r06946003 湯忠憲

Computation in Data Science - Background Removal

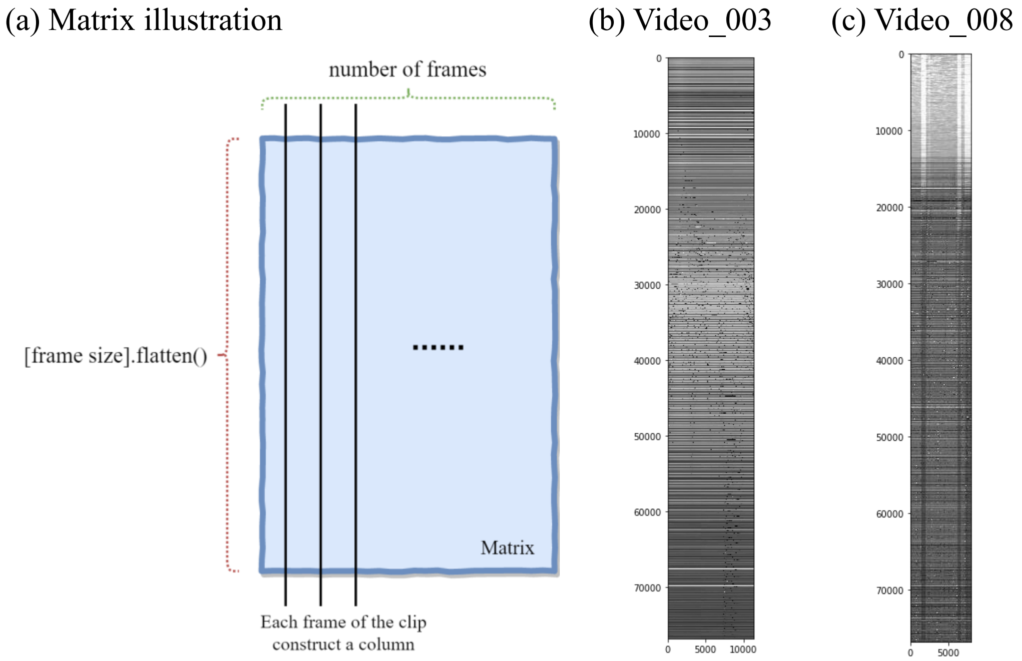
Try to analysis in better resolution.:

Here I try original resolution (320 * 240 for every frame size) on two videos.

○ Select two videos to analyze, please specify which videos you used

I select *Video_008* (“Traffic during windy day”) and *Video_003* (“Wandering students”) form Background Models Challenge[1].

