

中图分类号: TP391
学科分类号: 5206040

论文编号: 163102403
密 级:

天津理工大学研究生学位论文

基于随机探测和鲁棒主成分分析的 子空间学习方法及其视频处理应用

(申请博士学位)

一级学科: 计算机科学与技术

研究方向: 图像处理与计算机视觉

作者姓名: 李阳

指导教师: 陈胜勇 教授 刘光灿 教授

2020 年 06 月

**Thesis Submitted to Tianjin University of Technology
for the Doctor's Degree**

**Subspace Learning via Random Sample Probing
and Robust PCA and its Application in Video
Processing**

By
Yang Li

Supervisor
Shengyong Chen
Guangcan Liu

June 2020

摘要

视频处理是人工智能中最基本的任务之一，其结果直接影响到后续的图像识别、分类、行为检测等结果。大量的研究表明，高维的视频数据并不是杂乱无章，通过研究我们发现高维的数据都会存在于低维的子空间中。子空间学习是视频分析中一个被广泛使用且十分有效的技术，已被成功应用于视频压缩、目标识别等任务中。本文主要对子空间分割、子空间恢复和子空间压缩在视频处理中存在的四个问题进行研究，具体贡献如下：

当前广泛使用的基于谱聚类的方法，如稀疏子空间聚类（Sparse Subspace Clustering, SSC）等，时间复杂度为 $O(n^3)$ ，也不能很好地解决大规模数据分割问题。为了实现对大规模数据的快速聚类，本文提出了一种名叫随机探测（Random Sample Probing, RANSP）的快速子空间分割方法，其基本思想是通过随机选择种子点，并使用岭回归（Ridge Regression, RR）计算其它点与种子点的相关性，从而快速得到一个子空间的样本。通过使用 Woodbury 公式优化 RR 求解过程，RANSP 的时间复杂度可达到线性，因此可以对大规模数据集进行快速子空间分割。

鲁棒主成分分析（Robust Principal Component Analysis, RPCA）的目标在于从污染的数据中获得低秩和稀疏部分，已被广泛地用于计算机视觉中、图像处理等领域中。然而，RPCA 还存在可以被改进的地方，比如对核范数和稀疏项的约束。为了提高 RPCA 模型的子空间恢复能力，我们提出了一种截断加权（Truncated Weighting）的 RPCA 模型（TWRPCA）。在 TWRPCA 中，我们保留前 10% 的奇异值不变，其它的奇异值分别给予不同的权重。最后我们使用非精确增广拉格朗日乘子法（Inexact Augment Lagrangian multiplier, IALM）对 TWRPCA 模型进行了求解。

基于 RPCA 子空间学习的方法及其扩展的模型被成功地应用到运动目标检测中，但是不能较好地处理运动目标缓慢的情况，容易造成“空洞”的现象，尤其是在复杂背景下，如晃动的树叶、波动的湖面等。为了能够在复杂环境中检测运动缓慢的目标，针对动态背景使用了视频分割约束，针对缓慢运动采取了显著性约束，最后本文将视频分割（Segmentation）和显著（Salience）性约束融入到 RPCA 模型中得到 SSC-RPCA 模型。与其他文献不同，SSC-RPCA 中的显著信息不需要提前计算而是在模型求解过程中得到的。

自编码器被广泛地使用在图像和视频压缩中，但传统的自编码器需要存储大量的网络参数。而且传统的自编码器需要特定的数据来训练网络，因此其泛化能力较弱。本文提出了一种 3 阶张量自编码器（3D Tensor Auto Encoder, 3DTAE）

子空间压缩方法，视频被压缩到非线性子空间中。在传统的基于自编码器的视频压缩方法中，视频被表示为网络参数和向量，而在本文方法中，视频被直接表示为网络参数。假设输入数据的维度是 $O(n)$ ，那么 3DTAE 网络参数的维度将是 $O(n^{1/3})$ ，因此可以满足视频压缩的需求。另外，由于视频被直接压缩为网络参数，基于 3DTAE 的视频压缩方法不存在测试过程，因此也不存在泛化问题。

关键词：子空间学习，子空间分割，子空间恢复，运动目标检测，张量网络

Abstract

Video processing is one of the most basic tasks in artificial intelligence, and its results directly affect subsequent image recognition, image classification, behavior detection and so on. A large number of studies show that the high-dimensional video data is not disordered. Through the study we find that the high-dimensional data exist in the low-dimensional subspace. Subspace learning is a widely used and effective technique in video analysis and has been successfully applied in video compression, target recognition and so on. We mainly study the four problems of subspace segmentation, subspace recovery and subspace compression in video processing. Our contributions are as follow:

The current subspace segmentation method based on spectral clustering, such as: Sparse Subspace Clustering (SSC) can not satisfy the segmentation of large data points, the time complexity is $O(n^3)$. For fast segmenting large scale data, this paper proposes a fast subspace segmentation method called Random Sample Probing (RANSP), which randomly selects seed points and uses Ridge Regression (RR) to calculate the correlation between other points and this point, so that you can get a subspace. The WoodBury formula is used to optimize the RR solution so that the time complexity of RANSP is linear, so large-scale data points can be processed quickly.

Robust principal component analysis (RPCA) aims to recover low rank and sparse parts from polluted data, which has been widely used in computer vision, image processing and so on. However, there are still areas could be improved in RPCA model. In order to improve the subspace recovery ability of RPCA model, we propose a Truncated Weighted RPCA (TWRPCA) model. In TWRPCA model, we keep the first 10% of the singular value unchanged, and give different weights to the other singular values. At last the Inexact Augument Lagrangian Multiplier (IALM) is used to solve the TWRPCA model.

RPCA subspace learning method and its extended model are successfully applied to moving object detecion, but can not deal with the slow moving target, which is easy to cause "cavity", especially in complex backgrounds, such as swaying leaves, fluctuating lakes, etc. In order to detect slow moving targets in complex environments, video segmentation constraints are used for dynamic backgrounds, and salience constraints are imposed for slow motions. Finally, this paper integrates video segmentation and salient constraints into the RPCA model. The SSC-RPCA model is obtained. Unlike other literatures, the significant information of this paper is not

calculated in advance but obtained by solving the SSC-RPCA model.

Auto-encoder has been widely used in image and video compression, but traditional auto-encoder needs to store a large number of network parameters. Traditional auto-encoder requires specific data to train the network, so their generalization capabilities are weak. This paper proposes a 3D Tensor Auto Encoder (3DTAE) subspace compression method, in which video is compressed into a nonlinear subspace. In the traditional auto-encoder methods, the video is represented as network parameters and vectors. In this method, the video is directly represented as a network parameter. Assuming the dimension of the input data is n , and then the dimension of the network parameter is also $O(n^{1/3})$, which can meet the needs of video compression. In addition, because the video is compressed directly into network parameters, there is no test process in 3DTAE, so there is no generalization problem.

Key words: Subspace Learning, Subspace Segmentation, Subspace Recovery, Moving Object Detection, Tensor Network

目 录

第一章 绪论.....	1
1.1 引言.....	1
1.2 研究内容.....	2
1.3 我们的贡献.....	5
1.4 章节安排.....	7
第二章 国内外研究现状.....	9
2.1 子空间分割.....	9
2.2 子空间恢复.....	12
2.3 运动目标检测.....	14
2.4 视频压缩.....	17
2.5 总结.....	19
第三章 子空间学习基础算法研究.....	21
3.1 基于随机探测的快速子空间分割算法.....	21
3.1.1 引言.....	21
3.1.2 RANSP 模型.....	22
3.1.3 子空间分割实验.....	28
3.2 基于截断加权 RPCA 的子空间恢复算法.....	35
3.2.1 引言.....	35
3.2.2 TWRPCA 模型.....	37
3.2.3 子空间恢复实验.....	39
3.3 总结.....	42
第四章 子空间学习在视频处理中的应用研究.....	43
4.1 基于分割和显著约束的 RPCA 子空间恢复算法.....	43
4.1.1 引言.....	43
4.1.2 SC-RPCA 模型与 SSC-RPCA 模型.....	44
4.1.3 运动目标检测实验.....	50
4.2 基于 3D 张量自编码器的子空间压缩算法.....	58
4.2.1 引言.....	58
4.2.2 张量符号与基础知识.....	59
4.2.3 3DTAE 模型.....	60
4.2.4 视频压缩实验.....	66

4.3 总结.....	74
第五章 总结与展望	75
5.1 总结.....	75
5.2 展望.....	76
参考文献	79
发表论文和科研情况说明	93
致 谢	95

第一章 绪论

1.1 引言

随着信息技术的不断发展，城市的信息化水平不断提高，智慧城市的建设应运而生，许多发达国家积极开展建设智慧城市。迪比克市是美国的第一个智慧城市，同样也是世界上第一个智慧城市，其与国际商业机器公司进行合作建设，应用了智能化技术对城市进行建设，使用物联网技术将城市中所有的数字设备进行连接，比如在小区中将城市中的各种资源（水、电、交通等）连接起来，分析数据，做好调度工作，这样便可以更加高效和有效地服务市民同时可以对资源进行有效的预算进而减少资源的浪费，更好的服务市民。2006 年，新加坡启动“智慧国 2015 计划”，通过物联网等信息技术把新加坡建设成为经济、生态一流的国际化都市，其中在智能交通信息方面取得了引人注目的成绩。在“智慧城市”中，人民将尽享智慧家居、路网监控、智能医院、数字生活等带来的生活便利，如图 1-1。

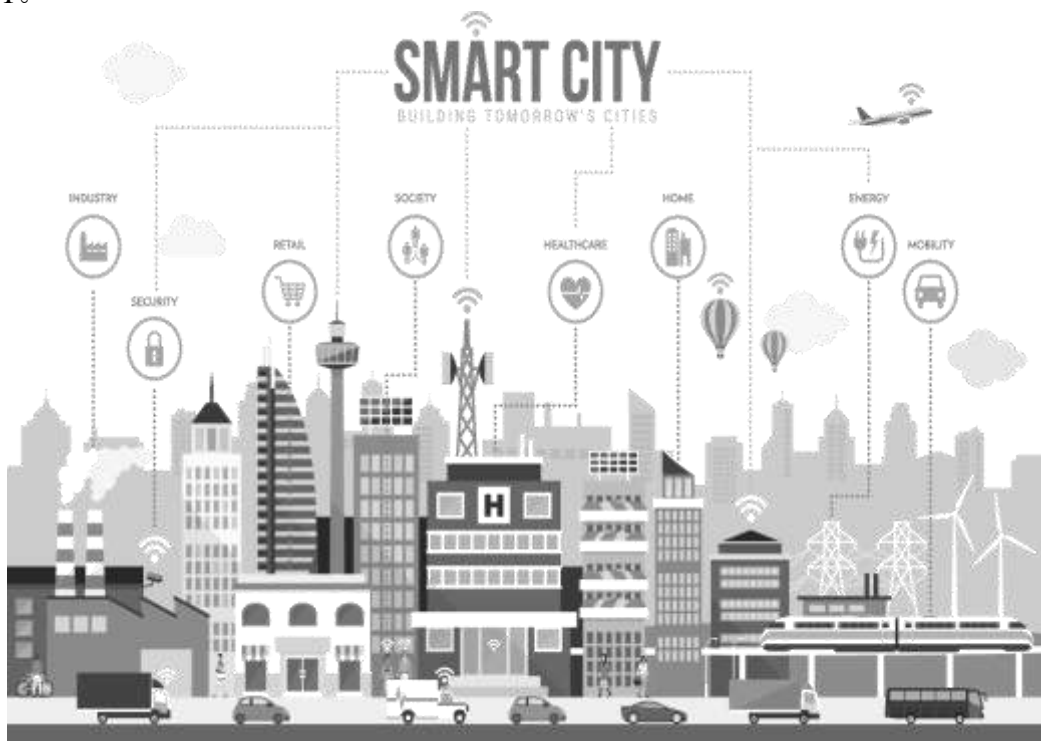


图 1 - 1 智慧城市（图片链接：<https://vmaker.tw/archives/32484>）

随着我国改革开放的推进，老百姓生活水平显著提高，人们对于生活的质量

要求也越来越高，但同时随着城市人口比例不断提高，随之而来的如交通拥堵、安全隐患、资源短缺等困扰着各个城市，为了让城市更加健康的发展，政府启动了国家智慧城市试点工作^[1]，智能城市的应用项目主要有智能安居服务、智能交通、智慧公共服务等等。随着我国智慧城市^[2]的不断推进建设，对信息技术的要求也越来越高。截至目前，全球已经有超过 1000 个城市明确提出或者即将要建设智慧城市，其中有一半的智慧城市建设在中国。

在智慧城市的建设中，主要关注安防、交通、社区家庭和商业，其中最重要的是安防。安全对于任何一个城市来说都是最重要的，安全是城市稳定发展的基础、同样也是社会健康发展的保障。在城市安防建设中，不可或缺的技术就是视频技术，其发挥的作用至关重要。视频技术不单单可以用于安防，视频技术还可以用于智慧城市建设的其它方面，从普通大众的日常，如视频点播、视频会议等，再到社会的安全，如交通监控、车站安防、智能交通、环境监测，再大到国家安全，如雷达探测、及时天气预测等等方面，被广泛的应用。

截止目前，学者们对视频技术的研究主要有视频获取、视频增强、视频去模糊、视频编码、视频显著检测、运动目标检测、目标检测、目标跟踪、视频场景理解、视频预测等等。本文根据对视频研究的智能化程度将视频研究的方向分为两个部分：低级视频研究（视频处理）和高级视频研究（视频理解），具体如表 1-1，视频处理不关注视频中有什么，而是为了后面的视频理解提供高质量有意义的数据，只是做了一些预处理，而视频理解则需要关注视频中有什么？在做什么？即将要做什么？主要在于视频的认知层面上。

表 1 - 1 视频研究方向

智能化程度	研究方向
低	视频增强、视频去噪、视频去模糊、视频显著检测、运动目标检测、视频压缩、视频分割、质量评价、场景深度估计
高	视频预测、行人再识别、目标检测、视频检索、异常行为检测

1.2 研究内容

近些年，子空间学习被成功且广泛的使用在视频处理^[3]、计算机视觉^{[4][5]}、模式识别中^[6]。本文主要使用子空间学习对视频处理存在的以下四个问题进行研究。这四个问题从理论上主要分为三个部分：子空间分割、子空间恢复和子空间压缩，其中我们在子空间分割中做了创新工作，提出了子空间分割模型 **RANSP**，在子空间恢复中提出了改进子空间恢复能力的工作，提出了 **TWRPCA** 模型；在子空间恢复中做了改进工作，提出了 **SSC-RPCA** 模型，并应用到了运动目标检

测中；在子空间压缩中提出了全新的 3DTAE 子空间压缩模型，并将它应用到视频压缩中，具体的关系如图 1-2：

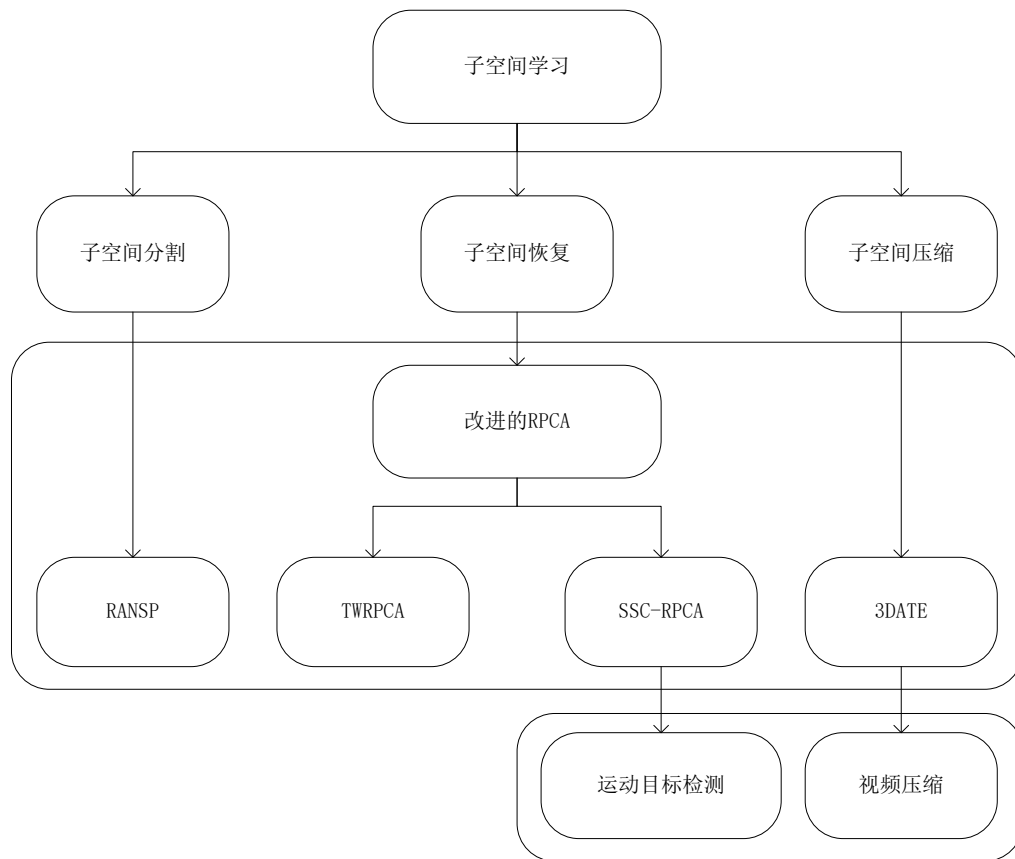


图 1 - 2 研究内容

（1）子空间分割。在实际的图像和视频处理中，常常面临一个问题就是数据来自于多个类别或者是属于多个类别，因为不同类别的数据有着不同的特点，在处理它们时需要采用不同的策略和方法，因此需要对数据进行聚类，再对每个类别进行分别处理，从而保证对不同的数据采用不同的策略进行处理，做到有针对性的处理，实现精准处理。除了以上的这个问题，还会遇到另一个问题，因为视频数据属于高维数据，占用空间非常大，需要消耗大量的资源进行存储和传输，因此需要对视频数据在不丢失主要信息的前提下进行压缩，减少资源的消耗。在处理以上两个问题时，子空间分割是一个非常有力的工具。

子空间分割是图像视频聚类、视频压缩的基础分析方法，子空间分割通常以数据全局线性为前提，将每个样本点表示为其它样本点的线性组合，其被广泛的使用在机器学习^[7]、计算机视觉、数据挖掘、信号处理、模式识别^[8]等应用中。比如在数据压缩中，子空间学习可以寻找高维数据到低维空间的映射。在数据挖掘中，子空间可以发现数据点的低维表示，可以实现数据的分类与聚类，比如在图像分割中也得到了应用。

(2) 子空间恢复。从受到污染的数据中发现数据的低维结构就是子空间恢复，如图 1-3 所示。在许多领域的应用中，都需要进行大量的数据收集，然后对收集的数据进行分析，大量的数据包含了丰富的信息，但是同时也给分析工作增加了工作量。且每个数据之间还存在联系，不能主观随意地丢弃一些数据，因此需要对数据进行适当地进行降维，在保证原有数据的主要信息时，尽可能地舍弃影响很小或者没有影响的数据，这样虽然增大了挑战性但大大提高了后期数据处理的工作效率。

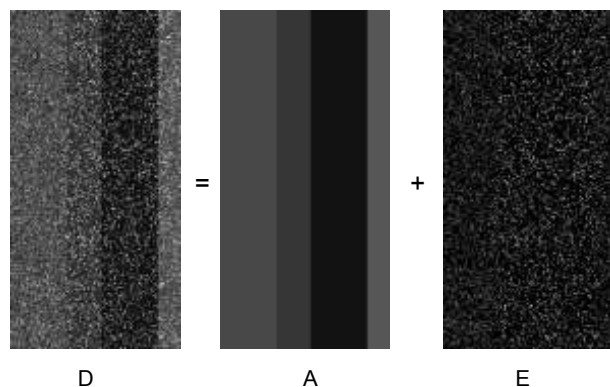


图 1 - 3 子空间恢复

常见且有效的数据降维方法如 PCA，当数据受到密集的、小的高斯噪声污染时，PCA 可以有效地得到数据的主要成分，即有效地进行子空间恢复。当数据受到大的离群点噪声污染时，如椒盐噪声，PCA 则不能有效地进行子空间恢复。

(3) 运动目标检测。前面的子空间学习理论研究是为了后期的视频研究做铺垫，是一个基础性的工作，为了视频分析提供了有力的帮助。在当今视频技术的广泛使用下，比如安防，成千上万的摄像头每天采集了海量的数据，在这些数据中，包含了很多重要的有价值的信息，同时也包含了大量的无用的信息。如果只依靠人工进行视频数据的分析虽然准确率较高但是效率则非常低下，而且如果人工在长时间工作后会出现疲劳则会造成工作失误，解决问题的方法就是智能视频分析。视频智能化发展是指将机器学习技术对视频数据进行目标分析，在监控视频中加入视频分析模块，借助计算机强大的计算能力来过滤掉无关的信息，从视频中抽取关键有用的信息，从而有效地进行事前预警及后期的取证^[9]，如在监控画面中实时的犯罪行为等。从传统的被动监控转变成为主动监控，不但可以将人工从繁重的任务上解救出来，还可以更加快速、准确地达到目的。

运动目标检测^[10]是从视频中将前景和背景进行分离，如图 1-4 所示，在智慧视频分析中，其是一个基础的且重要的步骤，被广泛的运用在虚拟现实、行为分析和场景理解中。在很简单的场景下，运动目标检测则是一个非常简单的任务，

检测的正确率也会非常高。然而在实际的场景则非常复杂，比如：晃动的树叶、摇摆的旗帜、波动的湖面、缓慢的运动物体等，这样便把运动目标检测变成了一个非常具有挑战性的问题。



图 1 - 4 运动目标检测

（4）视频压缩。随着视频技术的快速发展，视频得到了广泛的使用。目前视频的数据量占据了互联网所有数据量的 80%，成为了全球最大的数据消费来源。且随着对视频服务质量需求的不断提高，视频向高清化和网络化发展，摄像头能获取高分辨率的图片，从而可以提取到更多有效的信息，才能让我们能看的更清晰，视频在网络中进行传输，数据量越来越大，每年视频数据在呈爆炸式的增长。这对互联网的传输和存储，多媒体的计算提出了新的挑战。比如：一分钟未经压缩的分辨率为 1024×1024 、帧率为 25 的三通道彩色视频需要的存储空间为 4.4GB，这对视频的压缩技术提出了新的难题。



图 1 - 5 视频压缩

常见的数据压缩分为两种，有损压缩和无损压缩。其中无损压缩主要有霍夫曼编码、游程编码和算术编码等等，这些方法可以毫无损失地对原始数据进行压缩表示，但是压缩率有限，压缩比一般为 2:1 到 5:1。如果对视频采用无损压缩，压缩后的数据仍然需要占据很大的空间，因此对视频压缩需要采用有损压缩，如图 1-5 所示。视频有损压缩就是在不减少视频中重要信息的前提降低计算机中所存储的或者网络链路中所传播的数据的冗余性，从而提高传输效率，及时发送网络视频帧数据，及时对数据进行译码，降低传输成本^{[11][12]}。

1.3 我们的贡献

子空间分割在机器学习、计算机视觉、数据挖掘等领域中拥有广泛应用。早期的 RANSAC^[13] 方法、K-subspace^[14]、稀疏子空间聚类^[15] (Sparse Subspace Clustering, SSC) 等不能很好地解决大规模数据分割问题。为了实现对大规模数据的快速聚类,本文提出了一种名叫随机探测(Random Sample Probing, RANSP)的快速子空间分割方法,其基本思想是通过随机选择种子点,并使用岭回归(Ridge Regression, RR)计算其它点与种子点的相关性,从而快速得到一个子空间的样本,这样如果有 n 个子空间,我们只需要进行 n 次以上的步骤就可以得到 n 个子空间。然后我们计算每个不属于任何子空间的点到已经得到子空间的距离,并将这些点划分到最近的子空间。其中通过使用 Woodbury 公式优化 RR 求解过程, RANSP 的时间复杂度可达到线性,因此可以对大规模数据集进行快速子空间分割。

鲁棒主成分分析^[16] (Robust principal component analysis, RPCA) 目标在于从污染的数据中获得低秩和稀疏部分,已被广泛地用于计算机视觉中、图像处理等领域。然而, RPCA 还存在可以被改进的地方,比如对核范数和稀疏项的约束。为了提高 RPCA 模型的子空间恢复能力,我们提出了一种截断加权(Truncated Weighting)的 RPCA 模型(TWRPCA)。在 TWRPCA 中,我们保留前 10% 的奇异值不变,其它的奇异值分别给予不同的权重。这样做的优势在于,可以给予较大的奇异值较大的权重,给予较小的奇异值较小的权重,值越大的奇异值将会受到越小的惩罚,更易于保持数据的主要方向。最后我们使用非精确增广拉格朗日乘子法(Inexact Augment Lagrangian multiplier, IALM)对 TWRPCA 模型进行了求解。

运动目标检测在计算机视觉中扮演着重要的角色,其被广泛的使用在虚拟现实、行为分析和场景理解中。基于 RPCA 子空间学习的方法及其扩展的模型被成功地应用到运动目标检测中,但是不能较好地处理运动目标缓慢的情况,容易造成“空洞”的现象,尤其是在复杂背景下,如晃动的树叶、波动的湖面等。为了能够在复杂环境中检测运动缓慢的目标,我们提出了一种基于分割和显著约束的 RPCA 运动目标检测算法。因为运动目标具有时间和空间的连续性,所以对动态背景使用了视频分割约束;因为运动目标具有在场景中具有显著特性,所以对缓慢运动目标采取了显著性约束,最后本文将视频分割(Segmentation)和显著(Saliency)性约束融入到 RPCA 模型中得到 SSC-RPCA 模型。与其他文献不同, SSC-RPCA 中的显著信息不需要提前计算而是在模型求解过程中得到的。

视频压缩可以有效地减少视频的存储空间,以便于视频的存储、传输等。自编码器被广泛地使用在图像和视频压缩中,但传统的自编码器需要存储大量的网络参数,具体地,当输入数据的维度是 $O(n)$,那么网络参数的维度也是 $O(n)$ 。

而且传统的自编码器采用特定的数据来训练网络，因此泛化能力较弱。本文提出了一种 3 阶张量自编码器（3D Tensor Auto Encoder, 3DTAE）视频压缩方法，原始视频被压缩到非线性子空间中。在传统的基于自编码器的视频压缩方法中，视频被表示为网络参数和向量，而在本文方法中，视频被直接表示为网络参数。假设输入数据的维度是 $O(n)$ ，那么 3DTAE 网络参数的维度将是 $O(n^{1/3})$ ，因此可以满足视频压缩的需求。另外，由于视频被直接压缩为网络参数，基于 3DTAE 的视频压缩方法不存在测试过程，因此也不存在泛化问题。

1.4 章节安排

本文接下来的章节安排大致如下：

第二章介绍了子空间分割、子空间恢复视频压缩和运动目标检测的国内外研究现状。

第三章介绍了子空间学习基础算法研究，第一部分是基于随机探测的快速子空间分割算法：子空间分割的定义和当前子空间分割方法面临的问题，接着引入了本文的方法，分别从不含噪声的子空间分割和包含噪声的子空间分割进行介绍，然后使用 WoodBury 公式对本文方法进行了优化，使本文方法的时间复杂度为线性，并给出了最终的时间复杂度，再使用 PIE 人脸数据库^[17]、Fashion-MNIST 数据库^[18]和合成的数据库对本文方法和对比方法进行了对比实验。第二部分是基于截断加权 RPCA 的子空间恢复算法，介绍了 RPCA 在子空间恢复上的模型，为了提高 RPCA 模型的子空间恢复能力，我们提出了一种截断加权（Truncated Weighting）的 RPCA 模型（TWRPCA）。具体地，保留前 10% 的奇异值不变，其它的奇异值分别给予不同的权重。然后再使用 IALM 对 TWRPCA 模型进行了求解，接着在合成数据库中进行了对比实验。最后我们进行了总结。

第四章介绍了子空间学习在视频处理中的应用研究，主要分为以下两个部分：第一部分提出了基于视频分割和显著特性约束的 RPCA 子空间恢复算法，并将其应用目标检测方法，我们首先介绍了 RPCA 在运动目标检测中的优点及不足之处，然后将运动目标的时间和空间连续性纳入 RPCA 模型中得到了 SC-RPCA 模型，并对这个模型进行测试以及与 RPCA 模型比较，再将显著特征纳入 SC-RPCA 模型中得到了 SSC-RPCA 模型，最后详细地介绍了 SSC-RPCA 模型的求解推导以及模型的优化，并在 CDNET2014^[19]数据库上进行实验测试，分析了 SSC-RPCA 模型与其他模型的优势和差距。第二部分是基于 3D 张量自编码器的子空间压缩算法，并将其应用到视频压缩中，使用非线性子空间来表示视频，自编码器的压缩方法的特点和一些本文方法的基础知识，再引入了本文的方法-基

于 3D 张量自编码器的深度学习视频压缩方法，本文网络框架模型及其优势，模型的目标函数以及反向传播在本文方法中的参数更新，模型的优化与加速和模型的自动生成机制，最后测试了公共数据库中的视频序列。最后进行了总结。

第五章介绍了对全文进行了总结并进行了展望，主要介绍了本文方法的特点及其不足之处，并给出了下一步改进的计划。

第二章 国内外研究现状

本章介绍了子空间分割、子空间恢复、运动目标检测和视频压缩的国内外研究现状。

2.1 子空间分割

在过去的一些年中，子空间分割得到了广泛关注，大量的方法被提出，得到了实际的应用。1981 年，Fischler 等人提出了 Random Sample Consensus (RANSAC)^[13]方法，其被广泛地应用在计算机图形学中。该方法是一种基于统计的方法，在一个包含异常值的数据点集中找到一个合适的子空间的方法，在每一次的迭代中，RANSAC 随机地选择 r 个点来计算出一个模型，并用这个模型来测试其它的数据，如果某个点适用于这个模型，那么这个点被认为是局内点，如果有足够多的点被认为是局内点，那么得到的模型就足够合理，由于其高效性和有效性被广泛使用。但是，当其处理高维数据点时，计算复杂度呈级数增长，不适合处理高维数据，并且随着数据维度的增加，其分割的准确性不断下降。

2003 年，Ho 等人提出了 K-Subspace^[14]方法，K-subspace 是一种类似于 K-means^[20]的方法，其首先随机初始化子空间的基，然后根据子空间的基来计算数据点到每个子空间的距离，再将数据点分配到距离其自己最近的子空间，并用新的数据点来更新子空间的基，再重复以上步骤，不停迭代直到最大迭代次数或者误差小于设定的阈值，最终得到所有的子空间。K-Subspace 可以进行高维数据点的子空间分割处理，但是容易产生局部最小值，因此在许多的子空间分割任务中限制了其能力。

2005 年，Vidal 等人提出了广义主成分分析 (Generalized Principal Component Analysis, GPCA)^[21]，GPCA 是一种基于代数法的方法，其将样本聚类到不同的低维子空间中，被广泛地使用到了数据聚类、分局分割中，但是其算法的复杂度为指数级，无法高效地处理高维数据。

基于谱聚类的方法如：稀疏子空间聚类 (Sparse Subspace Clustering, SSC)^[15]、低秩子空间聚类 (Low-Rank Representation, LRR)^[22]、最小二乘回归 (Least Squares Regression, LSR)^[23]和阈值岭回归 (Thresholding Ridge Regression, TRR)^[29]等，基于谱聚类的子空间分割方法主要分为两个步骤，第一步是先学习到一个相似矩阵，这个矩阵包含了数据点的子空间隶属度，然后再使用谱聚类 Ncut^[24]

分割相似性矩阵 $|Z| + |Z^T|$ ，这些方法的不同之处在于获得相似性矩阵的步骤。

2009 年，Elhamifar 等人^[15]提出了 SSC 方法，SSC 假设高维数据分布在若干个低维子空间上，在适当的字典下，对高维数据的表示具有稀疏性，其核心是设计能够揭示高维数据真实子空间结构的表示模型，使得到的表示系数及由此构造的相似度矩阵有助于精确的子空间聚类，其时间复杂度为 $O(n^3)$ ，且在运算时需要大量的内存空间，其形式如下：

$$\min \|Z\|_1, \quad \text{s.t. } X = XZ, \quad \text{diag}(Z) = 0 \quad (2-1)$$

