Low-Rank Spatio-Temporal Video Segmentation

Alasdair Newson
alasdairnewson@gmail.com
Duke University
Durham, NC, USA
Mariano Tepper
http://www.marianotepper.com.ar
Guillermo Sapiro
http://www.ee.duke.edu/faculty/guillermo-sapiro

Abstract

Robust Principal Component Analysis (RPCA) has generated a great amount of interest for background/foreground estimation in videos. The central hypothesis in this setting is that a video's background can be well-represented by a low-rank model. However, in the presence of complex lighting conditions this model is only accurate in localised spatio-temporal regions. Following this observation, we propose to model the background with a piecewise low-rank approximation. To achieve this, we introduce the piecewise low-rank segmentation problem. Starting from a carefully designed cost function which assesses the low-rank coherence of two video regions, the segmentation is obtained with an efficient graph-clustering algorithm. We show that this segmentation, when used to establish a local RPCA per segment, leads to improved quantitative and qualitative results for background/foreground estimation in challenging videos.

1 Introduction

The technique known as Robust Principle Component Analysis (RPCA) has recently become an extremely popular method to solve a wide range of computer vision problems from background estimation to face recognition. The goal of this technique is to decompose a matrix into a low-rank and a sparse component. In the case of background estimation these two components correspond, respectively, to the background and the foreground. The low-rank requirement means that the background estimation is robust to global lighting changes. However, in challenging scenarios the global low-rank hypothesis is not necessarily verified, in particular in scenarios which contain local lighting changes. In the standard RPCA approach, this is dealt with by increasing the rank of the background. However, this is dangerous, as it increases the chance of including a foreground element in the background, especially if the foreground is static for a short while. In this case, it is better to describe the background using a *piecewise* low-rank representation, where each local representation can be of lower rank than a global model.

Accordingly, we present and address a new problem here: low-rank video segmentation, where we seek to segment a video into regions whose backgrounds are each well-represented

Pages 103.1-103.12
DOI: https://dx.doi.org/10.5244/C.29.103

by low-rank matrices, when considered independently. The motivating goal of this segmentation is to improve the background/foreground estimation which can subsequently be carried out, however the segmentation is interesting in its own right and we show that it can be applied to other computer vision problems such as detecting scene transitions.

The segmentation problem is made particularly difficult by the fact that we want to segment the video domain with respect to a criterion (how well a region is represented with a low-rank approximation) which itself can only be determined after an optimisation. We address this segmentation challenge by reformulating the problem in terms of graph clustering. We propose a cost function between two adjacent regions which reflects how well the backgrounds of these regions are approximated by a single low-rank matrix. Throughout this paper, we refer to regions which are well-represented in this manner as *coherent* regions.

The main contributions of this paper are the following:

- we introduce and describe the new problem of low-rank video segmentation, to find regions where the background hypothesis of RPCA is more appropriate;
- we consider the problem from a graph clustering perspective and introduce a cost function which determines how well two regions respect the low-rank hypothesis;
- with quantitative and qualitative comparisons to previous work, we show that the use of this segmentation improves background/foreground separation in challenging situations.

1.1 Related work

The problem which we address includes elements of segmentation and background estimation, which are vast fields of computer vision in their own right, and each naturally possesses a very large literature. We present here the contributions that are most relevant to our context.

The separation of videos into background and foreground components is an important pre-processing step of many computer vision problems (tracking, object recognition, *etc*.). Many early approaches used simple differences in greyscale to separate background from foreground, while trying to adapt to changes in lighting conditions, for example with the Kalman filter [13]. Wren et al. [17] use a single Gaussian distribution to model the background. A significant step forward was made by Stauffer and Grimson [16] who proposed to model the background as a mixture of Gaussian distributions, which are dynamically updated. This is still a very popular method due to its generality and flexibility. More recently, Candès et al. [3] introduced a convex optimisation problem which can be applied to background/foreground estimation. The main goal of this work is to separate an input matrix, which represents the video data, into two components: a low-rank component (the background) and a sparse component (the foreground).