○ Plot matrices


(a) Matrix illustration

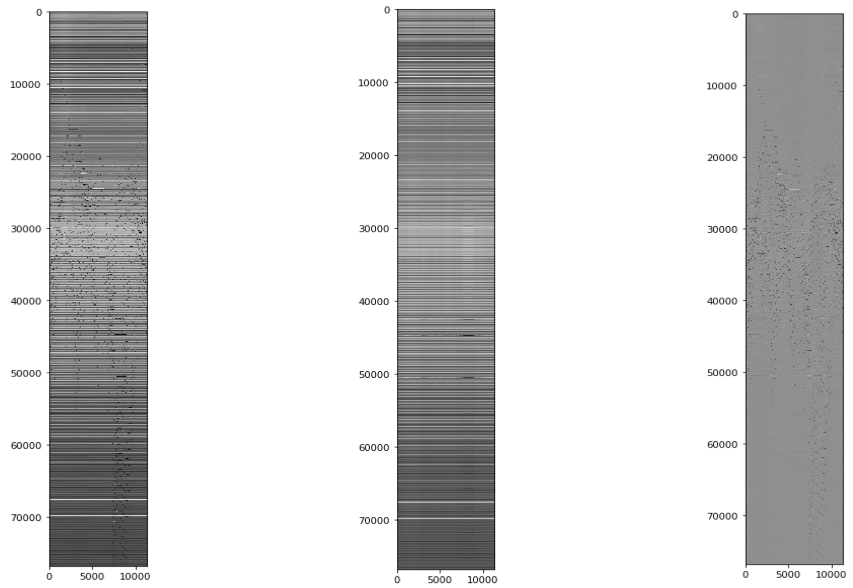
(b) Video_003

(c) Video_008

Figure 1. Matrix of flatten frames

[1] http://bmc.iut-auvergne.com/?page_id=24

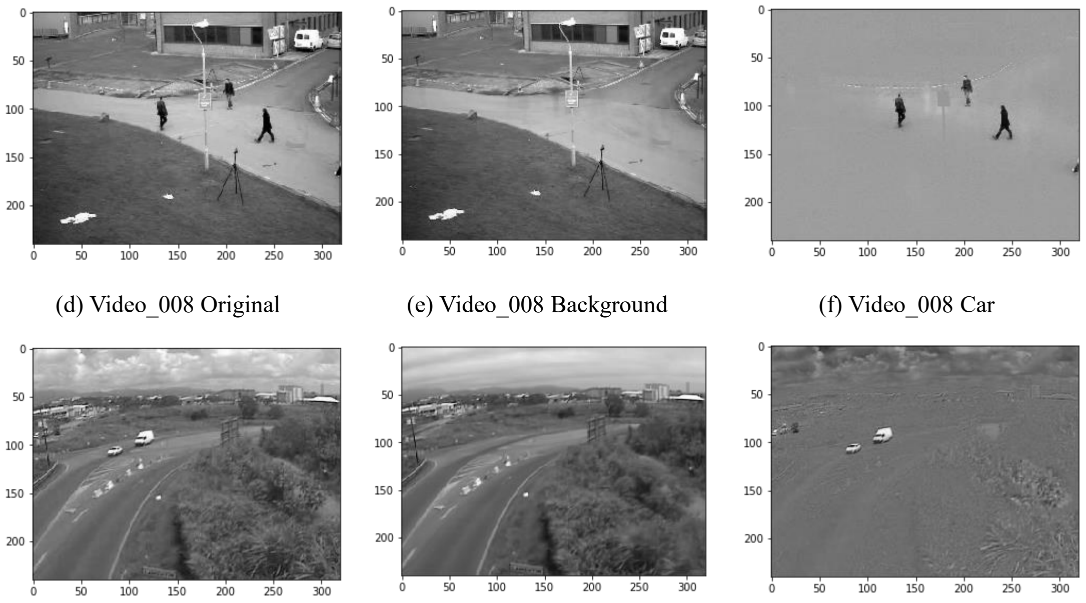
○ Background subtraction comparison



(a) Original matrix (b) Low rank approximation (c) Residual

Figure 2. Matrix decomposition

The above figure shows the decomposition of flatten frame matrix. Figure 2(b) is the approximation matrix after implement SVD on matrix in Figure 2(a). Then subtract the low-rank approximation matrix from the original matrix, we can get a matrix that contains the information of “People” or “Car”. The results are shown below:

