

图像拼接算法研究

摘要

本文给出了图像拼接的基本流程，将研究方向分为最关键的两部分——图像配准与图像融合，分别详细介绍了图像配准和图像拼接的步骤，并对每个步骤分别给出了几种较为常见的算法。最后，分别从图像配准与图像拼接的角度对现有算法进行优化，并提出一种更适合自然景象的图像拼接算法。

关键词：图像拼接，图像配准，图像融合

ABSTRACT

This paper presented the basic flow of image mosaics and separated the research into two key distinctive parts, which are image registration techniques and image fusion techniques. Finally it came up with a new algorithm of image mosaics, which improved both the registraton and the fusion part.

KEY WORDS: image mosaics, image registration, image fusion

1 绪论

1.1 图像拼接技术概况

图像拼接技术就是将数张有重叠部分的图像拼成一幅大型的无缝高分辨率图像的技术。将多张拍摄自不同时间、不同视角甚至是不同的传感器的影像根据其重叠部分拼接起来是本文所要讨论的话题。

1.2 选题意义与研究目标

目前的普通摄像器械只能拍摄有限范围有限视角的景象，能拍摄全景图的相机一般价格昂贵。然而随着图像处理技术的发展，使用普通相机拍摄多张不同视角的相片也能够得到效果比较好的全景图。我们的研究目标是将现有的图像拼接方法进行整理并加以筛选，最后提出一种能够实现光照鲁棒性的图像拼接方法。

1.3 图像拼接的主要问题

图像拼接所要研究的问题主要有两个——图像配准与图像融合。

图像配准目前比较流行的方法可以分为两大类——基于特征点与基于区域的方法。基于区域的方法其实现算法较为简单，然而局限性很大。它的计算量很大，另外该方法对尺度、亮度以及旋转变化较为敏感，一般只适用于平移。而基于特征点的配准方法其速度快，并且对于灰度变化有较强鲁棒性，不过，精确度和稳定性还是要寻找更佳的方法来解决。基于特征点的方法的关键在于寻找特征点，较为常见的算法有 Harris 角点检测法与 SIFT 算法，这两种算法将在下文详细介绍。

图像融合目前的方法都比较简单，普遍使用线性加权平均法。由于图像的光照随机性比较大，要找出一种统一的调整光照的方法相当困难。本文将介绍几种其他的融合方法，并对其作出比较，最后提出一种新的光照鲁棒拼接方法。

2 研究内容及方法

2.1 图像配准算法介绍

图像拼接技术的发展很大程度上取决于图像配准技术的创新。近十年来，各种基于特征的图像配准技术被提出，其中包括 1999 年 David G.Lowe 提出的 SIFT 尺度不变特征转换算法。

图像配准分为 4 个流程：首先对待匹配的两幅图像提取特征点，然后对这些特征点进行描述，再根据描述将两幅图像中具有相似度较高的特征点匹配起来，最后消除错配。概括起来就是：提取特征点、特征点描述、特征点匹配与消除错配。下面将详细介绍每一个流程的具体算法。

2.2 特征点提取算法

2.2.1 提取范围的确定

在图像配准之前，必须先确定图像提取特征点的范围，即两幅相似图像的重叠区域。两幅图像中除了重叠区域，都是图像所独有的，因此在重叠区域之外提取特征点不仅是无意义的，甚至会妨碍特征点的正确匹配。

根据本课题所考虑的图像特点，图像的重叠部分只可能出现在图像的左部或右部。假设 $width$ 和 $height$ 分别表示图像的宽度和高度，本文提取特征点的范围为图像右侧宽 $width*5/12$ ，高 $height$ 的矩形区域内或左侧宽 $width*5/12$ ，高 $height$ 的矩形区域内。

2.2.2 Harris 算法提取角点

角点是图像特征点中非常重要的一类，简单的定义是至少两个方向上图像灰度直变化较大的点。在实际图像中，轮廓的拐角、线段的末端等都可能是角点。比较常见的角点检测方法有：Moravec 角点检测算法、SUSAN 角点检测算法以及 Harris 角点检测算法^[1]。

Harris 角点检测算法是目前比较主流的特征角点检测方法。Harris 算子是由 C.Harris 和 M.J.Stephens 在 1988 年提出的一种特征点提取算子。其处理过程表示如下：

确定特征提取范围后，在该范围内考虑采用式(1)Harris[2]矩阵算子提取角点：

$$H = \begin{bmatrix} \left(\frac{\partial I}{\partial x}\right) & \left(\frac{\partial I}{\partial x}\right)\left(\frac{\partial I}{\partial y}\right) \\ \left(\frac{\partial I}{\partial y}\right)\left(\frac{\partial I}{\partial x}\right) & \left(\frac{\partial I}{\partial y}\right) \end{bmatrix} * L \quad \text{式(1)}$$

