

北京工业大学

硕士学位论文

图像拼接技术研究

姓名：高新杰

申请学位级别：硕士

专业：测试计量技术及仪器

指导教师：李德胜

20070601

摘 要

图像拼接技术是由于摄像器材的视角限制，无法一次拍出很大图片而产生的，它是将一系列具有部分重叠的图像进行无缝拼接而得到超宽视角图像的技术。

本文总结了国内外图像拼接技术的研究现状，详细叙述了数字图像拼接的全部过程，包括图像拼接的理论基础、图像拼接的获取和几何校正、图像拼接的预处理、图像配准和图像拼接等步骤，重点研究了图像配准的两大经典算法：基于像素灰度级的方法和基于频域的方法。在此基础上，论文针对现有拼接算法的不足及图片的实际拍摄情况，提出了一种改进的基于比值法和模板匹配法的图像拼接算法，该算法综合考虑了图像拼接的精度和速度，对两幅具有重叠区域的灰度或彩色图像，先通过设置拼接参数对话框，在另一幅图像中找到最佳匹配点，最后利用平滑因子对两幅图像的重叠区域进行数据融合操作，实现了图像的快速自动拼接。实验表明，算法原理简单、拼接速度快。最后指出了算法的使用范围，比较了算法的优缺点，并对图像拼接算法的发展进行了展望。

关键词 图像拼接；图像配准；图像融合；比值法；模板匹配

Abstract

As we know, the angular field of view in camera is in limit, we could not shoot a image which has a very large coverage , thus the image mosaics is generated, and used widely.The image mosaics is an active method to be used to generate a new image which has a large angular field of view.

The dissertation surveys image mosaics technology, and the whole process of digital image mosaid is introduced. The principle of the image mosaic, the acquisition of the image and the revising of the digital image, the pre-processing of the image mosaic, the image registration and the image mosaic are the main parts of the dissertation. The pixe gray level-based stitching algorithm and the fft-based stitching algorithm for image registration are emphatically discussed.. On the base of it, by analyzing the available disadvantages of image stitching algorithms and image acquisition, an improved algorithm used in gray or color image mosaic based on ratio method and template matching is presented in this paper. This algorithm took into account the precision of image stitching as well as the speed of it .By setting dialog box for mosaing two images which have overlapped area,then search a best matching point in another image,finally merge overlapped image area data by using smooth factors.A lot of experiments show that this algorithm is effective for most overlapped images. Extensive experimental results demonstrate the simplicity ,speediness, and efficiency of the algorithm. According to this ,we expound the limitations and suitable areas of these algorithms.Finally, we summarize the existing problems and point out some promising directions for the future research.

Keywords image mosaic; image registration; image composition; ratio method; template matching

独 创 性 声 明

本人声明所呈交的论文是我个人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得北京工业大学或其它教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

签名： 高新杰 日期： 2007.6.11

关于论文使用授权的说明

本人完全了解北京工业大学有关保留、使用学位论文的规定，即：学校有权保留送交论文的复印件，允许论文被查阅和借阅；学校可以公布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论文。

（保密的论文在解密后应遵守此规定）

签名： 高新杰 导师签名： 李法胜 日期： 07.6.11

第1章 绪论

1.1 引言

一台相机的视野范围是有限的,虽然利用扫帚式相机和广角镜头可部分解决相机视角不足这一问题,但广角镜头的边缘会产生难以避免的扭曲变形,为了获得范围更广的视野,人们自然考虑将多个照片拼接成一幅大的照片以增强相机的视野。图像拼接技术就是把针对同一场景的相互有部分重叠的一系列图片合成一张大的宽视角的图像,并且要求拼接后的图像最大程度地与原始图像接近,失真尽可能小,没有明显的缝合线^[1]。随着数字图像处理理论的丰富,近年来的发展趋势是利用PC机通过一定的算法来完成多幅图像的拼接,从而生成一幅完整的大图像。拼接示意图可参考图1-1。2003年,美国“勇气号”和“机遇号”火星探测器发回了大量的火星地面照片,科学家们就是运用图像拼接技术合成了火星表面的宽视角全景图像。因此,研究并提出一种精确而高速的图像拼接算法具有十分重要的现实意义。

图像拼接技术已广泛应用在宇宙空间探测、医学、气象、军事、视频压缩和传输,档案的数字化保存,视频的索引和检索,物体的3D重建,数码相机的超分辨率处理等领域^[2]。图像拼接的广泛应用,图像拼接理论不断得到丰富。本课题分析了国内外常用图像拼接理论,根据课题背景,有目的地选取基于像素灰度级的拼接方法,并在该算法基础上加以改进,提出了一种基于比值法和模板匹配法相结合的拼接算法,可适用于灰度图像和彩色图像拼接。

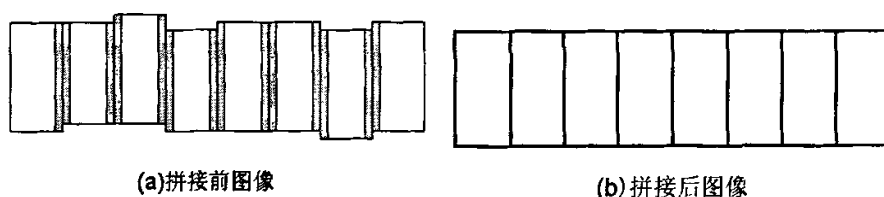


图 1-1 图像拼接示意图

Fig.1-1 Image mosaic experiment

1.2 国内外研究现状

图像拼接的核心技术就是图像配准技术,早在 1992 年,剑桥大学的 LisaGottesfeld Brown 就已经总结了各领域图像配准技术的基本理论以及它们的主要方法。1996 年, Richard Szeliski (Microsoft Corporation), 提出了

基于运动的全景图像拼接模型, 该模型采用迭代非线性最小化方法 (简称 L-M 算法), 主要利用得到的图像之间的几何关系来进行图像配准, 该算法突出优点是收敛速度快, 且可处理具有平移量、旋转量的源图像, 因此成为图像拼接领域的经典算法^[3,4]。2000 年, Shmuel Peleg(member IEEE), Benny Rousso, Alex Rav-Acha, 和 Assaf Zomet 在 Richard Szeliski 提出的算法的基础上加以改进, 提出了自适应的图像拼接模型, 该模型根据相机的不同运动, 自适应选择拼接模型, 通过把图像分成狭条进行多重投影来完成图像的拼接。这一研究成果无疑推动了图像拼接技术的进一步发展, 自适应问题也从此成为图像拼接领域研究的新热点。

同时, 基于相位相关度法和基于像素灰度级的图像配准算法也得到了足够的发展。早在 1975 年, Kuglin 和 Hines 就提出了具有场景无关性的相位相关度法。后来, 随着傅立叶变换的广泛应用和不断发展, De Castro 和 Morandi 将傅立叶变换应用到拼接中。对于图像的平移, 计算两图像功率谱的逆傅立叶变换可得一个脉冲函数, 该函数仅在平移量处不为零。对于旋转, 可使用极坐标方式表示, 用相同方法计算图像间的旋转角度。相位相关度法充分利用了傅立叶变换的良好性质, 即函数平移, 旋转和变尺度在频率域都有其对称性。基于像素灰度级的方法, 是以两幅图像重叠部分所对应像素在 RGB 系统中灰度级的相似性为配准准则, 自动寻找图像的配准位置。该算法不需对图像进行变换处理, 充分利用了图像自带信息, 常用的方法有比值法、模板匹配法, 采用不同的评价函数, 模板匹配法又可分为最小差值法和极大相关系数法。基于像素灰度级的方法拼接精度和速度基本能满足实际要求, 是目前国内外常用的方法。

基于几何特征的图像配准方法是图像拼接技术的另一研究热点。1994 年, Blaszk T. Tachid Deriche 通过二维高斯模糊过滤可以得到一些低级特征模型, 如边模型、角模型和顶点模型^[5~7]。

以上讨论的拼接方法多用于静止拍摄的图像拼接, 为了使场景具有动态的特征并极大地增强漫游的真实感, 近年来已经出现了对动态全景图的拼接研究, 另外随着视频编码技术的迅速发展和极大应用, 图像拼接技术也和其结合起来, 这主要应用于 MPEG-4 编码标准中的 Sprite 技术, 它主要对某段视频编码序列中的背景图像进行全景图的拼接, 生成的完整背景图也叫做 Sprite 图像。

1.3 图像拼接技术研究的核心问题

图像拼接流程主要包括两个关键环节, 即图像配准和图像融合。对于图像融合部分, 由于其耗时不大, 且现有的几种常用方法效果差别也不大, 总的来说算法比较成熟。图像配准则是整个图像拼接流程的核心, 它直接关系到图像

拼接算法的成功率和运行速度，因此图像配准算法的研究一直是图像拼接研究的重点。

在机器识别过程中，常需把不同传感器或同一传感器在不同时间、不同成像条件下对同一景物获取的两幅或多幅图像在空间上对准，或根据已知模式到另一幅图中寻找相应的模式，这就叫做图像配准（图像匹配）。图像配准，即是要准确的找到相邻图像间重叠区域的位置以及范围，以便将相邻的具有重叠区域的图像“缝合”成一个新的更大画面的视图。传统的图像拼接思想是通过图像匹配，准确找出两幅图像的重叠部位。图 1-2 中给出待拼接的两幅图像 L 和 R，它们之间存在 30%—50% 的重叠度。拼接时，在左边图像 L 上选取一个匹配模板 T，利用模板 T 在右边的图像 R 中搜索与之匹配的位置，最后根据匹配位置完成图像的拼接。所以，在图像的拼接过程中图像的匹配是关键，它既决定拼接的效率，又决定拼接的准确度。

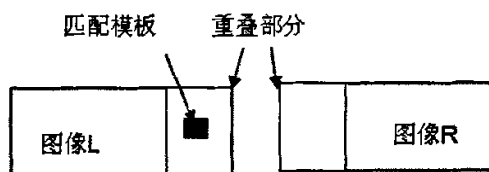


图 1-2 图像匹配示意图

Fig.1-2 Image matching experiment

近年来，研究人员在图像匹配技术方面做了大量的研究工作，提出了一系列行之有效的算法。基于像素灰度级的算法可分为两大类：

1) 灰度相关匹配。在第一幅图像中选取一块子图像作为匹配模板，根据某个相似度评价函数对第二幅图像中的像素灰度值进行比较，找出两幅图像的对准点。

2) 特征相关匹配。在第一幅图像中选取包含某个特征的子图像作为匹配模板，在第二幅图像中进行这一特征的匹配。

1.4 本课题的主要研究内容

1.4.1 主要工作

本课题旨在对静止拍摄图像序列进行全景图拼接的研究。研究的重点是基于像素灰度级相关的图像拼接算法，分析了该方法的不足，并提出了改进措施，提出了一种自动的快速图像拼接算法。论文后期对算法进行实验，分析算法的合理性和有效性。

研究工作分为以下个阶段进行：

(1) 前期工作中，阅读了国内外大量的关于图像拼接的参考文献，对当前图像拼接算法发展应用有了足够的认识，并掌握了常用图像配准技术的基本理论。

(2) 重点研究静止拍摄图像序列的拼接技术，分别讨论了频域法和基于像素灰度级相关的算法，并加以对比，指出后者的可行性。

(3) 本设计重点研究比值法和网格匹配法，并以此理论为基础，利用 VisualC++6.0 软件设计了一套可行的算法，从而实现灰度图像和彩色图像的准确拼接。

(4) 对该算法进行大量图像拼接实验，分析所得数据，对算法的精度和速度加以评价，并进一步指出算法的不足和改进措施。

1.4.2 论文章节安排

全文共分为六章。各章节内容安排如下：

第一章：绪论。阐述了图像拼接技术的发展概况，指出了图像拼接技术研究的核心问题以及本课题的主要研究工作。

第二章：图像拼接及其关键技术。按照图像拼接流程，分别介绍了图像获取、预处理、图像匹配算法和图像融合算法。重点介绍了典型的图像配准算法，并总结了常用的图像融合算法。

第三章：基于频域的图像拼接法。详细介绍了相位相关度法原理和具体实现，并指出了该方法的不足。

第四章：基于像素灰度级的图像拼接法。本设计以基于像素灰度级相关算法为理论基础，所以这是本论文的重要部分。本章详细介绍了比值法和模板匹配法的原理，并指出了两种方法的缺陷，对于两种方法的改进以及具体实现会在第五章里提到。

第五章：图像拼接具体实现。简要介绍了数字图像处理基础和 VisualC++ 在图像拼接中的应用，提出了一种基于比值法和模板匹配相结合的方法，重点介绍了灰度图像和彩色图像拼接的具体实现。

第六章：实验与数据分析。对不同图像进行拼接实验，分析所得实验数据，讨论了算法的精度和速度问题。指出算法合理性的同时进一步探讨算法存在的不足，最后是对本文工作的总结和图像拼接算法的研究展望。

第2章 图像拼接及其关键技术

在利用电荷耦合器件 (CCD) 摄像机等图像采集装置对图像进行采集和测量时, 常常遇到被测目标面积比 CCD 摄像机视场要大的情况, CCD 所采集的每一幅图像不能完全反映目标的全貌。实际中可采集多幅图像, 利用图像拼接技术将目标的不同部分拼接成一幅完整图像来反映目标的整体特征。图像拼接是利用边界部分重叠的图像共同特征匹配实现对接^[6]。图像拼接技术典型地应用于医学图像分析, 遥感数据处理, 航拍图处理、模式识别等领域。全景拼图技术是指利用相机平移或旋转, 拍摄得到一个相邻照片之间有部分重叠的图像序列, 通过适当的算法组合这些序列图像, 从而生成一幅完整图像的场景绘制方法。图像拼接技术是一个崭新的研究领域, 具有非常广阔的发展前景。

图像拼接流程大致可分为: 图像获取、图像预处理、坐标变换、图像配准 (图像匹配) 和图像融合等基本步骤。图像拼接基本流程如图 2-1 所示。首先获取待拼接的图像, 然后经过预处理 (滤波等), 再进行统一坐标变换, 即将所有图像序列变换到统一的坐标系中, 不同的变换方式对应不同的拼接流程, 接着进行图像匹配和图像融合, 最终得到全景拼接图。其中, 预处理和统一坐标变换两步不是必须的, 可视具体情况进行选择。

本章首先介绍图像获取的主要方法, 然后介绍统一坐标变换, 最后重点介绍图像匹配和图像融合这两个关键环节。

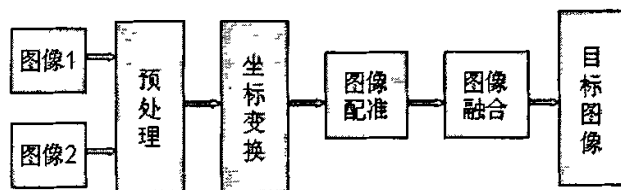


图 2-1 图像拼接流程

Fig.2-1 Flow process chart of image mosaic

2.1 图像获取

原始图像质量的好坏是影响图像拼接算法的最重要因素。原始图像获取的方式不同, 得到的图像整体性质也有很大差异。

图像输入获取装置主要有摄像机、扫描仪、数码照相机和遥感图像获取设备等。现在一般采用数码相机作为图像获取工具。数码相机是一种能够进行景物拍摄, 并以数字格式存放拍摄图像的特殊照相机。它的核心部件是 CCD 图像

传感器, CCD 可以对亮度进行分级, 但并不能识别颜色, 为此, 数码照相机用红、绿和蓝三个彩色滤镜, 当光线从红、绿、蓝滤镜中穿过时, 就可以得到各种色光的反应值, 再通过软件对得到的数据进行处理, 从而确定每一个像素点的颜色。图 2-2 从左到右分别为数码照相机、专业图像采集摄像机和 CCD 三彩色滤镜示意结构图。



图 2-2 图像输入设备

Fig.2-2 Equipment of image inputting

以数码照相机为例, 分析图像获取的三种情况: (1) 照相机固定在三角架上, 旋转照相机拍摄; (2) 照相机放置于一个滑轨上, 平行移动照相机进行拍摄; (3) 是一种普通的情况, 人手持照相机, 站在原地拍摄四周, 或者沿着照相机的光轴垂直方向走动拍摄^[9]。

1、旋转照相机拍摄

在这种情况下, 放置照相机的三脚架在拍摄过程中一直在同一位置。拍摄时, 照相机绕垂直轴旋转, 每旋转一定的角度, 拍摄一张照片, 理想的情况下, 照相机不绕照相机的光轴旋转。

拍摄得到一系列照片中相邻两张必须有部分重叠, 重叠区域的大小是影响图像能否成功拼接的因素。建议相邻图像之间重叠比例达到 20% 以上^[2]。重叠比例越大, 拼接相对容易, 但是需要的照片越多。旋转照相机拍摄由于照相机固定, 不需要恢复过多参数, 较容易实现。但是, 拍摄图像不在一个平面上, 需要投影到同一个平面上, 投影过程将会导致图像质量下降。

2、平移照相机拍摄

平移照相机指的是照相机在一个平行于成像平面的方向上平移。在固定焦距的情况下, 照相机放置在一个滑块上移动拍摄。物体和照相机的距离远近, 或者拍摄物体的大小变化都会影响到最后的拼接结果。这种情况的特点是: 拍摄的相片在一个平面上, 但全景图的三维感觉不如旋转拍摄的好。

3、手持照相机拍摄

这种方法比较容易做到, 手持照相机原地旋转拍摄, 或按一定的路线平行于对象拍摄。但是, 拼接手持照相机拍摄的照片是很困难的, 因为在拍摄过程中, 照相机的运动非常复杂。原地旋转拍摄类似于固定照相机旋转拍摄, 但是角度控制、旋转控制都很差。沿一定路线移动时, 类似于平移拍摄, 控制距离和保持相同的成像平面很困难。为了减少这些影响, 可以增加重叠比例, 使照

相机旋转角度、平移减小,因而减小相邻图像之间的不连续程度。

光强差异是影响图像处理的一个重要因素。图像拼接要求待拼接图像重叠区域光强差异不大,否则会增加图像拼接的难度,差异太大会导致拼接失败。图像获取最常见的问题就是相邻图像之间光强有较大的变化。理想情况下,相同的区域应该有相同的光强,但实际中光源变化、相机移动以及反光区域带来的影响都会导致光强的差异。所以在拍摄照片时,尽量保证每张照片亮度、色度以及对比度差异不大,并且保证相邻照片有足够的重合部分,保证获取的图像质量较高。

2.2 统一坐标变换

本节简要的讨论平面对应(planar homography)的几何学问题。平面对应是由平面透视成像引起的映射问题。图像获取方法不同得到的图像就不同。旋转拍摄得到的图像不在同一个平面里,而平移照相机拍摄,可保证获得的图像在同一平面里。如果采用旋转照相机拍摄方式,对所得图像进行拼接前就需要统一坐标变换。旋转图像拼接按照拼接基准表面的映射模型可分为平面、球表面和柱面等各种类型,它是通过将所有图像投影到这些简单表面上完成的^[10]。

1、平面拼接(Planar Mosaic)

平面拼接就是以序列图像中一幅图像的坐标系为基准,将其它图像都投影变换到这个基准坐标系中,使相邻图像的重叠区域对齐,称由此形成的拼接为平面拼接^[11]。

2、球面拼接(Spherical Mosaic)

人们在观察周围世界时,首先通过视网膜获取真实世界的图像信息,然后将图像信息通过透视变换投影到眼球部分。因此,最自然的全景图模型就是以视点为中心的单位球。球面拼接与眼模型最接近的一种全景描述,但它有以下缺点:球面不能展开在一个平面上,球面上数据不宜存储,图像处理难度大^[12]。

3、柱面拼接(Cylindrical Mosaic)

柱面拼接是指将采集到的图像数据重投影到一个以相机焦距为半径的柱面,在柱面上进行全景图的拼接。对于旋转相机拍摄的图片,采用柱面投影可以保证拼接完成的图像和肉眼观察到的图像在角度上的一致性。在绝大多数的应用中,横向 360 度的环视环境即可较好地表达出空间信息,所以圆柱体全景图模型是较为理想的一种选择。这种映射方式是目前应用最为广泛的一种映射方式。

在生成柱面全景图的过程中,如果对拍摄的原始图像序列直接进行拼合,将会破坏实际场景的视觉一致性,无法满足实际场景中各对象的几何关系。为

了保持实际场景中的空间约束关系，将拍摄得到的原始图像序列映射到一个标准投影即柱面投影站，得到柱面图像序列，再进行拼接得到柱面全景图。

本设计可采用两个（或多个）CCD 摄像机，分别获取被测物不同局部图像，保证被测物不动，CCD 摄像机沿同一方向移动，控制 CCD 位置即可控制重叠区域范围，这样保证扫描后的图像在同一水平面内，本设计中不需要坐标变换这一步骤。

2.3 典型图像配准算法研究

图像拼接的质量主要依赖图像的配准精度。配准算法既要保证配准的精度，又要保证一定的速度，计算量不能太大，其核心问题是：寻找一个变换，使图像间相互重叠的部分对准，并“缝合”成为一个新的大视角的视图。精确配准的关键就是要有能很好的描述两幅图像之间转化关系的模型。下面分别介绍了常用的图像配准算法。

