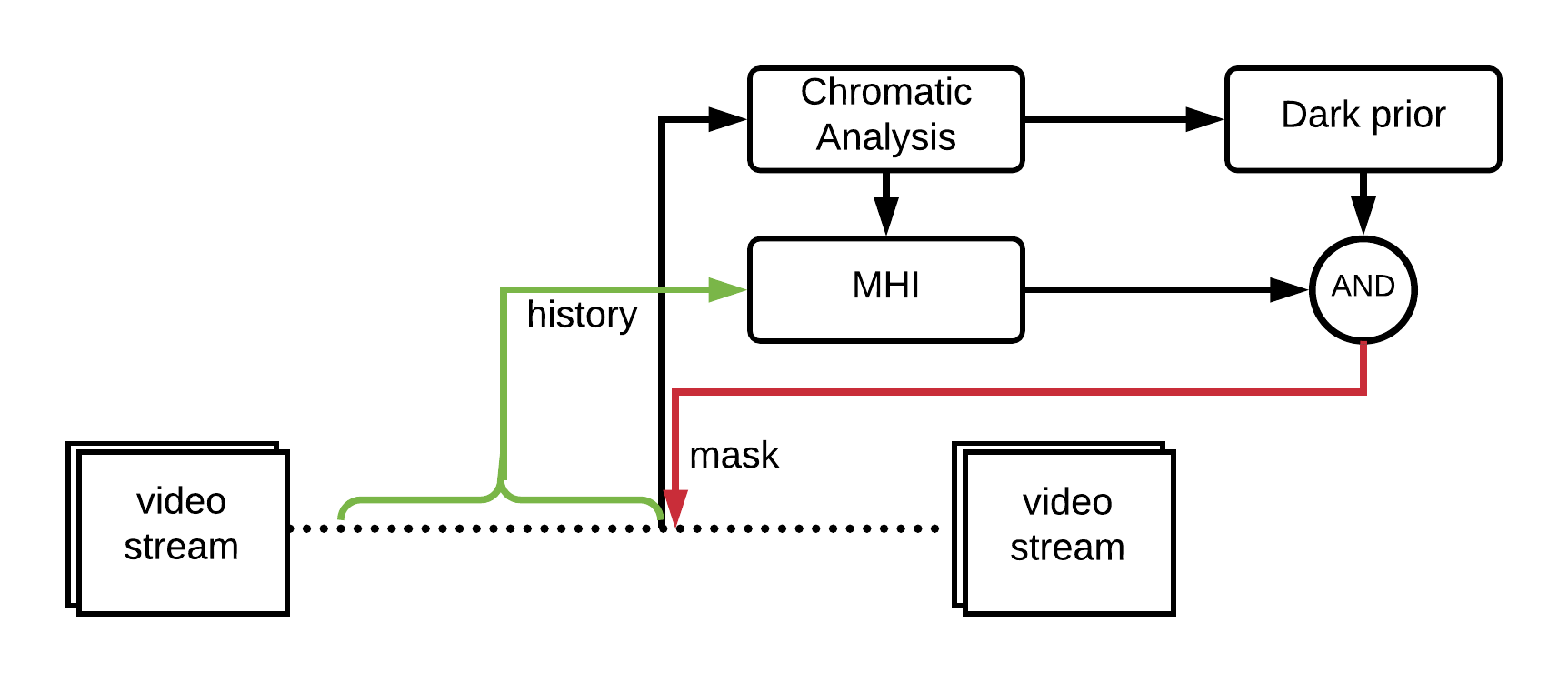
Original Dark channel



Transmission Transmission refined

**Implementation**

In our implementation, we combine three methods described above to be the core of our smoke detection algorithm. We use Chromatic analysis as the first layer to roughly picking all potential smoke pixels as the input to our consequent methods. Since dark channel prior is relatively computational expensive especially when guided image filtering is applied, only compute dark channel on potential smoke pixels can optimize the performance. After Chromatic analysis, we compute dark channel and MHI separately into two distinct image templates. The overlapping of the two image templates are the final result points to be considered as smoke. Figure below shows the basic structure of the workflow:



As shown on the figure, after applying chromatic analysis on the extracted frame from the video stream, we input the result into both MHI and dark prior analysis as a reference of which pixels need to be considered. MHI extracts history images from the stream and outputs an image template with value 0 in pixel indicating no motion in the history at all and 1 indicating motion is happened between current and last frame. Intermediate motions are represented by factor on n where n denotes the maximum number of history frames can be recorded in the MHI. Since we only need previous MHI and current frame difference to compute next MH  
I and MHI is basically a single image with same size as the video, increasing n does not affect time and space performance. By knowing n, we can retrieve any range of history up to n by thresholding over the MHI.

For dark channel analysis, we first compute the dark channel image and then get the transmission image from it. Then we use guided filter to refine the transmission image. The resulting transmission image supposed to have pixel values between 0 and 1. We threshold the image using T=0.4 and return the result as potential smoke regions after normalizing.

