

密级:_____



中国科学院大学
University of Chinese Academy of Sciences

硕士学位论文

高阶张量奇异值分解人脸识别算法研究

作者姓名: _____ 计雨含

指导教师: _____ 王晓东 研究员

_____ 中国科学院成都计算机应用研究所

学位类别: _____ 工程硕士

学科专业: _____ 软件工程

研 究 所: _____ 中国科学院成都计算机应用研究所

二零一五年四月

Research on face identification that based on higher-order tensor singular value decomposition

A Thesis Submitted to

The University of Chinese Academy of Sciences

In partial fulfillment of the requirement

For the degree of

Master of Software Engineering

By

Ji Yuhan

Directed By

Wang Xiaodong

Chengdu Institute of Computer Applications

Chinese Academy of Sciences

April, 2014

关于学位论文使用权声明

任何收存和保管本论文各种版本的单位和个人，未经著作权人授权，不得将本论文转借他人并复印、抄录、拍照、或以任何方式传播。否则，引起有碍著作权人著作权益之问题，将可能承担法律责任。

.....

关于学位论文使用授权的说明

本人完全了解中国科学院成都计算机应用研究所有关保存、使用学位论文的规定，即：中国科学院成都计算机应用研究所有权保留学位论文的副本，允许该论文被查阅；中国科学院成都计算机应用研究所可以公布该论文的全部或部分内容，可以采用影印、缩印或其他复制手段保存该论文。

（涉密的学位论文在解密后应遵守此规定）

签 名： 导师签名： 日 期：

.....

关于学位论文原创性声明

本人郑重声明：所呈交的学位论文是本人在导师指导下，独立进行研究工作所取得的成果。尽我所知，除文中已经注明引用的内容外，本学位论文的研究成果不包含任何他人享有著作权的内容。对本论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确方式标明。

签 名： 导师签名： 日 期：

摘要

为了使人脸识别算法不受拍摄条件的影响,本文在人脸识别领域应用高阶张量奇异值分解 (High-ordered Tensor Singular Value Decomposition, HOSVD)来进行人脸特征的表示和提取,这种方法克服了其它特征提取方法过于依赖拍摄条件的缺点。HOSVD 将数据转换成三维线性模型,能避免二维线性方法中条件改变则精确度下降的问题,使得识别算法在变化的条件下获得了相对稳定的结果,该方法还可以根据需要将算法扩展到更高维度,使其与更多的拍摄因素无关。然后在此基础上对算法进行了优化,利用矩阵分解方法,在不影响算法正确率的情况下,有效减少计算量,提高算法效率。然后再将小波运算与高阶张量奇异值分解算法结合,先用小波函数处理图像,保留人脸图像的重要信息,滤去多余信息,提高了识别准确度和效率。为了便于展示和编辑数据库,为算法实现了图形展示界面,可以直观快捷的进行操作演示。

基于 Matlab 进行了对比实验,首先将 HOSVD 进行人脸识别的结果与用主成分分析法 (Principal Component Analysis, PCA) 方法得到的结果对比,验证了 HOSVD 算法在表情有很大变化条件下显著的正确性以及稳定性;然后,对优化的 HOSVD 算法进行了实验验证,在数据量较大的情况下,该算法速度明显提高;最后,对与小波结合后的 HOSVD 与原始 HOSVD 算法进行了对比试验,证明结合后的算法在时间效率和准确性上都有明显提高。

本文在鹰眼智能视频检索项目中有着实际的应用:对于公安机关在重要地点对于在逃人员的追捕具有重要的意义,使对犯罪分子的自动识别不受表情、光照等条件因素的影响,是鹰眼项目的重要组成部分。

关键词: 人脸识别,高阶奇异值分解,张量分解,特征提取,模式识别

Abstract

In order to make the face recognition algorithm not affected by the different shooting conditions. This paper used Higher-ordered Singular Value Decomposition (HOSVD) for face feature representation and extraction in the field of face recognition. This method overcame the shortcomings of other feature extraction methods that rely too much on the shooting conditions. HOSVD transforms data into three dimensional linear model, it can avoid the reduce of accuracy when conditions change in two dimensional linear method, This recognition algorithm gets a relatively stable result under changing conditions, and if needed, the algorithm can be extended to higher dimensions, making it not affected by more shooting factors. Then on this basis, the algorithm is optimized, using the method of matrix decomposition. The accuracy is not reduced, but the amount of calculation is reduced, so the efficiency of algorithm is improved. Then it combined higher-order tensor singular value decomposition with wavelet analysis algorithm. Process images using wavelet function first, keep important information of face image, filter the redundant information to improve the recognition accuracy and efficiency. In order to display and edit database, here I made a graphic display interface, which is easy for operating and demonstrating.

This paper carried on the contrast experiment based on the platform of Matlab. Firstly compared recognition result of HOSVD with the result of Principal Component Analysis (PCA) method, the correctness and stability of HOSVD is verified under the condition when there is a big change in expressions. Then the efficiency of the optimized HOSVD algorithm was verified by experiment, especially when the amount of data is large, the algorithm speed increased significantly. Finally, this paper compared the wavelet combined HOSVD and original HOSVD through the contrast test, it showed an obvious improvement in time efficiency and accuracy.

This paper has practical applications in the Eagle eye intelligent video retrieval project: it is very important for the public security organ to hunt for fugitive personnel in the important place. The success of automatic identification of criminals in different

expression and illumination conditions is an very important part of the Eagle Eye project.

KEY WORDS: Face recognition, Higher-order singular value decomposition, Tensor decomposition, Feature extraction, Pattern recognition;

目录

摘要	I
Abstract.....	II
目录	IV
图目录	VI
表目录	VII
1 绪论	1
1.1 人脸识别问题描述.....	1
1.2 人脸识别研究历史及国内外研究现状	2
1.3 人脸识别主要算法综述	3
1.4 人脸识别领域主要挑战	5
1.5 论文的主要研究内容和章节安排	6
2 高阶张量奇异值分解方法	7
2.1 张量相关基本概念	7
2.2 高阶张量奇异值分解	9
2.3 用 HOSVD 获得一个张量的近似	12
2.4 小波变换的基本思想	13
2.4.1 小波简介	13
2.4.2 小波变换的基本理论	14
3 高阶张量奇异值分解与小波分析相结合的人脸识别方法	17
3.1 db2 小波函数处理人脸图像	17
3.2 将人脸识别问题转换到张量领域	19
3.3 应用 HOSVD 提取人脸特征	20
3.4 基于 HOSVD 的人脸识别过程	23
3.5 基于压缩的 HOSVD 优化识别算法	25
4 基于 HOSVD 和小波结合的人脸识别算法实现	27
4.1 程序架构	27

4.2 张量工具包	28
4.3 基本后台程序	33
4.3.1 实验数据	33
4.3.2 数据库构建与特征提取	33
4.3.3 被测图片的识别	35
4.4 图形界面	37
5 算法实验和结果分析	43
5.1 内存分析	43
5.2 算法实验	45
5.2.1 表情无关 HOSVD 与 PCA 算法识别准确度对比实验.....	45
5.2.2 优化的 HOSVD 和原始 HOSVD 对比实验.....	46
5.2.3 与小波结合的 HOSVD 与原始 HOSVD 算法对比.....	52
6 算法在项目中的应用	53
6.1 鹰眼智能视频检索软件 V1.1	53
6.2 系统环境	53
6.3 表情无关人脸识别算法在鹰眼智能视频检索软件中的应用	54
7 总结与展望	57
7.1 总结	57
7.2 展望	58
参考文献	59
硕士研究生学习期间科研成果	63
致谢	65

图目录

图 2- 1 张量示意图	7
图 2- 2 纤维示意图	7
图 2- 3 切面示意图	8
图 3- 1 小波分解前后图像对比	19
图 3- 2 HOSVD 的其中一种解释	21
图 3- 3 HOSVD 的另外一种解释	21
图 4- 1 程序架构流程图	27
图 4- 2 部分实验数据图	33
图 4- 3 YALE 数据库识别结果	36
图 4- 4 图形界面	37
图 5- 1 YALE 数据库实验结果	45
图 5- 2 PCA 算法识别 YALE 数据库实验结果	45
图 5- 3 HOSVD 识别实验室同学图片实验结果	46
图 5- 4 ORL 数据库识别结果	46
图 5- 5 图像模态下的奇异值分布	49
图 5- 6 表情模态 (+) 和人物模态 (°) 下的奇异值分布	49
图 6- 1 系统界面图	54
图 6- 2 案件添加方式	55
图 6- 3 视频压缩方法	55

表目录

表 5.1 矩阵优化与 HOSVD 压缩加速效果.....	51
表 5.2 与小波结合后的算法与原始算法时间和精度对比.....	52
表 6.1 系统硬件环境	53

1 绪论

人脸识别指的是利用计算机分析处理人脸图像并从中提取有效信息从而进行身份识别的计算机技术。人类仿佛生来就具有“人脸识别”的能力，赋予计算机同样的能力是人脸识别研究的最终目标^[1]。人脸识别是一个跨学科的研究课题，它包括了数字图像处理、模式识别、计算机视觉、神经网络、统计学习、人工智能、心理学以及数学等诸多学科的内容。同时人脸识别又是一个拥有非常广阔应用前景的领域，这项技术可以用于公安、安全验证系统、医学、金融、视频会议、交通量控制等生活的方方面面。随着其重要性的日益凸显，人脸识别已成为当前人工智能领域和模式识别的一个研究热点。具体来讲，身份认证识别、监视公共场合的人、图像数据库的检索等都需要先进准确的人脸识别技术。由于计算机并不像人那样的“智能”，所以当人脸表情变化丰富，随年龄的增长而变化，图像受光照、成像角度及成像距离等影响时，计算机有时并不能像人类一样进行有效的识别，这也正是人脸识别领域面临的巨大挑战^[2]。目前尚没有非常好的描述人脸的三维模型，至今人脸识别技术尚未能取得像指纹识别技术那样十分令人满意的结果，还没有通用成熟的人脸自动识别系统出现。因此计算机人脸识别技术的研究也就成为一项极具挑战性的工作。

1.1 人脸识别问题描述

通常，一个完整的人脸识别流程包含以下几个部分^[3]：1.图像采集与人脸检测：在这一个过程中，通过摄像头等采集人脸图像，并通过一定的算法检测其中人脸的位置、大小等信息，以便后续步骤的进行。尤其是近几年以来，为了实用的考虑，人脸识别需要不受环境和其它因素的影响，因此人脸检测得到了更加深入的发展，甚至已成为一个独立的研究领域。2.人脸图像预处理：由于所采集的原始人脸图像常常都到各种干扰或者限制，通常并不能直接供识别过程使用，需要一定的预处理工作。其中包括噪声过滤、灰度校正等。3.人脸特征提取：特征提取是人脸识别过程中最重要的一个环节，前述采集与预处理等环节都是为了此步骤服务，而后续的匹配识别过程也是建立在特征提取的基础上。对于人类来说，人脸是肌肉皮肤还有骨骼器官等的组合，然而对于计算机来说，人脸则是不同像素组成的灰度值矩阵，同一张人脸，受光照、

遮挡、胡须、眼镜、化妆等等因素的影响也可以表现为完全不同的矩阵，所以虽然经过多年的发展，但是如何利用人脸图像进行不受外部环境影响的人脸识别仍然是当前研究的重点，科研人员也一直致力于如何攻克这一问题。4.人脸图像匹配与识别：一旦提取出被测图片中的人脸特征，接下来就可以与数据库中的特征进行比对，如果误差小于给定的阈值则人物匹配，否则被测图片所代表的人物不在数据库中，当然如何对比选定误差阈值也有很多相关的研究。

1.2 人脸识别研究历史及国内外研究现状

人类对人脸识别的研究有着比较悠久的历史，例如 1888 年《Nature》上就有一篇利用人脸进行身份识别的文章，此文章研究的是人类自身对人脸的识别能力。1990 年以来，随着计算机技术以及整个产业的发展，在计算机进行自动人脸识别方面取得了巨大的进展和突破^[4]。现在几乎每所高校都有专门从事人脸识别研究的课题组。

早期的人脸识别主要应用几何特征进行识别。该类方法一般先定位面部的主要器官，如眼睛、眉毛、鼻子、嘴巴等。再选择能够反映这些器官之间角度、距离等结构配置的特质。这些方法以面部特征点直接的距离等作为最后的识别依据。

上世纪 80 年代，人脸识别采用表观进行描述，例如利用图像灰度、小波变换等特征。虽然这些特征其实并没有只直观的物理含义，但是包含的信息丰富，具有很好的鲁棒性。

上世纪 90 年代有人注意到人脸图像像素点直接有很强的相关性，因此利用相关性进行了大量深入的研究。比如 MIT 的 Turk 和 Pentland 提出的主成分分析法(Principal Component Analysis) 为后来人脸识别其它方法奠定了基础。此后的很多方法都是从这个方法的基础上提出来的。例如 MIT 的 Moghaddam 用概率度量取代了简单的欧氏距离度量方法。Belhumeur 也在此基础上提出了线性判别分析方法 (Linear Discriminant Analysis, LDA)。90 年代产生的另一类方法是 Lades 等提出的弹性图匹配方法。利用连通的属性图来描述人脸图像，脸部关键点用图的顶点来表示，其属性是该特征点的 Gabor 变化特征。特征点这件的距离用边的属性来描述。该方法用最后得到的人脸图像的属性图与数据库中的图像的属性图进行对比来完成识别。该方法非常耗时，因此后来有人针对该方法进行了效率上的提高，节省了更多的时间。隐马尔可夫模型也是 90 年代有代表性的方法，运用人脸各部分之间的状态转移关系来

描述人脸。支持向量机方法 (Support Vector Machines, SVM) 和 Boosting 算法等基于统计学习的方法发展与 90 年代后期。20 世纪初, 还出现了针对光照建模的光锥模型、图商像法, Retinex 方法等综合的创新的创新的研究方法。

国内外有许多专家学者活跃在人脸识别这个领域^[5]。如美国 Texas at Dallas 大学的 Abdi 和 Tool 小组, 由 Stirling 的 Bruce 教授和 Glasgow 大学的 Burto 合作领导的小组等;再比如如英国的 Graw 小组和荷兰 Groningen 大学的 Petkov 小组, MIT 的 Media 实验室的 Pentland 小组, 还有 C.vonder Malsburg 小组, 他领导了美国的 Southern California 大学和德国的 Bochum 大学合作, 采用动态链接结构和弹性图像匹配等方法。国内主要有中科院自动化所、清华大学、浙江大学等高校及科研院所。另外已经由一些公司开发出走在时代前沿的人脸识别应用系统, 如美国的 L-1 Identity Solution 公司和 A4Vision 公司的产品。

