REGRESSION PRINCIPAL COMPONENT ANALYSIS

by

Huyunting Huang

A Thesis

Submitted to the Faculty of Purdue University

In Partial Fulfillment of the Requirements for the Degree of

Master of Science

Department of Computer and Information Technology

West Lafayette, Indiana

December 2019

THE PURDUE UNIVERSITY GRADUATE SCHOOL

STATEMENT OF COMMITTEE APPROVAL

Prof. Dmitri Gusev, Chair

Department of Computer and Information Technology

Prof. Baijian Yang, Co-Chair

Department of Computer and Information Technology

Prof. Byung-Cheol Min

Department of Computer and Information Technology

Prof. Dominic Kao

Department of Computer and Information Technology

Approved by:

Dr. Eric T. Matson

Head of the Graduate Program

Dedicated to my family.

ACKNOWLEDGMENTS

I wish to gratefully acknowledge Prof.Yang and Prof.Zhang for their guidance of my research. I also want to acknowledge Prof.Gusev, Prof.Kao and Prof.Min for their feedback and comments of the points that I did not noticed first when I was writing the thesis.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

PCA	principal component analysis
RegPCA	regression principal component analysis
RPCA	robust principal component analysis
RegRPCA	Regression RPCA
SPCA	sparse principal component analysis
SS	sufficient statistics
SSE	sum square error
POD	proper orthogonal decomposition
SVD	singular value decomposition
EVD	eigenvalue decomposition
i.i.d	independent and identically distributed
ALM	augmented Lagrange multiplier
MLE	maximum likelihood estimator
PCP	principal component pursuit

GLOSSARY

Principal Component Analysis – Principal component analysis is a dimension reduction method that can reduce the dimension of a matrix without losing much information. (Pearson, 1901)

ABSTRACT

Principal Component Analysis (PCA) is a widely used dimensional reduction method that aims to find a low dimension sub space of highly correlated data for its major information to be used in further analysis. Machine learning methods based on PCA are popular in high dimensional data analysis, such as video and image processing. In video processing, the Robust PCA (RPCA), which is a modified method of the traditional PCA, has good properties in separating moving objects from the background, but it may have difficulties in separating those when light intensity of the background varies significantly in time. To overcome the difficulties, a modified PCA method, called Regression PCA (RegPCA), is proposed. The method is developed by combining the traditional PCA and regression approaches together, and it can be easily combined with RPCA for video processing. We focus the presentation of RegPCA with the combination of RPCA on video processing and find that it is more reliable than RPCA only. We use RegPCA to separate moving object from the background in a color video and get a better result than that given by RPCA. In the implementation, we first derive the explanatory variables by the background information. we then process a number of frames of the video and use those as a set of response variables. We remove the impact of the background by regressing the response against the explanatory variables by a regression model. The regression model provides a set of residuals, which can be further analyzed by RPCA. We compare the results of RegRPCA against those of RPCA only. It is evident that the moving objects can be completely removed from the background using our method but not in RPCA. Note that our result is based on a combination of RegPCA with RPCA. Our proposed method provides a new implementation of RPCA under the framework of regression approaches, which can be used to account for the impact of risk factors. This problem cannot be addressed by the application of RPCA only.

CHAPTER 1. INTRODUCTION

The problem's background, including the history and its pros and cons, are introduced in this section. Then we give the details of our problem, including its statement, significance, scope, assumption, limitation and delimitations.

1.1 Background

In 1901, Principal component analysis (PCA) was proposed by Karl Pearson (Pearson, 1901) and in 1930s Harold Hotelling independently developed this method (Hotelling, 1992).

In data analysis, PCA is a method that aims to find a hyperplane in data space with much lower dimension and the projection of data on this hyperplane will contain most, for example, 90% of the information of the data.

In different areas, PCA method has many different names, for example, singular value decomposition (SVD) of data matrix$\mathbf{X}$ (Golub & Van Loan, 1985), proper orthogonal decomposition (POD) in mechanical engineering, eigenvalue decomposition of covariance matrix $\mathbf{X}^\top\mathbf{X}$ in linear algebra (Jolliffe, 2003).

This method is now mainly used in areas like machine learning, video processing, image processing as a data analysis tool. In these areas, one typical situation is that the number of features might be too large while most of the information is concentrated on only a few of them. As a result, some methods including PCA are proposed to evaluate the features and pick those that have the most information to simplify the further analysis (Wold, Esbensen, & Geladi, 1987).

Effective as it is, this method also has many issues when scientists applied it to more and more different areas. For example, it can't be applied to the area of big data directly since the computer can't load the whole data at once when the data size is too big. In order to resolve this memorial barrier issue, Zhang and Yang proposed a method which makes use of sufficient statistics (SS) to avoid loading whole data sets all at once (Zhang & Yang, 2016).

Another typical issue in classical PCA is that it's sensitive to the large scale outliers, which is significant in video processing. To resolve this, many scientists proposed some modifications such as removing outliers, covariance matrix with 0-1 weight (Xu & Yuille, 1995), weighted SVD (Gabriel & Zamir, 1979), robust error function (De la Torre & Black, 2001), etc.

Although the concepts behind these methods are simple, they do not guarantee optimality. Then, in 2011, Candes, Li, Ma and Wright developed a method called Robust PCA (RPCA) on which can apply convex optimization (Candès, Li, Ma, & Wright, 2011).

The RPCA has a good property that can divide the data matrix of videos into sparse and low-rank components. When the light intensity of background in a video is stable, the information of the background can be explained by low rank term and the information of the moving object can be described by the sparse part of the data matrix. Based on this method, scientists have done many works in video separation. He, Balzano and Szlam developed a method called Grassmannian Robust Adaptive Subspace Tracking Algorithm (GRASTA) (He, Balzano, & Szlam, 2012) that supports estimation from subsampled data. Also, Moore, Gao and Nadakudity (Moore, Gao, & Nadakuditi, 2019) developed a method called Panoramic Robust PCA to do separation where the camera is moving.

In this research, we extend RPCA to color videos. In a color video, both the objects and the color of the objects matter. In this situation, although RPCA can extract the information of moving objects of the video into sparse component quickly, it cannot extract the color information from the original video with acceptable speed.

To overcome this difficulty, we propose a new modified PCA method, called Regression PCA (RegPCA). RegPCA is developed by combining the traditional PCA with regression approaches. Theoretically, it can be easily combined with any other PCA methods, such as the RPCA. According to the results of the experiments that we have conducted, we find that the combination of the RegPCA and RPCA can provide a more reliable result, which is a more separable result of foreground and background than that given by RPCA only.

This research will focus on our new method, RegPCA, a framework that can combine Regression approaches and PCA method, and its possible applications.

1.2 Problem Statement

Regression approaches and PCA are two types of data processing methods that are both widely used in many areas. The exploration of the combination of these two methods is meaningful work. In the previous works, PCA is often used to process the data before doing the regression, so combining these two methods in reverse order and exploring the application of the resulting novel hybrid method is natural to be attempted.

Meanwhile, RPCA, one of the most famous modified versions of PCA, shows good performance in video separation task. But it is very time-consuming to get optimal results because it needs an extremely high number of iterations to achieve that. Besides, in a color video, although it can quickly extract the information of the shape of the moving objects, it can not extract the color information in a few iterations. As a result, how to improve its performance in terms of color is a problem that can be explored.

1.3 Research Question

This research contributes answers for the following questions:

1. Is the new method able to be extended to other existing modifications of Regression and PCA? And how to extend it?

2. Can the new method improve upon the performance of RPCA on the color video separation task, along with correctness of the information on color and moving objects?

1.4 Significance

This study makes the following main contributions:

1. This research proposes a new method called Regressive Principal Component Analysis (RegPCA) which is a new way of combining linear regression approach and classical PCA method. This new method can easily be modified to combine other existing modifications of Regression and PCA such as Polynomial regression, RPCA, etc.

2. We develop a simple modification of RegPCA called RegRPCA which is the combination of Linear Regression and Robust PCA and show a possible application of this modification by three color video separation experiments. The results show that RegRPCA can improve the performance of RPCA in the color video separation tasks.

3. To evaluate the performance, color deviation, a new subjective image metric with choice of pixels, is proposed to evaluate how well the color information is kept after processing.