2.3.1 基于特征匹配法

利用图像的自身特性，先提取图像中的一些特征点，然后借助这些特征点进行匹配，这就是基于特征的匹配算法。特征点可以是图像上的某些特殊点，如边缘点、角点、拐点等。由于图像背景十分复杂，而且图像有旋转，考虑工程实际的需要和算法的通用性，则图像拼接算法的实现步骤如下^[12]：(1) 分析图像，找出两幅待拼接图像中共同部分的两个特征区域块；(2) 采用图像分割和边缘检测技术从图像中将该特征块提取出来；(3) 计算特征块的形心位置，将特征块的匹配问题转换为特征点的匹配问题；(4) 利用形心位置实现两幅有旋转的图像之间的平滑连接。

(1) 边缘检测和分割算法：

精确定位特征区域块的边缘有利于提高图像测量的精度，其中边缘检测算法的选取至关重要。具体实现方法如下：

a. 对于数字图像 $\{f(i, j)\}$ ，利用 *Sobel* 边缘检测算子检测图像的水平 and 垂直边缘分量 S_x 和 S_y 。水平和垂直边缘检测模板分别为

$$M_x = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad \text{和} \quad M_y = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad (2-1)$$

b. 水平和垂直边缘检测后的分量合成为

$$S(i, j) = (s_x^2 + s_y^2)^{\frac{1}{2}} \quad (2-2)$$

形成边缘检测后的图像 $\{S(i, j)\}$ ，在实际使用时，如果用选定的阈值进行二值化，使得边缘图中幅值较小的边缘丢失了，为克服这个缺陷，对 $S(i, j)$ 引入衰减因子 d ，即

$$S(i, j) = (s_x^2 + s_y^2)^{\frac{1}{2}} / d \quad (2-3)$$

从而保留了图中较小边缘的数值，不需进行二值处理，得到不失真的灰阶边缘图。

c. 对灰阶边缘图进行内插运算。选用三次多项式

$$w(\omega) = \begin{cases} 1 - 2|\omega|^2 + |\omega|^3, & |\omega| < 1 \\ 4 - 8|\omega| + 5|\omega|^2 - |\omega|^3, & 1 \leq |\omega| \leq 2 \\ 0, & |\omega| > 2 \end{cases} \quad (2-4)$$

d. 对插值后灰阶图像，利用自适应全局阈值法选取最优阈值进行图像分割，然后再进行细化处理，使目标边缘定位在亚像素级。

(2) 形心位置确定：

图像经过边缘提取、分割后获得特征块的形状特征，采用链码方法确定特征块形心位置 $o(x_c, y_c)$ 。特征块对 x 轴（水平方向）积分的面积为：

$$S = \sum_{i=1}^n a_{i0}(y_{i-1} + a_{i2}/2) \quad (2-5)$$

式中 $y_i = y_{i-1} + a_{i2}$ ， y_0 是初始点的纵坐标， a_{i0} 和 a_{i2} 分别是链码第 i 环的长度在水平和垂直方向的分量，对于封闭链码（初始点坐标等于终点坐标）， y_0 可任意选择，则特征块形心坐标 $o(x_c, y_c)$ 为

$$x_c = M_1^x / S, \quad y_c = M_1^y / S, \quad (2-6)$$

式中 M_1^x 和 M_1^y 分别是链码关于 x 轴和 y 轴的一阶矩， M_1^x 的计算方法为

$$M_1^x = \sum_{i=1}^n \frac{a_{i0} [y_{i-1}^2 + a_{i2}(y_{i-1} + a_{i2}/3)]}{2} \quad (2-7)$$

M_1^y 的计算方法是先将链码旋转 $\pi/2$ ，即可对链的每一环旋转 $\pi/2$ 的变换。

$ai' = ai + 2(\text{mod } 8)$ ， $i=1, 2, \dots, n$ ，然后利用公式 (2-7) 进行计算。

(3) 图像匹配拼接：

如图 2-3 所示，有 A, B 两幅图像，它们有共同的重叠部分，假设在重叠部分中，图像 A 中的点 $P(x_i, y_i)$ ， $P(x_s, y_s)$ 与图像 B 中的点 $P(x_s, y_i)$ ， $P(x_i, y_s)$ 在实际物空间中分别是两个不同的特征对应点，图像匹配拼接方法为：

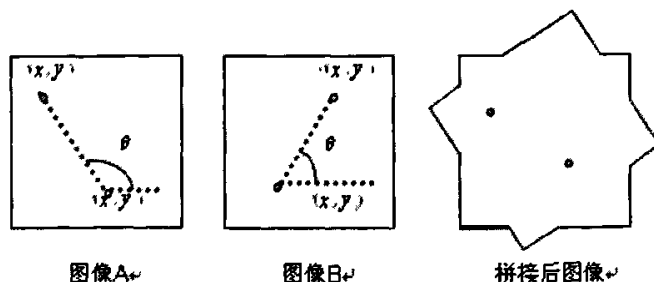


图 2-3 特征匹配

Fig.2-3 Feature matching

首先计算图像 A , B 之间的旋转角度, 设 $P(x_1, y_1)$ 和 $P(x_2, y_2)$ 的连线长度为 r_1 , 与水平向右方向的夹角为 θ_1 ; $P(x_3, y_3)$, $P(x_4, y_4)$ 连线长度为 r_2 , 与水平向右方向的夹角为 θ_2 , 则有

$$\begin{cases} \gamma_1 = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \\ \cos \theta_1 = (x_1 - x_2) / \gamma_1 \\ \sin \theta_1 = (y_1 - y_2) / \gamma_1 \end{cases}, \begin{cases} \gamma_2 = \sqrt{(x_3 - x_4)^2 + (y_3 - y_4)^2} \\ \cos \theta_2 = (x_3 - x_4) / \gamma_2 \\ \sin \theta_2 = (y_3 - y_4) / \gamma_2 \end{cases} \quad (2-8)$$

设两幅图像之间的旋转角度为 θ , 则

$$\begin{cases} \cos \theta = \cos(\theta_1 - \theta_2) = \cos \theta_1 \cos \theta_2 + \sin \theta_1 \sin \theta_2 \\ \sin \theta = \sin(\theta_1 - \theta_2) = \sin \theta_1 \cos \theta_2 - \cos \theta_1 \sin \theta_2 \end{cases} \quad (2-9)$$

接着图像旋转。将图像 B 旋转 θ 角度后为 B' , 设 (x, y) 为图像旋转前某点坐标, 旋转后对应坐标为 (x', y') , 其对应关系如下式所示:

$$\begin{cases} x' = x \cos \theta - y \sin \theta \\ y' = x \sin \theta + y \cos \theta \end{cases} \quad (2-10)$$

最后图像平移平滑拼接。图像 B 中的点 $P(x_3, y_3)$ 和 $P(x_4, y_4)$ 经过旋转后分别对应图像 B' 中的点 $P(x_3', y_3')$ 和 $P(x_4', y_4')$, 则图像 A 中的点 $P(x_1, y_1)$ 和 $P(x_2, y_2)$ 连线平行于图像 B' 中的点 $P(x_3', y_3')$ 和 $P(x_4', y_4')$ 连线, 并且方向一致。在实际物空间中, 图像 B' 中的点 $P(x_3', y_3')$ 和 $P(x_4', y_4')$ 分别对应于图像 A 中点 $P(x_1, y_1)$ 和 $P(x_2, y_2)$ 。找出匹配特征点后, 对于重叠部分进行平滑连接。

2.3.2 基于频域的图像拼接算法

由文献[1,13~17]知, 相位相关度法最早是由 Kuglin 和 Hines 在 1975 年提出来的, 是一种非线性、基于傅氏功率谱的频域相关算法, 由于采用该方法只取互功率谱中的相位信息, 因而减少了对图像内容的依赖性, 而且所获得的相关峰尖锐突出, 因此, 位移检测范围较大, 有很高的匹配精度。另外, 相位相关

技术对图像灰度依赖较小,具有一定抗干扰能力。但它对图像旋转特别敏感。而采用旋转不变性相位相关法能够对带有旋转的两幅图判别其是否相关,同时能测量出两个图像的旋转角度而不必考虑旋转轴的位置。

该方法充分利用了傅立叶变换的良好性质,即函数平移,旋转和变尺度在频率域都有其对称性,对于图像的平移,计算两幅图像功率谱的逆傅立叶变换可得一个脉冲函数,该函数仅在平移量处不为零。对于旋转,可使用极坐标方式表示,用相同方法计算图像间的旋转角度。假如图像运动是平移和旋转的混合运动,我们可分两步运算:先计算旋转后计算平移。该方法对小平移量和旋转及变尺度图像拼接非常适合。随着傅立叶变换的广泛应用以及硬件的发展,相位相关度法得到了快速发展,该算法可适合多传感器和光源变化采集的图像,相对于基于像素灰度级的方法,该算法计算量大,在注重拼接精度的场合可以不考虑速度,可采用此法。

具体算法分析如下:设两幅图像 $f(x, y)$ 与 $f(x', y')$ 之间有一旋转、平移关系,即:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos \theta_0 & \sin \theta_0 \\ -\sin \theta_0 & \cos \theta_0 \end{bmatrix} \begin{bmatrix} x - x_0 \\ y - y_0 \end{bmatrix} \quad (2-11)$$

(θ_0 为旋转量; (x_0, y_0) 为平移量) 其对应的傅立叶变换满足如下关系:

$$F'(\omega_1, \omega_2) = F(\omega_1 \cos \theta_0 + \omega_2 \sin \theta_0, -\omega_1 \sin \theta_0 + \omega_2 \cos \theta_0) \times e^{-j[(\omega_1 \cos \theta_0 + \omega_2 \sin \theta_0)x_0 + (-\omega_1 \sin \theta_0 + \omega_2 \cos \theta_0)y_0]} \quad (2-12)$$

由此可见,空域中的平移在频域中只反映在相位变化,频域中的幅谱只反映空域中的旋转量,而幅谱经过极坐标变换后对应的旋转量转换成了平移量,即在极坐标系中有:

$$|F'(\rho, \theta)| = |F(\rho, \theta - \theta_0)|, \text{ 其中 } \rho = \sqrt{\omega_1^2 + \omega_2^2}, \theta = \arctan \frac{\omega_2}{\omega_1} \quad (2-13)$$

因此,两图像的平移量可以通过做相位合成后再做傅立叶反变换即可得到相关表面,其最大峰值超过某一阈值时表示两图像相关,对应峰值的位置偏移量精确地反映了图像的二维平移量,其 POC 算法流程图如图 2-4 所示。而用来测量旋转量的 RIPOC 算法流程图如图 2-5 所示,其中对幅谱取对数运算是为了降低其动态范围。当两幅图像既有平移量又有旋转量时,可按照图 2-6 所示的流程处理;同时也不难看出,对于带有尺度变换的图像只要在 RIPOC 算法流程中的极坐标变换后加上取对数运算即可求出其尺度变换比。

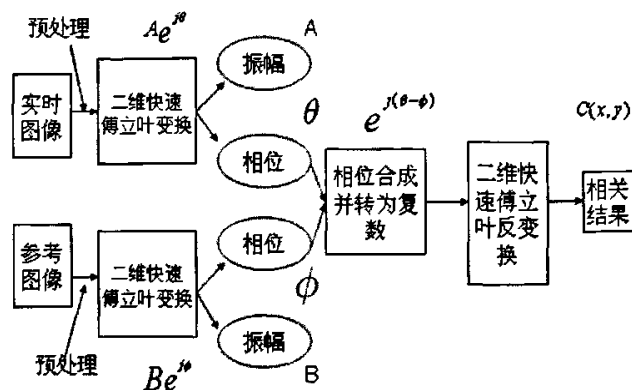


图 2-4 POC 算法流程

Fig.2-4 Flow process chart of POC algorithm

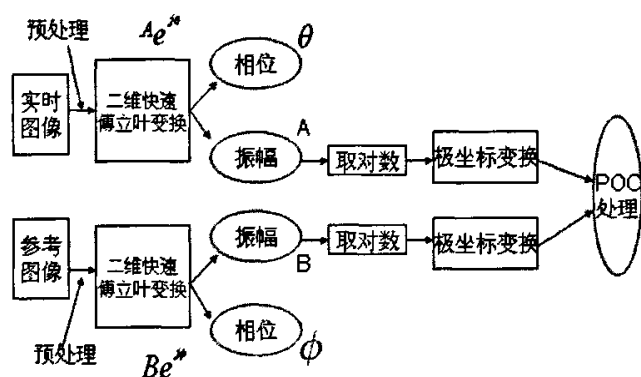


图 2-5 RIPOC 算法流程图

Fig.2-5 Flow process chart of RIPOC algorithm

相位相关度法只利用了互功率谱中的相位信息进行图像配准，对图像间的亮度变化不敏感，具有一定的抗干扰能力，而且所获得的相关峰尖锐突出，位移检测范围大，具有较高的匹配精度，但该算法需要计算量大，且需要变换，时间开销大，如果图像质量好，采用此法拼接速度不能满足实际要求^[18,19]。

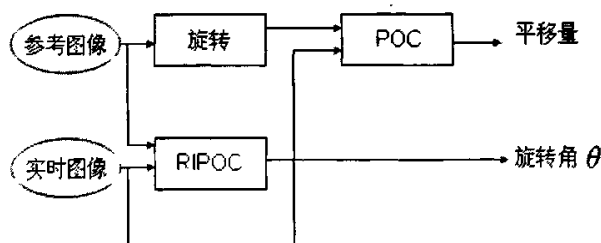


图 2-6 总流程图

Fig.2-6 The gross process

2.3.3 基于像素灰度级的方法

基于像素灰度级的方法：一般是取出待匹配图像的重叠部分有特征的像素，通过评价函数给出待配准图像与基准图像之间的相似度值。通常基于像素灰度级的方法主要有^[2]：①比值匹配法，计算量不大，精度低；②模板匹配法。对于模板匹配法，如果采用的评价函数不同，得到的模板也不同，可分为最小差值法和极大相关系数法。灰度图像拼接主要是根据两图像重叠区域所对应点像素灰度值的相似性，自动寻找匹配位置，基于像素灰度级的匹配算法非常适合灰度图像拼接。下面分别介绍比值法和模板匹配法。

(1) 比值法：

比值法的原理比较简单，就是在第一幅图像的重叠区域里取出两列像素，以它们的比值作为模板，在第二幅图像中找到对应位置，从而确定重叠范围。

设待拼接源图像为 *image1* 和 *image2*，图像重叠区域在两幅源图像里所占比例在 20%~50%。设 $T[i][j]$ 表示图像 *T* 上第 *i* 行、第 *j* 列像素的灰度值，则比值匹配模板的构造就是在 *image1* 重叠区域中找到间隔一定距离的两列上的部分像素，将两列对应像素的比值的集合作为模板。比值法初始模板用 *M* 表示， $M[k]$ 表示模板 *M* 中第 *K* 个成员，*span* 表示这两列像素的水平间隔，则模板的生成公式为：

$$M[k]=S[i][k]/S[i][k+span] \quad (2-14)$$

初始模板提取后，接着在 *image2* 上进行相似模板的搜索，相似模板用 *N* 表示，则对于 *image2* 上间距为 *span* 的两列，计算出由此构成的相似模板 *N*，并且逐一同初始模板 *M* 比较。可用 *M* 与 *N* 对应的像素点的差的绝对值和 *E* 作为评价函数来计算 *M* 与 *N* 之间的相似程度，如公式 (2-15)：

$$E = \sum_{i=0}^{length} (|N[i] - M[i]|) \quad (2-15)$$

两个模板的相似程度越高 *E* 值就越小。当 *E* 取最小值时，认为已经找到正确匹配的位置，它是以第一列向量的第一个像素成员在待匹配图像中的坐标位置为准^[20~22]。

(2) 模板匹配法：

模板就是一幅已知的小图像。模板匹配就是在一幅大图像中搜寻目标，已知该图中有要找的目标，且该目标同模板有相同的尺寸、方向和图像，通过一定的算法可以在图中找到目标，确定其坐标位置。

在 *image_1* 中选取一个正方形网格作为初始模板 *T*，*T* 在 *image_2* 上遍历，通过一定的算法可以在图 *image_2* 中找到目标，确定其坐标位置。常用的两种测度：初始模板与 *image_2* 对应区域像素误差的平方和，平方和最小值对应点

为匹配点；初始模板与 image_2 对应区域像素误差的绝对值之和，绝对值和最小值对应点为匹配点。只要保证待拼接图像质量好，模板匹配总能找到匹配点，但模板匹配法计算量大。

以 8 位图像（其 1 各像素由 1 各字节描述）为例，模板 T ($m \times n$ 个像素) 叠放在被搜索图 S ($W \times H$ 个像素) 上平移，模板覆盖被搜索图的那块区域叫子图 S_{ij} , i, j 为子图左上角在被搜索图 S 上的坐标。搜索范围是：

$$\begin{cases} 1 \leq i \leq W - m \\ 1 \leq j \leq H - n \end{cases} \quad (2-16)$$

通过比较 T 和 S_{ij} 的相似性，完成模板匹配过程。

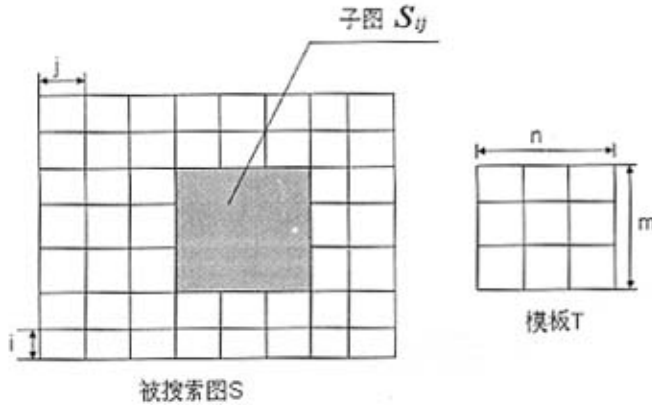


图 2-7 模板匹配示意图

Fig.2-7 Experiment of Template Matching

注意：图像的数据是从下到上、从左到右排列的。模板匹配法评价函数主要有两个：极大相关系数和最小差值。

可以用下式衡量 T 和 S_{ij} 相似性：

$$\begin{aligned} D(i, j) &= \sum_{m=1}^M \sum_{n=1}^N [S^{ij}(m, n) - T(m, n)]^2 \\ &= \sum_{m=1}^M \sum_{n=1}^N [S^{ij}(m, n)]^2 - 2 \sum_{m=1}^M \sum_{n=1}^N S^{ij}(m, n) \times T(m, n) + \sum_{m=1}^M \sum_{n=1}^N [T(m, n)]^2 \end{aligned} \quad (2-17)$$

上式的第一项为子图的能量，第三项为模板的能量，都与模板匹配无关。第二项是模板和子图的互相关，随 (i, j) 而改变。当模板和子图匹配时，该项有极大值。将其归一化，得模板匹配的相关系数：

$$R(i, j) = \frac{\sum_{m=1}^M \sum_{n=1}^N S^y(m, n) \times T(m, n)}{\sqrt{\sum_{m=1}^M \sum_{n=1}^N [S^y(m, n)]^2} \sqrt{\sum_{m=1}^M \sum_{n=1}^N [T(m, n)]^2}} \quad (2-18)$$

当模板和子图完全一样时, 相关系数 $R(i, j)=1$ 。在被搜索图 S 中完成全部搜索后, 找出 R 的最大值 $R_{max}(i_m, j_m)$, 其对应的子图 S_{i_m, j_m} 即为匹配目标。显然用这种公式做图像匹配计算量大, 速度较慢。

另一种评价函数是衡量 T 和 S_{ij} 的误差, 其公式为:

$$E(i, j) = \sum_{m=1}^M \sum_{n=1}^N |S^y(m, n) - T(m, n)| \quad (2-19)$$

$E(i, j)$ 为最小值处即为匹配目标。为提高计算速度, 取一个误差阈值 E_0 , 当 $E(i, j) > E_0$ 时就停止该点的计算, 继续下一点计算。

两种方法模板一样, 只是搜索匹配点采用的评价函数不一样, 因此拼接时间也不一样。

2.4 图像融合算法研究

图像融合研究就是为了消除拼接点的缝隙。在不同时刻的图像采样, 光强都会有差异, 得到的图像就会存在亮度差, 这就很容易导致图像拼接的边界处有明显的缝隙。图像拼接点找到后, 如果仅仅是简单的叠加, 会造成图像的模糊和明显的边界, 这是不能容忍的。图像融合就是要消除图像光强或色彩的不连续性。常用的融合算法有^[12]: ①取平均值法; ②加权平均法; ③基于欧氏距离的有效权重法; ④双线性插值法。

2.4.1 取平均值法

令 $I_1(x, y)$, $I_2(x, y)$ 和 $I(x, y)$ 分别表示第一幅图像, 第二幅图像和融合后图像在点 (x, y) 处的像素值, 则融合图像中各点的像素值按公式(2-20)来计算。

$$I(x, y) = \begin{cases} I_1(x, y) & (x, y) \in R_1 \\ \frac{1}{2}(I_1(x, y) + I_2(x, y)) & (x, y) \in R_2 \\ I_2(x, y) & (x, y) \in R_3 \end{cases} \quad (2-20)$$