总之, 在外界条件不理想的情况下, 大规模数据的人脸识别仍然为当前的研究热点问题, 因此本文旨在提高非理想情况下的识别精度, 提高大规模人脸识别的效率。

1.3 人脸识别主要算法综述

一般地, 可以分为基于几何特征提取、基于代数特征提取、基于神经网络的方法等等。几何特征^[6]一般用人脸的特定两点之间的距离、角度、曲率等等分量以及特定器官的大小等分量来表示, 该方法的基本思想是: 把人脸看做是由各种部件组成的, 因此把各个部件独立拆分开来, 如眼睛、鼻子等, 单独测量其几何特征, 包括大小、形状等, 然后基于这些特征来进行人脸识别。主要的基于几何的提取方法包括: 局部特征分析法、弹性模板、活动轮廓等。局部特征分析法利用一些主要器官间的方位比例距离等作为参数, 与已有的人脸图像进行比较, 如果差距小于一定的值, 则可以识别为同一个人; 弹性模板也是一种基于几何特征的人脸识别方法, 模板在匹配时, 可以旋转、平移以及变换形状, 该方法需要构造能量函数并且使能量函数的值达到最小; 活动轮廓模型的计算量比较小, 并且具有很好的鲁棒性, 尤其是当图像有倾斜的时候。该方法简单直接, 然而缺点也比较明显。它只获得了人脸局部细节特征, 而忽略了整体性, 可以说丢失了人脸图片中大量的信息, 在此基础上, 可以想见, 其准确性会受

到相当大的影响。此外，一旦人脸图片拍摄条件，如角度、表情等发生变化，则该算法的稳定性将大受影响。

而基于代数特征的思路则是通过映射进行分类。该方法通过变换投影，将图像投影到另一个新的空间，新的维度上^[7]。即是“特征脸”所形成的降维空间的投影。常见的代数特征提取方法则包括：独立成分分析法、主成分分析法、KL 变换、奇异值分解、隐马尔科夫等。独立成分分析法（Independent Component Analysis）的基本思想是寻找在特种空间上数据相互独立的方向，独立是该方法的核心，即分解后的各分量是独立的，该方法又称为特征脸方法；PCA 方法实现简单，计算量较小，然而，该算法只有在所有人脸图片都是在类似的条件下拍摄时，方能取得较好的效果。一旦数据库图片或者被测图片的拍摄条件（光照、角度等）发生改变，那么识别结果的准确率将受到较大的影响。

独立成分分析法是一种分离技术，利用了信号的高阶统计特性；Turk 和 Pentland 于 1991 年提出主成分分析方法（Principal Component Analysis），该方法又称为特征脸方法，该方法是统计学中的一种重要的方法，亦可以称为特征脸法，旨在在特征空间中寻找那些方向，使得该方向上的数据误差最小，从而解释数据的本质特征，同时进行了降维。

每一张人脸图片是一个矩阵。在该方法中^[8]，将一个图片矩阵的所有列向量堆叠起来，形成一个列向量。这样，任何人脸图片都可以看成是某一个线性空间里面的一个向量^[9]。这个线性空间由一些基本向量（该线性空间的基底）张成。具体的人脸识别的过程分为两部分：一是根据已有数据库来计算基底，以及数据库中每个人在该组基底下的坐标；二是计算被测图片在该组基底下的坐标，并将其与已知任务的坐标进行比对，进行身份识别。

具体计算过程如下：

假设有 n 个人，每人一张图片，图片的像素数为 m （长度为 m 的列向量），则所有图片形成一个 $m \times n$ 的矩阵 X 。计算矩阵 S 的特征向量，其中：

$$S = XX^T \quad (1-1)$$

取前 M 个特征向量，作为整个线性空间的基底。每一个特征向量，可以认为代表一种人脸的特征，即所谓的“特征脸”。然后将被测图片投射到该线性空间，求出其在该组基底下的坐标，通过与已知人物进行比对，完成人脸识别。

PCA 方法实现简单，计算量较小；然而，该算法只有在所有人脸图片都是在类似的条件下拍摄时，方能取得较好的效果。一旦数据库图片或者被测图片的拍摄条件（光照、角度等）发生改变，那么识别结果的准确率将受到较大的影响。奇异值分解算法就是在该方法的基础上发展而来的；奇异值分解（Singular Value Decomposition）是一种十分有效基于代数特征的提取方法，奇异值特征是人脸图像的本质特征，因此它是稳定的，内在的，旋转不变的，镜像不变的。该方法是目前非常主流并且高准确度的识别方法。另外，该技术也应用在信号处理、模式识别等各个方面^[10]。

神经网络因为具有记忆功能，所以可以用来处理比较复杂的检测问题，准确性也比较高。有很多利用神经网络识别人脸的方法，比如概率决策神经网络等方法。但是人脸是高维的，有很多训练节点，所以需要大量的训练样本，导致了该方法的局限性在于着重于系统的优化训练，当下流行的方法还包括支持向量机的方法等等^[11]。

随着小波理论的日趋成熟，利用小波分析对图像进行识别也是当下研究的热点。小波分析可以在时间和频率域的变化，可以对图像做多尺度的分析。例如通过伸缩平移等在图像中提取信息。该方法通过使用不同带宽的函数来将信号分解，并且舍弃高频分量降维低频分量。小波变换较傅里叶变换有许多明显的优势，解决了后者所不能解决的很多问题，在调和与分析领域具有重要的意义^[12]。

1.4 人脸识别领域主要挑战

人脸识别技术的关键，就在于人脸特征的表示与提取^[13]。人脸的检测与预处理，都是为了特征提取服务。而一旦特征提取完毕，则匹配与识别也是水到渠成。特征提取，作为人脸识别程序最重要的一环，也是最难的一环。

尽管不同人的面貌千差万别，然而，严格来讲，这些差别大多是细节上的差别。在宏观上，不同个体的脸部轮廓，五官分布等等，其实差别不大^[14]。另一方面，同一个体，在不同条件下，面部图片可以发生很大的变化。不同的表情、光照、拍摄角度等，都可以带来巨大的变化。那么，如何表示以及提取特征，使得这种特征可以放大不同个体之间的差别，缩小同一个体在不同条件下的差别，对准确、快速的进行人脸识别至关重要。这一问题，正式当前人脸识别研究领域面临的主要挑战，也是研究者们全力攻克的方向^[15]。

1.5 论文的主要研究内容和章节安排

人脸识别是一个拥有广阔应用前景并且充满挑战的课题，在许多领域都有很实际的应用。本文在充分调研与 SVD 相关算法的基础上，查阅了大量的资料，了解了人脸识别领域主流的算法。提出了一种高效的表情无关的人脸识别方法。

第一章是绪论部分，主要对人脸识别研究的发展历史做了概述，并且综述了一些当前主流的人脸识别研究方法。

第二章主要是对文章相关的概念的表述。首先对高阶张量奇异值分解的相关概念和基础做了介绍，介绍了张量相关的概念，然后介绍如何把张量奇异值分解方法扩展到三维领域，引出 HOSVD 的概念。最后介绍了小波理论的相关概念和基本理论。

第三章主要介绍了高阶张量奇异值分解与小波分析相结合的人脸识别方法。侧重于对算法理论的分析 and 研究。该算法首先用 db2 小波处理人脸图像，然后将人脸问题转换到张量领域，用 HOSVD 进行人脸特征提取并识别，最后提出优化的 HOSVD 算法，在不改变算法准确率的基础上提高了算法的效率

第四章是对上述算法的实现过程，包括系统流程图以及算法实现的部分重要代码，为了便于展示以及编辑数据库，为系统设计了图形界面。

第五章实验和实验结果的分析，用三组数据进行了对比实验。对比了 HOSVD 和普通特征提取算法的识别精度；优化的 HOSVD 与原始的算法的时间效率；对比了 HOSVD 与小波结合后的算法与原算法的精确度和时间效率，验证了优化算法的有效性。

第六章是结语，在总结文章主要内容的基础上对未来的工作进行了展望。

2 高阶张量奇异值分解方法

本章将重点介绍关于高阶张量奇异值分解的相关概念和理论基础。首先了解张量相关的基本概念，然后介绍如何进行高阶张量的奇异值分解，最后介绍了小波的基础知识。这里的理论基础工作主要是为了以后的实际应用打基础。

2.1 张量相关基本概念

张量 (tensor): 一般地，一维数组称为向量，二维数组称为矩阵，而三维及三维以上数组，则称为张量。张量的每一个维度被称为模态 (mode) [16]。

图 2-1 是一个张量示意图：

$$\mathbf{X} = (\mathbf{x}_{ijk}) \in \mathbb{R}^{I \times J \times K}$$

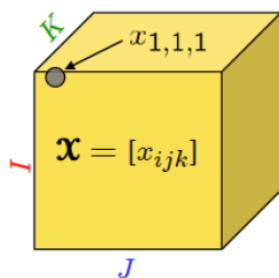


图 2- 1 张量示意图

纤维：张量中有纤维，纤维类似矩阵的行和列。在三维张量中，纵向的向量维为第一模态的纤维，横向为第二模态纤维。

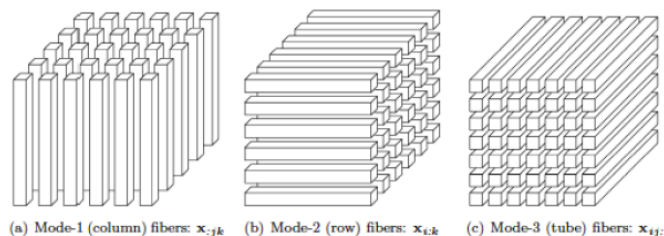


图 2- 2 纤维示意图

切面：如同矩阵包含向量，张量也包含切面。如图 2-3 展示了张量的各个切面。

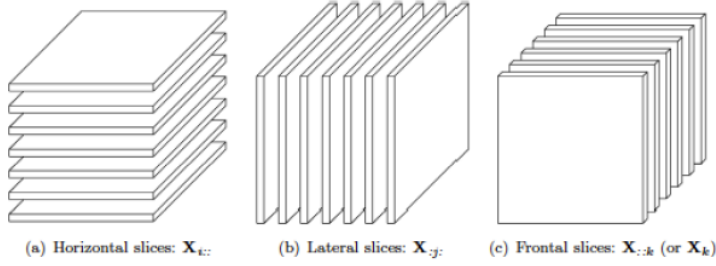


图 2- 3 切面示意图

矩阵化：矩阵化即将张量转换为矩阵的过程。张量 $X \in R^{I_1 \times I_2 \times \dots \times I_n}$ 在第 n 个模态下的矩阵化，就是以张量在该模态下的纤维作为矩阵的列，将张量展开成矩阵，其结果记为 $X_{(n)}$ ^[17]。以 $X \in R^{3 \times 4 \times 2}$ 为例，设其前切面为：

$$X_1 = \begin{bmatrix} 1 & 4 & 7 & 10 \\ 2 & 5 & 8 & 11 \\ 3 & 6 & 9 & 12 \end{bmatrix}, \quad X_2 = \begin{bmatrix} 13 & 16 & 19 & 22 \\ 14 & 17 & 20 & 23 \\ 15 & 18 & 21 & 24 \end{bmatrix}$$

该张量在第一模态下矩阵化所得结果为：

$$X_{(1)} = \begin{bmatrix} 1 & 4 & 7 & 10 & 13 & 16 & 19 & 22 \\ 2 & 5 & 8 & 11 & 14 & 17 & 20 & 23 \\ 3 & 6 & 9 & 12 & 15 & 18 & 21 & 24 \end{bmatrix}$$

张量与矩阵相乘：张量 $X \in R^{I_1 \times I_2 \times \dots \times I_n}$ 在第 n 个模态下与矩阵 $U \in R^{I_n \times J_n}$ 相乘，记为 $X \times_n U$ ，所得结果是一个张量。其维度为： $I_1 I_2 \dots I_{n-1} \times I_j \times I_{n+1} \times \dots \times I_N$ ，计算过程如下：

$$(X \times_n U)_{i_1, i_2, \dots, i_{n-1}, j, i_{n+1}, \dots, i_N} = \sum_{i_n=1}^{I_n} x_{i_1, i_2, \dots, i_{n-1}, i_n, i_{n+1}, \dots, i_N} u_{i_n, j} \quad (2-1)$$

具体来讲，就是将张量在该模态下展开，令矩阵与展开后的张量相乘，然后将所得结果折叠，形成新的张量。该过程也可以写成如下形式：

$$Y = X \times_n U \Leftrightarrow Y_{(n)} = U X_{(n)} \quad (2-2)$$

仍以前述张量 $X \in R^{3 \times 4 \times 2}$ 为例，设有矩阵 U ，

$$U = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$$

则张量与该矩阵在第一模态下相乘的结果为（以前切面表示）：

$$Y_1 = \begin{bmatrix} 22 & 49 & 76 & 103 \\ 28 & 64 & 100 & 136 \end{bmatrix} \quad Y_2 = \begin{bmatrix} 130 & 157 & 184 & 211 \\ 172 & 208 & 244 & 280 \end{bmatrix}$$

由定义易知，若多个张量分别于多个矩阵在不同模态下相乘，则最终结果与运算顺序无关，即张量与矩阵在不同模态下的乘法具有交换性：

$$X \times_n A \times_n B = X \times_n B \times_n A \quad (m \neq n) \quad (2-3)$$

若模态相同，则有：

$$X \times_n A \times_n B = X \times_n (BA) \quad (2-4)$$

张量与向量相乘：张量 $X \in R^{I_1 \times I_2 \times \dots \times I_N}$ 与向量 $v \in R^{I_n}$ 在第 n 模态下相乘，记为 $X \times_n v$ ，

其结果为 $(N-1)$ 维张量，维度为 $I_1 I_2 \dots I_{n-1} \times I_{n+1} \times \dots \times I_N$ ，计算如下：

$$(X \times_n v)_{i_1 i_2 \dots i_{n-1} i_{n+1} \dots i_N} = \sum_{i_n=1}^{I_n} x_{i_1 i_2 \dots i_N} v_{i_n} \quad (2-5)$$

2.2 高阶张量奇异值分解

SVD 是矩阵奇异值分解对于张量的推广。它有许多不同的形式，而我们这里将要描述的形式，类似于矩阵的主成分分析^[18]。这种推广常常被称为高阶 SVD (HOSVD)。

张量 $A \in R^{I_1 \times I_2 \times \dots \times I_N}$ 可以写成如下形式：

$$A = S \times_1 U^{(1)} \times_2 U^{(2)} \times_3 U^{(3)} \dots \times_N U^{(N)} \quad (2-6)$$

其中 $U^{(1)}, U^{(2)}, U^{(3)}$ 都是正交矩阵。S 是一个张量，它与张量 A 维度相同，它也具有全正交性：S 的任意两个切面相互正交，其内积为零：

$$\langle S(i, :, :), S(j, :, :) \rangle = \langle S(:, i, :), S(:, j, :) \rangle = \langle S(:, :, i), S(:, :, j) \rangle = 0 \quad (2-7)$$

其中 $i \neq j$ 。核心张量 S 在 1-模态下的奇异值定义如下：

$$\sigma_j^{(1)} = \|S(i, :, :)\|_F = \sqrt{\sum_{k=1}^{I_2} \sum_{l=1}^{I_3} |S(i, k, l)|^2} \quad (2-8)$$

这些值具有顺序性^[19]，即：

$$\sigma_1^{(1)} \geq \sigma_2^{(1)} \geq \sigma_l^{(1)} \quad (2-9)$$

其他模态下的奇异值与此类似，也具有顺序性。下面给出证明。

证明 这里给出计算正交因子矩阵和张量 \mathbf{S} 的方法。

首先计算张量 \mathbf{A} 在各个模态下展开后所得矩阵的 SVD 分解：

$$\mathbf{A}_{(i)} = \mathbf{U}^{(i)} \Sigma^{(i)} (\mathbf{V}^{(i)})^T, i=1,2,3 \quad (2-10)$$