其中 $\|\cdot\|_1$ 可以看做是矩阵的 L1 范数。

2010 年，Liu^[22]提出了 LRR 方法，LRR 方法与 SSC 方法非常类似，区别就在于 LRR 方法寻找的目标是数据的低秩表示，而 SSC 方法则是寻找数据的稀疏表示，LRR 将秩范数表示为核范数形式，通过求解核范数最小化实现优化求解，其形式如下：

$$\min_{Z, E} \|Z\|_* + \lambda \|E\|_{2,1}, \quad \text{s.t. } X = AZ + E \quad (2-2)$$

其中 $\|\cdot\|_*$ 是矩阵的核范数。

学者们不断跟进，提出了一些扩展版本^{[25][26]}，如 2012 年，Liu 等人^[27]提出了一种隐式的低秩子空间分割方法 Latent LRR，如下：

$$\min_Z \|Z\|_* + \|L\|_* + \lambda \|E\|_1, \quad \text{s.t. } X = XZ + LX + E \quad (2-3)$$

其中 $L \in R^{m \times n}$ 为投影矩阵。

当观测数据矩阵数据不充分或者严重损坏时，可能会降低 LRR 的性能，为此通过使用观察到的和未观察到的隐藏数据来构造字典，结果表明，通过求解核范数极小化问题，可以近似地恢复隐藏数据的影响，核范数极小化问题是凸的，可以有效地求解。该方法将子空间分割和特征提取无缝地集成到一个统一的框架中，为子空间分割和特征提取提供了一种解决方案。作为子空间分割算法，LATLRR 是 LRR 的增强版本并且优于 LRR。Lu 等人^[23]提出了一种鲁棒的基于最小二乘回归（LSR）的子空间分割方法，充分利用了数据之间的相关性，在处理有噪声数据时，其定义如下：

$$\min \|X - XZ\|_F^2 + \lambda \|Z\|_F^2, \quad \text{s.t. } \text{diag}(Z) = 0 \quad (2-4)$$

其中类似于 SSC，LSR 同样也使用 F 范数， λ 用于平衡两部分，其具有以下的解

析解：

$$Z^* = -D(diag(D))^{-1} \text{ and } diag(Z^*) = 0 \quad (2-5)$$

其中 $D = (X^T X + \lambda I)^{-1}$ 。

当公式 (2-5) 中的约束 $diag(Z) = 0$ 移除后，其就变成如下的形式：

$$\min \|X - XZ\|_F^2 + \lambda \|Z\|_F^2 \quad (2-6)$$

其中上式同样也具有一个解析解：

$$Z^* = (X^T X + \lambda I)^{-1} X^T X \quad (2-7)$$

从公式 (2-5) 和公式 (2-7) 可以看出，LSR 两种形式解析解都需要进行对矩阵求逆操作，当处理的数据点的个数为 n ，其时间复杂度 $O(n^3)$ 。

2013 年，Wang 等人^[28]提出了一种低秩稀疏模型用来同时捕获数据的全局和局部结构。2015 年，Peng 等人^[29]提出了一种新的鲁棒子空间分割方法，该方法从投影空间而不是输入空间消除误差对聚类的影响。2018 年，Brbic 等人^[30]提出了多视图谱聚类框架（Multi-view Low-rank Sparse Subspace Clustering, MLRSSC），同时学习数据的低秩和稀疏特性。L1 范数保证了相似性矩阵中的子空间的保持性（簇间没有连通），很少考虑到连通性（簇内完全连通）。Li 等人^[31]结合了稀疏表达和扩散过程提出了一种新的相似性矩阵的学习方法，可以捕获局部的流形过程。

传统的子空间聚类方法都是通过测量给定特征空间中点的相似性来进行聚类，因此子空间的选择和聚类的过程是紧密耦合的。Zhu 等人^[32]提出了一种名为共享子空间聚类的方法（Clustering by Shared Subspaces, CSSub），它可以根据共享子空间的数量来聚集相邻的核心点，子空间的选择和聚类划分为两个独立的过程。在一般情况下，谱聚类方法是非常有效的子空间分割方法，但是当数据点的数量巨大时，其效率则非常低下，不能满足应用。

在大数据的背景下，快速的子空间分割方法成为了一种迫切的需要。为了达到快速的子空间分割效果，2013 年，Peng 等人提出了扩展的稀疏子空间聚类（Scalable SSC, SSSC）、扩展的低秩子空间聚类（Scalable LRR, SLRR）和扩展的最小二成回归（Scalable LSR, SLSR）^[33]，这些方法首先从全部的数据点中选取一小部分数据点并对这部分数据点进行子空间分割，然后未被选取的数据点被分配给距离上一步得到的子空间中最小残差的子空间，在此方法中，SSC 步骤的时间复杂度仍然相对较高。

在 2014 年，Zhang 等人^[34]提出了另一种快速子空间分割方法，主要是对子

空间聚类方法中时间复杂度比较高的步骤进行了优化，其分为三个步骤：第一步使用部分奇异值分解（Partial Singular Value Decomposition, PSVD）来逼近低秩子空间的解；第二步使用局部敏感哈希（Locality Sensitive Hashing, LSH）建立稀疏相似矩阵；第三步使用快速的归一化割来得到子空间分割结果，该方法时间复杂度为 $O(n \log(n))$ 。Feng 等人^[35]提出了一种大规模优化的随机拟牛顿方法。2015 年，Xiao 等人^[36]提出了一种快速的 LRR 求解方法，它使用了分布式的框架，在现有的 LRR 基础上实现了数量级的提升，在理论上保证了获得的解是最优的。在 2016 年，Shen 等人^[37]提出了一种在线建模的低秩子空间聚类方法（Online Low-Rank Subspace Clustering, OLRSC），该方法有效地将 LRR 对内存的需要从 $O(n^2)$ 减少到了 $O(pd)$ ， $d < p \leq n$ 。

现有的方法使用浅层模型来估计数据点的子空间，但浅层模型的表达能力有限，Peng 等人^[38]提出了一种新的基于结构化自编码器的子空间分割方法，同时保持子空间的全局和局部结构，具有较强的表达能力。这些方法都明显地降低了原有方法的时间复杂度。

2.2 子空间恢复

在对数据进行恢复时，主要通过针对核范数和稀疏项采用不同的策略来提高恢复的性能^{[39][40]}。在核范数最小化的过程中，所有的奇异值没有被区分而是都被一起最小化，不利于近似秩，为了克服核范数最小化不能较好地逼近核范数的问题，截断核范数正则化方法（Truncated Nuclear Norm Regularization term, TNNR）^{[41][42]}被提出，主要通过保留最大的 r 个奇异值不变，最小化最小的 $\min(m, n) - r$ 个奇异值，如公式（2-8）所示：

$$\min_X \|X\|_r \quad \text{s.t.} \quad X_\Omega = M_\Omega \quad (2-8)$$

作者使用两种高效的优化算法对模型进行求解，分别是交替方向乘子法（Alternating Direction Method of Multipliers, ADMM）和加速近端梯度方法（Accelerated Proximal Gradient Line Search, APGL）方法。然而这个模型也有不足之处，比如在实际的应用中，矩阵的秩难以估计。

为了解决这个问题，Gu 等人^[43]通过给予不同奇异值不同的权重来获得更好性能的子空间恢复，提出了加权核范数最小化（Weighted Nuclear Norm Minimization, WNNM），其形式如下：

$$\hat{X} = \operatorname{argmin}_X \|X - Y\|_F^2 + \lambda \|X\|_{w,*} \quad (2-9)$$

其中 λ 是一个大于 0 的正则化参数, $\|X\|_{w,*} = \sum_{i=1}^r w_i \sigma_i(X)$, $w_i = \frac{1}{\sigma_i(X) + \varepsilon}$ 代表了第 i 个奇异值 $\sigma_i(X)$ 的权重。

不同的奇异值拥有不同的权重, 奇异值的值越大权重越大, 奇异值的值越小权重越小。这样做的优势在于, 值越大的奇异值将会受到越小的惩罚, 更易于保持数据的主要方向。最后作者将这个模型成功地应用到图像去噪的应用中, 并取得了良好的效果。

Xie 等人^{[44][45]}提出了加权 Schatten-p 范数最小化约束模型 (Weighted Schatten p-norm Minimization, WSNM), 该模型的形式如下:

$$\hat{X} = \underset{X}{\operatorname{argmin}} \quad \|X - Y\|_F^2 + \lambda \|X\|_{w,S_p}^p \quad (2-10)$$

其中 λ 是一个大于 0 的正则化参数, $\|X\|_{\Omega,S_p}^p = \sum_{i=1}^r w_i \sigma_i(X)^p$, $p \in (0, 1]$,

$$w_i = \frac{1}{\sigma_i(X)^p + \varepsilon}。$$

主成分分析是计算数据分析中的一项基本操作, 它的应用范围广泛, 被广泛地应用在计算机视觉和图像分析领域中。然而, 它的性能和在实际场景中的应用受到对离群或损坏的观测缺乏鲁棒性的限制。为了解决这个问题, RPCA^[16]被提出, RPCA 模型充分考虑到了数据中存在大的离群点这个情况, RPCA 假设数据矩阵 $D \in \mathbb{R}^{m \times n}$ 是由一个低秩矩阵 $A \in \mathbb{R}^{m \times n}$ 和一个稀疏矩阵 $E \in \mathbb{R}^{m \times n}$ 相加得到, 它的模型如下:

$$D = A + E \quad (2-11)$$

为了求解模型中的低秩矩阵 A 和稀疏矩阵 E , 可以将上述问题进行转化, 得到如下的公式:

$$\arg \min \quad \operatorname{rank}(A) + \lambda \|E\|_0, \quad \text{s.t. } D = A + E \quad (2-12)$$

其中 λ 是一个大于 0 的正则化参数, $\|E\|_0$ 代表稀疏矩阵 E 中非零元素的个数。

这个问题是一个非凸的问题, Candès 等人提出了一种基于主成分追踪 (Principal Component Pursuit, PCP) 来解决这个问题, 使用凸松弛的方法将这个问题转化为以下形式:

$$\arg \min \quad \|A\|_* + \lambda \|E\|_1, \quad \text{s.t. } D = A + E \quad (2-13)$$

为了能够更加准确地逼近核范数, Peng 等人^[46]提出了加权低秩矩阵恢复模

型，用于增强低秩约束和稀疏约束，定义如下：

$$\arg \min_{A,E} \|A\|_{\Omega,*} + \lambda \|W \otimes E\|_1, \quad s.t. \quad D = A + E \quad (2-14)$$

其中 λ 是一个大于 0 的正则化参数， $\|A\|_{\Omega,*} = \sum_{i=1}^r w_i \sigma_i(A)$ 表示对奇异值的惩罚。

每个奇异值的权重计算如下 $w_i = \frac{1}{\sigma_i(X) + \varepsilon}$ ，除此之外， $\|W \otimes E\|_1 = \sum_{i=1}^{mn} w_i |e_i|$ 是稀疏矩阵 E 中元素的约束，每个元素的权重计算如下 $w_i = \frac{1}{|e_i| + \varepsilon}$ 。

Schatten L_p 范数^{[42][47]}被成功地应用到的 RPCA 模型中，Dong 等人^[42]提出了双重加权的 RPCA 模型（Dual Reweighted Lp-norm, DWLP），模型如下：

$$\arg \min_{A,E} \|A\|_{\Omega,S_p}^p + \lambda \|W \otimes E\|_q^q, \quad s.t. \quad D = A + E \quad (2-15)$$

其中 λ 是一个大于 0 的正则化参数， $\|A\|_{\Omega,S_p}^p = \sum_{i=1}^r w_i \sigma_i(A)^p$ 表示对奇异值的惩罚。

权重计算方法为 $w_i = \frac{1}{\sigma_i(A)^p + \varepsilon}$ ， $\|W \otimes E\|_q^q = \sum_{i=1}^{mn} w_i |e_i|^q$ 表示稀疏约束，它的权重矩阵计算如下 $w_i = \frac{1}{|e_i|^q + \varepsilon}$ ， $q \in (0,1]$ 。

Liu^[47]等人将使用 Schatten-p 范数和 L_p 范数替代 RPCA 模型中的秩函数和 L_0 范数，得到的模型如下：

$$\arg \min_{A,E} \|A\|_p^p + \lambda \|E\|_{p,p}^p, \quad s.t. \quad D = A + E \quad (2-16)$$

其中 $D \in R^{m \times n}$ ， $\|A\|_p^p$ 为矩阵的 Schatten-p 范数，即 $\|A\|_p = \left(\sum_{i=1}^m \sigma_i^p(A) \right)^{1/p}$ ， $\|E\|_{p,p}$ 为矩阵的 L_p 范数，即 $\|E\|_{p,p} = \left(\sum_{i=1}^m \sum_{j=1}^n |e_{ij}|^p \right)^{1/p}$ 。

2.3 运动目标检测

在 1997 年，Wren 等人^[48]提出了一种单高斯模型来对背景进行建模的方法，这个模型可以较好地处理室内运动目标检测，但是由于室外多模态的背景较为复杂，不能较好地完成室外运动目标检测任务。1999 年，Stauffer 等人^[49]使用高斯混合模型（Gaussian Mixture Model, GMM）对背景进行建模，在这个模型中，视频中的每个像素都是独立地通过多高斯分布来进行处理，不同于单高斯模型一

样, GMM 充分利用了像素的历史信息, 因此 GMM 可以较好地处理多模态环境, 然而其计算的复杂度较高且不能较好地处理场景突然的变化。

Toyama^[50]为了背景建模开发了一种三级模式的模型名为 Wallflower, 分为三个尺度, 分别为像素级、区域级和帧级, 其中在像素级, Wallflower 为每一个像素的背景都建立和维护一个模型, 像素级处理背景像素和前景像素的初步分类问题, 可以处理并适应背景变换问题, 如: 目标位移、舞动的旗帜、晃动的树叶等, 在进行像素级处理时, 每个像素都是被单独处理的, 忽略其邻域像素变化信息。区域级处理考虑到了邻域像素之间的关系, 避免了前景孔径的问题。最后帧级处理可以克服光切换问题。

2000 年, Elgammal 等人^[51]采用了核密度估计 (Kernel Density Estimation, KDE) 来进行建模, 此方法使用高斯核计算每个像素概率密度分布并得到概率分布特性, 再对新的像素进行概率密度计算以此来判断像素是否属于前景或者背景, 此方法运算速度快, 取得了良好的效果。2002 年, Lin 等人^[52]使用一种概率的支持向量机 (Support Vector Machine, SVM) 来实现自动的背景提取, 这个方法依靠概率支持向量机和光流的计算, 先初始化模型, 再使用支持向量机计算每一帧图像的所有像素的概率, 依靠概率来判断其是否为背景像素。

2003 年, Butler 等人^[53]使用 K-means 来构建一组簇用来表示帧中的每一个像素, 根据模型的可能性进行聚类排序, 使用簇来为输入的像素进行匹配, 如果匹配则被认为是背景的一部分, 可以用来较好地处理动态背景和光照变换, 运算速度相对较快。2005 年, Kim 等人^[54]提出了一种实时性的背景与前景分割方法, 该方法为每一个像素在时间域中都建立了一个 CodeBook 结构, 可以较好地抑制动态背景, 但是其消耗内存较多。Messelodi 等人^[55]提出了一种基于卡尔曼滤波的视频背景建模方法, 这个方法能够处理渐变和突然的全局光照变化, 其基本思想是测量全局光照变换, 并将其用来控制卡尔曼滤波。SACON (sample consensus)^[56]是一种基于样本一致性的方法, 对每一个像素都进行一致性检验来判断是否属于背景像素, 其主要分为四个部分, 分别是邻域帧间差分、样本一致性检测、空洞填充和阴影去除处理和背景模型更新。

2009 年, Barnich 和 Droogenbroeck^[57]提出了一种名为视觉背景提取 (Visual Background Extractor, ViBE) 的方法, 该方法假设在一个空间局部区域中每个像素都和其它像素有同样的特性, 使用区域像素来进行背景建模, 该方法的主要优势有占用内存小、运算速度快、不需要复杂的参数设置、抗噪能力较强, 性能优于 GMM、KDE 方法。现在的方法大都基于空间域方法进行建模, 2011 年, Biswas 等人^[58]提出了一种新的基于小波变换的空间背景建模。2012 年, 为了弥补 GMM 运算速度慢的不足, Evangelio 等人^[59]提出了一种分裂的高斯混合模型

(Splitting Gaussian in Mixture Models, SGMM), 其使用两个更新率不同的互补的高斯混合模型, 一个用于精确地检测运动目标, 另一个用于建立背景。相对于 GMM、SGMM 不但提高了检测准确率同时也降低了计算的时间。Hofmann 等人^[60]提出了一种像素自适应分离方法 (Pixel-Based Adaptive Segmenter, PBAS), 该方法结合了 SACON 和 VIBE 两个方法的优势, 并在关键步骤进行了改进, 比如虽然 PBAS 也采用了邻域更新, 但与 VIBE 方法不同, PBAS 的背景模型中的邻域像素是依靠当前像素的邻域像素来实施更新的, 而 VIBE 中邻域的像素是采取当前的像素来更新。PBAS 是基于之前像素梯度的无参数方法, 对缓慢的光照变化具有较好的鲁棒性, 性能优于 SACON 和 VIBE 方法; 然而该方法没有考虑到空间连续性, 因此当处理持久动态背景的情况时性能会下降。2014 年, Lu 等人^[61]提出了一种非参数背景建模方法, 在此方法中动态场景中的每个位置由历史和邻域像素在不同尺度下的信息组成。2014 年, Sedky 等人^[62]提出了一种基于物理的双色反射模型的变化检测技术, 这个方法从相机输出使用图像成像模型来估算颜色描述符, 然后测量前景和背景像素之间的全谱相似度, 最后将前景信息从背景中分割出。2016 年, Wang 等人^[63]提出了一种高精度的半自动分割前景运动目标的方法, 该方法给定有限的用户干预用来提供足够精确的结果。Cao 等人^[64]提出了一种多层背景模型策略, 每个像素都需要经过多个模型的验证从而提高检测的准确性。2017 年, Jiang 等人^[65]使用一些可变权重的样本来迅速适应场景的变化。

2014 年, Liang 和 Kaneko^[66]提出了一种在线的对突然光照变化和动态背景鲁棒的运动检测方法, 其是一种简单直观的基于像素空间相关性的深度学习的统计模型, 对于每个像素都是用一组具有高相关性的像素来支持建模。Gregorio 等人^[67]采用加权神经网络 (Weightless Neural Networks, WNN) 进行动态建模实现运动目标检测, 对每个像素进行单独处理, 不需要像素的邻域信息, 其运算速度较快。2018 年, Babaei 等人^[68]将卷积神经网络 (Convolutional Neural Network, CNN) 运用到了运动目标检测中, 并取得了良好的效果, 该方法的优势是网络参数不需要提前设置, 参数都在训练的过程中学习得到, 可以处理较为复杂的场景且能够满足实时性的要求。

2000 年, Oliver 等人^[69]提出了基于主成分分析的子空间学习 (Subspace Learning Principal Components Analysis, SL-PCA) 方法来对 N 张图片进行建模, 其由平均图像和投影矩阵的前 p 个显著特征向量组成。在 2011 年, Candès 等人提出了鲁棒主成分分析 (Robust Principal Component Analysis, RPCA), RPCA 是一种子空间学习的方法, 并被广泛地应用在运动目标检测中。当理想的状态下, 运动目标运动速度相对较快、场景为静态时, RPCA^[70]可以很好地实现运动目标

检测,在RPCA运动目标检测中,前景信息受到L1范数约束被当成是稀疏噪声,而背景信息受到核范数约束被当作是低秩背景,该模型可以通过非精确增广拉格朗日乘子法进行迭代求解。然而当传统的RPCA进行缓慢运动目标检测时,则检测到的结果中会产生“空洞”,当背景为动态时,比如晃动的树叶、湖面的波浪等,则检测到的前景信息中包含了很多的背景信息。

为此,经过学者们的不断努力,一些改进的RPCA模型被提出^[71-85]。2013年,Zhou等人^[84]在原有的RPCA模型中增加了一个约束,假设前景是有许多连续的小区域组成,其将局部连续性加入了RPCA模型中。Oreifej等人^[77]为了能够更加凸现出前景信息,将原始视频分为三个部分,分别是:背景信息、噪声信息和前景信息,然后光流法进行显著性检测检测出候选的前景信息,并将候选信息加入到RPCA模型中。2015年,Sobral等人^[78]同时将边缘信息和前景的显著信息加入到RPCA的模型中,增加了对前景目标的约束,与文献^[77]不同的是显著信息获取的方式。2017年,Chang等人^[86]提出了一种低秩和加权稀疏分解的运动目标检测方法,其主要思想是对稀疏RPCA分解过程中的稀疏矩阵进行加权,对于视频中运动缓慢的物体或者占据视频帧面积较大的物体给与较小的加权系数,增强前景矩阵的稀疏性。这些方法在处理动态背景和缓慢运动目标检测时均取得了比传统RPCA好的效果。但是他们并没有充分地利用好这些先验信息,比如在对显著性先验知识的使用时,这些方法都是先计算先验信息,然后再将其纳入模型中。

2.4 视频压缩

视频数据占据了互联网数据的大部分^[87],而且视频的应用需求在不断增长,因此需要对视频压缩算法提出了新的要求,不但要有高的压缩率同时要有良好的保真度。

显然视频压缩可以通过图像压缩的方式来实现,这种方式被称为基于图像压缩的方法,其中主要包括JPEG^[88],JPEG2000^[89]和WEBP^[90]。传统图像压缩方法主要是基于离散余弦变换^[88],小波变换^[89]和多级树集合分裂方法。JPEG因为运算时间复杂度较小,现已经被广泛应用在图像压缩领域,它的主要压缩步骤如图2-1所示。

Amiri等人^[91]提出了一种方法来抑制JPEG块状效应。JPEG2000主要由预处理,离散小波变换,量化和信息熵编码组成,其在JPEG基础上提升了30%的压缩率。这些方法依赖于固定的编码器和解码器,且没有运用视频的时间冗余信息,因此压缩能力有效。WebP^[90]是由Google最新提出的基于块预测的方法。Better

Portable Graphics^[92] (BGP) 是基于 High Efficiency Video Coding (HEVC) 的标准上提出的, 其具有比 JPEG, JPEG XR^[93] 和 WebP 的更好的压缩效果。

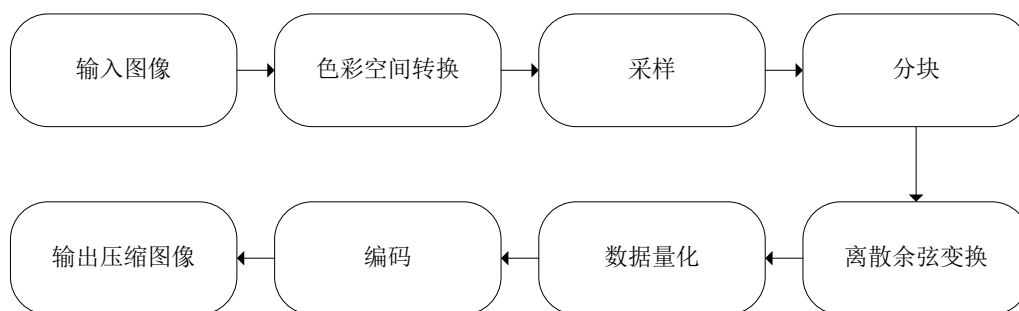


图 2-1 JPEG 图像压缩步骤

近些年, 神经网络^[94-102]被广泛地应用到图像压缩中, 这些方法首先使用大量的图片来训练一个编码器和解码器, 然后再将图像输入到网络中对其进行编码, 最后将编码后的短码和网络保存下来实现视频压缩。这些低维度的短码占据很小的存储空间, 因此实现了视频压缩。

最近, Toderici 等人^[95]提出了一种基于卷积和反卷积的长记忆性循环网络的可变压缩率的图像压缩方法。Toderici 等人^[96]提出了一种全变分有损图像压缩。自编码压缩器^[98]是一种有损压缩方法, 其效果要优于最近提出的基于递归神经网络。基于神经网络的图像压缩方法虽然取得了良好的效果, 但是其因为训练时使用了特定的数据, 因此其具有较弱的泛化能力。基于图像的方法由于忽略了视频帧之间冗余信息, 因此它们的压缩率相对较低。

稀疏表达^[103]和字典学习^[104]被成功地应用在视频压缩中, 受到时空残差图像比原始图像更加稀疏的启示, Song 等人^[105]使用时空残差字典来实现视频压缩。

H.264^[106]是面向图像块处理的视频压缩的行业标准, 也称为 Advanced Video Coding (AVC), 它的基本步骤如图 2-2。

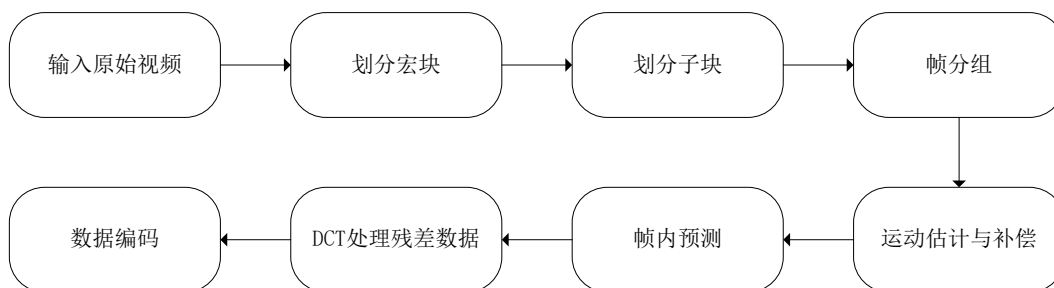


图 2-2 AVC 视频压缩步骤

H.265^[107]是 H.264 的升级版, 也称为 HEVC, 其在同等视频质量下, H.265 可以获得比 H.264 高一倍的压缩率。H.265 使用编码树单元 (CTU) 来进行压缩。

然而这些方法只考虑了邻域块较近的相似块，而忽略了较远的相似块。卷积神经网络被成功地应用到 H.265 中的压缩关键步骤中，如帧内编码^[108]，编码单元^[109]和解码质量^[110]，被用于提高压缩率和压缩效率。最近，下一代视频编码标准 H.266^[111]已被提出，相对于 H.265 其提高了 40% 的编码效率，但同时编码的复杂度也大幅度提高。

张量分解^[112-114]被广泛使用在数据压缩中。Friedland 等人^[115]提出了一种统一框架张量压缩用于压缩高阶张量。Du 等人^[116]提出了一种多路张量投影算法，作为传统的主成分分析法的扩展，用于高光谱数据压缩和重建。

基于神经网络的视频压缩方法相对较新，是近几年刚提出的方法，其因为考虑到了帧间冗余信息，因此相对于基于神经网络的图像压缩方法，其具有较高的压缩率。Chen^[117]展示了一种 DeepCoder 方法，这种方法使用深度神经网络预测图像块的运动信息。Wu 等人^[118]提出了一种依靠帧插值的方法，这种方法获得了和 H.264 相近的性能。这些方法主要都是基于块的方法，因此它们具有与现有方法相近的性能。相反，在 Han 等人^[119]的工作中，提出了一种基于序列变分自编码器的压缩方案，该方法将原始视频转换为低维信息表示同时进行时间概率模型熵编码。Kim 等人^[120]基于软边缘检测提出了对抗生成网络的视频压缩框架。与此同时，Li 等人提出了基于视频预测的视频压缩方法^[121]。虽然基于深度学习的视频压缩方法^[118-120]取得了显著的效果，但是他们需要特定的数据进行训练，因此泛化能力较弱。

2.5 总结

本章介绍了子空间分割的前沿研究热点、视频压缩的相关工作和运动目标检测的研究内容，在接下来的章节中，本文将在第三章中介绍子空间学习基础算法研究，主要包括基于随机探测的子空间分割算法和基于截断加权 RPCA 的子空间恢复算法，本文将在第四章介绍子空间学习在视频处理中的应用研究，包括基于 RPCA 的显著和分割约束的子空间分割算法及其在运动目标检测中的应用和基于 3 阶张量自编码器的子空间压缩方法及其在视频压缩中的应用。在第五章中，我们对本文进行了总结和展望。

第三章 子空间学习基础算法研究

在本章中，将介绍我们在子空间学习基础算法研究中所做的工作，分为两个部分，第一部分为基于随机探测的快速子空间分割算法，第二部分为基于截断加权 RPCA 的子空间恢复算法。

3.1 基于随机探测的快速子空间分割算法

3.1.1 引言

主成分分析就是一个传统的子空间方法，其通过从高维数据中提取出低维的主要信息，其在模式识别^[7]、数据压缩等领域有着广泛的使用^[122]。然而在实际应用中，数据点可能来自多个子空间，那么如何将每个子空间聚类到多个的子空间中，找到属于其自己的低维子空间，这个问题就是子空间分割^[38]。在子空间中，并不是数据点距离近，它们就是属于同一个子空间，如图 3-1^[34]中：

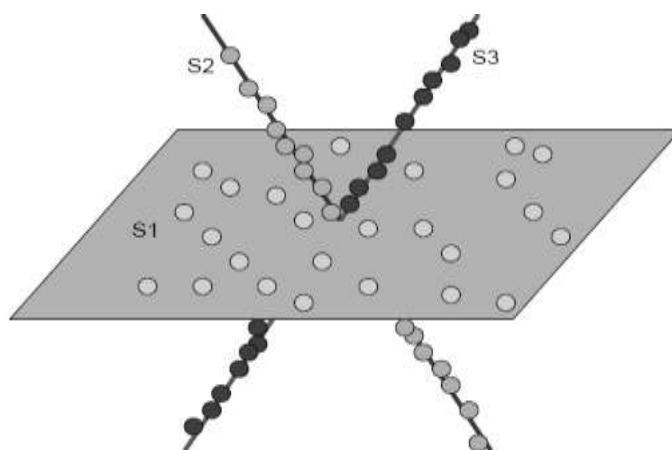


图 3 - 1 子空间：其中 S1、S2 和 S3 属于三个不同的子空间

子空间分割就是将给定的数据点划分到不同的簇中，每个簇中的点属于一个子空间。现有的方法如 SSC 和 LRR 可以有效地进行子空间分割，具有较高的准确性，但是时间复杂度较高，因为它们都要使用谱聚类算法来处理相似性矩阵，谱聚类的时间复杂度非常高，假设相似性矩阵的大小为 $n \times n$ ，那么使用谱聚类^[123]来处理的时间复杂度为 $O(n^3)$ ，因此基于谱聚类的方法不能一次有效地处理大量的数据点。

为此, 本文提出了一种基于迭代的随机探测的子空间分割方法 **RANSP**。在每一次迭代中, 本文方法只分割出一个子空间, 因此本文方法可以处理较多的数据点。**RANSP** 每次随机选择一个点作为种子点, 然后使用岭回归计算种子点与其它点的相似性, 通过相似性值判断得到一个子空间, 其中我们使用 **WoodBury** 来优化岭回归的求解, 使得 **RANSP** 的时间复杂度为线性的, 其时间复杂为 $O(n)$ 。实验结果表明, 本文方法可以有效地分割包含较多点的数据库。

3.1.2 RANSP 模型

在本节中, 我们将详细介绍本文的方法 **RANSP**, 使用 **RANSP** 对一组具有 n 个点且每个点的维度为 m 的数据矩阵 $X \in R^{m \times n}$ 进行聚类, 类别数为 k 。在每一次的随机探测中 **RANSP** 都会得到一个子空间, 那么 k 个子空间只需要 $k-1$ 次就可以全部得到。在正常情况下, 数据分为两种情况: 有噪声和无噪声。接下来将介绍 **RANSP** 处理这两种情况下的数据。

3.1.2.1 数据中不包含噪声

为了便于理解本文的方法, 先从理想的情况来介绍本文的方法。在理想的情况下数据中不包含任何噪声。假设 $y \in R^m$ 是从数据矩阵 X 中随机选出的一个种子点, 为了发现 X 中所有与 y 属于同一个子空间的点, 则可以将此问题转化为 $L2$ 最小化的问题:

$$\min_w \|w\|_2, \text{ s.t. } y = Xw \quad (3-1)$$

其中 $\|\cdot\|_2$ 为一个向量的 $L2$ 范数, 这个问题具有一个闭式解:

$$\hat{w} = X^+ y \quad (3-2)$$

其中 $(\cdot)^+$ 表示一个矩阵的伪逆。

如果数据点中不包含噪声且每个子空间时相互独立的, 那么可以通过 \hat{w} 来识别与 y 同一个子空间的点, 如定理 3-1。

定理 3 - 1^[28]: 令 $X = [x_1, x_2, \dots, x_n]$, 假设 $\hat{w} = [w_1, w_2, \dots, w_n]$ 是问题 3-1 的最优解, 当每个子空间时相互独立时, X 不是满秩且数据点是从每个子空间随机选择的, 那么当 x_i 与 y 不属于同一个子空间时 w_i 为 0, 那么当 x_i 与 y 属于同一个子空间时 w_i 不为 0。

