MSFT: Background Subtraction - HW #1

Due on October 9, 2014

Dro Désiré Sidibé

Emre Ozan Alkan

October 9, 2014

Contents

Problem 1

In problem 1, we asked to use frame difference method. We've given 2 sequence of images. In my method It starts from second frame and using only previous frame to achieve my results instead of using median with long history.

Initially code doesn't produce good results, however with time and learning, It can track the cars in *Car* and *Highway* sequences. You can see Table 1 on the page 3 indicating the parameters of algorithm. Learning rate of *Highway* sequence is two times bigger than *Car* sequence due to changing position of the camera, lightning conditions and cluttered background.

Image Sequence	**Learning Rate**	**Threshold**
Car	0.05	50
Highway	0.1	50

Table 1: Learning rate and thresholds for sequences

Highway sequence has 1700 images. We've asked to use first 470 image training our background and using rest of the images to test and detect cars in the scene. In Figure 1 you can see the results of the method on different image sequences.

(a) Frame difference method on *Car* sequence

(b) Frame difference method on *Highway* sequence

Figure 1: Results of frame difference method

Problem 1 continued on next page. . .

In order to achieve results on the Figure 1 on the preceding page, we used many morphological operations as demonstrated in Code Listing 1

Listing 1: Morphological operations used for different sequences

```
% Morphological operations for Car sequence
Ob2 = bwareaopen(Ob, 50, 8);
Ob3 = bwmorph(Ob2, 'dilate');
Ob4 = imfill(Ob3, 'holes');
Ob5 = bwmorph(Ob4, 'erode', 2);

% Morphological operations for Highway sequence
ObFiltered = bwmorph(ObFiltered, 'bridge', 'Inf');
ObFiltered = imfill(ObFiltered, 'holes');
ObFiltered = bwmorph(ObFiltered, 'bridge', 'Inf');
ObFiltered = bwareaopen(ObFiltered, 8, 8);
ObFiltered = bwmorph(ObFiltered, 'dilate', 1);
ObFiltered = medfilt2(ObFiltered, [9 9]);
ObFiltered = imfill(ObFiltered, 'holes');
ObFiltered = bwmorph(ObFiltered, 'bridge', 'Inf');
ObFiltered = bwmorph(ObFiltered, 'erode', 1);
```

We also calculated precision, recall and f-score of the results obtained from *Highway* sequence shown in Table 2 .

	Score
Precision	%84.17
Recall	%73.92
F-Score	%78.22

Table 2: Precision, recall and f-score scores

Overall, this method looks good on the given *Car* and *Highway* sequences. However It will be not enough for more advanced scenarios like high illumination change, cluttered background.

Problem 2

In problem 2, we asked to use running average Gaussian method. We've given 2 sequence of images. In this method we had mean and sigma images updated continuously. Mean image is used to removed from the current image, and sigma image to get foreground model with some threshold. Sigma image is also updated with learning rate continuously. Like previous method, this method also had problems on initializing and getting stable. After dozen of frames, it becomes more stable and less noisy on *Car* and *Highyway* sequences. Table 3 on the page 4 indicating the parameters of algorithm.

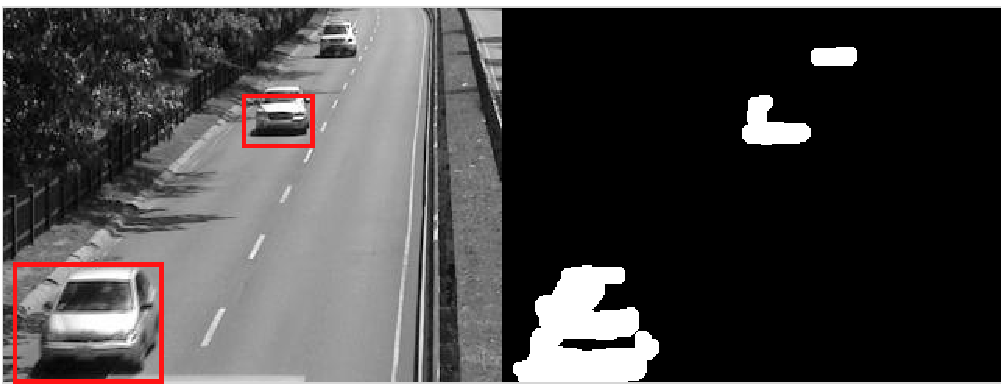
Image Sequence	**Learning Rate**	**Threshold**
Car	0.02	3
Highway	0.02	2.5

Table 3: Learning rate and thresholds for sequences

Problem 2 continued on next page...

Highway sequence has 1700 images. We've asked to use first 470 image training our background and using rest of the images to test and detect cars in the scene. In Figure 2 you can see the results of the method on different image sequences.

(a) Running average Gaussian method on *Car* sequence



(b) Running average Gaussian method on *Highway* sequence

Figure 2: Results of Running average Gaussian method

In order to achieve results on the Figure 2, we used many morphological operations as demonstrated in Code Listing 1 .