然后计算张量 \mathbf{S} ，其定义如下：

$$\mathbf{S} = \mathbf{A} \times_1 (\mathbf{U}^{(1)})^T \times_2 (\mathbf{U}^{(2)})^T \times_3 (\mathbf{U}^{(3)})^T \quad (2-11)$$

剩下只需证明张量 \mathbf{S} 的切面两两相互正交，并且在 i -模态下的奇异值具有（非减）顺序性。

全正交的张量 \mathbf{S} 一般被称为核心张量。HOSVD 的直观图像如下图所示：

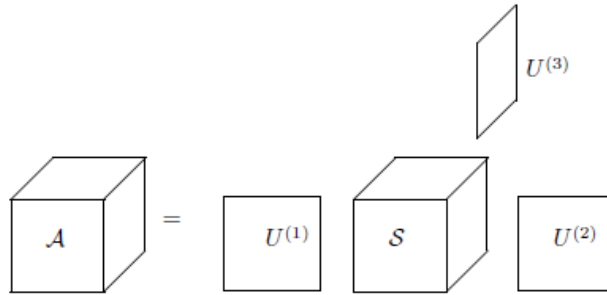


图 2- 4 HOSVD

上面的 HOSVD 分解也可以写作如下形式：

$$A_{ijk} = \sum_{p=1}^I \sum_{q=1}^m \sum_{s=1}^n u_{ip}^{(1)} u_{jq}^{(2)} u_{ks}^{(3)} S_{pqs} \quad (2-12)$$

其含义如下：整个张量的元素的变化情况，由元素 S_{pqs} 和奇异值向量 $u_p^{(1)}, u_q^{(2)}, u_s^{(3)}$ 共同决定^[20]。

HOSVD 的计算十分直观，可以由下面的 Matlab 代码直接实现。


```

function [U1,U2,U3,S,s1,s2,s3]=svd3(A);
% Compute the HOSVD of a 3-way tensor A

[U1,s1,v]=svd(unfold(A,1));
[U2,s2,v]=svd(unfold(A,2));
[U3,s3,v]=svd(unfold(A,3));

S=tmul(tmul(tmul(A,U1',1),U2',2),U3',3);

```

其中，函数 $\text{tmul}(A, X, i)$ 用来计算张量 A 在 i -模态下与矩阵 X 相乘所得的张量，即 $A \times_i X$ 。

如果令矩阵 V 是与 U_i 维度相同的正交阵，则下面的等式：

$$S \times_i U^{(i)} = S \times_i (U^{(i)} V V^T) = (S \times_i V^T) \times_i U^{(i)} V \quad (2-13)$$

表面上看，HOSVD 分解似乎并不唯一。然而，在这个线性变换中，核心张量 S 发生了改变，并且破坏了 i -模态下奇异值所具有的顺序性。所以，本质上 HOSVD 是唯一的。

在一些实际的应用中^[21]，可能出现张量某一个模态的维度远远大于其他所有模态维度的乘积的情况。例如，假设对张量 $A \in R^{l \times m \times n}$ ，有 $l > mn$ ，则可以证明，核心张量满足如下条件：

$$S(i, :, \neq) \neq 0, i \quad (2-14)$$

我们可以忽略核心张量那些全部为零的切面，将 HOSVD 重新写成精简的形式，即：

$$A = \hat{S} \times_1 \hat{U}^{(1)} \times_2 \hat{U}^{(2)} \times_3 \hat{U}^{(3)} \quad (2-15)$$

其中 $\hat{S} \in R^{mn \times m \times n}$ ，并且 $\hat{U}^{(1)} \in R^{l \times mn}$ 。

2.3 用 HOSVD 获得一个张量的近似

使用 SVD，矩阵可以表示为一系列的（矩阵的）秩-1 的量的和的形式^[22]。根据张量-矩阵的乘法的定义，我们可以推导出一个类似的关系式，来获得一个张量的表示：张量 $A \in R^{l \times m \times n}$ 可以写成一系列矩阵与奇异值向量的乘积的和的形式：

$$A = \sum_{i=1}^n A_i \times_3 u_i^{(3)}, A_i = S(:, :, i) \times_1 U^{(1)} \times_2 U^{(2)} \quad (2-16)$$

其中， $u_i^{(3)}$ 是矩阵 $u^{(3)}$ 的第 i 个列向量。 A_i 既可以看做一个 $R^{m \times n}$ 的矩阵，也可以看成是一个 $R^{m \times n \times 1}$ 的张量。该表示形式的直观图如下图所示：

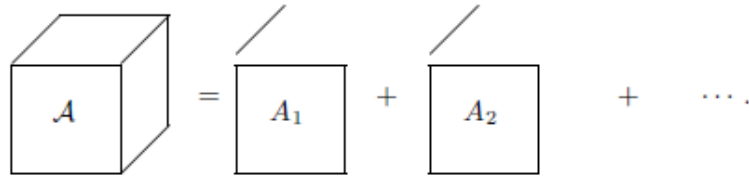


图 2- 5 HOSVD 近似表示张量

其他模态下的表示与此相似。

容易证明不同的矩阵 A_i 之间相互具有正交性，即内积为零：

$$\langle A_i, A_j \rangle = \text{tr} \left[U^{(2)T} S(:, :, i) (U^{(1)} U^{(1)T}) S(:, :, j) U^{(2)} \right] = 0, \quad (2-17)$$

现在，我们可以将张量的该种表示形式解释如下：张量 A 在第三模态下的全部切面，我们可以将其视为一组正交的基底 $(A_i)_{i=1}^{r_3}$ ，其中 r_3 是第三模态下张量 A 的正奇异值数量。可以得到：

$$A(:, :, j) = \sum_{i=1}^{r_3} z_i^{(j)} A_i \quad (2-18)$$

其中 $z_i^{(j)}$ 是 $u_i^{(3)}$ 的第 j 个元素，并且这里也得到了一个 A_i 的由正交因素矩阵组成的分解：

$$A_i = S(:, :, i) \times_1 U^{(1)} \times_2 U^{(2)} \quad (2-19)$$

并且，由于核心张量 S 在每个模态下的奇异值都具有顺序性^[23]，因此，对于每个切面 $S(:, :, i)$ ，其“质量”都集中在其左上角。

2.4 小波变换的基本思想

2.4.1 小波简介

小波变换是一个数学分支，是法国地球物理学家 Morlet 在上世纪 80 年代作为一种用于信号分析的数学工具提出的，后来又有法国学者将小波理论引入了工程领域。传统的傅里叶变化是一种全局变换，要么全在时域，要么全在频域，无法表述时频域，所以小波变换很好的解决和处理了这个问题，使得它可以用来处理非平稳信号。小波变换是一种时间-尺度分析法，在时域和频域都有很强的表达信号局部特征的能力^[24]。在低频部分有较高的频域分辨率和较低的时间分辨率。同样的，在高频部分具有高的时间分辨率和低的频率分辨率，这种多分辨的观点是小波变换的基础。Gabor 在 1946 年提出了 Gabor 变换，Burt 在 1982 年提出了图像金字塔、子带编码等都丰富了小波变换的应用意义，促进了小波变换的进一步发展。目前小波理论已经由一维发展到了多维，它是一种全新的信号处理分析方法，被称为多分辨率理论的基础。多分辨率理论涉及多个分辨率下的信号表示和分析，这种方法的明显优势是：在某种分辨率下无法检测的特性在另一种分辨率下将很容易检测。小波的本质就是一个滤波过程，若是把不同分解尺度上信号的能量求解出来，则可以将这些能量值按尺度顺序排列形成特征向量供识别使用^[25]。

小波分析的应用领域十分广泛，它包括：数学领域的许多学科；信号分析、图像处理；量子力学、理论物理；军事电子对抗与武器的智能化；计算机分类与识别；音乐与语言的人工合成；医学成像与诊断；地震勘探数据处理；大型机械的故障诊断等方面；例如，在数学方面，它已用于数值分析、构造快速数值方法、曲线曲面构造、微分方程求解、控制论等。在信号分析方面的滤波、去噪声、压缩、传递等。在图像处理方面的图像压缩、分类、识别与诊断，去污等。在医学成像方面的减少 B 超、CT、核磁共振成像的时间，提高分辨率等。

2.4.2 小波变换的基本理论

小波变换从定义的角度来分，可以分成连续和离散两大类。而连续小波又可以分成：单尺度小波变换和多尺度小波变换^[26]。

连续小波变换：所谓小波函数 $\phi(t)$ ，它局部地存在于时域轴上，其波形可在特定的位置上发生变化：当对低频信号分析时，小波函数可以放大 a 倍；当对高频信号分析时，小波函数可以缩小 a 倍。小波函数 $\phi_a(t)$ 是具有震荡特性、能迅速衰减到零的一类函数，可表示为：

$$\phi_a(t) = \frac{1}{\sqrt{a}} \phi\left(\frac{t-\tau}{a}\right) \quad (2-20)$$

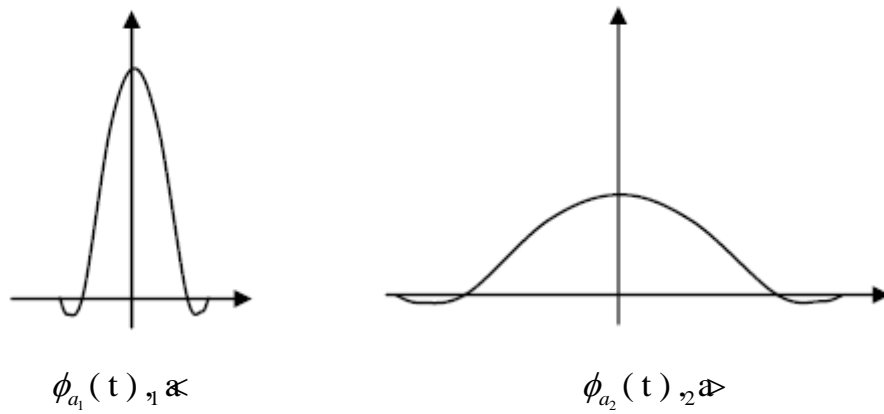


图 2- 6 小波函数曲线图

连续小波的定义：设 $x(t) \in L^2(\mathbb{R})$ 是平方可积函数， $\phi(t)$ 是基小波，则有：

$$WT_x(a, \tau) = \frac{1}{\sqrt{a}} \int x(t) \phi\left(\frac{t-\tau}{a}\right) dt \quad \text{称为 } x(\tau) \text{ 的连续小波变换 (CWT), 公式中}$$

$WT_x(a, \tau)$ 与参数 a 和 τ 相关，其中 a 称为尺度因子， τ 反映了小波函数在变换中的位移，且 $\int_{-\infty}^{+\infty} \phi(t) dt = 0$ ，其中 $\phi(t)$ 是带通滤波器。

连续小波变换具有如下性质^[27]：（1）小波变换是一个满足能量守恒方程的线形运算，它把一个信号分解成对空间和尺度的独立贡献，同时又不失原信号所包含的信息；

（2）小波变换相当于一个具有放大、缩小和平移等功能的数学显微镜，通过检查不同放大倍数下信号的变化来研究其动态特性；（3）小波变换不一定要要求是正交的，小

波基不唯一。小波函数的时宽-带宽积很小，且在时间和频率轴上都很集中，即展开系数的能量很集中；(4) 小波变换巧妙地利用了非均匀的分辨率，较好地解决了时间和频率分辨率的矛盾；在低频段用高的频率分辨率和低的时间分辨率（宽的分析窗口），而在高频段则用低的频率分辨率和高的时间分辨率（窄的分析窗口），这与时变信号的特征一致；(5) 小波变换将信号分解为在对数坐标中具有相同大小频带的集合，这种以非线性的对数方式而不是以线形方式处理频率的方法对时变信号具有明显的优越性；(6) 小波变换是稳定的，是一个信号的冗余表示。由于 a 、 b 是连续变化的，相邻分析窗的绝大部分是相互重叠的，相关性很强；(7) 小波变换同傅立叶变换一样，具有统一性和相似性，其正反变换具有完美的对称性。小波变换具有基于卷积和 QMF 的塔形

离散小波变换的定义：设 $\phi(t) \in L^2(\mathbf{R})$ ， $\phi_{jk}(t) = a^{-j/2} \phi(a^{-j}t - k)$ ，若 $\{\phi_{j,k}\}_{j,k \in \mathbf{Z}}$ 称为 $L^2(\mathbf{R})$ 的标准正交基，则称函数 ϕ 为正交小波， $\{\phi_{j,k}\}_{j,k \in \mathbf{Z}}$ 为正交小波基，称 $W_j = \text{span}\{\phi_{j,k}(t)\}_{j,k \in \mathbf{Z}}$ 为小波子空间。由内积空间理论可知，通过空间中的基函数的线性组合可把空间中的任意函数展开成级数形式。所以在小波分析中，当 $\{\phi_{j,k}\}_{j,k \in \mathbf{Z}}$ 为 $L^2(\mathbf{R})$

的规范正交基时，对于任意的 $L^2(\mathbf{R})$ 都可以用小波基展开成小波级数：

$$x = \sum_{j,k \in \mathbf{Z}} d_{j,k} \phi_{j,k} \quad (2-21)$$

其中小波系数 $d_{j,k}$ 可通过 x 和 $\phi_{j,k}$ 的内积来计算：

$$d_{j,k} = \langle x, \phi_{j,k} \rangle = \int_{\mathbf{R}} x \phi_{j,k} dk \quad (2-22)$$

从连续小波变换的讨论中我们可以看出，对一维信号 $x(t)$ 作小波变换的结果为二维的 $WT_x(a, \tau)$ ，其信息是有冗余的。从节减小计算量的角度来看，一般采用离散尺度和位移来计算小波变换^[28]。通常小波变换尺度是按照幂级数作离散化的：若令 a 依次取 $1, 2, 2^2, 2^3 \dots$ 则对应的小波基函数为 $\phi_{jk}(x) = 2^{-j/2} \phi(2^{-j}t - k)$ 。所以离散小波变换可以表示为：

$$WT_x(j, k) = \int x(t) \phi_{jk}(t) dt \quad (2-23)$$

通常情况下，时间 t 都是以离散的形式出现的。

3 高阶张量奇异值分解与小波分析相结合的人脸识别方法

本章详细描述了用小波分析与高阶张量奇异值分解算法结合的表情无关的人脸识别算法的基本方法和流程。首先利用简单而有效的 db2 小波函数处理人脸图像，然后把人脸识别问题转换到张量领域，利用处理后的人脸图像进行高阶张量奇异值分解来提取特征，进行人脸图像的识别。最后在此基础上对算法进行了优化。

3.1 db2 小波函数处理人脸图像

db2 小波函数的概念：Daubechies 函数是由世界著名的小波分析学者 Inrid Daubechies 构造的小波函数，除 db1(即 haar 小波)外，其它小波没有明确的表达式，但转换函数 h 的平方模是很明确的。