其中 $I(x,y)$ 是图像点 (x,y) 处的灰度值， $\left(\frac{\partial I}{\partial x}\right)$ 、 $\left(\frac{\partial I}{\partial y}\right)$ 分别是图像在 x 和 y 方向的一阶导数。

L 为一个线性低通 $N*N$ 的高斯模板，用该模板对 H 的四个因子进行掩模滤波即可消除一定得噪声。经过高斯滤波后，通过式(2)角点响应函数 R 求取角点响应值：

$$R = \det(H) - k(\text{trace}) \quad \text{式(2)}$$

其中， $\det(H)$ 为矩阵 H 的行列式， $\text{trace}(H)$ 为矩阵 H 的迹， k 为一阈值系数，可以用来控制特征

点的个数。

求得角点响应值 R 后，将其与阈值 T 比较，只有当 $R > T$ 时，则将其视为初始特征点。

Harris 算子在图像旋转、灰度变化、噪声影响和视角变化的时候具有一定的鲁棒性，然而对于尺度变化较为敏感。

2.2.3 SIFT 算法提取特征点

SIFT算法是一种在不同的尺度空间下提取图像的特征点的方法，即尺度不变特征变换(scale invariant feature transform)。SIFT算法的过程比较复杂，在此对其基本步骤和目的做简单介绍，具体的特征点提取方法可参照相关文献。

2.2.3.1 Gaussian 金字塔和 DOG 金字塔的构造

在进行极值点检测之前，先对图像预处理，去除噪声，然后对图像进行线性插值放大，接着构造Gaussian金字塔和DOG金字塔对图像进行滤波。Gaussian金字塔和DOG金字塔的构造如图1所示。

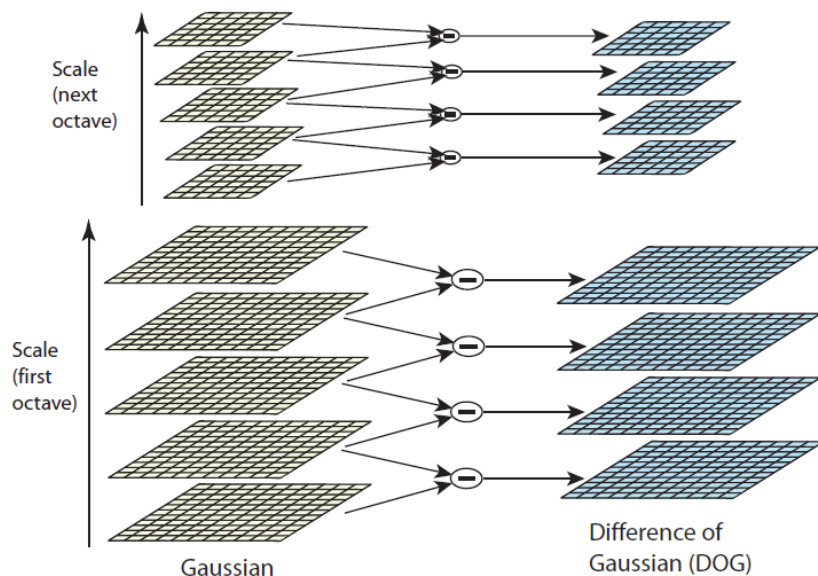


图1 Gaussian金字塔和DOG金字塔的构造

2.2.3.2 极值点的检测

极值点是指与其在同一层的周围8个点和上、下两层各9个点相比较，如果均小于其它点或均大于其它点，则称为极值点。

检测出的极值点并非都是想要的特征点，如一些低对比度的点对噪声敏感，并且DOG函数对边界有强烈的反应，因此要进一步去除这些点。

2.2.3.3 方位确定

经过去除低对比度点和消除边界影响，我们得到SIFT关键点。每个关键点由3个因素决定：模大小、所处尺度和方向。因此还需为每个关键点指定方位和相应模的大小。

$$m(x,y) = \sqrt{L(x+1,y) - L(x-1,y)^2 + L(x,y+1) - L(x,y-1)^2} \quad \text{式(3)}$$

$$\theta(x,y) = \arctan\left(\frac{L(x,y+1) - L(x,y-1)}{L(x+1,y) - L(x-1,y)}\right) \quad \text{式(4)}$$