Listing 2: Morphological operations used for different sequences

```
% Morphological operations for Car sequence
foreground = medfilt2(foreground, [5 5]);
foreground = bwareaopen(foreground, 20, 8);

% Morphological operations for Highway sequence
foreground = bwareaopen(foreground, 30, 8);
foreground = medfilt2(foreground, [5 5]);
foreground = bwmorph(foreground, 'bridge', 'Inf');
foreground = imfill(foreground, 'holes');
se = strel('disk', 5);
foreground = imdilate(foreground, se);
```

We also calculated precision, recall and f-score of the results obtained from *Highway* sequence shown in Table 4 .

	Score
Precision	%80.46
Recall	%33.94
F-Score	%44.74

Table 4: Precision, recall and f-score scores

Overall, this method looks more noisy then the previous method especially on *Highway*. It was challenge for me to detect objects with the method, and unfortunately I didn't able to get stable results. It's also obvious that our recall and f-score is very low. We think we should work on the method to make it more robust against noise and maybe better morphological filtering.

Problem 3

In problem 3, Mixture of Gaussians methods was given us, however it was optional to investigate furthermore background subtraction methods. So we implemented it to see and experiment. We only tested the method on *Car* sequence. So no training and testing is done for *Highway* sequence. We think it was most the complicated method among others. It was using the multiple Gaussian distribution to model each pixel. We used 3 Gaussian distribution on testing the method on *Car* sequence. It was also the slowest method among others, however result was more better than running average Gaussian method.

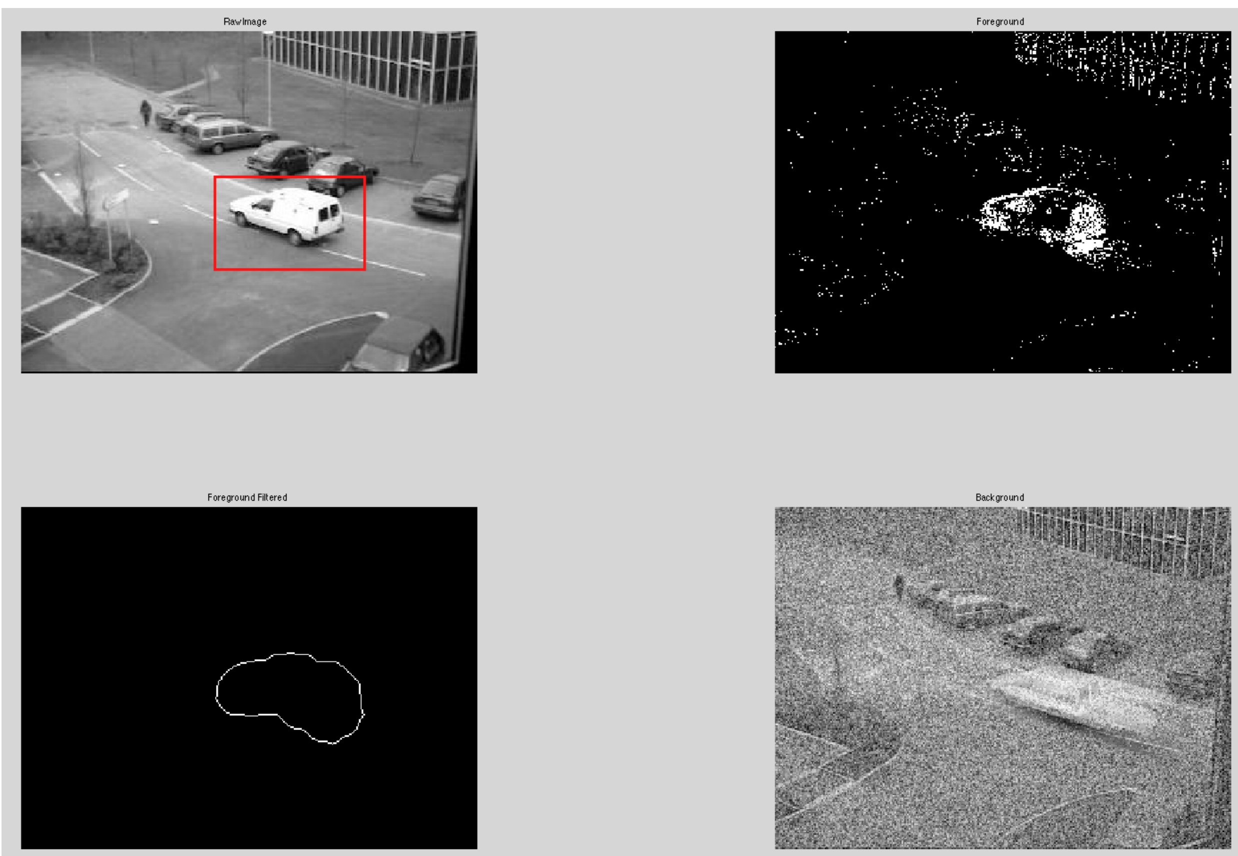


Figure 3: Result of mixture of Gaussians method on *Car* sequence

In order to achieve results on the Figure 3, we used many morphological operations as demonstrated in Code Listing 3 .