The problem of segmentation is also a very old and important topic in the image processing and computer vision communities. An important early contribution was made by Mumford and Shah [8] who proposed a functional which basically evaluates how "good" a given piece-wise constant approximation of an input image is. The optimisation of this functional [1] provides a segmentation solution. Kass et al. [5] introduced another well-known segmentation model: active contours. This model evaluates a given segmentation curve with respect to the boundary smoothness and also to its proximity to object boundaries. Caselles et al. [4] extended this model, using tools from geometric curve evolution. Shi and Malik [14] recast the segmentation problem as a *graph clustering* problem, introducing the normalised cut criterion. Another common approach to segmentation is that of region merging [9] or splitting [10]. In this work, we draw inspiration from these ideas to achieve our goal of low-rank video segmentation.

The more recent subject of video segmentation is usually concerned with segmenting *moving objects* in videos, which is quite a different goal to ours. An estimation of the background itself is not important, only identifying a region corresponding to a moving object. Brox et al. [2] proposed to first establish a dense optical flow, and then to cluster the motion vectors, which provides the segmentation. Papazoglou and Ferrari [10] also determine optical flow vectors and optimise a discrete cost function which indicates whether a pixel is inside or outside of a moving object. Again, it must be emphasised that the goal of these works is very different than ours, and also that they consider situations such or fast moving backgrounds which often do not apply to background estimation as we look at in this work.

2 A piece-wise low-rank video background model

We first set out the notation and introduce the concepts required for RPCA as described by Candès et al. [3]. We denote the input video data matrix with $\mathbf{X} \in \mathbb{R}^{m \times n}$. Each column of this matrix corresponds to the greyscale information contained in one frame, or features derived from it, in a vectorised form. Each frame contains m pixels, and there are a total of n frames in our video. The goal of RPCA is to decompose $\mathbf{X}$ into a low-rank and a sparse component. The former corresponds to the background, and the latter to the foreground. Formally, we have $\mathbf{X} \approx \mathbf{L} + \mathbf{S}$, where $\mathbf{L}$ is the low-rank matrix and $\mathbf{S}$ is the sparse matrix. This can be formulated as an optimisation problem:

$$\underset{\mathbf{L},\mathbf{S} \in \mathbb{R}^{m \times n}}{\operatorname{argmin}} \frac{1}{2}\|\mathbf{X} - \mathbf{L} - \mathbf{S}\|_F^2 + \lambda_* \operatorname{rank}(\mathbf{L}) + \lambda \|\mathbf{S}\|_1, \tag{1}$$

where λ_* and λ are scalar optimisation parameters, $\|\cdot\|_F$ is the Frobenius matrix norm and $\|\cdot\|_1$ is the ℓ^1 matrix norm, which induces sparsity in the foreground matrix.

Unfortunately, using the rank makes this problem non-convex, meaning that the quality of the solutions will be greatly affected by the initialisation procedure. A key insight of RPCA is to use the *nuclear norm* as a convex surrogate for the rank of a matrix. Thus a convex problem is then reformulated as:

$$\min_{\mathbf{L},\mathbf{S} \in \mathbb{R}^{m \times n}} \frac{1}{2}\|\mathbf{X} - \mathbf{L} - \mathbf{S}\|_F^2 + \lambda_* \|\mathbf{L}\|_* + \lambda \|\mathbf{S}\|_1, \tag{2}$$

where $\|\mathbf{L}\|_* = \sum_i \sigma_i(\mathbf{L})$ is the nuclear norm of $\mathbf{L}$ and $\sigma_i(\mathbf{L})$ is the i^{th} singular value of $\mathbf{L}$.

2.1 Piece-wise low-rank video segmentation

We recall that, in the presence of complex lighting conditions, the RPCA model is only *locally* accurate, and this observation is the main motivation of our piece-wise low-rank video segmentation problem. Our goal, then, is to identify spatio-temporal regions in the video which are each well-represented by a low-rank background, plus a sparse foreground (i.e. coherent regions). This new model provides greater robustness when dealing with videos with complex lighting conditions and foreground objects which are temporarily motionless.