假设 $P(y) = \sum_{k=1}^{N-1} C_k^{N-1+k} y^k$ ，其中 C_k^{N-1+k} 为二项式的系数，可以得到

$$|m_0(\omega)|^2 = \left(\cos^2 \frac{\omega}{2} \right)^N P \left(\sin^2 \frac{\omega}{2} \right) \quad (3-1)$$

$$\text{其中, } m_0(\omega) = \frac{1}{\sqrt{2}} \sum_{k=0}^{2N-1} h_k e^{-ik\omega}$$

Daubechies 小波具有如下性质：小波函数 φ 和尺度函数 ϕ 的有效支撑长度为 $2N-1$ ，小波函数 φ 的消失矩阶数为 N ；dbN 大多数不具有对称性；对于有些小波函数，不对称性是非常明显的；正则性随着序号 N 的增加而增加。函数具有正交性。

Daubechies 小波函数提供了比 Harr 函数更有效的分析和综合。Daubechies 系中的小波基记为 dbN, N 为序号，且 $N=1,2,\dots,10$

db2 小波在数字图像分析中有很好的应用，它可以用于边界的处理与滤波、时频分析、信噪分离与提取弱信号等。在这里，我们可以认为人脸图像受到外界拍摄条件的影响，有许多次要的信息集中的高频部分，所有需要对人脸图像进行 db2 小波变换，提取其中的低频成分，舍弃其中的高频成分。这样不仅可以去除多余的高频成分对算

法识别精读的影响，还可以缩小图片，减小计算量，大大提高了算法的效率。在 matlab 中，此步骤可以通过下面的代码来实现：

```
clear all
clc
close all
X = imread('D:\test\ORL\s1_1.bmp');%读入图像
[C,S] = wavedec2(X,2,'db2');%进行 2 层小波分解
ca2 = appcoef2(C, S, 'db2', 2);%提取 2 层的低频系数
ca1 = appcoef2(C, S, 'db2', 1);%提取 1 层的低频系数
[h2, v2, d2] = detcoef2('all', C, S, 2);%提取 2 层的所有高频系数
[h1, v1, d1] = detcoef2('all', C, S, 1);%提取 1 层的所有高频系数
cah1=[ca1,v1;h1,d1];
cah2 = [ca2, h2; v2, d2];
%显示原图像
figure;
imshow(X);
title('原图像')
%显示一级小波分解图
figure;
subplot(1,2,1)
imshow(cah1,[]);
title('一级小波分解图')
%显示二级小波分解图
subplot(1,2,2)
imshow(cah2,[]);
title('二级小波分解图')
%将小波分解后的低频系数保存用于算法后续步骤
CA1 = uint8(ca1);
imwrite(CA1, '1.bmp', 'bmp');
```


%ca1 尺度 1 上的低频系数

%h1 尺度 1 上的水平高频系数

%v1 尺度 1 上的垂直高频系数

%d1 尺度 1 上的对角线高频系数

经小波分解后的图像与原图像的对比如下图 3-1 所示



图 3- 1 小波分解前后图像对比

经过 db2 小波处理的图像具有原图像的重要特征，并且脱离了无用的信息，可以用小波处理后的图像进行高阶张量奇异值分解

3.2 将人脸识别问题转换到张量领域

在人脸识别问题中，需要有数据库和被测图片两部分的数据^[29]。在基于 SVD 的人脸识别算法中，数据库有两重意义：一是建立人脸特征的线性空间，二是获得数据库中每个人在该线性空间下的坐标。有了这些数据之后，只需要计算被测图片在线性空间下的坐标，并且与数据库中已知人物的坐标进行比对，如果误差小于某一给定的阈值，则认为匹配成功，否则，被测人物不在数据库中。下面是具体的实现过程。

首先，数据库是一个图片的集合，假设其中的图片属于 n_p 个人物。每一张图片是一个矩阵，其大小为 $m_{i_1} \times m_{i_2}$ ，我们令 $m_{i_1} \times m_{i_2} = n_i$ 。现在，把一张人脸图像用 db2 小波处理后，把形成的新的图像所有列向量按照从左到右的顺序从上到下拼接起来，形成一个列向量。以下所说的图皆指经过小波处理后的图片。每一张图片就变成了一个长度为 n_i 的单一向量。进一步地，再假设数据库中，每个人拍摄 n_e 张不同表情

的面部照片（在这里，为了推导过程的简化，将拍摄角度、光照等等不同的条件都视作不同的表情）。通常情况下，我们有 $n_i \geq 5000$ ，并且，可以认为 $n_i > n_e n_p$ 。这样，整个数据库就可以存储在一个张量 A 中， $A \in R^{n_i \times n_e \times n_p}$

在这里，把该张量的三个模态分别称为图像模态（第一模态），表情模态（第二模态）还有人物模态（第三模态）。

如果拥有更多的图片，比如每个人在不同拍摄角度，不同光照条件下等拍摄的照片，那么整个数据库可以表示为一个更高维的张量（比如五维）。在这里，为了简化讨论，数据库中同一个人的照片只有表情的变化，于是我们将研究范围限定在了三维张量。不过，三维的情况到更高维情况的推广时十分直接的。此外，使用得另一个简化就是忽略色彩，全部图片都使用灰度图^[30]并且全部先经过小波处理。

3.3 应用 HOSVD 提取人脸特征

接下来，将 HOSVD 应用到我们的数据库张量 A 。首先，规定记号的规范。为了方便说明以及明确定义，假设张量 A 的各个模态的顺序与上面的介绍相同。不过，为了强调各模态的顺序具有任意性，这里使用 x_e 记号来代表矩阵与张量在表情-模态下进行乘法，其他模态的记号与此相似。在真正使用 HOSVD 之前，要使用第一个简化，精简的 HOSVD。首先，假设 $n_i \gg n_e n_p$ ，根据精简的 HOSVD，有

$$A = S \times_i F \times_e G \times_p H \quad (3-2)$$

其中， $S \in R$ 是核心张量， G 与 H 都是正交阵，而 F 是列正交阵。

为了不同的使用目的，可以将 HOSVD (3-2) 进行不同的解释，则得到不同的结果^[31]。首先有关系式：

$$A = D \times_e G \times_p H \quad (3-3)$$

其中， $D = S \times_i F$ 。直观图如下所示：

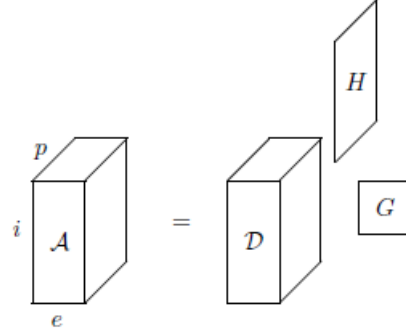


图 3- 2 HOSVD 的其中一种解释

现在，根据矩阵与张量相乘的定义，考虑第二模态，也即表情模态下的相乘：

$$(D \times_e G)(i_1, j, i_3) = \sum_{k=1}^{n_e} g_{j,k} d_{i_1,k,i_3} \quad (3-4)$$

如果将表情固定为某个确定的值，例如， $j = e_0$ ，那么相当于只使用了矩阵 G 的第 e_0 行^[32]。接下来在人物模态下进行类似的工作，则有

$$A(:, e_0, p_0) = D \times_e g_{e_0} \times_p h_{p_0} \quad (3-5)$$

其中， h_{p_0} 代表 H 的第 p_0 个行向量， g_{e_0} 代表 G 的第 e_0 个行向量。下图形象的展示了式（3-5）中的计算：

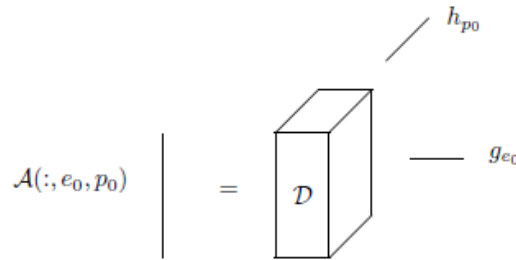


图 3- 3 HOSVD 的另外一种解释

将上述内容总结如下：

如果我们将张量 D 分别与向量 h_{p_0} 和 g_{e_0} 在其对应的人物模态和表情模态相乘，那么我们就可以得到第 p_0 个人物的第 e_0 个表情的图片。而且可以发现，对于同一个表情 e_0 ，当人物变化时，向量 g_{e_0} 不变；同理，对于同一个人物 p_0 ，当表情变化时，向量 h_{p_0} 不变。由此可知，在我们的模型中，向量 g_{e_0} 唯一的代表了表情 e_0 ，而向

量 h_{p_0} 唯一的代表了人物 p_0 ，并且两者相互独立，互不影响。具体的计算过程如下式所表示：

$$D \times_e g \times_p h \quad (3-6)$$

接下来，开始考虑人脸特征的提取。为了推导需要，将 HOSVD (3-6) 写成如下形式：

$$A = C \times_p H, \quad C = S \times_e F \times_e G \quad (3-7)$$

在上式中，如果将表情的值固定，则对于某个确定的 e 值，有

$$A(:, e, :) = C(:, e, :) \times_p H \quad (3-8)$$

在这里， $A(:, e, :)$ 和 $C(:, e, :)$ 是张量的切面，显然可以将它们看作矩阵，这里记为 A_e 和 C_e 。于是，对于任意的表情 e ，都有如下线性关系

$$A_e = C_e H^T, \quad e = 1, 2, \dots, n_e \quad (3-9)$$

相同的正交矩阵 H 出现在了全部的 n_e 个线性关系里。若令 $H^T = (h_1 \cdots h_{n_e})$ ，则式 (3-9) 中，其第 p 列可以写成如下式

$$a_p^{(e)} = C_e h \quad (3-10)$$

接下来，通过仔细分析式 (3-9) 与 (3-10)，对这两个式子作出解释如下：

A_e 的第 p 列所包含的数据，是第 p 个人物的第 e 张表情的图片。接下来该列向量由矩阵 C_e 和列向量 h_p 完全决定。此外，对于同一个表情 e ，当人物 p 变化时，矩阵 C_e 不变。同理，对于同一个人物 p ，当表情 e 变化时，向量 h_p 不变。那么，我们可以认为，矩阵 C_e 的所有列向量，共同组成了表情 e 的一组基底（表情坐标系）。也就是说，数据库中所有人物的第 e 个表情，都存在于该组基底所张成的线性空间中。那么，每一个人物在的表情 e ，都存在一个对应该组坐标系下的坐标，而这个坐标，就是 h_p 。不仅如此，在所有表情坐标系中，人物 p 的坐标都是 h_p 。这样，通过将数据组织成张量的形式并进行 HOSVD 分解，就提取出了人物的与表情无关的特征信息 h_p 。接下来，只需要提取出被测图片中的这种与表情无关的信息^[33]，并将其与数据

库中的已知身份信息进行比对, 就可以实现人脸识别。更重要的是, 由于我们提取的人物特征信息是与表情无关的, 因此可以预期该算法能够针对表情变化的情况下工作而不受影响, 获得一定的稳定性。

3.4 基于 HOSVD 的人脸识别过程

接下来考虑被测图片的特征提取与识别的过程。假设向量 $z \in R^n$ 是一张未知人物

的未知表情的照片 (该表情不在数据库的 n_e 个表情中), 称其为被测图片。目的是想要识别该张照片中的人物身份, 判断其是否在数据库中。显然, 如果该图片在数据库中, 例如, 假设它是第 p 个人物的第 e 张表情的图片, 则 z 在该表情的坐标系下的线性表示的坐标应该与向量 h_p 相等。于是, 可以通过计算该图片在每一个表情坐标系下的坐标, 并且将其与人物矩阵 H 的每一行进行比对, 来进行身份识别^[34]。如果它与 H 的任何一行的差的范数小于某一个给定的阈值, 就认为匹配成功; 或者, 它在任何一个坐标系下都无法与任何一个人匹配, 则匹配失败^[35]。

图片 z 表情 e 的坐标系下的坐标可以通过计算下面的最小二乘问题来求解

$$\min_{\alpha_e} \|C_e \alpha_e - z\|_2 \quad (3-11)$$

该算法可以通过下面的伪代码实现:

```
% z is a test image.
for e = 1, 2, ..., n_e
    Solve  $\min_{\alpha_e} \|C_e \alpha_e - z\|_2$ 
for p = 1, 2, ..., n_p
    If  $\|\alpha_e - h_p\|_2 < tol$ , then classify as person p and stop.
end
end
```

不难看出, 这个算法拥有较大的计算工作量。两重循环, 外层循环求解最小二乘问题, 内层循环求解向量范数。下面给出而最小二乘问题计算过程的一个简化。

首先，对于每一张被测图片 z ，都要计算 n_e 个最小二乘问题，其中 $C_e \in R^{n_i \times n_p}$ 。

因为 $C = S \times_i F \times_e G$ ，则对于某个确定的表情 e 有

$$C_e = FB_e \quad (3-12)$$

其中 $B_e \in R^{n_i n_p \times n_p}$ 等于 $(S \times_e G)(:, e, :)$ 。此外， $F \in R^{n_i \times n_i n_p}$ ，假设 $n_i \gg n_e n_p$ 。为了方便分析，将矩阵 F 扩展为正交方阵

现在，在范数计算式中加入因子 \hat{F}^T ，有：

$$\|C_e \alpha_e - z\|_2^2 = \|B_e \alpha_e - F^T z\|_2^2 + \|(F^-)^T z\|_2^2 \quad (3-13)$$

于是，为了计算 n_e 个最小二乘问题，我们可以先算出 $F^T z$ ，接下来再计算

$$\min_{\alpha_e} \|B_e \alpha_e - F^T z\|_2, e = 1, 2, \dots, n_e \quad (3-14)$$

其中，矩阵 B_e 的规模为 $n_e n_p \times n_p$ ，于是，式（3-14）与（3-11）相比，大大降低了计算量。更进一步地，可以计算出矩阵 B_e 的 QR 分解，继续减少计算量。于是，得到了下面的优化版本算法，降低了工作量，但其准确率不变。

% z is a test image.