This study offers a framework that can produce several new data processing methods. Also, one of the modifications of RegPCA, RegRPCA offers an effective way of doing color video separation task, which can be further explored.

1.5 Statement of Scope

The RegPCA can have many different modifications because of the large number of modification of regression approaches and PCA. This paper just develops its simplest version which combines linear regression and classical PCA. Then the study develops a simple modification of RegPCA, RegRPCA, the combination of linear regression and Robust PCA, to show the ease of extension.

The study also shows a possible application of RegRPCA, the color video separation task. Finally, three experiments are conducted to evaluate the performance of the new method and RPCA to show the improvement of the new method in this area.

1.6 Assumption

The assumption for this study is:

- RPCA is effective and can offer optimum result in video separation tasks.

1.7 Limitation

The limitation of this study includes:

- The videos for the experiment have conditions that their camera are fixed and light intensity does not vary too much.

1.8 Delimitations

The delimitations for this study include:

- The implemented algorithms in this studies are all run on the CPU and do not use parallel computation technique.
- Due to the number of the existing regression methods and PCA methods, the research is first limited to the simplest combination of linear regression and classical PCA to show the way to combine these two types of methods.
- Then, a simple modification, RegRPCA which is the combination of linear regression and robust PCA, is proposed to show the ease of modification.

1.9 Summary

The research is briefly introduced in this chapter. The proposed RegPCA innovatively applies regression before employing PCA. This paper shows how this approach can be developed and extended to separate the moving objects from a video. Experiments are also designed to evaluate its effectiveness.

CHAPTER 2. REVIEW OF LITERATURE

This chapter first provides a review of the literature about PCA, including classical PCA method, kernel PCA method, Sparse PCA method and Robust PCA method. Then we give a short review of Linear Regression and some existing evaluation metrics for image quality.

2.1 Principal Component Analysis and Its Modifications

In this section, the theory of classical PCA and some of its typical modifications are introduced. Specifically, RPCA is reviewed in more details since it is the method used in the experiment of this research.

2.1.1 Classical Principal Component Analysis

In practice, PCA is a dimension reduction and feature extraction method. This method is aimed to use only a few features to tell the most information about the data. The classical PCA method has 3 kinds of picture: Dimension reduction, Geometry interpretation and singular value decomposition (Golub & Van Loan, 1985; Pearson, 1901; Wall, Rechtsteiner, & Rocha, 2003).

In terms of dimension reduction, PCA aims to find a subspace of original data space such that the variation of projection in this subspace is maximized. The corresponding formula is as follows.

Let $\mathbf{X^T}$ be an $\mathbf{p} \times \mathbf{n}$ data matrix. Here, p is the number of variables and n is the number of observations. By performing a translation transformation, the mean value of data can be set to zero without affecting the variance. The variance of data along axis $\mathbf{v}$ can then be written as.

$$\begin{aligned} \sigma^2 &= \frac{1}{n}\sum_{i=1}^{n}(v^T x_i - 0)^2 = \frac{1}{n}\sum_{i=1}^{n}(v^T x_i)(v^T x_i) \\ &= \frac{1}{n}\sum_{i=1}^{n}(v^T x_i)(v^T x_i)^T = \frac{1}{n}v^T(\sum_{i=1}^{n} x_i x_i^T)v = \frac{1}{n}v^T C v \end{aligned} \tag{2.1}$$

Where $\mathbf{C} = \mathbf{X^T X}$ is the covariance matrix. And the optimization problem is:

$$v = \arg \max_{v,|v|=1} v^T C v \tag{2.2}$$

After the first principal component is calculated, the rest of the principal components can be calculated as follows:

$$\begin{aligned} v_{s+1} &= \arg \max_{v,|v|=1} v^T C v \\ subject\ to\ v^T v_l &= 0, 1 \leq l \leq s \end{aligned} \tag{2.3}$$

By using Lagrange Multiplier Method, one can prove that this is equivalent to find first s eigenvectors of $\mathbf{C}$ (Jolliffe, 2003).

In terms of geometric interpretation, The procedure of PCA method can be understood as finding an axis that the sum of distance form data points to this axis is minimized. Based on this, the optimization problem can be rewritten as:

$$v = \arg \min_{v,v^T v=1} \sum_{i=1}^{n} |x_i - vv^T x_i|^2 \tag{2.4}$$

One can prove they are equivalent:

$$\begin{aligned} |x_i - vv^T x_i|^2 &= (x_i - vv^T x_i)^T (x_i - vv^T x_i) \\ &= x_i^T x + x_i^T vv^T vv^T x_i - x_i^T vv^T x_i - x_i^T vv^T x_i \\ &= x_i^T x_i - x_i^T vv^T x_i = x_i^T x_i - (v^T x_i)^T (v_i^T x_i) \\ &= x_i^T x_i - (v^T x_i)(v^T x_i)^T = x_i^T x_i - v^T x_i x_i^T v \end{aligned} \tag{2.5}$$

The first term is constant, the second term is the opposite number of the term in (2.1) so they're equivalent.

Finally, consider the SVD.

Let $\mathbf{X}$ be an $n \times p$ matrix. By SVD, it can be expressed as

$$\mathbf{X} = \mathbf{U}\mathbf{D}\mathbf{V}^\top \tag{2.6}$$

where $\mathbf{U} = (\mathbf{u}_1, \ldots, \mathbf{u}_p)$ is a $n \times p$ orthogonal matrix satisfying $\mathbf{U}^\top\mathbf{U} = \mathbf{I}_p$. $\mathbf{V} = (\mathbf{v}_1, \ldots, \mathbf{v}_p)$ is a $p \times p$ orthogonal matrix for loadings satisfying $\mathbf{V}^\top\mathbf{V} = \mathbf{I}_p$, and $\mathbf{D}_\varepsilon = diag(d_1, \ldots, d_p)$ is a $p \times p$ diagonal matrix for singular values and $d_p \geq d_2 \geq, \ldots, \geq d_p \geq 0$. The principal components (PCs) are the columns of $\mathbf{UD}$ and the corresponding loadings are the columns of $\mathbf{V}$. The i th PC is $\mathbf{PC}_i = d_i\mathbf{u}_i$ and the sample variance is d_i^2/n. For any integer $h \leq p$, let

$$\mathbf{X}_h = \sum_{i=1}^{h} d_i \mathbf{u}_i \mathbf{v}_i^\top = \mathbf{U}_h \mathbf{D}_h \mathbf{V}_h^\top, \tag{2.7}$$

where $\mathbf{U}_h = (\mathbf{u}_1, \ldots, \mathbf{u}_h)$, $\mathbf{D}_h = diag(d_1, \ldots, d_h)$, and $\mathbf{V}_h = (\mathbf{v}_1, \ldots, \mathbf{v}_h)$. The variation of $\mathbf{M}_h$ is $\sum_{i=1}^{h} d_i^2/n$. The ratio to the total variation is

$$\lambda_h = \frac{\sum_{i=1}^{h} d_i^2}{\sum_{j=1}^{p} d_j^2} \tag{2.8}$$

If there exists one small h that can make $\lambda_h \approx 1$, the dimension of the data matrix can be reduced from p to h with most variations contained in $\mathbf{M}_h$. Then, we can use $\mathbf{M}_h$ in the next stage analysis.

2.1.2 Kernel Principal Component Analysis

PCA has many extensions. For example, in order to solve the non-linear condition, one needs to map input space to a higher dimensional feature space by some non-linear function. Although constructing a hyperplane that divides the points into arbitrary linearly separated clusters is easy, it also increases the amount of time to compute. Therefore, the kernel method is introduced and kernel PCA has been developed (Schölkopf, Smola, & Müller, 1997). Kernel PCA has been applied in the areas like face recognition (Kim, Jung, & Kim, 2002), image modelling (Kim, Franz, & Scholkopf, 2005), nonlinear process monitoring (Lee, Yoo, Choi, Vanrolleghem, & Lee, 2004) and so on.