We note the desired partitioning as $\mathcal{P} = \{\Phi_i\}_{i=1\ldots|\mathcal{P}|}$, where each Φ_i is a spatio-temporal region of the video. In the most general setting, the desired partitioning is the solution of the following minimisation problem:

$$\min_{\mathcal{P}} \sum_{i=1}^{|\mathcal{P}|} \min_{\mathbf{L}_i,\mathbf{S}_i} \frac{1}{2}\|\mathbf{X}_i - \mathbf{L}_i - \mathbf{S}_i\|_F^2 + \lambda_* \|\mathbf{L}_i\|_* + \lambda \|\mathbf{S}_i\|_1, \tag{3}$$

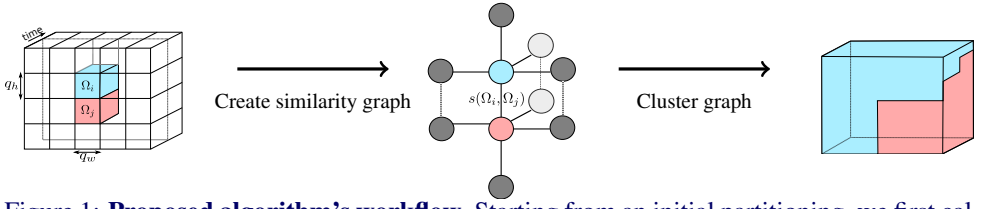


Figure 1: **Proposed algorithm's workflow.** Starting from an initial partitioning, we first calculate the cost of merging each adjacent region, using a low-rank approximation. From these costs, we build and then cluster a weighted graph, which produces the desired segmentation.

where $\mathbf{X}_i$ corresponds to the video data in the region Φ_i, and $\mathbf{L}_i$ and $\mathbf{S}_i$ are the associated decomposition matrices.

This problem contains two optimisations, one over the possible partitionings, and one corresponding to the RPCA itself. Optimising over the discrete set of possible partitions makes the problem non-convex and thus more difficult to solve.

As mentioned in Section 1.1, a wide range of approaches to segmentation exist. At a first glance, the framework of this problem recalls the Mumford-Shah functional, however the criteria to optimise (total variation, boundary length *etc.*) are replaced by quantities which themselves require the resolution of an optimisation problem. Consequently, a search for the optimal solution with a variational approach is likely to be very difficult. We look to other approaches in the literature to tackle this challenging problem. In particular, we use the ideas of region merging and graph clustering [14].

3 Proposed approach

The proposed algorithm is based on the idea of merging an initial set of small spatio-temporal regions into the largest regions possible which are coherent. Firstly, we set up a regular grid on our video domain which defines the *initial* regions. We denote these regions with Ω_i. Each Ω_i is a spatio-temporal block of size $q_w \times q_h \times t$. For an illustration, see Figure 1.

We then transform this grid into an undirected, weighted graph. Each vertex of this graph corresponds to a single Ω_i and the weights between two vertices represent some cost of merging the two regions. Clearly, a critical part of the algorithm is how to carefully design a reliable and meaningful criterion which indicates how coherent two regions are with each other. This cost should penalise the merging of incoherent regions. Finally, the graph is clustered and the output segmentation corresponds to the clusters of the graph.

3.1 Creating and clustering the graph

The first step in our algorithm is the creation and clustering of the graph, which provides the output segmentation. From the initial grid, we create an undirected, weighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of all the vertices of the graph, and $\mathcal{E}$ are the edges between these vertices. Each vertex corresponds to a single region in the initial grid. We shall refer to a vertex in the graph with Ω_i, in the same manner as a region of the video domain, even though this is an abuse of notation, strictly speaking. Accordingly, the weight (or cost) of an edge between two connected vertices is defined with Equation (7), which we will explain in

detail further on. We choose a six-connectivity for our graph: each region is connected to the region directly left, right, above, below, before, and after itself.

The graph clustering process which we will use takes a graph's *similarity matrix* as an input. This similarity is obviously the contrary of the cost; the greater it is, the more coherent the two regions are. We define the similarity of two regions/vertices in a classical fashion:

$$s(\Omega_i, \Omega_j) = \exp(-d(\Omega_i, \Omega_j)^2/(2\beta^2)) + \varepsilon, \tag{4}$$

where β is a kernel width and ε is a very small scalar which avoids two adjacent regions being disconnected in the similarity matrix. The cost function $d(\Omega_i, \Omega_j)$ will be explained in Section 3.2.

We now have the elements to cluster the graph. We use a method by Zelnik-Manor and Perona [19] which automatically finds an optimal number of regions, given a maximum number of regions.