式中， $m(x,y)$ 是每个像素点梯度模的大小； $\theta(x,y)$ 是 (x,y) 所处梯度的方向。L所用的尺度为每个像素点各自所在的尺度。

在以关键点为中心的邻域窗口内采样，并用直方图统计邻域像素的梯度方向。对每一个关键点，每个邻近的像素点对关键点的贡献是不一样的，用一个带权重的高斯分布窗口乘以相应每个像素点的梯度大小，通过这种方式来确定模的大小。

关键点的方位用梯度直方图来确定，将梯度直方图36等分，每份是10度。直方图的峰值代表了该关键点处邻域梯度的主方向，作为该关键点的方向。在梯度方向直方图中，当存在另一个相当于主峰值80%能量的峰值时，则将这个方向看作是该关键点的辅方向。

确定了关键点的3个主要信息：位置、所处尺度和方向后，由此得到一个SIFT特征区域。一个关键点可以被指定具有多个方向（一个主方向，一个以上辅方向），这可以增强匹配的鲁棒性。匹配时只要把图像旋转到主方向上，就可以达到旋转不变的目的。

2.2.4 特征点描述算法

特征点的描述方法比较多样化，有基于像素差平方和(SSD)的描述符，有基于互相关(CC)系数的描述符。这些描述符在大多数平滑的自然图像中所取得的效果大同小异，然而在对精度要求较高的科研图像中还需要更加详细的描述符。Hu 不变矩描述法有 21 维的特征向量来描述特征点，SIFT 算法中用到了 128 维的特征向量来描述特征点。然而随着特征向量的维数增多，计算量也将增大，因此找到描述精确性与匹配高效性的平衡是一个有待解决的问题。

特征点匹配是将不同图像上由同一场景点投影而成的特征点对找出来。目前出现的匹配算法可以分为两类：

- (1) 直接利用特征点周围区域的灰度信息求取归一化相关系数来判别是否匹配。
- (2) 首先利用特征点周围区域的灰度信息求取某种特征向量，再通过特征向量间的相似性来判别是否匹配。

由于特征向量对旋转、对比度、缩放有较大的鲁棒性，本文选择第二种方法。

2.2.4.1 特征向量的生成

特征向量的生成有多种方法，如 SIFT 算法中利用特征点为中心 8×8 的矩形框内的相位信息形成 32 维的特征向量，以及以 Hu 不变矩作为特征向量。这两种算法各自存在优缺点，本文采取的是根据图像的特点来自适应地选用其中一种算法进行特征向量的生成和匹配。下面对这两种方法分别进行介绍。

2.2.4.2 SIFT 特征向量

具体步骤如下：

- (1) 以关键点为中心取一个 8×8 的窗口，每个小格代表关键点邻域所在尺度空间的一个像素，箭头方向代表该像素的梯度方向，箭头长度代表梯度模值，图中蓝色的圈代表高斯加权的范围(越靠近关键点的像素梯度方向信息贡献越大)。
- (2) 在 4×4 的小块上计算 8 个方向的梯度方向直方图，计算每个梯度方向的累加值，即可形成一个种子点，如图 2 所示。

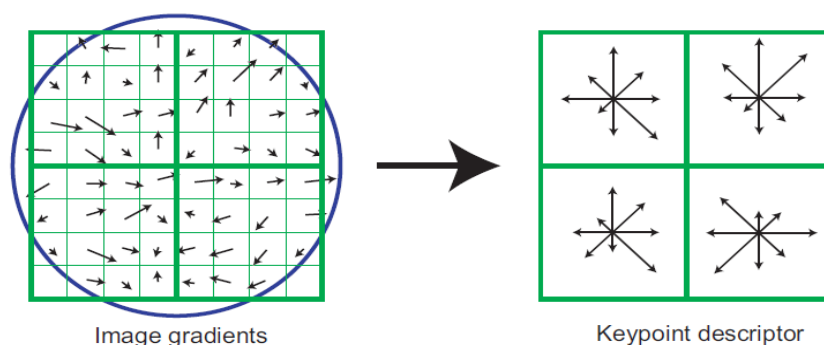


图2 关键点特征向量的生成

(3) 每一个关键点由2x2共4个种子点组成，每个种子点有8个方向向量信息。这种邻域方向性信息联合的思想增强了算法抗噪声的能力，同时对于含有定位误差的特征匹配也提供了较好的容错性。实际计算过程中，为了增强匹配的稳健性，可以使用4x4共16个种子点来描述，这样对于一个关键点就可以产生128个数据，即最终形成128维的SIFT特征向量。

2.2.4.3 Hu 不变矩特征向量

选择以特征点为中心半径为 r 的圆形区域分别计算该区域的红，绿，蓝三种分量的七维 Hu 不变矩，并将这三个 Hu 矩合并为一个 21 维的特征向量。其具体步骤如下：