After getting the partial history MHI and threshold dark prior image, we can find smoke regions as the overlapping highlighted areas of both images (as the “AND” component represented in the figure). We draw masks on the corresponding locations into the video stream.

In case of real time video detection, we have two options to handle processing latency under the condition that latency is not significantly large. One is to keep the video streaming going and after smoke region is detected, show the region on consequent image frames. As shown in the workflow figure, the red arrow indicating draw back of the smoke region will take place later than the frame extract arrow inputting the frame into chromatic analysis. Which means the draw back will perform on a different frame from the one we analyzed. As a result, this method might have slight mismatch in smoke and labeled regions in case of fast-moving smoke, but in general the detection algorithm is efficient enough to hide those minor delay. Another approach is to further reduce the execution time by sacrificing the smoke region refinement. The most time-consuming step in the algorithm is the guided filter refinement of the transmission image in dark prior. We can remove this step to have faster processing. Under this circumstance, the region will not be as precise as refined version, so for this method we choose to only show whether there is smoke in sense or not by printing a smoke indicator on the video instead of showing the precise smoke regions.

**Parameter tuning**

This section will talk about initial parameters and their meanings used in the implementation on functions mostly from my contribution:

Image size: 500

Program will limit the input largest side into 500 if it is over 500 and shrink the size proportionally. We choose image size of 500 to optimize the performance of analysis. Increase image size limit may increase processing cost.

MHI\_DURATION: 5

The MHI algorithm will only traverse 5 previous frames as the MHI history. Increase number of durations can expose more smoke candidates to the result by traversing more history motions.

DEFAULT\_THRESHOLD: 30

The minimum difference in intensity MHI allowed for two pixels to be considered as a motion. Increase this threshold can expose more smoke candidates per motion history by considering lower absolute difference as motion.

Colorth: 85

The maximum RGB difference in a pixel to be considered as smoke candidate in Chromatic analysis. Increase this number will eliminate more (non-greyish) pixels from been a smoke candidate.

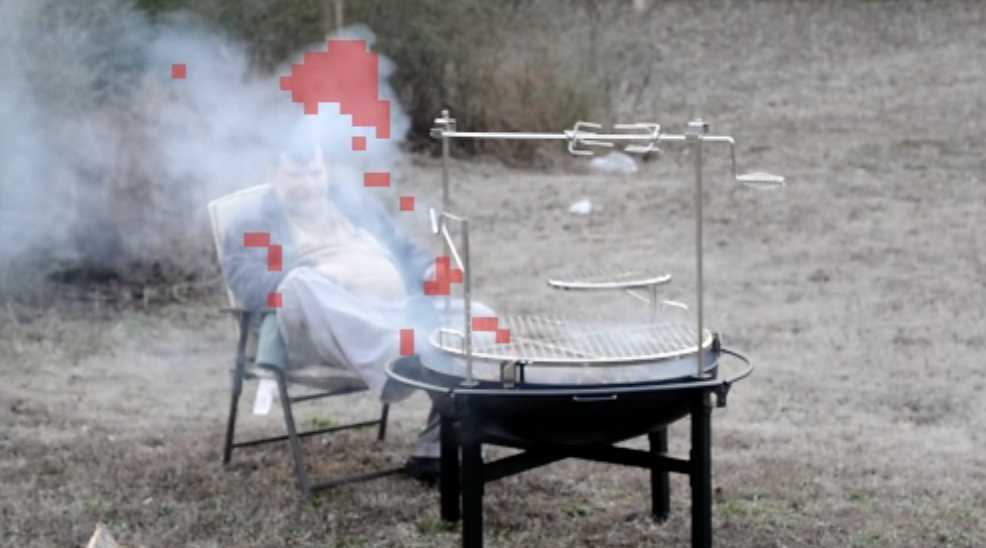
minI: 100

The minimum intensity in a pixel to be considered as smoke candidate in Chromatic analysis. Increase this number will eliminate more dark pixels from been a smoke candidate.

**Test results**

This section we show test results on real videos and performance analysis.



****

From the test result we can see that partial smokes are identified and labeled in red. This is because based on the default parameter, only the actively moving smoke patterns are identified as smoke to eliminate noises.

In terms of performance, each frame takes roughly 0.25 seconds to process without refinement which makes real time processing not very smooth but applicable. However, the guided filter library is not a fast guided filter so using refinement will increase per frame execution time to roughly 4 seconds.

**Future works**

1. Implement or import a faster guided filter algorithm to speed up the refinement.
2. Add motion orientation on top of MHI to have a histogram-based motion analysis to further identify the movement pattern in terms of direction of smoke.
3. Analyze block of images instead of each pixel to achieve smoother real time smoke detection.

**References**

[1] Miao Ligang, Chen Yanjun, Wang Aizhong; Video Smoke Detection Algorithm Using Dark Channel Priori; Proceedings of the 33rd Chinese Control Conference; July 28-30, 2014, Nanjing, China.

[2] Kaiming He, Jian Sun, Xiaoou Tang; Single Image Haze Removal Using Dark Channel Prior; 978-1-4244-3991-1/09/2009 IEEE.