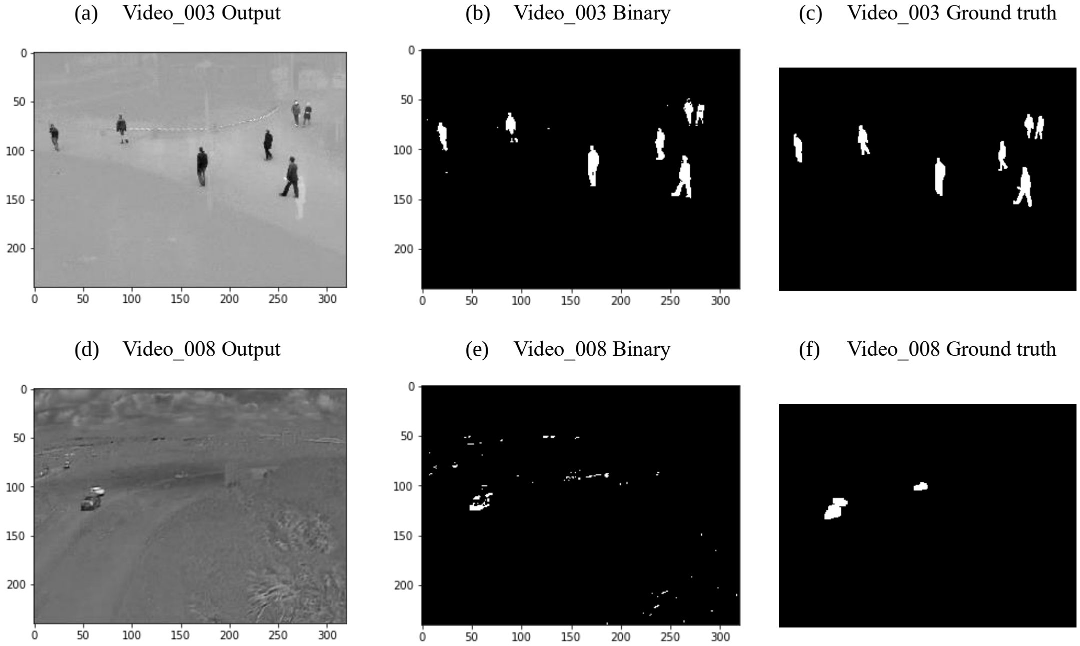


(a) Video_003 Original (b) Video_003 Background (c) Video_003 People

(d) Video_008 Original (e) Video_008 Background (f) Video_008 Car

Figure 3. Background removal results

○ Try to analysis performance


(a) Video_003 Output

(b) Video_003 Binary

(c) Video_003 Ground truth

(d) Video_008 Output

(e) Video_008 Binary

(f) Video_008 Ground truth

Figure 4. Performance evaluation for SVD

For evaluation the performance of background removal using SVD, we're asked to use dice coefficient as a measurement. The formula is shown below:

$$S = \frac{2|X \cap Y|}{|X| + |Y|}$$

Besides color frames for testing, black and white ground truth pictures are also provided. However, the outputs of model are in gray scale, see Figure 4(a) and (d). Therefore, we have to determine a threshold to transform them to binary representation. The threshold would be a parameter for tuning. The results of binary representation are shown in Figure 4(b) and (e).

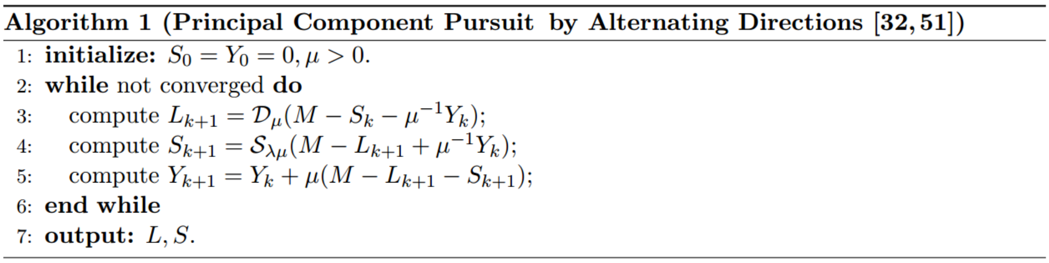
After that, we could compare the similarity with ground truth. The average dice coefficients of testing sets are listed below:

- Video_003: Average dice Coefficient= **0.869**
- Video_008: Average dice Coefficient= **0.346**

The performance looks good in Video_003. On the other hands, Video_008 remains more noise. I think the main difference of these two videos is that Video_008 contains more movable element, e.g. grass, cloud. Another reason is that the camera using to record Video_008 is wobbling and make the frames of video unstable.