为了方便理解, 我们称 \hat{w} 为相似性向量, $|w_1|, |w_2|, \dots, |w_n|$ 被称为相似性距离。如图 3-2 所示, 数据点矩阵 X 包含了 100 个点, 其中每个点的维度为 100, 包含了 5 个独立的子空间, 每个子空间的秩为 4。第 1 个点到第 20 个点属于同一个

子空间，第 21 个点到第 40 个点属于同一个子空间，第 41 个点到第 60 个点属于同一个子空间，第 61 个点到第 80 个点属于同一个子空间，第 81 个点到第 100 个点属于同一个子空间，第 25 个点为随机选择的种子点。

可以看出与种子点同一个子空间的点的相似性距离都大于 0，而和种子点不属于同一个子空间点的相似性距离都为 0。因此，当数据点中不包含噪声且每个子空间相互独立时，我们可以通过相似性距离是否大于 0 来判断点与点是否属于同一个子空间。本文无噪声数据子空间聚类步骤具体见算法 3-1。

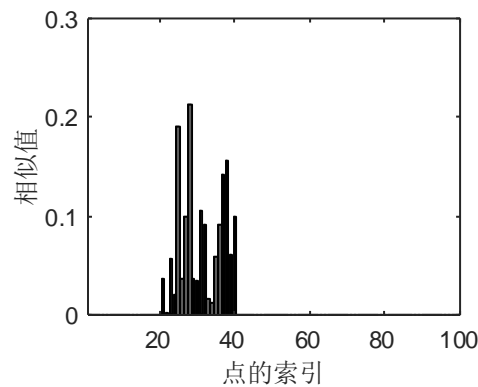


图 3 - 2 无噪声情况下相似向量 \hat{w}

算法 3 - 1 无噪声数据子空间聚类

输入：数据点集 X 和子空间数量 k

输出：子空间分割结果

1. 标记所有点都为“未分配的”
2. For $I = 1$ to $k-1$
 - 随机选择一个未分配的点作为种子点
 - 计算相似性向量
 - 将相似性向量中值大于 0 对应的点和种子点归为第 i 个子空间
- Next i
3. 将所有未分配的点归为第 k 个子空间

3.1.2.2 数据中包含噪声

尽管上述情况下，方法被证明是有效的，但是在实际的情况中，数据中往往包含噪声。为了在噪声情况下得到有效的子空间分割，我们需要对之前的方法进行修正，修正后的步骤具体如下：

为了处理噪声数据，首先 $L2$ 最小化问题需要被岭回归替代：

$$\min_w \|y - Xw\|_F^2 + \lambda \|w\|_2^2 \quad (3-3)$$

其中 λ 是个大于 0 的参数，上式包含了一个闭式解：

$$\hat{w} = (X^T X + \lambda I)^{-1} X^T y \quad (3-4)$$

其中 $(\cdot)^{-1}$ 是非奇异矩阵的逆， I 为单位矩阵，我们再一次定义相似性向量 $\hat{w} = [w_1, w_2, \dots, w_n]^T$ 。

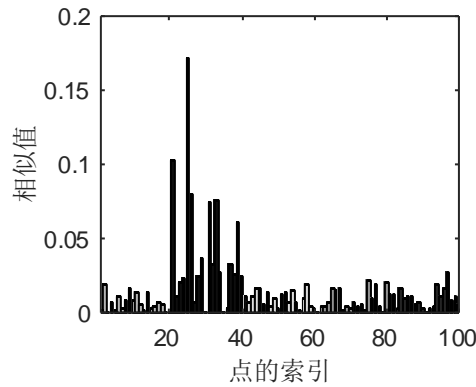


图 3 - 3 噪声情况下的相似性向量

图 3-3 给出了噪声情况下的相似性向量，其中数据矩阵 $X = X_0 + N$ ， X_0 为包含了 100 个数据点，属于 5 个独立的子空间，每个子空间的秩为 4， N 为 0 均值标准差为 0.15 的高斯噪声。第 21 个点第 40 个点属于同一个子空间，第 25 个点为随机选择的种子点。从中可以看出，通过相似性值仍然可以大致判断其它点与种子点是否属于同一个子空间。然而不能够通过之前的相似性值大于 0 来判断是否属于同一个子空间，很难通过相似性值来找到所有与种子点属于同一个子空间的点。但我们可以找到一个子空间所有点的子集，为了得到子集，我们使用如下的阈值：

$$th = \mu \|\hat{w}\|_\infty \quad (3-5)$$

其中 μ 是一个大于 0 的参数， $\|\cdot\|_\infty$ 为一个向量中的所有元素绝对值中的最大值。

根据经验我们设置 μ 为 0.4 左右。对于某个点，如果相似性值大于阈值 th 则认为它是和种子点属于同一个子空间，尽管还有一些和种子点属于同一个子空间的点没有被选择出，但是通过阈值我们可以保证得到的子空间的准确性。

图 3-4 中数据矩阵 $X = X_0 + N$ ，其中 X_0 为包含了 100 个数据点，属于 5 个独立的子空间，每个子空间的秩为 4， N 为 0 均值标准差为 0.15 的高斯噪声。第 21 个点第 40 个点属于同一个子空间。

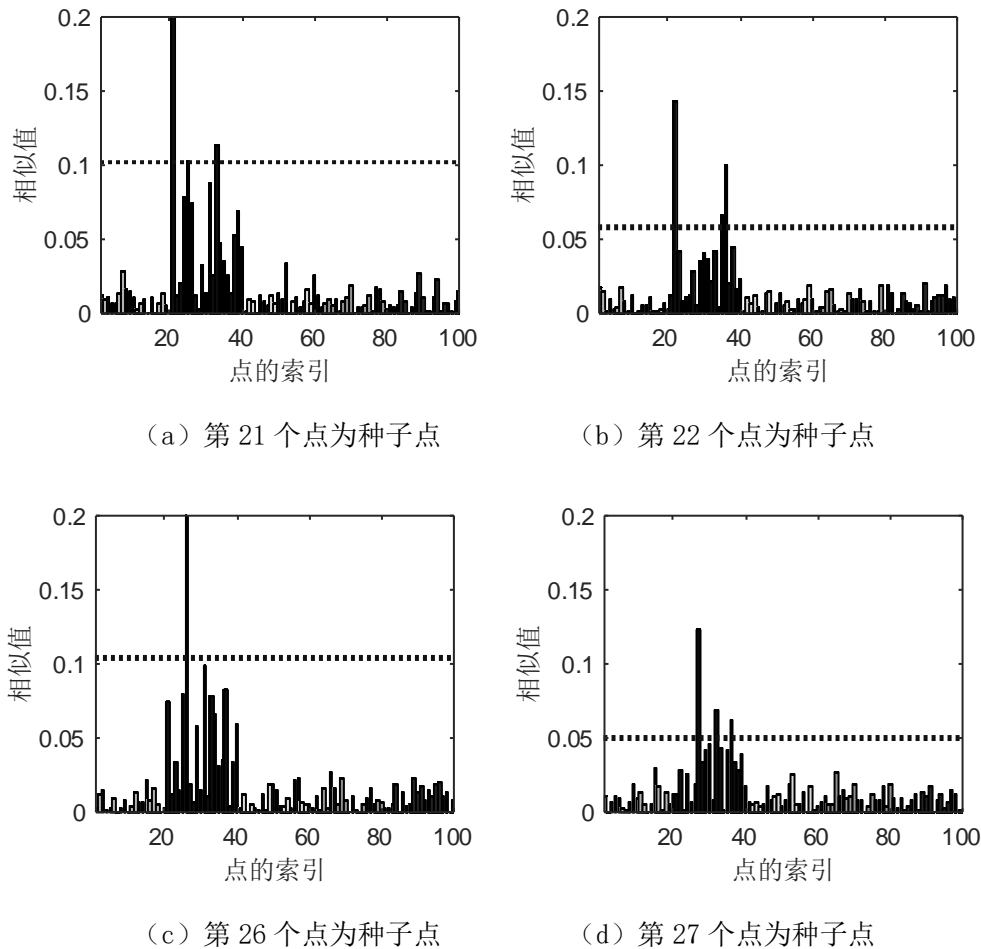


图 3-4 种子点的影响

从图 3-4 中可以看出，相似性向量的质量取决于种子点的选择，当种子点选择的“好”时，则可以发现较多与种子点属于同一个子空间的点，当种子点选择的“坏”时，则可以找出很少与种子点属于同一个子空间的点。图 3-4 (c) 是一种极端的情况，只找到了 0 个和它同一个子空间的点。

为了解决这个问题，我们提出了一种随机多重探测的机制，它是由如下的重复迭代的步骤组成：第一步：从所有的数据点中随机选择一个点作为种子点；第二步：计算相似性向量，并计算阈值；第三步：根据阈值得到与种子点属于同一个子空间的点，记录这些点的编号。重复上面的步骤可以得到有效的种子点，在本文中。我们设置了固定的随机探测的次数 f ， f 值越大，种子点越好，在实际情况中 $f=100$ 已经可以取得了很好的效果。

经过算法 3-2 处理后，我们可以得到每个子空间的子集，如图 3-5 所示，从中可以看出还有部分点不属于任何的子空间。

为此我们使用了后续处理步骤，主要分为以下步骤：使用主成分分析得到每个子空间的基，然后再将不属于任何子空间的点分配到距离其最近的子空间，如图 3-6。

算法 3 - 2 随机探测

输入：数据点集 X ，阈值参数 μ 和随机探测次数 f

输出：属于一个子空间的点集

1. For $i=1$ to f

 随机选择一个点作为种子点

 计算相似性向量

 计算阈值

 将相似性向量中大于阈值对应的点作为一个子空间并记录该点集

Next i

2. 选择点集中包含最多数点作为一个子空间

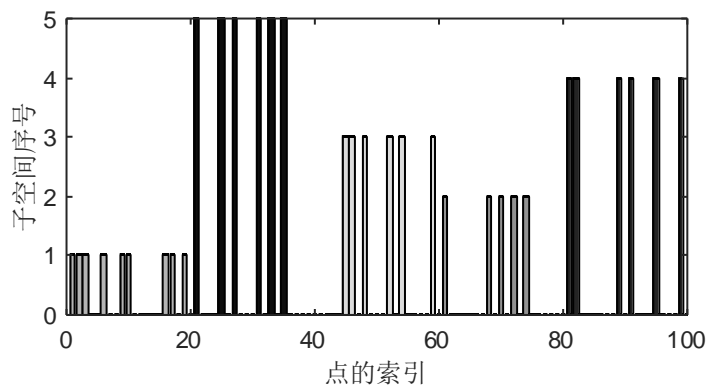


图 3 - 5 正类点

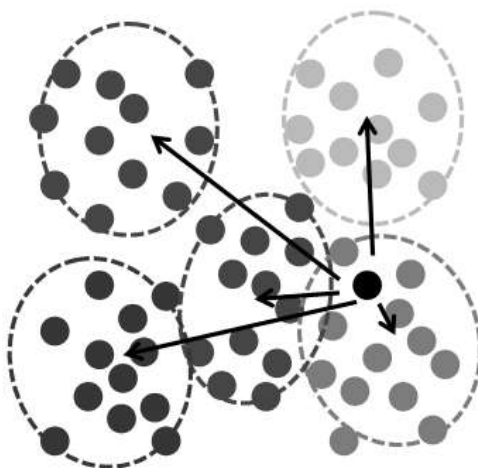


图 3 - 6 寻找最近子空间

点到子空间的距离使用如下的公式进行计算：

$$\|z - UU^T z\|_2^2 \quad (3-6)$$

其中 $U \in R^{m \times d}$ 是子空间的基矩阵。

算法 3 - 3 RANSP 子空间分割

输入：数据点集 X ，子空间类别数 k ，正则化参数 λ ，阈值参数 μ ，随机探测次数 f ，每个子空间的秩 d

输出：子空间分割结果

1. 标记所有点都为未分配的点
2. For $i=1$ to k
 - 使用算法 3 - 2 得到一个子空间并记录该子空间 X_i
 - 标记 X_i 为已分配的点
 - 通过不完全奇异值分解得到第 i 子空间的基矩阵
- Next i
3. For $i=1$ to 未分配的点
 - 计算该点到每个子空间的距离
 - 将该点分配到最近的子空间
- Next i

3.1.2.3 计算复杂度及优化

在本文方法 RANSP 中，时间复杂度最高的地方是岭回归的求解，时间复杂度为 $O(n^3)$ ，当数据量很大时，则不能高效地进行处理。大量的方法被学者们提出用来减少岭回归的时间复杂度，如梯度下降^[124]，拟牛顿法^[125]，随机梯度下降^[126-128]，主成分分析^{[124][129][130]}。但是以上的方法还需要消耗大量时间来进行优化，在本文工作中，RANSP 被用来处理数据点的维度远小于数据点的数量的情况。根据 Woodbury Matrix Identify^[131] 公式我们可以得到：

$$(X^T X + \lambda I)^{-1} X^T \approx X^T (X X^T + \lambda I)^{-1} \quad (3-7)$$

装载矩阵 $L \cong (X^T X + \lambda I)^{-1} X^T$ 的时间复杂度为 $O(m^2 + m^3)$ 。此方法的优势是只需要计算一次装载矩阵，为了获得随机点 y 的相似性矩阵，我们只需要计算 Ly ，只需要 $O(mn)$ 的时间。

在随机探测步骤中，我们需要计算 Ly 的相似性向量 f 次。因此寻找 k 个子空间需要的时间复杂度为 $O(kfmn)$ 。假设得到的每个子空间点的个数为 n_1 ，通常 n_1 远小于 n/k 且其是一个很小的值。为了得到 k 个子空间的基，我们需要计算每个子空间部分 d -SVD，每个子空间的大小为 $m \times n_1$ 的矩阵，其时间复杂度为 $O(mdn_1)$ ，得到 k 个子空间的基的时间复杂度为 $O(kmdn_1)$ ，其最大为 O

(mnd)。不属于任何子空间的点的数量为 $n-kn1$ 。因此后续处理步骤的时间复杂度为 $O(m-kn1)md$ ，其小于 $O(mnd)$ 且远小于 $O(nm^2)$ ，因为 d 小于 m 。所以在处理有噪声的数据点时，RANSP 的时间复杂度为 $O(nm^2+m^3+kfmn)$ 。

3.1.3 子空间分割实验

本文所有实验都是在一个拥有志强 E5-2620 2.1GHZ 的 CPU 和 128GB 内存的服务器上运行的，操作系统为 Ubuntu 16.04，编程环境为 MATLAB R2016b。

3.1.3.1 实验数据

第一个数据库是使用文献^[22]中的方法生成了一个包含 100 万个数据点，每个数据点的维度为 200 的合成数据库。具体地，我们建立了 10 个独立的子空间，每个子空间的基矩阵 $\{U_i\}_{i=1}^{10}$ 是使用 $U_{i+1} = TU_i$ ， $1 \leq i \leq 9$ ，其中 T 为一个随机的旋转， U_i 是一个 200×5 的随机正交矩阵，每个子空间的秩为 5。我们通过 $X_i = U_i Q_i$ 从每个子空间采样了 10^4 个点， $1 \leq i \leq 10$ ，其中 Q_i 是一个 5×100000 大小的矩阵。我们在数据中加入了 $N(0, \rho)$ 的噪声，在实验中设置了不同的噪声等级 ρ 用来测试算法的抗噪性能。

第二个是 Fashion-MNIST^[18] 的数据库，此数据库拥有与 MNIST 数据库相同大小的尺寸和格式。不同于 MNIST 数据库，Fashion-MNIST 数据库是一个包含了衣服鞋帽包等的数据库，如图 3-7，每三行属于一个类别。这个数据库共有 10 个类和 28×28 大小的 60000 张图片。为了提高效率，本文使用主成分分析将数据降到了 100 维。我们从此数据库中分别挑选了前 6 个类、前 8 个类和所有的 10 个类用作实验。

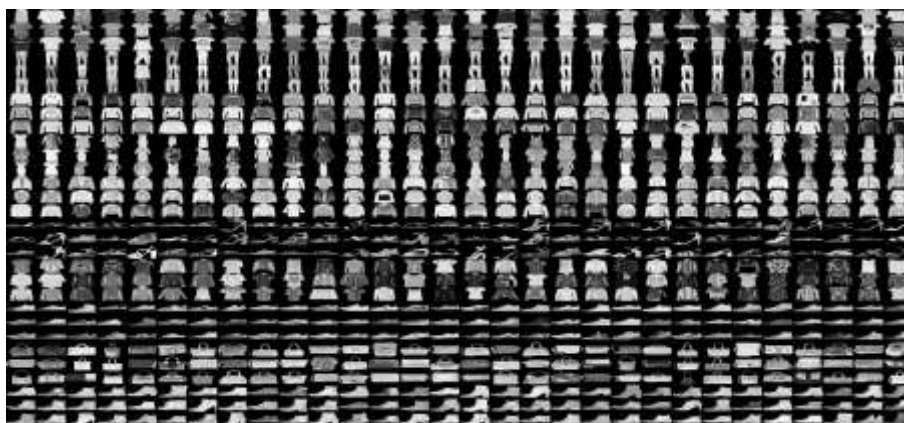


图 3 - 7 Fashion-MNIST 数据库部分图片^[18] (图片链接:
<https://github.com/zalandoresearch/fashion-mnist>)

第三个数据库是 CMU PIE^[17] 人脸数据库，如图 3-8 所示，其包含了 68 个人脸照片，一共 11554 张照片。我们从中分别选取了前 50 个类，前 60 个类和所有

68 个类用于实验，同样地我们使用主成分分析将数据维度降到了 100。



图 3 - 8 PIE 数据库部分图片^[17]

具体的实验数据信息在如下的表中：

表 3 - 1 三个数据库的具体信息

数据库	类别数	维度	点数
合成数据库	10	200	1000000
	6	100	36000
Fashion-MNIST	8	100	48000
	10	100	60000
	50	100	8494
PIE	60	100	10194
	68	100	11554

3.1.3.2 实验设置

算法的运行时间、子空间的准确性和归一化互信息（Normalized Mutual Information， NMI）^[132]被用来评价每个算法的效率和有效性。分割的准确性是用来衡量每个点被分配正确的度量。

NMI 的范围是 0-1，数值越大效果越好，其被用来衡量分类结果与标准结果的差别，如果没有差别则结果为 1，如果差别非常大则结果接近于 0，其形式具体如下：

$$U(X,Y)=2\frac{I(X;Y)}{H(X)+H(Y)} \tag{3-8}$$

$$I(X;Y)=\sum_{y\in Y}\sum_{x\in X}p(x,y)\log\left(\frac{p(x,y)}{p(x)p(y)}\right) \quad (3-9)$$

$$H(X)=-\sum_{i=1}^np(x_i)\log_2p(x_i) \quad (3-10)$$

$$H(Y)=-\sum_{i=1}^np(y_i)\log_2p(y_i) \quad (3-11)$$

实验中参数的设置如表 3-2 和表 3-3, 其中表 3-2 为 PIE 和 Fshion-MNIST 数据的参数设置, 表 3-3 是合成数据库参数设置。

表 3 - 2 PIE 和 Fshion-MNIST 数据的参数设置

方法	PIE			Fasnion-MNIST		
	50 类	60 类	68 类	6 类	8 类	10 类
RANSP		$\lambda = 0.01$			$\lambda = 1$	
		$u = 0.5$			$U = 0.46$	
		$d = 35$			$d = 20$	
		$f = 100$			$f = 100$	
K-Subspace		$\text{maxLoop} = 1000$			$\text{maxLoop} = 1000$	
		$d = 35$			$d = 20$	
LSR1		$\lambda = 0.004$			$\lambda = 0.001$	
LSR2		$\lambda = 0.004$			$\lambda = 0.001$	
LRR		$\lambda = 0.01$			$\lambda = 0.002$	
SSC		$\lambda = 10$			$\lambda = 3$	
SSSC		$\text{landmark} = 1000$			$\text{landmark} = 1000$	
OLRSC		$\text{Epochs} = 2$			$\text{Epochs} = 2$	

表 3 - 3 合成数据库参数设置

方法	合成数据库
RANSP	$\lambda = 100, u = 0.45, d = 5, f = 100$
K-Subspace	$\text{maxLoop} = 3000, d = 5$

为了验证对比 RANSP 方法的有效性, 本文选取了 LRR^[22], SSC^[15], SSSC^[33], OLRSC^[37], LSR1^[23], LSR2^[23]和 K-subspace^[14]作为比对方法。LRR、SSC、SSSC、OLRSC、LSR1 和 LSR2 是基于谱聚类的子空间分割方法, 其中 SSSC 和 OLRSC 是快速子空间聚类方法。

3.1.3.3 结果与分析

在合成的数据库中, 通过改变噪声级别来进行 RANSP 和 K-Subspace 算法的

对比。在这个数据库中，我们没有将 LRR, LSR1, LSR2, SSC, SSSC 和 OLRSC 方法纳入比对，因为处理的数据非常多，这些方法时间复杂度非常高。虽然相对于 SSC, SSSC 的速度已经提高了很多，但是其仍然需要进行谱聚类操作，时间复杂度仍然很大，K-subspace 方法是目前速度最快的子空间分割方法。

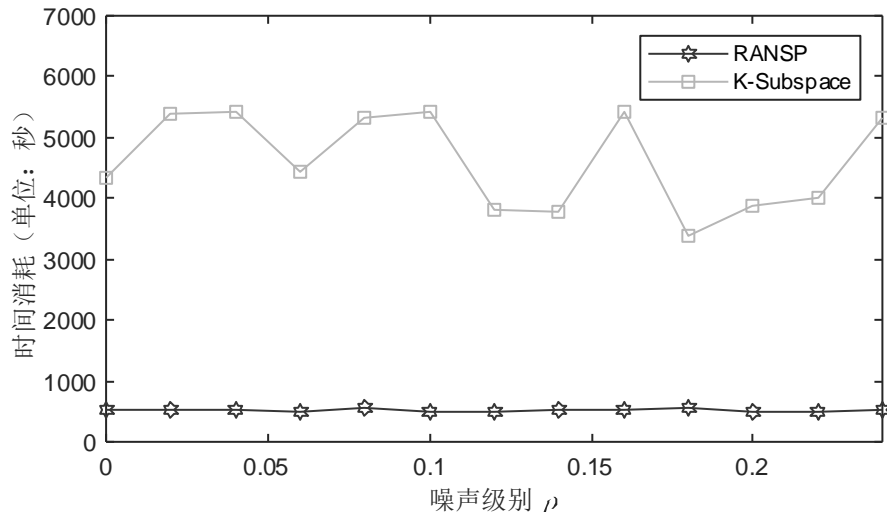


图 3 - 9 合成数据库时间消耗

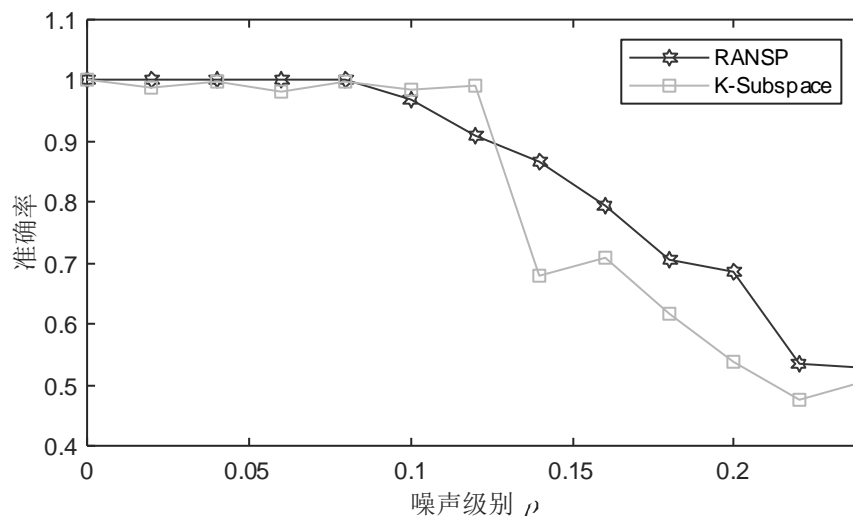


图 3 - 10 合成数据库准确率

图 3-9 给出了本文方法 RANSP 和 K-subspace 方法在合成数据库上的实验时间信息，实验中加入了不同级别的噪声，从图 3-9 中可以看出，本文的方法在处理 100W 个数据点时所需要的时间仅为 500 秒左右，而目前速度最快的 K-subspace 方法大概需要 5000 秒左右，本文的 RANSP 方法仅为它的十分之一左右。

图 3-10 给出了本文方法 RANSP 和 K-subspace 方法在合成数据库上的实验分割结果准确度信息，从中可以看出当噪声级别在 0-0.1 这个区间时，RANSP

方法和 K-subspace 的准确度很接近，都接近于 1，在 0.1 到 0.13 这个区间时，K-subspace 的效果略好与 RANSP，当噪声级别大于 0.13 时，则 RANSP 的准确率则高于 K-Subspace。

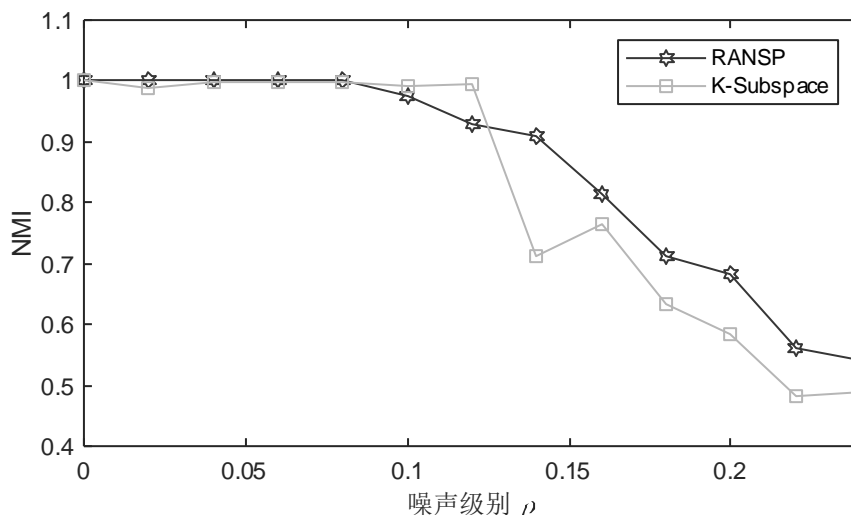


图 3 - 11 合成数据库 NMI

图 3-11 给出了本文方法 RANSP 和 K-subspace 方法在合成数据库上的实验分割结果 NMI 信息，从中可以看出当噪声级别在 0-0.1 这个区间时，RANSP 方法和 K-subspace 的准确度很接近，都接近于 1，在 0.1 到 0.13 这个区间时，K-subspace 的效果略好与 RANSP，当噪声级别大于 0.13 时，则 RANSP 的准确率则高于 K-Subspace。

从结果中可以看出在效率和性能上，RANSP 优于 K-Subspace。因为 K-Subspace 要处理大量的数据点，需要大量的时间来进行模型迭代更新，因此其运行效率低于 RANSP。

表 3 - 4 PIE 和 Fashion-MNIST 数据库时间消耗（单位：秒）

Method	PIE			Fashion-MNIST		
	50 类	60 类	68 类	6 类	8 类	10 类
RANSP	3.2	4.3	6.0	12.8	16.9	21.0
K-Subspace	38.6	42.3	62.1	50.1	98.8	154.6
LSR1	119.2	198.6	276.7	4145.2	5541.5	8472.1
LSR2	122.0	179.6	258.9	3552.9	5054.2	8861.7
LRR	146.7	233.4	292.1	3845.8	5342.0	7248.6
SSC	1297.5	1686.3	2315.8	29194.2	56421.5	84653.9
SSSC	42.3	48.1	52.0	185.7	197.5	211.2
OLRSC	76.1	114.8	158.6	2620.4	3865.6	5827.5

从表 3-4 到表 3-6 可以看出在 PIE 和 Fashion-MNIST 数据库中, LSR1 和 LSR2 需要运行谱聚类对相似性矩阵进行分割, 因此其具有比 RANSP 较高的时间复杂度。为了得到 $n \times n$ 的参数矩阵, LRR^[133] 没有直接在 $n \times n$ 的矩阵上计算 SVDs, 而是计算了 $m \times n$ 的矩阵, $m < n$, 因此 LRR 的时间复杂度为 $O(nlm^2) + O(lm^3)$, 其中 l 为迭代次数。通常情况下, LRR 需要至少上百次的迭代才能收敛, 因此其运行效率低于 RANSP。OLRSC 时间复杂度低于 LRR, 但是它还是需要计算谱聚类。在运行时间, 聚类准确性和 NMI 这三个指标中, RANSP 方法取得了最好的性能。

表 3 - 5 PIE 和 Fashion-MNIST 数据库准确率

Method	PIE			Fashion-MNIST		
	50 类	60 类	68 类	6 类	8 类	10 类
RANSP	0.460	0.452	0.418	0.673	0.543	0.501
K-Subspace	0.276	0.273	0.272	0.634	0.483	0.549
LSR1	0.458	0.436	0.424	0.639	0.547	0.522
LSR2	0.470	0.431	0.421	0.654	0.527	0.489
LRR	0.437	0.412	0.393	0.652	0.564	0.553
SSC	0.421	0.416	0.409	0.658	0.585	0.572
SSSC	0.340	0.328	0.311	0.577	0.573	0.569
OLRSC	0.326	0.294	0.281	0.604	0.438	0.405

表 3 - 6 PIE 和 Fashion-MNIST 数据库 NMI

Method	PIE			Fashion-MNIST		
	50 类	60 类	68 类	6 类	8 类	10 类
RANSP	0.562	0.559	0.533	0.634	0.554	0.529
K-Subspace	0.461	0.475	0.480	0.615	0.545	0.559
LSR1	0.581	0.568	0.574	0.611	0.586	0.557
LSR2	0.584	0.567	0.556	0.624	0.554	0.539
LRR	0.556	0.545	0.529	0.627	0.520	0.507
SSC	0.502	0.478	0.462	0.616	0.592	0.581
SSSC	0.461	0.445	0.447	0.603	0.598	0.574
OLRSC	0.384	0.357	0.335	0.626	0.466	0.427

超参数敏感性分析使用的数据为 PIE 前 50 个类进行测试。

图 3-12 给出了子空间维度 d 的大小对分割准确率和 NMI 值的影响。从中可以看出, d 从 5 开始递增到 35 左右时, 准确率和 NMI 值都在同时提高, 当 d 的值在 35-50 时, 准确率和 NMI 值趋向于稳定。

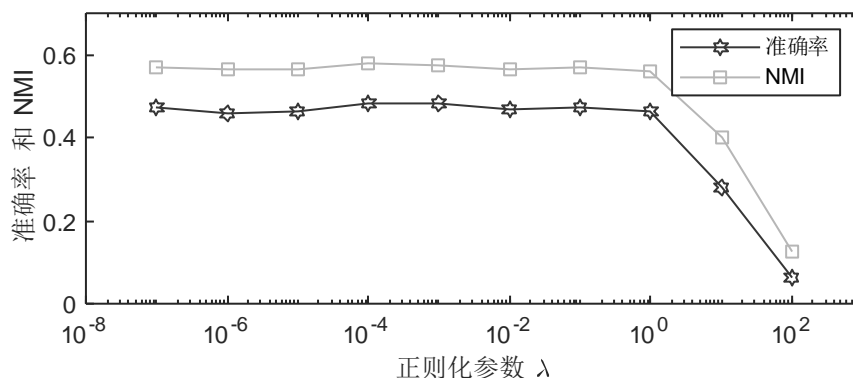


图 3-12 子空间维度敏感性分析 (其中 $\lambda = 0.0001$, $\mu = 0.46$, $f = 100$)