Compute $\hat{z} = F^T z$.

for $e=1, 2, \dots, n_e$

Solve $R_e \alpha_e = Q_e^T \hat{z}$ for α_e

for $p=1, 2, \dots, n_p$

If $\|\alpha_e - h_p\|_2 < tol$, then classify as person p and stop.

end

end

该算法首先计算出 $F^T z$ ，然后将矩阵 B_e 进行 QR 分解，最后求解最小二乘问题^[38]，得到被测图片在表情坐标系 e 下的坐标。在不同模态下，核心张量的奇异值具有顺序性。利用这一性质，我们可以截断核心张量，得到压缩的 HOSVD，并且该压缩后的 HOSVD 仍然是原张量 A 的一个很好的近似。

3.5 基于压缩的 HOSVD 优化识别算法

在前面的分析中提到，在不同模态下，核心张量的奇异值具有顺序性^[36]。利用这一性质，我们可以截断核心张量，得到压缩的 HOSVD，并且该压缩后的 HOSVD 仍然是原张量 A 的一个很好的近似。我们定义 $F_K = F(:, 1:k)$ ，假设 k 大于 n_p ，但是远小于 n_i 。扩展矩阵使其成为正交方阵^[37]，以便于分析：

$$\hat{F} = (F_K, F^\sim), \hat{F}^T \hat{F} = 1 \quad (3-15)$$

接下来，使用与截断矩阵相类似的方法，我们截断核心张量，有

$$\hat{C} = (S \times_e G)(1:k, :, :) \times_i F_K \quad (3-16)$$

由于核心张量在各个模态下的奇异值具有顺序性，此外，可以证明，核心张量 S 与矩阵 G 在表情模态下相乘，并不影响其在图像模态下奇异值的顺序性，因此我们有

$$\|\hat{C} - c\|_F^2 = \sum_{v=k+1}^{n_i} \sigma_v^i \quad (3-17)$$

由该算式我们可以得知，只要在图像模态下，奇异值递减的速度足够快，则即使对核心张量进行了压缩，我们仍然能够获得较好的识别结果。如果我们使用 \hat{C} 代替前述算法中的 C ，则我们得到如下最小二乘问题

$$\min_{\alpha_e} \|\hat{C}_e \alpha_e - z\|_2 \quad (3-18)$$

其中， \hat{C}_e 的定义显然。于是，由式 (3-16)，我们得到 $\hat{C}_e = F_K \hat{B}, \hat{B}_e \in R^{k \times n_p}$ 。在该范数算式中加入因子 \hat{F} 则得到

$$\left\| \hat{C}_e \alpha_e - z \right\|_2^2 = \left\| \hat{B}_e \alpha_e - F_k^T z \right\|_2^2 + \left\| (F^\sim)^T z \right\|_2^2 \quad (3-19)$$

在这个“压缩”版本的识别算法中，使用 $\hat{z}_k = F_k^T z$ 替换了 $\hat{z} = F^T z$ ，因此，循环中的每一个最小二乘问题，其规模都变小了。

4 基于 HOSVD 和小波结合的人脸识别算法实现

本章的内容是算法的具体实现环节，展示了程序的基本架构和所用到的张量工具包。然后以著名的人脸数据库为例，说明后台程序的运行过程。最后为了便于编辑数据库和展示，为系统实现了图像界面。

4.1 程序架构

程序的基本框架可以用下面的流程图图 4-1 来表示：

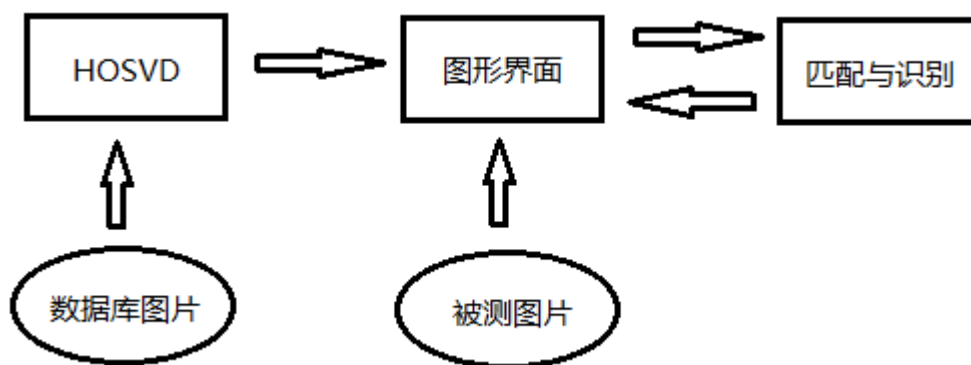


图 4-1 程序架构流程图

其中，方框代表程序，而椭圆代表数据，图片数据都是经过小波滤波的。HOSVD 程序由 Matlab 代码实现，接受人脸数据库图片作为输入，将其构建为张量并且应用 HOSVD 分解来获取数据库中人物的人脸特征，也就是 HOSVD 分解中的因素矩阵，并且将这些因素矩阵作为输出传递给前端图形界面。

图形界面程序由 C# 代码组成，接受被测图片数据以及 HOSVD 分解程序计算出的因素矩阵（数据库中人物的特征）作为输入，并且将其传递给后台的匹配与识别程序。

匹配与识别程序由 Matlab 代码组成，接受图形界面传递的 HOSVD 因素矩阵（数据库中人物的特征）以及被测图片作为输入，将被测图片中的人物特征提取后与数据库中的人物特征进行比对，并且将识别结果返回给图形界面程序。

4.2 张量工具包

本算法中包含有大量基于张量以及矩阵的运算。众所周知, Matlab 对于矩阵运算, 提供了非常强大的支持^[38]。基于以上原因, 选择 Matlab 作为主要的实验平台来进行算法的实现与验证工作。

除了支持向量与矩阵之外, Matlab 对于更高维的数组 (亦即张量) 也提供了一定的支持。然而其提供的功能并非十分完善, 尤其对于一些复杂的运算, 比如本算法中要用到的 HOSVD 等等, 并不能提供完全的支持。不过有很多张量相关工作的研究者开发了相应的扩展工具包^[39], 可以支持更多复杂的张量运算, 其中比较著名的有 N-way Toolbox 和 Tensor Toolbox 等。本实验中选用了 Tensor Toolbox, 下面重点介绍一下该工具包, 包括实验中将会使用的一些重要功能。

Tensor Toolbox 的开发者是 Bader 和 Kolda, 两人都是隶属于 Sandia National Laboratories。他们开发这个工具包的目的是, 为了扩展 Matlab 对于各种张量运算的支持。它支持稠密张量与稀疏张量, 其中内置了张量的各种基本预算, 比如矩阵与张量的乘法, 向量与张量的乘法, 张量的范数、内积等等, 也支持张量的各种分解, 尤其是对于 HOSVD 分解中求解主成分以及因素矩阵等提供了相当完善的支持。下面介绍实验中将用到的一些核心方法。

张量的创建: Tensor Toolbox 中提供了多种方式用来创建张量, 比如, 可以使用 tensor 函数创建空张量或者利用数组元素创建张量, 也可以使用 tenrand、tenzeros、tenones 等函数创建元素值经过初始化的张量 (从函数名易知初始化值)^[40]。张量具有 2 个属性, 一个是 size, 另一个是 data^[41]。执行下面的代码, 则可以得到结果如下:

```
X = tenrand([4 3 2]);
X.data
X.size
% 结果
ans(:,:,1)
    0.8147 0.6324 0.9575
    0.9058 0.0975 0.9649
```

```

0.1270 0.2785 0.1576
0.9134 0.5469 0.9706
ans(:,:,2) =
0.9572 0.4218 0.6557
0.4854 0.9157 0.0357
0.8003 0.7922 0.8491
0.1419 0.9595 0.9340
ans =
4 3 2

```

如上例所示，函数为了生成一个三维的张量，函数 `tenrand` 需要三个参数，分别指定了三个模态下的维度。然后函数会根据规则初始化元素的值。

张量的运算：Tensor Toolbox 支持多种张量运算^[42]，其中主要包括张量与张量的乘法运算（`tensor times tensor`，函数 `ttt`），还有矩阵与张量的乘法运算（`tensor times matrix`，函数 `ttm`），以及向量和张量的乘法运算（`tensor times vector`，函数 `ttv`）。下面以矩阵与张量相乘为例，进行说明：

```

rand('state',0);
X = tensor( floor( 5*rand (2,3,2)));
A = [2 1;3 2];
Y = ttm(X,A,1)
% 结果
Y isatensorofsize2x3x2
Y(:,:,1) =
    9    8   11
   14   13   18
Y(:,:,2) =
    4   10    9
    6   16   15

```

如上例所示，函数 `tmm` 表示张量与矩阵相乘，需要指定三个参数，分别是参与运算的张量，参与运算的矩阵，以及运算进行的模态。其中参与运算的张量与矩阵维度必须匹配，否则无法进行运算。

张量展开（矩阵化）：在前面的章节中，我们提到了张量的展开，亦即矩阵化。我们知道，张量的展开（以及折叠）是张量的基础且重要的操作，它是实现张量与矩阵以及向量等乘法的基础^[43]。Tensor Toolbox 提供了 `tenmat` 函数，用来将张量展开成矩阵，并且得到一个 `tenmat` 对象。该对象具有三个属性，分别为 `data`，`rindicies` 以及 `cindicies`。具体代码如下：

```
X = tensor(1:24,[3 4 2])
tenmat(X,1)
tenmat(X,2)
tenmat(X,3)
% 结果
X isatensorofsize3x4x2
X(:, :, 1) =
    1  4  7 10
    2  5  8 11
    3  6  9 12
X(:, :, 2) =
   13 16 19 22
   14 17 20 23
   15 18 21 24
ans is amatrixcorrespondingtoatensorofsize3x4x2
ans.rindices = [1] (modes of tensor corresponding to rows)
ans.cindices = [2 3] (modes of tensor corresponding to columns)
ans.data =
    1  4  7 10 13 16 19 22
    2  5  8 11 14 17 20 23
```

```

3 6 9 12 15 18 21 24

ans is amatrixcorrespondingtoatensorofsize3x4x2
ans.rindices = [2] (modes of tensor corresponding to rows)
ans.cindices = [1 3] (modes of tensor corresponding to columns)
ans.data=

    1    2    3    13 14 15
    4    5    6    16 17 18
    7    8    9    19 20 21
   10 11 12 22 23 24

ans is amatrixcorrespondingtoatensorofsize3x4x2
ans.rindices = [3] (modes of tensor corresponding to rows)
ans.cindices = [1 2] (modes of tensor corresponding to columns)
ans.data =

    1    2    3    4    5    6    7    8    9    10 11 12
   13 14 15 16 17 18 19 20 21 22 23 24

```

张量的 HOSVD 分解：事实上，张量的 HOSVD 分解并不是 Tensor Toolbox 所直接提供的一个操作。不过，我们可以利用该工具包提供的各种基本运算，求得 HOSVD 分解中的各个因素矩阵以及核心张量，最后将其相乘，则得到了所需要的 HOSVD 分解。

具体步骤是：首先，使用工具包提供的函数 `nvecs` 计算 HOSVD 分解中各个模态下的因素矩阵。其形式如下：

```
U1=nvecs(X,1,4)
```

其中，`U1` 是计算得到的结果，该函数有三个参数，第一个参数指定被分解的张量，第二个参数指定计算的因素矩阵所属的模态，第三个参数指定因素矩阵的列数（若该参数与该模态维度相同，则得到完整因素矩阵；若参数小于该模态维度，则得到的是压缩的因素矩阵）。

接下来，使用张量 X 分别在三个模态下与三个因素矩阵的逆矩阵相乘，这里使用到 `pinv`，求伪逆矩阵的函数。最后，使用 `ttensor` 函数，构建原张量的近似张量，则 HOSVD 分解完成。具体代码如下：

```

rand('state',0)
X = tenrand([4 3 2]); %<- Generate data
U1 = nvecs(X,1,4); %<- Mode 1
U2 = nvecs(X,2,3); %<- Mode 2
U3 = nvecs(X,3,2); %<- Mode 3
S = ttm( X,{ pinv(U1),pinv(U2),pinv(U3) } ); %<- Core
Y = ttensor( S,{ U1,U2,U3} ) %<- HOSVD of X
% 结果
Y is attensorofsize4x3x2

Y.core is attensorofsize4x3x2

Y.core(:,1) =

    2.8891    0.0067   -0.0058
    0.1217    0.4834   -0.1743
   -0.0902    0.3462    0.0787
    0.0081   -0.0615    0.2800

Y.core(:,2) =

    0.0378    0.6458    0.0183
    0.1159   -0.5625    0.1706
    0.0387    0.3866    0.4122
    0.3752   -0.0555    0.0168

Y.U{1} =

    0.6703   -0.2673   -0.1760    0.6695
    0.4886   -0.0239   -0.5802   -0.6512
    0.3826    0.8903    0.2444    0.0366
    0.4069   -0.3680    0.7568   -0.3554

Y.U{2} =

```

$$\begin{aligned} &0.5870 \ -0.0848 \ 0.8051 \\ &0.6728 \ -0.5020 \ -0.5434 \\ &0.4503 \ 0.8607 \ -0.2376 \\ Y.U\{3\} = & \\ &0.7053 \ 0.7089 \\ &0.7089 \ -0.7053 \end{aligned}$$

4.3 基本后台程序

4.3.1 实验数据

网络上有很多的人脸数据库，都是研究者收集发布的，可以很方便的使用。其中比较著名的有 yale 人脸数据库，orl 人脸数据库，以及 MIT、CMU 等等。这里选用 yale 人脸数据库。图片为灰度图，数据规模为：选取十个人物，每人十张表情，构成整个实验的人脸数据库。另外，每人选取一张额外的（不在实验数据库）中的表情，作为被测图片，用来验证算法的正确性^[44]。下图所示为第一个人物的部分图片（注意不同图片之间表情、光照等的不同）。



图 4- 2 部分实验数据图

4.3.2 数据库构建与特征提取

按照之前算法的推导，接下来的过程是通过数据库构建张量，将人脸识别问题转换到张量领域。具体代码如下：

```
T=tenzeros([isize etrain ptrain]);
```

```

for j=1:ptrain %number of persons
    Z=zeros(isize,etrain); %create a empty tensor
    for i=1:etrain %number of expressions
        X=imread(strcat(yaleface/s',pids(j,:),'/','eids(i)'.bmp'));
        X=imresize(X,[nrows,ncls]); %compress the image to save memory
        Y=reshape(X,ysize,1);% numel=number of elements
        Z(:,i)=Y;%add a column to Z
    end
    T(:,j)=Z; %add a slice to tensor T
end

```

两层循环，内层循环读取一张图片，并且将图片从矩阵转换为一个列向量。经过 `etrain`（即每个人的表情个数）次循环后，每个人的所有表情的图片构成了

一个矩阵（张量的切面）。外层循环将 `ptrain`（即人物个数）个人物的矩阵拼接成一个张量。这样，从数据库到张量的构建工作完成。

接下来是对张量进行 HOSVD 分解，进行人脸特征提取，获得需要的核心张量以及因素矩阵。具体代码如下：

```

F = nvecs(T,1,ptrain*etrain);%<-- Mode 1 - image mode
G = nvecs(T,2,9); %<-- Mode 2 - expression mode
H = nvecs(T,3,ptrain); %<-- Mode 3 - person mode
S = ttmt(T,{pinv(F),pinv(G),pinv(H)}); %<-- Core Tensor
A = ttensor(S,{F,G,H}); %<-- HOSVD of T

D=ttmt(S,F,1);%<-- D=S * F in Mode 1
C=ttmt(S,{F,G},[1 2]);%<-- C=S*F*G

%保存 HOSVD 的数据
save('orlfaces.mat','T','A','S','F','G','H','D')

```


至此，数据库张量的构建与特征提取完成，接下来开始进行被测图片的实际识别过程。

4.3.3 被测图片的识别

为了完成被测图片的识别，首先，需要提取出被测图片中人脸的特征，然后，将该特征与数据库中的已知人脸的特征进行对比，如果匹配，则识别成功；否则匹配失败，人物不在数据库中。具体代码如下：

```
for i=1:ptrain
    Z=imread(strcat(yaleface/s',pids(i,:),','11.bmp'));%read
    Z=imresize(Z,[nrows,ncls]);%compress it to save memory
    Z=double(reshape(Z,numel(Z),1));
    mine=1;%用来保存最接近的表情的编号
    minp=1;%用来保存最接近的人物的编号
    mindif=100;%不会超过 100 [观察结果一般都小于 2]
    for e=1:etrain% number of expressions
        Ce=double(C(:,e,:));%得到 Ce
        Alpha=Ce\z;%求最小二乘解
        for p=1:ptrain%number of persons
            res=norm(alpha-H(p,:)',2);%求出误差的范数值
            if(res<mindif)% 若新误差小于最小误差 则更新
                mindif=res;
                mine=e;
                minp=p;
            end
        end
    end
end
```