3.2 A reliable criterion for region merging

We now describe our region merging cost function. Recall that this cost function should be designed so that the cost of merging two coherent regions is very *low*. For this, consider two regions Ω_i and Ω_j, whose coherence we wish to assess. Let $\Omega_{i\cup j}$ be the concatenation of the two regions. We propose to use the RPCA decomposition itself as an indicator of the coherence of the regions. The most direct way to proceed would be to apply the RPCA to Ω_i, Ω_j, and $\Omega_i \cup \Omega_j$, and observe the ranks of the background components of each decomposition. Unfortunately, the rank is a relatively unstable and sensitive to changes in the optimisation parameters. Alternatively, we could use the nuclear norm as a criterion. Again, there is no obvious way to compare $\|\mathbf{L}_i\|_*$, $\|\mathbf{L}_j\|_*$ and $\|\mathbf{L}_{i\cup j}\|_*$, as the nuclear norm is non-separable.

To design a more reliable merging criterion, we propose to modify the RPCA decomposition of each region. We redefine the low-rank decomposition of a region Ω_i as:

$$\{\mathbf{L}_i, \mathbf{S}_i\} = \underset{\mathbf{L},\mathbf{S}}{\operatorname{argmin}} \tfrac{1}{2}\|\mathbf{X}_i - \mathbf{L} - \mathbf{S}\|_F^2 + \lambda\|\mathbf{S}\|_1 \quad \text{subject to} \quad \operatorname{rank}(\mathbf{L}) \leq r. \tag{5}$$

This formulation has two major advantages. Firstly, we have fixed the maximum rank of $\mathbf{L}$ in the decomposition. This means that the nuclear norm does not play a role in the energy of the decomposition, making comparisons more reliable. Secondly, the rank-constraint r is more easily interpretable than the nuclear norm weight λ_*. This is a significant advantage, as in practice these parameters have a great influence on the decomposition and there is no trivial way to set them.

The new rank-constrained problem is non-convex; giving up convexity is the price to pay for having a decomposition which leads to a reliable merging criterion. Nevertheless we can address it using an alternating approach. Accordingly, we perform a minimisation firstly over $\mathbf{L}$, and then over $\mathbf{S}$. The first problem can be addressed [12] by decomposing the low-rank matrix into the product of two submatrices $\mathbf{L}_i = \mathbf{U}_i\mathbf{V}_i$, with $\mathbf{U}_i \in \mathbb{R}^{m\times r}$, $\mathbf{V}_i \in \mathbb{R}^{r\times n}$, and alternately minimising the Frobenius norms of the submatrices. The second may be solved with soft thresholding. We give details of these minimisation processes in Section 3.4.

Let e_i denote the quadratic error of the low-rank/sparse approximation:

$$e_i = \|\mathbf{X}_i - \mathbf{L}_i - \mathbf{S}_i\|_F^2. \tag{6}$$

Algorithm 1: Alternating minimization scheme for solving Equation (5), with $r = 1$

input : Data $\mathbf{X}_i$ to decompose, parameter λ, step size τ.
output : Sparse matrix $\mathbf{S}_i$, rank one background matrix $\mathbf{L}_i$
$\mathbf{S} \leftarrow \mathbf{0}$
Initialise $\mathbf{u}$ as the temporal median of $\mathbf{X}_i$
repeat
  $\mathbf{v} \leftarrow \left(\mathbf{u}^{\mathrm{T}}\mathbf{u}\right)^{-1}\left(\mathbf{u}^{\mathrm{T}}(\mathbf{X}_i - \mathbf{S})\right)$
  $\mathbf{u} \leftarrow \left((\mathbf{X}_i - \mathbf{S})\mathbf{v}^{\mathrm{T}}\right)\left(\mathbf{v}\mathbf{v}^{\mathrm{T}}\right)^{-1}$
  $\mathbf{S} \leftarrow \mathrm{shrink}_{\lambda}(\mathbf{S} + \tau(\mathbf{X}_i - \mathbf{u}\mathbf{v}))$ // $\left(\mathrm{shrink}_{\lambda}(\mathbf{A})\right)_{p,q} = \begin{cases} 0 & \texttt{if } (\mathbf{A})_{p,q} < \lambda \\ (\mathbf{A})_{p,q} - \lambda & \texttt{otherwise} \end{cases}$
until *convergence*
$\mathbf{S}_i \leftarrow \mathbf{S}$; $\mathbf{L}_i \leftarrow \mathbf{u}\mathbf{v}$