公式(2-20)中, R_1 表示第一幅图像中去掉重叠部分剩下的图像区域对应像素值, R_2 表示第一幅图像和第二幅图像重叠的图像区域对应像素值, R_3 表示第二幅图像中去掉重叠部分剩下的图像区域对应像素值。

如果待拼接图像质量很好, 可以采用此法, 取两幅图像的平均值作为重叠

区域的像素效果一般不能令人满意，在融合部分有明显的带状感觉，用眼睛观察有明显的亮度上的差异^[23, 24]。

2.4.2 加权平均法

加权平均法思想就是在重叠部分由前一幅图像慢慢过渡到第二幅图像并删去垂直方向错开的图像部分，常用空域的线性加权^[25~29]。

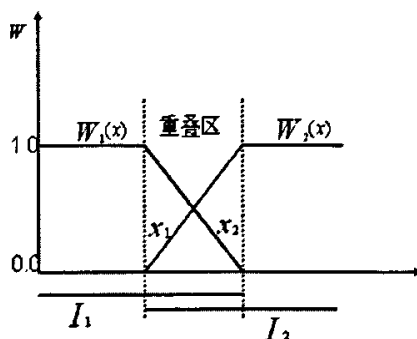


图 2-8 加权平均法

Fig.2-8 the weighted average method

如图 2-8 所示，相邻图像 I_1 , I_2 在区间 $[x_1, x_2]$ 上重叠， $W_1(x)$ 、 $W_2(x)$ 为加权函数，一个常用的加权函数如式 (2-21)。那么重叠图像 I 在这个区间上 x 点的值如式 (2-22)。

$$W_2(x) = 1 - W_1(x) = 1 - i/W \quad (2-21)$$

这里的 $0 \leq i \leq W-1$, W 为重叠区域的宽度：

$$I(x) = I_1(x) \times W_1(x) + I_2(x) \times W_2(x) \quad (2-22)$$

2.4.3 基于欧氏距离的有效权重法

图像的每个像素都分配权重。这个权重与到边缘（或最近不可见点）的距离成比例。在拼接时，主要目的是减少离边缘近的像素点的光强贡献。融合算法中计算距离映射 $d(x, y)$ ，利用块距离和欧氏距离，计算到最近的透明点 ($\alpha=0$) 或边的距离。融合所有变形图像公式：

$$C(x, y) = \frac{\sum_k \omega(d(x, y)) I_k(x, y)}{\sum_k \omega(d(x, y))} \quad (2-23)$$

$\omega(x)$ 是单调函数, $\omega(x)=x$ 。 I_k 是第 k 幅变形图像的光强函数。 d 的计算很简单, 取离矩形四条边距离的最小值^[30, 31]。

2.4.4 双线性插值法

双线性插值法比较简单, 设图像 I_1 和 I_2 交叠区的 x 和 y 轴最大和最小值分别为 $X_{\max}, X_{\min}$ 和 $Y_{\max}, Y_{\min}$, 则融合图像 I 在交叠区的颜色值为

$$I = \beta I_1(x, y) + (1 - \beta) I_2(x, y) \quad (2-24)$$

其中, $\beta = (x_{\max} - x) / (x_{\max} - x_{\min})$, 这种方法在多数情况下都可以取得比较好的效果。本设计就是采用此法来做图像融合^[32~35]。

2.5 本章小结

本章是图像拼接的理论部分, 在本章中分析了图像拼接的基本步骤及拼接关键技术。图像匹配是研究的重点, 分别介绍了基于特征匹配法、相位相关度法和基于像素灰度级的方法, 其中, 相位相关度法和基于像素灰度级的方法是我们讨论的核心, 而基于像素灰度级的方法更是本课题研究的关键。下面的第三章和第四章就相位相关度法和基于像素灰度级的方法作了更进一步的讨论。最后介绍了常用的图像融合算法。

第3章 基于频域的图像拼接算法

1994年Chen等人提出了一种基于Fourier-Mellin变换的图像配准方法。该方法假设待配准的两幅图像，其中一幅是另一幅经过旋转(rotation)、尺度伸缩(scaling)和平移(translation)变换后的图像(简称RST变换)。该方法通过计算两幅图像功率谱的反Fourier变换所对应的峰值位置求取它们的相对平移，通过对图像幅度谱进行对数-极(Log-Polar)变换，在对数-极空间用类似于计算相对平移的方法来求得相对旋转角度和尺度缩放因子，从而实现了满足RST变换关系的两幅图像之间的配准。

Chen和Reddy等人还就该算法针对遥感图像、医学图像、目标识别等应用进行了大量实验，结果发现，该方法除具有较高的精度外，对诸如一致性的照明变化、两幅图像内容的部分不相同、部分遮挡、系统校正误差、加性噪声等的影响具有很强的鲁棒性^[15]。

下面首先介绍了离散傅立叶变换基础，接着论述了相位相关度法原理，以及其实现步骤，分析了相位相关度法的优缺点，最后总结了基于频域的图像配准算法。

3.1 离散傅立叶变换基础

离散傅立叶变换(Discrete Fourier Transform, DCT)是一种常见的正交变换，它在数字图像处理中有着非常广泛的应用。

傅立叶变换在数学中的定义是严格的。设 $f(x)$ 为 x 的函数，如果 $f(x)$ 满足下面的狄里克莱条件：

1. 具有有限个间断点；
2. 具有有限个极值点；
3. 绝对可积。

则有下列两式成立：

$$F(u) = \int_{-\infty}^{+\infty} f(x)e^{-j2\pi ux} dx \quad (3-1)$$

其逆变换公式为：

$$f(x) = \int_{-\infty}^{+\infty} F(u)e^{j2\pi ux} du \quad (3-2)$$

其中， x 为时域变量， u 为频域变量。通常把以上公式称为傅立叶变换对。

3.1.1 一维离散傅立叶变换

设一维连续函数 $f(x)$ 的傅立叶变换为:

$$F(u) = \int_{-\infty}^{+\infty} f(x)e^{-j2\pi ux} dx \quad (3-3)$$

则可用一个离散和去逼近这一积分, 从而得到一维离散函数的傅立叶变换为:

$$F_l = \sum_{k=0}^{N-1} f_k W^{-lk} \quad (3-4)$$

其中, $l=0, 1, 2, \dots, N-1$; $W=e^{j2\pi/N}$ 。

F_l 是对 $F(u)$ 等间距采样得到的 N 个样本 $F_0, F_1, F_2, \dots, F_{N-1}$, (x_0, y_0) 是对 $f(x)$ 等间距采样得到的 N 个样本 $f_0, f_1, f_2, \dots, f_{N-1}$ 。

同样的, 可以定义逆变换公式为:

$$f_k = \frac{1}{N} \sum_{l=0}^{N-1} F_l W^{lk} \quad (3-5)$$

由以上公式可以看出, 离散傅立叶变换的结果是一个复数表达式。设 $R(u)$ 为其实部, $I(u)$ 为其虚部, 则:

$$F(u) = R(u) + jI(u) \quad (3-6)$$

写成指数形式即为:

$$F(u) = |F(u)|e^{j\phi(u)} \quad (3-7)$$

其中,

$$|F(u)| = \sqrt{R(u)^2 + I(u)^2} \quad (3-8)$$

$$\phi(u) = \arctan \frac{I(u)}{R(u)} \quad (3-9)$$

通常 $|F(u)|$ 为 $f(x)$ 的傅立叶幅度谱, $\phi(u)$ 为 $f(x)$ 的相位谱。幅度谱的平方称为能量谱或功率谱, 记作 $E(u)$:

$$E(u) = R(u)^2 + I(u)^2 \quad (3-10)$$

3.1.2 二维离散傅立叶变换

傅立叶变换可推广到二维函数。在数字图像处理中用的最多的实二维离散傅立叶变换, 如果一幅二维离散图像 $f(x, y)$ 的大小为 $M \times N$, 则二维离散傅立叶

变换定义为:

$$F(u, v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-j2\pi(\frac{ux}{M} + \frac{vy}{N})} \quad (3-11)$$

其中, $u=0, 1, \dots, M-1$; $v=0, 1, \dots, N-1$ 。

与一维时的情况相似, 可定义二维离散傅立叶变换的幅度谱、相位谱和能量谱为:

$$|F(u, v)| = \sqrt{R(u, v)^2 + I(u, v)^2} \quad (3-12)$$

$$\phi(u, v) = \arctan \frac{I(u, v)}{R(u, v)} \quad (3-13)$$

$$E(u, v) = R(u, v)^2 + I(u, v)^2 \quad (3-14)$$

3.1.3 二维离散傅立叶变换性质

傅立叶变换有许多重要的性质, 这些性质为实际运算处理提供了极大的便利。图像处理用的是二维傅立叶变换, 所以下面列出了二维傅立叶变换的重要性质:

(1) 可平移性

离散傅立叶变换和逆变换对的位移性质是指:

$$F(u - u_0, v - v_0) \sim f(x, y) e^{j2\pi(\frac{u_0 x + v_0 y}{N})} \quad (3-15)$$

$$f(x - x_0, y - y_0) \sim F(u, v) e^{j2\pi(\frac{ux_0 + vy_0}{N})} \quad (3-16)$$

通过上面的对应关系可以看出, 如果再空间域中图像平移到点 (x_0, y_0) 处, 则其对应的傅立叶变换要乘上一个系数 $e^{-j2\pi j(ux_0 + vy_0)}$, 即表明空间域中图像的平移对应于频域中的相移, 因为 $e^{-j2\pi j(ux_0 + vy_0)}$ 的幅值为 1, 所以频域中发生相移, 其傅立叶的幅值不变。

(2) 共轭对称性

如果离散函数 $f(x, y)$ 的傅立叶变换为 $F(u, v)$, $F^*(-u, -v)$ 为 $f(-x, -y)$ 离散傅立叶变换的共轭函数, 那么:

$$F(u, v) = F^*(-u, -v) \quad (3-17)$$

$$|F(u, v)| = |F(-u, -v)| \quad (3-18)$$

离散傅立叶变换的共轭性说明离散函数 $f(x, y)$ 经过正交变换后得到的 $F(u, v)$ 是以原点为中心对称的。利用该特性, 只要求出半个周期内的值就可以

得到整个周期的值。

(3) 旋转不变性

将离散函数 $f(x, y)$ 及其离散傅立叶变换 $F(u, v)$ 转换为极坐标形式 $f(r, \theta)$ 和 $F(k, \phi)$ 。其中, $x=r\cos\theta, y=r\sin\theta, u=k\cos\phi, v=k\sin\phi$ 。如果空间域中离散函数旋转角度 θ_0 , 则在变换域中该离散函数的离散傅立叶变换函数也将旋转同样的角度。反之, 如果频域 $F(u, v)$ 旋转某一角度 θ_0 , 则 $f(x, y)$ 在空间域也旋转同样的角度 θ_0 。即有:

$$f(r, \theta + \theta_0) \Leftrightarrow F(k, \phi + \theta_0) \quad (3-19)$$

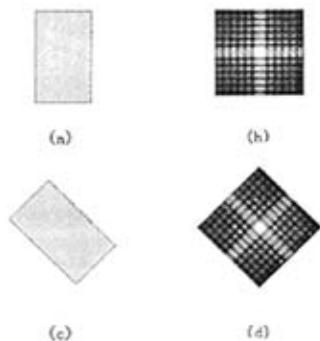


图 3-1 旋转不变性

Fig.3-1 The theorem of revolution invariant

(4) 帕斯维尔定理

这个性质也称为能量保持定理。如果 $F(u, v)$ 是 $f(x, y)$ 的傅立叶变换, 那么有下式成立:

$$\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} |f(x, y)|^2 = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} |F(u, v)|^2 \quad (3-20)$$

这个性质说明变换前后并不损失能量。

(5) 相关定理

如果 $f(x), g(x)$ 为两个一维时域函数, $f(x, y), g(x, y)$ 为两个二维空域函数, 那么, 定义下两式为相关运算:

$$f(x) \circ g(x) = \int_{-\infty}^{+\infty} f(\alpha) g(x + \alpha) d\alpha \quad (3-21)$$

$$f(x, y) \circ g(x, y) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(\alpha, \beta) g(x + \alpha, y + \beta) d\alpha d\beta \quad (3-22)$$

式中“o”符号表示相关运算。由以上定义可引出傅立叶变换的一个重要性质, 那就是相关定理, 即:

$$f(x) \circ g(x) \Leftrightarrow F(u, v) * G^*(u, v) \quad (3-23)$$

$$f(x, y) \circ g^*(x, y) \Leftrightarrow F(u, v) \circ G(u, v) \quad (3-24)$$

式中 $F(u, v)$ 是 $f(x, y)$ 的傅立叶变换, $G(u, v)$ 是 $g(x, y)$ 的傅立叶变换, $G^{*(u, v)}$ 是 $G(u, v)$ 的共轭, $g^{*(x, y)}$ 是 $g(x, y)$ 的共轭。

(6) 卷积定理

如果 $f(x)$, $g(x)$ 为两个一维时域函数, $f(x, y)$, $g(x, y)$ 为两个二维空域函数, 那么, 定义下两式为相关运算, 即:

$$f(x) * g(x) = \int_{-\infty}^{+\infty} f(\alpha)g(x-\alpha)d\alpha \quad (3-25)$$

$$f(x, y) * g(x, y) = \int_{-\infty}^{+\infty} f(\alpha, \beta)g(x-\alpha, y-\beta)d\alpha d\beta \quad (3-26)$$

式中 “*” 符号表示卷积运算。由此可得到傅立叶变换的卷积定理如下:

$$f(x) * g(x) \Leftrightarrow F(u, v) \bullet G(u, v) \quad (3-27)$$

$$f(x) \bullet g(x) \Leftrightarrow F(u, v) * G(u, v) \quad (3-28)$$

式中 $F(u, v)$ 和 $G(u, v)$ 分别是 $f(x, y)$ 和 $g(x, y)$ 的傅立叶变换。

在一些涉及相关或者卷积的运算中, 一般都是利用此性质将求它们在时域的卷积转化为求频域内的乘积来简化运算过程。

3.2 相位相关度法原理

基于频域变换的配准方法是利用了图像在频域上的一些重要性质: 平移、旋转、缩放不变性, 也就是说图像之间在空域上的平移、旋转、缩放都可以在频域上找到一一的对应。快速傅立叶变换在很大程度上提高了算法的运算效率, 同时傅立叶变换对于频率相关的噪声表现出了良好的鲁棒性。因此相位相关的方法是实现图像配准的一种比较有效的算法:

假设图像 f_1 和 f_2 之间只是存在着 (x_0, y_0) 的平移:

$$f_2(x, y) = f_1(x + x_0, y + y_0) \quad (3-29)$$

它们对应的傅立叶变换则会有如下的关系:

$$\frac{F_2(w_x, w_y)}{F_1(w_x, w_y)} = e^{j(w_x x_0 + w_y y_0)} \quad (3-30)$$

两幅图像的互功率谱则定义为:

$$\frac{F_1(w_x, w_y)F_2^*(w_x, w_y)}{|F_1(w_x, w_y)| |F_2^*(w_x, w_y)|} = e^{j(w_x x_0 + w_y y_0)} \quad (3-31)$$

其中, F^* 表示的是 F 的共轭。而且转移理论保证了互功率谱的相位等于图像之间相位的不同。通过对上式做反傅立叶变换将会得到一个脉冲函数 (图 3-2);

$$F^{-1}\left(\frac{F_1(w_x, w_y)F_2^*(w_x, w_y)}{|F_1(w_x, w_y)||F_2^*(w_x, w_y)|}\right) = \delta(x - x_0, y - y_0) \quad (3-32)$$

也就是说, 除了在需要配准的平移位置, 其余位置它的值几乎都是 0。

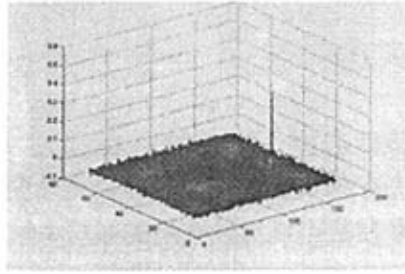


图 3-2 相位相关峰值

Fig.3-2 Peak point of phase only correlation

对于两幅图像之间的旋转角度, 根据傅立叶变换的旋转属性, 把图像从直角坐标系转成极坐标系, 旋转的计算就可以转换成简单的图像平移求取。假设图像 f_1 和 f_2 之间只是存在着 θ_0 角度的旋转:

$$f_2(x, y) = f_1(x \cos \theta_0 + y \sin \theta_0, -x \sin \theta_0 + y \cos \theta_0) \quad (3-33)$$

它们的傅立叶变换存在以下关系:

$$F_2(w_x, w_y) = F_1(w_x \cos \theta_0 + w_y \sin \theta_0, -w_x \sin \theta_0 + w_y \cos \theta_0) \quad (3-34)$$

把它们转换成极坐标系之后, 求取傅立叶变换的模, 则:

$$M_2(\rho, \theta) = M_1(\rho, \theta + \theta_0) \quad (3-35)$$

从上式发现, 通过极坐标系的变换之后旋转角度的计算就可以降为对平移的计算, 然后根据前面的式子可以容易的找到旋转角度。

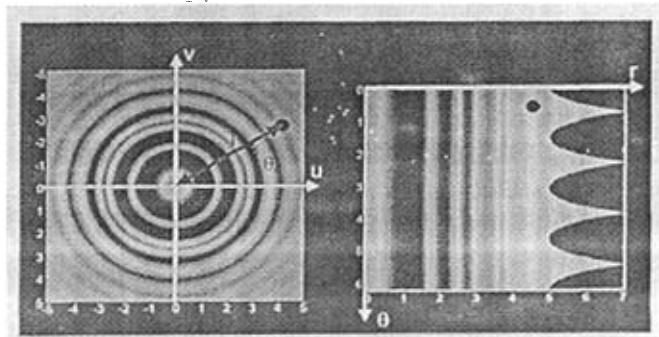


图 3-3 直角坐标系到极坐标系的示意图

Fig.3-3 Conversion between rectangular coordinate system and polar coordinate system

对于存在低频噪声的图像进行拼接, 可以采用基于傅立叶变换的相位相关

度法。根据前面分析, 相位相关度法具有对噪声有较高的容忍度、检测结果与照度无关以及受几何失真小等突出优点。但要注意一个问题, 理论上, 傅立叶频谱随着图像旋转而旋转, 当旋转角度大于 5 度时其相关峰值衰减为 0。而实际上, 由于图像之间存在着的不同区域、不同的灰度以及一定的噪声, 通常旋转角度不到 5 度时就已经衰减为 0。所以在既存在着旋转又存在着平移的图像之间的配准时首先需要对旋转进行校正^[16]。

3.3 相位相关度法实现步骤

一般的图像对齐算法都是基于空间域的, 这里提到的相位相关度法从另一个角度出发, 即从频域考虑如何实现图像对齐。充分利用了二维傅立叶变换的平移性质, 两幅图像的平移量可以通过它们的互功率谱的相位直接计算出来。

基于傅立叶变换的相位相关度法实现步骤如下^[1]:

1. 读入两幅图像 f_1 和 f_2 (函数输入);
2. 分别对 f_1 和 f_2 作二维傅立叶变换 F_1 和 F_2 ;
3. 根据前面的公式计算规格化互功率谱;
4. 对互功率谱进行傅立叶反变换, 并求其幅值;
5. 找出幅值最大值的位置即为两幅图像 f_1 和 f_2 的平移量。
6. 对两幅图像进行融合算法处理, 拼接为一幅完整的图像。

设只含 (x_0, y_0) 平移关系的参考图像 f_1 和 f_2 对应的傅立叶变换分别为 $G(\xi, \zeta)$ 和 $F(\xi, \zeta)$, 则:

$$C(\xi, \zeta) = \frac{\angle F(\xi, \zeta)}{\angle G(\xi, \zeta)} = \frac{F(\xi, \zeta)G^*(\xi, \zeta)}{|F(\xi, \zeta)G^*(\xi, \zeta)|} = \exp(-j2\pi(\xi * x_0 + \zeta * y_0)) \quad (3-36)$$

再对 $C(\xi, \zeta)$ 进行傅立叶反变换即可得到相关函数 $c(x, y)$, 相关函数理想情况下为 δ 函数, 其峰值代表两图像的相关度, 峰值所对应的坐标即为 (x_0, y_0) 。对应的拼接流程图如图 3-4 所示。

如果待拼接的两幅图像之间既有平移量又有旋转量, 可采取先求旋转量, 接着根据这个旋转角度旋转回来, 然后再求平移量的方法。含有平移量和旋转量的相位相关度法流程图可参考图 3-5 所示。