(1) 在半径为 r 的圆形区域内分别计算红，绿，蓝三种分量的七维 Hu 不变矩 R 、 G 、 B 。

(2) 调整 R 各维的值，使得 $R(i) = |\log(R(i))|$ ， G 和 B 做同样操作。

(3) 合并不变矩 R 、 G 、 B 得到特征点 21 维的特征向量 $V=[R G B]$ ，并按照式(3)将向量模值进行归一化。

2.2.5 特征点初步匹配

此部分内容确定两特征点是否匹配。具体步骤如下：

(1) 在匹配图像中选择一个特征点 $P_r(i)$ ($1 \leq i \leq N_r$)， N_r 为匹配图像特征点的数目，假设其特征向量为 $V_r(i)$ 。在模板图像中选择一个特征点 $P_t(j)$ ($1 \leq j \leq N_t$)， N_t 为模板图像特征点的数目，假设其特征向量为 $V_t(j)$ ，通过式(4)所表示两特征向量各维的相对距离之和作为这两个特征点的相似性度量 S 。

$$S = \sum_{i=1}^n \frac{|V_t(i) - V_r(i)|}{|V_t(i)| + |V_r(i)| + \varepsilon} \quad \text{式(5)}$$

ε 为任意小整数，防止分母为 0 的情况发生。

(2) 对模板图像中所有的特征点计算它们与特征点 $P_r(i)$ 的相似性度量。记录相似性度量值最小的特征点 $PI(j)$ ，以及最小相似性度量值 Min ，并记录第二小的相似性度量值 $SMin$ 。如果 $Min < Thr * SMin$ ，则初步认为 $PI(j)$ 为 $P_r(i)$ 的匹配点，否则认为 $P_r(i)$ 在模板图像中无匹配特征点。其中 Thr 为一阈值，一般取值 0.8~0.9。

(3) 对于步骤 2 获得的 $P_r(i)$ 的匹配点 $PI(j)$ ，返回到匹配图像中通过步骤 1 和步骤 2 寻找 $PI(j)$ 在匹配图像中的匹配点 $P_r(i')$ 。如果 i 等于 i' ，则最终认为 $P_r(i)$ 与 $PI(j)$ 匹配，如果 i 不等于 i' ，则最终认为 $P_r(i)$ 在模板图像中无匹配点。

(4) 对匹配图像中所有的点按照步骤 1、2、3 求取它们在模板图像中的匹配点。所有的匹配点对构

成初始匹配集。

2.2.6 匹配方案的选择

以上两种特征向量的生成方式在生成过程、维数、描述精确度和计算复杂度方面都不相同，我们根据它们的特点，确定了两套初步匹配方案。

方案一：SIFT 算法提取特征点并生成特征点描述向量。

方案二：Harris 角点法提取特征点，生成 Hu 不变矩特征描述向量。

通过 matlab 实验，我们可以从下表看出两种方案之间的差别。

表一 两种匹配方案的对比表

	方案一	方案二
提取特征点用时		3.327s (harris)
特征点数	1600	159
生成特征矩阵耗时	6.167s (包括提取特征点)	0.288s
匹配耗时	3.846	0.044s
匹配点对数	219	30

*注：表中的数据是特定图片在特定计算机上的运算结果，数据没有绝对意义，只有相对意义。

显然，方案一提取特征点数量多于方案二，并且通过匹配得到了较多的匹配点对，也即意味着其匹配精度更高，能够满足一些特殊情况下的图像匹配，如尺度相差较大的图像和匹配度较低的图像。但方案一耗时远远大于方案二，在大多数情况下，图像的匹配不需要这么高的精确度，采用方案二更加合适。为了使匹配方案适用于各种情况，我们采用以下方法来进行特征点的提取和初步匹配：

(1) 用方案二对图像进行初步匹配。

(2) 得到的特征点数如果满足：匹配点数/图像总像素数 <0.0002 ，则认为精确度太低，无法进行匹配，转步骤(3)；否则初步匹配结束。

(3) 重新用方案一进行初步匹配。

由于方案一可以满足大多数待匹配图像的匹配精度要求，因此上述方案即可以满足普通匹配的时间要求，又能够在特殊匹配情况下正确地找出匹配点。

2.2.7 利用空间方向约束剔除误配点

对于摄像机绕其中心在某一平面旋转而不发生绕主轴旋转而拍摄的两幅图像，当将他们在有重叠部分的一端排在一起时，正确匹配点之间线段的倾斜角和长度都极为相近。因此在获得初始匹配集后可以利用倾斜角和距离去除误匹配点的方法。