We propose to use this error for the cost of merging two adjacent regions. Assuming that the sparse foreground elements appear equally in $\mathbf{S}_i$, $\mathbf{S}_j$, and $\mathbf{S}_{i\cup j}$, logically the energy due to the ℓ^1 term $\|\mathbf{S}\|_1$ will have no influence on the merging decision. For these reasons we argue that if the two sub-regions of $\Omega_{i\cup j}$ are coherent, then this cost will be very low, which is our goal. Indeed, the Frobenius norm is separable, which means that for two adjacent, coherent regions (two regions where the low-rank assumption is accurate) we should have $e_i + e_j = e_{i\cup j}$. Thus the quadratic error provides a meaningful comparison of the coherence of two regions.

We now give the formal definition of the cost of merging two regions:

$$d(\Omega_i, \Omega_j) = \frac{\left|e_i + e_j - e_{i\cup j}\right|}{\phi_{i\cup j}}, \tag{7}$$

where $\phi_{i\cup j}$ is the following scaling factor:

$$\phi_{i\cup j} = \begin{cases} 1 & \text{if } \Omega_i \text{ and } \Omega_j \text{ are spatially adjacent} \\ q_w q_h \sigma^2 & \text{if } \Omega_i \text{ and } \Omega_j \text{ are temporally adjacent.} \end{cases} \tag{8}$$

We discuss this factor $\phi_{i\cup j}$ in detail in the next section. We recall that q_w and q_h are the spatial width and height of the initial blocks.

3.3 Comparing spatial and temporal merging fairly

In Equation (7), we have assumed that when we try to merge two regions which are coherent, the quadratic errors of the two regions will be similar, so that our merging cost is reliable. Unfortunately, when we are merging two temporally adjacent regions, this assumption is not quite correct.

Consider two adjacent regions Ω_i and Ω_j which contain the same static background and no lighting changes, plus some Gaussian noise of (estimated) variance σ. Let us also suppose that $r = 1$, which is reasonable in this case. In such a setting, the expected value of the merging cost is very different if we merge spatially or temporally. For spatially adjacent regions, the cost will be zero. However, when we merge two temporally adjacent regions, we have:

$$\mathbb{E}\left(\left|e_i + e_j - e_{i\cup j}\right|\right) \approx q_w q_h \sigma^2. \tag{9}$$

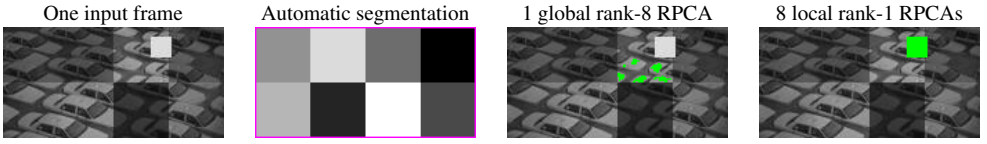


Figure 2: **Foreground estimation in a challenging, synthetic video using the proposed segmentation. The estimated foreground is highlighted in green.** In this case, the background contains several regions whose contrast varies independently, and a foreground component (the white square). We detect the coherent regions, indicated in the second image with the grey squares, and use them to carry out localised RPCA decompositions.

These results are proven in supplementary material available on the project webpage. Intuitively, the costs in the spatial and temporal merging situations are dissimilar for the following reason. Two temporally adjacent, coherent regions, contain different (noisy) observations of the same variables/pixels. In the case of spatial adjacency, we have twice the number of variables, without increasing the number of observations.

In order to counter this effect, and make sure that merging is not favored in either the spatial or temporal directions, we need to set the scaling factor $\phi_{i\cup j}$ of Equation (8) correctly. Given the previous reasoning, we scale the temporal merging with $\phi_{i\cup j} = q_w q_h \sigma^2$. In the spatial merging case, we do not scale the cost function, i.e., we set $\phi_{i\cup j} = 1$.

3.4 Minimisation of Equation (5) and choice of r

It is clear that the most expensive operations of our algorithm are the local RPCA decompositions in each spatio-temporal region. To speed this up, we propose to choose $r = 1$ in Equation (5). In fact, this restriction makes sense; in one coherent region there should logically be only one "true" background. We distinguish this special case by denoting the matrices $\mathbf{U}_i$ and $\mathbf{V}_i$ with lowercase letters, since they are now vectors, so that $\mathbf{L}_i = \mathbf{u}_i \mathbf{v}_i$. We have used this speedup in all of our experiments, with good results. The minimisation algorithm to decompose $\mathbf{X}$ in this case is shown in Algorithm 1.