Instead of using the mapping function $\Phi(x),\ \Phi : R^P \longrightarrow R^n$ explicitly, which needs to work in a space with high dimension, the kernel PCA avoids it by using kernel function:

$$K = \begin{bmatrix} k(x_1,x_1) & k(x_1,x_2) & ... & k(x_1,x_n) \\ k(x_2,x_1) & k(x_2,x_2) & ... & k(x_2,x_n) \\ ... & ... & ... & ... \\ k(x_n,x_1) & ... & ... & k(x_n,x_n) \end{bmatrix} \tag{2.9}$$
$$where\ k(x,y) = (\Phi(x),\Phi(y)) = \Phi(x)^T\Phi(y)$$

Here the data matrix is redefined as $\mathbf{X} = (\Phi(x_1)^T, \Phi(x_2)^T, \Phi(x_3)^T ... \Phi(x_n)^T)^T$. The eigenvalue problem of $\mathbf{K} = \mathbf{X}\mathbf{X}^T$ is $\mathbf{X}\mathbf{X}^T u = \lambda u$. Multiply $\mathbf{X}^T$ on the left-hand side to get $\mathbf{X}^T K u = \mathbf{X}^T \lambda u$ which means $(\mathbf{X}^T\mathbf{X})\mathbf{X}^T u = \lambda X^T u$. So the $v = \mathbf{X}^T u$ is the eigenvector of the co-variance matrix defined in this high-dimension space.

Do normalization to v to get:

$$v_i = \frac{\mathbf{X}^T u_i}{\sqrt{\lambda_i}} \tag{2.10}$$

Because Φ is unknown, to avoid using $\mathbf{X}$, we can directly calculate the projection on normalized v by using Kernel function.

$$\begin{aligned} v_i^T\Phi(x_j) = \frac{1}{\sqrt{\lambda_i}} u_i^T \mathbf{X}\Phi(x_j) &= \frac{1}{\sqrt{\lambda_i}} u_i^T (\Phi(x_1)^T, \Phi(x_2)^T, \Phi(x_3)^T ... \Phi(x_n)^T)^T \Phi(x_j) \\ &= \frac{1}{\sqrt{\lambda_i}} u_i^T (k(x_1,x_j), k(x_2,x_j), k(x_3,x_j) ....., k(x_n,x_j)) \end{aligned} \tag{2.11}$$

This successfully avoids computing the Φ directly and gives the possibility to compute the non-linear principal component.

2.1.3 Sufficient Statistics Method of Classical PCA

As the size of data is increasing, the classical PCA approaches can not directly be applied to the application of big data because of the limited memory of a computer. One solution is deriving a sufficient statistics array. Then there is no need to compute the principal component directly (Zhang & Yang, 2016).

When sufficient statistics are available, the observed data is then not necessary for the following analysis (Fisher, 1922). For example, if we know the mean value of a data set, then there is no need to use the whole data to calculate the sum of the data.

When it comes to PCA, it is derived from the SVD form of PCA. Let the principal components be $U_k D_k = (PC_1, ...PC_k)$ and the loading vectors be $V = (v_1, v_2, \ldots v_k)$. Because **U** and the data matrix **X** has the same order of magnitude in terms of size, the same memory issue appears when directly using **U**. So it is necessary to think about using **V** and **D** to resolve this issue.

To get **V** and **D**, consider the following variables:

$$\mathbf{C}_{S,XX} = \mathbf{X}_S^T \mathbf{X}_S = \mathbf{V}\mathbf{D}^2\mathbf{D}^T \tag{2.12}$$

The **V** and **D** can be obtained by computing the eigenvalue and eigenvectors of $\mathbf{C}_{S,XX}$ and the size of $\mathbf{C}_{S,XX}$ is $p \times p$ which is way smaller than the size of original data matrix $(n \times p)$ in terms of big data (n is extremely large). And in order to get $\mathbf{C}_{S,XX}$, the sufficient statistical variables we need are:

$$\begin{aligned} \mathbf{C}_{XX} &= \sum_{i=1}^{n} \mathbf{X}_i^T \mathbf{X}_i \\ \mathbf{C}_X &= \sum_{i=1}^{n} \mathbf{X}_i \end{aligned} \tag{2.13}$$

Let $\mathbf{C}_{s,xx,ij}$ be the (i,j) th entry of the $\mathbf{C}_{S,XX}$, $\mathbf{C}_{XX,ij}$ be the (i,j)th entry of the $\mathbf{C}_{XX}$, $\mathbf{C}_{X,i}$ be the i th entry of $\mathbf{C}_X$. The $\mathbf{C}_{S,XX,ij}$ can be computed as follows:

$$\mathbf{C}_{S,XX,ij} = \frac{\mathbf{C}_{XX,ij} - \frac{\mathbf{C}_{X,i}\mathbf{C}_{X,j}}{n}}{s_i s_j} \tag{2.14}$$

Here $s_i = \mathbf{C}^2_{xx,ii} - \mathbf{C}^2_{x,i}/n$ is the sum square error of the ith variable for i=1,2,3....p.

Since $\mathbf{C}_{xx}$ and $\mathbf{C}_x$ can be computed row by row procedurally and do not require loading all of the data into memory in one time, this formulation makes it possible to apply classical PCA method in the area of big data.

As for the $\mathbf{U}$, $\mathbf{U}$ is not available, it depends on the specific question we need to analyze. And it's very similar to the kernel PCA where transformed data matrix $\mathbf{X}$ is not available, we directly compute the projection along the axis without using Φ in Kernel PCA.

Here the strategy is similar. Since the PCA method is usually followed with other data analysis procedure such as regression analysis. This is also the typical way of combining PCA methods and regression approaches. For example in linear regression, $\mathbf{U}y$ is directly computed instead of using $\mathbf{X}$ alone, and this avoids using $\mathbf{U}$ directly (Zhang & Yang, 2016).

2.1.4 Robust PCA

Finally, classical PCA also has an issue that it is very sensitive to the large scale outlier. To deal with this issue, some scientists proposed some methods based on some simple concepts such as removing outlier(Hastings, Mosteller, Tukey, Winsor, et al., 1947), using weighted SVD (Gabriel & Zamir, 1979), using robust error function (De la Torre & Black, 2001) and so on. However, these methods can not guarantee the optimality of the result. Then Candes, et al. developed a method called Robust PCA which can robustify the PCA method with optimality (Candès et al., 2011).

Note that one of the goals of the PCA method is dimension reduction and is dealing with data matrix which is still a matrix. The PCA problem can be reformulated as:

Given data matrix $\mathbf{X}$, recover a low-rank matrix $\mathbf{L}$ from $\mathbf{X}$ to minimize the error: $\mathbf{E} = \mathbf{X} - \mathbf{L}$.

$$\mathbf{L} = \arg \min_{\mathbf{L}, rank(\mathbf{L}) \leq r} \|\mathbf{X} - \mathbf{L}\|_F \tag{2.15}$$

$\| \, \|_F$ is Frobenius norm (Murphy, 2014), defined as $\|\mathbf{X}\|_F = \sqrt{\sum_j \sum_j \mathbf{X}_{i,j}^2}$ Since this is not robust, Wright, et al. formulated a Robust PCA problem as follows (Wright, Ganesh, Rao, Peng, & Ma, 2009):

$$\begin{gathered}(\mathbf{L}, \mathbf{E}) = \arg \min_{\mathbf{L}, \mathbf{E}} rank(\mathbf{L}) + \lambda_0 |\mathbf{L}|_0 \\ subject\ to\ \mathbf{L} + \mathbf{E} = \mathbf{X}\end{gathered} \tag{2.16}$$

However, since the rank and l_0 norm is not continuous, this is not a convex problem. They then reformulated the problem to make it convex:

$$\begin{gathered}(\mathbf{L}, \mathbf{E}) = \arg \min_{\mathbf{L}, \mathbf{E}} \|\mathbf{L}\|_* + \lambda_0 |\mathbf{E}|_1 \\ subject\ to\ \mathbf{L} + \mathbf{E} = \mathbf{X}\end{gathered} \tag{2.17}$$

Here $\| \, \|_1$ is the summation of the absolute value of elements of matrix and $\| \, \|_*$ is defined as the summation of the singular value of matrix. Since these two norms are convex, it is possible to apply convex optimization method on it. When **E** is sufficiently sparse but not low rank and **L** is sufficiently low rank but not sparse, the solution is exact.