在拼接时，所有图像要经过投影在柱面或球面后再进行特征点匹配等过程。经过投影后两幅图像已经处于同一个曲面，如果将两幅图像贴在曲面上，两者重叠部分理论上将会完全相同，所以两幅图像正确的匹配点都在曲面的同一个位置。按照本文方法将两幅图在含有重叠区域的一侧排成一行就相当于在曲面上平移图像，显然，平移过后，所有正确匹配点间的线段拥有完全相同的倾斜角和长度。如果没有经过投影而直接将图像匹配，我们只能说正确匹配点间的线段拥有近似相同的倾斜角和长度。

利用空间约束剔除误匹配点的具体做法如下：

算法中数组 $\phi[i]$ 记录出现过的第 i 种倾斜角， $d[i]$ 记录与 $\phi[i]$ 对应的距离值， $n[i]$ 记录在倾斜角误差 $thr1$ 范围内具有倾斜角 $\phi[i]$ 并在距离误差 $thr2$ 范围内具有距离 $d[i]$ 的匹配点对的数目， num 记录数组 $\phi[]$ 中元素的个数。 $\theta[i]$ 记录第 i 组匹配点对的倾斜角， $D[i]$ 记录第 i 组匹配点对的距离。 Np 为初始匹配集中点对的总数。

初始时数组 $\phi[]$ 、 $d[]$ 、 $n[]$ 、 $\theta[]$ 和 $D[]$ 都是空数组

(1) 依照式(5)计算匹配集中第 1 组匹配点对对应的倾斜角 $\theta[1]$ 和距离 $D[1]$ ，并令 $\phi[1]=\theta$ ， $d[1]=D[1]$ ， $n[1]=1$ 。

(2) 依照式(5)计算第 i 组 ($2 \leq i \leq Np$) 匹配点的倾斜角 $\theta[i]$ 和距离 $D[i]$ 。

(3) 将 $\theta[i]$ 和 $D[i]$ 与数组 $\phi[]$ 和 $d[]$ 中所有值做比较，如果存在某个 j 值满足式(6)，则认为这组倾斜角和距离值已经被记录过，按照式(7)更新相应数组的第 j 个元素，其余元素不变；如果数组 $\phi[]$ 和 $d[]$ 中所有值都不满足式(6)，则认为 $\theta[i]$ 和 $D[i]$ 为新一组倾斜角和距离值，按照式(8)将其记录到数组 $\phi[]$ 和 $d[]$ 中。

$$\begin{cases} |\theta[i] - \phi[j]| < thr1 \\ \frac{|D[i] - d[j]|}{d[j]} < thr2 \end{cases}, \quad 2 \leq i \leq Np, \quad 1 \leq j \leq num \quad \text{式(6)}$$

$$\begin{cases} \phi[j] = (\phi[j] \cdot n[j] + \theta[i]) / (n[j] + 1) \\ d[j] = (d[j] \cdot n[j] + D[i]) / (n[j] + 1) \\ n[j] = n[j] + 1 \end{cases}, \quad 2 \leq i \leq Np, \quad 1 \leq j \leq num \quad \text{式(7)}$$

$$\begin{cases} num = num + 1 \\ \phi[num] = \theta[i] \\ d[num] = D[i] \\ n[num] = 1 \end{cases}, \quad 2 \leq i \leq Np \quad \text{式(8)}$$

(4) 所有初始匹配集中的点对都进行步骤 2 和 3 的操作后，选择数组 $n[]$ 中值最大的下标 tag ，那么 $angleV = \phi[tag]$ ， $distV = d[tag]$ 为标准的倾斜角和距离值。再将所有匹配点对的倾斜角和距离值与这两个值比较。其中满足式(9)的点对为匹配正确的点对将其保留下来，不满足的点对将其剔除。

$$\begin{cases} |\theta[i] - angleV| < thr1 \\ \frac{|D[i] - distV|}{distV} < thr2 \end{cases}, \quad 1 \leq i \leq Np \quad \text{式(9)}$$

经过上述步骤后匹配点对中仍然有可能存在匹配错误的点对，称为野值，匹配正确的点对称为内点。在所有内点中也并不是所有点对都能够精确地满足同一单应矩阵，这种情况和野值的存在会严重影响单应矩阵的估计，因此必须将野值剔除并提取出在一定误差内满足同一单应矩阵的匹配点数最多的匹配点集，即最大内点集。目前提取最大内点集的算法主要有 RANSAC 和最小中值平方算法 (LMS)。由于当多于一半的数据是野值时，LMS 算法需要利用动态的距离来确定内点，故本课题选择了 RANSAC 算法来进行算法的实现。RANSAC 算法的具体实现步骤在相关论文上都有详细叙述，在此不做介绍。