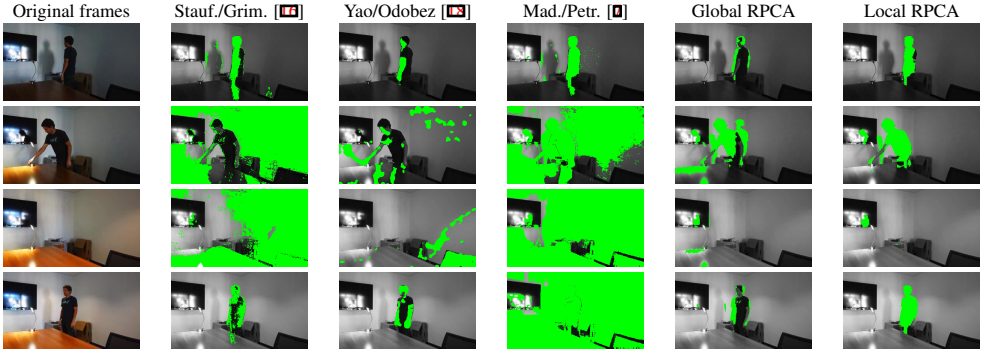
3.5 Segmentation overlap

The choice of a non-overlapping initial grid, imposes a lower limit on the granularity of the segmentation, which can be problematic, especially in the temporal direction. The main goal of our segmentation is to carry out background/foreground estimation locally in each region. For this purpose, we do not need the segmentation to be pixel-precision. Instead, we dilate each segmented region to a half of the initial grid precision and perform the final low-rank/sparse decomposition in these dilated regions. Then, for the pixels in overlapping regions, we choose the model which best fits the data for that pixel.

4 Experimental results

We now show some results of our segmentation algorithm on synthetic and real data. In particular, we show that commonly used background/foreground estimation algorithms and the standard RPCA fail when faced with difficult situations including both variable lighting

Qualitative (visual) evaluation



Quantitative evaluation (recall, precision and f1-score)

	Stauffer/Grimson [16]	Yao/Odobez [13]	Maddalena/Petrosino [9]	Global RPCA	Local RPCA
Recall	70.83	67.88	61.69	50.27	**74.97**
Precision	39.35	68.16	05.97	60.57	**81.26**
f1-score	50.60	68.02	09.97	54.94	**77.99**

F1-score as function of time

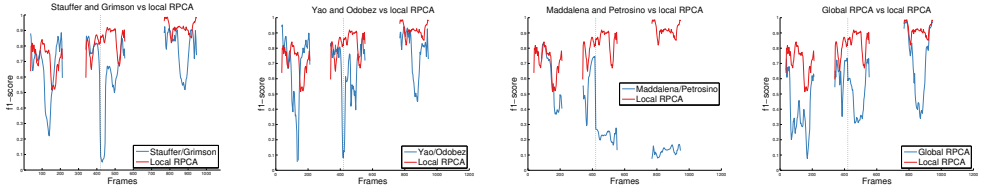


Figure 3: **Foreground estimation in variable lighting conditions.**. We segment the video using the proposed algorithm, and carry out a local, rank one RPCA in each region. We compare with three other well-known background subtraction algorithms [7, 16, 13]. The black dotted vertical line in the f1-score plots indicates a local lighting change when the foreground person is present. The segmentation may be seen on the project webpage.

and foreground which may be static for a while. The proposed algorithm exhibits significant improvement over these other approaches in such scenarios. The results in this section, and other video results, are available on the project webpage[1].

Firstly, in Figure 2, we show a synthetic example where it is impossible to correctly separate the foreground and background with a standard, global RPCA. The video contains eight regions which are each illuminated independently, and a sparse component which moves around before finally staying in one position for a short while. In this example, the initial blocks are chosen to span the entire temporal extent of the video. Our algorithm finds the correct segmentation. The main point here is that *whatever* the rank of the global approximation, the classical RPCA will not be able to recover the background and foreground correctly. Let us further illustrate this issue with a real example.