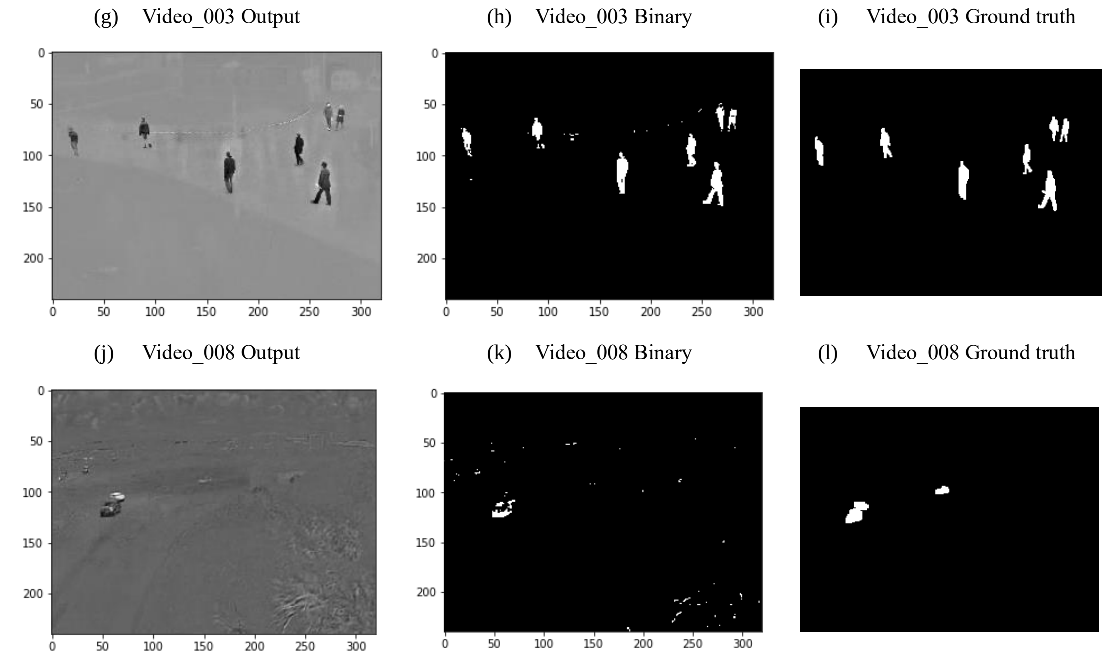
○ How to improve performance

For improvement, I also try to use the general primary component pursuit algorithm from Robust PCA paper[2]. The algorithm is in table 1. Refer to the paper for more detail.



```
Algorithm 1 (Principal Component Pursuit by Alternating Directions [32,51])
initialize: $S_0 = Y_0 = 0, \mu > 0$.
while not converged do
    compute $L_{k+1} = \mathcal{D}_\mu(M - S_k - \mu^{-1}Y_k)$;
    compute $S_{k+1} = \mathcal{S}_{\lambda\mu}(M - L_{k+1} + \mu^{-1}Y_k)$;
    compute $Y_{k+1} = Y_k + \mu(M - L_{k+1} - S_{k+1})$;
end while
output: $L, S$.
```

Table 1. Robust PCA algorithm



(g) Video_003 Output

(h) Video_003 Binary

(i) Video_003 Ground truth

(j) Video_008 Output

(k) Video_008 Binary

(l) Video_008 Ground truth

Figure 5. Performance evaluation of Robust PCA

Figure 5 shows the results of experiment using PCA. The average dice coefficients of testing sets are listed below:

- Video_003: Average dice Coefficient= **0.865**
- Video_008: Average dice Coefficient= **0.396**

It seems that Video_008 has been improved!

○ Summary

Based on the above experiments, both SVD and Robust PCA perform well as an unsupervised way to remove background. The latter method takes more time on computation and outperform on Video_008.

[2] https://arxiv.org/pdf/0912.3599.pdf