There are several ways to solve this optimization problem. For example, principal component pursuit (PCP) (Candès et al., 2011), iterative thresholding (Cai, Candès, & Shen, 2010), accelerated proximal gradient (Bao, Wu, Ling, & Ji, 2012), augmented Lagrange multipliers (Hale, Yin, & Zhang, 2007; Lin, Chen, & Ma, 2010), and so on. To make it easy to understand, here we demonstrate the way of augmented Lagrange multiplier (ALM) which combines the penalty method and Lagrange multiplier method to solve Robust PCA. The Robust PCA can be reformulated as:

$$(\mathbf{L},\mathbf{E}) = \arg\min_{\mathbf{L},\mathbf{E}} \|\mathbf{L}\|_* + \lambda_0|\mathbf{E}|_1 + <\mathbf{Y},\mathbf{X}-\mathbf{L}-\mathbf{E}> + \frac{\mu}{2}\|\mathbf{X}-\mathbf{L}-\mathbf{E}\|_F^2 \quad subject \quad to \quad \mathbf{L}+\mathbf{E}=\mathbf{X} \tag{2.18}$$

Here $<\mathbf{A},\mathbf{B}>=\sum_i\sum_j \mathbf{A}_{ij}\mathbf{B}_{ij}$ and $\mathbf{Y}$ is the Lagrange multiplier. Solving L with fixed E will get $\mathbf{L} = \mathbf{U}\mathbf{T}_{\frac{1}{\mu}}(\mathbf{D})\mathbf{V}^T$ and solving E with fixed L will get $\mathbf{E} = \mathbf{T}_{\frac{\lambda}{\mu}}(\mathbf{X}-\mathbf{L}+\frac{\mathbf{Y}}{\mu})$. T_ε is soft threshold operator which is defined as:

$$T_\varepsilon(x) = \begin{cases} x-\varepsilon & if\ x>\varepsilon \\ x+\varepsilon & if\ x<\varepsilon \\ 0 & otherwise \end{cases} \tag{2.19}$$

The full algorithm of ALM for RPCA is in *algorithm* 2.1:

Algorithm 2.1 RPCA via ALM

function ALM FOR RPCA($\mathbf{X}$)
    Initialize $\mathbf{L} = 0, \mathbf{E} = 0$
    Initialize $\mathbf{Y}, \mu > 0, \rho > 1$
    while *not converge* **do**
        while *not converge* **do**
            $\mathbf{U}, \mathbf{S}, \mathbf{V} = SVD(\mathbf{X} - \mathbf{E} + \mathbf{Y}/\mu)$
            $\mathbf{L} = \mathbf{U}\mathbf{T}_{\frac{1}{\mu}}(\mathbf{D})\mathbf{V}^T$
            $\mathbf{E} = \mathbf{T}_{\frac{\lambda}{\mu}}(\mathbf{X} - \mathbf{L} + \frac{\mathbf{Y}}{\mu})$
        end while
        update $\mathbf{Y} = \mathbf{Y} + \mu(\mathbf{D} - \mathbf{L} - \mathbf{E})$
        update $\mu = \rho\mu$
    end while
    output $\mathbf{L}, \mathbf{E}$
end function

RPCA method has strong constraints in the data matrix. As mentioned above, it requires $\mathbf{L}$ is sufficiently low rank but not sparse and $\mathbf{E}$ is sufficiently sparse but not low rank. In practice, such as video processing, this means the camera is steady, and the variation among images is small, or this method will diverge. So developing a method that has a relatively weaker constraint in data set is a potential direction to explore.

Because of the ability to separate the data into sparse part and low-rank part, RPCA is very effective in performing video separation task. When the video is camera-fixed and low light variance, it can quickly extract the information of moving objects from the video. However when the video has complicated color distribution. RPCA might spend extremely high iterations to get a complete separation. Improving its time performance is thus a necessary thing to do.

2.2 Regression analysis and Linear Regression

In statistics, Regression analysis is a statistical analysis method to model the quantitative relationship among variables (Draper & Smith, 1998). It aims to describe the behavior of dependent variables changes when independent variables vary.

Regression analysis including many different models. For example, Linear Regression which tries to use a linear function to describe the relationship with least-squares estimations (Seber & Lee, 2012), Polynomial Regression which extends the linear function to polynomial ones (Kleinbaum, Kupper, Muller, & Nizam, 1988), and Lasso and Ridge Regression which use the linear function with different norms (Hoerl & Kennard, 1970; Tibshirani, 1996).

Among these models, Linear Regression is the simplest model in regression analysis. It models the relationship between a scalar response and the explanatory variables.

Assume the dataset contains n observations each of which has $p-1$ features. Linear regression is written as:

$$y = \mathbf{X}\beta + \varepsilon, \tag{2.20}$$

Where $\mathbf{X} = (x_1^\top, x_2^\top, \ldots, x_n^\top)^\top$ is a $n \times p$ matrix of explanatory variables, $y = (y_1, y_2, \ldots, y_n)^\top$ is a $n \times 1$ vector of the response, $\beta = (\beta_0, \beta_1, \ldots, \beta_{p-1})^\top$ is a $p \times 1$ linear coefficient vector. $\varepsilon = (\varepsilon_1, \ldots, \varepsilon_n)^\top$ is the error term which is a $n \times 1$ vector following the normal distribution $\mathbf{N}(0, \sigma^2\mathbf{I})$.

By maximizing loglikelihood function:

$$\mathscr{L}_{lr}(\beta, \sigma^2) = -\frac{n}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\|\mathbf{y} - \mathbf{X}\beta\|_2^2 \tag{2.21}$$

where $\|\cdot\|_2$ is an ℓ_2 norm. And the maximum likelihood estimator (MLE), the solution of this model is:

$$\begin{aligned} \hat{\beta} &= (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y} \\ \hat{\sigma}^2 &= \frac{1}{n}(\mathbf{y} - \mathbf{X}\hat{\beta})^\top(\mathbf{y} - \mathbf{X}\hat{\beta}) \end{aligned} \tag{2.22}$$

In video separation, linear regression and other more complex regression tools can be used to describe the correlation among frames. In a set of frames of a video that is camera fixed, most pixels in a frame are very similar to the those in other frames. Due to this fact, it is reasonable to apply regression approaches to describe this dependence.

2.3 Discussion about Typical Combination of Regression and Classical PCA

Classical PCA, as a dimension reduction method, is usually applied in the areas of data science as a pre-processing tool. Scientists often apply PCA before doing Regression analysis. When the data size is large and each sample has a large number of features, directly applying analysis will be time-consuming and might lose some explainability of the model. Those features, which do not have much variance, are more likely to be the consistent features shared by whole data instead of features that classify the data. For example, in a list of people, they might be classified by their gender or the age, but since they are all humans, there is no need to add a feature called human into the dataset.

And in such condition, the classical PCA is often used first to extract the features that have the biggest variance and ignore those flat features to accelerate the processing speed. And it might also make the model more explainable.

2.4 Quality Index

To evaluate how well the color information can be extracted from the original video, a metric is necessary. This metric should be used to compare the processed results with the original one.

There are two types of ways to evaluate the quality of an image. The first type is objective metrics, such as Signal-to-Nose Ratio (Johnson, 2006) and Mean Squared Error which are objective evaluation metrics. This kind of metrics is to evaluate the image itself and has no comparison with other images.

Another kind of metric is the subjective metric, which is to compare the processed image with the original one, one of these metrics is quality index (Wang, Bovik, & Lu, 2002). Its definition is as follows.

Let the source image signals be $\mathbf{x} = \{x_i | i = 1, 2, ..., N\}$ and the processed image signals be $\mathbf{y} = \{y_i | i = 1, 2, ..., N\}$. Then quality index can be defined as

$$Q = \frac{\sigma_{xy}}{\sigma_x \sigma_y} \cdot \frac{2\bar{x}\bar{y}}{(\bar{x})^2 + (\bar{y})^2} \cdot \frac{2\sigma_x \sigma_y}{\sigma_x^2 + \sigma_y^2}. \tag{2.23}$$

Where

$$\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i, \qquad \bar{y} = \frac{1}{N}\sum_{i=1}^{N} y_i,$$
$$\sigma_x^2 = \frac{1}{N-1}\sum_{i=1}^{N}(x_i-\bar{x})^2, \sigma_y^2 = \frac{1}{N-1}\sum_{i=1}^{N}(y_i-\bar{y})^2, \tag{2.24}$$
$$\sigma_{xy} = \frac{1}{N-1}\sum_{1=1}^{N}(x_i-\bar{x})(y_i-\bar{y}).$$

This metric includes three factors, mean distortion, loss of correlation, and variance distortion.

