RPCA BASED MOVING ENTITY DETECTION FROM A MOVING CAMERA

Ahmad Chatha, Koku Egbetoke, Robert Kwiatkowski, Aaron Zakem

May 2, 2016

Abstract

Robust Principal Component Analysis (RPCA) works well for detecting moving objects in a series of images taken from a stationary camera by separating the static background from the dynamic objects. However, RPCA does not perform well at this task when the images were taken from a moving camera. This paper investigates the accuracy of two different approaches for detecting moving objects in images taken from a moving camera. The first approach is Tensor-RPCA, and performs reasonably well on short sequences of images. The second approach implements motion compensation techniques to generate the low rank and sparse matrix. The code is available on github.[1]

1. Introduction

Distinguishing the moving objects from the static background in a series of images or video has numerous useful applications in the field of computer vision, particularly with respect to security, topographical analysis of satellite images, and analyzing the movement of crowds in public spaces. Traditional approaches to moving object detection based on Principal Component Analysis perform moving object detection by separating the images series or video into low-rank and sparse components, with the low-rank component comprising the static background and the sparse component containing the moving objects. However, this approach suffers from degraded performance when the images are taken from a moving camera, as the movement of the camera is reflected in the dynamic nature of the background in the resulting images. Improved object detection for images taken from a moving camera would have applications to automated surveillance using panning or mobile cameras, as well as collision avoidance for automobiles and analysis of video feed taken from Unmanned Aerial Vehicles, both military and commercial. This paper explores two different approaches to modifying traditional Principal Component Analysis for detecting moving objects in video taken from a moving camera, and uses a short video taken from a commercial UAV for purposes of evaluation.

2. Previous Work

As set forth by forth by Candès et al., separating the static background from the moving objects in video taken from a stationary camera can be achieved using Robust Principal Component Analysis (RPCA) [1]. Pursuant to RPCA, the low-rank matrix L and sparse component S can be recovered by solving:

$$\min ||L||_* + \lambda ||S||_1$$
$$\text{subject to } L + S = M$$

where M is the matrix comprising the original images or video. RPCA will recover the low-rank and sparse components of M subject to certain "weak" assumptions regarding incoherence, namely

[1] https://github.com/rjk2147/RPCA-BASED-MOVING-ENTITY-DETECTION-FROM-A-MOVING-CAMERA/

that the low-rank component is not sparse, and that the sparsity pattern of the sparse component is selected uniformly at random [1].

In order to motivate exploration of modifications and alternatives to RPCA for moving object detection in the context of moving recorders, let us first examine the results of standard RPCA when applied to detect moving objects in a short video taken from a camera mounted to a commercially available UAV. The video focuses on a woman walking in a grassy field, and is recorded from a UAV that circles her while flying overhead (note, the video has been converted to greyscale for the experiments performed in this paper). We applied the fast method for solving RPCA as described in Aravkin, et al. [2] to the first 40 frames of the video to separate the sparse and low-rank components; a sample frame of the result is shown below:

M L S

As can be seen in the figure above, traditional RPCA does a relatively poor job of isolating the moving component of the image (the woman and her shadow) from the stationary background. Since the girl appears around the middle of the frame throughout the video, RPCA considers her to be part of the low rank matrix. However, we as humans have no difficulty figuring out what objects actually move in the non-static frames. Also, significant portions of the background remain in the sparse component S since they appear . Thus, modification to RPCA appears necessary to achieve satisfactory separation.

One approach to modifying RPCA to account for motion in the camera is to treat the video data as a tensor of rank 3, and to separate the tensor into it's sparse and low-rank components by solving a modified version of RPCA that utilizes tensor norms. Zhang et. al [3] describes a successful application of "Tensor RPCA" to recover the original video from a noisy version that has been corrupted by the addition of noise to randomly selected pixels. In section 3.1, we discuss the Tensor RPCA method described by Zhang and its application to moving object detection. In section 4.1, we discuss the results of applying Tensor RPCA to the test video shown used in our baseline evaluation of RPCA above.