为了对齐两幅以上的图像, 对齐位置可以转化为图像对之间的位置偏移, 累加相邻图像间的局部偏移就得到每幅图像的全局偏移量。例如, t_{12} 和 t_{13} 表示图像对 I_1 和 I_2 和图像对 I_1 和 I_3 的偏移位置, 那么 I_2 和 I_3 之间的位置偏移是 $t_{13} - t_{12}$ 。在输入图像序列中, 可以任意选取最左图像作为全局偏移位置的参考帧。任意图像 i 的全局偏移位置 P 为它之前两两相邻图像间局部偏移的累加:

$$p_i = \sum_{j=1}^{i-1} t_{j,j+1} \quad (3-37)$$

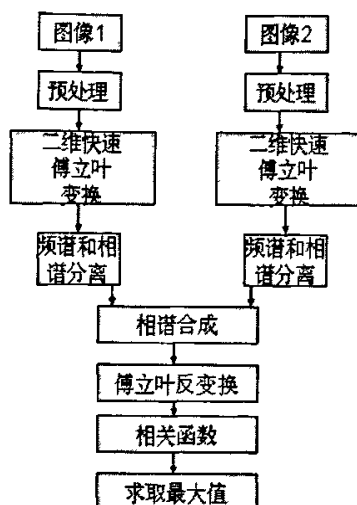


图 3-4 只含平移量的相位相关拼接流程图

Fig.3-4 Flow process chart of POC algorithm

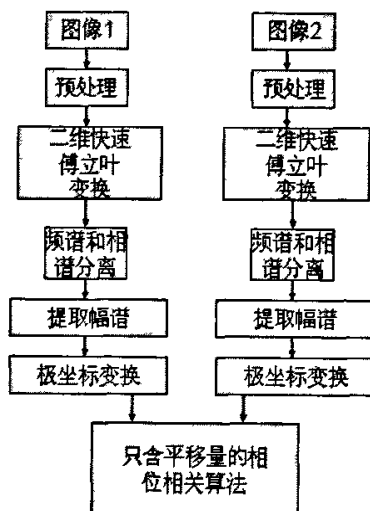


图 3-5 含有旋转量的相位相关算法流程

Fig.3-5 Flow process chart of RIPOC algorithm

3.4 基于 2 幂子图像的 FFT 对齐算法

在快速傅立叶算法 (FFT) 中, 维数 N 必须维可以分解为一些较小整数的乘积, 由文献[31]可知当 N 是 2 的幂 (即 $N=2^p$, 其中 p 是整数) 时, 效率最高,

实现起来也最简单。因此如果图像的长和宽都是 2 的幂，FFT 计算就可以达到最高效率。在图像对齐算法中，采用幂子图像也不会影响最终的对齐结果。

3.4.1 2 幂子图像

2 幂子图像的定义：图像 I 的宽为 W ，高为 H ，则 I 的 2 幂子图像 S 的宽 w 和高 h 符合以下条件：

$$\begin{cases} w = 2^m \leq W \leq 2^{m+1} \\ h = 2^n \leq H \leq 2^{n+1} \end{cases} \quad (3-38)$$

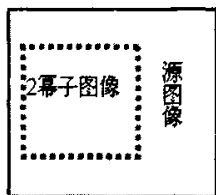


图 3-6 2 幂子图像

Fig.3-6 Two-power sub-image

在相邻的两幅图像对齐时，2 幂子图像取源于图像中间左边或右边的部分。前一幅图像 I_1 的 2 幂子图像 S_1 取源于图像的右边部分，后一幅图像 I_2 的 2 幂子图像 S_2 取源于图像的左边部分，如图 3-7 所示。

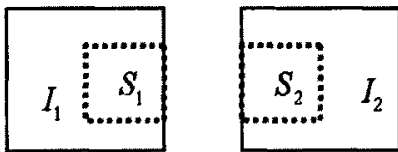


图 3-7 2 幂子图像对齐模型

Fig.3-7 Matching module of two-power sub-image

3.4.2 2 幂子图像的对齐

相邻的两幅图像对齐时，源图像的偏移可根据它们的 2 幂子图像偏移得到。在源图像中选取 2 幂子图像时，2 幂子图像在源图像中的位置也可计算出。前一幅源图像 I_1 的 2 幂子图像 S_1 的位置：

$$\begin{cases} x_l^p = W_l - w_l \\ y_l^p = (H_l - h_l)/2 \end{cases} \quad (3-39)$$

后一幅源图像 I_r 的 2 幂子图像 S_r 的位置：

$$\begin{cases} x_r^p = W_r - w_r \\ y_r^p = (H_r - h_r)/2 \end{cases} \quad (3-40)$$

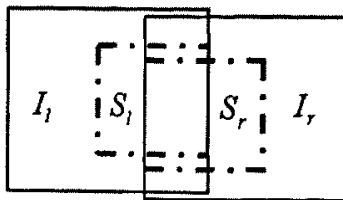


图 3-8 2 幂子图像的对齐

Fig3-8 Registration of two-power sub-image

假设由 FFT 计算出相邻图像的 2 幂子图像对齐时的位置偏移是 (t_x, t_y) ，则源图像间对齐偏移 (T_x, T_y) 是 $(t_x + x_l', t_y)$ 。

利用 2 幂子图像可以更快速计算出图像的偏移位置，从而对大量的采样图像序列对齐有着明显的速度优势，而且不会影响最终的源图像对齐偏移位置。

很明显，采用了 2 幂子图像后，相邻图像间的重叠率要求降低了，10%~25% 的重叠率就可很容易拼接了。

采用改进的基于 2 幂子图像的 FFT 对齐方法，不仅降低了对相邻图像间的重叠率的要求，而且加快了算法的计算速度。

3.5 本章小结

本章讲述了基于频域的图像匹配算法，在讨论离散傅立叶变换的基础上提出了相位相关度法，分析了相位相关度法原理及实现步骤。最后提出了一种基于 2 幂子图像的 FFT 对齐算法，该算法可提高图像拼接速度。总的来说基于频域的方法拼接效果不错，但对图像重叠率要求不低于 20%，否则无法找到匹配点。基于频域的方法需要傅立叶变换，直接影响了拼接速度。

第4章 基于像素灰度级的图像拼接算法

在基于像素灰度级的图像匹配算法中, 图像重叠范围的确定主要是基于两幅图像重叠区域对应像素的相似性, 即在 RGB 或 YUV 空间上具有基本相同的值。常用的基于像素灰度级相关的算法有^[11]: 比值法; 模板匹配法 (有的文献上也叫网格匹配法)。

4.1 比值匹配法

设有两幅待拼接源图像, 如果取前一幅图像中的一块作为模板, 在第二幅图像中搜索具有相同 (或相似) 值的对应块, 来确定两幅图像的重叠范围, 这种方法计算量大; 如果只利用一行或一行像素又会容易造成误匹配, 不能正确匹配。比值匹配法的思路是利用相隔一定间隔的两列上的部分像素的比值作为模板, 在第二幅图像中来搜索最佳匹配位置。比值法示意如图 4-1 所示。

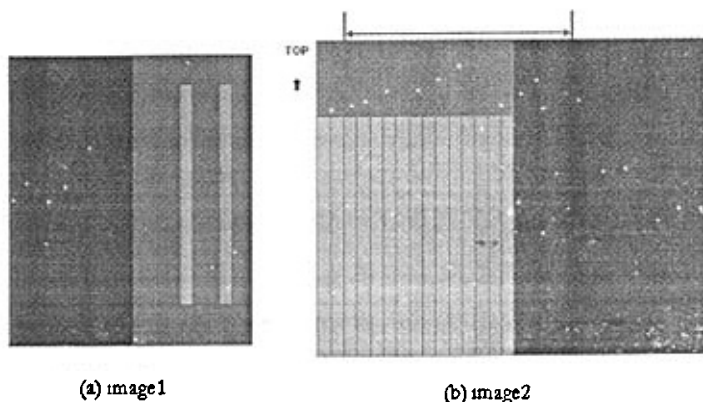


图 4-1 比值匹配法示意图

Fig.4-1 Experiment of ratio template matching

4.1.1 比值匹配法原理

比值法算法简单。设待拼接源图像为 *image1* 和 *image2*, 图像重叠区域在两幅源图像里所占比例在 20%~50%。设 $T[i][j]$ 表示图像 *T* 上第 *i* 行、第 *j* 列像素的灰度值, 则比值匹配模板的构造就是在 *image1* 重叠区域中找到间隔一定距离的两列上的部分像素, 将两列对应像素的比值的集合作为模板。比值法初始模板用 *M* 表示, $M[k]$ 表示模板 *M* 中第 *k* 个成员, *span* 表示这两列像素的水平间隔, 则模板的生成公式为:

$$M[k]=T[i][k]/T[i][k+span] \quad (4-1)$$

初始模板提取后,接着在 *image2* 上进行相似模板的搜索,相似模板用 *N* 表示,则对于 *image2* 上间距为 *span* 的两列,计算出由此构成的相似模板 *N*,并且逐一同初始模板 *M* 比较。可用 *M* 与 *N* 对应的像素点的差的绝对值和 *E* 作为评价函数来计算 *M* 与 *N* 之间的相似程度,如公式 (4-2):

$$E = \sum_{i=0}^{length} (|N[i] - M[i]|) \quad (4-2)$$

两个模板的相似程度越高 *E* 值就越小。当 *E* 取最小值时,认为已经找到正确匹配的位置,它是以第一列向量的第一个像素成员在待匹配图像中的坐标位置为准。

4.1.2 改进:加入自相关属性

基于比值模板的匹配是以前后图像重叠区域的两列像素的灰度比值作为匹配模板,这是一种相对关系,如同对于比值 3 可以认为是 6 除以 2 的值,也可以认为是 240 除以 80 的值。如果仅依靠这种相对关系来进行匹配,在亮度及色差变化较大时就容易出现误匹配。但是,如果在模板构造中引入了一些比值分子或分母自身的信息,比如在模板中添加分母的信息为“其值介于 77~85 之间的整数”,则容易确定与模板 3 匹配的是 240 除以 80^[21]。现面将详细说明如何构造这种具有自相关属性的模板。

设新构造出来的模板为 M_{new} , 它是一个二维向量,具体表达式如下:

$$M_{new} = \{M, TM\} \quad (4-3)$$

其中, *M* 由公式 (4-1) 构造,代表模板的相对关系; *TM* 则蕴涵着图 *S* 第 *i* 列的部分信息,构造方法如下:

$$TM[y] = T[i][y] / T[i][y+1] \quad (4-4)$$

其中, $0 \leq y \leq length$ 。

得到新的模板 M_{new} 后,可以类似地在待匹配图像上构造模板 $N_{new} \{N, TM\}$ 。然后利用下面的式子来判断这两个模板的相似程度:

$$E_{new} = (1 - \mu)E + \mu E_{TM} \quad (4-5)$$

上式中,因子 μ 为一个自相关因子,取值在 $[0, 1]$ 之间,经验值为 0.7。 E_{TM} 可根据下面的式子来计算:

$$E_{TM} = \sum_{i=0}^{length} (|TN[i] - TM[i]|) \quad (4-6)$$

当 E_{new} 取得最小值时,可以认为找到匹配模板。

在基于比值模板匹配原理的基础上,增加了模板自相关属性,比原算法更

能实现图像的准确拼接，并且对于所获取图像在亮度、色差等方面变化较大时也可以实现无误拼接，说明此算法对图像获取环境要求更为宽松。

4.1.3 比值匹配法的缺陷

图 4-2 为两幅待拼接的源图像，采用比值匹配法进行拼接，拼接后结果为图 4-3，从图 4-3 来看，拼接结果有误，本节就以此为例，来分析一下比值法的缺陷。

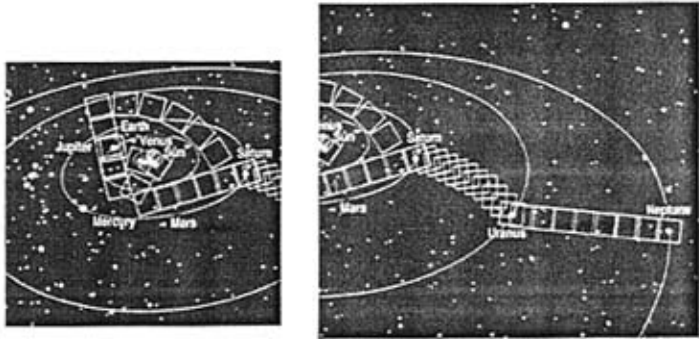


图 4-2 源图像

Fig.4-2 Source images

比值模板构造时，求两列对应像素的比值，涉及到除法，必然涉及到分母有效性的问题，也就是说分母不能为零。比值法最大的缺陷在于，如果模板 M 中第二列出现黑色像素点（灰度值为 0），比值为无穷大，用以表示模板 M 和 N 的评价函数 E 的值为无穷大，比值匹配法失效，找不到匹配点，以前的参考文献大都忽略了这个问题。分析图 4-2 可知，该图中黑色像素点很多，分布不均匀，比值模板中出现黑色像素点概率非常大，从而导致模板失效，无法实现正确拼接。

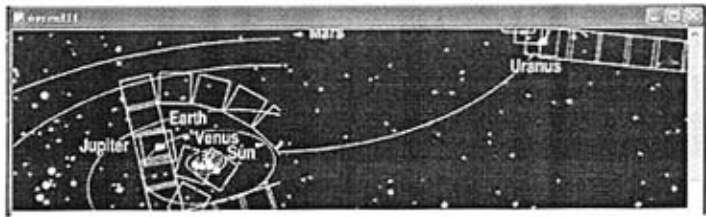


图 4-3 拼接结果

Fig.4-3 Result of two images matching

从图 4-3 可知，拼接结果不正确，这也就是本文提出的比值法的缺陷。对于比值法，可以改进算法，例如增加一个条件，必须保证作分母的像素点的灰度值不为零，这就添加了一个条件去搜索有无黑色像素点，使比值模板建立条

件更苛刻,无疑增加了算法的计算量,降低了拼接速度,对于实时性要求很高的地方增加判断分母是否为零显得冗余麻烦。

就比值法失效这个问题,可以换个思路来解决,能否采用另一种方法来弥补比值法无法拼接的问题,这也是一个不错的解决方法。采用比值法主要是因为该方法拼接速度快,实时性高,如果不去考虑速度问题,只考虑精度问题,就可以解决比值法无法实现拼接的问题。

4.2 模板匹配法

在机器识别事物的过程中,根据一致模式在另一幅图像中寻找相应的模式,这叫匹配。若在被搜索的图像中由待搜索模板,且同模板有一样的尺寸和方向,可以通过相关函数的计算来找到它以及在被搜索图中的坐标位置。设待拼接的两幅图像为 $image1, image2$ 。模板匹配就是在 $image1$ 中选取一个正方形网格作为初始模板 T , T 在 $image2$ 上遍历,通过一定的算法可以在图 $image2$ 中找到目标,确定其坐标位置。常用的两种测度:初始模板与 $image2$ 对应区域像素误差的平方和,平方和最小值对应点为匹配点;初始模板与 $image2$ 对应区域像素误差的绝对值之和,绝对值和最小值对应点为匹配点。实际上这两种测度都是求最小值,因为差值最小意味着两幅图像模板最相似,差值最小处坐标即是匹配点坐标,但模板匹配法计算量大。模板匹配示意图如图 4-4 所示:

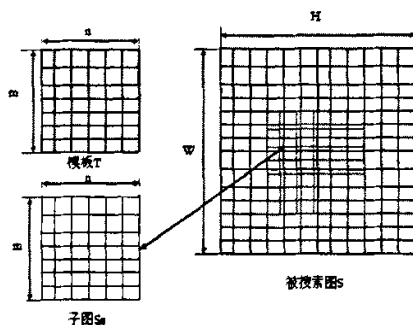


图 4-4 模板匹配法示意图

Fig.4-4 Experiment of template matching

4.2.1 模板匹配法原理

设模板 T 在被搜索图 S 上平移,模板覆盖下的那块搜索图叫做子图 S' , (i, j) 是这块子图的左下角像素点在 S 中的坐标,叫做参考点。设模板的高和宽都为 M , 被搜索图像的高和宽都为 N , 则 i, j 的取值范围是: $1 \leq i, j \leq N - M + 1$ 。比较 T 和 S' 的内容时,若两者一致,则 T 和 S' 之差为零。所以可以用下列测度来衡

量 T 和 $S^{i,j}$ 的相似度:

$$D(i, j) = \sum_{m=1}^M \sum_{n=1}^M [S^{i,j}(m, n) - T(m, n)]^2 \quad (4-7)$$

将上式展开, 则可得:

$$D(i, j) = \sum_{m=1}^M \sum_{n=1}^M [S^{i,j}(m, n)]^2 - 2 \sum_{m=1}^M \sum_{n=1}^M S^{i,j}(m, n) \times T(m, n) + \sum_{m=1}^M \sum_{n=1}^M [T(m, n)]^2 \quad (4-8)$$

式中, $\sum_{m=1}^M \sum_{n=1}^M [T(m, n)]^2$ ——表示模板的总能量, 与 (i, j) 无关;

$\sum_{m=1}^M \sum_{n=1}^M [S^{i,j}(m, n)]^2$ ——表示模板覆盖下那块子图像的能量;

$2 \sum_{m=1}^M \sum_{n=1}^M S^{i,j}(m, n) \times T(m, n)$ 表示子图像和模板的互相关, 随着 (i, j) 改变。当 T

和 $S^{i,j}$ 匹配时这一项的取值最大, 因此可以用下列相关函数作为相似性测度:

$$R(i, j) = \frac{\sum_{m=1}^M \sum_{n=1}^M S^{i,j}(m, n) \times T(m, n)}{\sum_{m=1}^M \sum_{n=1}^M [S^{i,j}(m, n)]^2} \quad (4-9)$$

或者归一化处理为:

$$R(i, j) = \frac{\sum_{m=1}^M \sum_{n=1}^M S^{i,j}(m, n) \times T(m, n)}{\sqrt{\sum_{m=1}^M \sum_{n=1}^M [S^{i,j}(m, n)]^2} \sqrt{\sum_{m=1}^M \sum_{n=1}^M [T(m, n)]^2}} \quad (4-10)$$

在实际应用中, 常用的是归一化后的相似性评价函数。根据施瓦兹不等式:

$$[\sum_{i=1}^n a_i b_i]^2 \leq [\sum_{i=1}^n a_i^2] [\sum_{i=1}^n b_i^2] \quad (4-11)$$

可知上式中 $0 \leq R(i, j) \leq 1$, 并且仅在比值 $S^{i,j}(m, n)/T(m, n)$ 为常数时, $R(i, j)$ 取极大值 1。

当模板和子图完全一样时, 相关系数 $R(i, j) = 1$ 。在被搜索图 S 中完成全部搜索后, 超出 $R(i, j)$ 的最大值 $R_{max}(i_s, j_s)$, 其对应的子图 S_{i_s, j_s} 即为匹配目标。

用归一化互相关求匹配的计算量很大, 速度很慢, 因为模板要在 $(M-i+1) \times (N-j+1)$ 个参考位置上做相关计算, 其中, 除最佳匹配点外, 其余做的都是无效运算, 所以有必要对其进行改进, 以提高运算速度。

上面用的测度函数为差值的平方和, 当然可以求差值的绝对值, 即求 T 和 $S^{i,j}$ 的误差, 其公式如下:

$$E(i, j) = \sum_{m=1}^M \sum_{n=1}^M |S^{i,j}(m, n) - T(m, n)| \quad (4-12)$$

$E(i, j)$ 为最小值处即为匹配目标。为提高计算速度, 取一个误差阈值 E_0 , 当 $E(i, j) > E_0$ 时就停止该点的计算, 继续下一点计算。

结果表明: 被搜索图越大, 匹配速度越慢; 模板越小, 匹配速度越快。误差法速度较快, 阈值的大小对匹配速度影响大, 和模板的尺寸有关。

4.2.2 改进的模板匹配算法

在误差算法的基础上设计了二次匹配误差算法:

第一次匹配是粗略匹配。取模板的隔行隔列数据, 即四分之一的模板数据, 在被搜索图上进行隔行隔列扫描匹配, 即在原图的四分之一范围内匹配。由于数据量大幅度减少, 匹配速度显著提高。

为了合理的给出一个误差阈值 E_0 , 采用的一个确定误差阈 E_0 的准则:

$$E_0 = e_0 \times (m+1)/2 \times (n+1)/2 \quad (4-13)$$

式中: e_0 为各点平均的最大误差, 一般取 40~50 即可, m, n 为模板的长和宽。

第二次匹配是精确匹配。在第一次误差最小点 (i_{min}, j_{min}) 的邻域内, 即在对角点为 $(i_{min}-1, j_{min}-1)$, $(i_{min}+1, j_{min}+1)$ 的矩形内, 进行搜索匹配, 得到最后结果。

表 4-1 相关法、误差法和二次误差法比较

Table4-1 Coparison experiment

被搜索图尺寸	模板名称	模板大小	卷积法 时间 (s)	误差法时 间 (s)	二次匹配法 时间 (s)
256*256	corner	16*16	0.94	0.55	0.06
	eye	13*11	0.55	0.38	0.03
	hell	15*22	1.16	0.44	0.05
640*480	finger	13*17	4.12	2.30	0.22
	nose	11*17	3.41	2.20	0.16