实验结果

待匹配图像如下图。



(a)



(b)

图 3 待匹配图像

采用方案一（SIFT）的实验结果：

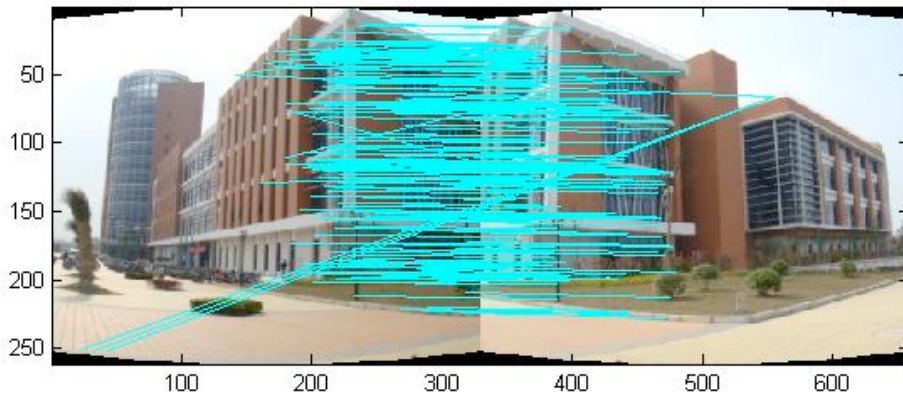


图 4 方案一初匹配后



图 5 方案一按空间约束剔除误配点后

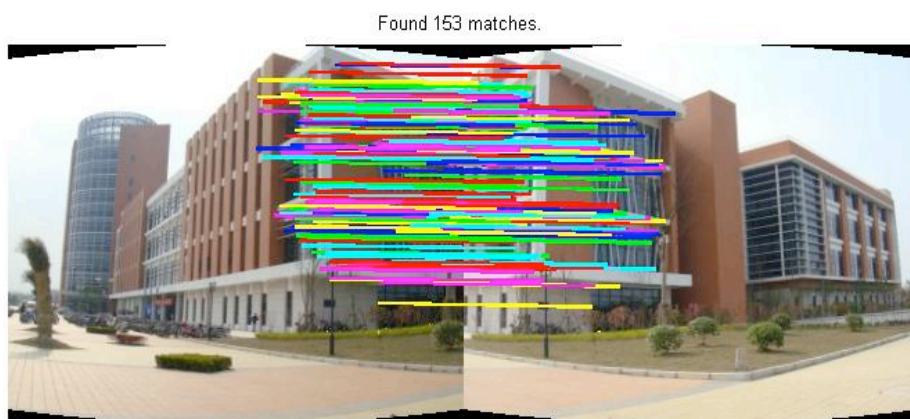


图 6 方案一 RANSAC 提取最大内点集后

采用方案二（Harris + Hu 不变矩）的实验结果：

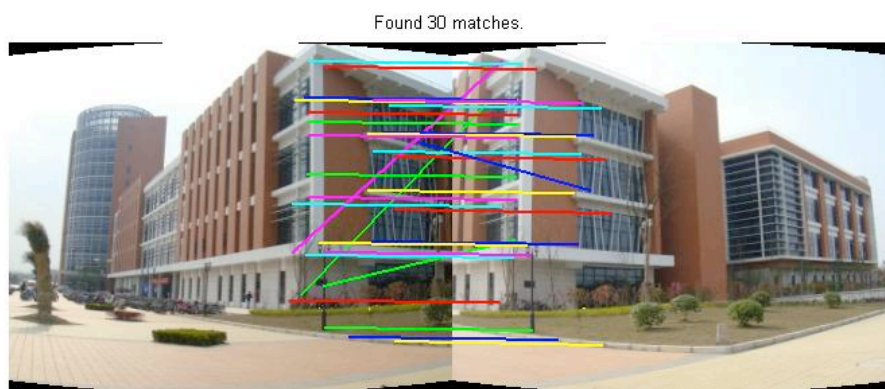


图 7 方案二初匹配后



图 8 方案二按空间约束剔除误配点后



图 9 方案二 RANSAC 提取最大内点集后

2.3 图像融合算法介绍

图像融合要处理三方面的基本问题：形变处理、光照调整与接缝处的色差处理。其基本过程是先从匹配点中选取有代表性的若干点，通过这些点来计算两幅图像之间的形变关系，再对图像作几何变换。然后调整两幅图像的曝光程度使其光照条件相近，最后通过融合方法是两幅图像自然地拼接在一起。

