Robust PCA for Moving Object Detection in Video

Chris HC Nguyen and James M Shihua
The Hong Kong University of Science and Technology

Codes used in this project is available at:
https://github.com/cchrisnguyen/VideoRPCA

Abstract

The utilization of Robust Principal Component Analysis (RPCA) for the detection and removal of moving objects in a video is exemplified by this project. By employing the RPCA algorithm, each frame of the image sequence is separated into a low-rank background matrix and a sparse moving object matrix, which enables the identification and elimination of moving objects from the background. This technique is efficacious in detecting and removing moving objects from the video background.

Introduction

RPCA is a mathematical technique used to separate a data matrix into two components: a low-rank component representing the underlying structure of the data, and a sparse component representing the noise or outliers in the data [1]. This technique is particularly useful for applications where the data contains outliers or is corrupted by noise.

In the context of video processing, RPCA can be used to separate a video sequence into a background component (stationary objects) and a foreground component (moving objects). By separating the video into these two components, it becomes possible to detect and remove the moving objects from the video background.

In this project, RPCA is augmented by a Lagrange multiplier and the gradient is calculated by its dual. This method is significantly more efficient than the traditional methods like alternating direction method of multipliers (ADMM).

[1] Candès, E. J., Li, X., Ma, Y., & Wright, J. (2011). Robust principal component analysis?. Journal of the ACM (JACM), 58(3), 1-37.

[2] Lin, Z., Chen, M., & Ma, Y. (2010). The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. arXiv preprint arXiv:1009.5055.

Data

The data used to test the RPCA algorithm is a video captured from a close–circuit television (CCTV), which has a resolution of 320x256 pixels and 1000 frames. To facilitate the RPCA operation, the first two dimensions of the video are flattened, and the video is represented by a matrix with a dimension of (1000, 81920).

Methodology

The algorithm of RPCA in this project is implemented with Intel optimized Python and Numpy library with Intel MKL, for optimal efficiency.

```
Input: Data matrix X of size m x n
       Regularization parameter λ
       Maximum number of iterations T

Output: Low-rank matrix L
        Sparse matrix S

Initialization: L_0 = S_0 = 0

For t = 1 to T do
    Solve for L_t and S_t using the
following update rules:
        L_t = argmin_L 1/2||X-L-S_{t-
1}||_F^2 + λ||L||_*
        S_t = argmin_S λ||S||_1 subject
to ||S||_0 <= k

    Update the Lagrange multipliers:
        μ_t = min(μ_{t-1} * rho, μ_max)

    Check for convergence:
        If ||L_t - L_{t-1}||_F /
||L_t||_F < tol and
           ||S_t - S_{t-1}||_F /
||S_t||_F < tol then
            break
        End if

End for

Output L = L_T and S = S_T
```

Augmented Lagrange Multiplier

The RPCA problem can be formulated as

$$\min_{A,E} \|A\|_* + \lambda \|E\|_1$$

Subject to

$$D = A + E.$$

ALM is applied by identifying

$$X = (A, E),$$
$$f(X) = \|A\|_* + \lambda \|E\|_1,$$
$$h(X) = D - A - E.$$

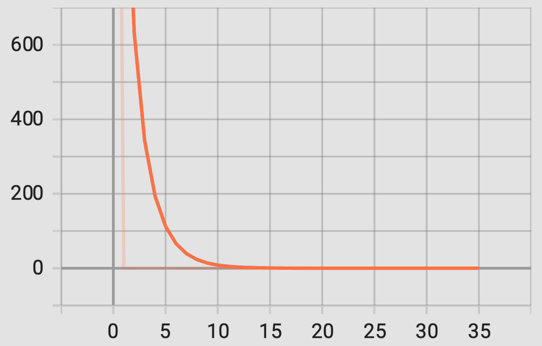
The Lagrange function is

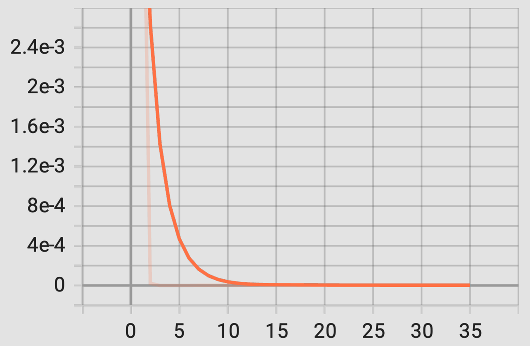
$$L(A, E, Y, \mu) = \|A\|_* + \lambda \|E\|_1 + \langle Y, D - A - E \rangle + \frac{\mu}{2} \|D - A - E\|_F .$$