A second approach to the moving object detection problem in the context of a moving camera is to attempt to determine the direction of motion for the camera, and compensate for the estimated motion when identifying moving objects in recorded video. Super-resolution contends with a similar problem when trying to produce a super-resolute image from a series of images each with their own camera motion, and there have been papers attempting to solve this problem using motion compensation [4].

3. Our Approach

3.1 Tensor RPCA Solution

The first approach we evaluated was Tensor RPCA as described in Zhang, et al. [2]. In this paper, multilinear rank and a related tensor nuclear norm was used to characterize informational and structural complexity of multilinear data, based on tensor-SVD (t-SVD). It was shown that

videos with linear camera motion can be represented and recovered using t-SVD and tensor nuclear norm penalized algorithm for video completion from missing entries.

Essentially the t-SVD decomposition is based on an operator theoretic interpretation of third-order tensors as linear operators on the space of oriented matrices. This notion can be extended recursively to higher order tensors. This decomposition, associated with the notion of tensor multi-rank and its convex relaxation to the corresponding Tensor Nuclear Norm (TNN) for completion and recovery of multilinear data, produced better results compared to earlier methods.

The Tensor Nuclear Norm (TNN) of a tensor A is the sum of the singular values of all the frontal faces of , where is obtained by taking the Fourier transform of A along the third dimension [2]. Proof that TNN is valid norm was presented in an earlier work by Semerci, et al. [5]. The 1,1,2 Norm of tensor A is the sum of the Frobenius norms of the tube fibers of A [2].

The key idea behind these methods is that under the assumption of low-rank of the underlying data (thereby constraining the complexity of the hypothesis space), it should be feasible to recover data (or equivalently predict the missing entries) from a number of measurements in proportion to the rank. Such analysis and the corresponding identifiability results are obtained by considering an appropriate complexity penalized recovery algorithm under observation constraints, where the measure of complexity, related to the notion of rank, comes from this particular factorization of the data.

Given M $\in \mathbb{R}^{n_1 \times n_2 \times n_3}$, the t-SVD of M is:

$$M = U * S * V^T$$

Where U and V are orthogonal tensors of size $n_1 \times n_1 \times n_3$ and $n_2 \times n_2 \times n_3$ respectively. S is a rectangular f-diagonal tensor of size $n_1 \times n_2 \times n_3$, and denotes the t-product. This decomposition is obtained by computing several matrix SVDs in the Fourier domain [2, 6].

Application of the proposed algorithm for video recovery from missing entries is shown to yield a superior performance over existing methods. Applied to the problem of RPCA for de-noising 3-D video data from sparse random corruptions, the t-SVD method showed superior performance compared to the matrix robust PCA method [2]. Zhang et al. successfully applied Tensor-RPCA to recover a basketball video (obtained from a laterally mobile camera) from a corrupted version of the video obtained by adding noise to pixels selected uniformly at random.

The minimization for Tensor-RPCA is stated in terms of the Tensor Nuclear Norm and $S_{1,1,2}$ norm, as given below:

$$\begin{aligned} &\min ||L||_{TNN} + ||S||_{1,1,2} \\ &\text{subject to } M = L + S \end{aligned}$$

ADMM is used to solve this convex optimization problem with the following update rules executed until convergence:

$$\begin{aligned} L^{k+1} &= \arg \min_L ||L||_{TNN} + \rho/2||L + S^k - M + W^k||_F^2 \\ S^{k+1} &= \arg \min_S \lambda||S||_{1,1,2} + \rho/2||L^{k+1} + S - M + W^k||_F^2 \\ Y^{k+1} &= Y^k + L^{k+1} + S^{k+1} - M \end{aligned}$$

Where $W = \rho Y$ [2]. The following algorithm is used to solve the minimization problem via the ADMM update rules set forth above.