下面将逐步介绍融合过程的方法。

2.3.1 形变处理

2.3.1.1 图像坐标变换模型^[3]

如果对同一个物体在不同的方向进行拍摄获得两幅图片，其中一幅图像可以通过对应的二维映射变换得到另一幅图。一般的投影变换形式是： $X'=MX$ ，其中 X' 和 X 是两幅图像中的同一景物， M 矩阵为投影变换矩阵。下式是一个可以应用于任何平面投影关系的 8 参数矩阵方程。平面投影变换通过下面公式把一幅图像变换成了另一幅，解析出 8 个系数，就能确定两幅图像之间的投影关系，从而完成具有相同场景的两幅图像之间的转换

刚性变换与仿射变换是投影变换中比较特殊的两类，但也是较为常见的两类。它们基于 8 参数模型有简化的表达方式。

刚性变换的变化方程可表示为：

仿射变换的变化方程可表示为：

对原始图像进行变换矩阵操作后，并不能完整地转为目标图像。下面将说明原因以及解决方法。

2.3.1.2 前向映射与逆向映射

由于图像的像素点坐标都是整数表示，而映射后的像素的坐标不能保证是整数，若进行简单取舍，则图像中会出现明显“空洞”和“重叠”。

图像变换主要有两种实现方法，前向映射和逆向映射^[4]。这两种方法对变换的结果有着不同的影响。设变换前输入的像素坐标是 (x, y) ，相应的变换后输出图像的像素坐标为 (u, v) ，变换的前向映射和逆向映射分别定义为：

$$(u, v) = (U(x, y), V(x, y))$$

$$(x, y) = (U^{-1}(u, v), V^{-1}(u, v))$$

前向映射扫描输入的图像，把每个图像像素 (x, y) 的亮度值按照上式拷贝到 (u, v) 指定的像素位置。

逆向映射从输出图像点 (u, v) 出发，通过 (U^{-1}, V^{-1}) ，找到它在输入图像中的位置 (x, y) ，并把该位置处像素的亮度值传给图像中的 (u, v) 像素点。

逆向映射避免了空洞的出现，它保证了输出图像的的点都可以被扫描到。

2.3.1.3 图像插值

由于图像处理过程中需要对原始图像进行重新的缩放、旋转及视角变换等，此时常常会出现因变换导致非整数坐标点对应像素值的确定问题，图像插值技术是用来确定该类像素值最有效的方法之一。

下面比较两种常见的插值技术方法——最邻近点插值法和双线性插值法^[5]。

最邻近点插值法其思想是取插值点的灰度值与其最邻近点的灰度值相同。这种方法的优点是简单、快速。

计算过程如下：

- 1) 首先计算邻域内各点与插值点之间的欧式距离

- 2) 其次取与插值点之间距离最小的点的灰度值作为插入值

不过，由于采用这种插值法需要的参考依据很少，因此引入的误差比较大，插值点像素有可能会失真。与邻近点像素连在一起后很容易出现马赛克现象。如果要减少马赛克的情况需要一种涉及更多参考点来确定像素值的插值方法，这就是下面将提到的双线性插值法。

双线性插值法的思想是根据输入图像网格上的像素值来确定插值点的灰度值。

- 1) 在 x 方向上通过一阶线性插值求得 $f(x, 0)$ 与 $f(x, 1)$

$$f(x, 0) = f(0, 0) + x[f(1, 0) - f(0, 0)]$$

$$f(x, 1) = f(0, 1) + x[f(1, 1) - f(0, 1)]$$

- 2) 对 y 方向进行一阶线性插值，则可以求得 $f(x, y)$

$$f(x, y) = f(x, 0) + y[f(x, 1) - f(x, 0)]$$

- 3) 对图像中进行二阶线性插值

2.3.2 光照处理

分析图像的光照情况是一个非常复杂的过程。许多文献中提出的方法都较为简单，效果不佳，仅适用于少数情况。当一幅照片与另一幅照片整体曝光不同时，文献中通常提出的解决方案是：设一幅图作为参照图，将它的参数赋给另一幅。取两幅图像的重叠区域平均值之差 $m_1 - m_2$ ，另一幅图减去这个差^[6]。

这个方案适合于在一幅图光照正常另一幅过亮或过暗的情况下，可以人工确定以正常的一幅为参照图，这样拼接后的光照效果较好。其缺陷在于调整范围如果较大的话，会导致灰度值超出阈值 (0,255) 而导致失真，并且无法自动判别出以哪一幅图作为参照图。