三层循环，第一层循环共 ptrain 次，对应每个人的被测图片。首先，读取被测图片，然后按照构建数据库时相同的方式，将该图片矩阵转化为列向量。接下来，第二层循环，对于所得到的列向量，求出其在 etrain 个表情坐标系中，每一个里面所对应的坐标。最后，第三层循环，用得到的坐标与矩阵 \mathbf{H} 中的每一列，亦即数据库中每一个人的坐标进行比对，求出误差。保存误差最小值（在这里，为了简化讨论，认为被测图片与误差最小者身份匹配）。

最后，利用 Matlab 的 `plot` 函数，将所得结果输出。代码如下所示：

```
dgh=ttv(D,{G(mine,:)',H(minp,:)'},[2 3]);% 这种方式等价与两次与向量相乘
imagedata=uint8(reshape(double(dgh),nrows,ncls));% 从 SVD 构建图片
subplot(2,ptrain,i),imshow(Z) % 输出被测图片
subplot(2,ptrain,ptrain+i),imshow(imagedata) % 输出匹配图片
```

输出结果如下图：



图 4-3 yale 数据库识别结果

其中，第一行是被测图片，第二行是该算法求出的数据库中与被测图片最接近的图片。

从该结果我们可以看出，十个被测人物中，九个识别正确，只有一个错误，该算法的准确率还是比较高的。

4.4 图形界面

利用 Matlab 实现了基本的后台算法后, 下一步实现一个基于 C#的图形界面。之所以实现图形界面, 首先是为了便于操作演示。另外, 在图形界面中内嵌了图片数据库的编辑整理功能, 也是为后面在项目中数据库的编辑功能打下基础。因为有时候人脸数据库中的图片分类、命名等杂乱无章, 尤其当数据量较大时, 如果手工整理则耗时耗力。此外, 对于已经创建好的数据库, 也可能由于人物的增加、减少、表情数量的改变等而需要经常性的编辑。这里创建一个可复用类库用来管理图片数据库, 并将其编辑整理功能整合到图形界面中。

如图所示即为该程序的图形界面。图形界面的菜单栏包括以下几个选项卡: 功能、数据库、历史记录、设置、帮助。可以在左侧通过选择“...”图标来选择要识别的照片, 之后可以形成一个照片的预览, 点击开始识别, 就可以连接到后台数据库, 查找数据库中匹配的图片。识别结果在右侧框中出现。本系统还提供了生成记录以及查看历史记录等功能。



图 4-4 图形界面

由于图形界面基于 C#实现, 其代码比较简单, 在这里不再过多赘述。下面详细描述 C#与 Matlab 混合编程的具体方法。

虽然图形界面本身的实现并不复杂,然而,完成图形界面后,将其与后台的 Matlab 算法链接起来,是比较困难的。

C#与 Matlab 的混合编程中,若希望 C#代码能够利用 Matlab 组件,则对 Matlab 的处理,通常有生成 COM 组件,生成.net 组件等方式。这里采用后者。

利用 Matlab 代码生成.net 组件的过程简要概括如下:

- 1、安装相应的 C#编译器。
- 2、执行 `mbuild -setup` 命令配置 Matlab 编译环境。
- 3、执行 `deploytool` 命令,创建工程,添加类与函数,以及相关资源文件,编译生成.net 组件 (dll 文件)。

另一方面,C#代码若想利用 Matlab 生成的.net 组件,也许一定的配置过程,简要概括如下:

- 1、引用 `MWArray.dll` 文件,该文件包含了 Matlab 与 C#之间用于数据传递的中间数据结构 `MWArray` 等的定义。
- 2、引用上一步中生成的.net 组件,这里是 `myfunc.dll`。
- 3、在 C#文件中手动加入下面的代码:

```
using myfunc;  
using MathWorks.MATLAB.NET.Utility;  
using MathWorks.MATLAB.NET.Arrays;
```

然后就可以在 C#代码中使用自己定义的 Matlab 函数 `myfunc` 了。

不过,在该函数真正能够运行之前,还有一个问题。

这里自定义的人脸识别算法 `myfunc` 函数,依赖于一定的外部输入,包括以下两个方面:一是 `HOSVD` 分解得到的因素矩阵,亦即识别过程所需要的,从人脸数据库中提取出的人脸特征;二是识别过程所需要的被测图片。

虽然在 Matlab 的 `deploytool` 中^[45],提供了将外部输入整合到生成的 dll 文件中的功能,然而一来,这样将程序的接口固定化之后,不便于输入的修改,导致程序难于使用;二来,在实际运行中发现使用该方法后,C#代码无法正确处理变量类型,导

致程序出错，无法执行。解决方案就是把该函数的输入与 C#程序的接口合并，统一由 C#程序读取外部输入，然后传递给 myfunc 函数进行运算。具体过程如下。

针对第一部分输入，亦即 HOSVD 分解得到的因素矩阵，首先在 Matlab 中将计算结果输出为文本文件（具体代码在这里略去），然后在 C#程序中读取该文本文件中的数据并存储，具体代码如下：

```
double[,] c = newdouble[8736, 100];
StreamReader objreader2 = newStreamReader("c.txt");
string line2 = "";
for (int i = 0; i < 10; i++)
    for (int j = 0; j < 8736; j++)
        {
            line2 = objreader2.ReadLine();
            string[] tmp2 = line2.Split('\t');
            for (int k = 0; k < 10; k++)
                {
                    c[j, i * 10 + k] = double.Parse(tmp2[k]);
                }
        }
```

该段代码读取因素矩阵 C 并且存储在数组中，其他因素矩阵的读取与此类似。

针对第二部分输入，亦即被测图片，处理方法与前述类似，具体代码如下：

```
Bitmap bmp = newBitmap(picname);
int width = bmp.Width;
int height = bmp.Height;
double[,] matrix = newdouble[height, width];
for (int j = 0; j < width; j++)
    {
        for (int i = 0; i < height; i++)
```

```

    {
        Color cl = bmp.GetPixel(j, i);
        matrix[i, j] = cl.R;
    }
}

```

该段代码读取图片数据，并存入相应数组中。

有了函数的输入数据，接下来我们可以调用函数 `myfunc`，接受并且分析其返回结果，具体代码如下：

```

myfunc.Class1 my1= new myfunc.Class1();

MWEArray a = (MWEArray)(my1.myfunc((MWNumericArray)h, (MWNumericArray)c,
(MWNumericArray)matrix));

```

```

MWNumericArray temp1 = (MWNumericArray)a[1];
double[,] res1 = (double[,])temp1.ToArray(MWEArrayComponent.Real);

```

```

MWNumericArray temp2 = (MWNumericArray)a[2];
double[,] res2 = (double[,])temp2.ToArray(MWEArrayComponent.Real);

```

```

string person = "subject";
string filename = "yalefaces/subject";
if (res2[0, 0] <= 9)
    {
        filename += "0";
        filename += res2[0, 0].ToString();
        person += "0";
        person += res2[0, 0].ToString();
    }
else
    {
        filename += "10";
    }

```

```
        person += "10";  
    }
```

首先，定义一个新的 `myfunc.Class1` 对象（Matlab 函数所在的类的对象），然后利用上面获得的数据（因素矩阵 `h`，`c`，被测图片 `matrix`）作为输入，调用 `myfunc` 函数，利用中间数据类型 `MWArray` 接受函数返回数据，分析数据获得最终的识别结果。这样，就完成了 C#图形界面与 Matlab 后台算法的链接。

5 算法实验和结果分析

5.1 内存分析

在开始进行实验结果的准确性分析之前，需要描述一下在实验中遇到的一个问题：内存不足。

前述实验一的代码在运行的时候，遇到了内存不足的问题，具体情况如下：

```
Error using '
Out of memory. Type HELP MEMORY for your options.

Error in eigs>ishermitian (line 1392)
tf = isequal(A,A');

Error in eigs/checkInputs (line 469)
    issymA = ishermitian(A);

Error in eigs (line 94)
[A,Amatrix,isrealprob,issymA,n,B,classAB,k,eigs_sigma,which, ...

Error in tensor/nvecs (line 42)
[u,d] = eigs(Y, r, 'LM', eigsopts);

Error in hosvd (line 37)
F = nvecs(T,1,100); %<-- Mode 1 - image mode
```

注意最后一行，可以看到，出错部分的代码的功能是利用 Tensor Toolbox 提供的 `nvecs` 函数求解因素矩阵。观察该函数代码，发现其中包含如下内容：

```
Xn = double(tenmat(X,n));
```

```
Y = Xn*Xn';
```

该函数将待分解的张量展开成矩阵后，求出其协方差矩阵（即，令矩阵与自身转置相乘。）由于该矩阵是数据库张量在图像模态下的展开，因此相乘后，所得到的矩阵是一个行数与列数都为 n_i 的方阵。在这个实验中，虽然原始数据规模并不大（大概只有 10Mb 左右），然而由于图片模态下的维度其大小大概是表情模态与人物模态维度乘积的 100 倍，因此，最后代码所需要的内存空间峰值达到了 1Gb 的规模，再加上 Matlab 运行时本身需要的内存空间，其总和已经接近甚至超过了可用内存的极限（在 32 为 win7 系统中，系统限制每个进程最大可用内存为 1.5G 左右），因此出现了上面的内存不足问题。解决方法有两个，一是压缩图片大小，降低原始数据量。另一个方案是开启 3GB 开关，这样每个进程可用内存增加到 3G，足以满足程序的需求。

实验一的内存问题虽然解决了，然而后续还将进行更大规模的实验。实验一的数据规模为 10*10 共 100 张图片，然而后续两个实验分别基于 orl 人脸数据库以及 yaleB 人脸数据库，问题规模分别为 40*10 以及 40*60 的规模。实验一中内存空间尚且勉强满足要求，一旦数据增加的后续实验的规模，内存空间的问题该如何解决？

为了处理这个问题，需要仔细分析该算法中，决定空间复杂度的真正因素。在实验一中，内存空间所以会急剧上升为原始数据量的 100 倍，是因为实验一中图像模态维度远大于其他两个模态维度的乘积。然而，在后续实验中，随着图像模态与其他两个模态维度乘积的比值的减小，内存空间需求与原始数据的比也迅速下降。实际上通过分析可以发现，算法的空间复杂度，唯一决定于原始数据中每张图片的像素个数，与人物个数以及表情个数完全无关（除非表情模态与人物模态维度的乘积大于图像模态的维度）。因此，至少在理论分析中，后续基于 orl 与 yaleB 人脸数据库的两个实验，其所需内存空间完全与实验一相当，在当前实验平台运行完全不会遇到内存空间不足的问题。有了这样一个理论分析基础，我们就可以进行后续的实验。

以上分析是基于原始的 HOSVD 算法。然而，使用小波处理图像后，由于图像的大小变为原来的四分之一，所以算法相应的空间复杂度也降低，则内存不足的问题可以避免。

5.2 算法实验

5.2.1 表情无关 HOSVD 与 PCA 算法识别准确度对比实验

在前述的实验中,已经看到了采用 HOSVD 对 yale 数据库识别的结果如下图:

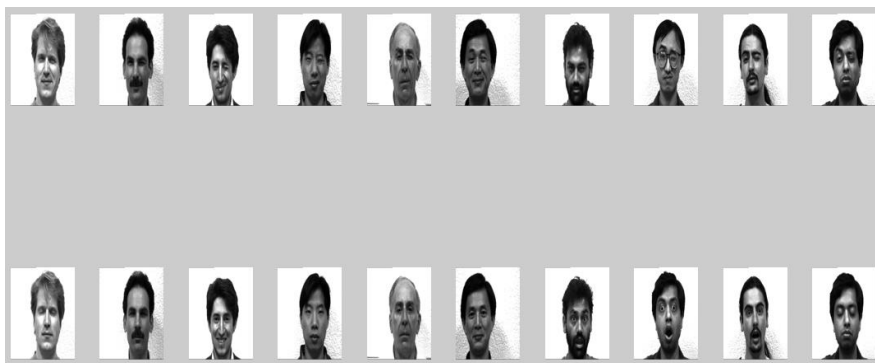


图 5- 1 yale 数据库实验结果

可以看到,在十个人物中,只有一个人物识别错误,其余就个人全部识别正确,正确率高达 90%,而且被测图片的表情都是人物不在数据库中。可以看出,该算法的准确率较高,而且在面对人物图片拍摄条件的变化时,也能获得一定的稳定性。

再用 PCA 算法,选取任务不在数据库中的表情进行识别,可以得到下面的识别结果

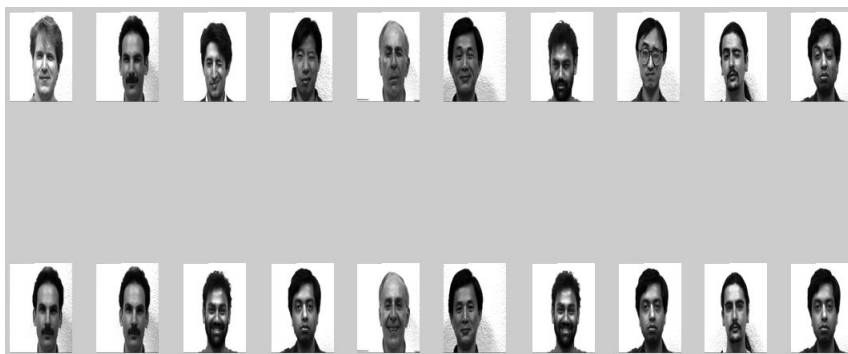


图 5- 2 PCA 算法识别 yale 数据库实验结果

受到表情和光照发型等各种因素的影响, yale 数据库中人脸匹配的正确率只有 60%。

为了验证算法的准确性,采集了实验室同学的图片进行识别,发现在多个表情下,能够全部识别成功。



图 5- 3 HOSVD 识别实验室同学图片实验结果

在用 yale 数据库进行实验后，基于数据规模更大的 orl 数据库和 yaleB 数据库，又进行了两组实验，使用表情无关的 HOSVD 对 orl 数据库识别的结果（部分截图）如下所示：



图 5- 4 orl 数据库识别结果

实验结果具体如下：

在 orl 数据库的实验中，40 个被测人物，其中 4 个错误，36 个正确，正确率达到 90%，基本与实验一保持一致。在实验三使用 yaleB 数据库的实验中，40 个人物全部识别正确。使用 PCA 对 orl 数据库中图片识别的正确率为 70%，明显低于 HOSVD 的识别精度。YaleB 数据库也取得了相似的结果。