Algorithm 1 Tensor ADMM

procedure L UPDATE
  for each frontal face i of L, S, M, and Y: **do**
    Let $\hat{A(i)}$ = Fast Fourier Transform of $-(S(i)^k - M(i) + Y(i)^k)$ along the third dimension.
    calculate $U(i)\Sigma(i)V(i)^T = SVD(\hat{A(i)})$.
    Threshold the diagonal values of $\Sigma(i)$ by $1/\rho$.
    Set corresponding frontal face i of tensors U, Σ, V
  end for
  Perform inverse Fast Fourier Transform on U, Σ, and V
  $L^{k+1} = U * \Sigma * V^t$, using t-product.
end procedure
procedure S UPDATE
  for each tube fiber i of L, S, M, and Y: **do**
    Let $B(i) = -(L(i)^{k+1} - M(i) + Y(i)^k)$
    $S(i)^{k+1} = (1 - \lambda/\rho||B(i)||_F)_+ * B$
  end for
end procedure
procedure Y UPDATE
  $Y^{k+1} = Y^k + \rho(L^{k+1} + S^{k+1} - M)$
end procedure

In order to apply Tensor-RPCA to the moving object detection problem, we implemented the algorithm above on a tensor of rank 3 comprising the greyscale video data for the UAV test video, and experimented with tuning the sparsity (λ) and convergence parameters to achieve satisfactory results.

3.2 Spatial Transformation Solution

We introduce a spatial transformation vector T and a sparse vector K. The purpose of T is to take into account affine linear transformations such as rotation, translation, scaling and shear which can happen between two sequential image sets. K incorporates a selection of these features and hence we impose L1 norm on it so that we can kill useless transformations. The basic idea is to make the image sets similar to each other and as a result act similar to a low rank matrix, which when subtracted from the original image should generate the Sparse matrix we desire. This idea was inspired by similar image transformations used for super-resolution by Chung, Haber, and Nagy [4].

Let $d_i \in R^{n\times 1}$ and $d_{i+1} \in R^{n\times 1}$ be two image sets. Let $K \in R^{1\times 6}$ be the sparse transformation selection vector and $T \in R^{6\times 1}$ be the transformation vector. The transformation of a pixel j from frame d_i to d_{i+1} looks something like this:

$$\begin{pmatrix} d^j_{i,1} \\ d^j_{i,2} \end{pmatrix} = \begin{pmatrix} \gamma_{i,1} & \gamma_{i,2} \\ \gamma_{i,3} & \gamma_{i,4} \end{pmatrix} \begin{pmatrix} d^j_{i+1,1} \\ d^j_{i+1,2} \end{pmatrix} + \begin{pmatrix} \gamma_{i,5} \\ \gamma_{i,6} \end{pmatrix}$$

Here $\gamma_{i,1}$ to $\gamma_{i,4}$ incorporate rotation, scaling and shear while $\gamma_{i,5}$ and $\gamma_{i,6}$ are for translations. Since not all of the transformations will be active at any given pixel, the K sparse vector will choose the best ones. Given the intuition behind T and K, we arrive at our optimization objective function:

$$\epsilon = \min_{T,K} \frac{1}{2n}||d_i - d_{i+j}KT||_2^2 + \mu||K||_1 + \lambda||T||_2^2$$

Comparing this to the regular RPCA optimization object, we can see that ϵ behaves like the Sparse matrix, $d_{i+j}KT$ is the low rank and d_i is the original M we are trying to decompose. Differentiating with respect to K and T, we arrive at the following updates:

$$\hat{T} = ((d_{i+j}K)^T(d_{i+j}K) + \lambda I)^{-1}((d_{i+j}K)^T d)$$

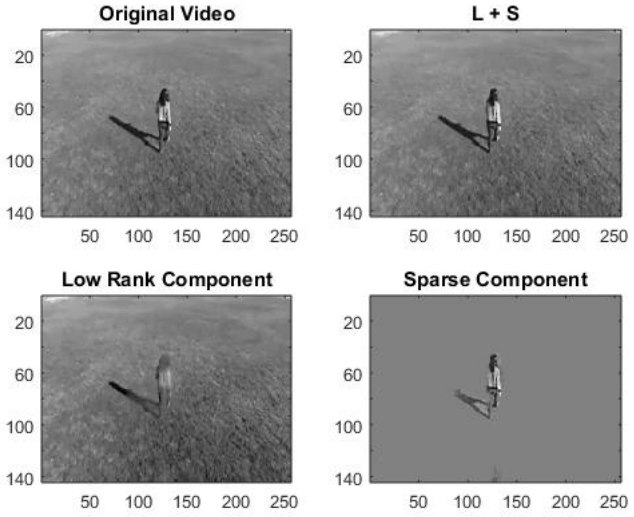
$$\hat{K} = \frac{T^T(T({d_{i+j}}^T d) - sign(K)\mu)T^T(TT^T)^{-1}}{(T^T T)({d_{i+j}}^T d_{i+1})}$$

Solving for T and K in an iterative fashion eventually yields a transformation which minimizes the difference between successive frames d and d_{i+j} in a video.