对此，我们提出了一种折中的改良算法：将调整系数设为两幅图像素平均值之差的一半，一幅图减去这个系数，另一幅图加上这个系数。这样不但使图像拼接后的效果更为柔和，并且无需指定参照图。

通过多次对比实验后我们发现，并不是所有光照条件不统一的图像都需要进行光照调整处理。对图像进行光照处理是具有一定风险的，很有可能使一些点的像素值超出阈值而导致丢失色彩失真。例如匹配物体因拍摄角度不同而造成的局部亮度不同，这种情况下用线性加权的融合方法过渡即可，不需要进行光照处理，因为光照处理加大了像素值失真的可能性。

另外，当图像本身因曝光不合适而导致部分像素点全白或全黑的情况时，要还原正常图像也是非常困难的，这里不是我们的研究范围不作讨论。

2.4 优化算法

综合上文提到的图像配准与图像融合方法，本文选取适应自然图像拼接的方法组合起来，达到优化算法的目的。

在图像配准阶段，我们用 Harris 算法提取特征点。由于 SIFT 方法所介绍的特征点描述方法开销过大，我们选用 Hu 不变矩描述特征点，得到 21 维的特征向量。然后对描述后的特征点进行配对，利用空间方向约束剔除误匹配点。

在图像融合阶段，我们采用线性加权法对图像接缝处像素值进行处理。而在光照调整方面，本文所提到的几种处理方案在处理不同情况导致的光照不统一问题时各有所长，因此我们结合了几种常见的处理方法，由用户自行比较哪一个更优。

3 研究结果及讨论

3.1 实验结果

我们选取在同一地点拍摄的两张具有重叠部分的图像。由于两张图像是在同一地点旋转一定角度后拍摄得到，因此在进行融合前线要对其进行形变处理，将它们投影到柱面上。

原始图像：

柱面投影：

之后，对其分别进行三种融合处理。第一种是直接线形加权融合，不作任何光照调整；第二种是在对图像线性加权融合后，将一幅图像的光照参数赋给另一幅，使两幅图达到光照一致；第三种是在图像线性加权融合后，将一幅图像光照参数提升光照差的一半，另一幅图像的光照参数降低光照差的一半。

拼接效果图：

第一种方法：

第二种方法：

第三种方法：

上面三种方法得到的效果图相差不大，因为两幅图像的光照程度差不多。接下来我们使用两张光照差别较大的图像来进行实验，其中一张曝光过度。

原始图像：

第一种方法：

第二种方法:

第三种方法:

由于曝光过度导致像素值的 **rgb** 值比例失衡, 导致难以精确计算出光照参数, 但就效果而言, 第三种方法得到的图像最为自然。

因此, 对于光照差别较大的两幅图像我们可以使用折中的像素值调整方法。

3.2 技术展望

随着图像拼接技术的发展, 其在科研领域、日常生活以及其他领域的作用性越来越重要。图像拼接技术已融入许多前沿科技中, 如: 空间探测、遥感图像处理、医学图像分析、视频压缩传输、虚拟现实技术以及超分辨率重构技术等。对于图像拼接, 人们也不断提出了新的要求, 自动化、实时性、立体图像拼接是未来图像拼接技术的发展方向。

4 结论

本文就图像拼接的两个主要技术——图像配准与图像融合, 结合国内外现有技术展开比较与讨论, 选择一种高效的适应自然图像的拼接组合。同时提出一种新的光照处理方法, 减缓由曝光问题导致的色彩失真现象。

参考文献

- [1] HARRIS C, STEPHENS M. A combined corner and edge detector [C]//Proc of the 4th Alvey Vision Conference. Manchester:[s.n.], 1988: 147-151
- [2] LOWE D.G Distinctive image features from scale-invariant keypoints [J].International Journal of Computer Vision, 2004, 60(2):91.110.
- [3] Jang Kyun Ho, Jung Soon Ki. Constructing cylindrical panoramic image using equidistant matching[J]. IEEElectronics Letters, 1999, 35(20):1715-1716.
- [4] Su Mingshing, Hwang Wenliang, Cheng Kuoyong. Analysis on mutlresolution mosaic images[J]. IEEE Transactions on Image Processing, 2004, 13(7):952-959.
- [5] Szeliski R, Shum H Y. Creating full view panoramic image mosaics and texture-mapped models[A].In Proceedings of SIGGRAPH 97[C], Los Angeles, California, USA, 1997: 251-258.
- [6] Reinhar Erik, Ashikhmin Michael, Bmee Gooh, et a1. Color transfer between images[J]. IEEE Computer Graphics and Applications, 2001, 21(5): 34-41.