图 3-13 给出了子空间维度 λ 的大小对分割准确率和 NMI 值的影响,从中可以看出当 λ 小于 1 时,准确率和 NMI 值分别在 0.57 和 0.49 左右波动,当 λ 大于 1 时准确率和 NMI 值迅速下降。

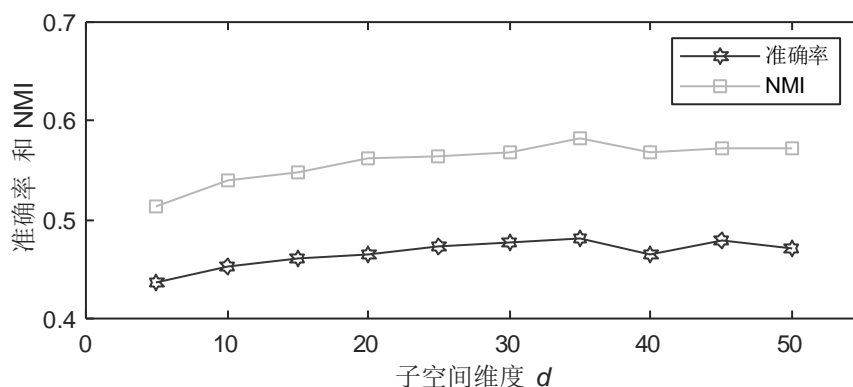


图 3-13 正则化参数 λ 敏感性分析 (其中 $d = 35$, $\mu = 0.46$, $f = 100$)

图 3-14 给出了阈值参数 μ 的大小对分割准确率和 NMI 值的影响,当 μ 从 0.2 到 0.45 时,准确率和 NMI 值都在不断提高,当 μ 大于 0.45 时,准确率和 NMI 开始下降。

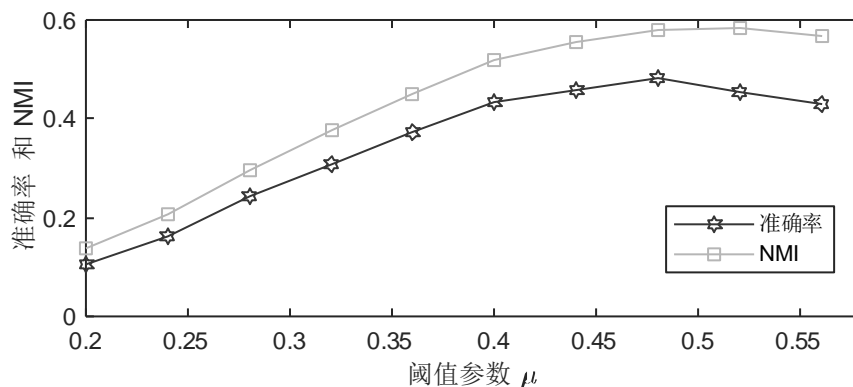


图 3-14 阈值参数 μ 敏感性分析 (其中 $d = 35$, $\lambda = 0.0001$, $f = 100$)

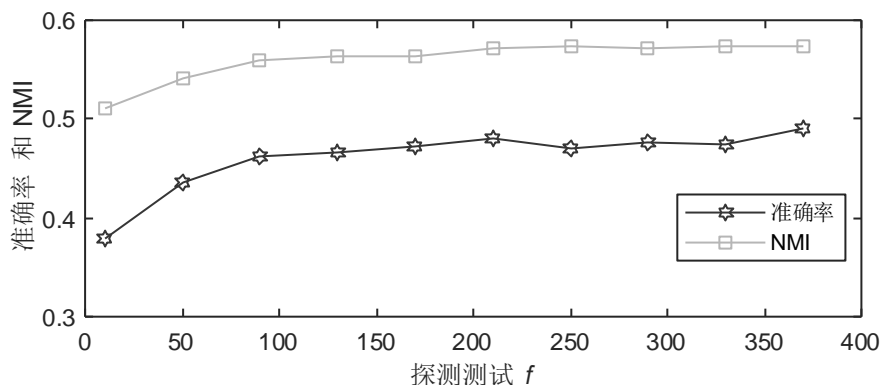


图 3-15 探测次数 f 敏感性分析 (其中 $\lambda = 0.0001$, $d = 35$, $\mu = 0.46$)

图 3-15 给出了探测次数 f 对分割准确率和 NMI 值的影响, 当 f 从 20 到 100 时, 准确率和 NMI 值都在不断提高, 当 f 大于 100 时, 准确率和 NMI 趋于稳定。

从超参数敏感测试中可以看出, 当 λ , μ , f , d 处于较大范围中, RANSP 可以取得良好的效果, 参数容易设置为实际应用提供了便捷。相对来说参数 μ 需要仔细设置, 参数过小, 则得不到较多的与种子点同一个子空间的点, 无法得到子空间的基, 参数过大, 则会将不属于同一个子空间的点归为同一个子空间, 这样便得到了错误的基。

3.2 基于截断加权 RPCA 的子空间恢复算法

3.2.1 引言

主成分分析 PCA 在进行投影时保留着数据中最主要的结构和元素, 可以实现数据的降维, 寻找高维数据中最主要的信息。但是 PCA 的前提是数据的噪声是高斯噪声, 当数据中存在大的离群点时, PCA 则会受到大离群点的影响, 不能有效的进行数据从高维到低维的投影, 因此难以发现数据中的低维子空间。

为了解决这个问题, RPCA^[16]被提出, RPCA 模型充分考虑到了数据中存在大的离群点, RPCA 假设数据矩阵 D 是由一个低秩矩阵 A 和一个稀疏矩阵 E 相加得到, 它的模型如下:

$$D = A + E \quad (3-12)$$

为了求解模型中的矩阵 A 和矩阵 E , 将上述问题进行转化, 得到如下形式:

$$\arg \min \quad \text{rank}(A) + \lambda \|E\|_0 \quad (3-13)$$

其中 λ 是一个大于 0 的正则化参数, $\|E\|_0$ 代表稀疏矩阵 E 中非零元素的个数。

为了求解这个非凸问题, Candès 等人^[16]使用基于主成分追踪 (Principal

Component Pursuit, PCP) 来解决这个问题, 使用凸松弛的方法将这个问题转化为以下形式:

$$\arg \min \|A\|_* + \lambda \|E\|_1 \quad (3-14)$$

其中, $\|A\|_*$ 为低秩矩阵 A 的核范数, 代表奇异值的和, 即 $\|A\|_* = \sum_{i=1} \sigma_i(A)$ 。 $\|E\|_1$ 为稀疏矩阵的 1 范数, 表示矩阵 E 的列范数的和, 即:

$$\|E\|_1 = \max_j \left(\sum_{i=1}^m |a_{ij}| \right) (j = 1, 2, \dots, n) \quad (3-15)$$

这个问题 $\arg \min \|A\|_* + \lambda \|E\|_1$ 可以使用非精确增广拉格朗日乘子法来求解^[16], 此问题对应的拉格朗日形式如下:

$$L(A, E, Y, \mu) = \|A\|_* + \lambda \|E\|_1 + \langle Y, D - A - E \rangle + \frac{\mu}{2} \|D - A - E\|_F^2 \quad (3-16)$$

交替更新求解 A 和 E 。更新 A 时:

$$\arg \min_A \|A\|_* + \frac{\mu}{2} \|D - A - E + \mu^{-1} Y\|_F^2 = D_{\mu^{-1}}(D - E + \mu^{-1} Y) \quad (3-17)$$

更新 E 时:

$$\arg \min_E \|E\|_1 + \frac{\mu}{2} \|D - A - E + \mu^{-1} Y\|_F^2 = S_{\frac{\lambda}{\mu}}(D - A + \mu^{-1} Y) \quad (3-18)$$

交替更新直到收敛, 便可以得到低秩矩阵 A 和稀疏矩阵 E 。

$$\min_{A, E} \|A\|_* + \lambda \|E\|_1, s.t. D = A + E \quad (3-19)$$

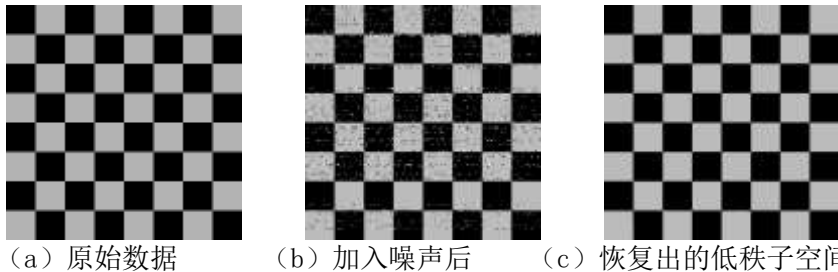


图 3 - 16 RPCA 子空间恢复 (20%行加入噪声, 噪声程度为 0.2)

从图 3-16 和图 3-17 中可以看出, 当 20%的行加入椒盐噪声, 噪声程度 Rho 为 0.2 时, RPCA 在进行子空间恢复时, 误差为 10^{-9} , 可以说明 RPCA 能够准确

地恢复出子空间结构。但是当 60% 的行加入椒盐噪声，噪声程度 Rho1 为 0.8 时，噪声误差为 0.26，RPCA 则不能有效地恢复子空间结构。其中，误差计算的方式在下面实验部分有具体说明。虽然 RPCA 在子空间恢复上取得了很好的效果，但仍然具有可以改进的地方。

算法 3-4 是非精确增广拉格朗日乘子法求解 RPCA 模型的步骤。

算法 3 - 4 非精确增广拉格朗日乘子法求解 $D=A+E$

输入：观测矩阵 D

输出：低秩矩阵 A 和稀疏矩阵 E

1. 初始化： $D=E=0$
2. $(U, \Sigma, V) = \text{svd}(D - S_k + \mu_k^{-1} Y_k), A_{k+1} = U S_{\mu_k^{-1}}(\Sigma) V^T$
3. $E_{k+1} = S_{\lambda \mu_k^{-1}}(D - A_{k+1} + \mu_k^{-1} Y_k)$
4. $Y_{k+1} = Y_k + \mu_k (D - A_{k+1} - E_{k+1})$
5. 更新拉格朗日乘子 Y^k 至 Y^{k+1} ，更新 μ_k 至 μ_{k+1}
6. $k \leftarrow k+1$

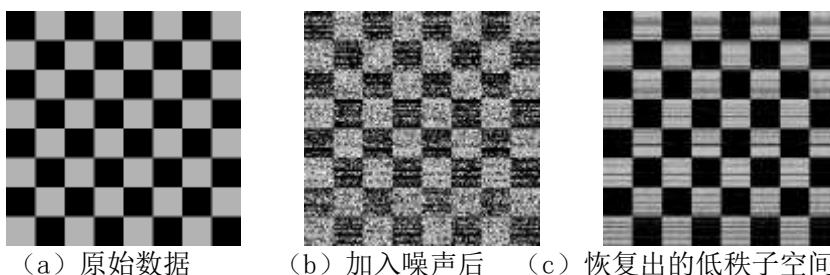


图 3 - 17 RPCA 子空间恢复（60%行加入噪声，噪声程度为 0.8）

近些年，有学者将加权 $L1$ 范数和核范数融入到 RPCA 中并取得了较好的结果^[39-47]。我们受到学者们的启发，提出了一种截断加权 RPCA，在进行核范数处理中，我们对大的奇异值直接不进行收缩，保留前 10% 的奇异值不变，对于剩余 90% 小的奇异值计算它对应的加权系数进行收缩。

3.2.2 TWRPCA 模型

本文提出的加权 RPCA 模型如下：

$$\min_{A, E} \|A\|_{\Omega, *} + \lambda \|E\|_1, \text{ s.t. } D = A + E \quad (3-20)$$

对应的拉格朗日形式如下，

$$L(A, E, Y) = \|A\|_{\Omega, *} + \lambda \|E\|_1 + \langle Y, D - A - E \rangle + \frac{\mu}{2} \|D - A - E\|_{2,1} \quad (3-21)$$

可以使用 ADM 来交替求解 A, E。

固定其他参数, 求解 A

$$\begin{aligned}
 A^{k+1} &= \arg \min_A L(A^k, E^k, Y^k) = \arg \min \|A\|_{\Omega,*} + \\
 &\langle Y, D - A - E \rangle + \frac{\mu}{2} \|D - A - E\|_{2,1} \\
 &= \arg \min \frac{1}{\mu} \|A\|_{\Omega,*} + \frac{1}{2} \left\| D - A - E + \frac{Y}{\mu} \right\|_{2,1} = \hat{U} T_{\frac{1}{\mu}}(\text{diag}(\Sigma)) \hat{V}^T
 \end{aligned} \quad (3-22)$$

其中 $\Sigma = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n, 0, \dots, 0)$, 为 $D - A - E + \frac{Y}{\mu}$ 的奇异值矩阵

在此, 为了求解 A, 我们需要使用不同的阈值对奇异值进行操作, 每个奇异值更新的权重是不一样的, 大的奇异值保存了重要的信息。我们对于大的奇异值不进行收缩操作, 对于小的奇异值进行权重收缩, 收缩操作如下公式所示:

$$\lambda(i) = \begin{cases} \lambda(i) & \text{前10\%奇异值} \\ \lambda(i) - \frac{\beta}{\mu(\lambda(i) + \varepsilon)} & \text{其它} \end{cases} \quad (3-23)$$

固定其他参数, 求解 E:

$$\begin{aligned}
 E^{k+1} &= \arg \min_A L(A^{k+1}, E^k, Y^k) = \arg \min \lambda \|E\|_1 + \\
 &\langle Y, D - A - E \rangle + \frac{\mu}{2} \|D - A - E\|_{2,1} \\
 &= S_{\lambda \mu_k^{-1}} \left(D - A - E + \frac{Y^k}{\mu} \right)
 \end{aligned} \quad (3-24)$$

算法 3 - 5 截断加权非精确增广拉格朗日乘子法求解 $D=A+E$

输入: 观测矩阵 D

输出: A, E

1. 初始化: $D=E=0$
2. 计算 $\Sigma = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n, 0, \dots, 0)$
3. $A^{k+1} = \arg \min L(A^k, E^k, Y^k)$
4. $E^{k+1} = \arg \min L(A^{k+1}, E^k, Y^k)$
5. 更新拉格朗日乘子 Y^k 至 Y^{k+1}
6. 更新 μ_k 至 μ_{k+1}
7. $k \leftarrow k + 1$

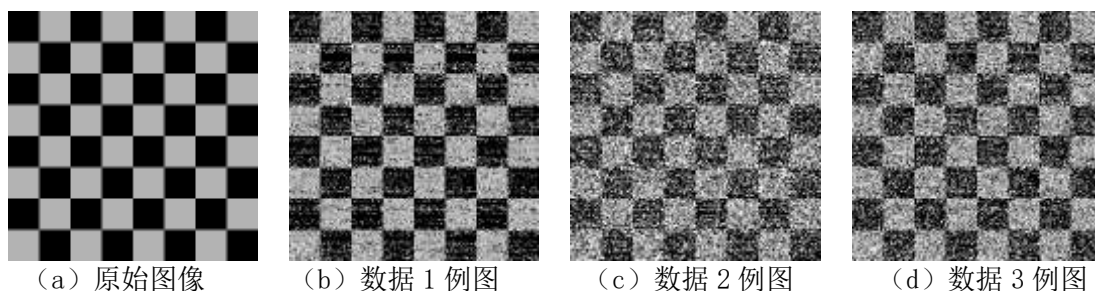


图 3 - 18 数据示例

3.2.3 子空间恢复实验

3.2.3.1 实验准备

使用 MATLAB 生成了图 3-18 (a)，图像大小为 256*256，然后使用不同的策略对图像加入噪声得到不同的实验数据。加入噪声的方法是使用 MATLAB 中 imnoise 函数。

数据 1：对图 3-18 (a) 的行加入不同程度 Rho1 的椒盐噪声，得到数据 1。图 3-18 (b) 为在 3-18 (a) 的基础上加入噪声的图像，具体是对图 3-18 (a) 中 50% 的行加入了噪声程度 Rho1 为 50% 的椒盐噪声。

数据 2：对图 3-18 (a) 加入不同程度 Rho2 的椒盐噪声，得到数据 2。如图 3-18 (c) 是在图 3-18 (a) 的基础上对整副图像加入了噪声程度 Rho2 为 0.7 的结果。

数据 3：先对图 3-18 (a) 中 50% 的行加入不同程度 Rho1 的椒盐噪声，再对整副图像加入不同程度 Rho2 的椒盐噪声，得到数据 3。图 3-18 (d) 是在图 3-18 (b) 的基础上对整副图像加入了噪声程度 Rho2 为 30% 的噪声。

数据 4：对图 3-18 (a) 中加入不同方差 Rho3 的高斯噪声，得到数据 4。

评价指标：使用恢复误差 $Error = \|D_{recover} - D_0\|_2 / \|D_0\|_2$ 作为算法精度的评价指标。 $D_{recover}$ 和 D_0 分别为恢复后的数据和原始数据。

对比方法：本文将 TWRPCA 方法与 RPCA 和 DWLP 方法进行对比，其中 DWLP 方法分别对低秩和稀疏部分来进行约束，实现子空间的恢复。

3.2.3.2 实验结果与分析

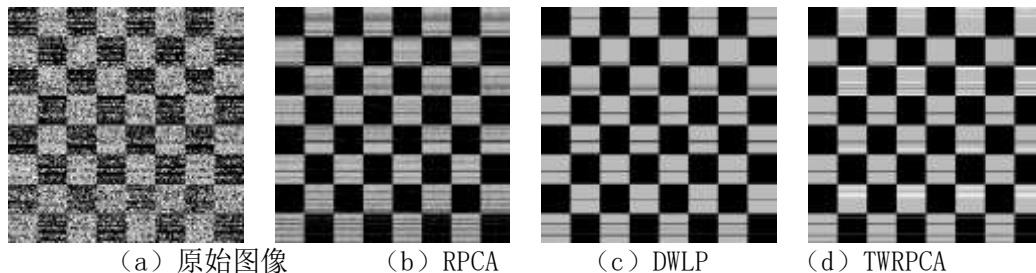


图 3 - 19 子空间恢复结果 (60%的行加入噪声，噪声程度 Rho1=0.8)

表 3 - 7 数据 1 实验误差结果

Method	30%行加入噪声			60%行加入噪声		
	Rho1=0.2	Rho1=0.5	Rho1=0.8	Rho1=0.2	Rho1=0.5	Rho1=0.8
RPCA	10^{-9}	10^{-8}	0.11	10^{-9}	10^{-8}	0.26
DWLP	10^{-9}	10^{-8}	0.05	10^{-9}	10^{-8}	0.21
TWRPCA	10^{-9}	10^{-8}	0.05	10^{-9}	10^{-8}	0.17

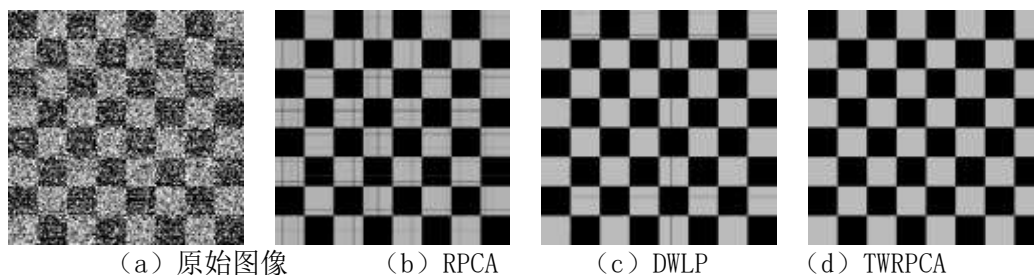


图 3 - 20 子空间恢复结果（整幅图像加入噪声程度 Rho2 为 0.7 的椒盐噪声）

表 3 - 8 数据 2 实验误差结果

Method	Rho2=0.1	Rho2=0.3	Rho2=0.5	Rho2=0.7
RPCA	10^{-9}	10^{-9}	10^{-8}	0.10
DWLP	10^{-9}	10^{-8}	10^{-8}	0.06
TWRPCA	10^{-9}	10^{-8}	10^{-8}	0.02

表 3 - 9 数据 3 实验误差结果

Method	Rho1=0.3			Rho1=0.6		
	Rho2=0.2	Rho2=0.4	Rho2=0.6	Rho2=0.2	Rho2=0.4	Rho2=0.6
RPCA	10^{-9}	10^{-8}	0.09	0.02	0.14	0.33
DWLP	10^{-9}	10^{-8}	0.08	10^{-7}	0.12	0.22
TWRPCA	10^{-9}	10^{-8}	0.06	10^{-7}	0.12	0.18

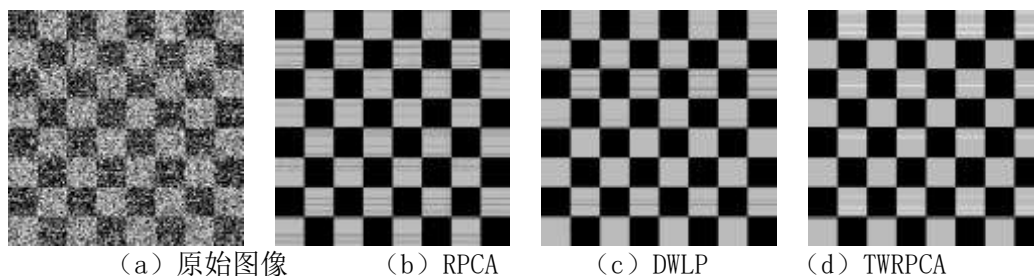


图 3 - 21 子空间恢复结果（整幅图像加入噪声程度 Rho1 为 0.6 的椒盐噪声和 50%的行加入噪声程度 Rho2 为 0.4 的椒盐噪声）

从数据 1 到数据 3 的实验结果我们可以看出，在图像加入较小的椒盐噪声时（比如：数据 1 中 30%行加入噪声，噪声程度 Rho1 小于等于 0.5 时），TWRPCA、

DWLP 和 RPCA 的误差在相同数量级上,说明在弱噪声下,三者具有相同的子空间恢复能力。而当噪声加大时(比如:数据 3 中,50%的行加入了噪声程度 Rho1 为 0.6 的噪声,除此之外再在整副图像上加入了噪声程度 Rho2 为 0.6 的椒盐噪声),RPCA 和 DWLP 进行子空间恢复的误差要大于 TWRPCA 方法,说明 TWRPCA 具有更好的子空间恢复能力。从数据 4 的实验结果可以看出,在图像加入高斯噪声时,DWLP 方法在进行子空间恢复时,误差要小于 RPCA 和 TWRPCA 方法,主要是因为它对低秩部分和稀疏部分都进行了约束。

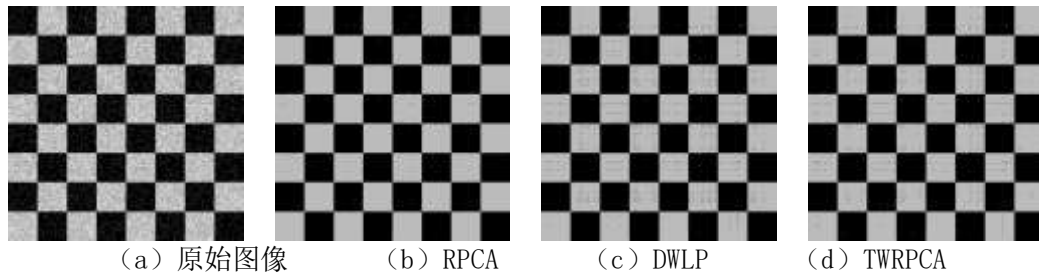


图 3 - 22 子空间恢复结果(整幅图像加入方差 Rho3 为 50 的高斯噪声)

表 3 - 10 数据 4 实验误差结果

Method	$\text{Rho3}=5$	$\text{Rho3}=20$	$\text{Rho3}=35$	$\text{Rho3}=50$
RPCA	0.04	0.05	0.05	0.08
DWLP	0.02	0.02	0.04	0.06
TWRPCA	0.02	0.03	0.04	0.07

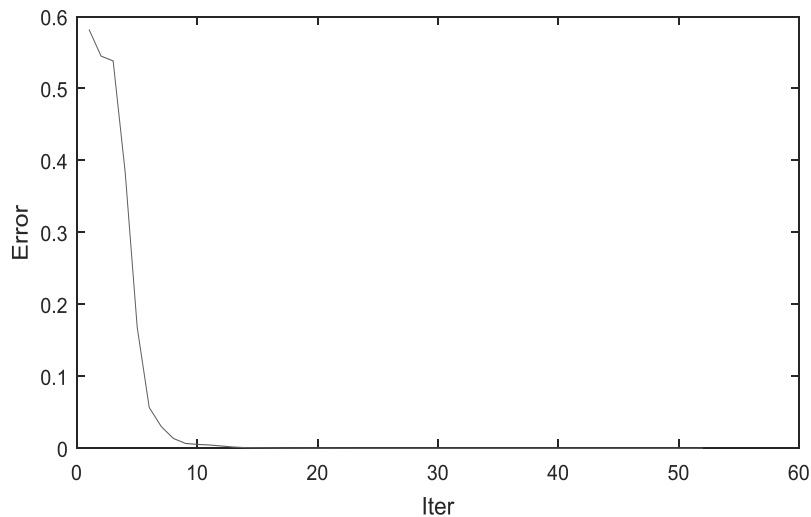


图 3 - 23 迭代损失(60%的行加入噪声,噪声程度 $\text{Rho1}=0.5$)

图 3-23 给出了 TWRPCA 模型在使用 IALM 进行求解的损失值,从中可以看出随着迭代次数的增加,损失值越来越小,说明 IALM 可以有效地求解 TWRPCA 模型。

3.3 总结

本章介绍了子空间学习基础算法研究,第一部分是基于随机探测的快速子空间分割算法:分别从不含噪声的子空间分割和包含噪声的子空间分割进行介绍,然后使用 WoodBury 公式对本文方法进行了优化,使本文方法的时间复杂度为线性,并给出了最终的时间复杂度,再使用 PIE 人脸数据库、Fashion-MNIST 数据库和合成的数据库对本文方法和对比方法进行了对比实验,实验表明本文方法 RANSP 可以对大规模数据点进行分割。第二部分是基于截断加权 RPCA 的子空间恢复算法,介绍了 RPCA 在子空间恢复上的模型,为了提高 RPCA 模型的子空间恢复能力,我们提出了一种截断加权(Truncated Weighting)的 RPCA 模型(TWRPCA)。具体地,保留前 10%的奇异值不变,其它的奇异值分别给予不同的权重。然后再使用 IALM 对 TWRPCA 模型进行了求解。接着在合成数据库中进行了对比实验,实验结果表明本文方法 TWRPCA 可以有效地提高子空间恢复的能力。在下一章我们将介绍子空间学习在视频处理中应用研究。

第四章 子空间学习在视频处理中的应用研究

在本章中，我们将介绍子空间学习方法在视频处理中的应用研究，第一部分为基于分割和显著约束的 RPCA 子空间恢复算法，并将该算法应用到运动目标检测中，第二部分为基于 3D 张量自编码器的子空间压缩算法，并将该算法应用到视频压缩中。

4.1 基于分割和显著约束的 RPCA 子空间恢复算法

4.1.1 引言

许多计算机视觉应用，如视频监控、光学运动捕捉和多媒体应用等都需要首先对背景进行建模，然后再检测移动的目标^[134]，因此获取一个不含任何目标的背景图像是迫切需要的。基于此，许多背景建模方法被提出，这些方法主要可以分为以下几种：基础背景建模、统计背景建模、模糊背景建模^[135]、背景聚类、基于神经网络的背景建模、基于小波变换的背景建模和背景估计。

近些年基于子空间学习的方法如 RPCA^[16]被成功应用到了运动目标检测中，RPCA 将视频帧拉成一行再将连续的列组成新的矩阵 D ，然后将矩阵 D 分解成为低秩矩阵 A 和稀疏矩阵 E ：

$$\min_{A,E} \|A\|_* + \lambda \|E\|_1, \text{ s.t. } D = A + E \quad (4-1)$$

其中 $\| \cdot \|_*$ 是矩阵的核范数， $\| \cdot \|_1$ 是矩阵的 L1 范数， λ 是一个大于 0 的参数， D 是视频序列矩阵，每一列都是一帧拉成的向量， A 为低秩背景， E 为稀疏前景。

在理想的状态下，背景为静态且运动目标运动相对较快，此时的运动目标检测则变成了一个很轻松的任务，RPCA 可以有效地实现运动目标检测。但是在某些场景中是无法使用的，比如背景总会不断改变、照明情况、缓慢运动的物体等。如图 4-1 所示，当 RPCA 检测波动的水面时，检测出很多异常的区域，比如水面，当检测缓慢运动目标时，运动目标的中间区域出现了空洞，发生了漏检。

为此本文提出了一种基于视频分割和显著约束的 RPCA 运动目标检测方法，为了克服动态背景，本文将运动目标和背景的时间空间的连续性使用视频分割融入到 RPCA 模型中，为了克服缓慢运动目标，本文将运动目标的显著性融入到 RPCA 模型中，最终实现了动态背景和缓慢情况下的运动目标检测。

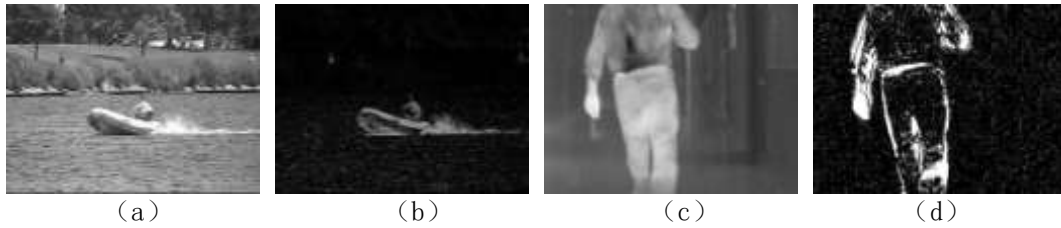


图 4 - 1 RPCA 检测结果：（a）波动的水面；（b）波动的水面检测结果；（c）缓慢运动；（d）缓慢运动检测结果

4.1.2 SC-RPCA 模型与 SSC-RPCA 模型

在本节中，我们将详细介绍本文提出的方法 **SSC-RPCA** 在运动目标检测中的使用。我们先介绍一种叫做 **SC-RPCA** 的方法，这种方法将视频分割融入到 **RPCA** 模型中；然后我们再将显著特征融入到 **SC-RPCA** 模型中，得到了 **SSC-RPCA** 模型。

4.1.2.1 分割约束与 SC-RPCA

为了提高 **RPCA** 模型在动态背景下运动目标检测的鲁棒性，我们将运动目标和背景的时间与空间的连续性纳入 **RPCA** 模型中。时间和空间的连续性是通过如下的视频分割方式得到的，如算法 4-1 所示，我们首先使用超像素分割^[136]方法得到图像的超像素块，这样就得到了运动目标和背景的局部连续性，然后再将局部的超像素块进行合并便得到了运动目标和背景的空间连续性^[137]，最后通过判断相邻块的距离再将相邻帧的相似块进行连接最终实现了视频分割，最终实现了视频的时间连续性，如图 4-2 所示：

算法 4 - 1 视频分割

输入：连续的视频帧 $\{I_1 \cdots I_N\}$

输出：分割结果

1. For $k=1$ to N
 - 对图像 I_k 进行超像素分割
 - 对超像素块进行合并Next k
 2. Repeat
 - 在 N 帧视频中寻找邻接区域
 - 对邻接区域进行合并Until 没有区域需要合并
-

从图4-2中可以看出，超像素分割将附近邻域的相似的像素点进行了有效的聚类，而图像分割则可以将超像素块再进行相似性合并，这样可以得到更大的

图像相似块。

经过视频分割后，我们可以得到运动目标视频背景的时间与空间连续性特征，为了便于表示，我们将视频结果表示为 C ，其中 C_i 表示为视频分割中第 i 个组， m 是由视频分割算法自动生成的，它为组的个数。

$$C = C_1 \cup C_2 \cup \dots \cup C_m \quad (4-2)$$



图 4-2 获得视频分割约束：(a) 输入原图；(b) 超像素分割^[136]结果；(c) 图像分割^[137]结果；(d) 视频分割结果

在同一个组中的像素具有相同的特征，要么都为运动目标要么都为背景。有了视频分割的信息后，便于我们对运动目标完整性进行约束，同时也能约束动态背景。因此我们将视频分割结果融入到 RPCA 模型中，得到如下的 SC-RPCA 模型：

$$\min_{A,E} \|A\|_* + \lambda \|E\|_{C(2,1)}, s.t. D = A + E \quad (4-3)$$

其中 λ 为大于0的超参数，SC-RPCA 模型和现有的一些基于 RPCA 的方法相类似，比如块稀疏 RPCA^[74]， $\|E\|_{C(2,1)}$ 的定义如下：

$$\|E\|_{C(2,1)} = \sum_{i=1}^m \sqrt{|C_i| \sum_{(j,k) \in C_i} ([E]_{jk})^2} \quad (4-4)$$