通过简单的分析，可以认为，本算法能够在人脸图片拍摄条件发生变化时，获得较好的稳定性，该识别算法具有较高的准确率。

5.2.2 优化的 HOSVD 和原始 HOSVD 对比实验

在前述算法推导过程中，我们提到，可以通过在最小二乘问题的范数计算式中加入因子 \hat{F}_T ，来缩小问题规模，简化计算量。具体代码如下：

```

for i=1:ptrain% number of persons

    Z=imread(strcat(yaleface/s',pids(i,:),','11.bmp'));%read
    Z=imresize(Z,[nrows,ncls]);%compress it to save memory
    Z=double(reshape(Z,numel(Z),1)); % reshape to a vector
    Zhat=F'*z;% z(hat) -> 100*1

    mine=1;%用来保存最接近的表情的编号
    minp=1;%用来保存最接近的人物的编号
    mindif=100;%不会超过 100 [观察结果一般都小于 2]

    for e=1:etrain% number of expressions

        Be=ttm(S,G,2);
        Be=double(Be(:,e,:));% 100*10 double matrix
        [Qe,Re]=qr(Be,0);% Qe:100*10 || Re:10*10
        alpha=Re\((Qe'*Zhat);%求最小二乘解

    for p=1:ptrain%number of persons

        dif(e,p)=norm(alpha-H(p,:)',2);%求出误差的范数值

    if(dif(e,p)<mindif)% 若新误差小于最小误差 则更新

        mindif=dif(e,p);
        mine=e;
        minp=p;

    end

end

end

end

```

该算法完全对应前述推导过程。首先计算出 \hat{F}_T ，然后将矩阵 B_e 进行 QR 分解，最后求解最小二乘问题，得到被测图片在表情坐标系 e 下的坐标。

核心张量在各个模态下的奇异值具有顺序性，可以通过下面一段 Matlab 代码，显示核心张量在三个模态下的奇异值分布：

```
figure
imode=tenmat(A,1); %在图像模态下展开
idata=imode.data;
[Ui,Si,Vi]=svd(idata,0); %进行 SVD 分解
plot(diag(Si),'+b') %画出奇异值分布图

emode=tenmat(A,2); %在表情模态下展开
edata=emode.data;
[Ve,Se,Ue]=svd(edata',0); %进行 SVD 分解

pmode=tenmat(A,3); %在人物模态下展开
pdata=pmode.data;
[Vp,Sp,Up]=svd(pdata',0); %进行 SVD 分解

figure
t=[1:10]';
plot(t,diag(Se),'+r',t,diag(Sp),'ob') %画出奇异值分布图
```

分别将张量 A 在三个模态下展开，并对其进行 SVD 分解，求出其对应的因素矩阵，则该矩阵 SVD 分解的奇异值就代表了张量 A 在各个模态下的奇异值分布。其具体分布情况如下图所示：

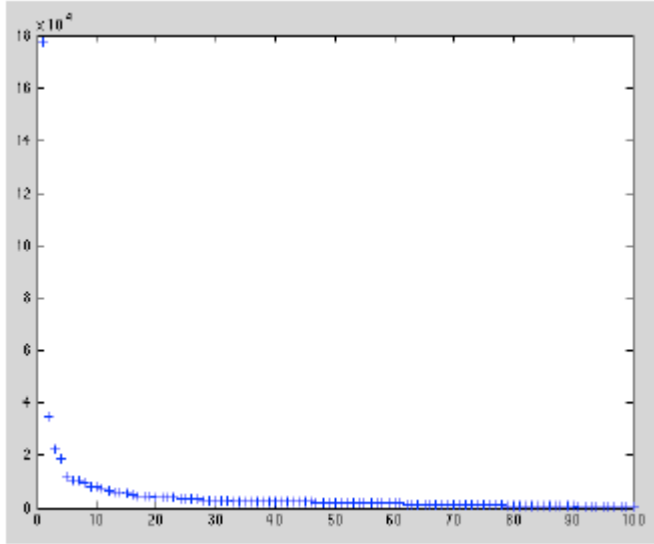


图 5- 5 图像模态下的奇异值分布

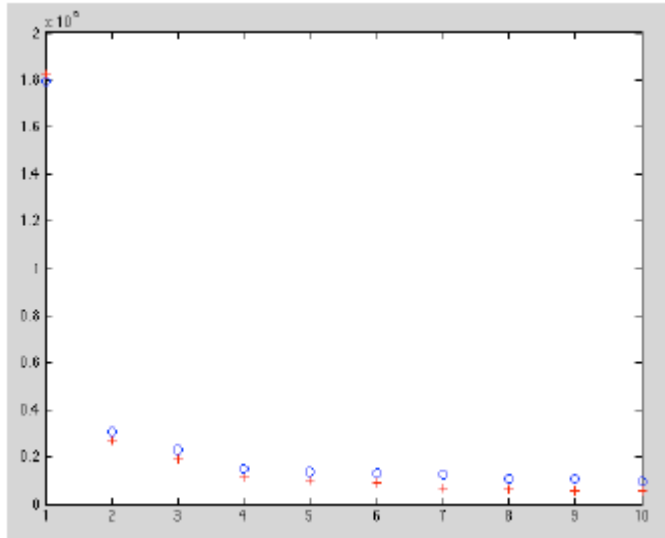


图 5- 6 表情模态 (+) 和人物模态 (o) 下的奇异值分布

从图中我们可以看出，张量 \mathbf{A} 在各个模态下的奇异值，都集中在前面少数几个值上。而这正提供了我们进行压缩的理论基础

前面章节已经对压缩算法进行了理论推导，基于 QR 分解加速的过程，将插入范数算式的因子 $\hat{\mathbf{z}} = \mathbf{F}^T \mathbf{z}$ 替换为 $\hat{\mathbf{z}}_k = \mathbf{F}_k^T \mathbf{z}$ ，得到最小二乘问题如下：

$$\left\| \hat{\mathbf{C}}_e \alpha_e - \mathbf{z} \right\|_2^2 = \left\| \hat{\mathbf{B}}_e \alpha_e - \mathbf{F}_k^T \mathbf{z} \right\|_2^2 + \left\| (\mathbf{F}^\sim)^T \mathbf{z} \right\|_2^2 \quad (5-1)$$

在这个算式中，各个矩阵都是经过压缩的，因此问题规模变小，计算量得到化简。
具体的实现代码如下：

```

k=10;%压缩值
Fk=F(:,1:k);%进行压缩，矩阵 F 共 100 列，取 10 列，压缩比为十分之一

figure
for i=1:ptrain% number of persons
    Z=imread(strcat('yaleface/s',pids(i,:),'.bmp'));
    Z=imresize(Z,[nrows,ncls]);% reshape the image to a vector
    z=double(reshape(Z,numel(Z),1));% 8736*1
    zhat=Fk'*z;
    mine=1;
    minp=1;
    mindif=100;% 不会超过 100 [观察结果一般都小于 2]
for e=1:etrain% number of expressions
    Be=ttm(S,G,2);
    Be=double(Be(1:k,e,:));%k=10，执行压缩
    [Qe,Re]=qr(Be,0);% Qe:k*10 || Re:10*10
    alpha=Re\((Qe'*zhat);% 求最小二乘解
for p=1:ptrain% number of persons
    dif(e,p)=norm(alpha-H(p,:)',2);% 求出误差的范数值
if(dif(e,p)<mindif)
        mindif=dif(e,p);
        mine=e;
        minp=p;
end
end
end
end

```


可以看到，压缩版算法的代码与前述基本相同，除了 F_k 与 B_e 两个矩阵都使用了原矩阵的压缩版本，压缩系数为 $k=10$ ，压缩比为十分之一。

通过调整 k 值，改变压缩比，进行多次实验并分析其结果如下：

当 k 大于等于 10 时，识别结果正确率都大于或等于 90%，可以认为 $k=10$ 是一个阈值，在此值之上，压缩并未带来算法正确率的影响。

当 k 小于 10 时，算法正确率开始受到一定影响，比如当 $k=5$ 时，十个人物中只有六个人物识别正确。

由此可知，只要我们保证 k 大于等于一定的值，则我们可以压缩 HOSVD，减小计算量，提高算法效率，同时并不会带来准确率上显著的损失。使用压缩后的具体运行时间见下表：

表 5.1 矩阵优化与 HOSVD 压缩加速效果

	未加速 (s)	使用因子矩阵加速 (s)	同时使用因子矩阵与 HOSVD 压缩 (s)	数据规模 (人数*图片数)
实验一	1.82	1.55	1.48	10*10
实验二	12.96	4.53	4.23	40*9
实验三	53.01	22.34	14.19	39*63

其中，HOSVD 压缩 k 的取值都保证压缩算法不损失原算法的准确率。可以看出，数据量越大的时候，压缩算法的优化效果越明显。

此外，这里需要注意的是，由于我们的实验中，出了图像模态维度较大之外，人物模态与表情模态都非常小，没有压缩的必要。然而，在实际的应用中，很可能人物模态与表情模态都相当的大。由于 HOSVD 中，核心张量在各个模态下都具有顺序性，因此在实际的应用中，我们可以将三个（甚至更多个）模态全部进行压缩，因此与未压缩的算法相比，可以极大的减小数据量，提高算法性能。

5.2.3 与小波结合的 HOSVD 与原始 HOSVD 算法对比

先用 db2 小波函数对人脸图片进行滤波, 舍弃高频分量保留低频分量, 处理后的图像只有原图像的四分之一大小, 所以需要处理的数据量大大减小了, 而且舍弃高频成分以后, 无关的信息被剔除, 主要特征得以彰显, 所以识别精度得以提高。

对于实验一原本未识别的那张图片由于无用信息被剔除, 所以能够全部识别, 识别精度为 100%, 比原来提高了 11%。

对于实验二的 orl 数据库, 数据规模有了明显的提高, 加速和精度提高的效果也凸显了出来, 可以看出, 识别精度在 40 张图片中, 对于不在数据库中新的表情, 准确的识别出了 37 张, 准确率为 92%, 所用的时间也有明显的提升。

对于更大规模的实验三, 这种算法的优点就更加明显了。具体实验数据如下表所示:

表 5.2 与小波结合后的算法与原始算法时间和精度对比

	数据规模	原识别精度	小波结合后识别精度	原 HOSVD 用时	小波结合后用时
实验一	10*10	90%	100%	1.82	1.30
实验二	40*9	90%	92%	12.96	4.56
实验三	39*63	100%	100%	53.01	12.29

6 算法在项目中的应用

6.1 鹰眼智能视频检索软件 V1.1

随着经济和科学技术的发展，监控摄像头几乎遍布中国大地的每个街头,在一些重点行业，摄像头的清晰度也越来越高。昼夜不停的监控录像，产生了海量的信息。然而，仅仅将画面记录下来只是第一步。当事件发生后，想在海量的视频中查找我们需要的信息，无疑是大海捞针。它往往需要一个庞大的团队来进行集中查看和分析捕获的视频录像，用于调查取证。这样不仅使监控成本提高，且高强度的查看分析极易使人疲劳，出现纰漏。如何快速的从监控视频中找到关键信息，成了目前视频监控应用的迫切需求。因此，鹰眼智能视频检索软件 V1.1 将实现图像的自动查找、自动识别，减少人的工作量，提高正确捕获分析我们需要信息的能力。

鹰眼智能视频检索软件 V1.1 包括案件管理、视频摘要压缩、关注目标、图像处理、地图管理、内容检索等模块。可广泛应用于公安、监狱、边防、银行、国家重点保护单位等行业的监控系统，比如公安行业的关键道路、卡口视频；交通行业的重点路段、收费闸口视频；监狱看守所的重点监舍、人员交接班视频等等；实现海量视频的浓缩管理，大大节省了人力成本，减少了由于种种原因造成的人为遗漏的风险，有效地解决海量视频快速查看比对的难题。

6.2 系统环境

系统运行需要使用 Windows7 以上版本，只支持 64 位系统，不支持 32 位。需要 OpenGL 驱动。其它硬件环境见下表：

表 6.1 系统硬件环境

	单机最底配置	单机推荐配置	服务器推荐配置
CPU	Intel i5 3.0G	Intel i7 3.3G	Intel Xeon E5 2.8G
内存	4G	8G	16G
显卡	无要求	AMD 系列显卡	无要求

硬盘	1T 7200 转/分	2T 7200 转/分	500G 10000 转/分
固态硬盘	无要求	120G	120G*3
显示分辨率	1366*768	1600*900	无要求

6.3 表情无关人脸识别算法在鹰眼智能视频检索软件中的应用

在一些重点行业，尤其是公安领域，对在逃犯罪分子的识别对于警察进行有效的抓捕活动有着重要的意义。如果能在重点道路、卡口、收费闸口、监舍等地点通过较清晰的监控视频将犯罪分子识别出来，无疑能大大提高公安干警的工作效率。然而，由于室外拍摄条件多变，受光照、表情、人脸遮盖物（帽子、眼镜等）的影响，对其进行人脸识别有一定的困难。所以，研究一种不受拍摄条件影响的人脸识别算法显得尤其重要。

在这样的需求下，本文提出了一种基于高阶张量奇异值分解的人脸识别算法，该算法在人脸领域用高阶奇异值分解来进行人脸特征的表示和提取,这种方法克服了其它特征提取方法过于依赖拍摄条件的缺点。HOSVD 将数据转换成三维或更高维线性模型,能避免二维线性方法中条件改变则精确度下降的问题，使得识别算法在变化的条件下获得了相对稳定的结果。

系统初始界面如图所示，



图 6- 1 系统界面图

首先在系统中右侧选择人脸识别模块，然后按下图的方式添加案件：

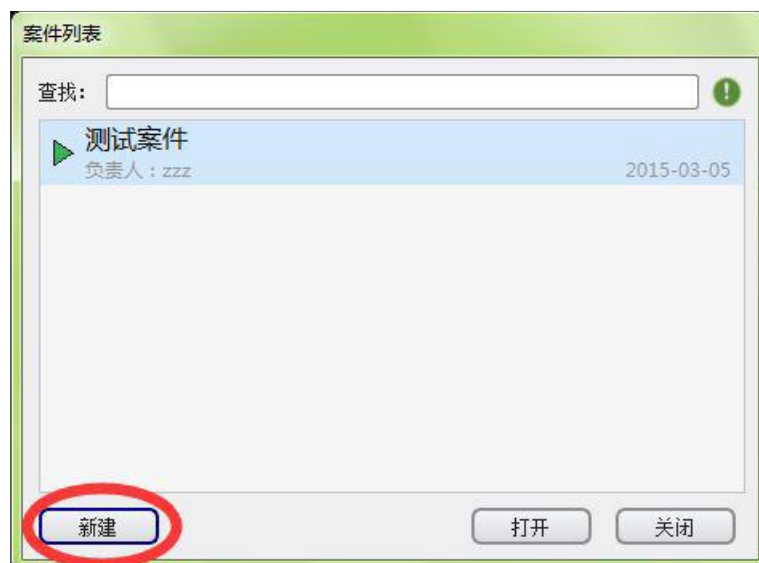


图 6-2 案件添加方式

在进行人脸检测和识别之前，先运行视频压缩功能。

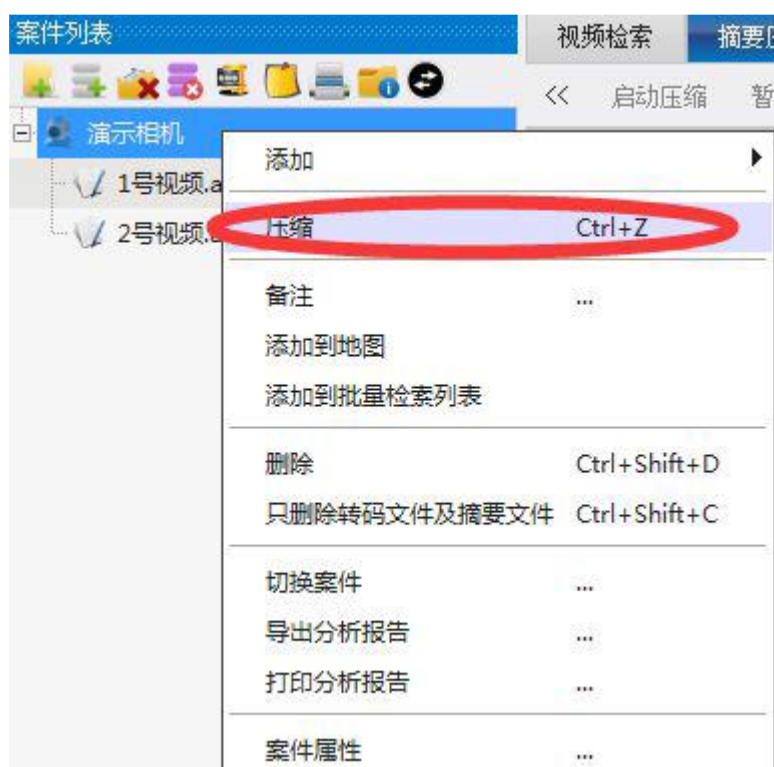


图 6-3 视频压缩方法

因为要想快速的对视频的内容进行检索查看，那首先要对视频文件进行摘要压缩，对视频内的运动目标进行分割和识别，然后对这些目标的外形特征跟踪，运动方

向跟踪等。压缩完成后会生成一个以 `casit` 为扩展名的摘要文件。这个摘要文件存储了从原始视频中分析出来的所有有效目标信息。后期进行视频内容检索的过程实质上就是以不同的搜索条件对摘要文件进行搜索。