表 4-1 是相关法、误差法、二次匹配误差法这三种模板匹配算法对两幅图像进行模板匹配的比较结果, 从表中数据分析, 很明显二次匹配误差法最快。

4.3 比值法和模板匹配法比较

基于像素灰度级相关算法, 不同于基于频域的方法。基于频域的相位相关度法需要变换, 而基于像素灰度级的算法仅仅利用像素的相似性, 不需要去做

变换,所以基于像素灰度级相关的算法在实际中应用广泛。上两节分别讨论了比值法和模板匹配法。比值法仅仅取用了间隔一定距离的两列中的部分像素,思路简单,对于图像质量很好的待拼接图像,对图像进行正确拼接,得到满意的图像;模板匹配法,构造模板也不是很复杂,根据前面提到的,模板建立好后,评价函数有两个:求差的绝对值之和(改进的方法是二次匹配误差法);求极大相关系数。在后面我们一般称对应的模板匹配算法为最小差值匹配法和极大相关系数匹配法,两者模板选取一样,唯一差别在于计算匹配点坐标算法不一样。

模板匹配法有重要应用,以军事应用为例,图像匹配制导是一种非常先进的制导方式,通过遥感图像把导弹自动引向目标,通常用于远程导弹的末端制导。这种制导方式事先在目标附近选择地貌特征明显的地区作为景象匹配区,将侦察获得的匹配区的图像以数字方式预存于导弹的计算机内,当导弹飞经景象匹配区时导弹上的摄像机拍摄的景物图像经过数字化处理形成遥感实时图像与预存的数字景象地图进行比较来确定导弹相对于目标的位置,纠正导弹的飞行方向。因此图像匹配制导精度很高。

模板匹配算法复杂,运算量大,如果图像质量很好,比值法可以满足精确匹配,就没必要采用模板匹配法,但考虑到前面提到的比值法的缺陷,对于采用比值法无法实现的图像匹配,可以考虑采用模板匹配法来弥补比值法的不足。本设计的思路也是基于此,在后面的章节里重点讨论了拼接的具体实现,采用了一种基于比值法和模板匹配法相结合的方法,提高了图像拼接的成功率。

4.4 本章小结

本章详细介绍了基于像素灰度级的图像匹配算法,重点讨论了比值法和模板匹配法,并对这两种方法加以比较,指出比值法拼接速度快,但模板匹配法拼接效果好。

第5章 图像拼接具体实现

5.1 数字图像处理基础

我们接触到的现实世界中的图像多为模拟图像，但是计算机只能处理数字信息，因此，必须将其数字化，转换成适合计算机表示的形式，才能由计算机进行处理。为了便于信息交换和方便使用，数字图像必须以一定的格式存储，如常见的 BMP、JPEG、GIF 等图像文件格式。从颜色来看，数字图像有线画稿、灰度图像、彩色图像、真彩色图像等种类。要表示丰富多彩的颜色信息，就需要用到各种颜色模型，如 RGB、HSL、CMY 等。在实现图像拼接之前，有必要了解和掌握这些数字图像处理的基础知识。本节将介绍图像数字化技术、数字图像的存储格式、图像的种类、颜色模型等。

5.1.1 图像数字化技术

我们日常生活中见到的图像一般是连续形式的模拟图像，所以数字图像处理的一个先决条件就是将连续图像离散化，转换为数字图像。

图像的数字化包括采样和量化两个过程。

设连续图像 $f(x, y)$ 经数字化后，可以用一个离散量组成的矩阵 $g(i, j)$ （即二维数组）来表示。

$$g(i, j) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,n-1) \\ f(1,0) & f(1,1) & \dots & f(1,n-1) \\ \dots & \dots & \dots & \dots \\ f(m-1,0) & f(m-1,1) & \dots & f(m-1,n-1) \end{bmatrix} \quad (5-1)$$

矩阵中的每一个元素称为像素。而 $g(i, j)$ 代表 (i, j) 点的灰度值，即亮度值。

图像在空间上的离散化称为采样。也就是用空间上部分点的灰度值代表图像，这些点称为采样点。由于图像是一种二维分布的信息，为了对它进行采样操作，需要先将二维信号变为一维信号，再对一维信号完成采样。具体做法是，先沿垂直方向按一定间隔从上到下顺序地沿水平方向直线扫描，取出各水平线上灰度值的一维扫描。而后再对一维扫描线信号按一定间隔采样得到离散信号，即先垂直方向采样，再沿水平方向采样这两个步骤完成采样操作。

模拟图像经过采样后，在时间和空间上离散化为像素。但采样得到的像素值仍是连续量。把采样后所得的各像素的灰度值从模拟量到离散量的转换称为图像灰度的量化。

5.1.2 数字图像类型

为了方便地处理数字图像，根据数字图像的特性将其分成不同的类型。静态图像可分为矢量(Vector)图和位图(Bitmap)，位图也称为栅格图像。

矢量图是用一系列绘图指令来表示一幅图，如 AutoCAD 中的绘图语句。这种方法的本质是用数学公式描述一幅图像。图像中每一个形状都是一个完整的公式，称为一个对象。公式化表示的矢量图具有两个优点：一是它的文件数据量很小；二是图像质量与分辨率无关。这意味着无论将图像放大或缩小了多少次，图像总是以显示设备允许的最大清晰度显示。但矢量图有一个明显的缺点，就是不易制作色调丰富或色彩变化太多大图像，而且绘出来的图像不是很逼真。位图是通过许多像素点表示一幅图像，每个像素具有颜色属性和位置属性。位图主要分为四种：线画稿(LineArt)、灰度图像(GrayScale)、索引颜色图像(IndexColor)和真彩色图像(True Color)。

(1)线画稿 线画稿只有黑白两种颜色，也成二值图像；

(2)灰度图像 在灰度图像中，像素灰度级用 8 bit 表示，所以每个像素都是介于黑色和白色之间的 256($2^8=256$)种灰度中的一种。灰度图像只有灰度颜色而没有彩色。我们通常所说的黑白照片，其实包含了黑白之间的所有灰度色调。从技术上来说，就是具有从黑到白的 256 种灰度色域的单色图像。

(3)索引图像 打开该图像文件类型时，构成该图像具体颜色的索引值就被读入程序里，然后根据索引值找到最终的颜色。

(4)真彩色图像 真彩色图像将像素的色彩能力推向了顶峰。“真彩色”是 RGB 颜色的另一种流行的叫法。从技术角度考虑，真彩色是指写到磁盘上的图像类型，而 RGB 颜色是指显示器的显示模式。在真彩色图像中，每一个像素由红、绿和蓝三个字节组成，每个字节为 8bit，表示 0 到 255 之间的不同的亮度值，这三个字节组合，可以产生 1670 万种不同的颜色。

5.1.3 图像文件格式

数字图像有多种存储格式，每种格式一般由不同的开发商支持。要进行图像处理，必须了解图像文件的格式，即图像文件的数据构成。每一种图像文件均有一个文件头，在文件头之后才是图像数据。文件头的内容由制作该图像文件的公司决定，一般包括文件类型、文件制作者、制作时间、版本号、文件大小等内容。数字图像处理大多采用 BMP 图像文件格式，BMP(BitMap Picture)文件格式是 Windows 系统交换图形、图像数据的一种标准格式。BMP 图像的数据由四部分组成：位图文件头、位图信息头、调色板和位图数据。下面讲述一

下调色板和位图数据，其他可参考相关文献。

①调色板(Palette)是对哪些需要调色板的位图文件而言的。真彩色图像是不需要调色板的，位图信息头后直接是位图数据。调色板实际上是个数组，共有 `biClrUsed` 个元素。数组中每个元素的类型是一个 `RGBQUAD` 结构，占 4 个字节，其定义如下：

```
typedef structure tagRGBQUAD{
    BYTE    rgbBlue;//该颜色的蓝色分量
    BYTE    rgbGreen;//该颜色的绿色分量
    BYTE    rgbRed;//该颜色的红色分量
    BYTE    rgbReserved;//保留值
}RGBQUAD;
```

②位图数据就是实际的图像数据。对于用到调色板的位图，图像数据就是该像素颜色在调色板中的索引值，对于真彩色图像，图像数据就是实际的 R、G、B 值。下面就 2 色、16 色、256 色和真彩色位图分别介绍。

对于 2 色位图，用 1 位就可以表示该像素的颜色（一般 0 表示黑，1 表示白），所以一个字节可以表示 8 个像素。

对于 16 色位图，用 4 位可以表示一个像素的颜色，所以一个字节可以表示 2 个像素。

对于 256 色位图，一个字节刚好可以表示一个像素。

但必须注意：每一行的字节数必须是 4 的整数倍，如果不是，则需要补齐。BMP 文件的数据存放是从下到上，从左到右的。也就是说，从文件中最先读到的是图像最下面一行的左边第一个像素，然后是左边第二个像素，接下来是倒数第二行左边第一个像素，左边第二个像素。依次类推，最后得到的是最上面一行的最右边的一个像素。

5.1.4 颜色模型

为了科学地定量描述和使用颜色，人们提出了各种颜色模型。目前常用的颜色模型按用途可分为两类，一类面向诸如视频监视器、彩色摄像机或打印机之类的硬件设备。另一类面向以彩色处理为目的的应用，如动画中的彩色图形。面向硬件设备的最常用彩色模型是 RGB 模型，而面向彩色处理的最常用模型是 HSI 模型。另外，在印刷工业上和电视信号传输中，经常使用 CMYK 和 YUV 色彩系统。

1) RGB 模型：

RGB 模型用三维空间中的一个点来表示一种颜色，如图所示。每个点有三个分量，分别代表该点颜色的红、绿、蓝亮度值，亮度值限定在 $[0, 1]$ 。

在 RGB 模型立方体中, 原点所对应的颜色为黑色, 它的三个分量值都为零。距离原点最远的顶点对应的颜色为白色, 它的三个分量值都为 1。从黑到白的灰度值分布在这两个点的连线上, 该线称为灰色线。立方体内其余各点对应不同的颜色。彩色立方体中有三个角对应于三基色—红、绿、蓝。剩下的三个角对应于三基色的补色—黄色、青色、品红。

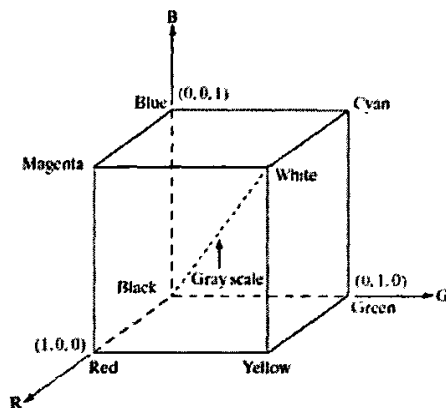


图 5-1 RGB 模型

Fig.5-1 RGB module

2) HIS 模型:

HIS 模型是 Munseu 提出的, 它反映了人的视觉系统观察彩色的方式, 在艺术上经常使用 HIS 模型。HIS 模型中, H 表示色调(Hue), S 表示饱和度(Saturation), I 表示亮度(Intensity, 对应成像亮度和图像灰度)。这个模型的建立基于两个重要的事实: ①I 分量与图像的彩色信息无关; ②H 和 S 分量与人感受颜色的方式是紧密相联的。这些特点使得 HIS 模型非常适合借助人的视觉系统来感知彩色特性的图像处理算法。

YUV 彩色电视信号传输时, 将 R、G、B 改组成亮度信号和色度信号。PAL 制式将 R、G、B 三色信号改组成 Y、U、V 信号, 其中 Y 信号表示亮度, U、V 信号是色差信号。

RGB 与 YUV 之间的对应关系如下:

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.148 & -0.289 & -0.437 \\ 0.615 & 0.515 & -0.100 \end{bmatrix} \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (5-2)$$

在本设计中, 用到了公式, Y 代表亮度信息, 由 RGB 信息可得亮度信息为: $Y=0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B$, 这样可以由 R、G、B 三色得到一个评价量, 即亮度信息。

5.1.5 BMP 位图文件的存储格式

BMP(Bitmap 的缩写)图像是指文件名后缀为 BMP 的位图图像。位图图像在计算机中使用很广泛,例如在 Windows 中,记事本、写字板中的文字就是用位图图像表示出来的。许多以其他格式存储的图像,就是在位图图像的基础上,进行优化处理后得到的,例如 JPEG 图像等。

在数字图像处理中,许多算法主要是针对灰度位图和 24 位真彩色位图设计的。因此,很有必要介绍以下位图文件的这两种存储格式。

1) 24 位真彩色位图存储格式:

24 位真彩色位图的二进制数据包括:位图文件头,位图信息头和位图阵列三部分。位图文件头用来记录标志文件大小的一些信息,在文件中占 14 个字节。位图信息头记录和位图相关的一些信息,在文件中占 40 个字节。位图阵列,即像素表示部分,每个像素点由 3 个字节的数据组成,按照从左到右的顺序,分别表示蓝色、绿色、红色。

2) 灰度图像存储格式:

灰度图像存储格式与 24 位真彩色图像存储格式基本一样。最大的差别是,灰度位图比 24 位真彩色位图增加了颜色索引表。同时灰度位图用一个字节来表示一个像素。这样,灰度位图的像素阵列小了三分之二。

颜色索引表定义如下:

在实际的编程中,读取数据的方式发生了变化,除了要读取文件头,文件信息头,还要设置变量读取颜色索引表。除此以外,灰度图像的编程读取显示方式与 24 位真彩色位图的方式完全相同,在此就不再叙述。

5.2 VisualC++数字图像处理基础

VisualC++最大的特色是给面向对象技术以强大支持,通过Microsoft基础类库MFC(Microsoft Function Class)把大部分与用户界面设计有关的 Windows API函数封装起来,提供程序开发人员使用,从而提高了代码的可重用性。

5.2.1 设备无关位图 (DIB)

这里首先引入一个概念就是设备无关位图 (DIB)。在 Windows 3.0 中引入了与设备无关的位图 (DIB),提供了适用于交换的图像格式。DIB 内的位图几乎都没有被压缩。如果想在程序中操作位图,这实际上是一个优点。DIB 不像 GIF 和 JPEG 格式,Windows API 直接支持 DIB。如果在内存中有 DIB,就可以提

供指向该 DIB 的指针作为某些函数的参数，来显示 DIB 或把 DIB 转化为 DDB。BMP 文件包含了一个 DIB，至于 BMP 文件的格式在前面详细讨论过，因此这里不再展开。值得强调的是，对其结构的了解是进行 BMP 编程的基础。

在 Windows 的 GDI 中，有一些用于访问 DIB 的重要函数，本节列出了这些在操作 DIB 图像时比较重要的函数和详细的参数信息。

① SetDIBitsToDevice() 和 StretchDIBits()

前者可以直接在显示器或者打印机上显示 DIB。在显示的时候不进行压缩处理。

② CreateDIBitmap()

该函数用来从 DIB 创建 GDI 位图。

③ CreateDIBSection()

该函数主要用于创建一个应用可以直接操作的设备无关位图 (DIB)。其第 4 个参数用来返回创建的 DIB 的位图位的指针。

5.2.2 构造 CDib 类

在对 DIB 位图进行操作时，通常有如下的一些任务要完成：装载位图、获取位图文件头信息、获取位图信息、获取颜色表信息、获取位图数据信息、获取位图尺寸、获取颜色总数、获取颜色位数、创建位图调色板、绘制位图等。微软并没有提供对 DIB 操作的类，为了操作的方便，这里需要自己动手编写一个用于操作的类—CDib。

构造一个 CDib 类主要包括以下内容：

1. CDib 类声明

本小节构建了 CDib 类，它对操作 DIB 的部分 GDI 函数进行了封装，它是后续图像处理的基石。

2. CDib 类实现

本小节给出了 CDib 类的完整的定义部分，主要是对上节中声明的类的成员函数的定义。其中包括一个用于判断位图标志的宏 DIB_MARKER，这在实现文件的起始部分中。

DrawDib()：该函数主要用来绘制 DIB 对象；

ConstructPalette(HGLOBAL hDIB, CPalette* pPal)：该函数按照 DIB 创建一个逻辑调色板；

GetBits(LPSTR lpdib)：计算 DIB 像素的起始位置，并返回指向它的指针；

GetWidth(LPSTR lpdib)：该函数返回 DIB 中图像的宽度；

GetHeight(LPSTR lpdib)：该函数返回 DIB 中图像的高度；

GetPalSize(LPSTR lpdib): 该函数返回 DIB 中调色板的大小;
GetColorNum(LPSTR lpdib): 该函数返回 DIB 中调色板的颜色的种数;
GetBitCount(LPSTR lpdib): 该函数返回 DIBBitCount;
CopyObject (HGLOBAL hGlob): 该函数复制指定的内存区域;
SaveFile(HGLOBAL hDib, CFile& file): 将指定的 DIB 对象保存到指定的 CFile 中;
LoadFile(CFile& file): 将指定的文件中的 DIB 对象读到指定的内存区域中;
GetReqByteWidth(int bits): 获取需要的行字节数, 应为 4 的倍数;
GetRectWidth(LPCRECT lpRect): 获取矩形区域的宽度;
GetRectHeight(LPCRECT lpRect): 获取矩形区域的高度;
InitMembers() : 初始化类的成员变量;
ClearMemory() : 复位类的成员变量;

5.3 灰度图像拼接具体实现

前面两节讲述了一些数字图像处理的基础以及 VisualC++应用于数字图像处理的知识, 这些知识是下面图像拼接具体实现的基础, 本设计需要建立一个多文档, 用于打开多幅图像, 在实际操作中, 增加了一个登录对话框设置, 运行软件前弹出, 需要输入用户名和密码, 确定后才能进入软件界面, BMP 文件是没有经过压缩处理的图像文件, 非常适合于图像处理, 本设计就是对 BMP 文件进行图像拼接的, 所以在拼接中需要判断打开的图像文件是不是 BMP 文件, 由于灰度图像和彩色图像的存储是不一样的, 所以, 还需要判断图像文件色彩是否一致, 若不一致, 还需要重新输入图像。本设计创新点在于添加了拼接初始参数设置对话框, 可用于动态修改拼接参数, 这样有利于图像拼接成功。灰度图像拼接采用比值法和模板匹配法实现, 默认为比值法, 但比值法拼接不成功时候可采用模板匹配法, 同前面讲的, 模板匹配法有两个: 最小差值法和极大相关系数法。

这里还有一个条件必须满足才能保证图像拼接成功, 就是获取的图像在同一个平面内, 也就是说获取图像时, 相机保持垂直距离不变, 只是水平移动, 这样采集的图像不需要做坐标变换, 可直接进行拼接。

5.3.1 多文档应用程序工程的建立

运行 VC++6.0 软件, 在 MFC AppWizard-Setup1 中选择 Multiple documents,

然后点击 Next, 在 MFC AppWizard2of6~MFC AppWizard 5of6 中选择默认设置。在 MFC AppWizard 6of6 中, 选择列表中的 CImageProcessView 中的 Base class 的 CscrollView, 然后点击 Finish 按钮。在随后的对话框中点击 OK 按钮, 完成多文档应用程序工程的建立。

所建工程共包括六个类:

CaboutDlg 类是关于对话框类, 用于显示程序基本信息和版权信息;

CchildFrame 类是子框架类, 用于管理程序子框架;

CImageProcessView 类是应用程序类, 是整个程序的入口;

CImageProcess DOC 类是是文档类, 管理程序文档;

CImageProcdssView 类是视图类, 用于文档视图的显示;

CmainFrame 类是主框架类, 显示程序运行时的主界面。

5.3.2 导入 CDib 类

首先将 Dib.cpp 和 Dib.h 两个文件拷贝到工程所在目录下。选择菜单 Project-Add To Project-Files..., 进入 Insert Files into Project 对话框, 选种上述两个文件, 点击 OK 按钮, 这样 CDib 类就导入到本工程中。

虽然导入了 CDib 类, 但还不能应用它, 需要对文档、视图类作进一步的修改, 才能实现位图的读取、显示和保存。下面分别讲述需要修改的类以及相应修改措施。

1. CImageProcessDoc 类的修改

在 DImageprocessDoc.h 加入#include "Dib.h"包含 Cdib 类, 加入 extern m_dib 语句使变量 m_dib 在整个文件中都可以被引用。

添加成员变量和成员函数: 在 ClassView 中的 CImageProcessDoc 上点击右键选择 Add Member Variable, 进入 Add Member Variable 对话框, 输入变量类型、变量名称, 选择变量属性, 点击 OK, 完成类成员变量的添加; 在 ClassView 中的 CImageProcessDoc 上点击右键选择 Add Member Function, 进入 Add Member Function 对话框, 输入函数类型、函数名称, 选择函数属性, 点击 OK, 完成类成员函数的添加。

5.3.3 为主框架添加菜单及消息处理函数

在 Workspace 中选择 ResourceView, 点击 Menu 前面的加号, 出现 IDR_DIMAGETYPE, 双击 IDR_DIMAGETYPE, 出现菜单设置界面。在最后面的虚线框上点击右键, 选择 Properties, 进入 Menu Item Properties 对话框, 按下图所示设置菜单属性: “灰度图像拼接”下的菜单属性如下图 5-2 所示。其余