公式(4-3)可以使用非精确增广拉格朗日乘法 (IALM) 进行交替迭代求解，公式(4-3)的拉格朗日形式为：

$$L(A, E, Y, \mu) = \|A\|_* + \lambda \|E\|_{C(2,1)} + \langle Y, D - A - E \rangle + \frac{\mu}{2} \|D - A - E\|_F^2 \quad (4-5)$$

其中 Y 是拉格朗日乘子， μ 是惩罚系数。

公式(4-5)是一个凸函数具有闭式解，可以使用奇异值收缩 (Singular Value Thresholding, SVT)^[16]算子进行求解，得到如下的求解步骤：

$$A = \arg \min_A L(A, E, Y, \mu) = \arg \min_E \frac{\lambda}{\mu} \|A\|_* + \frac{1}{2} \left\| A - \left(D - E + \frac{Y}{\mu} \right) \right\|_F^2 \quad (4-6)$$

$$E = \arg \min_E L(A, E, Y, \mu) = \arg \min_E \frac{\lambda}{\mu} \|E\|_{C(2,1)} + \frac{1}{2} \left\| E - \left(D - A + \frac{Y}{\mu} \right) \right\|_F^2 \quad (4-7)$$

令 $M = D - A + \frac{Y}{\mu}$, 假设 $(j, k) \in C_i$, $\|[M]_{C_i}\|_2 = \sqrt{\sum_{(a,b) \in C_i} ([M]_{ab})^2}$, 上式的解如下:

$$[E]_{jk} = \begin{cases} \frac{\|[M]_{C_i}\|_2 - \frac{\lambda \sqrt{|C_i|}}{2\mu}}{\|[M]_{C_i}\|_2} [M]_{jk}, & \text{if } \|[M]_{C_i}\|_2 > \frac{\lambda \sqrt{|C_i|}}{2\mu} \\ 0, & \text{otherwise} \end{cases} \quad (4-8)$$

图 4-3 给出了 SC-RPCA 的部分实验结果, 我们将它与图 4-1 中 RPCA 的检测检测结果进行比较, 从中可以看出, 通过视频分割约束后, 在动态背景下, SC-RPCA 模型可以有效地对背景进行抑制, 比如波动的湖面和红外图像的噪点。

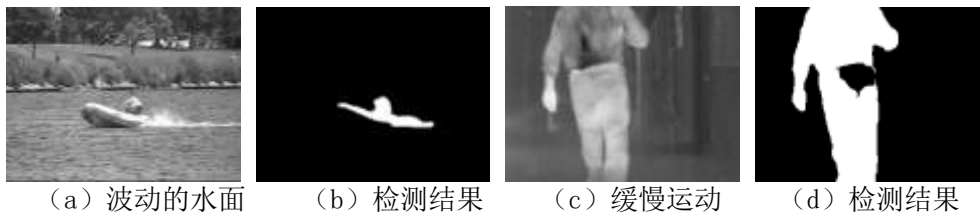


图 4 - 3 SC-RPCA 检测结果

算法 4 - 2 SC-RPCA 运动目标检测

输入: 图像序列 $\{I_k\}_{k=1}^N$, 视频分割结果 C

输出: A, E

1. 初始化: $A = E = Y = 0$, $\mu = 10^{-6}$
2. While 不收敛 do

固定其他参数, 更新 $A = \arg \min_A L(A, E, Y, \mu)$
 固定其他参数, 更新 $E = \arg \min_E L(A, E, Y, \mu)$
 固定其他参数, 更新 $Y = Y + \mu(D - A - E)$

End while



图 4 - 4 RPCA 显著检测: (a) 输入图像, (b) 显著检测结果

4.1.2.2 显著约束与 SSC-RPCA

尽管 SC-RPCA 模型相对于 RPCA 模型取得了良好的效果,但是其仍有不足之处,比如缓慢运动目标的检测,容易产生一些空洞,造成了漏检。为了克服问题,本文将运动目标的显著性融入到 SC-RPCA 模型中,运动目标显著性是从单张图片中获取的,因此可以提供一些约束来增加缓慢运动目标检测的鲁棒性。

然而显著信息的获取仍然是一个具有挑战性的任务。面对一个图像序列,本文将同时实现运动目标检测和显著信息提取,而不是像文献^[78]需要单独提取显著信息,然后再纳入到 RPCA 模型中,受到了模型^[138]的启发,我们将使用 RPCA 来计算图像序列的显著信息:

$$\min_{L_k, S_k} \|L_k\|_* + \alpha \|S_k\|_1, \text{ s.t. } I_k = L_k + S_k \quad (4-9)$$

其中 α 为大于 0 的参数, I_k 为单幅图像。

图 4-4 给出了显著信息检测的结果,从中可以看出显著信息可以准确地定位前景目标。

现在我们将显著信息提取融入到 SC-RPCA 模型中,得到了 SSC-RPCA 模型:

$$\begin{aligned} \min_{A, E} \quad & \|A\|_* + \lambda \|E\|_{C(2,1)} + \beta \sum_{k=1}^N (\|L_k\|_* + \alpha \|S_k\|_1) \\ \text{s.t.} \quad & D = A + E, I_k = L_k + S_k, k = 1, \dots, N, \\ & A = [\text{vec}(L_1), \dots, \text{vec}(L_N)], E = [\text{vec}(S_1), \dots, \text{vec}(S_N)] \end{aligned} \quad (4-10)$$

其中 $\{I_k\}_{k=1}^N$ 是一个具有 N 帧的视频序列, C 为视频分割结果, VEC(:) 是一个将矩阵变成向量的操作算子, $D = [\text{vec}(I_1), \dots, \text{vec}(I_N)]$, λ 、 α 和 β 是超参数。

图 4-5 为 SSC-RPCA 检测结果,相对于 SC-RPCA 模型,在“boats”视频中,SSC-RPCA 可以获得更为完整的船体,在“corridor”视频中,SSC-RPCA 可以检测出更完整的人体,没有造成“空洞”现象,因此从中可以看出 SSC-RPCA 通过显著信息可以获得更加完整的运动目标检测,降低了漏检率。

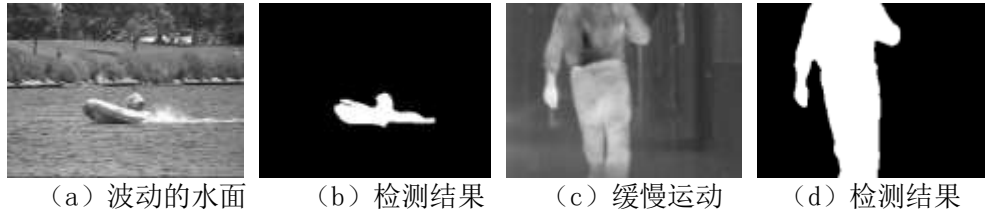


图 4 - 5 SSC-RPCA 检测结果

4.1.2.3 SSC-RPCA 模型优化求解

虽然 SSC-RPCA 模型看起来比较复杂,但是其是凸函数,和 SC-RPCA 模型

相类似，可以使用许多基于梯度的方法来进行求解。为了方便起见，我们选择经典的非精确增广拉格朗日乘子法^[139]来进行求解，首先我们根据 SSC-RPCA 模型定义拉格朗日函数：

$$\begin{aligned} Lag = & \|A\|_* + \lambda \|E\|_{C(2,1)} + \beta \sum_{k=1}^N (\|L_k\|_* + \alpha \|S_k\|_1) + \\ & \langle D - A - E, Y \rangle + \sum_{k=1}^N (\langle I_k - L_k - S_k, W_k \rangle) + \\ & \langle \text{mat}([A]_{\cdot k}) - L_k, V_k \rangle + \langle \text{mat}([E]_{\cdot k}) - S_k, H_k \rangle + \frac{\mu}{2} \|D - A - E\|_F^2 \\ & + \frac{\mu}{2} \sum_{k=1}^N (\|I_k - L_k - S_k\|_F^2 + \|\text{mat}([A]_{\cdot k}) - L_k\|_F^2 + \|\text{mat}([E]_{\cdot k}) - S_k\|_F^2) \end{aligned} \quad (4-11)$$

其中 $[\cdot]_k$ 是矩阵的第 k 列， $\text{mat}(\cdot)$ 是 $\text{vec}(\cdot)$ 操作算子的逆操作， μ 为惩罚系数， Y ， $\{W_k\}_{k=1}^N$ ， $\{V_k\}_{k=1}^N$ 和 $\{H_k\}_{k=1}^N$ 为拉格朗日乘子。

求解上式最小化，可以交替迭代进行求解。

固定其他参数：求解 $\{L_k\}_{k=1}^N$ ，使用如下公式更新 $\{L_k\}_{k=1}^N$

$$\begin{aligned} & \beta \|L_k\|_* + \langle -L_k, Y_k \rangle + \langle -L_k, W_k \rangle + \\ & \frac{\mu}{2} \|L_k - (f_k(D) - S_k)\|_F^2 + \frac{\mu}{2} \|L_k - f_k(A)\|_F^2 \end{aligned} \quad (4-12)$$

$$\begin{aligned} & \beta \|L_k\| - \langle L_k, Y_k + W_k \rangle + \frac{\mu}{2} \|L_k\|_F^2 - \\ & \mu \langle L_k, f_k(D) - S_k \rangle + \frac{\mu}{2} \|L_k\|_F^2 - \mu \langle L_k, f_k(A) \rangle \end{aligned} \quad (4-13)$$

$$\begin{aligned} & \frac{\beta}{\mu} \|L_k\| - \left\langle L_k, \frac{Y_k + W_k}{\mu} \right\rangle + \|L_k\|_F^2 - \\ & \langle L_k, f_k(D) - S_k \rangle + \frac{\mu}{2} \|L_k\|_F^2 - \langle L_k, f_k(A) \rangle \end{aligned} \quad (4-14)$$

$$\frac{\beta}{\mu} \|L_k\|_* - \left\langle L_k, \frac{Y_k + W_k}{\mu} + f_k(D) - S_k + f_k(A) \right\rangle + \|L_k\|_F^2 \quad (4-15)$$

$$\min_{L_k} \frac{\beta}{2\mu} \|L_k\|_* + \frac{1}{2} \left\| L_k - \left(\frac{W_k + V_k}{2\mu} + \frac{I_k - S_k + \text{mat}([A]_{\cdot k})}{2} \right) \right\|_F^2 \quad (4-16)$$

根据奇异值阈值^[140] (Singular Value Thresholding, SVT) 上式具有闭式解。

固定其他参数：求解 $\{S_k\}_{k=1}^N$

$$\min_{L_k} \frac{\alpha\beta}{2\mu} \|S_k\|_* + \frac{1}{2} \left\| S_k - \left(\frac{W_k + H_k}{2\mu} + \frac{I_k - L_k + \text{mat}([E]_{jk})}{2} \right) \right\|_F^2 \quad (4-17)$$

上式根据收缩算子求解。

固定其他参数：求解 A ，定义 $L = [\text{vec}(L_1), \dots, \text{vec}(L_N)]$ ， $V = [\text{vec}(V_1), \dots, \text{vec}(V_N)]$

$$\|A\|_* - \langle A, Y \rangle + \sum_{i=1}^N \langle f_i(A), V_i \rangle + \frac{u}{2} \|A - (D - C)\|_F^2 + \frac{\mu}{2} \sum_{i=1}^N \|f_i(A) - L_i\| \quad (4-18)$$

$$\|A\|_* - \langle A, Y - V \rangle + \frac{\mu}{2} \|A - (D - C)\|_F^2 + \frac{u}{2} \|A - L\|_F^2 \quad (4-19)$$

$$\|A\|_* - \langle A, Y - V \rangle + \mu \|A\|_F^2 - \mu \langle A, D - C \rangle - \mu \langle A, L \rangle \quad (4-20)$$

$$\min_A \frac{1}{2\mu} \|A\|_* + \frac{1}{2} \left\| A - \left(\frac{Y - V}{2\mu} + \frac{D - C + L}{2} \right) \right\|_F^2 \quad (4-21)$$

上式使用 SVT 进行求解。

固定其他参数：求解 E ，定义 $S = [\text{vec}(S_1), \dots, \text{vec}(S_N)]$ ， $H = [\text{vec}(H_1), \dots, \text{vec}(H_N)]$

$$\min_E \frac{1}{2\mu} \|E\|_* + \frac{1}{2} \left\| E - \left(\frac{Y - V}{2\mu} + \frac{D - A + S}{2} \right) \right\|_F^2 \quad (4-22)$$

上式通过如下的步骤进行求解，定义

$$M = \frac{Y - H}{2\mu} + \frac{D - A - S}{2} \quad (4-23)$$

$$\|[M]_{C_i}\|_2 = \sqrt{\sum_{(j_1, j_2) \in C_i} ([M]_{j_1 j_2})^2} \quad (4-24)$$

根据文献^[22]，可以如下的方式进行求解

$$[E]_{jk} = \begin{cases} \frac{\|[M]_{C_i}\|_2 - \frac{\lambda\sqrt{|C_i|}}{2\mu}}{\|[M]_{C_i}\|_2} [M]_{jk}, & \text{if } \|[M]_{C_i}\|_2 > \frac{\lambda\sqrt{|C_i|}}{2\mu} \\ 0, & \text{otherwise} \end{cases} \quad (4-25)$$

固定其他参数：求解拉格朗日乘子

$$\begin{aligned}
 Y &= Y + \mu(D - A - E) \\
 W_k &= W_k + \mu(I_k - L_k + S_k) \\
 V_k &= V_k + \mu(\text{mat}([A]_{:,k}) - L_k) \\
 H_k &= H_k + \mu(\text{mat}([E]_{:,k}) - L_k) \\
 \mu &= \rho\mu
 \end{aligned} \tag{4-26}$$

其中 ρ 是用来控制收敛速度，在本文中其被设置为 1.1。

算法 4 - 3 求解 SSC-RPCA 运动目标检测

输入：图像序列 $\{I_k\}_{k=1}^N$ ，视频分割结果 C

输出：A, E

1. 初始化： $A = E = Y = 0$ ， $W_k = V_k = H_k = S_k = L_k = 0$ ， $1 \leq k \leq N$ ， $\mu = 10^{-6}$
2. While 不收敛 do
 - 固定其他参数，更新 $L_k (1 \leq k \leq N)$
 - 固定其他参数，更新 $S_k (1 \leq k \leq N)$
 - 固定其他参数，更新 A
 - 固定其他参数，更新 E
 - 固定其他参数，更新拉格朗日乘子

End While

整个 SSC-RPCA 模型的优化求解过程在上面算法中，因为 SSC-RPCA 同样需要对视频序列计算 SVD 分解，因此其具有和 RPCA 相同的时间复杂度。

4.1.3 运动目标检测实验

4.1.3.1 实验设置

实验视频：为了验证 SSC-RPCA 模型的有效性，本文在 CDNET 2014^[19] 公共数据集上进行了实验。一共在 14 个视频上进行了对比，这些视频中包含了具有挑战性的场景，缓慢运动的目标，动态背景（晃动的树枝树叶、波动的湖面），下雪天。本文将这些视频分为三组：正常运动（“fall”，“fountain02”，“blizzard”，“skating”，“snowfall”）、缓慢运动（“boats”，“copyMachine”，“corridor”，“twoPositionPTZCam”）、非常缓慢运动（“canoe”，“peopleInshade”，“library”，“overpass”，“winterDriveway”），本文将每个视频进行编号，信息见表 4-1。

对比方法：为了验证方法的优越性，本文选取了 15 种运动目标检测方法，包括 DeepBS^[68]，M4CD^[141]，SWCD^[142]，WeSamBE^[143]，SCM^[78]，SBBS^[144]，MST^[145]，

MBS^[146], ED^[147], pROST^[148], SMoG^[149], GMM^[49], ThreeWayDec^[77], Decolor^[84], and Godec^[83]。其中 SCM, ThreeWayDec, Decolor 和 Godec 是基于 RPCA 模型的。

参数设置: SSC-RPCA 中的参数 α 可以通过文献^[70]中的方法设置, 具体如下: 假设输入图像维度为 $p \times q$, 我们设置 $\alpha = 1/\sqrt{\max(p, q)}$, 参数 λ 是用来权衡前景矩阵与背景矩阵的, 它的范围是 $0.1/\sqrt{pq}$ 至 $10/\sqrt{pq}$, 参数 β 用来控制显著信息约束能力, 它的范围为 0.001 至 0.2。

评价指标: 本文从检测的速度与准确度来对算法进行评价, 检测速度采用的是 Frame Per Second (FPS), 准确率采用的是 True Positive (TP)、True Negative (TN)、False Positive (FP) 组成的召回率 (Recall, R)、精确率 (Precision, P) 和 F-measure (F) 值来对实验结果进行评价, 其中 TP 为被正确分类的前景像素点的个数, TN 为被正确分类的背景像素点的个数, FP 为被错误分类的背景像素点的个数, FN 为错误分类的前景像素的个数^[19]。

$$\begin{aligned} \text{Recall} &= \frac{TP}{TP + FN} \\ \text{Precision} &= \frac{TP}{TP + FP} \\ F - \text{measure} &= 2 \frac{\text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} \end{aligned} \quad (4-27)$$

4.1.3.2 结果与分析

SC-RPCA 与 SSC-RPCA 实验比较: 图 4-3 和图 4-5 已经通过实验证明了视频分割和显著约束对运动目标的有效性。为了清晰起见, 我们将以定量的方式再给出 SC-RPCA 与 SSC-RPCA 的比较, 使用的实验视频为慢动作场景“boats”, 选取了连续的 50 帧进行评估, 在这 50 帧中有缓慢运动的物体, 也有动态的湖面背景。从图 4-6 中可以看出, 通过调节不同的参数 λ , SC-RPCA 方法的 F 的值要高于 RPCA 方法的 F 值的, 说明 SC-RPCA 的效果优于 RPCA 的效果, 证明了显著约束和视频分割约束对于提高动态背景下缓慢运动目标检测是有效的, 通过调节不同的 β , 说明了 SSC-RPCA 在增加了显著约束后, SSC-RPCA 的 F 的值高于 SC-RPCA 的 F 的值, 效果明显得到了提升, 说明显著约束能够有效地提升检测结果。与其它方法的比较: 表 4-2 到表 4-7 展示了这 16 种方法的实验客观评价结果, 从中我们可以看出, 从 F 评价指标进行评价, 在 14 个测试视频中, 本文 SSC-RPCA 效果要优于其他基于 RPCA 的方法 (SCM, ThreeWayDec, Decolor 和 Godec)。SBBS, pROST, 和 MBS 方法取得了相对较高的 F 值。ED 取得了较低的 F 值。除了“library”视频, SMoG 和 GMM 方法取得了良好的结果。MST,

DeepBS, M4CD, SWCD 和 WeSamBE 在大多数的视频中获得了令人满意的结果。在 10 个视频中, SSC-RPCA 取得了最高的 F, 这说明了视频分割和显著信息可以明显提高 RPCA 的运动目标检测的鲁棒性。

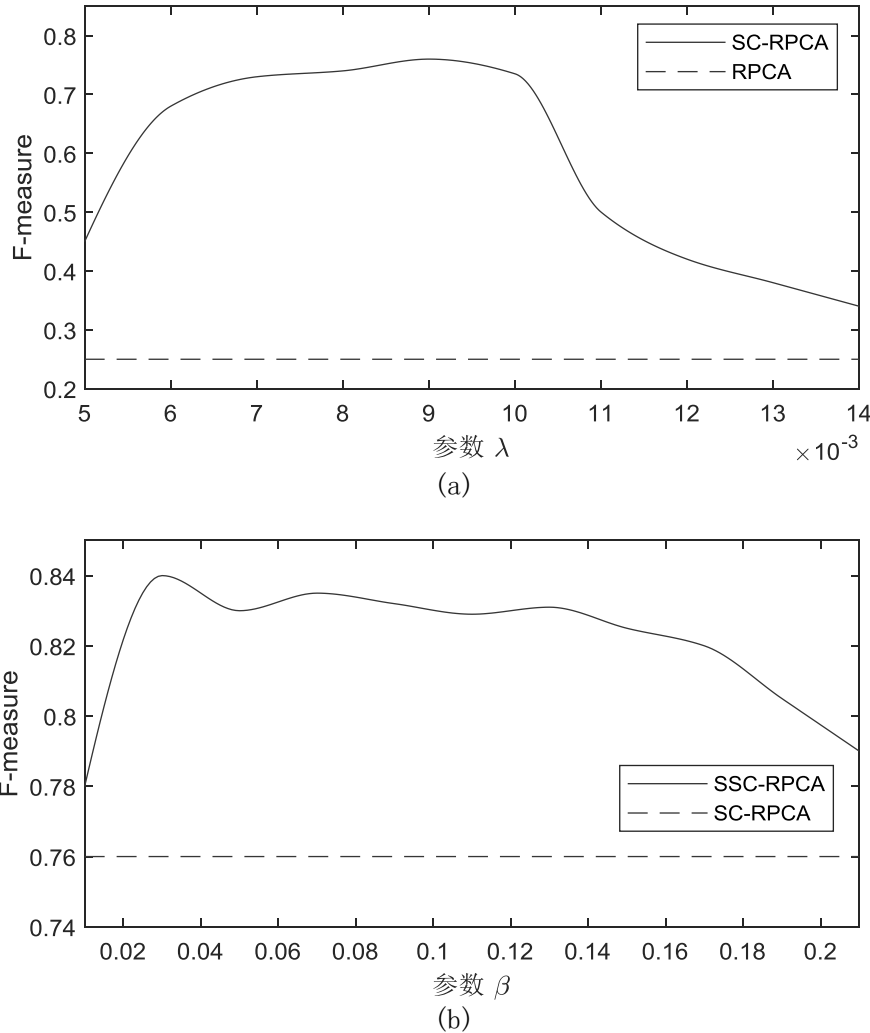


图 4-6 SSC-RPCA 与 RPCA、SC-RPCA 方法的比对: (a) SC-RPCA 与 RPCA 的比对, 在不同的参数 λ 下, (b) SSC-RPCA 与 SC-RPCA 的比对, 在不同的参数 β 下

从图 4-9 到图 4-18 给出了一些方法的实验结果图, 从中可以看出在大多数的视频中, 只有 ThreeWayDec 和 SCM 方法漏检率较高, 基本上没有检测到运动目标。受到晃动的树叶的影响, 在“fall”视频中 ThreeWayDec, Decolor 和 Godec 错误地将很多背景检测为前景。在“boats”视频中, ThreeWayDec 和 Godec 受到波动的水面的影响, 检测结果中包含了很多的背景。当检测“corridor”视频时, 因为目标运动缓慢, MST, GMM, ThreeWayDec, Decolor 和 Godec 方法在检测结果中造成了很多的空洞。在“library”视频中, 只有 GMM 和 SSC-RPCA 方法检测到了目标。除了“skating”, “blizzard”和“library”视频, DeepBS 使用 CNN 检测到了大多数的运动目标。

SSC-RPCA 因为分割不精确导致其丢失了运动目标的部分区域,如图 4-19 中,在“fountain02”视频中,车轮被分割到和路面一个区域中,在“copyMachine”视频中,人物鞋子的颜色和地面的颜色很接近,因此被划分到和地面一个区域中,在“library”视频中,这个视频为红外视频,因为书的温度和桌子的温度比较接近,因此其被划分到和桌面一个区域中,在检测中被认为是背景。相对于其他方法,SSC-RPCA 能够较好地检测出整个运动目标,不会造成“空洞”现象,同时能够抑制复杂的背景信息(波动的水面、晃动的树叶等等)。

表 4-8 给出了所有对比方法处理一张分辨率为 320×240 图像的时间。从中可以看出,ED 因为其只需要执行简单的计算并且使用了 C++ 编程,因此其运算速度非常快。pROST 采用了图形处理器(Graphics Processing Unit, GPU)加速,有效地提高了运行速度,ED 和 pROST 可以满足实际的应用。MST, GMM, SMoG, SBBS, MBS, DeepBS 和 SWCD 运算速度相对较快。其中, GMM 本身时间复杂度较低, DeepBS 采用了 GPU 加速。基于 RPCA 的方法运算速度较慢,是因为其在进行运动目标检测时需要进行矩阵奇异值分解。ThreeWayDec 因为要基于粒子平流和光流计算可信度地图预处理,所以运算成本高。SSC-RPCA 中采用的超像素分割需要进行大矩阵的求逆操作,因此运算速度较慢。

表 4-1 实验视频信息及编号

正常运动	缓慢运动	非常缓慢运动
Fountain02 (001)	Boats (006)	Canoe (010)
Fall (002)	Copymachine (007)	PeopleInhade (011)
Blizzard (003)	Corridor (008)	Library (012)
Skating (004)	Twopositionptzcam (009)	Overpass (013)
Snowfall (005)		Winterdriveway (014)



图 4-7 输入图片: 分别为实验中 14 个视频中的一张图片, 从左至右, 从上到下, 分别为编号 001 到编号 014



图 4-8 Ground Truth

表 4 - 2 正常运动部分实验结果

Video	M	MST	ED	GMM	pROST	SMoG	SBBS	MBS	DeepBS
001	R	0.770	0.570	0.758	0.640	0.707	0.715	0.738	0.895
	P	0.983	0.966	0.995	0.997	0.998	0.999	0.996	0.966
	F	0.863	0.716	0.860	0.779	0.828	0.833	0.848	0.929
002	R	0.413	0.206	0.491	0.704	0.840	0.752	0.695	0.792
	P	0.978	0.950	0.978	0.984	0.977	0.993	0.669	0.917
	F	0.543	0.327	0.625	0.804	0.896	0.824	0.618	0.823
003	R	0.958	0.955	0.800	*	0.847	0.842	0.831	0.973
	P	0.783	0.292	0.980	*	0.352	0.843	0.986	0.577
	F	0.862	0.447	0.881	*	0.528	0.841	0.901	0.724
004	R	0.670	1.000	0.984	*	0.945	0.936	0.930	0.992
	P	0.948	0.558	0.951	*	0.920	0.988	0.995	0.864
	F	0.778	0.681	0.967	*	0.932	0.962	0.962	0.920
005	R	0.986	0.931	0.880	*	0.906	0.900	0.812	0.968
	P	0.553	0.420	0.641	*	0.720	0.680	0.954	0.763
	F	0.708	0.561	0.740	*	0.803	0.774	0.881	0.851

表 4 - 3 缓慢运动部分实验结果

Video	M	MST	ED	GMM	pROST	SMoG	SBBS	MBS	DeepBS
006	R	0.582	0.691	0.714	0.646	0.731	0.735	0.708	0.726
	P	0.989	0.826	0.683	0.975	0.903	0.800	0.833	0.967
	F	0.731	0.750	0.696	0.774	0.804	0.760	0.761	0.820
007	R	0.737	0.426	0.789	0.355	0.813	0.667	0.930	0.943
	P	0.918	0.828	0.892	0.923	0.881	0.971	0.906	0.967
	F	0.817	0.562	0.834	0.513	0.843	0.782	0.917	0.955
008	R	0.955	0.960	0.919	0.915	0.932	0.928	0.925	0.917
	P	0.836	0.843	0.946	0.846	0.879	0.927	0.993	0.826
	F	0.891	0.898	0.933	0.879	0.905	0.928	0.958	0.869
009	R	0.245	0.595	0.614	*	0.726	0.811	0.771	0.925
	P	0.557	0.554	0.650	*	0.740	0.414	0.892	0.775
	F	0.340	0.516	0.632	*	0.730	0.541	0.825	0.838

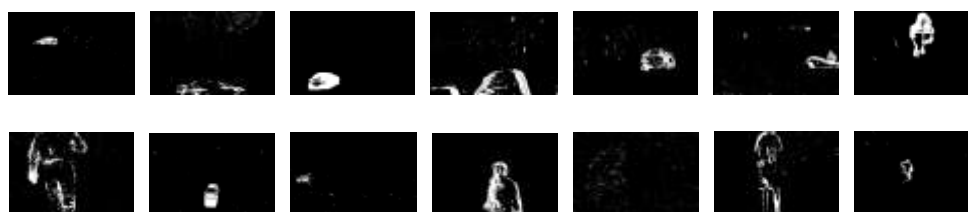


图 4 - 9 SCM 检测结果

表 4 - 4 非常缓慢运动部分实验结果

Video	M	MST	ED	GMM	pROST	SMoG	SBBS	MBS	DeepBS
010	R	0.754	0.840	0.638	0.897	0.981	0.846	0.997	0.939
	P	0.985	0.763	0.994	0.936	0.769	0.993	0.779	0.990
	F	0.854	0.792	0.776	0.916	0.879	0.913	0.874	0.908
011	R	0.883	0.900	0.999	0.983	0.998	0.913	0.993	0.936
	P	0.977	0.940	0.866	0.967	0.652	0.986	0.885	0.996
	F	0.928	0.920	0.928	0.975	0.787	0.948	0.935	0.965
012	R	0.953	0.786	0.588	0.900	0.529	0.987	0.899	0.827
	P	0.527	0.390	0.010	0.459	0.003	0.843	0.918	0.154
	F	0.697	0.471	0.020	0.608	0.008	0.909	0.908	0.261
013	R	0.830	0.822	0.814	0.725	0.798	0.793	0.752	0.802
	P	0.827	0.764	0.746	0.858	0.871	0.793	0.866	0.899
	F	0.828	0.791	0.778	0.785	0.831	0.792	0.788	0.845
014	R	0.492	0.746	0.769	0.793	0.476	0.335	0.601	0.480
	P	0.989	0.819	0.852	0.883	0.585	0.997	0.886	0.986
	F	0.656	0.780	0.807	0.834	0.516	0.502	0.711	0.645

表 4 - 5 正常运动部分实验结果

Video	M	M4CD	SWCD	We SamBE	SCM	Three WayDec	Decolor	GoDec	SSC RPCA
001	R	0.730	0.763	0.738	0.754	0.587	0.334	0.515	0.861
	P	0.999	0.999	0.998	0.764	0.172	0.996	0.862	0.912
	F	0.844	0.865	0.848	0.758	0.266	0.501	0.645	0.880
002	R	0.431	0.799	0.855	0.600	0.704	0.456	0.529	0.974
	P	0.989	0.985	0.935	0.203	0.579	0.859	0.786	0.841
	F	0.563	0.856	0.888	0.304	0.636	0.590	0.632	0.898
003	R	0.823	0.811	0.782	0.804	0.857	0.678	0.918	0.926
	P	0.962	0.938	0.997	0.927	0.710	0.997	0.966	0.973
	F	0.886	0.865	0.875	0.858	0.776	0.801	0.939	0.948
004	R	0.944	0.794	0.949	0.562	0.702	0.871	0.753	0.995
	P	0.996	0.992	0.975	0.400	0.625	0.999	0.690	0.947
	F	0.970	0.878	0.962	0.454	0.663	0.931	0.712	0.970
005	R	0.891	0.861	0.883	0.658	0.846	0.744	0.799	0.949
	P	0.748	0.808	0.977	0.466	0.466	0.855	0.476	0.931
	F	0.812	0.833	0.927	0.543	0.601	0.793	0.588	0.939

表 4 - 6 缓慢运动部分实验结果

Video	M	M4CD	SWCD	We SamBE	SCM	Three WayDec	De color	Go Dec	SSC- RPCA
006	R	0.715	0.696	0.717	0.554	0.156	0.468	0.284	0.743
	P	0.994	0.978	0.966	0.629	0.628	0.665	0.691	0.956
	F	0.827	0.811	0.822	0.589	0.250	0.548	0.400	0.833
007	R	0.763	0.870	0.899	0.744	0.866	0.955	0.953	0.998
	P	0.979	0.973	0.978	0.582	0.839	0.707	0.710	0.792
	F	0.853	0.918	0.937	0.650	0.846	0.808	0.871	0.883
008	R	0.921	0.908	0.890	0.704	0.401	0.820	0.608	0.954
	P	0.979	0.958	0.999	0.361	0.071	0.718	0.238	0.984
	F	0.949	0.932	0.942	0.477	0.121	0.749	0.326	0.969
009	R	0.817	0.828	0.853	0.813	0.812	0.416	0.772	0.901
	P	0.861	0.864	0.905	0.733	0.768	0.963	0.658	0.920
	F	0.838	0.840	0.875	0.769	0.789	0.576	0.705	0.920