In general, the background subtraction literature uses examples which are simple in terms of varying lighting conditions. Either no lighting changes happen, or they are relatively

[1] https://goo.gl/JSwVmt

Figure 4: **Detection in an example from the database of [6].** In this case, the lighting variations are global, so the proposed algorithm performs similarly to the standard RPCA.

global. In Figure 3 we provide a more complex example. A person is walking in and out of the video, while different lights are turned on (a lamp, and then an overhead light). We annotated the foreground of each frame of this video by hand to provide a ground truth. We segmented the video with our algorithm, which was able to locate the different points in time and space of the lighting changes. We then carried out a local rank-one RPCA in each region.

We compared the resulting local RPCA foreground detection with three other popular background subtraction methods from the literature [2, 16, 18], and also with the results of the global RPCA. Our local background models are all of rank-one. We used the implementations of the background subtraction library from [15]. Figure 3 analyses four frames of the video, which clearly illustrate the advantages of the local RPCA. At the moment of sudden lighting changes (second and third rows), two of the algorithms from the literature strongly over-detect, whereas both the global and the local RPCAs are robust to this effect. However, the global RPCA achieves this at a cost: to represent the locally varying lighting, the low-rank model must also include the person who remains static for a short while in the background. The local rank-one models are robust to the temporarily static person; we detect the person well whenever he is present.

Our quantitative evaluation is carried out in terms of recall, precision and f1-score. The f1-score is defined as $\mathrm{f1} = 2\frac{\mathrm{recall}\cdot\mathrm{precision}}{\mathrm{recall}+\mathrm{precision}}$. A high f1-score implies both a high recall and precision, and is thus a better way to evaluate algorithms than using recall or precision alone. We show the recall, precision and f1-score taken over the whole video in the table of Figure 3. The local, rank-one RPCA has an f1-score of 77.99%, compared with 68.02% of the best other method [18]. As a complement to this, we also show the f1-score per frame. The goal of this is to illustrate the lack of robustness which other methods exhibit. We do not show the f1-score on frames where few or no pixels are labelled as foreground in the ground truth, as the scores are quite unstable or meaningless for all the algorithms in this case. It is clear that the other approaches suffer from a lack of robustness either to strong lighting changes, or to the foreground person which is incorporated into the background. The local RPCA maintains a good f1-score during in these challenging situations.

In Figure 4, we see results from a standard video from the literature [6]. In this example, our local RPCA performs similarly to the global one, since the illumination changes are global. On this video (160×128, 50 seconds), the segmentation took 7m, and the subsequent local RPCA took 6m24s, on a machine with an Intel Core i5 processor. Interestingly, the graph clustering only took 0.5s, meaning that most of the work goes into calculating the edge weights. This process could obviously be carried out completely in parallel, which would greatly decrease execution times.

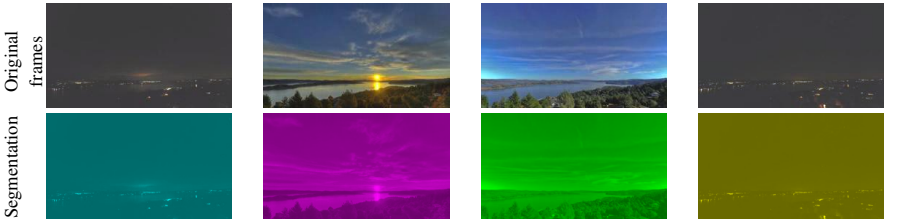


Figure 5: **Segmentation in a time-lapse example. The segmentation is indicated by varying colours.** Here, the background changes dramatically depending on the time of day. Our algorithm picks up these changes, and this is reflected in the segmentation. The standard RPCA considers that there is only one background, which does not make sense here.

In Figure 5, we show another interesting segmentation result on a timelapse video. In this example, our segmentation algorithm is able to identify the different temporal segments which contain coherent lighting. This lighting varies throughout the video, as the timelapse goes from sunrise to sunset.

Other applications Another interesting application of this work is the detection of *scene cuts* in videos. Indeed, a scene cut may be viewed as a change from one low-rank representation to another. With our framework, it is possible to identify both the temporal and spatial locations of scene transitions such as scene wipes. An example of a scene wipe transition is given in the project webpage.