菜单的添加同上。

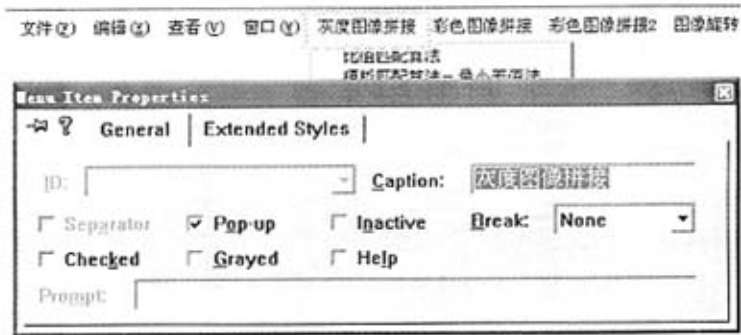


图 5-2 添加灰度图像拼接菜单

Fig.5-2 Add-on menu for gray image mosaicing

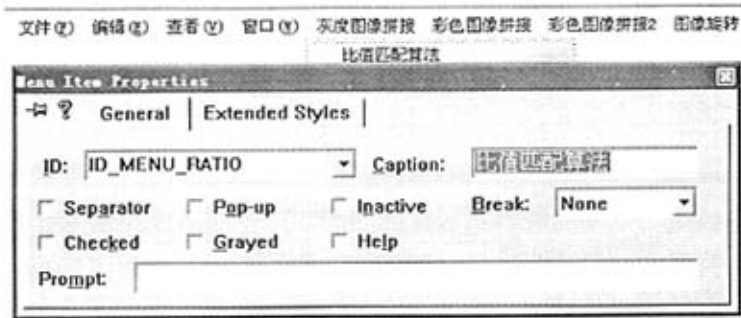


图 5-3 添加比值法菜单

Fig.5-3 Add-on menu for ratio method

本设计中，添加了灰度图像拼接菜单和后面就要提到的彩色图像拼接菜单，灰度图像拼接菜单下面又有三个菜单：比值匹配法、最小差值法和极大相关系数法三个菜单。

在菜单项“比值匹配算法”上点击右键，选择 ClassWizard，弹出 MFC ClassWizard 对话框，在 Class name 中选择 CDImageProcessView，在 Object Ids 中选择 ID_MENU_RATIO，在 Messages 中分别选择 COMMAND 和 UPDATE_COMMAND_UI，点击 Add Function 按钮后点击 OK 按钮完成添加消息处理函数，如图 5-4 所示。同样为其它菜单添加消息处理函数。



图 5-4 添加消息处理函数

Fig.5-4 Adding message disposing function

5.3.4 添加登录对话框

在 Workspace 中选择 ResourceView, 在 Dialog 上点击右键, 选择 Insert Dialog。在对话框上点击右键, 选择 Properties, 设置如下图:

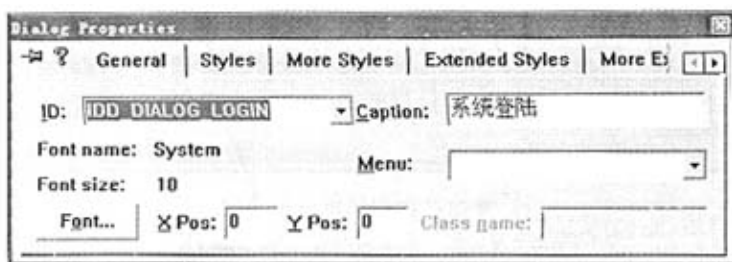


图 5-5 添加登录对话框

Fig.5-5 Adding accession frame

要在系统主框架启动前弹出系统登录对话框，先在 DImageProcess.cpp 文件的上部添加#include "LoginDlg.h"，然后在 CDImageProcessApp 类的初始化函数 InitInstance() 中添加代码，如下语句：

```
//进入登录对话框
```

```
CLoginDlg dlg;           //定义对话框类
```

下一句语句调用显示对话框的函数显示登录对话框，判断是否点击了确定按钮，如果点击了确定按钮程序继续往下执行，否则程序退出。

```
if(IDOK != dlg.DoModal())
{
    return FALSE;
}
```

以上语句在程序入口函数 `CDImageProcessApp::InitInstance()` 中，在执

行主框架程序前调用上述语句显示登录对话框。

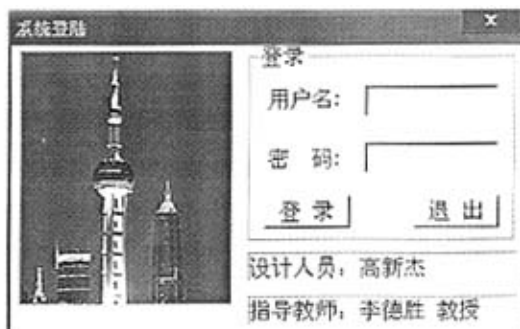


图 5-6 登录对话框

Fig.5-6 accession frame

5.3.5 拼接前预处理

在图像拼接前，必须做到三点：判断图像是否是 BMP 图像、判断图像大小是否大于 100*100 像素以及图像颜色是否一致，这就是拼接前预处理。当图像不是 BMP 位图时，弹出菜单说明图像不是 BMP 图像，需要重新打开 BMP 图像，如果打开的 BMP 图像大小低于 100*100 像素，同样弹出菜单，需要重新输入更大的图像。如果图像是 BMP 图像且图像大小满足要求，这时候还得判断图像色彩是否一致，比如是否都为灰度图像或彩色图像。下面是判断图像是否为位图以及图像大小是否满足要求的程序。

CDImageProcessDoc::OnOpenDocument(LPCTSTR lpszPathName)中用于图像拼接预处理的语句如下：

```
if(bmfHeader.bfType != ((WORD)('M' << 8) | 'B'))
{
    AfxMessageBox("不是位图文件!");
    return FALSE;
}

if(file.Read((LPSTR)&bmiHeader, sizeof(bmiHeader)) !=
sizeof(bmiHeader))
{
    return FALSE;
}

if((bmiHeader.biWidth < 100) || (bmiHeader.biHeight < 100))
{
    AfxMessageBox("图像太小! \n必须为 100 * 100 像素或更大。");
```

```

return FALSE;
}

```

5.3.5 初始模板设置

对于比值法和模板匹配法，都涉及到初始模板是否存在的问题。对于比值法，比值模板中两列的水平间隔应不小于 20 个像素单位，间隔太小，灰度变化不明显，易造成误匹配，如果间隔太大，第二列可能越界。比值法最大的缺陷在于，如果模板 M 中第二列出现黑色像素点（灰度值为 0），比值为无穷大，用以表示模板 M 和 N 的评价函数 E 的值为无穷大，比值法失效，找不到匹配点，以前的参考文献都忽略了这个问题。对于模板匹配法，只要保证初始模板存在于图像重叠区域就可以。为了保证初始模板存在，我们在两种方法拼接前，添加了一个模板参数设置对话框。下面以比值模板参数设置对话框为例进行说明，比值法初始模板参看图 5-7。

影响比值法模板选取的三个因素：①：模板 M 第一列第一个像素坐标 (Ratio_M_x , Ratio_M_y)；②：模板 M 中两列间距 Ratio_M_span ；③：模板 M 长度 Ratio_M_length 。默认参数设置为： $\text{Ratio_M_x} = \text{width1} * 5/7$ (width1 为 image1 的宽度)； $\text{Ratio_M_y} = \text{height1}/2$ (height1 为 image1 的高度)； $\text{Ratio_M_span} = 20$ ； $\text{Ratio_M_length} = \text{height1}/5$ ，使 M 位于 image1 右中间。比值算法初始模板设置范围为： $\text{Ratio_M_x} + \text{Ratio_M_span} < \text{width1}$, $\text{Ratio_M_y} + \text{Ratio_M_length} < \text{height1}$ 。模板 M 设置参照图 5-7。模板动态设置优点：直观；拼接效果不好时，可修改模板，重新找最佳匹配点；模板大小不固定，可适用于大小不同图像。 N 搜索范围为 image2 的左半区域，减少了搜索时间。

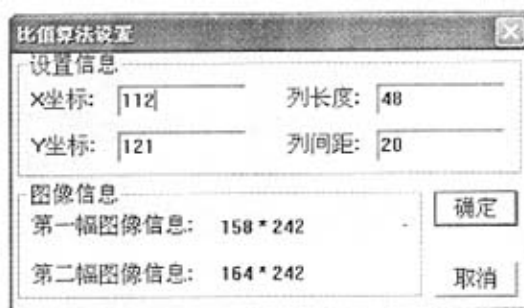


图 5-7 比值法设置模板

Fig.5-7 The set-up frame for ratio method

将 image1 和 image2 拼接为一幅完整图像 image3 。在 image2 中找到匹配点 (Ratio_N_x , Ratio_N_y) 后，接下来根据 (Ratio_M_x , Ratio_M_y), (Ratio_N_x , Ratio_N_y) 以及 image1 、 image2 大小来确定 image3 大小。设 $d1$ 代表

$image1_d1$, $image2_d1$ 之间的小者, $d2$ 代表 $image1_d2$ 和 $image2_d2$ 之间的小者, 这里 $image1_d2$ 等于 $Ratio_M_y$, $image2_d2$ 等于 $Ratio_N_y$, $image1_d1$ 等于 $height1-length-Ratio_M_y$, $image2_d1$ 等于 $height2-length-Ratio_N_y$ 。 $d1$, $d2$ 都取最小者, 多余部分截掉, $image2$ 匹配点左边图像截掉, 拼接成一幅形状为矩形的图像 $image3$, 比值拼接示意图如图5-8所示。

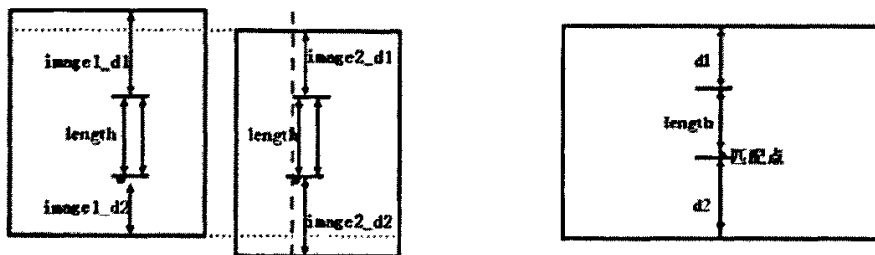


图 5-8 比值法

Fig.5-8 Ratio method

5.3.6 比值法匹配点搜索

首先搜索匹配点：在第一幅图像中右部选择一起始点 $(x1, y1)$ ，选择一定间隔的两列像素，在第二幅图像中从左下角开始向上向右查找，选择最佳匹配点 $(x2, y2)$ ，确定与之匹配的两列像素在第二幅图像中的起始点。接着确定第三幅图像的大小：设点 $(x1, y1)$ 到第一幅图像的上边缘的距离为 $h1$, $(x2, y2)$ 到第二幅图像的上边缘的距离为 $h2$ ，取 $height1$ 为 $h1$ 、 $h2$ 两者的最小值，设点 $(x1, y1)$ 到第一幅图像的下边缘的距离为 $h3$, $(x2, y2)$ 到第二幅图像的下边缘的距离为 $h4$ ，取 $height2$ 为 $h3$ 、 $h4$ 两者的最小值。则第三幅图像的高度 $height = height1 + height2$ ；设 $(x1, y1)$ 到第一幅图像的左边缘的距离为 $w1$, $(x2, y2)$ 到第二幅图像的右边缘的距离为 $w2$ ，则第三幅图像的宽度 $width = w1 + w2$ 。最后生成第三幅图像：取第一幅图像中点 $(x1, y1)$ 左边距离 $w1$ ，到上边距离 $height1$ ，到下边距离 $height2$ 的部分复制到第三幅图像的左半部分；取第二幅图像中点 $(x2, y2)$ 右边距离 $w2$ ，到上边距离 $height1$ ，到下边距离 $height2$ 的部分复制到第三幅图像的右半部分；拼接完成，拼接示意图如上面图 5-8。

比值算法有个缺陷，就是当某个点的像素值为 0 时，另外一个像素点与该点像素值的比值为无穷大，无法找到匹配点。为了解决这个问题，在查找前弹出一个对话框，用于设置拼接参数，可以避免因比值无穷大而找不到匹配的缺陷。具体过程如下图：前两行代码设置默认 $(x1, y1)$ 的位置，从第 4 行到第 18 行弹出参数设置对话框，并将默认参数传递给对话框，最后如果点击了确定按钮，对话框里的参数值就返回，程序按照对话框给

定的参数进行查找。

5.3.7 拼接流程

根据前面讲的,本设计采用 VisualC++6.0 软件编程,进行图像拼接时,比值法和模板法可选,考虑到比值法运行速度快,所以默认为比值法。灰度图像拼接流程如图 5-9 所示。

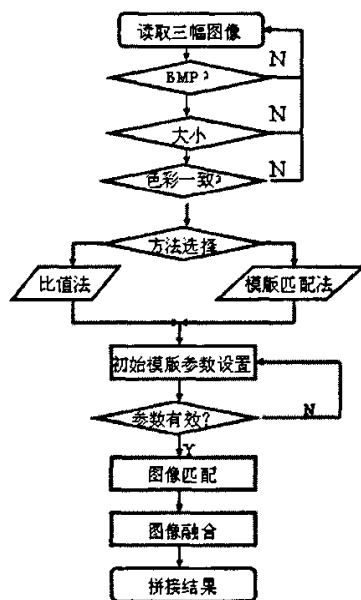


图 5-9 灰度图像拼接流程图

Fig.5-9 The flow process chart of gray image mosaic

5.4 彩色图像拼接

5.4.1 真彩色数字位图存储

一个完整的真彩色 DIB(Device-Independent Bitmap 设备无关位图)一般包括以下几个部分:位图文件头、位图信息头、位图阵列信息。位图文件头指明文件类型,文件大小即从文件头到实际的位图数据的偏移字节数,位图信息头指明图像的宽度以及高度等信息。位图的阵列信息就是实际的图像数据的 R、G、B 值,长度视图像的大小而定。一般来说, BMP 文件的数据是从下到上、从左到右的,即假设一幅位图共有 m 行 n 列,在位图阵列信息中实际存放的格式是首先依次存放 $(m-1)$ 行的 $(m-1, 0), \dots, (m-1, n-1)$ 像素的颜色值,其次为 $m-2$ 行,直

到位图阵列信息的最后行依次存放 $(0, 0), \dots, (0, n-1)$ 像素的颜色值。

假设记录一行像素需 Linebytes 个字节, 则位图阵列信息的大小为 Linebytes*Height 个字节。上式中的 Linebytes 为 4 (32 位) 的整数倍, 其取值规则为: $\text{Linebytes} = ((\text{Width} * 8 + 31) / 32 * 4) * 3$, 故指向 DIB 第 i 行, 第 j 个像素的指针为: 指向原 DIB 图像指针加上 $\text{Linebytes} * (\text{Height} - i - 1) + j * 3$ ^[37~39]。

5.4.2 真彩色位图处理

对位图文件进行处理, 必须首先找到位图数据指针, 才能读取位图图像数据。通常有两种方法: 一是在资源文件中定义位图, 利用 Windows 资源函数来装载位图; 另一种是对位图文件进行操作, 直接打开文件, 按 DIB 位图文件的结构读出相应的位图信息。本设计采用第二种方法实现, 具体步骤为: 首先通过 GetDocument() 函数获取文档, 再将所要读取的位图图像输出到一个内存对象 lpDIB (包括位图信息头和调色板信息 (真彩色位图没有调色板)), 由 GlobalLock() 实现, 获得指向 DIB 的指针 lpDIB 后就可以很容易的计算得到指向该位图数据块的指针, 方法是将 lpDIB 后移信息头大小的字段再加上调色板大小的字段 (真彩色位图为 0) 就获得指向 DIB 像素的指针 lpDIBBits, 这样便可方便地提取位图图像数据。

5.4.3 真彩色图像拼接

图像拼接是根据数字图像的特点, 即利用图像重叠部分具有相似的像素值这一特征, 将具有重叠部分的若干幅图像拼接成一幅图像。具有重叠部分的若干幅图像由于在色彩、亮度等方面不可能完全一致, 所以具有相同内容的图像部分的像素值会有一定的差别。利用数字算法寻求两幅图像内容上相重叠的部分, 然后将得到的值加权处理, 从而形成一幅图像^[40, 41]。

因此, 寻找图像重叠部分是图像拼接的关键因素, 在第 5.3 节详细讲述了灰度图像拼接具体实现, 提到了比值法和模板匹配法。寻找到匹配点后, 将不同的部分叠加起来, 形成一幅完整图像。相对于灰度图像, 真彩色图像只是每个像素需要字节数不一样, 由于真彩色图像每个像素对应 R, G, B 三个分量, 由前面知道 YUV 色系中 Y 表示亮度, Y 可以由 R、G、B 通过一定比例得到, $Y = 0.299 * R + 0.587 * G + 0.114 * B$, 可以看到 G 分量所占比例最大, 达到一半以上, 在本设计中就是基于每个像素点 G 分量来寻找匹配点的, 当然对于质量很高的图像, 可以采用每个像素的 R 和 B 分量来寻找匹配点。也有一些参考文献考虑整个 Y 来寻找匹配点, 这样又多了一步求和运算, 降低了算法速度, 本设计分别采用这两种方法来实现彩色图像拼接, 但在实际应用中一般基于 G 分量信息

来搜索匹配点, 毕竟采用此算法速度快, 实时性高。

由此可见, 彩色图像拼接和灰度图像拼接差别在于匹配点寻找这一步, 其他原理和算法都一样, 前面认真讨论了灰度图像拼接步骤, 关于彩色图像拼接步骤就不展开讨论了。在本设计软件中下拉菜单会分别出现彩色图像拼接和灰度图像拼接, 用于不同色彩的图像处理, 彩色图像拼接下拉菜单如图 5-10 所示, 这样针对不同色彩的图像采取不同的处理。

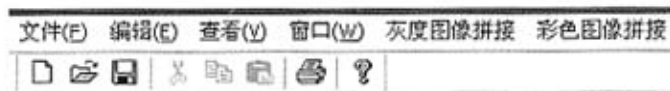


图 5-10 添加彩色拼接菜单

Fig.5-10 Add-on menu for color image mosaic

既然两个图像的亮度、色度等特征不可能完全一致, 如果将两幅图像的相同部分简单叠加, 拼接处会产生明显的接缝。为了消除拼接缝隙, 在两幅图的重叠区域, 将两帧图的像素值按一定的权值合成到新图。如图 5-11 所示, 在某一条扫描线上, 图像 I 和 I' 在区间 $[x_1, x_2]$ 上重叠, 那么新图 I'' 在这个 $[x_1, x_2]$ 区间上 x 点处的取值如下:

$$I''(x) = I(x) * W(x) + I'(x) * W'(x) \quad (5-3)$$

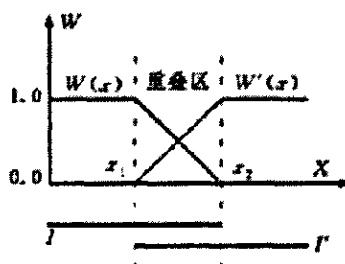


图 5-11 图像合成的函数

Fig.5-11 Image stitching function

为了达到平滑连接的目的, 需要对重叠区域进行图像融合这一步, 在 2.4 节讲述了常用的 4 种图像融合方法, 这里采用双线性插值法, 彩色图像可将 R、G、B 值分开计算即可。

为了进一步增强拼接的过渡效果, 在文献中提到在拼接的地方引入了微调系数 a ($0 \leq a \leq 1$) 即:

$$I''(x) = I(x) * W(x) + I'(x) * W'(x) + a \quad (5-4)$$

这样可以通过调节 a 的值实现拼接效果。

5.5 本章小结

本章首先介绍了数字图像处理的基础知识，接着介绍了 VisualC++ 软件在数字图像处理中的应用，重点介绍了灰度图像和彩色图像拼接具体实现过程，采用 VisualC++ 软件设计了一套适合本设计的算法。

第 6 章 实验与数据分析

在 第 5 章详细讲述了灰度图像和彩色图像拼接具体, 本章通过大量拼接实验, 从而分析本设计所采用算法的可行性。评价一个算法好坏的重要标准有两个: 精度和速度。在实际应用中, 首先考虑精度问题, 其次考虑在满足精度的前提下如何提高算法的速度的问题。本章就算法的精度和速度加以讨论分析, 并指出算法的不足以及改进措施。下面分别对灰度图像和彩色图像进行拼接实验, 并对同样图像采用不同算法拼接后拼接点位置和拼接时间进行比较, 从而对算法的精度和速度加以评价。

实验中采用的数据均是采用相机拍摄的, 为了保证待拼接源图像在同一个视角平面, 所以保持相机垂直距离不变, 只是水平向移动, 拍摄多幅具有重叠的图像, 这样图像拼接时候就减少坐标变换这个步骤。