Results

We test the RPCA algorithm with a maximum iteration of 100. The tolerance is set to 10^{-7}. In all test cases, the algorithm converges within 50 iterations and the tolerance criteria is satisfied.

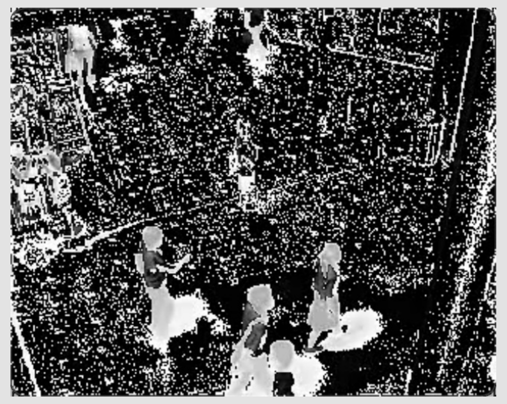
Convergence trend of the primal error and dual error:



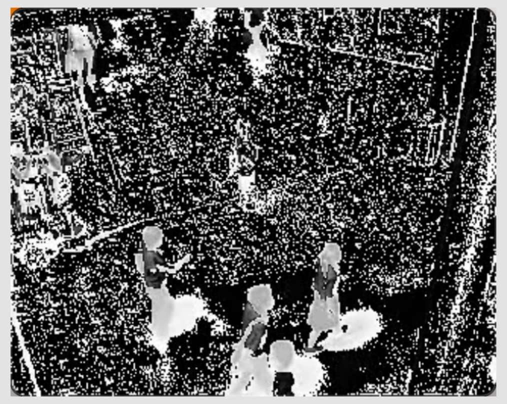


Extracted foreground of the

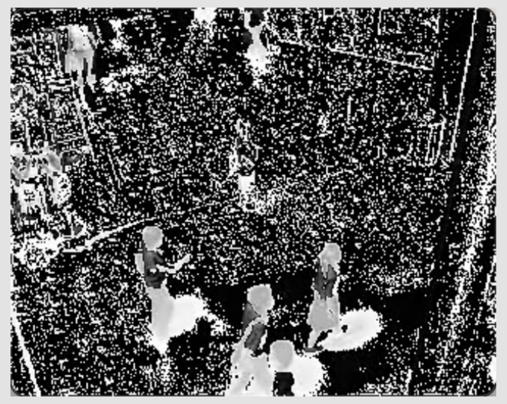
5^{th} iteration



10^{th} iteration



30^{th} iteration



35^{th} iteration

Extracted background from the

5^{th} iteration

10^{th} iteration

30^{th} iteration

35^{th} iteration

We clearly see that the proposed RPCA algorithm effectively separates the video into a foreground and background components. However, it can be observed that the foreground part contains much more noise than the background part. This might be due to that the foreground objects are time dependent and have weaker correlations.

Conclusion

In this project, we demonstrated a Robust PCA algorithm to extract the foreground and background components from a video. The video is represented by a matrix of N-pixels by M-frames. In our test case, N=81920 and M=1000. We find that the algorithm converges efficiently within tens of iterations and reached the convergence criteria at 35^{th} iteration, which is significantly performant than the ADMM algorithm (takes ~1000 iterations).

From the output foreground and background components, we see that the algorithm successfully extracted the moving objects (people) and stationary objects (floor, shelves) with a satisfactory level of accuracy. The background is very sharp and clean. However, the foreground part has a certain level of noise, and the reason might be a weaker correlations between the moving objects.