5 Conclusion

We have introduced the problem of segmenting videos into regions which are well-represented by a low-rank background model plus a sparse foreground. We address this problem by creating and clustering a graph from an initial grid of spatio-temporal regions. We carefully design a cost function to determine whether two adjacent regions are coherent, in terms of their low-rank approximations. With this clustering/segmentation, we can carry out several *local* RPCAs instead of one global one. Using quantitative and qualitative comparisons with the standard RPCA and several state-of-the-art algorithms from the background subtraction literature, we show that the new piece-wise low-rank model produces significantly better background/foreground estimation in challenging situations.

For the present application, background estimation, it is not necessary for the segmentation to have a great precision, since we can simply dilate the final regions, as explained in Section 3.5. However, for other uses it could be necessary to have a higher precision. We would like to explore different options for achieving this in the future. Another clear disadvantage inherent in any RPCA-based approach is that it cannot deal with *dynamic* background, in other words backgrounds with rustling leaves or moving water. An interesting future direction would be to take this into account into the RPCA minimisation process, by modifying the quadratic error term, for example.

Acknowledgments: Work partially supported by NGA, DHS (APL), ONR, ARO, NSF, and AFOSR (NSSEFF).

References

[1] Luigi Ambrosio and Vincenzo M. Tortorelli. Approximation of Functional Depending on Jumps by Elliptic Functional via t-Convergence. *Communications on Pure and Applied Mathematics*, 43(8):999–1036, 1990.

[2] Thomas Brox and Jitendra Malik. Object segmentation by long term analysis of point trajectories. In *European Conference on Computer Vision*, pages 282–295, 2010.

[3] Emmanuel J. Candès, Xiaodong Li, Yi Ma, and John Wright. Robust Principal Component Analysis? *Journal of the ACM*, 58(3):1–37, 2011.

[4] Vicent Caselles, Ron Kimmel, and Guillermo Sapiro. Geodesic Active Contours. *International Journal of Computer Vision*, 22(1):61–79, 1997.

[5] Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes: Active Contour Models. *International Journal of Computer Vision*, 1(4):321–331, 1988.

[6] Liyuan Li, Huang Weimin, Irene Y.H. Gu, and Qi Tian. Foreground object detection from videos containing complex background. In *International Conference on Multimedia*, pages 2–10, 2003.

[7] Lucia Maddalena and Alfredo Petrosino. A Fuzzy Spatial Coherence-based Approach to Background/Foreground Separation for Moving Object Detection. *Neural Computing & Applications*, 19(2):179–186, 2010.

[8] David Mumford and Jayant Shah. Optimal Approximations by Piecewise Smooth Functions and Associated Variational Problems. *Communications on Pure and Applied Mathematics*, 42(5): 577–685, 1989.

[9] Richard Nock and Frank Nielsen. Statistical Region Merging. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(11):1452–1458, 2004.

[10] Ron Ohlander, Keith Price, and D. Raj Reddy. Picture Segmentation Using a Recursive Region Splitting Method. *Computer Graphics and Image Processing*, 8(3):313–333, 1978.

[11] Anestis. Papazoglou and Vittorio Ferrari. Fast objectcsegmentation in unconstrained video. In *IEEE International Conference on Computer Vision*, pages 1777–1784, 2013.

[12] Benjamin Recht, Maryam Fazel, and Pablo A. Parrilo. Guaranteed Minimum-Rank Solutions of Linear Matrix Equations via Nuclear Norm Minimization. *SIAM Review*, 52(3):471–501, 2010.

[13] Christof Ridder, Olaf Munkelt, and Harald Kirchner. Adaptive Background Estimation and Foreground Detection using Kalman-Filtering. pages 193–199, 1995.

[14] Jianbo Shi and Jitendra Malik. Normalized Cuts and Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[15] Andrews Sobral. BGSLibrary: An OpenCV C++ Background Subtraction Library. In *IX Workshop de Visao Computacional*, 2013.

[16] Chris Stauffer and W. E. L. Grimson. Adaptive Background Mixture Models for Real-Time Tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 246–252, 1999.

[17] Christopher Wren, Ali. Azarbayejani, Trevor. Darrell, and Alex Pentland. Pfinder: Real-Time Tracking of the Human Body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):780–785, 1997.

[18] Jian Yao and Jean-Marc. Odobez. Multi-Layer Background Subtraction Based on Color and Texture. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.

[19] Lihi Zelnik-Manor and Pietro Perona. Self-tuning Spectral Clustering. In *Advances in Neural Information Processing Systems*, pages 1601–1608, 2004.