6.1 灰度图像拼接实验及数据

分别选取不同大小的 6 幅灰度图像进行拼接实验, 并比较不同方法所得到图像的拼接点位置和匹配时间。算法的精度可以用图像匹配点位置来衡量, 如果针对同样的图像, 采用不同的方法, 得到的图像匹配点位置坐标仅仅有几个像素的差距, 说明图像匹配算法合理。在实验中, 为了验证图像匹配算法的合理性, 看其能否适合不同图像拼接, 在实验中选取了不同的图像进行实验对比。实验图像数据如表 6-1 所示。

表 6-1 拼接源图像数据
Table6-1 The data of source images

名称	图像 1 尺寸(pixel)	图像 2 尺寸(pixel)
标准混凝土图像	100*576	160*576
小女孩	151*255	185*256
小熊	144*183	187*195
经典图像	158*242	164*242
太阳	264*444	372*518
梅花	370*402	473*566
航天飞机	341*369	409*460
巴黎凯旋门航拍图	529*609	563*612
行星图	467*457	641*576

本设计在模板设置对话框里添加了图像信息, 可以显示图像大小, 比值模板设置对话框可以显示图像坐标 (X, Y), 列长度, 列间距, 这样比值初始模板

有四个参数可以修改,从而修改初始比值模板大小,有利于比值法拼接成功率,也加强了图像拼接算法的可适用性。模板匹配法设置对话框同样提供了图像大小信息以及模板坐标,模板匹配法里采用正方形模板,所以高度和宽度信息意义一样,这里只显示高度信息就足够了。下面以图像拼接中经典图像为例,进行图像拼接实验分析,待拼接图像如图 6-1 所示。



图 6-1 源图像 1

Fig.6-1 Source image one

对图 6-1 分别采用比值法和模板匹配法进行拼接,拼接前比值法初始模板和模板匹配法初始模板设置分别如图 6-2 所示。



图 6-2 初始设置

Fig.6-2 The initial set-up

只要点击“确定”按钮就可以进行图像拼接,本设计中初始模板可以修改,因为图像质量不一样,可能初始模板无法正确拼接,初始模板可动态修改,增强了算法的适用性,这也是本设计的创新点,两种拼接方法结合使用,可提高拼接成功率,因为前面提到有些图像使用比值法可能无法实现正确拼接。

首先采用比值法对图 6-1 进行拼接,比值法拼接结果如图 6-3 所示,可以看到拼接后显示信息为:图像拼接用时 351 毫秒,最小值为 0.000 000,匹配点坐标为 (23, 121)。



图 6-3 比值法拼接结果

Fig.6-3 The mosaicing result of ratio method

其次采用模板匹配法对图 6-1 进行拼接实验, 这里采用最小差值法, 拼接结果如图 6-4 所示, 可以看到模板匹配法用时 1072 毫秒, 最小值也是 0.000 000, 匹配点坐标也是 (23, 121)。



图 6-4 最小差值法拼接结果

Fig.6-4 The mosaicing result of the minimum D-value

从图像拼接质量看, 两种方法拼接效果都不错, 但比值法和模板匹配法的差别主要体现在拼接时间上, 很明显比值法用时比模板匹配法少, 这也是为什么本设计默认拼接算法用比值法的原因。从对图 6-1 拼接结果来看, 图像拼接点一样, 说明图像拼接算法达到了一定的精度, 满足设计要求, 从图像拼接时间来看, 图像拼接算法满足一定速度, 比值法用时最少, 速度最快, 拼接前应该首先采用此方法。

下面以表 6-1 所列图像为待拼接源图像进行拼接实验比较, 来评价算法的精度和速度, 从而进一步指出算法的不足和改进的措施。表 6-2 列出了表 6-1 所列图像拼接后结果。

由表 6-2 数据可以看出, 对于某些比值法无法实现拼接的图像, 可采用模板匹配法正确拼接, 这里分析一下比值法无法拼接的原因, 表 6-1 中的行星图如图 6-5 所示。

表 6-2 图像拼接数据 (时间单位为 ms)

Table6-2 The data of image mosaics

测试图像	比值法坐标	比值法时间	模板法坐标	模板法时间
标准混凝土图像	(22, 102)	50	(22, 102)	103
小女孩	(37, 127)	411	(37, 127)	1 042
小熊	(8, 99)	220	(8, 99)	791
经典图像	(23, 121)	300	(23, 121)	971
太阳	(43, 244)	3 104	(43, 244)	14 160
梅花	(68, 316)	5 408	(68, 316)	38 456
航天飞机	(52, 230)	2 574	(52, 230)	21 902
巴黎凯旋门航拍图	(72, 308)	7 541	(72, 308)	99 123
行星图	无法拼接	无法拼接	(89, 275)	74 818

在前面提到, 采用比值法进行拼接涉及一个比值模板是否存在的问题, 如果比值模板中出现分母为零, 也就是说分母中至少出现一个像素点的灰度值为零, 可以看出行星图中出现很多黑色像素点, 且分布非常广泛, 导致比值模板创建失败。

图 6-6 比值法拼接结果显示最小值为一个无穷大数, 比值匹配法失效, 在前面提到的比值法缺陷就是这个问题, 解决办法就是采用模板匹配法, 来完成图像拼接, 从而弥补比值法不足。

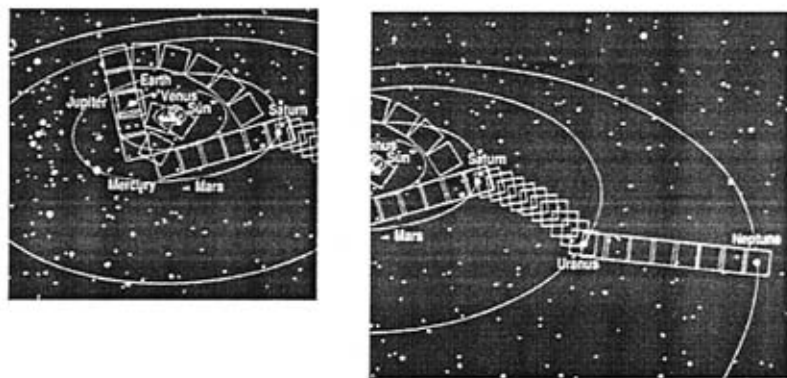


图 6-5 行星图

Fig.6-5 The planet image

从比值法拼接后结果图 6-6 中也可以看出, 图像拼接失败, 虽然拼接完毕, 但拼接结果是错误的。



图 6-6 比值法拼接结果

Fig.6-6 The mosaicing result of ratio method

由于图 6-5 待拼接图像里黑色像素点太多，很容易导致比值法失效，即使修改比值设置模板也无效，前面还提到模板匹配法主要是用时多，但模板匹配法不存在找不到拼接点的问题，所以可以采用模板匹配法来弥补比值法的不足。图 6-7 是采用模板匹配法中最小差值法拼接结果。

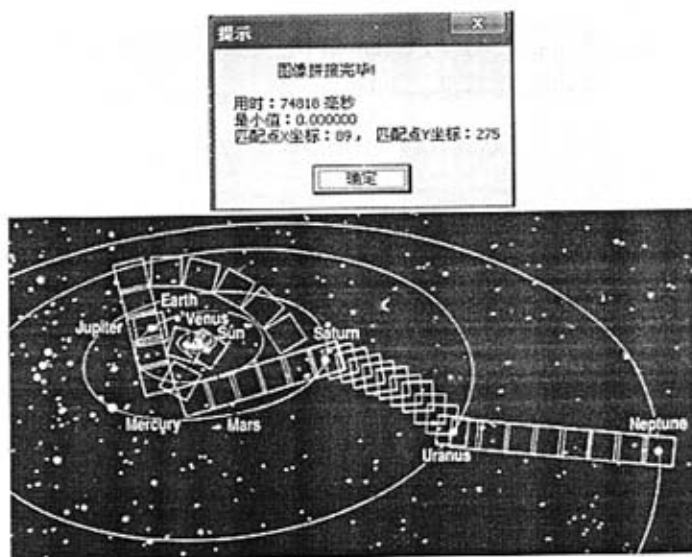


图 6-7 最小差值法拼接结果

Fig.6-7 The mosaicing result of the minimum D-value

比较两种方法拼接后的拼接结果图 6-6 和图 6-7，从视觉效果上可以判断出模板匹配法是正确的，模板匹配法虽然耗时多，但基本能找到匹配点，从而可以弥补比值法在某些情况下无法实现拼接的不足。

6.2 修改初始模板实验及数据

电脑硬件速度也是影响拼接速度的一个重要因素,本设计基于 Pentium(R) 4 CPU,主频 1.70G HZ,640MB 内存。如果在拼接的时候,还有其他软件占用系统资源,拼接速度也会受到影响,实验中,只运行拼接软件,最大程度提高拼接速度。这里,我们不去考虑电脑硬件的问题,在实际中完全可以改善硬件设备来提高软件运行速度。当然还有一个问题,就是图像重叠率的问题,重叠率太低也不能保证图像正确拼接,在本设计中,获取图像时候已经保证待拼接图像都满足 20% 以上的重叠率,所以重叠率问题在本设计中也不作考虑的因素。

6.2.1 修改比值法模板参数实验及数据分析

下面的实验主要是考虑修改初始模板来进行实验,验证算法的可行性。比值法初始模板有四个参数可以修改: X 坐标值, Y 坐标值,列长度,列间距。修改任何一个就可以修改模板,对于比值法,比值模板中两列的水平间隔应不小于 20 个像素单位,间隔太小,灰度变化不明显,易造成误匹配,如果间隔太大,第二列可能越界。比值法最大的缺陷在于,如果模板 M 中第二列出现黑色像素点(灰度值为 0),比值为无穷大,用以表示模板 M 和 N 的评价函数 E 的值为无穷大,比值法失效,找不到匹配点,以前的参考文献都忽略了这个问题。对于模板匹配法,只要保证初始模板存在于图像重叠区域就可以。为了保证初始模板存在,我们在两种方法拼接前,添加了一个模板参数设置对话框。下面以比值模板参数设置对话框为例进行说明。

影响比值法模板选取的三个因素:①:模板 M 第一列第一个像素坐标 (Ratio_M_x , Ratio_M_y);②:模板 M 中两列间距 Ratio_M_span ;③:模板 M 长度 Ratio_M_length 。默认参数设置为: $\text{Ratio_M_x} = \text{width1} * 5 / 7$ (width1 为 image1 的宽度); $\text{Ratio_M_y} = \text{height1} / 2$ (height1 为 image1 的高度); $\text{Ratio_M_span} = 20$; $\text{Ratio_M_length} = \text{height1} / 5$,使 M 位于 image1 右中。比值算法初始模板设置范围为: $\text{Ratio_M_x} + \text{Ratio_M_span} < \text{width1}$, $\text{Ratio_M_y} + \text{Ratio_M_length} < \text{height1}$ 。模板动态设置优点:直观;拼接效果不好时,可修改模板,重新找最佳匹配点;模板大小不固定,可适用于大小不同图像。 N 搜索范围为 image2 的左半区域,减少了搜索时间。

下面以三组待拼接图像为例进行比值法拼接实验,对每组图像都进行四次实验,分别按照修改 X , Y 坐标值,修改列长度,修改列间距的顺序进行实验,并记录实验数据,并比较拼接效果。待拼接图像选取表 6-1 里的小熊、航天飞机以及行星图来进行拼接实验,这样选取是有原因的,小熊图像质量比较不错,

航天飞机图像质量一般，而行星图黑色像素点太多，对于比值法可能行不通。

首先进行第一组实验，实验对象如图 6-8 小熊图像，图 6-9 为其初始模板设置参数，参数有四个，其中 X 和 Y 代表坐标值， X 、 Y 任何一个变化都影响模板设置，这里只改变其中一个，坐标中两个变量同时改变的情况同一个改变的情况，当然，这四个参数可以同时修改，这里只做一个参数修改的情况，并且还有一个问题，就是修改参数也可能导致拼接不成功，这里列举的实验数据只是随机性实验得到的结果。



图 6-8 小熊

Fig.6-8 The little bear image

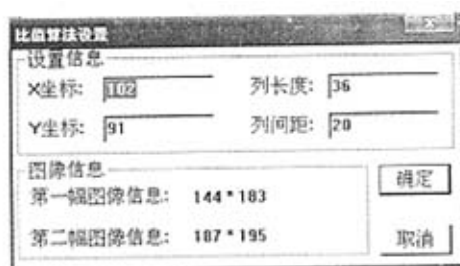


图 6-9 比值法初始设置

Fig.6-9 The initial set-up of ratio method

表 6-3 给出了修改坐标、列长度、列间距后实验数据，这里每次实验只修改一个参数，而保持其他参数不变。

表 6-3 小熊图像拼接实验

Table6-3 The experiment result of the little bear image

	参数修改	拼接点坐标(pixel)	拼接时间 (ms)
1	X: 110	(8, 99)	221
2	Y: 99	(8, 99)	210
3	列长度: 49	(8, 99)	260
4	列间距: 30	(8, 99)	200

比较以上进行的四次实验，可以得到一个结论，对于质量很高的图像例如实验中采用的图 6-8，初始模板基本可以保证拼接成功，修改拼接参数对拼接点影响不大，只要模板存在，基本能保证拼接成功，当然这里进行的实验是随

机的, 实际中需要进行很多次实验, 表 6-3 说明比值模板设置提高了拼接灵活性, 同时拼接时间不同说明参数修改也会影响拼接速度, 至于模板在那个位置拼接速度最快是不确定的。

第二组实验采用图 6-10 航天飞机图像, 采用比值匹配法, 拼接前默认参数设置为: 坐标 (243, 184), 列长度为 73 个像素, 列间距为 20 个像素。

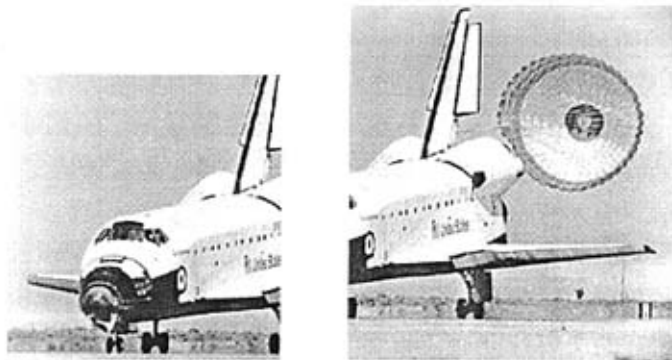


图 6-10 航天飞机图

Fig.6-10 The aerospaceplane image

修改四个参数, 进行实验得到数据如表 6-4 所示。

表 6-4 航天飞机实验数据

Table6-4 The experiment data of the aerospaceplane

	参数修改	拼接点坐标(pixel)	拼接时间 (ms)
1	X: 250	(59, 230)	2 524
2	Y: 180	(52, 226)	2 494
3	列长度: 100	(52, 230)	2 400
4	列间距: 30	(52, 230)	2 443

表 6-4 中拼接点有一点差距, 但不明显, 仅仅几个像素点的差距, 这与图像质量有关。对比表 6-4 和表 6-1, 可以看到增大列长度和列间距能保证拼接效果更好, 因为列长度或列间距增大, 这样采取的模板大了, 搜索的范围相对减小, 模板中的像素点多了, 有利于拼接成功。

第三组实验采用前面提到的行星图, 如图 6-5 所示, 比值法拼接前默认参数设置为: 坐标 (333, 228), 列长度 91 个像素, 列间距 20 个像素。拼接实验数据如表 6-5 所示。

由表 6-5 中实验数据可以看出, 对行星图采用比值法进行拼接, 即使修改初始模板参数设置也很难完成正确拼接, 这同时指出了比值法的缺陷, 因为一幅图像中黑色像素点分布是不均匀的, 假设算法修改为先寻找黑色像素点, 然后再进行拼接, 这样算法时间开销会很大, 且不能保证拼接就一定能成功, 所以需要改进比值法, 本设计的思路就是采用模板匹配法来弥补比值法的不足。在后面模板匹配法实验里可以看到模板匹配法可以完成图像拼接。

表 6-5 行星图实验数据

Table6-5 The experiment result of the planet image

	参数修改	拼接点坐标(pixel)	拼接时间 (ms)
1	X: 350	(0, 0)	30 604
2	Y: 300	(0, 0)	25 847
3	列长度: 180	(0, 0)	46 968
4	列间距:	(0, 0)	27 700

6.2.2 修改模板匹配法初始参数实验及数据分析

影响模板匹配法的参数有三个: X , Y 坐标和模板大小, 本设计模板采用正方形模板, 高度和宽度一样, 所以可以采用模板高度作为影响模板大小的参数之一。模板匹配有两个评价函数: 求模板差的绝对值之和的最小值和; 求极大相关系数。相对于比值法来说, 模板匹配法的缺点是时间开销大了, 但模板匹配法的优点是可以保证图像拼接成功, 来弥补比值法的缺陷。

下面的实验还是以图 6-8 小熊、图 6-10 航天飞机和图 6-5 行星图为例进行拼接实验。

第一组实验是对小熊图像进行模板匹配, 分别进行修改 X 坐标值, Y 坐标值和修改模板高度三次实验, 小熊图像模板初始参数为: (102, 91), 模板高度为 16。实验数据见表 6-6, 并比较表 6-2 和表 6-6, 采用比值法和模板匹配法得到的拼接点坐标都是 (8, 99), 分析得出修改模板初始参数可以得到正确的拼接效果, 说明模板匹配法对于图像质量比较好的图像进行拼接可适用性高, 从而说明本设计中模板匹配的算法合理, 有效。

表 6-6 小熊模板匹配参数

Table6-6 The template parameter of the little bear

	修改参数	拼接点坐标(pixel)	拼接时间 (ms)
1	X: 110	(8, 99)	772
2	Y: 89	(8, 99)	771
3	模板高度: 30	(8, 99)	2 503

第二组实验操作对象是图 6-10 航天飞机, 初始参数为: 坐标 (243, 184), 模板高度为 37 个像素。按照上面第一组实验的步骤进行实验, 得出的实验数据参考表 6-7, 并比较表 6-7 和表 6-2, 来分析修改模板参数后图像拼接效果。

表 6-7 航天飞机模板匹配参数

Table6-7 The template parameter of aerospaceplane

	修改参数	拼接点坐标(pixel)	拼接时间 (ms)
1	X: 250	(59, 230)	23 885
2	Y: 194	(58, 230)	25 176
3	模板高度: 50	(59, 230)	46 577

表 6-2 中数据显示比值法和模板匹配法拼接位置都是在坐标 (59, 230) 处, 修改模板初始参数得到拼接点位置大约也在这个位置, 误差仅为 1 个像素。

第三组实验是对采用比值法无法实现拼接的图 6-5 行星图进行的。模板拼接实验前默认参数值为: 坐标 (333, 228), 模板高度 50 个像素。实验数据为表 6-8。

表 6-8 行星图模板匹配参数修改数据

Table6-8 The template parameter of figure 6-5

	修改参数	拼接点坐标(pixel)	拼接时间 (ms)
1	X: 353	(89, 275)	76 430
2	Y: 240	(89, 280)	81 307
3	模板高度: 60	(89, 275)	108 907

对比表 6-8 和表 6-2, 分析修改模板参数对拼接的影响。表 6-2 里不修改初始模板得到拼接点坐标为 (89, 275), 修改后得到拼接点坐标基本在 (89, 275) 处, 从表 6-8 可以看出修改 Y 为 240 时, 得到拼接点坐标为 (89, 280), 拼接点 Y 坐标值有变化, 误差为 5 个像素, 但从视觉上我们看不出拼接后图像之间的差距, 毕竟 5 个像素的误差不大。

6.3 彩色图像拼接实验及数据

上面两节讲述了灰度图像拼接实验, 对大量图像进行了拼接实验, 文中只是举了三个典型的图像。本节进行彩色图像拼接实验, 并对拼接实验得到的数据进行分析, 指出拼接算法存在的问题。

前面讨论过彩色图像处理是根据 G 分量来实现的, 当然也可以根据三个分量的线性叠加来求取一个亮度值 Y , 然后按照灰度图像拼接点搜索方法进行拼接, 所以彩色图像拼接算法可以有两大类, 共六种方法。如果按照 RGB 中 G 分量来搜索匹配点, 则彩色图像拼接方法有: 比值法、最小差值模板匹配法和极大相关系数法; 如果根据 R, G, B 分量来求取亮度分量 Y , 则同样有: 比值法、最小差值模板匹配法和极大相关系数法。