However, this metric is developed based on the fact that both source and processed image are complete. Quality Index has no bias among pixels. In a video separation task, the result is only part of the original picture. Some modifications is necessary to reasonably evaluate the performance of RegPCA in the video separation experiments.

2.5 Summary

This chapter provides a literature review about the PCA method, Regression approaches and image evaluation. More specifically, it introduces the constraints of RPCA and explains why the existing image evaluation metrics can not be directly applied in our experiments.

CHAPTER 3. METHODLOGY

The study wants to develop a new method called RegPCA that based on PCA method and Regression Approaches. And we also want to find the possible application of the RegPCA.

This chapter provides:

- the development of Reg(R)PCA,
- The design of the experiments and metric for the evaluation of the method.

3.1 Development of Reg(R)PCA

The details of the method are introduced in this section. It includes the development of RegPCA in Section 3.1.1, the combination of RegPCA and RPCA in Section 3.1.2, and the implementation of our method to video separation in Section 3.1.3.

3.1.1 RegPCA

In video and image processing, the motivation of RegPCA is to account for a given image by a few underlying images. The given image is treated as a response. The underlying images are treated as explanatory variables. In order to fit those by a regression approach, we need to convert all the images into vectors by matrix unfolding methods. After the response is addressed by the explanatory variable, a residual vector is derived. The residual vector can be used to reflect the information contained by the given image after the impacts of the underlying images are removed. To carry out a dimension reduction approach, we need to convert the residual vector back to a matrix. Then, we obtained the RegPCA method.

Suppose that a matrix for a response variable has been unfolded to a vector, and a number of matrices for explanatory variables have also been unfolded to vectors. In practice, each matrix could be derived from an image. Then, the interest is to study the relationship between a given image for the response and several base images for the explanatory variables.

Let the unfolded vector for the given image be $\mathbf{y}$, and the unfolded vectors for the base images be $\mathbf{x}_1, \ldots, \mathbf{x}_p$. Then, the relationship is modelled by

$$\mathbf{y} = \mathbf{1}\beta_0 + \sum_{j=1}^{p} \mathbf{x}_j \beta_j + \varepsilon, \tag{3.1}$$

where $\mathbf{1}$ is a vector with all of its components equal to 1, and $\varepsilon \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I})$ is the error vector. We assume each image has n pixels, such that $\mathbf{y} = (y_1, \ldots, y_n)^\top$ and $\mathbf{X}_j = (x_{1j}, \ldots, x_{nj})^\top$ for all $j \in \{1, \ldots, p\}$. By matrices, (3.1) is written as

$$\mathbf{y} = \mathbf{X}\beta + \varepsilon \tag{3.2}$$

where $\beta = (\beta_0, \ldots, \beta_p)^\top$ and $\mathbf{X} = (\mathbf{1}, \mathbf{x}_1, \ldots, \mathbf{x}_p)$ is a $n \times p$ matrix and $\varepsilon \sim N(0, \sigma^2 \mathbf{I})$ is an $n-$ dimensional error vector.

In practice, as we are dealing with image, we just use MLE of β and calculate ε as $\hat{\varepsilon} = \mathbf{y} - \mathbf{X}\hat{\beta}$

The regression model above offers two parts for the image, linear function $\mathbf{X}\beta$ for the base image and error vector ε for the difference from the base image which would be further analyzed by traditional PCA method.

By folding the $\hat{\varepsilon}$ back to image matrix $\mathbf{M}_{\hat{\varepsilon}}$ with size $h \times w$, it is possible to apply PCA into it. In particular, we can set $\mathbf{M} = \mathbf{M}_{\hat{\varepsilon}}$ in (2.6), where $\mathbf{M}_{\hat{\varepsilon}}$ is a matrix expression of $\hat{\varepsilon}$ with the same format as the original image. By SVD, we obtain

$$\mathbf{M}_{\hat{\varepsilon}} = \mathbf{U}_\varepsilon \mathbf{D}_\varepsilon \mathbf{V}_\varepsilon^\top \tag{3.3}$$

where $\mathbf{U}_\varepsilon = (\mathbf{u}_{\varepsilon 1}, \ldots, \mathbf{u}_{\varepsilon w})$, $\mathbf{V}_\varepsilon = (\mathbf{v}_{\varepsilon 1}, \ldots, \mathbf{v}_{\varepsilon w})$, and $\mathbf{D}_\varepsilon = diag(d_{\varepsilon 1}, \ldots, d_{\varepsilon w})$ are defined similarly. For any integer $k \leq w$, we can similarly define

$$\mathbf{M}_{\hat{\varepsilon} k} = \sum_{i=1}^{k} d_{\varepsilon i} \mathbf{u}_{\varepsilon i} \mathbf{v}_{\varepsilon i}^\top = \mathbf{U}_{\varepsilon k} \mathbf{D}_{\varepsilon k} \mathbf{V}_{\varepsilon k}^\top \tag{3.4}$$

with $\mathbf{U}_{\varepsilon k} = (\mathbf{u}_{\varepsilon 1}, \ldots, \mathbf{u}_{\varepsilon k})$, $\mathbf{D}_{\varepsilon k} = diag(d_{\varepsilon 1}, \ldots, d_{\varepsilon k})$, and $\mathbf{V}_{\varepsilon k} = (\mathbf{v}_{\varepsilon 1}, \ldots, \mathbf{v}_{\varepsilon k})$. This can remove the impacts of base image and maintain the information of the different part of the original image.

The RegPCA method can be straightforwardly extended for videos. Note that only one frame is used in the above formulation. If multiple frames are used, then it is a method for videos. If only residual components are considered, then we can also combine RegPCA with other dimension reduction method. For example, we can use RPCA to analyze the residual components given by RegPCA, which provides a combination of RegPCA and RPCA method. This method will be introduced in Section 3.1.2.

3.1.2 Combination with RPCA

To deal with the video, which has multiple frames, the RegPCA method developed in 3.1.1 will be applied multiple times. For example, to process a video with k frames, the RegPCA will be applied for k times to process each frame.

To combine with RPCA(RegRPCA), the first step is combining all the residual components generated by the regression step of RegPCA to get a $n \times k$ residual matrix $\mathbf{X}_\varepsilon$ where n is the number of pixels of a frame and k is the number of frames of the video. We treat this $\mathbf{X}_\varepsilon$ as data matrix in RPCA. And then solve the following optimization problem

$$\begin{aligned}(\mathbf{L}_\varepsilon, \mathbf{E}_\varepsilon) = \arg\min_{\mathbf{L}_\varepsilon, \mathbf{E}_\varepsilon} \|\mathbf{L}_\varepsilon\|_* + \lambda_0 |\mathbf{E}_\varepsilon|_1 \\ subject\ to\ \mathbf{L}_\varepsilon + \mathbf{E}_\varepsilon = \mathbf{X}_\varepsilon\end{aligned} \tag{3.5}$$

where $\| \ \|_*$ is the nuclear norm which is the sum of the singular value of matrix and $\| \ \|_1$ is l_1 norm which is the sum of the absolute value of elements of matrix. $\mathbf{L}_\varepsilon$ is low-rank part of the residual matrix. $\mathbf{E}_\varepsilon$ is the sparse part of the residual matrix.

The above optimization problem could be easily solved by PCP method whose code is available online (Xiao et al., 2019). The low-rank part will be used for further analysis and the sparse part will be discarded.

3.1.3 Implementation in Video Separation

To apply RegPCA and its combination with RPCA, the first thing is converting video into a matrix. The way we convert video to data matrix is shown in *algorithm* 3.1, we first load video into memory, then we convert each frame into a matrix with size $h \times w \times 3$ where (h, w) represents the width and height of a frame and 3 represents the color channel. We then separate this matrix into three parts with size $h \times w$ representing R, G, B channel of the frame. Next, we unfold the matrices into three column vectors with size $(h \times w) \times 1$. Finally, we get three $N \times K$ video color channel matrices. The K is the number of frames of video. $N = h \times w$ is the number of pixels of one frame.