4. Results

4.1 Tensor RPCA Results

As discussed in section 3.1 above, the experiment reported in the Zhang et al. paper involved recovering a recording of a basketball game taken from a laterally mobile camera after the introduction of random pixel noise. We were able to recreate this de-noising experiment on the UAV test video using the parameters described by Zhang et al. [2]. However, detection of moving objects in the UAV video required tuning the sparsity coefficient λ, as well as the convergence conditions, in order to produce reasonably successful results. Zhang et al. reported that the optimal value for λ for Tensor RPCA was $\frac{1}{\sqrt{\max(n_1, n_2)}}$ in the context of the denoising experiment (with n_1 and n_2 as the number of rows and number of columns in the video frame, respectively). After experimentation, we found reasonably successful results in terms of separating the moving objects from the stationary background by scaling the value of λ to $\frac{.215}{\sqrt{\max(n_1, n_2)}}$, and running Tensor RPCA on the first 40 frames of the test video. The results are shown below:



As can be seen in the figure above, Tensor RPCA does not significantly improve on standard RPCA in terms of eliminating the moving object from the low-rank component L. However, Tensor RPCA shows significant improvement in isolating the moving object (the woman and portions of her shadow) in the sparse component S. The results are imperfect, but do show improvement over the results using standard RPCA. However, it should be noted that the results degraded significantly when Tensor RPCA was performed over longer segments of the test video, and required substantially more iterations to converge. When executed for the first 200 frames of the test video, Tensor RPCA was unable to isolate the moving objects in the sparse component S when the direction of motion of the moving object changed suddenly or differed substantially from the prevalent direction of motion throughout the video segment. Thus, it appears that Tensor RPCA produces better results for the moving object detection objective when applied to shorter segments of video data. Further, performing RPCA on longer video segments requires retuning

the sparsity coefficient λ as well as the convergence conditions.

4.2 Spatial Transformation Results

Due to the nature of our objective function's defining the sparse representation by the error in our objective function, large lambdas were required to see reasonable sparse representations. With lower regularization parameters the low rank is too close to the original, and the error too low that the sparse is either uniform about the entire image or almost entirely blank. These images were produced in 5 iterations using matlab with regularization parameters $\lambda = 1e4$ and $\mu = 1e4$, a j=25, on a 40 frame sample.

Figure 1: Original

Figure 2: Sparse

Figure 2 shows the sparse result which is equal to the error obtained after minimization. Figure 2 has also undergone some image manipulation to increase contrast, etc. so that the results are easier for the human eye to see. As you can see, it definitely outlines the girl and her shadow. However, upon further examination is it obvious that it also captures a lot of the grass patches which should be part of the background. Here, we omit the low rank part since our objective function does not model it explicitly. Moreover, the sparse part is enough to identify the moving objects in the frames.

The result is definitely more sparse than what was obtained with regular RPCA but is sparse in the foreground as well as background. As a result, the figure of the girl is not as sharply defined as we would like it to be. At a certain level, the optimization object reduces to the subtraction of two images and the sparse matrix is the result of the subtraction. If the sole aim is to get the movement of anything between the sets of frames, then this objective function achieves that well.

The L1 norm of the sparse matrix for the first frame came out to be $1.0129e+05$. Comparing this with that of tensor RPCA ($8.4405e+04$), we see that Tensor RPCA generates a more sparse matrix which captures only the moving object.