表 4 - 7 非常缓慢运动部分实验结果

Video	M	M4CD	SWCD	We SamBE	SCM	Three WayDec	De color	Go Dec	SSC- RPCA
010	R	0.816	0.755	0.831	0.689	0.005	0.581	0.282	0.859
	P	0.996	0.991	0.984	0.285	0.096	0.255	0.754	0.987
	F	0.896	0.856	0.900	0.404	0.009	0.352	0.385	0.918
011	R	0.894	0.869	0.891	0.598	0.837	0.618	0.609	0.884
	P	0.999	0.951	0.998	0.336	0.134	0.089	0.704	0.989
	F	0.943	0.908	0.941	0.406	0.231	0.148	0.624	0.933
012	R	0.966	0.920	0.782	0.842	0.785	0.833	0.344	0.910
	P	0.361	0.908	0.249	0.200	0.051	0.328	0.223	0.912
	F	0.526	0.914	0.378	0.323	0.095	0.326	0.271	0.911
013	R	0.764	0.740	0.614	0.382	0.794	0.739	0.651	0.892
	P	0.964	0.787	0.397	0.383	0.637	0.641	0.384	0.838
	F	0.851	0.762	0.461	0.376	0.707	0.685	0.479	0.860
014	R	0.425	0.353	0.286	0.232	0.829	0.562	0.287	0.741
	P	0.997	0.997	0.997	0.464	0.448	0.498	0.779	0.985
	F	0.595	0.520	0.444	0.310	0.582	0.534	0.420	0.845

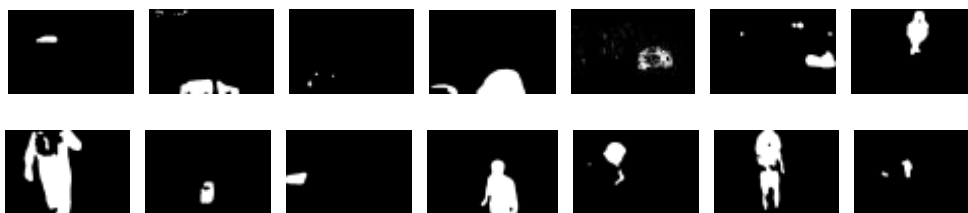


图 4 - 10 DeepBS 检测结果



图 4 - 11 MBS 检测结果

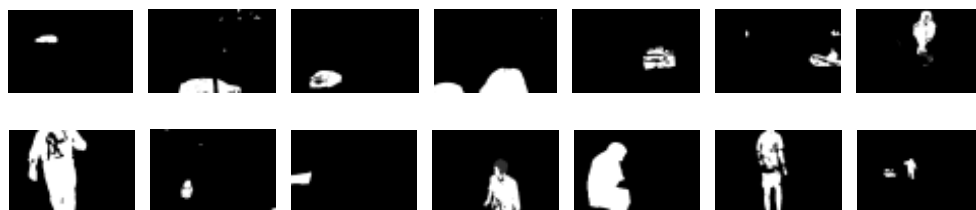


图 4 - 12 SBBS 检测结果



图 4 - 13 MST 检测结果



图 4 - 14 GMM 检测结果



图 4 - 15 ThreeWayDec 检测结果



图 4 - 16 Decolor 检测结果





图 4 - 17 GoDec 检测结果



图 4 - 18 SSC-RPCA 检测结果



图 4 - 19 部分图像分割结果：(a) “fountain02” 分割结果，(b) “copyMachine” 分割结果，(c) “peopleInShade” 分割结果，(d) “library” 分割结果，(e) “boats” 分割结果，(f) “overpass” 分割结果

表 4 - 8 运行时间 (320*240)

Method	Time	Running platform
MST	10fps	Matlab
ED	70fps	C++
GMM	21fps	C++
pROST	49fps	Python/Theano and GeForce GTX 560 Ti
SMoG	8fps	Matlab
SBBS	10fps	Matlab
MBS	8fps	Matlab
DeepBS	10fps	Python and GeForce Titan X GPU
M4CD	0.2fps	Matlab
SWCD	10fps	Matlab
WeSamBE	2fps	C++
SCM	0.7fps	Matlab
ThreewayDec	0.01fps	Matlab
Decolor	0.07fps	Matlab
GoDec	0.05fps	Matlab
SSC-RPCA	0.02fps	Matlab

4.2 基于 3D 张量自编码器的子空间压缩算法

4.2.1 引言

视频压缩就是在不减少视频中重要信息的前提下，消除视频中的冗余信息，减少视频的存储空间，以便于视频的存储，传输等。常见的基于图像的视频压缩方法忽略了帧与帧之间的冗余信息，因此压缩率不高。HEVC^[106]和 H.266^[111]视频编码方式取得了非常好的视频压缩效果，并得到了广泛的使用。

近些年来，基于深度学习的方法在图像、视频压缩中也得到了广泛的使用，其中比较著名的就是自编码器，但是传统的自编码器需要存储大量的网络参数，比如当输入数据的维度的复杂度为 n ，网络参数的参数复杂度也为 $O(n)$ 。在本文中，我们引入一种叫做 3D 张量自编码器 (3DTAE) 的子空间压缩算法，通过张量网络将高维视频压缩到低维子空间中，使用低维非线性子空间来表示视频。传统的自编码器在进行视频压缩时，是将视频压缩成一个向量，而本文的 3DTAE 是将视频当做一个三阶张量来处理，直接使用网络参数来表达视频实现视频压缩，是将压缩成一个网络和一个三阶核张量，网络参数和三阶核张量的参数复杂度为 $O(n^{1/3})$ ，这样刚好满足视频压缩的需求。最后再将网络参数与核张量进行量化编码实现视频压缩。

4.2.2 张量符号与基础知识

主要符号汇总表^[150]：在本文中，张量使用黑斜体来表示，比如： χ 。矩阵和向量分别使用大写字母和小写字母来表示，除此之外，小写字母还被用来表示一些整数，比如： $i, j, k, m, n, p, r, s, x, y, z$ 。比如一个三阶张量 X ， $[X]_{i,j,k}$ 表示表示 X 第 (i,j,k) 个元。

张量 Mode- n 积^[150]：张量从数值上来说是一个多维数组，张量的阶是维数，也称为模。张量之间可以进行多种乘法运算，本文只考虑张量与矩阵的乘法。张量 $\chi \in \Re^{I_1 \times I_2 \times \dots \times I_N}$ 与矩阵 $A \in \Re^{J \times I_N}$ 的 mode- n 乘积被定义为 $\chi \times_n A$ ，进行乘积后，大小为 $I_1 \times \dots \times I_{n-1} \times J \times I_{n+1} \times \dots \times I_N$ ，结果为：

$$(\chi \times_n A)_{i_1 \dots i_{n-1} j i_{n+1} \dots i_N} = \sum_{i_n=1}^{I_n} \chi_{i_1 i_2 \dots i_N} A_{j i_n} \quad (4-28)$$

从上式可以看出，三阶张量和矩阵的乘积结果还是一个三阶张量。

高阶奇异值分解^[150]：高阶奇异值分解是一种特殊的 Tucker 分解，如图 4-20 中所示，图中表示一个三阶张量 $X \in \Re$ 可以用三个矩阵 A, B, C 和一个三阶核张量 $Y \in \Re$ 近似表示，其中 A, B, C 和 $Y \in \Re$ 的参数复杂度为 $O(R^{1/3})$ 。

$$X \approx [Y; A, B, C] = Y \times_1 A \times_2 B \times_3 C = \sum_{r=1}^R y_r a_r \circ b_r \circ c_r \quad (4-29)$$

自编码器：自编码器^[94]已经成为了一种有效的数据压缩的端到端的工具，其主要由两部分组成：编码器和解码器，传统的自编码器结构如图 4-21 所示，编码器和解码器中的节点都是全连接的，假设输入数据的维度是 n ，那么网络参数的复杂度为 $O(n)$ 。传统的自编码器工作过程如下：当压缩数据时编码器将输入的数据（向量）转换为一个短码，使用短码来近似表达输入的数据，当解压缩数据时将使用解码器将短码进行变换变得到与原始输入数据相近的数据。自编码器是一个无监督的算法，因为训练网络模型时使用了特定数据，因此其只能压缩与原始数据相同的数据，因此其泛化能力较弱。自编码器通常使用均方误差来作为目标函数，通过最小化目标函数来实现训练，如公式（4-30）。

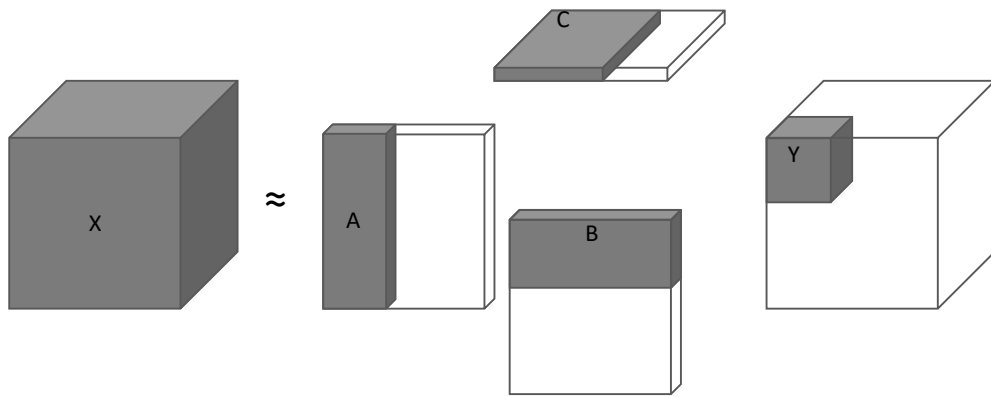


图 4 - 20 HOSVD 分解^[150]

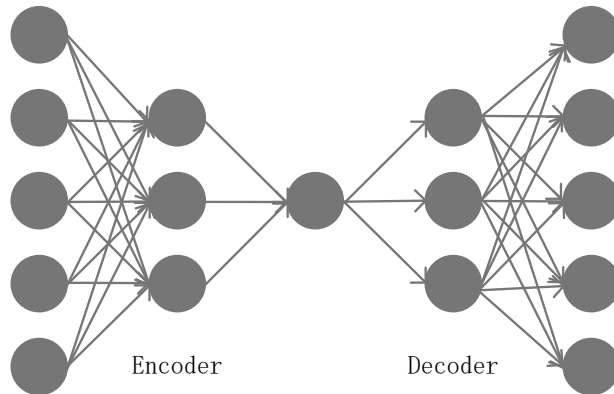


图 4 - 21 3 层全连接自编码器^[94]

$$L(d, d') = \|d - d'\|^2 = \|d - \sigma(W'(\sigma(Wd + b))) + b'\| \quad (4-30)$$

其中 d 和 d' 分别是输入和重构的输入，它们具有相同的维度， σ 为一个激活函数， W 为参数矩阵， b 为偏置。

4.2.3 3DTAE 模型

4.2.3.1 3DTAE 网络结构及其优势

与之前提到的深度神经网络的框架相比，我们的方法具有如下的优势：首先现有的基于深度神经网络的视频压缩方法具有泛化能力较弱的问题，为了克服这个问题，本文直接使用网络参数来表达视频，而不是先训练网络再将视频转换为短码。其次，本文的网络结构参数的维度小于输入数据的维度，因此可以实现视频压缩。

在我们的网络结构中，我们直接使用三维视频作为三阶张量来作为网络的输入，且没有使用全连接参数和偏置参数，而是使用三个维度较小的矩阵来连接张量，假设输入视频的维度为 n ，那么使用我们的网络压缩后的维度为 $O(n^{1/3})$ 。3DTAE 具有七层且是对称的结构，如图 4-22，它是由编码器和解码器组成。编码器负责将输入视频转换为一个短码（核张量），解码器负责使用短码来重建视频。编码器和解码器的参数是由训练得到的且在同时训练。

在我们的网络结构中，训练过程实际上也是一个压缩过程，我们不需要特定的数据来训练模型。我们直接将视频作为 3DTAE 网络的输入，接着训练这个模型得到网络参数使得误差最小，网络参数的求解的具体步骤如下。

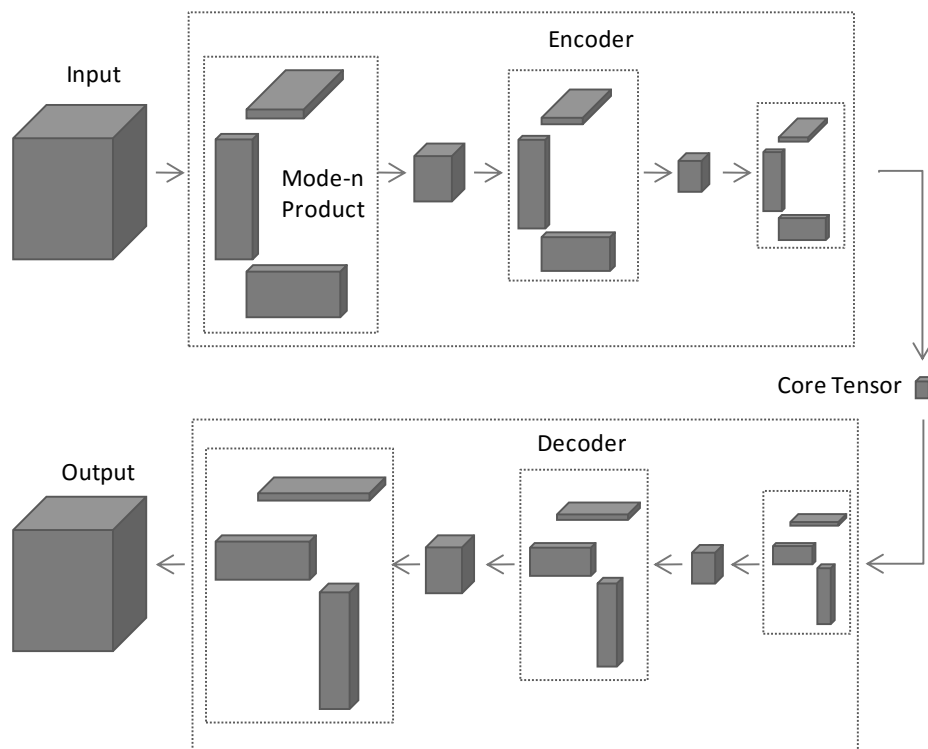


图 4 - 22 3DTAE 网络结构

4.2.3.2 损失函数

像传统的自编码器^[94]一样，本文使用均方误差作为损失函数，这也符合视频压缩的需求，使得压缩后视频与原始视频的误差最小，具体如下：

$$\begin{aligned}\mathcal{E}_{total}(y_{real}, y_{out}) &= \frac{1}{2} \|y_{real} - y_{out}\|^2 = \\ &= \frac{1}{2} \|y_{real} - \sigma(\cdots \sigma(\cdots \sigma(y_{real} \times_1^1 A \times_2^1 B \times_3^1 C) \cdots \times_1^i A \times_2^i B \times_3^i C \cdots) \times_1^6 A \times_2^6 B \times_3^6 C)\| \end{aligned} \quad (4-31)$$

其中 $1 \leq i \leq 6$, ${}_1^i A \times_2^i B \times_3^i C$ 表示第 i 个 mode-n 积。

4.2.3.3 参数初始化

我们使用 MATLAB 中 `randn` 函数来生成标准正态分布的随机数来初始化网络参数:

$$W = \alpha \times \text{randn}(h, w) \quad (4-32)$$

其中 α 设置为 0.5, h 和 w 为因子矩阵的大小。

4.2.3.4 前向传播

受到 Tucker 分解的启发, 相邻两层的张量是通过三个因子矩阵相连接的, 如图 4-20。 $A \in \mathbb{R}^{m \times r}$, $B \in \mathbb{R}^{n \times s}$, $C \in \mathbb{R}^{p \times t}$ 分别是三个因子矩阵, 它们是用来与上一层输入张量来进行 mode-1, mode-2 和 mode-3 积, 需要说明的是, 三个方向的模积具有平等的位置, 没有先后关系, 如下:

$$\hat{y} = y_{real} \times_1^1 A \times_2^1 B \times_3^1 C \quad (4-33)$$

其中 $y_{real} \in \mathbb{R}^{r \times s \times t}$ 为上一层输入张量, $\hat{y} \in \mathbb{R}^{m \times n \times p}$ 为下一层输入张量, 它们都是三阶张量。

当 mode-n 积之后, 我们将得到新的三阶张量, 然后使用 sigmoid 非线性映射函数进行映射。需要注意的是, 在映射时, 因为如果使用偏置那么就达不到数据压缩的目的, 网络参数的存储空间则会大于视频原本大小, 所以我们没有使用偏置, 如下所示:

$$f(\hat{y}) = \frac{1}{1 + e^{-\hat{y}}} = \frac{1}{1 + e^{-y_{real} \times_1^1 A \times_2^1 B \times_3^1 C}} \quad (4-34)$$

为了获得更强的网络表达能力, 我们将继续重复以上步骤 5 次。

4.2.3.5 反向传播

反向传播: 神经网络可以使用梯度下降来训练, 梯度是通过 BP 算法^[151]得到, 训练的主要目的是通过不断修正网络参数来使得损失函数最小。我们网络 BP 步骤如下所示:

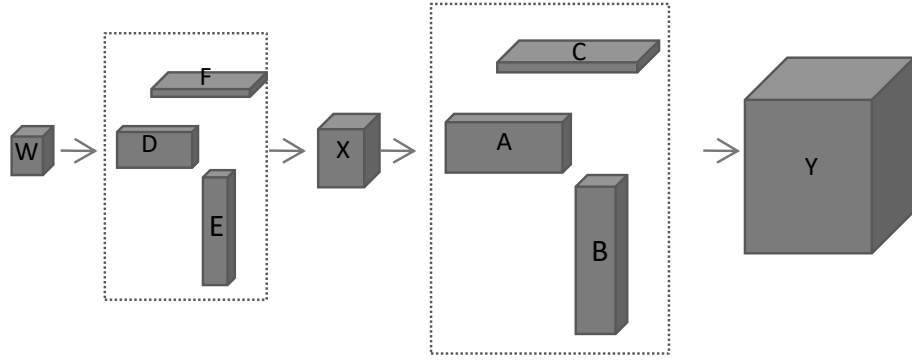


图 4 - 23 BP 说明示例

- 1) 隐含层到输出层的梯度: 如果 Y 为输出层, 那么假设 Y_{net} 和 Y_{out} 为经过激活函数前和经过激活函数后的层, 输出层和隐含层如图 4-23 所示。我们使用链式法则来获得输出层到隐含层的导数为:

$$\frac{\partial(\varepsilon_{total})_{x,y,z}}{\partial A_{x,i}} = \frac{\partial(\varepsilon_{total})_{x,y,z}}{\partial(y_{out})_{x,y,z}} \frac{\partial(y_{out})_{x,y,z}}{\partial(y_{net})_{x,y,z}} \frac{\partial(y_{net})_{x,y,z}}{A_{x,i}} \quad (4-35)$$

其中 $A \in \mathbb{R}^{m \times r}$, $Y_{net} \in \mathbb{R}^{m \times n \times p}$, $1 \leq x \leq m$, $1 \leq y \leq n$, $1 \leq z \leq p$, $1 \leq i \leq r$ 。

3DTAE 的损失如下式:

$$(\varepsilon_{total})_{x,y,z} = \frac{1}{2} \left((y_{real})_{x,y,z} - (y_{out})_{x,y,z} \right)^2 \quad (4-36)$$

其中 $(y_{real})_{x,y,z}$ 为输入矩阵, 所以:

$$\frac{\partial(\varepsilon_{total})_{x,y,z}}{\partial(y_{out})_{x,y,z}} = -((y_{real})_{x,y,z} - (y_{out})_{x,y,z}) \quad (4-37)$$

y_{out} 和 y_{net} 之间的关系可以使用如下的式子表达:

$$y_{out} = \frac{1}{1 + e^{-y_{net}}} \quad (4-38)$$

$$\frac{\partial(y_{out})_{x,y,z}}{\partial(y_{net})_{x,y,z}} = \partial(y_{out})_{x,y,z} (1 - \partial(y_{out})_{x,y,z}) \quad (4-39)$$

y_{net} 和 $A_{x,i}$ 之间的关系可以使用如下的式子表达

$$(y_{net})_{x,y,z} = \sum_{i=1}^r \sum_{j=1}^s \sum_{k=1}^t A_{x,i} B_{y,j} C_{z,k} (\chi_{out})_{i,j,k} \quad (4-40)$$

其中 $B \in \mathbb{R}^{n \times s}$, $C \in \mathbb{R}^{p \times t}$, $1 \leq j \leq s$, $1 \leq k \leq t$, $X_{net} \in \mathbb{R}^{r \times s \times t}$ 。

$$\frac{\partial(y_{net})_{x,y,z}}{\partial A_{x,i}} = \sum_{j=1}^s \sum_{k=1}^t B_{y,j} C_{z,k} (\chi_{out})_{i,j,k} \quad (4-41)$$

相同地,

$$\frac{\partial(\varepsilon_{total})_{x,y,z}}{\partial B_{y,j}} = \sum_{i=1}^r \sum_{k=1}^t \frac{\partial(\varepsilon_{total})_{x,y,z}}{\partial(y_{out})_{x,y,z}} \frac{\partial(y_{out})_{x,y,z}}{\partial(y_{net})_{x,y,z}} \frac{\partial(y_{net})_{x,y,z}}{\partial B_{y,j}} \quad (4-42)$$

$$\frac{\partial(\varepsilon_{total})_{x,y,z}}{\partial C_{z,k}} = \sum_{i=1}^r \sum_{j=1}^s \frac{\partial(\varepsilon_{total})_{x,y,z}}{\partial(y_{out})_{x,y,z}} \frac{\partial(y_{out})_{x,y,z}}{\partial(y_{net})_{x,y,z}} \frac{\partial(y_{net})_{x,y,z}}{\partial C_{z,k}} \quad (4-43)$$

2) 隐含层到隐含层的梯度: 隐含层到隐含层的梯度和隐含层到输出层的梯度计算方式有不同之处, 如下式:

$$\frac{\partial(\varepsilon_{total})_{x,y,z}}{\partial D_{i,a}} = \sum_{i=1}^r \sum_{j=1}^s \frac{\partial(\varepsilon_{total})_{x,y,z}}{\partial(\chi_{out})_{i,j,k}} \frac{\partial(\chi_{out})_{i,j,k}}{\partial(\chi_{net})_{i,j,k}} \frac{\partial(\delta_{net})_{i,j,k}}{\partial D_{i,a}} \quad (4-44)$$

其中 $D \in \mathbb{R}^{r \times u}$, $1 \leq \alpha \leq \mu$ 。

$$\frac{\partial(\varepsilon_{total})_{x,y,z}}{\partial(\chi_{out})_{i,j,k}} = \frac{\partial(\varepsilon_{total})_{x,y,z}}{\partial(y_{out})_{x,y,z}} \frac{\partial(y_{out})_{x,y,z}}{\partial(y_{net})_{x,y,z}} \frac{\partial(\delta_{net})_{i,j,k}}{\partial(\chi_{out})_{i,j,k}} \quad (4-45)$$

其中:

$$\frac{\partial(y_{net})_{x,y,z}}{\partial(\chi_{out})_{i,j,k}} = D_{x,i} E_{y,j} F_{z,k} \quad (4-46)$$

其中 $E \in \mathbb{R}^{s \times v}$, $F \in \mathbb{R}^{t \times w}$, $1 \leq b \leq v$, $1 \leq c \leq w$ 。

同样地, 我们有:

$$\frac{\partial(\chi_{out})_{i,j,k}}{\partial(\chi_{net})_{i,j,k}} = (\chi_{out})_{i,j,k} (1 - (\chi_{out})_{i,j,k}) \quad (4-47)$$

$$\frac{\partial(\chi_{net})_{i,j,k}}{\partial D_{i,a}} = \sum_{b=1}^v \sum_{c=1}^w E_{j,b} F_{k,c} (W_{out})_{a,b,c} \quad (4-48)$$

其中 $W_{out} \in \mathbb{R}^{\mu \times v \times w}$

相同地, 我们可以得到:

$$\frac{\partial(\varepsilon_{total})_{x,y,z}}{\partial E_{j,b}} = \sum_{a=1}^u \sum_{c=1}^w \frac{\partial(\varepsilon_{total})_{x,y,z}}{\partial(\chi_{out})_{i,j,k}} \frac{\partial(\chi_{out})_{i,j,k}}{\partial(\chi_{net})_{i,j,k}} \frac{\partial(\chi_{net})_{i,j,k}}{\partial E_{j,b}} \quad (4-49)$$

$$\frac{\partial(\varepsilon_{total})_{x,y,z}}{\partial F_{k,c}} = \sum_{a=1}^u \sum_{b=1}^v \frac{\partial(\varepsilon_{total})_{x,y,z}}{\partial(\chi_{out})_{i,j,k}} \frac{\partial(\chi_{out})_{i,j,k}}{\partial(\chi_{net})_{i,j,k}} \frac{\partial(\chi_{net})_{i,j,k}}{\partial F_{k,c}} \quad (4-50)$$

4.2.3.6 参数更新

由上面的步骤我们可以得到更新参数所需要的梯度信息，在本文中我们使用 Adam^[152]来更新优化参数，具体如下：

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \quad (4-51)$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \quad (4-52)$$

$$w_t = w_{t-1} - \alpha \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \varepsilon} \quad (4-53)$$

其中 m_t 为第一时刻的梯度均值， v_t 为第二时刻的梯度非中心方差值， α 为学习步长，本文设置为 0.001， β_1 和 β_2 分别设置为 0.9，0.999， ε 设置为 10^{-8} 。

4.2.3.7 模型优化

假设 3DTAE 的输入的张量为 $X \in \mathbb{R}^{n \times n \times n}$ ，那么前向传播的时间复杂度为 $O(n^4)$ 。在反向传播中，输出层到隐含层的复杂度为 $O(n^6)$ ，隐含层到隐含层的时间复杂度为 $O(n^9)$ ，因此 3DTAE 的模型参数更新的时间复杂度非常高。为了减少时间复杂度，受到随机梯度下降的启发，本文使用随机采样，选择部分的“点”来更新整个模型，而不是选择所有的“点”。其中，公式 (4-41) 的时间复杂度为 $O(n^2)$ ，我们随机地从矩阵 B 和 C 中选择第 r 列来更新模型，那么时间复杂度为 $O(m)$ 。相同地，公式 (4-46) 的复杂度为 $O(n^3)$ ，我们随机地从 Y 张量中选取 s 个点，那么公式 (4-46) 的复杂度则变为 $O(s)$ ，经过优化，在一次模型更新时的时间复杂度为 $O(n^6)$ 。

4.2.3.8 模型结构自动生成

本文使用 HOSVD^[153]来进行模型结构的自动生成，HOSVD 决定了张量分解的最佳秩。假设输入的张量为 $Y \in \mathbb{R}^{m \times n \times p}$ ，经过 HOSVD 分解后最佳秩为 (r_1, r_2, r_3) 。自动生成的结构如公式 (4-54) 所示。其中 λ 的参数范围是 $[0.6, 0.9]$ ，其决定了 3DTAE 的映射能力，值越大映射能力越强，视频的质量越高。3DTAE 的每一层都是一个三阶张量，相邻的三阶张量是通过三个因子矩阵连接，因此一共有 6×3 个因子矩阵，对应的大小如公式 (4-55) 所示。

$$\begin{bmatrix} m \lambda r_1 & \lambda^2 r_1 & \lambda^3 r_1 & \lambda^2 r_1 & \lambda r_1 & m \\ n \lambda r_2 & \lambda^2 r_2 & \lambda^3 r_2 & \lambda^2 r_2 & \lambda r_2 & n \\ p \lambda r_3 & \lambda^2 r_3 & \lambda^3 r_3 & \lambda^2 r_3 & \lambda r_3 & p \end{bmatrix} \quad (4-54)$$

$$\begin{bmatrix} \lambda r_1 \times m & \lambda r_2 \times n & \lambda r_3 \times p \\ \lambda^2 r_1 \times \lambda r_1 & \lambda^2 r_2 \times \lambda r_2 & \lambda^2 r_3 \times \lambda r_3 \\ \lambda^3 r_1 \times \lambda^2 r_1 & \lambda^3 r_2 \times \lambda^2 r_2 & \lambda^3 r_3 \times \lambda^2 r_3 \\ \lambda^2 r_1 \times \lambda^3 r_1 & \lambda^2 r_2 \times \lambda^3 r_2 & \lambda^2 r_3 \times \lambda^3 r_3 \\ \lambda r_1 \times \lambda^2 r_1 & \lambda r_2 \times \lambda^2 r_2 & \lambda r_3 \times \lambda^2 r_3 \\ m \times \lambda r_1 & n \times \lambda r_2 & p \times \lambda r_3 \end{bmatrix} \quad (4-55)$$

4.2.3.9 压缩与解压缩视频

视频的压缩步骤在算法 4-4，解压缩步骤在算法 4-5 中。

算法 4 - 4 视频压缩步骤

输入：原始未压缩的视频

输出：压缩后的视频

1. 对原始视频进行 RGB 三通道分离得到三个三阶张量
 2. 使用 3DTAE 算法对每个三阶张量进行压缩
 3. 重复以上步骤压缩所有三阶张量
 4. 量化和编码所有网络的解码器和核张量得到压缩视频
-

算法 4 - 5 视频解压缩步骤

输入：压缩视频

输出：解压后视频

1. 使用反编码和反量化得到量化的解码器和核张量
 2. 反量化得到所有解码器和核张量
 3. 使用张量 mode-n 积得到每个通道的三阶张量
 4. 使用所有三通道张量重构视频
-

4.2.4 视频压缩实验

4.2.4.1 实验设计

平台：本文所有实验都是在一个拥有志强 E5-2620 2.1GHZ 的 CPU 和 128GB 内存的服务器上运行的，编程环境为 MATLAB R2015b，Windows Server 2012，CUDA 7.5 和 TESLA K80 GPU。

对比方法：CAE^[98]、HEVC^[106]和 H.266^[111]，其中 HEVC 对应的软件是

HM-16.20^[154], H.266 对应的软件为 VTM-3.0^[111]。

数据库: 本文选取了标准的视频序列用于测试 3DTAE 的有效性, 因为本文提出的 3DATE 模型的时间复杂度很高, 所以本文只测试了 QCIF 格式视频, 测试视频共 8 个, 分别为 akiyo、bridge_close、bridge_far、claire、grandma、hall、mother_daughter 和 news, 每个视频取其前 300 帧进行测试, 帧率设置为 50。

超参数设置: 本文使用 Adam^[152] 来更新参数, 学习率设置为 1^{-8} , 迭代次数上限设置为 10^5 。3DTAE 模型中, 参数 λ 分别设置为 0.80, 0.85, 0.90 和 0.95。HM-16.20 软件中参数 QP 分别设置为 32, 35, 38 和 42。VTM-3.0 软件中参数 QP 分别设置为 29, 32, 35 和 38。CAE 模型中 iteration 分别设置为 0, 2, 5, 8 和 11。

评价指标: 本文选取了 Rate-Distortion 曲线, BD-Rate-PSNR, BD-Rate-SSMI, BP-PSNR 和 BD-SSIM^[155] 来进行结果评价, 除此之外还有编码和解码时间。PSNR 的计算公式如下:

