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]](#footnote-1).

**○ Plot matrices**

|  |  |  |
| --- | --- | --- |
| (a) Matrix illustration | (b) Video\_003 | (c) Video\_008 |
|  |  |  |

Figure 1. Matrix of flatten frames

**○ 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 Background | (b) Video\_003 People |
|  |  |
| (c) Video\_008 Background | (d) Video\_008 Car |
|  |  |

Figure 3. Background removal results

**○ Try to analysis performance**

|  |  |  |
| --- | --- | --- |
| 1. Video\_003 Output | 1. Video\_003 Binary | 1. Video\_003 Ground truth |
|  |  |  |
| 1. Video\_008 Output | 1. Video\_008 Binary | 1. 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:

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]](#footnote-2). The algorithm is in table 1. Refer to the paper for more detail.

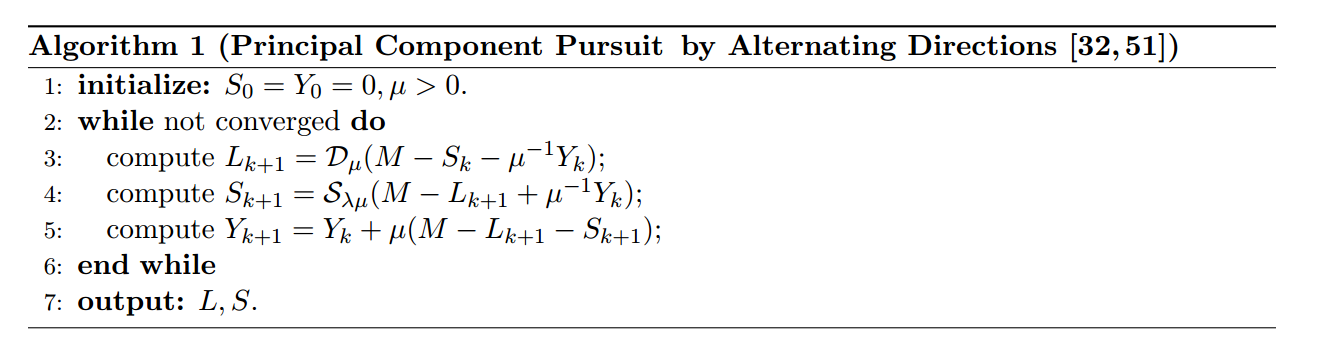


Table 1. Robust PCA algorithm

|  |  |  |
| --- | --- | --- |
| 1. Video\_003 Output | 1. Video\_003 Binary | 1. Video\_003 Ground truth |
|  |  |  |
| 1. Video\_008 Output | 1. Video\_008 Binary | 1. 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. The latter takes more time on computation.

1. http://bmc.iut-auvergne.com/?page\_id=24 [↑](#footnote-ref-1)
2. https://arxiv.org/pdf/0912.3599.pdf [↑](#footnote-ref-2)