The color channel matrices $\mathbf{R}, \mathbf{G}, \mathbf{B}$ are treated as three data matrices. And we apply RegPCA or RegRPCA to them respectively.

The algorithm for regression part of RegPCA are in *algorithm* 3.2, we use the first frame of video as the base image and regress the rest frames of video against the base image.

After applying RegPCA/RegRPCA to each video channel matrix, we combine the regression part of each channel matrix to form the regression part of the whole video which describes the background of the video. And we combine the post-residual part of each channel matrix to the post-residual part of video to describe the moving objects of the video. Thus we perform a video separation task through RegPCA/RegRPCA.

3.2 Experiment Design

This research simply chooses the first frame of the video to be the explanatory variable.

Based on the method, we perform video separation task on three color videos, a screen record from a mobile game (mihoyo, 2019), an airport video and a snow train video downloaded from the Youtube (satsumannoyaji, 2019; Spears, 2019). Some frames of videos can be seen via *figure* 3.1, *figure* 3.2 and *figure* 3.3 The details of the experiments are shown in the rest of this session and the results are shown in the next chapter. All the algorithms were implemented on Python.

3.2.1 Data Processing

In the experiments, the video is converted to three color channel matrices $\mathbf{M}_r, \mathbf{M}_g, \mathbf{M}_b$ created by *algorithm* 3.1.

Then in order to get the correct color information we do gamma correction (Hunt, 2004) to the elements of the color channel matrix:

$$\mathbf{X}'_{ij} = (\mathbf{X}_{ij}/255)^{2.2}, \tag{3.6}$$

where $\mathbf{X}$ is the original color channel matrix, $\mathbf{X}'$ is the the corrected color channel matrix. As a result, the corrected color channel matrices $\mathbf{M'}_r, \mathbf{M'}_g, \mathbf{M'}_b$ are generated.

After the Reg(R)PCA is applied, the gamma recover is done to each part of the video to recover the frame:

$$\mathbf{X}_{ij} = \mathbf{X'}_{ij}^{\frac{1}{2.2}} * 255, \tag{3.7}$$

3.2.2 Evaluation

As mentioned in the literature review, the existing subjective metrics for color image quality are developed based on the fact that the image is complete. They have no bias to pixels. And since this research is to evaluate parts of the image. It is necessary to develop a new metric that has the choice about pixels.

This research proposes a new metric called color deviation to evaluate how well the color information is kept after separation.

The color deviation C of a frame is defined as:

$$C = \frac{1}{3}\frac{1}{N_{eff}}\sum_{i=0}^{3}\sum_{j=0}^{n}\frac{|X_{ij} - X'_{ij}|^2 H(X'_{ij}) H(X_{ij})}{X_{ij}^2}, \tag{3.8}$$

Where X_{ij} is the ith color channel value of jth pixel of the frame, X'_{ij} is corresponding value of the residual part of this frame. N_{eff} is the total effective count of the pixels, i.e the pixel that has non-zero value in both images.

$$N_{eff} = \sum_{j=1}^{3} \frac{1}{\sum_{i=1}^{n} H(X'_{ij})H(X_{ij})} \tag{3.9}$$

$H(x)$ is the Heaviside step function:

$$H(x) = \begin{cases} 1 \text{ if } x > 0 \\ 0 \text{ if } x = 0 \end{cases}, \tag{3.10}$$

And the color deviation of a video with k frames is defined as:

$$C_{total} = \frac{1}{k} \sum_{l=1}^{k} C_l, \tag{3.11}$$

The lower C_{total} means smaller color deviation from the original videos.

3.3 Discussion about the Difference from the Classical Combination

Different from the classical way of combination which is $PCA + Regression$ mode, RegPCA is the $Regression + PCA$ mode. The different order results in different areas where the method is effective. The classical way is typically applied in the areas where regression is focused on such as big data, machine learning, and so on. And from the experiment, RegRPCA, a simple modification of RegPCA, might be effective in the area of video processing, at least in some video separation tasks.

More specifically, when PCA is first applied, it aims to extract the most meaningful information from the data. And when Regression is first applied, it aims to remove the known knowledge and extract the unknown knowledge from the data. The difference is that the most meaningful information might be already known. So the different order will not result in the same method.

Algorithm 3.1 Create Video Color Channel Matrix

Input:A video V with k frames, each frame is of size $h \times w$
Output: 3 video color channel matrices $\mathbf{R}, \mathbf{G}, \mathbf{B}$ with size $N \times K, N = h \times w$

read V
$\mathbf{R} = zeros(N,K)$,$\mathbf{G} = zeros(N,K)$,
$\mathbf{B} = zeros(N,K)$
for i in K **do**
    convert ith frame to image
    convert image to $N \times 3$ matrix $\mathbf{m}$
    $\mathbf{m}_r = m[:,:,1]$,$\mathbf{m}_g = m[:,:,2]$,$\mathbf{m}_b = m[:,:,3]$
    convert $\mathbf{m}_r, \mathbf{m}_g, \mathbf{m}_b$ into arrays $\mathbf{L_r}, \mathbf{L_g}, \mathbf{L_b}$ with size $N \times 1$
    $\mathbf{R}[:,i]$=$\mathbf{L_r}$,$\mathbf{G}[:,i]$=$\mathbf{L_g}$,$\mathbf{B}[:,i]$=$\mathbf{L_b}$
end for
Return $\mathbf{R}, \mathbf{G}, \mathbf{B}$

In RegRPCA, regression analysis is used as a pre-processing method to remove most of the background information from the original video. Then RPCA is used to do a complete separation in the residual part offered by a regression method. Since RPCA might spend an extremely long time to do a complete separation in a video whose color distribution is complex. (Linear) Regression can accelerate this process and give a much better initial condition for RPCA to start.

3.4 Summary

This chapter describes the details of Reg(R)PCA including how to combine Linear Regression and classical PCA, a simple extension to RPCA, and how to apply it to the color video separation tasks. It also explains the details of experiments for the evaluation of the new method, including experiment design and evaluation metrics. In the next chapter, the results of the experiment are presented.

Algorithm 3.2 Regression Part

Input:A Video Channel Matrix M
Output: Regression Part $\mathbf{R}_0$ and Residual part $\mathbf{R}_1$ of Video Matrix

$\mathbf{X} = \mathbf{M}[:, \mathbf{1}]$
$\mathbf{R}_0 = zeros(N, K)$
$\mathbf{R}_1 = zeros(N, K)$
$\mathbf{Y} = \mathbf{M}[:, 2 : K]$
for i from 2 to K **do**
    $\mathbf{y} = \mathbf{Y}[:, \mathbf{i} - \mathbf{1}]$
    $\beta = Linear\_Regression(\mathbf{y}, \mathbf{X})$
    $\mathbf{R}_0[:, i] = \beta \mathbf{X}$
    $\mathbf{R}_1[:, i] = \mathbf{y} - \mathbf{R_0}[:, \mathbf{i}]$
end for
Return $\mathbf{R}_0, \mathbf{R}_1$

Figure 3.1. Examples of Frames in Game Videos

Figure 3.2. Examples of Frames in Airport Videos

Figure 3.3. Examples of Frames in Airport Videos

CHAPTER 4. EXPERIMENTS

In this chapter, the results of the experiment and the conclusion are presented. Three experiments are conducted to evaluate the RegRPCA, first one is the color separation task on a screen record for a mobile game whose light intensity and camera position are totally fixed (mihoyo, 2019). Second and third experiment are color video separation task for realistic videos (satsumannoyaji, 2019; Spears, 2019).

The game video represents an ideal condition that the camera is fixed, the light intensity does not vary and only small objects are moving. The airport video represents a realistic condition. It has some small camera jitters, the light intensity slightly varies. The snow train video represents another condition. It has fewer camera jitters and light variance than airport video. But in a specific period, the moving objects cover a large area of the screen in the video which means the sparse part is now not very sparse.

4.1 Experiment One: Game Video

The first experiment is video separation task on the game video. The numeric results of color deviation are in $table$ 4.1 and the visual performance is in $figure$ 4.1. The experiment compares the performance of RegRPCA(10 iterations) and RPCA(10 iterations, 50 iterations and 100 iterations).