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i, j) - J(i, j)]^2 \quad (4-56)$$

$$PSNR = 10 \log_{10} \left(\frac{255^2}{MSE} \right) \quad (4-57)$$

其中 I, J 分别为原始图像和压缩后图像, PSNR 的值越大说明图像的质量越好。

SSIM 的计算步骤如下:

$$l(x, y) = \frac{2\mu_x \mu_y + c_1}{\mu_x^2 + \mu_y^2 + c_1} \quad (4-58)$$

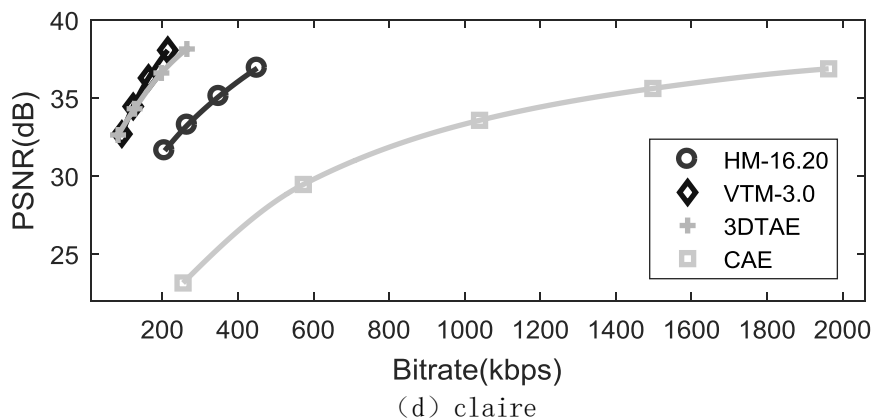
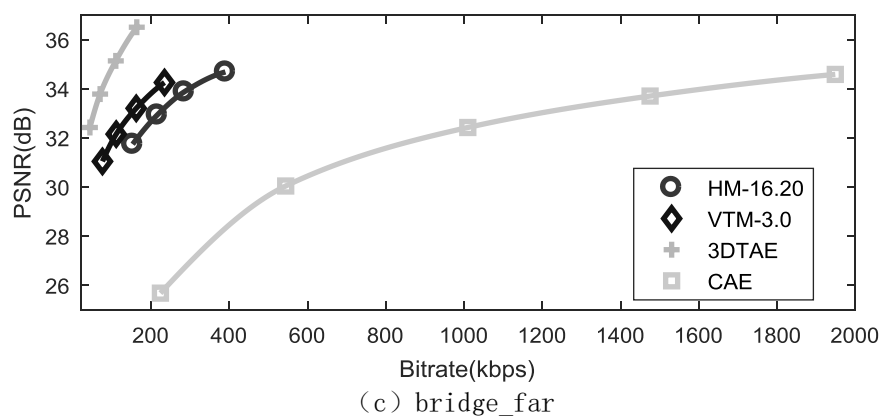
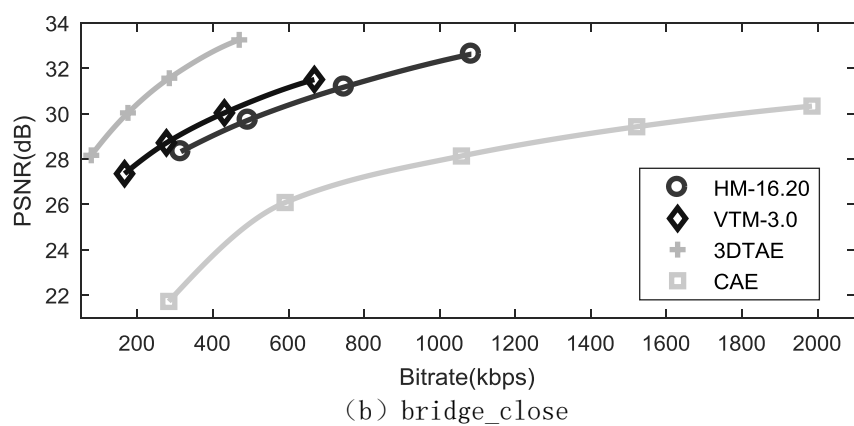
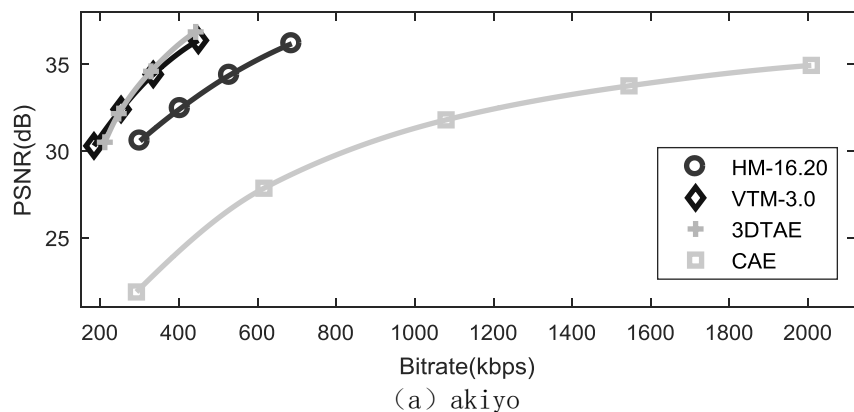
$$c(x, y) = \frac{2\sigma_x \sigma_y + c_2}{\sigma_x^2 + \sigma_y^2 + c_2} \quad (4-59)$$

$$s(x, y) = \frac{\sigma_{xy} + c_3}{\sigma_x \sigma_y + c_3} \quad (4-60)$$

$$SSIM(x, y) = [l(x, y)^\alpha \cdot c(x, y)^\beta \cdot s(x, y)^\gamma] \quad (4-61)$$

其中 μ_x 和 μ_y 分别为图像 x 和图像 y 的均值, σ_x^2 和 σ_y^2 分别是图像 x 和图像 y 的方差, σ_{xy} 是图像 x 和图像 y 的协方差, c_1 、 c_2 和 c_3 分别为常数。 α 、 β 和 γ 通常设置为 1。

4.2.4.2 结果与分析



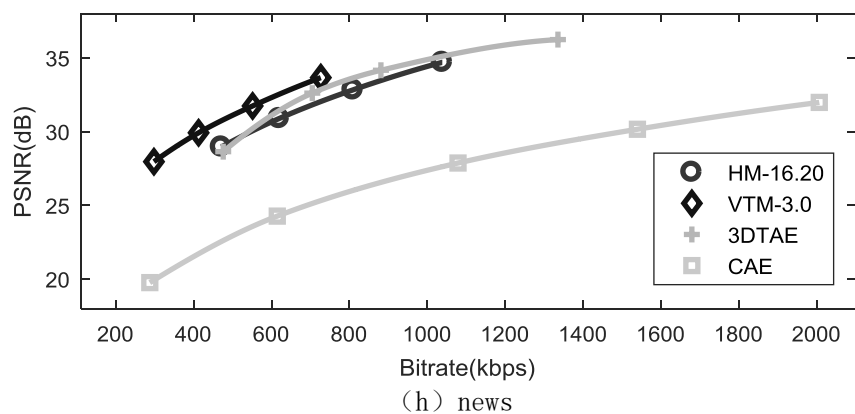
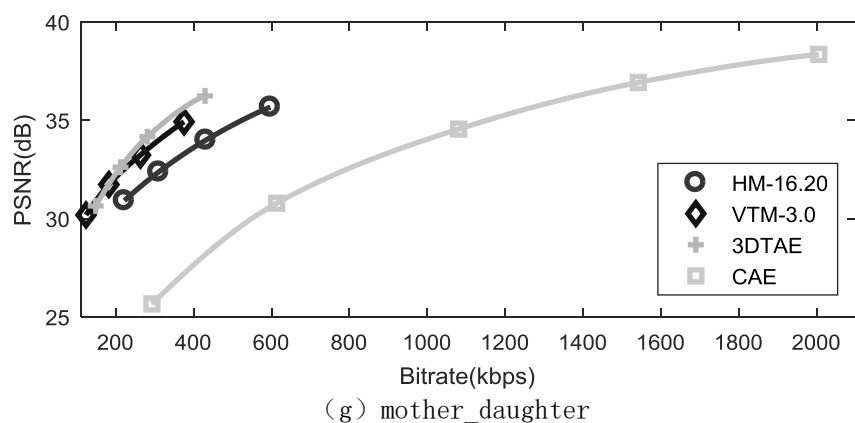
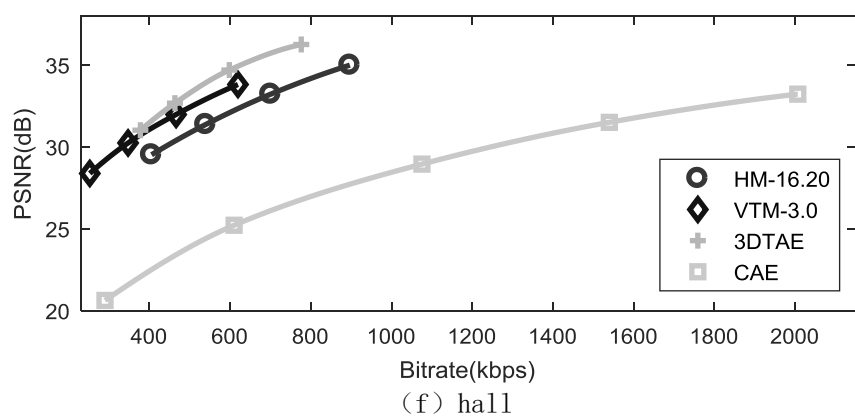
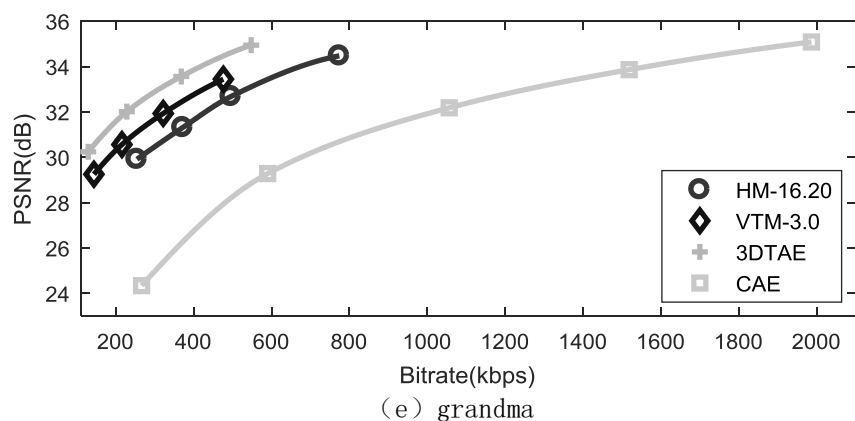
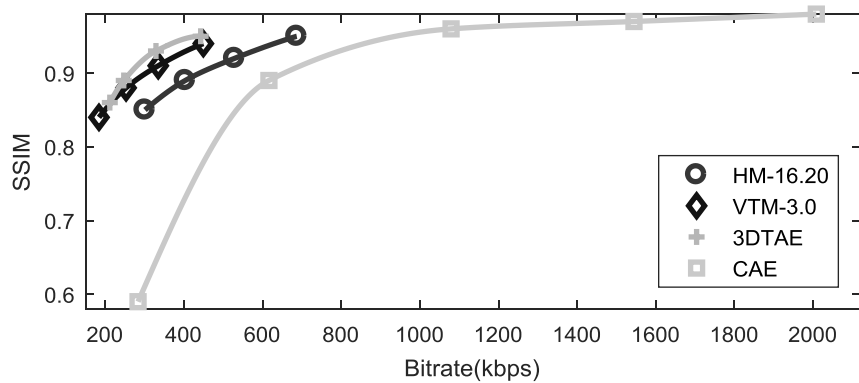
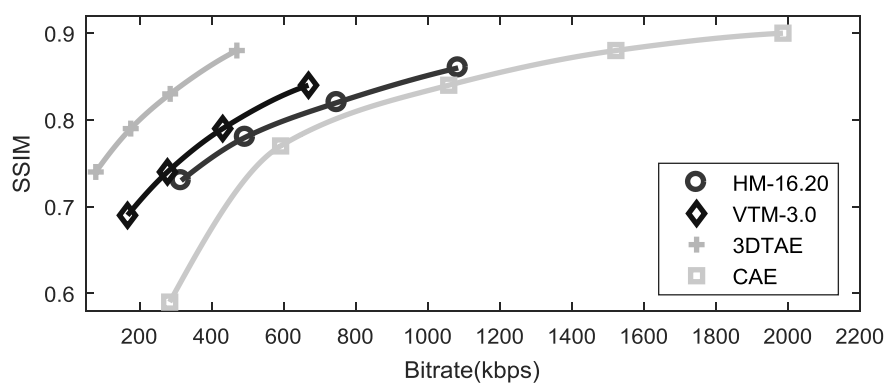


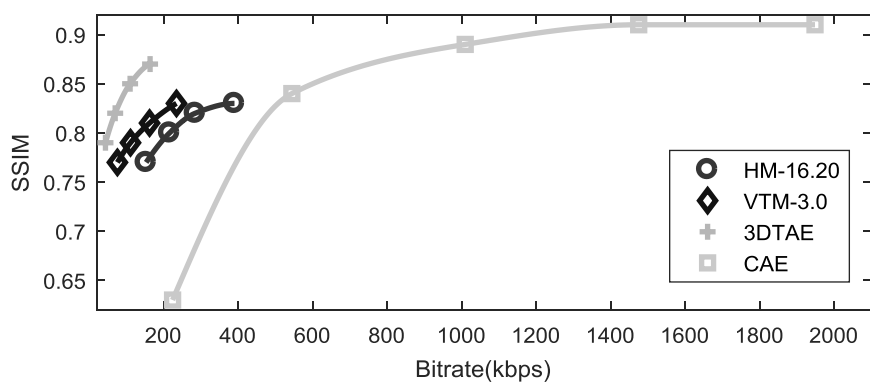
图 4 - 24 视频率失真曲线



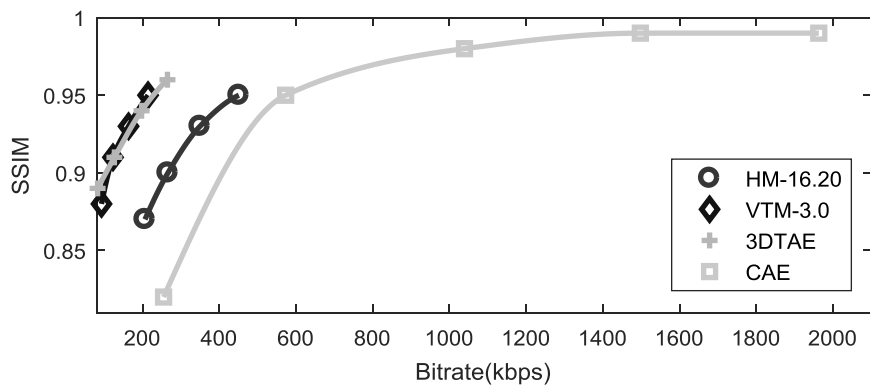
(a) akiyo



(b) bridge_close



(c) bridge_far



(d) claire

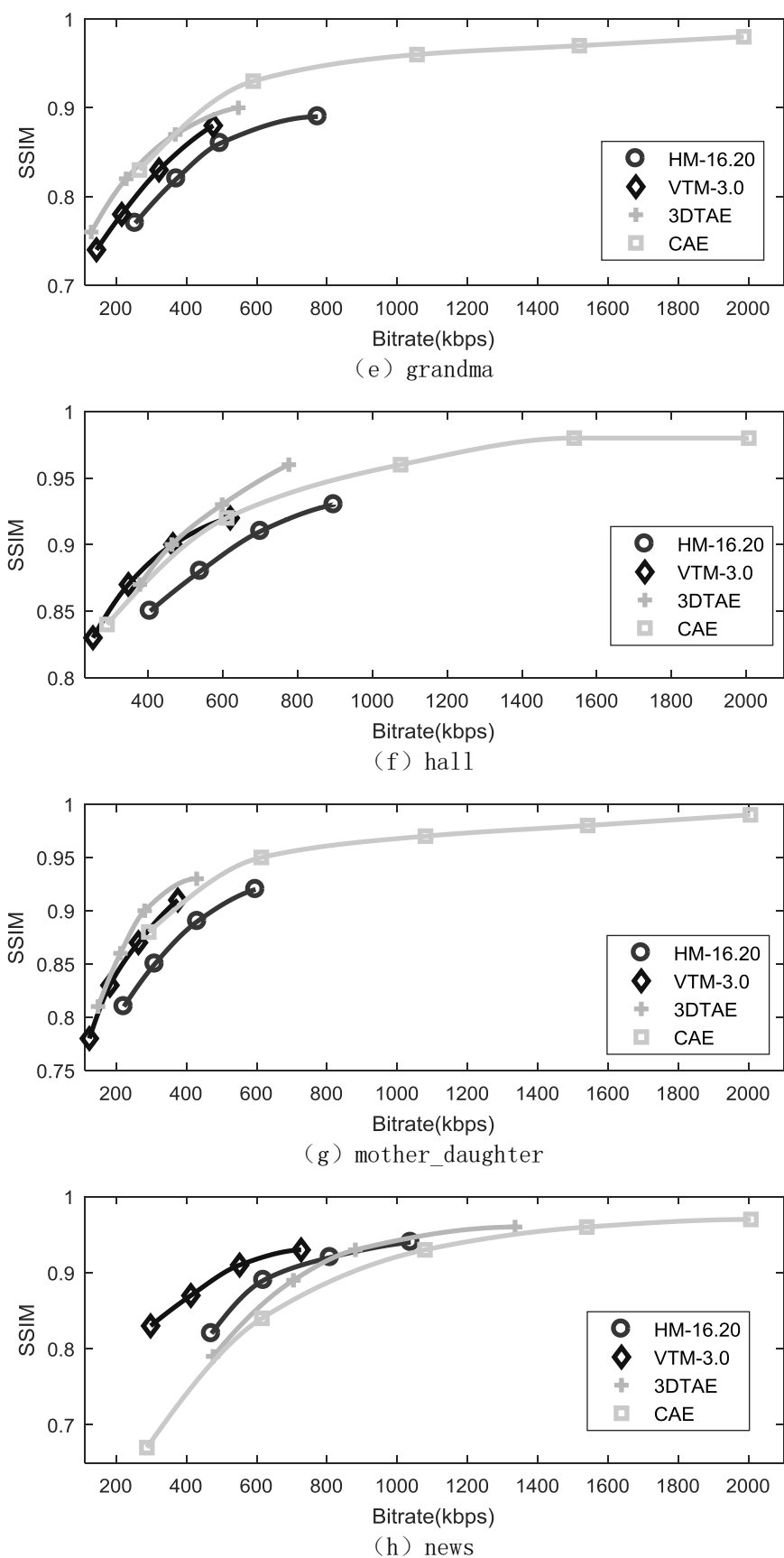


图 4 - 25 视频率失真曲线

表 4 - 9 BD-PSNR (dB)

Video	Vs.	CAE	HM-16.20	VTM-3.0
akiyo	dB	11.68	3.59	0.15
bridge close	dB	8.92	3.65	2.75
bridge far	dB	12.38	4.52	3.04
claire	dB	14.73	5.07	-0.44
grandma	dB	6.98	2.22	1.21
hall	dB	9.38	2.42	0.78
mother_daughter	dB	8.37	2.24	0.36
news	dB	7.22	0.52	-1.26
Avg	dB	9.98	3.02	0.82

表 4 - 10 BD-RATE-PSNR (%)

Video	Vs.	CAE	HM-16.20	VTM-3.0
akiyo	%	-79.42	-37.55	-1.20
bridge close	%	-91.23	-67.66	-60.48
bridge far	%	-94.45	-75.42	-64.82
claire	%	-89.46	-58.90	8.83
grandma	%	-76.6	-45.68	-30.67
hall	%	-74.51	-28.87	-9.98
mother_daughter	%	-72.72	-35.21	-6.24
news	%	-6.65	15.75	58.50
Avg	%	-73.38	-41.69	-13.28

表 4 - 11 BD-SSIM

Video	Vs.	CAE	HM-16.20	VTM-3.0
akiyo	SSIM	0.23	0.06	0.01
bridge close	SSIM	0.19	0.10	0.08
bridge far	SSIM	0.30	0.09	0.05
claire	SSIM	0.13	0.06	-0.01
grandma	SSIM	0.01	0.04	0.02
hall	SSIM	0.01	0.03	0.01
mother_daughter	SSIM	0.01	0.05	0.01
news	SSIM	0.02	-0.01	-0.06
Avg	SSIM	0.11	0.05	0.01

从图 4-24 和图 4-25 中给出了 Rate-distortion 曲线,从中可以看出 3DTAE 明显地相对于 VTM-3.0, HM-16.20 和 CAE 提高了编码效率。编码能力具体提高的信息在表 4-9、表 4-10、表 4-11 和表 4-12 中,如 BD-PSNR、BD-Rate-PSNR、BD-SSIM 和 BD-Rate-SSIM。在相同的峰值信噪比 PSNR 下,相对于 CAE、

HM-16.20 和 VTM-3.0, 3DTAE 分别提高了 73.38%、41.69% 和 13.28% 的编码效率。在相同的比特率下, 相对于 CAE、HM-16.20 和 VTM-3.0, 3DTAE 分别提高了 9.98dB、3.02dB 和 0.82dB。在相同的 SSIM 下, 相对于 CAE、HM-16.20 和 VTM-3.0, 3DTAE 分别提高了 37.57%、40.85% 和 13.47% 的编码效率。在相同的比特率下, 相对于 CAE、HM-16.20 和 VTM-3.0, 3DTAE 分别提高了 0.11、0.05 和 0.01。3DTAE 使用 mode-n 积来代替常见的自编码器全连接参数。假设输入数据的维度为 n , 3DTAE 参数的复杂为 $O(n^{1/3})$, 即压缩后的视频大小为 $O(n^{1/3})$ 。这是因为 3DTAE 是一个拥有多层网络的结构, 具有更强的非线性映射能力。图 4-26 给出了 3DTAE 在视频“akiyo”中 R 通道张量的迭代损失图, 从中可以看出, 随着迭代次数的增加, 目标函数的损失值不断减小, 说明了本文的模型优化方法是有效的。

表 4 - 12 BD-RATE-SSIM (%)

Video	Vs.	CAE	HM-16.20	VTM-3.0
akiyo	%	-58.12	-39.73	-12.83
bridge close	%	-74.05	-68.58	-60.75
bridge far	%	-85.30	-76.14	-60.93
claire	%	-79.11	-55.01	1.59
grandma	%	8.09	-38.24	-22.56
hall	%	-12.88	-27.78	0.21
mother_daughter	%	7.41	-37.14	-11.06
news	%	-6.65	15.75	58.50
Avg	%	-37.57	-40.85	-13.47

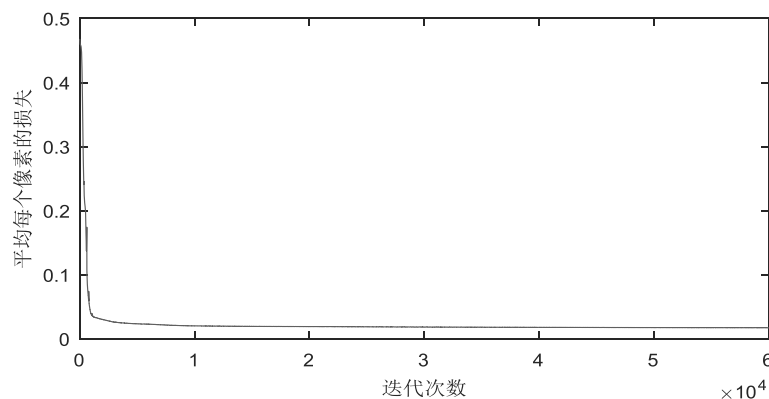


图 4 - 26 akiyo 视频张量的迭代损失曲线

表 4 - 13 编码解码时间信息 (单位: 秒)

平均时间	VAE	HM-16.20	VVC-3.0	3DTAE
编码	2.3	341.05	7241.72	15390.43
解码	1.8	6.15	8.38	0.36

4.3 总结

在本章中,为了克服 RPCA 在不能有效地处理动态背景下缓慢运动物体的运动目标检测任务,我们提出了一种基于视频分割和显著约束的 RPCA 子空间恢复算法。我们首先使用超像素分割方法得到图像序列的超像素块,然后再将图像中的超像素块进行合并,接着再将相邻图像块进行合并,最终完成了视频分割。属于同一个编号的像素具有相同的特征,要么都为运动目标要么都为背景信息。我们将视频分割信息融入到 RPCA 中得到了 SC-RPCA 中,其效果要明显优于 RPCA 的结果,但是其仍有不足之处,比如不能较好地处理缓慢运动的情况,受到其他学者研究的启发,本文将图像的显著信息融入到 SC-RPCA 模型中,得到了 SSC-RPCA 模型,本文的显著信息不需要提前计算,其是在 SSC-RPCA 模型求解时自动获得。最后我们将 SSC-RPCA 与主流的方法在 CDNET 2014 公共数据集上进行了实验,SSC-RPCA 取得了良好的效果。但是 SSC-RPCA 仍有不足之处,比如其时间复杂度较高,需要继续研究优化。除此之外,在进行超像素块合并时,有时会产生合并错误,将前景和背景混淆。

视频压缩是人工智能领域重要且基础的任务之一。在本章中,我们提出了一种 3D 张量子空间压缩方法,其主要是基于自编码器和 HOSVD 分解,此方法将视频表示为一个低维的非线性子空间。3DTAE 主要克服了两个问题,第一个是传统深度自编码器的泛化问题,在 3DTAE 中,我们不要使用特定的数据训练网络,我们直接将视频作为输入,使用网络参数来表达视频实现视频的压缩,所以 3DTAE 模型不存在泛化的问题。第二个问题是传统的深度自编码器参数复杂度和视频数据的复杂度相同,即假设传统自编码器的输入数据的维度是 n ,而自编码器的网络参数的复杂度也为 $O(n)$,因此其需要存储大量的网络参数,压缩性能有待进一步提高。而我们提出的 3DTAE 模型使用视频直接作为输入,然后再使用 mode- n 积来代替全连接层,这样网络的参数数量大大减少,假设输入数据的维度为 n ,3DTAE 参数的复杂为 $O(n^{1/3})$,即压缩后的视频大小为 $O(n^{1/3})$ 。3DTAE 在三个方向使用因子矩阵对数据从高维向低维进行投影消除冗余信息。

第五章 总结与展望

5.1 总结

子空间学习是处理视频的一种有效的工具之一,本文主要在子空间学习在视频处理中的三个方向进行了研究:子空间分割、子空间恢复和子空间压缩,本文的主要工作内容如下:

第一:针对当前基于谱聚类的子空间分割方法(SSC、LRR等)不能满足较大数据点的分割,基于RANSAC的子空间分割方法,随着数据维度的增加,时间复杂度呈指数级上升,基于K-subspace的子空间分割方法容易陷入局部最优值。为了得到更快速的子空间分割方法,本文提出了一种名叫随机探测(RANSP)的快速子空间分割方法。本文方法分为两种情况,第一种当数据中没有噪声时,我们将子空间分割转换为 L_2 最小化问题,如果有 k 个子空间需要分割,那么我们只需要进行 $k-1$ 次分割;第二种情况是当数据中有噪声时,通过随机选择种子点,并使用岭回归计算其它点与这个点的相关性,这样便可以得到一个子空间,通过多次重复便可以实现子空间分割,且RANSP的时间复杂度为线性,因此可以对大规模数据点进行处理。通过实验验证,RNASP运算时间仅仅是当前最快的子空间分割方法K-subspace十分之一。

第二:为了提高RPCA模型的子空间恢复能力,我们提出了一种截断加权的RPCA模型(TWRPCA)。大的奇异值保存着重要的数据方向信息,而小的奇异值被认为是稀疏噪声,在TWRPCA中,我们保留前10%的奇异值不变,其它的奇异值分别给予不同的权重。原则是:大的奇异值拥有大的权重,小的奇异值拥有较小的权重。最后再使用IALM对TWRPCA模型进行了求解。实验结果表明,TWRPCA在被椒盐噪声污染的数据上进行子空间恢复时,比DWLP和RPCA具有更好的鲁棒性,

第三:当前基于RPCA子空间学习的方法及其扩展的模型不能较好地处理运动目标缓慢的情况,尤其是在复杂背景下,如晃动的树叶、波动的湖面等。为了能够在复杂环境中检测缓慢的运动目标,本文将视频分割和显著性检测融入到RPCA模型中,提出了一种基于分割和显著约束的RPCA子空间分割算法(SSC-RPCA),并应用到运动目标检测任务中。在公共数据集CDNET 2014上进行了实验验证。实验表明,本文的方法比当前基于RPCA的运动目标检测模型具有更高的准确率。

第四：针对传统的自编码器在进行数据压缩时不单单要存储压缩后的短码，而且需要存储大量的网络参数，比如，当输入数据的维度是 $O(n)$ ，那么网络参数的维度也是 $O(n)$ 。本文提出了一种 3 阶张量自编码器的子空间压缩算法，高维视频被 3DTAE 模型压缩到低维的子空间中。在传统的自编码器中，视频被表示为网络参数和向量。在本文方法中，视频被直接表示为网络参数，且网络参数是由一些二维矩阵和一个核张量组成，假设输入数据的维度是 $O(n)$ ，那么网络参数的维度也是 $O(n^{1/3})$ ，这样便可以满足视频压缩的需求。通过实验验证，同等 PSNR 下，3DTAE 比 HM-16.20 和 VTM-3.0 分别降低了 41.69% 和 13.28% 的编码比特数。

5.2 展望

在未来的工作中，我将继续探讨子空间学习在视频处理中的应用。以后的工作将主要从以下几个方面进行展开：

本文的 RANSP 方法虽然能够处理大规模数据点的分割，主要是针对数据点的维度较小的情况，这样便于快速子空间分割，然而当数据点具有高维特征时，RANSP 的时间复杂度则快速增加，具体为，当数据的高维特征为 n ，则优化后的求解复杂度为 $O(n^3)$ 。因此未来我们将探讨大规模具有高维特征的数据点的子空间分割，主要在高维特征上降维做研究。

本文的 TWRPCA 模型虽然在椒盐噪声上效果优于 RPCA 模型和 DWLP 模型；但是在高斯噪声上，TWRPCA 模型恢复出的误差要高于 DWLP 模型。未来我们将继续优化 TWRPCA 模型，比如增加正则项用于针对高斯噪声，比如目前已经存在的模型： $D = A + E + S$ ，其中 A 为低秩部分，代表数据的低维子空间， E 为稀疏部分，代表大的离群点，比如椒盐噪声， S 代表较弱的噪声，比如高斯噪声，我们将主要对这三项进行不同策略的约束来提高子空间恢复的能力。除此之外，我们还要将此模型应用到图像去噪中。

本文的 SSC-RPCA 模型虽然检测正确率较高，但当前景目标与背景颜色相近时，则会出现误检，未来打算引入图像语义分割到模型中。而且 SSC-RPCA 模型的时间复杂度较高，不能完成实时的运动目标检测任务，这也是未来工作的重点。除此之外，SSC-RPCA 模型采取的是批量图片检测策略，未来打算将增量 RPCA 引入到 SSC-RPCA 模型中，用来配合实时运动目标检测。

本文的 3DTAE 模型速度较慢，而且模型所需要的内存也特别大，未能测试其在高清视频压缩任务中的性能，未来我们将重点研究加快模型的优化，重点分为两个方面，一方面是加快模型运行速度，另一方面是优化模型所需的存储空间。

间，这样便于未来测试高清视频序列。我们将打算将基于随机探测的快速子空间算法应用在视频压缩中去，主要分为两个方向，第一个方向 3DTAE 方法是将一个视频压缩到网络中，未来我们将多个视频压缩到同一个网络中，那么这个时候就需要第三章的方法对视频进行分类，将相近的视频分类到同一个类中，对同一个类中的视频进行压缩，压缩到同一个网络中，第二个方向是目前 3DTAE 方法的运行速度非常慢以至于不能测试高清视频，未来我们打算对图像的每一帧进行分块，然后使用第三章中的学习方法对图像块进行聚类，这样再使用 3DTAE 方法对每个类的图像块进行压缩。除此之外还需要设计视频传输方案。

参考文献

- [1] 黄松, 李燕林, 戴平娟. 智慧旅游城市旅游竞争力评价[J]. 地理学报, 2017, 72(2):242 - 255.
- [2] 陈能成, 刘迎冰, 盛浩. 智慧城市时空信息综合决策关键技术与系统[J]. 武汉大学学报:信息科学版, 2018, 43(12):2278 - 2286.
- [3] Li Y, Dai W, Zou J, et al. Scalable structured compressive video sampling with hierarchical subspace learning[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2019, 12(2):87 - 103.
- [4] 吕红伟, 王士同. 基于RPCA对高维数据子空间聚类的预测方法[J]. 计算机工程与科学, 2017, 39(3):553 - 561.
- [5] Fang X, Teng S, Lai Z, et al. Robust latent subspace learning for image classification[J]. IEEE Transactions on Neural Networks, 2018, 29(6):2502 - 2515.
- [6] Zheng Q, Wang Y, Heng P. Online subspace learning from gradient orientations for robust image alignment[J]. IEEE Transactions on Image Processing, 2019, 28(7):3383 - 3394.
- [7] 李小薪, 梁荣华. 有遮挡人脸识别综述从子空间回归到深度学习[J]. 计算机学报, 2018, 41(1):177 - 207.
- [8] Xia G, Sun H, Feng L, et al. Human motion segmentation via robust kernel sparse subspace clustering[J]. IEEE Transactions on Image Processing, 2018, 27(1):135 - 150.
- [9] 谢寒, 蒋阳升, 姚志洪, 等人. 多级分块的交通视频智能识别背景建模方法[J]. 哈尔滨工业大学学报, 2017, 49(9):40 - 46.
- [10] Chen Z, Wang R, Zhang Z, et al. Background - foreground interaction for moving object detection in dynamic scenes[J]. Information Sciences, 2019, 483:65 - 81.
- [11] 鲁志红, 郭丹, 汪萌. 结合帧率变换与 HEVC 标准的新型视频压缩编码算法[J]. 自动化学报, 2018, 44(9):92 - 102.
- [12] Hussain A J, Ahmed Z. A survey on video compression fast block matching algorithms[J]. Neurocomputing, 2019, 335:215 - 237.
- [13] Fischler M A, Bolles R C. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography[J]. Communications of the ACM, 1981, 24(6):381 - 395.
- [14] Ho J, Yang M H, Lim J, et al. Clustering appearances of objects

- under varying illumination conditions[C]. Computer Vision and Pattern Recognition. Madison, Wisconsin, USA, June 16-22, 2003, 1:11 - 18.
- [15] Elhamifar E, Vidal R. Sparse subspace clustering[C]. Computer Vision and Pattern Recognition. Miami, Florida, USA, June 20-25, 2009:2790 - 2797.
- [16] Wright J, Ganesh A, Rao S, et al. Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization[C]. Advances in Neural Information Processing Systems, Vancouver, Canada, December 10-15, 2009:2080 - 2088.
- [17] Sim T, Baker S, Bsat M. The CMU pose, illumination, and expression (PIE) database[C]. Automatic Face and Gesture Recognition, Washington, D.C., USA, May 20-22, 2002:53 - 58.
- [18] Xiao H, Rasul K, Vollgraf R. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms[J]. arXiv:1708.07747, 2017:1 - 6.
- [19] Wang Y, Jodoin P, Porikli F, et al. CDnet 2014: An expanded change detection benchmark dataset[C]. IEEE Conference on Computer Vision and Pattern Recognition Workshops, Columbus, OH, USA, June 23-28, 2014:393 - 400.
- [20] Krishna K, Murty M N. Genetic K-means algorithm[J]. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 1999, 29(3):433 - 439.
- [21] Vidal R, Ma Y, Sastry S. Generalized principal component analysis (GPCA)[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2005, 27(12):1945 - 1959.
- [22] Liu G, Lin Z, Yu Y. Robust subspace segmentation by low-rank representation[C]. Proceedings of the 27th international conference on machine learning (ICML-10), Haifa, Israel, June 21-24, 2010:663 - 670.
- [23] Lu C Y, Min H, Zhao Z Q, et al. Robust and efficient subspace segmentation via least squares regression[J]. European Conference on Computer Vision, Firenze, Italy, October 7-13, 2012:347 - 360.
- [24] Shi J, Malik J. Normalized cuts and image segmentation[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2000, 22(8):888 - 905.
- [25] Liu J, Cheng Y, Wang X, et al. Low rank subspace clustering via discrete constraint and hypergraph regularization for tumor molecular pattern discovery[J]. IEEE/ACM Transactions on Computational Biology and Bioinformatics, 2018, 15(5):1500-1512.