5 Future Work

There is an inherent difficulty in applying standard RPCA to moving camera frames. The object function does not map directly to the problem since the low rank part is ill-defined. While Tensor-RPCA achieves surprisingly good results, the spatial transformation technique fell a bit short. We suspect the reason has something to do with not using the nuclear norm in the objective function. So a logical thing to do is modify the minimization object so that it incorporates the nuclear norm on the transformed image. The new minimization function should look something like this:

$$\epsilon = \min_{T,K} \frac{1}{2n}||d_i - d_{i+j}KT||_2^2 + \tau||d_{i+j}KT||_* + \mu||K||_1 + \lambda||T||_2^2$$

While it does make the updates more complex, we think this will achieve better results than our naive implementation. Additionally, its presence makes the objective function closer to that of RPCA by having an explicit low-rank term.

In addition to improving the estimation of the low-rank term, the calculation of the sparse matrix as a function of the error is a method susceptible to many issues as well as one that re-

quires too much cross validation. A better method would involve solving for a sparse matrix that was the product of transformations as well as the low-rank. Implementing this technique would make this equation resemble more and more standard RPCA with the addition of transformations.

Another interesting thing we noticed during our experimentation was that the ADMM solution to RPCA and the fast method for solving RPCA as described in Aravkin, et al. [2] give different results for our example footage. The ADMM solution to standard RPCA gives much worse results that Fast RPCA. It generates an extremely sprase matrix in which the moving object is barely visible. We believe that the difference in results is due to fast RPCA having a slightly different objective function which minimizes the $\max(||L||_*, ||S||_1)$ instead of simply minimizing $||L||_* + ||S||_1$. If we use this slight modification in Tensor RPCA, then we might be able to achieve even better results.

With respect to Tensor RPCA, substantial experimentation with the sparsity coefficient λ and convergence conditions was required to achieve reasonably acceptable results, and optimal values would change depending on the number of frames in the tensor. Future work would attempt to automatically determine optimal parameters for Tensor RPCA based on the characteristics of the input tensor M. Further, our experiment was performed on low-resolution greyscale video data represented in a tensor of rank 3. Performing Tensor RPCA on high resolution video data, or full color video data represented in a tensor of rank 4, could potentially yield improved results.

Finally, all our experimentation and results are focused on the UAV footage example and they may not be relevant to other datasets. In order to figure out if they generalize well, one must apply these techniques to similar looking video examples.

6 Conclusion

Identifying moving objects in footage obtained from a moving camera is considered to be an easy task for humans. Various background subtraction algorithms exists to separate the moving foreground from the static background but they all require a static camera frame. Robust principal components analysis (RPCA) is one such algorithm which does an excellent job of decomposing the foreground and background into a sparse and low-rank part respectively. In this paper, we explored if using standard RPCA and variations on it can give relatively good results on a non-static footage.

Our approach applies two fairly standard techniques, for their respective domains, to a problem which they have never been applied. As a result, we have interesting and somewhat unique results. Tensor RPCA has results in the sparse matrix very close results of standard RPCA on stationary camera problems, whereas the Low-Rank is still not completely clear. Spatial Transformation, however, results in a slightly worse reconstruction of the sparse component both contains more background and less foreground than Tensor RPCA. While neither of these approaches are objectively perfect they are both superior to any other method yet applied to the problem.

References

1 Candès, E. J., Li, X., Ma, Y., and Wright, J. (2011, May). Robust Principal Component Analysis? In Journal of the ACM, Vol. 58, No. 3 (pp. 11-37).

2 Aravkin, A., Becker, S., Cevher, V., and Olsen, P. (2014). A Variational Approach to Stable Principal Component Pursuit. arXiv preprint arXiv:1406.1089.

3 Zhang, Z.M., Ely, G., Aeron, S., Hao, N., Kilmer, M.E. (2014). Novel methods for multilinear data completion and de-noising based on tensor-SVD. IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

4 J. Chung , E. Haber and J. Nagy, Numerical methods for coupled superresolution (2006). Inv. Probl., vol. 22, pp. 1261-1272.

5 O. Semerci, N. Hao, M. E. Kilmer, and E. L. Miller (2012). An iterative reconstruction method for spectral CT with tensor-based formulation and nuclear norm regularization. Second International Conference on Image Formation in X-Ray Computed Tomography.

6 C. Martin, R. Shafer, and B. LaRue (2013). An order-p tensor factorization with applications in imaging. SIAM Journal on Scientific Computing, 35(1):A474–A490.