The experiment results show that in such condition, the result of RegRPCA has systematically lower color deviation than RPCA only. It can also be seen from the results of the visual performance. The color of characters in the results of RegRPCA is similar to the original images although it has some ghost effect due to the matrix reduction. In RPCA, although the shape of characters can be figured out clearly, the color is not quite correct.

For the accuracy of separation, both RegRPCA and RPCA can extract the information of moving characters clearly.

4.2 Experiment Two: Airport Video

The second experiment is video separation task on the airport video. The numeric results of color deviation of the video are in $table$ 4.2 and the visual performance is in $figure$ 4.2. We compare the performance of RegRPCA (10 iterations) and RPCA (10 iterations, 50 iterations and 10000 iterations).

The experiment results show that in this video, RegRPCA has less color deviation than RPCA only. Even when the iteration of RPCA goes much higher, the color deviation of RPCA is still higher than RegRPCA. It can also be seen from the results of visual performance. The color of people in the results of RegRPCA is similar to the original images although the background is not clean due to the camera jitters and simple matrix reduction. In RPCA, the shape of moving objects can be figured out quickly. But when the iteration is low (10 to 100), the color of result is very different from the original one, and when iteration goes high (10000), the result color gets a little bit closer to the original one but not significantly. And in the experiment, running 10000 iterations of RPCA for 100 frames consumed over 3 days, so it is very time-consuming to get an ideal separation result if it can be reached.

As for the accuracy of separation, from the results, we find that RPCA can extract the shape of people well. And the RegRPCA can extract the information of moving objects although the background information is also partly extracted by this method.

4.3 Experiment Three: Snow Train Video

The third experiment is video separation task on the snow train video (satsumannoyaji, 2019). The color deviation scores of the methods are in $table$ 4.3 and the visual performance is in $figure$ 4.3

The numeric results show that in this video, RegRPCA also has less color deviation than RPCA. And the results of RPCA have much higher color deviation than the results of RPCA in other experiments. This is because that in this video, the moving objects contains snow which spreads out and covers a large area of the screen. The sparse part is not sparse enough in such situation, so RPCA is not as effective as it is in the previous experiments. According to the results of visual quality, the RPCA (10 to 50 iterations) can not provide quality results. The train and snow can hardly be figured out. The images also contain lots of noise. The RegRPCA extracts the color information well. Most of color information of the red train and white snow are extracted from the original video clearly.

As for the accuracy of separation, we find that RPCA can roughly depict the shape of train but the snow and noise can not be classified well. And in the results of RegRPCA, both train and snow are clearly presented.

4.4 Discussion

In this experiment, the results of RegRPCA have two things that need to be noticed. First of all, the result of RegRPCA in airport video is not clean because the method to produce the result is simply the matrix reduction. By applying a simple high-pass filter, the remaining background can be easily removed which is shown in $figure$ 4.4, although some other issues will also appear. Secondly, some ghost effect appears in the images. And it is also because the matrix reduction is too simple in the color video separation task.

While the method processed the data in RegRPCA is simple, it still has positive results comparing with RPCA. So we think this combination is effective when dealing with color video separation task.

4.5 Conclusion

This Chapter presents the results of three experiments conducted in three color videos with different condition. Under the metric of color deviation developed in this paper, the RegRPCA showed a better result in terms of color correctness which can also be seen by comparing the images offered by these two methods.

Table 4.1. *Color Deviation of the Game Video. The lower color deviation index indicates a better ability to keep the color information.*

method/channel	Red	Green	Blue	Average
RegRPCA (10 iter)	0.7660	0.6877	0.7265	0.7267
RPCA(10 iter)	0.9768	0.9705	0.9687	0.97203
RPCA(50 iter)	0.9747	0.9626	0.9604	0.9659
RPCA(100 iter)	0.9734	0.9578	0.9554	0.9622

Table 4.2. *Color Deviation of the Airport Video. The lower color deviation index indicates a better ability to keep the color information.*

method/channel	Red	Green	Blue	Average
RegRPCA (10 iter)	0.6951	0.7402	0.7172	0.7175
RPCA(10 iter)	0.9850	0.9801	0.9838	0.9830
RPCA(50 iter)	0.9731	0.9653	0.9707	0.9697
RPCA(10000 iter)	0.9687	0.9633	0.9690	0.9670

Table 4.3. *Color Deviation of the snowtrain*

method/channel	Red	Green	Blue	Average
RegRPCA	0.5492	0.5111	0.5322	0.5302
RPCA(10 iter)	1.0297	1.0298	1.0288	1.0295
RPCA(50 iter)	1.0245	1.0249	1.0244	1.0246

Figure 4.1. Results of Game Video. Row 1 is the original frames. Row 2 is the results of RegPCA, Row 3 is the result of RPCA with 10 iterations. Row 4 is 50 iterations of RPCA, Row 5 is 100 iterations of RPCA.

Figure 4.2. Results of Airport Video. Row 1 is the original frames. Row 2 is the results of RegPCA, Row 3 is the result of RPCA with 10 iterations. Row 4 is 50 iterations and Row 5 is 10000 iterations of RPCA.

Figure 4.3. Results of Snowtrain Video. Row 1 is the original frames. Row 2 is the results of RegPCA, Row 3 is the result of RPCA with 10 iterations. Row 4 is 50 iterations of RPCA.

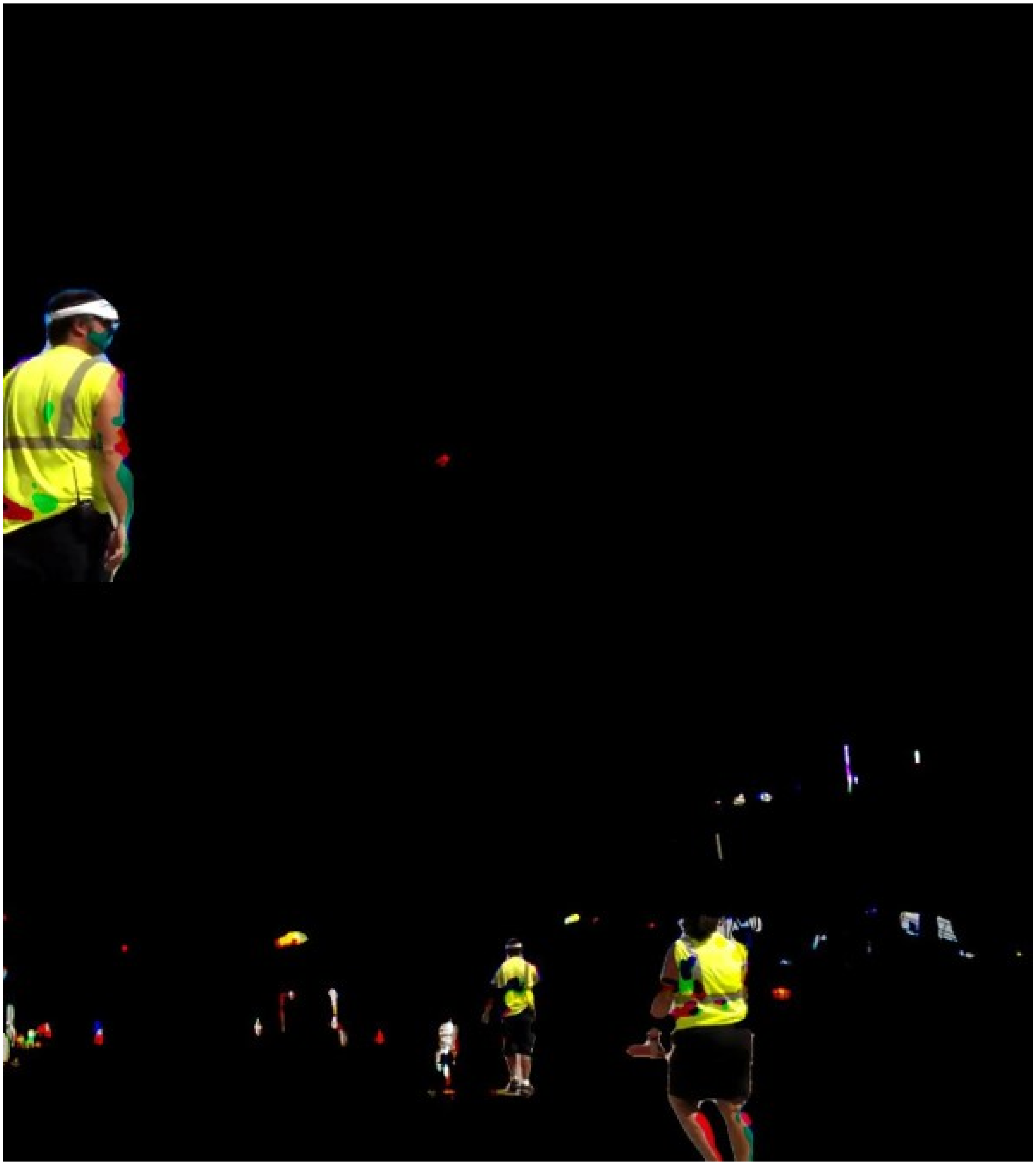

Figure 4.4. Results of RegRPCA in Airport Video with a simple high-pass filter.