- [26] Yin M, Xie S, Wu Z, et al. Subspace clustering via learning an adaptive low-rank graph[J]. IEEE Transactions on Image Processing, 2018, 27(8):3716 - 3728.
- [27] Liu G, Yan S. Latent low-rank representation for subspace segmentation and feature extraction[C]. IEEE International Conference on Computer Vision, Toronto, Ontario, Canada, May 27-30, 2012:1615 - 1622.
- [28] Wang Y X, Xu H, Leng C. Provable subspace clustering: When LRR meets SSC[C]. Advances in Neural Information Processing Systems, Lake Tahoe, Nevada, USA, December 13-14, 2013:64 - 72.
- [29] Peng X, Yi Z, Tang H. Robust subspace clustering via thresholding ridge regression[C]. the Association for the Advance of Artificial Intelligence, Austin, Texas, USA, Jan 25-29, 2015:3827 - 3833.
- [30] Brbic M, Kopriva I. Multi-view low-rank sparse subspace clustering[J]. Pattern Recognition, 2018, 73:247 - 258.
- [31] Li Q, Liu W, Li L. Affinity learning via a diffusion process for subspace clustering[J]. Pattern Recognition, 2018, 84:39 - 50.
- [32] Zhu Y, Ting K M, Carman M J. Grouping points by shared subspaces for effective subspace clustering[J]. Pattern Recognition, 2018, 83:230-244.
- [33] Peng X, Zhang L, Yi Z. Scalable sparse subspace clustering[C]. IEEE International Conference on Computer Vision, Amsterdam, Netherlands, October 8-16, 2016:3918-3927.
- [34] Zhang X, Sun F, Liu G, et al. Fast low-rank subspace segmentation[J]. IEEE Transactions on Knowledge and Data Engineering, 2014, 26(5):1293 - 1297.
- [35] Feng J, Lin Z, Xu H, et al. Robust subspace segmentation with block-diagonal prior[C]. IEEE Conference on Computer Vision and Pattern Recognition. Columbus, Ohio, USA, Jun 24-27, 2014:3818 - 3825.
- [36] Xiao S, Li W, Xu D, et al. FaLRR: A fast low rank representation solver[C]. Computer Vision and Pattern Recognition, Boston, Massachusetts, June 8-10, 2015:4612 - 4620.
- [37] Shen J, Li P, Xu H. Online low-rank subspace clustering by basis dictionary pursuit[C]. International Conference on International Conference on Machine Learning, New York City, USA, June 19-24, 2016:622 - 631.
- [38] Peng X, Feng J, Xiao S, et al. Structured auto encoders for subspace clustering[J]. IEEE Transactions on Image Processing, 2018, 27(10):5076-5086.

- [39] Candes E J, Wakin M B, Boyd S P. Enhancing sparsity by reweighted L1 minimization[J]. Journal of Fourier Analysis and Applications, 2008, 14(5):877 - 905.
- [40] Lu C, Zhu C, Xu C, et al. Generalized singular value thresholding[C]. Twenty-Ninth AAAI Conference on Artificial Intelligence, Austin, Texas, USA, Jan 25-29, 2015:1805 - 1811.
- [41] Zhang D, Hu Y, Ye J, et al. Matrix completion by truncated nuclear norm regularization[C]. Computer Vision and Pattern Recognition, Rhode Island, USA, June 16-21, 2012:2192 - 2199.
- [42] Donga H, Yua J, Xiaoa C. Dual reweighted lp-norm minimization for salt-and-pepper noise removal[J]. arXiv:1811.09173, 2018:1 - 13.
- [43] Gu S, Zhang L, Zuo W, et al. Weighted nuclear norm minimization with application to image denoising[C]. Computer Vision and Pattern Recognition, Columbus, Ohio, USA, Jun 24-27, 2014:2862 - 2869.
- [44] Xie Y, Qu Y, Tao D, et al. Hyperspectral image restoration via iteratively regularized weighted Schatten p -norm minimization[J]. IEEE Transactions on Geoscience and Remote Sensing, 2016, 54(8):4642 - 4659.
- [45] Xie Y, Gu S, Liu Y, et al. Weighted Schatten p -norm minimization for image denoising and background subtraction[J]. IEEE transactions on Image Processing, 2016, 25(10):4842 - 4857.
- [46] Peng Y, Suo J, Dai Q, et al. Reweighted low-rank matrix recovery and its application in image restoration[J]. IEEE Transactions on Cybernetics, 2014, 44(12):2418 - 2430.
- [47] 刘园, 王迪, 等人. 基于 RPCA 模型的 P 范数优化算法[J]. 温州大学学报 (自然科学版), 2016, 4:25 - 32.
- [48] Wren C R, Azarbayejani A, Darrell T, et al. Pfister: Real-time tracking of the human body[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1997, 19(7):780 - 785.
- [49] Stauffer C, Grimson W E L. Adaptive background mixture models for real-time tracking[C]. Computer Vision and Pattern Recognition. Ft. Collins, CO, USA, June 23-25, 1999, 2:246 - 252.
- [50] Toyama K, Krumm J, Brumitt B, et al. Wallflower: Principles and practice of background maintenance[C]. IEEE International Conference on Computer Vision. Corfu, Greece, September 21-22, 1999, 1:255 - 261.
- [51] Elgammal A, Harwood D, Davis L. Non-parametric model for background subtraction[C]. European Conference on Computer Vision. Dublin, Ireland, June 26-July 1, 2000:751 - 767.
- [52] Lin H H, Liu T L, Chuang J H. A probabilistic SVM approach for

- background scene initialization[C]. International Conference on Image Processing. Rochester, NY, USA, September 22-25, 2002, 3:893 - 896.
- [53] Butler D, Sridharan S, Bove V J. Real-time adaptive background segmentation[C]. IEEE International Conference on Acoustics, Speech, and Signal Processing, Hong Kong, China, April 6-10, 2003, 3:340 - 349.
- [54] Kim K, Chalidabhongse T H, Harwood D, et al. Real-time foreground - background segmentation using codebook model[J]. Real-time imaging, 2005, 11(3):172 - 185.
- [55] Messelodi S, Modena C M, Segata N, et al. A Kalman filter based background updating algorithm robust to sharp illumination changes[C]. International Conference on Image Analysis and Processing, Cagliari, Italy, September 6-8, 2005:163 - 170.
- [56] Wang H, Suter D. Sacon: a consensus based model for background subtraction[J]. Schools & Disciplines, 2005, 22(10):231 - 240.
- [57] Barnich O, Droogenbroeck M V. ViBE: A powerful random technique to estimate the background in video sequences[C]. IEEE International Conference on Acoustics, Speech and Signal Processing, Taipei, China, April 19-24, 2009:945 - 948.
- [58] Biswas S, Sil J, Sengupta N. Background modeling and implementation using discrete wavelet transform: a review[J]. International Journal on Graphics, Vision and Image Processing, 2011, 11(1):29 - 42.
- [59] Evangelio R H, Pätzold M, Sikora T. Splitting gaussians in mixture models[C]. International Conference on Advanced Video and Signal Based Surveillance (AVSS), Beijing, China, September 18-21, 2012:300 - 305.
- [60] Hofmann M, Tiefenbacher P, Rigoll G. Background segmentation with feedback: the pixel-based adaptive segmenter[C]. IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, Providence, Rhode Island, October 6-11, 2012:38 - 43.
- [61] Lu X. A multiscale spatio-temporal background model for motion detection[C]. IEEE International Conference on Image Processing (ICIP), Paris, France, October 27-30, 2014:3268 - 3271.
- [62] Sedky M, Moniri M, Chibelushi C C. Spectral-360: A physics-based technique for change detection[C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Columbus, Ohio, June 24-27, 2014:399 - 402.
- [63] Wang Y, Luo Z, Jodoin P M. Interactive deep learning method for segmenting moving objects[J]. Pattern Recognition Letters, 2017, 96:66 - 75.

- [64] 曹明伟, 余烨. 基于多层背景模型的运动目标检测[J]. 电子学报, 2016, 44(9):2126 - 2133.
- [65] Jiang S, Lu X. Wesambe: A weight-sample-based method for background subtraction[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2018, 28(9):2105 - 2115.
- [66] Liang D, Kaneko S. Improvements and experiments of a compact statistical background model[J]. Arxiv, 2014:1 - 12.
- [67] Gregorio M D, Giordano M. Change detection with weightless neural networks[C]. IEEE Conference on Computer Vision and Pattern Recognition Workshops, Columbus, Ohio, June 24-27, 2014:409 - 413.
- [68] Babaei M, Dinh D T, Rigoll G. A deep convolutional neural network for video sequence background subtraction[J]. Pattern Recognition, 2018, 76:635 - 649.
- [69] Oliver N M, Rosario B, Pentland A P. A bayesian computer vision system for modeling human interactions[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2000, 22(8):831 - 843.
- [70] Candès E J, Li X, Ma Y, et al. Robust principal component analysis?[J]. Journal of the ACM (JACM), 2011, 58(3):11 - 39.
- [71] Huang J, Zhang T, et al. The benefit of group sparsity[J]. The Annals of Statistics, 2010, 38(4):1978 - 2004.
- [72] 杨磊, 庞芳, 胡豁生. 低秩 - 稀疏与全变分表示的运动目标检测方法[J]. 控制理论与应用, 2020, 37(01):81-88.
- [73] Liu X, Zhao G, Yao J, et al. Background subtraction based on low-rank and structured sparse decomposition[J]. IEEE Transactions on Image Processing, 2015, 24(8):2502 - 2514.
- [74] Javed S, Mahmood A, Almaadeed S, et al. Moving object detection in complex scene using spatiotemporal structured-sparse RPCA[J]. IEEE Transactions on Image Processing, 2019, 28(2): 1007-1022.
- [75] Ebadi S E, Izquierdo E. Foreground segmentation via dynamic tree-structured sparse RPCA[C]. European Conference on Computer Vision. Amsterdam, The Netherlands, October 11-14, 2016:314 - 329.
- [76] Xu Y, Wu Z, Chanussot J, et al. Joint reconstruction and anomaly detection from compressive hyperspectral images using mahalanobis distance-regularized tensor RPCA[J]. IEEE Transactions on Geoscience and Remote Sensing, 2018, 56(28): 2919-2930.
- [77] Oreifej O, Li X, Shah M. Simultaneous video stabilization and moving object detection in turbulence[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2013, 35(2):450 - 462.

- [78] Sobral A, Bouwmans T, ZahZah E h. Double-constrained RPCA based on saliency maps for foreground detection in automated maritime surveillance[C]. IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Karlsruhe, Germany, Aug 25-28, 2015:1 - 6.
- [79] Lin L, Lin W, Huang S, et al. Group object detection and tracking by combining RPCA and fractal analysis[C]. Soft Computing, 2018, 22(1): 231-242.
- [80] Hu W, Yang Y, Zhang W, et al. Moving object detection using tensor-based low-rank and saliently fused-sparse decomposition[J]. IEEE Transactions on Image Processing, 2017, 26(2):724 - 737.
- [81] Xue J, Zhao Y, Liao W, et al. Total variation and rank-1 constraint RPCA for background subtraction[J]. IEEE Access, 2018, 6: 49955-49966.
- [82] Cao W, Wang Y, Sun J, et al. Total variation regularized tensor RPCA for background subtraction from compressive measurements[J]. IEEE Transactions on Image Processing, 2016, 25(9):4075 - 4090.
- [83] Zhou T, Tao D. Godec: Randomized low-rank and sparse matrix decomposition in noisy case[C]. International Conference on Machine Learning. Bellevue, Washington, USA, June 28-July 2, 2011:421 - 430.
- [84] Zhou X, Yang C, Yu W. Moving object detection by detecting contiguous outliers in the low-rank representation[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2013, 35(3):597 - 610.
- [85] Xin B, Tian Y, Wang Y, et al. Background subtraction via generalized fused lasso foreground modeling[C]. IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, June 7-11, 2015:4676 - 4684.
- [86] 常侃, 张智勇, 陈诚, et al. 采用低秩与加权稀疏分解的视频前景检测算法[J]. 电子学报, 2017, 45(9):2272 - 2280.
- [87] Kaplanyan A, Sochenov A, Leimkuhler T, et al. DeepFovea: neural reconstruction for foveated rendering and video compression using learned statistics of natural videos[J]. ACM Transactions on Graphics, 2019, 38(6):212-230.
- [88] Liang H, Zhang X, Cheng H, et al. Huffman-code based retrieval for encrypted JPEG images[J]. Journal of Visual Communication and Image Representation, 2019, 61: 149-156.
- [89] Thakur V S, Thakur K, Gupta S, et al. Improved optimum nonnegative integer bit allocation algorithm using fuzzy domain variance estimation and refinement for the wavelet-based image compression[J]. Circuits Systems and Signal Processing, 2019, 38(8): 3880-3900.

- [90] Agababov V, Buettner M, Chudnovsky V, et al. Flywheel: google's data compression proxy for the mobile web[C]. USENIX Symposium on Networked Systems Design and Implementation, Oakland, CA, USA, May 4-6, 2015, 15:367 - 380.
- [91] Amiri S A, Hassanpour H. Image compression using JPEG with reduced blocking effects via adaptive down-sampling and self-learning image sparse representation[J]. Multimedia Tools and Applications, 2018, 77(7):8677 - 8693.
- [92] Mohanty S P, Kougianos E, Guturu P, et al. SBPG: secure better portable graphics for trustworthy media communications in the IoT[J]. IEEE Access, 2018, 6: 5939-5953.
- [93] Suzuki T, Yoshida T. Lower complexity lifting structures for hierarchical lapped transforms highly compatible with JPEG XR standard[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2017, 27(12): 2652-2660.
- [94] Hinton G E, Salakhutdinov R R. Reducing the dimensionality of data with neural networks[J]. Science, 2006, 313(5786):504 - 507.
- [95] Toderici G, O' Malley S M, Hwang S J, et al. Variable rate image compression with recurrent neural networks[J]. arXiv:1511.06085, 2015:1 - 15.
- [96] Toderici G, Vincent D, Johnston N, et al. Full resolution image compression with recurrent neural networks[C]. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, Hawaii, USA, July 21-26, 2017:5435 - 5443.
- [97] Van den Oord A, Kalchbrenner N, Espeholt L, et al. Conditional image generation with pixelcnn decoders[C]. Advances in Neural Information Processing Systems, Barcelona, December 9-10, 2016. 2016:4790 - 4798.
- [98] Theis L, Shi W, Cunningham A, et al. Lossy image compression with compressive autoencoders[J]. International Conference on Learning Representations (ICLR), Toulon, France, April 24-26, 2017:1 - 19.
- [99] Dumas T, Roumy A, Guillemot C. Image compression with stochastic winner-take-all auto-encoder[C]. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Seoul, Korea, April 22-27, 2017:1512 - 1516.
- [100] Baig M H, Koltun V, Torresani L. Learning to inpaint for image compression[C]. Advances in Neural Information Processing Systems (NIPS), Long Beach, USA, December 4-9, 2017:1246 - 1255.
- [101] Cheng Z, Sun H, Takeuchi M, et al. Deep convolutional

- autoencoder-based lossy image compression[J]. arXiv:1804.09535, 2018:1 - 16.
- [102] Mentzer F, Agustsson E, Tschannen M, et al. Conditional probability models for deep image compression[C]. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, June 18-22, 2018:3 - 11.
- [103] Dai W, Shen Y, Tang X, et al. Sparse representation with spatio-temporal online dictionary learning for promising video coding[J]. IEEE Transactions on Image Processing, 2016, 25(10):4580 - 4595.
- [104] Irannejad M, Mahdavi-Nasab H. Block matching video compression based on sparse representation and dictionary learning[J]. Circuits, Systems, and Signal Processing, 2017, 21(49):1 - 21.
- [105] Song Y, Yang G, Xie H, et al. Residual domain dictionary learning for compressed sensing video recovery[J]. Multimedia Tools and Applications, 2017, 76(7):10083 - 10096.
- [106] Sullivan G J, Ohm J R, Han W J, et al. Overview of the high efficiency video coding(HEVC) standard[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2012, 22(12):1649 - 1668.
- [107] Zhang J, Kwong S T W, Zhao T, et al. Complexity control in HEVC intra coding for industrial video applications[J]. IEEE Transactions on Industrial Informatics, 2018, 13(99):1 - 14.
- [108] Li Y, Liu D, Li H, et al. Convolutional neural network-based block up-sampling for intra frame coding[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2018, 28(9):2316 - 2330.
- [109] Liu Z, Yu X, Gao Y, et al. CU partition mode decision for HEVC hardwired intra encoder using convolution neural network[J]. IEEE Transactions on Image Processing, 2016, 25(11):5088 - 5103.
- [110] Wang T, Chen M, Chao H. A novel deep learning-based method of improving coding efficiency from the decoder-end for hevc[C]. Data Compression Conference (DCC), Snowbird, Utah, USA, April 4-7, 2017:410 - 419.
- [111] Joint Video Exploration Team . VVC Test Model. [EB/OL]. (2019-08-12) [2019-11-20]. https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware_VTM/-/releases.
- [112] Fang L, He N, Lin H. CP tensor-based compression of hyperspectral images[J]. Journal of the Optical Society of America A, 2017, 34(2):252 - 258.

- [113] Zhao H, Ma L. Power distribution system stream data compression based on incremental tensor decomposition[J]. IEEE Transactions on Industrial Informatics, 2020, 16(4): 2469–2476.
- [114] Wang Q, Wei M, Chen X, et al. Joint encryption and compression of 3D images based on tensor compressive sensing with non-autonomous 3D chaotic system[J]. Multimedia Tools and Applications, 2018, 77(2):1715 – 1734.
- [115] Friedland S, Li Q, Schonfeld D. Compressive sensing of sparse tensors. [J]. IEEE Transactions on Image Processing, 2014, 23(10):4438 – 4447.
- [116] Du B, Zhang M, Zhang L, et al. Hyperspectral biological images compression based on multiway tensor projection[C]. IEEE International Conference on Multimedia and Expo (ICME), Chengdu, China, July 14–18, 2014:1 – 6.
- [117] Chen T, Liu H, Shen Q, et al. DeepCoder: A deep neural network based video compression[C]. Visual Communications and Image Processing (VCIP), St. Petersburg, FL, USA, December 10–13, 2017:1 – 4.
- [118] Wu C Y, Singhal N, Krhenbühl P. Video compression through Image Interpolation[C]. European Conference on Computer Vision (ECCV). Munich, Germany, September 8–14, 2018:425 – 440.
- [119] Han J, Lombardo S, Schroers C, et al. Deep Probabilistic Video Compression[J]. arXiv:1810.02845, 2018:1 – 9.
- [120] Kim S, Park J S, Bampis C G, et al. Adversarial video compression guided by soft edge detection[J]. arXiv:1811.10673, 2018:1 – 10.
- [121] Yingzhen L, Mandt S. Disentangled sequential autoencoder[C]. International Conference on Machine Learning (ICML), LStockholm, Sweden, July 13–19, 2018:5656 – 5665.
- [122] Liu Y, Liu K, Zhang C, et al. Entropy-based active sparse subspace clustering[J]. Multimedia Tools and Applications, 2018, 12(35):1 – 17.
- [123] Zhang H, Yang J, Shang F, et al. LRR for subspace segmentation via tractable Schatten- p norm minimization and factorization[J]. IEEE Transactions on Systems, Man, and Cybernetics, 2019, 49(5):1722 – 1734.
- [124] Lu Y, Foster D P. Fast ridge regression with randomized principal component analysis and gradient descent[J]. ArXiv:1405.3952, 2014:1 – 11.
- [125] Byrd R H, Hansen S L, Nocedal J, et al. A stochastic quasi-Newton method for large-scale optimization[J]. SIAM Journal on Optimization, 2016, 26(2):1008 – 1031.

- [126] Zhang T. Solving large scale linear prediction problems using stochastic gradient descent algorithms[C]. Proceedings of The Twenty-First International Conference on Machine learning, Banff, Alberta, Canada, July 4-8, 2004:116 - 123.
- [127] Johnson R, Zhang T. Accelerating stochastic gradient descent using predictive variance reduction[C]. Advances in Neural Information Processing Systems, Lake Tahoe, Nevada, USA, Dec 13-14, 2013:315 - 323.
- [128] Tan C, Ma S, Dai Y H, et al. Barzilai-Borwein step size for stochastic gradient descent[C]. Advances in Neural Information Processing Systems, Barcelona, Dec 9-13, 2016:685 - 693.
- [129] Artemiou A, Li B. On principal components and regression: a statistical explanation of a natural phenomenon[J], Statistica Sinica, 2009:1557 - 1565.
- [130] Lu Y, Dhillon P, Foster D P, et al. Faster ridge regression via the subsampled randomized hadamard transform[C]. Advances in Neural Information Processing Systems, Lake Tahoe, Nevada, USA, Dec 13-14, 2013:369 - 377.
- [131] Max A. Woodbury. Woodbury_matrix_identity. [EB/OL]. (2011-08-12) [2017-11-20]. http://en.wikipedia.org/wiki/Woodbury_matrix_identity.
- [132] Cai D, He X, Han J. Document clustering using locality preserving indexing[J]. IEEE Transactions on Knowledge and Data Engineering, 2005, 17(12):1624 - 1637.
- [133] Liu G, Lin Z, Yan S, et al. Robust recovery of subspace structures by low-rank representation[J]. IEEE Transactions on Pattern Recognition and Machine Intelligence, 2013, 35(1):171 - 184.
- [134] Vasamsetti S, Setia S, Mittal N, et al. Automatic underwater moving object detection using multi-feature integration framework in complex backgrounds[J]. IET Computer Vision, 2018, 12(6):770 - 778.
- [135] Sigari M H, Mozayani N, Pourreza H. Fuzzy running average and fuzzy background subtraction: concepts and application[J]. International Journal of Computer Science and Network Security, 2008, 8(2):138 - 143.
- [136] Mori G. Guiding model search using segmentation[C]. IEEE International Conference on Computer Vision, Beijing, China, Oct 17-20, 2005, 2:1417 - 1423.
- [137] Ma Y, Derksen H, Hong W, et al. Segmentation of multivariate mixed data via lossy data coding and compression[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2007, 29(9):331-345.
- [138] Lang C, Liu G, Yu J, et al. Saliency detection by multitask sparsity pursuit[J]. IEEE Transactions on Image Processing, 2012, 21(3):1327 -

1338.

[139] Lin Z, Chen M, Wu L, et al. The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices[R]. UIUC Technical Report UILU-ENG-09-2215, 2009:1-16.

[140] Cai J, Candes E, Shen Z. A singular value thresholding algorithm for matrix completion[J]. SIAM J. on Optimization, 2010, 20(4):1956 - 1982.

[141] Wang K, Gou C, Wang F Y. M4CD: a robust change detection method for intelligent visual surveillance[J]. IEEE Access, 2018, 6:15505 - 15520.

[142] Isk S, Özkan K, Günel S, et al. SWCD: a sliding window and self-regulated learning-based background updating method for change detection in videos[J]. Journal of Electronic Imaging, 2018, 27(2):23 - 32.

[143] Jiang S, Lu X. WeSamBE: A weight-sample-based method for background subtraction[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2018, 28(9):2105 - 2115.

[144] Varghese A, Sreelekha G. Sample-based integrated background subtraction and shadow detection[J]. Ispj Transactions on Computer Vision and Applications, 2017, 9(1):9 - 25.

[145] Lu X. A multiscale spatio-temporal background model for motion detection[C]. IEEE International Conference on Image Processing (ICIP), Paris, Franch, Oct 27-30, 2014:3268 - 3271.

[146] Sajid H, Cheung S C S. Universal multimode background subtraction[J]. IEEE Transactions on Image Processing, 2017, 26(7):3249 - 3260.

[147] Benezeth Y, Jodoin P M, Emile B, et al. Comparative study of background subtraction algorithms[J]. Journal of Electronic Imaging, 2010, 19(3):033003 - 033003.

[148] Seidel F, Hage C, Kleinsteuber M. pROST: a smoothed ℓ_1 ℓ_p -norm robust online subspace tracking method for background subtraction in video[J]. Machine Vision and Applications, 2014, 25(5):1227 - 1240.

[149] Varadarajan S, Miller P, Zhou H. Spatial mixture of Gaussians for dynamic background modelling[C]. IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Krakow, Poland, Aug 27-31, 2013:63 - 68.

[150] Kolda T G, Bader B W. Tensor decompositions and applications[J]. Siam Review, 2009, 51(3):455 - 500.

- [151] Rumelhart D E, Hinton G E, Williams R J. Learning representations by back-propagating errors[J]. Nature, 1986, 323(6088):533 - 552.
- [152] Kingma D P, Ba J. Adam: A method for stochastic optimization[J]. arXiv:1412.6980, 2014:390 - 405.
- [153] Bader B W, Kolda T G, et al. MATLAB Tensor Toolbox Version 2.6 . [EB/OL]. (2015-03-26) [2019-04-19]. <http://www.sandia.gov/tgkolda/TensorToolbox/index-2.6>.
- [154] Joint Video Exploration Team. HM-16.0. [EB/OL]. (2014-03-08) [2019-11-20]. http://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/tags/HM-16.0/.
- [155] Bjontegaard G. Calculation of average PSNR differences between RD-Curves[C]. ITU-T VCEG Meeting, Austin, TX, USA, April 2-4, 2001:1 - 36.

发表论文和科研情况说明

发表的论文：

- [1] **Yang Li**, Yubao Sun, Qingshan Liu, Shengyong Chen. Fast subspace segmentation via Random Sample Probing[J]. *Neurocomputing*, **SCI**, 2018, 319: 66-73.
- [2] **Yang Li**, Guangcan Liu, Yubao Sun, Qingshan Liu, Shengyong Chen. Moving object detection via segmentation and saliency constrained RPCA[J]. *Neurocomputing*, **SCI**, 2019, 323: 352-362.
- [3] **Li Yang**, Tang Tinglong, Huang Wei. A robust auto-focus measure based on inner energy[J]. *Optoelectronics Letters*, **EI(JA)**, 2017, 13(4): 309-313.

在审的论文：

- [1] **Yang Li**, Guangcan Liu, Yubao Sun, Qingshan Liu, Shengyong Chen. 3D Tensor Auto-Encoder with Application to Video Compression [J]. *ACM Transactions on Multimedia Computing, Communications and Applications*. (Major Revision).
- [2] **Yang Li**. Subspace Recovery based on Truncated Weighted RPCA [J]. *Information Processing Letters*. (Submitted to Journal).

专利：

- [1] 刘光灿, **李阳**, 陈胜勇. 基于高斯极值滤波和群稀疏 RPCA 的运动目标检测方法[P]. 江苏: CN107564029A, 2018-01-09.
- [2] 刘光灿, **李阳**, 陈胜勇. 一种基于三阶张量自编码网络的视频压缩方法[P]. 江苏: CN109302614A, 2019-02-01

致 谢

光阴似箭，岁月如梭。转眼间我的博士学习的生涯即将画上句号，在这读博四年中，老师、同学和家人给与了我很多的帮助和支持，在此我要向所有曾经给与我和帮助和关心的人致以最诚挚的感谢。

首先，我要感谢我的导师陈胜勇教授和南京信息工程大学的刘光灿教授。本论文的工作是在陈胜勇教授和刘光灿教授的悉心指导下完成的。他们学识渊博，在学术研究上都有很深的造诣，是我从事科研的引路人，从基础的选题和写作技巧，到论文的结构与内容，两位导师都给予我极大的帮助。他们教会了我如何去发现有意义的研究方向，更教会了我如何脚踏实地地去做科学研究。他们严谨的治学态度和科学的工作方法给了我极大的帮助和影响，这些深深地影响了我。承蒙恩师谆谆教导，在此衷心感谢四年来陈胜勇教授和刘光灿教授对我的在生活和学習上的关心和指导。同时，还要感谢南京信息工程大学刘青山教授对我在南京信息工程大学学习时期的指导和帮助。感谢北京大学林宙辰教授在我预答辩时给予的详细的指导。

其次，我要感谢课题组黄薇博士、赵萌博士等，在我读博期间，给予了我很多的关心与帮助。感谢课题组的小伙伴们：唐庭龙师兄、何煜师兄、李军伟师兄、程徐、杨延红、张哲、栾昊、彭飞、张阳阳、王良丹、赵宇峰、贾鑫、贾晨、游永、周哲哲、刘梦娜等在学习和生活上给予的帮助，是他们陪伴我度过了四年的博士学习生涯，使我的原本枯燥的学习生活变得丰富有趣。感谢贾鑫、杨延红、栾昊、张哲和赵宇峰的对本文的校稿工作。

我还想感谢所有授课和参与我博士论文评审、答辩的老师。感谢他们指出我论文中的不足，并提出宝贵的指导意见，为我未来的研究点明了方向。

最后，特别感谢我的家人，他们在背后一直默默的奉献，不求回报，在我二十多年的学习生涯中，一直在身后支持着我。在今后的工作和生活中，我将加倍努力，做一个对社会和家庭有用的人。