本设计分别采用这六种方法进行多幅图像拼接实验, 来验证算法是否适合彩色图像拼接, 并比较拼接效果。这里以两幅不同图像为例, 进行数据分析。

6.3.1 根据 G 分量的图像拼接实验及数据分析

实验 1 采用的图像如图 6-11 所示，图中左边图像大小为 394*476 个像素，右边图像大小为 408*517 个像素。

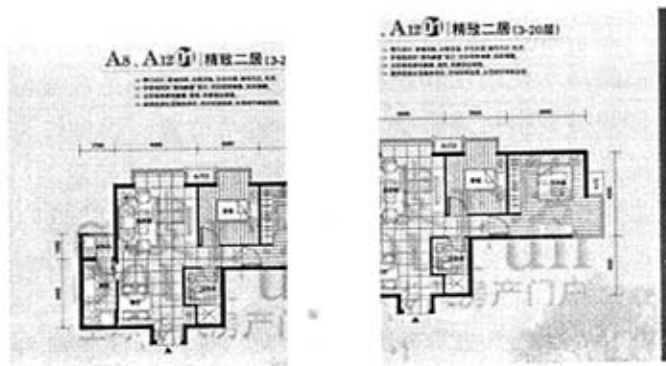


图 6-11 源图像 2

Fig.6-11 Source image two

比值法初始模板参数参考图 6-12。

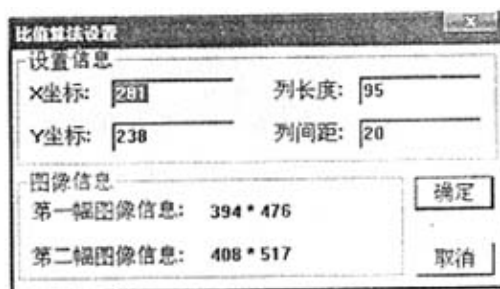


图 6-12 比值模板参数

Fig.6-12 The set-up of ratio method

可以看到比值法列间距还是 20 个像素点，这也说明了对于彩色图像拼接可以按照灰度图像拼接方法进行，只是亮度信息采用 G 分量。分别采用比值法、最小差值法和极大相关系数法进行实验，实验按照默认参数进行，实验结果如图 6-13 所示。

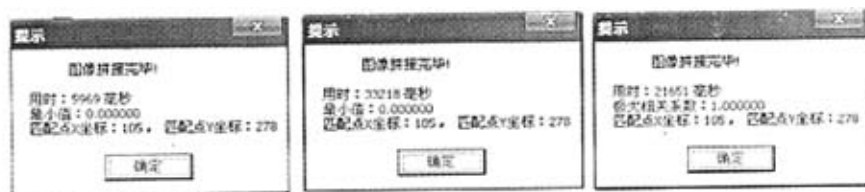


图 6-13 实验结果

Fig.6-13 The result of experiment

比较分析拼接点位置，可以看到拼接点一致都是 (105, 278)，拼接时间

分别为 5 969ms, 33 218ms 和 21 651ms, 这说明比值法速度最快, 最小差值法用时间最长, 因为这里涉及到求绝对值运算。拼接后效果如图 6-14 所示。



图 6-14 拼接效果图

Fig.6-12 Image mosaics effect

表 6-9 中记录了拼接实验得到的数据。

表 6-9 实验数据

Table6-9 Experiment data

采用方法	拼接点坐标	拼接时间 (ms)
1. 比值法	(105, 278)	5 969
2. 最小差值法	(105, 278)	33 218
3. 极大相关系数法	(105, 278)	21 651

第二组实验图像如图 6-15 所示, 待拼接图像中左边图像大小为 435*471 个像素, 右边图像大小为 464*426 个像素。从图 6-15 看出该图像色彩鲜明, 图像质量很好, 拼接结果见图 6-16, 图 6-17 为比值法拼接效果图, 从直觉上看拼接效果不错。

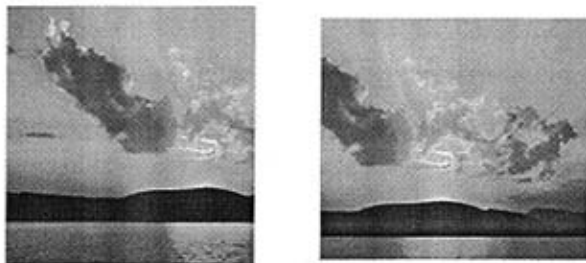


图 6-15 拼接源图像 3

Fig.6-15 Source image three



图 6-16 拼接结果

Fig.6-16 The mosaic result



图 6-17 拼接效果图

Fig.6-17 Image mosaic effect

拼接数据分析如表 6-10 所示。

表 6-10 实验数据

Table6-10 The result data

采用方法	拼接点坐标	拼接时间 (ms)
1. 比值法	(169, 196)	4186
2. 最小差值法	(169, 196)	36 622
3. 极大相关系数法	(169, 196)	23 895

由表 6-10 可知，三种方法拼接点搜索一致，说明算法合理，但搜索时间存在明显差距，并且最小差值法用时最多。

6.3.2 根据亮度信息的图像拼接实验及数据分析

亮度信息 Y 可以由公式 $Y=0.299R+0.587G+0.114B$ 来求得，这样综合考虑三个分量的变化，可以说精度提高了。

这里只进行一组实验，拼接源图像采用图 6-15，分别采用三种方法进行拼接实验，拼接结果如图 6-18。

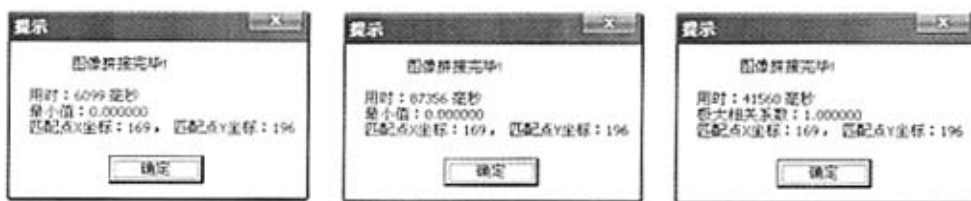


图 6-18 新的拼接结果

Fig.6-18 The new experiment result

拼接数据记录参考表 6-11。

表 6-11 新实验数据

Table6-11 The new experiment data

采用方法	拼接点坐标	拼接时间 (ms)
1. 比值法	(169, 196)	6 099
2. 最小差值法	(169, 196)	87 356
3. 极大相关系数法	(169, 196)	41 560

对比表 6-10 和表 6-11, 可以得出一个结论: 根据 G 分量和根据亮度信息来搜索拼接点位置一样, 但速度差距很大, 这主要是因为每次搜索都多了一步加法运算。

6.4 影响拼接效果的因素

影响拼接效果的因素有: 图像获取时光强是否一致; 图像重叠率大小; 图像获取方式; 图像黑色像素点分布 (针对灰度图像拼接)。本设计采用相同光强条件下获取图像, 所以光强变化对图像的影响不大。图像获取方式不同, 获取的图像就不同, 这存在一个图像是否在同一坐标面的问题, 如果图像不在同一个坐标面, 拼接前预处理中就涉及到坐标变换, 转变为同一个坐标面进行拼接。图像重叠率可以通过控制相机水平移动距离而加以控制, 可以保证获取的图像有较高的重叠率。黑色像素点分布对灰度图像拼接影响很大, 虽然比值法拼接效果不好, 但可以通过模板匹配法来完成拼接。

下面说明一下图像获取方式对拼接效果的影响, 以图 6-19 为例, 图 6-19 为北京工业大学综合楼照片, 获取源图像时, 保证了足够的光强 (光强变化不大), 且有足够的重叠率, 但水平移动相机有个角度差, 所以所获取的图像不在同一个平面内。

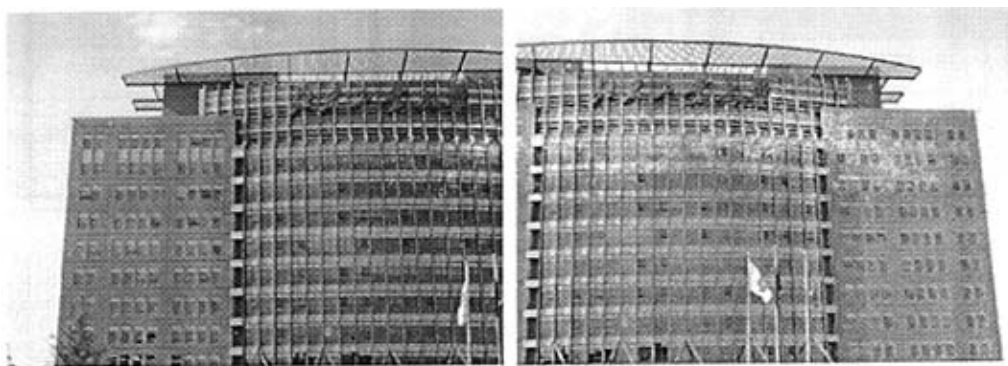


图 6-19 北京工业大学综合楼

Fig.6-19 The office-building of bjut

采用最简单的比值法进行拼接，拼接效果图如图 6-20 所示。

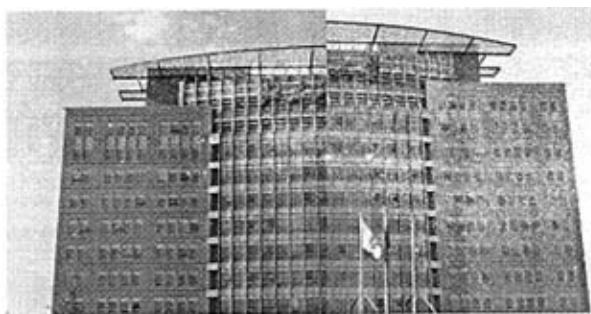


图 6-20 拼接效果图

Fig.6-20 The experiment effect

从图 6-20 的拼接效果来看，不在同一个平面的两幅图像，如果不采取坐标变换，得到的拼接结果是有问题的。固定相机，保持水平移动相机，重新获取两幅图像，拼接后结果如图 6-21 所示。

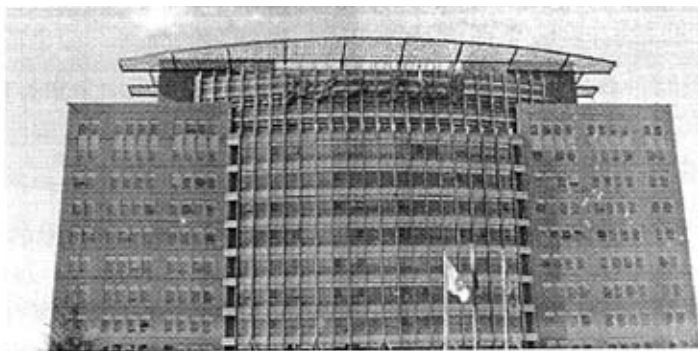


图 6-21 综合楼拼接结果

Fig.6-21 The mosaic result of office-building

本次设计总共实验了 300 多幅图像，在论文中只采用了几幅，以这几幅图像为例来分析算法的合理性。对于质量比较高的图像，比值法，最小差值法和

极大相关系数法都可以得到大致一样的拼接点，如果存在误差，误差仅仅为几个像素，算法不同，存在误差是合理的。本章也举了一个行星图的灰度图像拼接实验，比值法无法实现正确拼接，即使修改初始模板设置参数，也无法实现正确拼接，但采用模板匹配法可以实现正确拼接，这说明对于比值法无法实现拼接的图像，可以采用模板匹配法来实现拼接，从而弥补比值法的不足。

彩色图像拼接原理和灰度图像拼接原理基本一样，由于 G 分量占的比例最大，所以可以采用 G 分量来作为参考分量，当然也可以求亮度信息，即求 R, G, B 分量的线性叠加来，从实验的数据来看，采用 G 分量方法比后者速度快，这主要是后者每步搜索还涉及加法运算。彩色图像拼接也是采用比值法，最小差值法和极大相关系数法来搜索拼接点的，实验数据证明搜索的拼接点精度高，可适用性好。在满足精度的前提下，提高速度需要更深的研究。

6.5 本章小结

本章对灰度图像和彩色图像分别进行了拼接实验，实验中分别采用比值法、最小差值法和极大相关系数法，并比较了这三种算法的优点和不足，最后分析了影响拼接效果的因素。

结 论

本文主要研究了图像拼接技术,这是近年来图像处理领域的新兴技术,也是数字视频、虚拟现实场景及三维动画的基础,因此对它的研究具有十分重要的现实意义。

论文首先介绍了图像拼接的基本原理,总结了常用的图像拼接算法,分析了它们的性能特点,根据本研究的应用对象,选择了合适的算法,并利用 VisualC++6.0 软件实现了图像拼接。针对不同图像,进行拼接实验,分析了影响拼接精度和速度的各种因素,并对图像拼接得出了以下结论:

- 1) 图像拼接主要有三种方法:基于频域法、基于像素灰度级的匹配法和基于特征块的方法。基于频域法需要在时域和频域之间转换才能得到结果,基于特征块匹配法受到一定噪声干扰就很难实现拼接,且手动选取特征,操作不方便,时间开销大,而基于像素灰度级的匹配法利用像素之间的相似性来进行匹配,不需要变换。
- 2) 基于像素灰度级的匹配法有:比值法、最小差值法和极大相关系数法。比值法仅仅利用了间隔一定距离的两列的部分像素,拼接速度快,但有时无法实现正确拼接。最小差值法和极大相关系数法采用的模板是一样的,仅仅是评价函数不一样。分析了比值法的缺陷,并提出了改进措施,提出了一种基于比值法和模板匹配法相结合的新方法,增加了拼接成功率。
- 3) 对于一些图像质量很好的图像,采用三种方法可以很好的实现拼接,但如果图像获取不在同一个平面,就会导致拼接失败,出现误匹配,拼接要保证图像有严格的重叠率,至少保证在 20% 以上。

受实际条件和时间的限制,还有很多问题有待于进一步的研究:虽然实现了灰度图像和彩色图像拼接,考虑到实际应用中实时性要高并且图像获取时容易带进干扰,如何对含有干扰的图像进行正确拼接是下步研究的重点之一,另外如何提高拼接速度仍是研究的重点。在设计中,实验了 300 多幅图像,但这与实际应用还是有差距的,有些问题是不可预知的,必须加强实际应用实验,来提高拼接算法的适用性。研究动态图像拼接也是图像拼接的一个重要发展方向。

参考文献

1. 候舒维. 图像拼接技术研究. 西安电子科技大学硕士论文. 2005:1~3
2. 解凯, 郭恒业, 张田文. 图像 Mosaic 技术综述. 电子学报. 2004, 32(4):33~34
3. S. Richard. Video Mosaics for Virtual Environments. IEEE computer graphics and applications. 1996,16(2):22~30
4. 张峰, 贺贵明. 一种动态 Sprite 生成技术的研究与实现. 计算机工程与应用. 2003,12(3):119~120
5. S. E. Chen. QuickTime VR-an Image-Based approach to virtual environment navigation. Computer Graphics. 1995:29~38
6. R. Szeliski. Image mosaicing for tele-reality applications. IEEE Computer Graphics and Applications. 1994,13(6):44~53
7. C. T. Hsu, T. H. CHENG. Feature-based video mosaic. Image Processing. 2000,2(9):887~890
8. S. Peleg, J. Herman . Panoramic mosaics by manifold projection.Rico P. Proceedings of the Conference on Computer Vision and Pattern Recognition. Los Alamitos:IEEE CS Press. 1997:338~343
9. 屈志毅, 王玉珍, 钟声伟. 图像拼接中特征块的选取. 计算机工程. 2002, 28(9):51~54
10. 刘金根, 吴志鹏, 刘上乾, 殷世民. 一种基于特征区域分割的图像拼接算法. 西安电子科技大学学报(自然科学版). 2002, 29(6):16~17
11. 钟力, 胡晓峰. 重叠图像拼接算法. 中国图象图形学报. 1998, 3(5):44~47
12. 刘楠. 数字图像的自动拼接与合成技术研究. 南京理工大学硕士论文. 2004:6~8
13. E. Kang , I Cohen, G Medioru . A graph-based global registration for 2D mosaics. in:Proc. Of the 15 th Intl.Conf.on Pattern Recognition Barcelona,Spain.2000:221~223
14. P. Bao, D. Xu. Complex Wavelet-based Image Mosaics using Edge Preserving Visual Perception Modeling.Computers and Graphics. 1999,23(3):309~321
15. 仵建宁, 冯宗哲, 郭宝龙. 一种基于相位相关优化的图像拼接方法. 微电子学与计算机. 2006, 23(1):84~88
16. 邓红梅, 吴四夫. 基于相位相关算法的研究与实现. 信息技术. 2005, 4(10):21~22
17. Y. Kanazawa, K. Kanatani. Image moasicing by stratified matching. Image and Vision Computing. 2004,22(2):93~103
18. M. Morikawa, A. Katsumata, T. Higuchi, K.Kobayashi. An Image Processor Implementing Algorithms using Characteristics of Phase Spectrum of Two-Dimensional Fourier Transformation.In proc IEEE Int Symp. On Industrial Electronics,1999,3:1208~1209
19. 徐正光, 田清, 张利欣. 图像拼接方法探讨. 图像处理. 2006, 10(3):11~14
20. 李志刚. 边界重叠图像的一种快速拼接算法. 计算机工程. 2000, 26(5):37~38
21. 方青, 王博亮. 一种改进的基于比值模板匹配的显微图像拼接. 计算机工程. 2005, 31(24):77~79
22. 张红民. 基于模板匹配的彩色图像自动拼接方法. 微机发展. 2003, 13(7):112~113
23. 何斌, 马天予. VisualC++数字图像处理. 北京人民邮电出版社, 2001:88~124
24. P. Kumar , P. Anadan . Representation of scenes from collections of images. Proc of IEEE

- Workshop on Representations of Visual Scenes, Cambridge,Massachusetts,1995:10~17
25. 冈萨雷斯. 数字图像处理. 电子工业出版社, 2003:33~55
 26. B. S. Reddy, B. N. Chatterji. An FFT-based technique for translation,rotation,and scale-invariant image registration.IEEE Trans.Image Processing. 1996:201~206
 27. 殷飞, 桑农, 王洗. 一种新的序列图像匹配定位算法. 红外与激光工程. 2001, 30(6):422~425
 28. D. H. Kim, Y. I. Yoon, J. S. Choi. An Efficient Method to Build Panoramic Image Mosaics.Pattern Recognition Letters,2003,24(14):2421~2429
 29. P. Anadan. IEEE Workshop on Representations of Visual Scenes. Cambridge, Massachusets. IEEE Computer Society. 1995:
 30. 沈项军, 李峰, 陈金华. 一种基于特征跟踪的彩色序列图像拼接方法. 微型电脑应用. 2003, 19(3):55~57.
 31. 林木华, 游志胜, 邢霄飞. 基于 2 幂子图像的图像拼接算法. 中国民航飞行学院学报. 2006, 17(3):66~69
 32. 孙远, 周刚慧, 赵立初, 施鹏飞. 灰度图像匹配的快速算法. 上海交通大学学报. 2000, 34(5):702~704
 33. A. V. Oppenheim, R. W. Schafer, J. R. Buck. Discrete-Time Singal Processing. Pearson Computer Graphics. 1998:25~30
 34. C. Y. Chen, K. Reinhard . Image Stiching-Comparisons and New Tchniques.CAIP. 1999:615~622
 35. D. N. Wood, J. F. Hughes . Multiperspective panoramas for cell animation. Computer Graphics. 1997:243~250
 36. H. Steve, S. Harpreet, K. Rakesh. Automated mosaics via topology inference.IEEE Computer Graphics and Applications. 2002,22(2):44~54
 37. E. Noirfalise, J.T. Laprest, F. Jurie, M. Dhome . Real-time registration for image mosaicing. In Electronic Proc. of The 13 th BMVC,University of Cardiff, September 2002:617~625.
 38. H. Richard, Z. Andrew. Multiple View Geometry in Computer Vision. The Press Syndicate of The University of Cambridge,UK,2000:252~258
 39. 陈延国, 赵国良, 蒋彭龙, 郑勇斌. 真彩色位图图像的分解和合成. 应用科技. 2002, 29(5):110~112
 40. S.H. Sawhney. The true multi-image alignment and his application to mosaicing and lens distortion correction . IEEE Trans on PAMI. 1999,21(3):235~243
 41. 李睿, 张贵仓. 真彩色数字位图的处理. 哈尔滨师范大学自然科学学报. 2004, 20(1):44~46

攻读硕士学位期间取得的研究成果

已录用的学术论文

- 1 高新杰, 李德胜. 基于比值法和模板匹配法的灰度图像拼接. 微计算机信息. (已录用)

致 谢

首先，对导师李德胜教授表示衷心的感谢！从论文的选题、论证、研究到最后完成，无不凝聚着导师的心血。李老师专业知识渊博，治学态度严谨，为人师表，对待学生严格要求，为我们树立了求知和工作的榜样。本课题的研究，得到了李老师的悉心指导。攻读硕士的几年中，李老师在学术上给了我很多创新的灵感和耐心的帮助，在生活上给予了热情关心，不但调动了我学习的积极性，而且让我在实践中提高了动手能力和综合素质！

还要感谢在读硕期间给予悉心帮助的董天午老师，王跃宗老师和冯凡老师，是你们在科研和生活上给予的支持和关心使我顺利完成学业。

在此也非常感谢刘本东师兄、卢志远师兄、于丽杰师姐、聂林博士、高宗余博士、李翔、杨宇、杨建坤、种传波、叶乐志、刘振杰、刘喆、郭巧红、曾新新等同学给予的热情帮助和支持。

最后，特别感谢我的父母和女朋友李国珍，他们的爱是我生命的阳光、精神的支柱，他们给了我努力拼搏的力量！

图像拼接技术研究

作者：高新杰
学位授予单位：北京工业大学

本文链接：http://d.g.wanfangdata.com.cn/Thesis_Y1163271.aspx