CHAPTER 5. CONCLUSION AND FUTURE WORK

5.1 Conclusion

From the results of the experiments above, one can safely conclude that at least in some specific videos, although RPCA can quickly extract the information of moving objects, it can not keep the color information right until a huge number of iteration to get its optimal results. At the same time, RegRPCA, the combination of linear regression and RPCA, can extract the information of moving objects with much better color information.

5.2 Future Work

In the future, there are two directions to explore, which are methodology and technology. For the methodology, the RegPCA can be regarded as a new way of combination of Regression and PCA. These two types of methods both have many different modifications. So there are many possible combinations among them, for example, Ridge Regression and Sparse PCA (Zou, Hastie, & Tibshirani, 2006), Polynomial Regression and RPCA and so on. Developing these methods and find the possible application will have a long way to go.

As for technology, there are two things to be explored. First of all, the way to develop the explanatory variable in this paper is very simple, also the way generating the residual part is simply the reduction of matrix. Some complex methods might improve the performance in color video separation task. For example, applying median blur to the residual can effectively remove the noise point of the result although it will also affect the quality of the image. How to properly apply these methods to improve the performance is needed to be studied. Second, the generality of RegRPCA also needs to be considered. In this experiment, result is positive because the camera is almost fixed and light intensity does not vary. Its performance in other situations, such as obvious camera moving, is not evaluated yet. So it is necessary to explore how to generalize the method in order to fit other kinds of videos. And in terms of technology, generalization and proper application of other complex processing methods are two highly related directions because an improvement in one often means good progress in another one.

In conclusion, both methodology and technology of RegPCA have many things to be explored. And it is very likely to be a new effective tool for other researchers when it is highly developed.

REFERENCES

Bao, C., Wu, Y., Ling, H., & Ji, H. (2012). Real time robust l1 tracker using accelerated proximal gradient approach. In *2012 ieee conference on computer vision and pattern recognition* (pp. 1830–1837).

Cai, J.-F., Candès, E. J., & Shen, Z. (2010). A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, *20*(4), 1956–1982.

Candès, E. J., Li, X., Ma, Y., & Wright, J. (2011). Robust principal component analysis? *Journal of the ACM (JACM)*, *58*(3), 11.

De la Torre, F., & Black, M. J. (2001). Robust principal component analysis for computer vision. In *Proceedings eighth ieee international conference on computer vision. iccv 2001* (Vol. 1, pp. 362–369).

Draper, N. R., & Smith, H. (1998). *Applied regression analysis* (Vol. 326). John Wiley & Sons.

Fisher, R. A. (1922). On the mathematical foundations of theoretical statistics. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, *222*(594-604), 309–368.

Gabriel, K. R., & Zamir, S. (1979). Lower rank approximation of matrices by least squares with any choice of weights. *Technometrics*, *21*(4), 489–498.

Golub, G. H., & Van Loan, C. F. (1985). Matrix computations. 1983. *Johns Hopkins, Baltimore, MD*.

Hale, E. T., Yin, W., & Zhang, Y. (2007). A fixed-point continuation method for l1-regularized minimization with applications to compressed sensing. *CAAM TR07-07, Rice University*, *43*, 44.

Hastings, C., Mosteller, F., Tukey, J. W., Winsor, C. P., et al. (1947). Low moments for small samples: a comparative study of order statistics. *The Annals of Mathematical Statistics*, *18*(3), 413–426.

He, J., Balzano, L., & Szlam, A. (2012). Incremental gradient on the grassmannian for online foreground and background separation in subsampled video. In *2012 ieee conference on computer vision and pattern recognition* (pp. 1568–1575).

Hoerl, A. E., & Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, *12*(1), 55–67.

Hotelling, H. (1992). Relations between two sets of variates. In *Breakthroughs in statistics* (pp. 162–190). Springer.

Hunt, R. (2004). The reproduction of colour sixth edition. *John Wiley & Sons*, 48.

Johnson, D. H. (2006). Signal-to-noise ratio. *Scholarpedia*, *1*(12), 2088.

Jolliffe, I. (2003). Principal component analysis. *Technometrics*, *45*(3), 276.

Kim, K. I., Franz, M. O., & Scholkopf, B. (2005). Iterative kernel principal component analysis for image modeling. *IEEE transactions on pattern analysis and machine intelligence*, *27*(9), 1351–1366.

Kim, K. I., Jung, K., & Kim, H. J. (2002). Face recognition using kernel principal component analysis. *IEEE signal processing letters*, *9*(2), 40–42.

Kleinbaum, D. G., Kupper, L. L., Muller, K. E., & Nizam, A. (1988). *Applied regression analysis and other multivariable methods* (Vol. 601). Duxbury Press Belmont, CA.

Lee, J.-M., Yoo, C., Choi, S. W., Vanrolleghem, P. A., & Lee, I.-B. (2004). Nonlinear process monitoring using kernel principal component analysis. *Chemical engineering science*, *59*(1), 223–234.

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*.

mihoyo. (2019). Retrieved from `https://honkaiimpact3.mihoyo.com/global`

Moore, B., Gao, C., & Nadakuditi, R. R. (2019). Panoramic robust pca for foreground-background separation on noisy, free-motion camera video. *IEEE Transactions on Computational Imaging*.

Murphy, G. J. (2014). *C*-algebras and operator theory*. Academic press.

Pearson, K. (1901). Principal components analysis. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, *6*(2), 559.

satsumannoyaji. (2019). Retrieved from `https://www.youtube.com/watch?v=n3pTSvQSH_E`

Schölkopf, B., Smola, A., & Müller, K.-R. (1997). Kernel principal component analysis. In *International conference on artificial neural networks* (pp. 583–588).

Seber, G. A., & Lee, A. J. (2012). *Linear regression analysis* (Vol. 329). John Wiley & Sons.

Spears, B. (2019). *What happens on the ground when you are landing.* Retrieved from https://www.youtube.com/watch?v=nRQ7oIu7IYo&t=5s

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, *58*(1), 267–288.

Wall, M. E., Rechtsteiner, A., & Rocha, L. M. (2003). Singular value decomposition and principal component analysis. In *A practical approach to microarray data analysis* (pp. 91–109). Springer.

Wang, Z., Bovik, A. C., & Lu, L. (2002). Why is image quality assessment so difficult? In *2002 ieee international conference on acoustics, speech, and signal processing* (Vol. 4, pp. IV–3313).

Wold, S., Esbensen, K., & Geladi, P. (1987). Principal component analysis. *Chemometrics and intelligent laboratory systems*, *2*(1-3), 37–52.

Wright, J., Ganesh, A., Rao, S., Peng, Y., & Ma, Y. (2009). Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization. In *Advances in neural information processing systems* (pp. 2080–2088).

Xiao, W., Huang, X., He, F., Silva, J., Emrani, S., & Chaudhuri, A. (2019). Online robust principal component analysis with change point detection. *IEEE Transactions on Multimedia*.

Xu, L., & Yuille, A. L. (1995). Robust principal component analysis by self-organizing rules based on statistical physics approach. *IEEE Transactions on Neural Networks*, *6*(1), 131–143.

Zhang, T., & Yang, B. (2016). Big data dimension reduction using pca. In *2016 ieee international conference on smart cloud (smartcloud)* (pp. 152–157).

Zou, H., Hastie, T., & Tibshirani, R. (2006). Sparse principal component analysis. *Journal of computational and graphical statistics*, *15*(2), 265–286.