Listing 3: Morphological operations used for *Car* sequence

```
% Morphological operations for Car sequence
foregroundFiltered = bwareaopen(foreground, 100, 8);
se = strel('disk', 13);
foregroundFiltered = imdilate(foregroundFiltered, se);
foregroundFiltered = bwmorph(foregroundFiltered, 'bridge', 'Inf');
foregroundFiltered = medfilt2(foregroundFiltered, [5 5]);
foregroundFiltered = imfill(foregroundFiltered, 'holes');
foregroundFiltered = bwmorph(foregroundFiltered, 'erode', 5);
foregroundFiltered = bwmorph(foregroundFiltered, 'remove');
```

Overall, it was complex method to implement. It was running slow with large number of Gaussian distributions. In conclusion, its stable method and producing less noisy foreground model. It's more efficient implementation then current one is left as a future work :)

Problem 4

In problem 4, we asked to use Eigen Background method. We've given 2 sequence of images. In this method, we first calculated mean image for each sequence from each image. Then we normalized our images by subtracting mean image from them. Then applied PCA via SVD by finding some subspace that spanning our images. This method we think so produced stable results then others on *Car* and *Highway* sequences. We used our subspace size as number of images divided by number of images in sequence to fulfill $k << N$ constraint.

Highway sequence has 1700 images. We've asked to use first 470 image training our background and using rest of the images to test and detect cars in the scene. In Figure 4 on the following page you can see the results of the method on difference image sequences.

In order to achieve results on the Figure 4 on the next page, we used many morphological operations as demonstrated in Code Listing 4 .

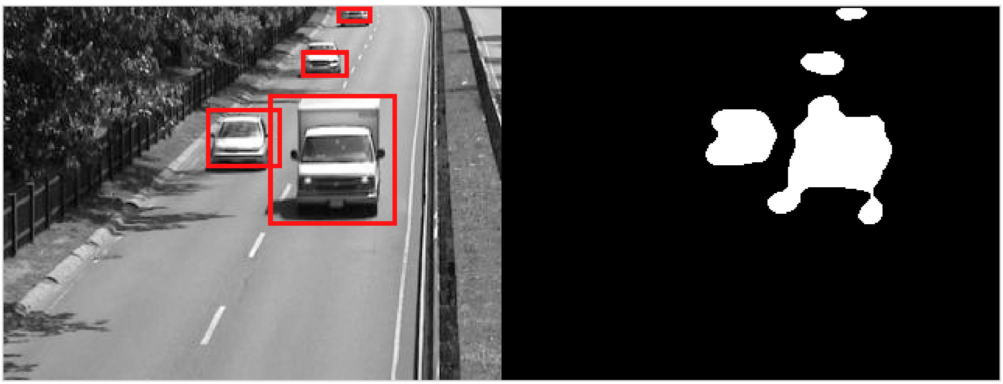
Listing 4: Morphological operations used for different sequences

```
% Morphological operations for Car and Highway sequence
foregroundImage = reshape(foreground, [imageHeight, imageWidth]);
foregroundImage = bwareaopen(foregroundImage, 16, 8);
foregroundImage = bwmorph(foregroundImage, 'bridge', 'Inf');
se = strel('disk', 7);
foregroundImage = imdilate(foregroundImage, se);
foregroundImage = medfilt2(foregroundImage, [9 9]);
foregroundImage = imfill(foregroundImage, 'holes');
foregroundImage = bwmorph(foregroundImage, 'erode', 5);
```

We also calculated precision, recall and f-score of the results obtained from *Highway* sequence shown in Table 5 .

Problem 4 continued on next page...

(a) Eigen Background method on *Car* sequence



(b) Eigen Background method on *Highway* sequence

Figure 4: Results of Eigen Background method

	Score
Precision	%95.71
Recall	%79.28
F-Score	%85.98

Table 5: Precision, recall and f-score scores

In conclusion, the Eigen Background method looks very stable and fast among others on *Car* and *Highway* sequences. It produce nice results with less noise. We think its also easy to implement. Whereas when the data-set is increasing by more images in sequences, it's getting hard to manage matrix operations, i.e., using SVD, where we had to use economy mode of it. As a result, it has the highest F-Score, it was easy to implement and fast. We would prefer this method to be used in our further works.

Conclusion

In this lab, we reviewed Frame differencing, Running average Gaussian, Mixture of Gaussians and Eigen Background. Eigen Background and Frame difference methods produced nice results while tracking the objects. However all methods had problems with backgrounds like in *Highway*, where there was trees, very small camera vibration. We think these methods alone cannot overcome the very cluttered backgrounds nor camera motion or very high illumination change.