然后运行人脸识别功能,由于本系统中可以导入公安机关的在逃犯罪人员的人脸图像数据库,所以可以在视频中将检测到的某帧人脸图像抽取出来,利用上述高阶张量奇异值分解算法进行识别,可以克服人脸表情变化的影响。另外,可以设置更高的维度,使其不受光照和遮盖物等条件的影响,从而有效的识别在逃犯罪人员。如果该人物确实是罪犯数据库中的人物或者是十分相似的人脸,在右侧的检索结果里就会出现检索结果人物列表,供公安人员进行判断,减少国家和人民的损失,减轻公安干警的工作负担,实现科技治安。

7 总结与展望

7.1 总结

在人脸识别算法中，人脸特征的表示与提取是最重要的一环。一旦找到了合适的形式进行人脸特征的表示与提取，则识别过程将水到渠成。传统的特征提取方法与主成分分析方法等，都具有过于依赖人脸图片拍摄条件（光照、表情、角度等）的问题，一旦条件发生改变，则算法稳定性大受影响。如何能够获得一种人脸特征的表示方式，使得该特征能够不受拍摄条件影响，在条件改变时获得稳定的结果，是人脸识别领域的一个挑战。

本文具体贡献如下：

将高阶张量 **SVD** 分解应用到人脸识别领域，推导了基于高阶张量 **SVD** 分解的人脸识别算法。利用该方法进行人脸特征的表示与提取，可以获得一种至于人物身份有关，而不依赖于具体拍摄条件的人脸特征信息。具体说来，本算法不受人脸表情的影响，只要数据库中的图片选取得当，则即使当被测图片拍摄条件发生改变时，识别算法依然可以获得较高的准确率，从而解决了传统的识别算法过于依赖被测图片拍摄条件的问题。

提出优化的 **HOSVD**：矩阵 **QR** 分解，以及张量 **HOSVD** 压缩等优化加速环节，在保证算法准确率不变的情况下，有效降低了算法的计算量，提高了工作效率。其中，张量 **HOSVD** 压缩是一个非常有效的优化方法，尤其当问题规模变大时，可以在出了图像模态之外的人物模态、表情模态等进行压缩，进一步简化算法工作量。

提出小波分解和张量 **HOSVD** 相结合的算法，先对人脸图像进行小波分解，可以去除噪声和无用的信息，使得识别结果更加准确，同时分解后图片变小，减小了识别过程中的计算量。由此，算法的效率和准确率都得到了显著的提高。

实现了一个 **C#** 的图形界面，便于数据库的编辑，人脸识别程序的演示等工作，此外解决了实现过程中 **C#** 与 **Matlab** 混合编程的问题。

7.2 展望

1.在进行人脸识别的过程中,由于多层循环分别针对不同的表情坐标系以及不同的被测图片进行计算,相互之间并无影响,因此可以通过将人物表情等进行一定划分,通过多线程并行进行加速,进一步提高程序效率。

2.由于本次实验主要针对与人脸识别过程中人脸特征提取与识别等步骤,之前的人脸检测步骤,所以准确率收到一定的影响。下一步准备和较为成熟的人脸图片检测技术结合起来,可以得到更好的效果。

参考文献

- [1] Eldén L. Matrix methods in data mining and pattern recognition[M]. SIAM, 2007
- [2] R. Baeza-Yates and B. Ribeiro-Neto. Modern Information Retrieval[M]. ACM Press, Addison-Wesley, New York, 1999.
- [3] R.Chellappa, C.L.Wilson, and Saad Sirohey. Human and Machine Recognition of faces: A Survey[J]. Proceedings of IEEE, 1995.5,vol.83,no.5
- [4] Gerrald J. Kaufman, and Kenneth J.Breeding. The automatic recognition of human faces from profile silhouettes[J]. IEEE Transaction on Systems, Man, and Cybernetics, 1976,6(2):113-121.
- [5] 山世光,人脸识别中若干关键问题研究[D].中国科学院博士学位论文, 2004.7
- [6] Kasamanis A, Pandreou G, Maragos P ,et al .Face active appearance modeling and speech acoustic information to recover articulation[J]. Audio Speech and Language Pro, IEEE Transaction on, 2009,17(3):411-422.
- [7] Hong Z. Algebraic feature extraction of Image for Pattern Recognition[J].1991 24 (3) :211-219.
- [8] Cheng Y , Liu K , Yang J ,et al. Human face recognition method based on the statistical model of small sample size[C]. Intelligent Robots and Computer Vision X: Algorithms and Techniques. Boston, 1991.1,1608:85-95.
- [9] M. W. Berry, S. T. Dumais, and G. W. O'Brien. Using linear algebra for intelligent information retrieval[J]. SIAM Rev., 1995, 37:573–95.
- [10] 周杰, 卢春雨, 张长水等, 人脸自动识别方法综述[J], 电子学报, 2002,vol.28,no.4:102-106.
- [11] N. Christianini and J. Shawe-Taylor. An Introduction to Support Vector Machines[M].Cambridge University Press, London, 2000.
- [12] Qiang Zhang, Baoxin Li, Discriminative K-SVD for dictionary learning in face recognition[J]. Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference, June 2010:2691-2698

- [13]周激流, 张晔, 人脸识别理论研究进展[J], 计算机辅助设计与图形学学报,1999,vol.11,no.2, 180-184.
- [14]祁泽宇, 人脸部特征的提取[D], 硕士学位论文, 上海交通大学, 1995:15-19
- [15] J. M. Conroy, J. D. Schlesinger, D. P. O’Leary, and J. Goldstein. Back to basics: CLASSY 2006[J]. DUC 02 Conference Proceedings, 2006.
- [16] C. Boutsidis and E. Gallopoulos. On SVD-Based Initialization for Nonnegative Matrix Factorization[R]. Technical Rep
- [17] L. De Lathauwer, B. De Moor, and J. Vandewalle. A multilinear singular value decomposition[J]. University of Patras, Greece, 2005.
- [18] M. A. Turk and A. P. Pentland. Eigenfaces for recognition[J]. Journal of Cognitive Neuroscience, 1991 23(3):71–86.
- [19] Manjunath B S , Shekhar C ,Chellappa R. A new approach to image feature detection with application[J]. Pattern Recognition April 1996, Volume 29, Issue 4 , 627–640 .
- SIAM. Matrix Anal, 2000,21:1253–1278.
- [20]边肇祺.《模式识别》.清华大学出版社, 2000 年.
- [21] WANG Hui-yuan, Leng Yan, Wang , Zeng-feng, et al. Application of image correction and bit-plane fusion in generalized PCA based face recognition[J]. Pattern Recognition Letters, 2007, 28(16):2352-2358.
- [22] P. Berkin. A survey on PageRank computing[J]. Internet Math., 2005, 2:73–120.
- [23] K. J. Cios, W. Pedrycz, and R. W. Swiniarski. Data Mining Methods for Knowledge Discovery[M]. Kluwer, Boston, 1998.
- [24] Mohammed Alwakeel, Zyad Shaaban, Face Recognition Based on Haar Wavelet Transform and Principal Component Analysis via Levenberg-Marquardt Backpropagation Neural Network[J]. European Journal of Scientific Research. ISSN ,2010:Vol.42 No.1, 25-31
- [25] M. W. Berry, M. Browne, A. Langville, V. P. Pauca, and R. J. Plemmons. Algorithms and Applications for Approximate Nonnegative Matrix Factorization[R]. Department of Computer Science, University of Tennessee, 2006.

- [26] 沈永增,陈勇.基于小波子图和决策融合的人脸识别算法[J], 浙江工业大学学报,2003,6,31(3),440-449.
- [27] K. Ramesha, K. B. Raja. Face Recognition System Using Discrete Wavelet Transform and Fast PCA[J]. Communications in Computer and Information Science, 2011:Volume 147, 13-18.
- [28] M. C. Burl, L. Asker, P. Smyth, U. Fayyad, P. Perona, L. Crumpler, and J. Aubele. Learning to recognize volcanoes on Venus[J]. Machine Learning, 1998,30:165–95
- [29] 王雅峰, 丁彦蕊。一种快速张量分解算法及其在人脸识别中的应用[J].电子设计工程.2013-10:Vol.21,No.19,30-35
- [30] 刘超远, 人脸识别技术与计算机视觉场模型[D], 博士学位论文, 上海交通大学, 1996:33-40
- [31] Kolda T G, Bader B W. Tensor decompositions and applications[J]. SIAM review, 2009, 51(3):455-500.
- [32] 许亦男,王士同.基于 SLPP 和张量分解相结合的人脸识别[J]. 计算机应用研究, 2011, 8
- [33] V. D. Blondel, A. Gajardo, M. Heymans, P. Senellart, and P. Van Dooren. A measure of similarity between graph vertices: Applications to synonym extraction and web searching[J].SIAM Rev., 2004,46:647–66.
- [34] Cox I J, Ghosn J, Yianilos P N. Feature-based face recognition using mixture-distance[C]. IEEE Computer Society Conf. on Computer Vision and patt.Recog, IEEE,1996:209-215
- [35] J.-P. Brunet, P. Tamayo, T. R. Golub, and J. P. Mesirov. Meta genes and molecular pattern discovery using matrix factorization[J]. PNAS, 2004, 101:4164–4169
- [36] LEE S, CHOI S. Two-dimensional canonical correlation analysis[J], IEEE, Signal Processing , 2007,14(10):735-738.
- [37] K. Blom and A. Ruhe. A Krylov subspace method for information retrieval[J].SIAM J. Matrix Anal. Appl. 2005, 26:566–82,.
- [38] 雷刚, 蒲亦菲, 张卫华, 周激流.张量典型相关分析及其在人脸识别中的应用[J]. 电子科技大学学报,2012-5,Vol.41,No. 3,109-115.

- [39] Daoqiang Zhang, Songcan Chen , Zhi-Hua Zhou. A new face recognition method based on SVD perturbation for single example image per person[J]. Applied Mathematics and Computation, Volume 163, Issue 2, 15 April 2005:895–907
- [40] 梅蓉蓉,吴小俊,冯振华.基于状态估计的张量分解人脸识别方法[J] 计算机工程与应用 2011:47(24),143-145
- [41] Qiang Zhang, Baoxin Li, Discriminative K-SVD for dictionary learning in face recognition[J]. Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference, June 2010:2691-2698
- [42] S. Brin and L. Page. The anatomy of a large-scale hyper textual web search engine[J]. Comput. Networks ISDN Syst. 1998,30:107–17.
- [43] T. A. Davis. Direct Methods for Sparse Linear Systems[J]. Fundamentals of Algorithms. SIAM,, 2006.
- [44] E. Anderson, Z. Bai, C. Bischof, et al. LAPACK Users' Guide, 3rd ed[M]. SIAM, Philadelphia, 1999.
- [45] Bader B W, Kolda T G. Efficient MATLAB computations with sparse and factored tensors[J]. SIAM Journal on Scientific Computing, 2007, 30(1): 205-231.

硕士研究生学习期间科研成果

1、论文情况

[1]计雨含，王晓东，姚宇，基于张量 SVD 分解的人脸识别方法研究[J]，《计算机应用》，2015 年（增刊）。（国内核心刊物）

2、科研项目情况。

2014 年 4 月-2015 年 4 月 鹰眼智能视频检索系统项目

致谢

时光飞逝，三年的研究生生活马上就要随着毕业论文的完成而结束了。这是一段路程的终点，也是另一段路程的开始。硕士论文的撰写和研究生的科研生活都是一个既艰辛又充满成就感的过程，最喜欢不断钻研，攻克难题的欢喜，而这样的快乐都是出现在不断的钻研努力之后。在这个过程中，我知道了科研工作的艰苦和伟大，正如爱迪生所说“在科学的道路上，没有平坦的大道，只有不畏劳苦，沿着陡峭山峰攀登的人，才有希望达到光辉的顶点”，在以后的学习和工作用，希望自己以在此论文撰写过程中体会到的科学精神勉励自己，不畏劳苦，实事求是，一丝不苟，专心致志。在论文的即将撰写完毕的时候，我由衷的感谢为这篇论文提供帮助过的人，感谢在实验室为我提供宝贵意见的兄弟姐妹们，同时也很感谢那些没有见过面的，在我的论文中引用的文章的作者们，没有借助你们的成果，这篇文章不可能这么顺利的完成。

借此论文，我要特别感谢我的导师王晓东老师，王老师平日里工作繁多，但在我做毕业设计的每个阶段都给予了我悉心的指导。尤其是王老师对待学生的包容和理解，使我深深为之感动，如果没有王老师帮助和指导，这篇论文不能如此顺利的完成。在此真诚的对您说声感谢。除了敬佩于老师的专业水平外，他的宽以待人的品格和严谨的治学态度是我永远学习的榜样，并将积极影响我今后的学习和工作。

感谢实验室的各位同学，和你们在一起学习和生活非常愉快，谢谢你们对我学习上和各个方面对我的帮助。正是因为有了你们的帮助和支持，我不仅学到了本次课题所涉及的知识，更感觉到了知识以外的东西，那就是团结的力量。还有感谢姚宇老师在论文撰写的各个阶段的指导和提出的宝贵建议。感谢汤琳老师一直以来对我学习和生活上的帮助。

最后我要感谢我的父母，你们给予我生命的权利，给予我一个温馨的家，近几年来我一直在外求学，没能陪在你们身边，但是我一定会努力工作生活，成为你们的骄傲。

谢谢所有关心，帮助，支持我的人！

计雨含

2015年3月31日