

指导教师： 杨涛

提交时间： 2015/3/29

The task of
Digital Image Processing

数字图像处理

School of Computer Science

No: 1

姓名： 许萌签

学号： 2012302455

班号： 10011203



可容忍视差的图像拼接

张帆, 刘峰

计算机科学系

波特兰州立大学{zhangfan, fliu}@cs.pdx.edu

摘要

在进行图像拼接时, 视差处理是一项富有挑战性的工作。我们观察到在进行图像拼接时, 不必要将输入图像的重叠区域全部都完美匹配, 恰恰相反的是我们只需要将输入图像中存在的能无缝地贴合起来的局部区域匹配。本篇论文基于以上观察提出了一个处理视差的局部图像拼接方法。在处理视差和避免不良的局部变形时, 为了保证算法的灵活性, 我们采取了将单应性矩阵和保持内容的图像翘曲相结合的混合匹配模式。本文提出了一个高效的具有随机性的算法, 这个算法结合了保持内容的图像翘曲, 用于搜索图像的单应性矩阵, 同时这种算法还优化了拼接效果。我们通过质量评测预测了单应性矩阵通过寻找合理接缝并利用该接缝对图像进行合理拼接的出色程度。我们研究得出了一个接缝搜索算法, 在同时保证几何匹配和保持图像内容的条件下, 得出粗略匹配的图像, 利用粗略匹配后的图像, 这个方法便能估算出合理的拼接接缝。接下来, 我们利用最优单应性矩阵预先匹配了输入图像, 然后再利用保持内容的图像翘曲来局部地完善匹配。最终我们运用标准化的割缝算法和多波段混合算法将匹配了的图像组合在一起。通过的实验表明, 我们

的方法能够拼接具有很大视差的图像, 这在现有的算法中是很难实现的。

1. 引言

图像拼接是一个深入研究了的课题[22]。它进行的第一步是匹配输入图像。早期的方法是估算出两张图像之间的二维转换关系(一般是但应性矩阵), 然后利用图像间的转换关系来匹配[23, 3]。由于单应性矩阵不能解析处理视差, 那些早期方法要求输入图像从同一视点获取或者图像的景象是大致平面的。否则, 对于不满足要求的图像, 任何单应性矩阵都无法用于匹配这些图像, 只会导致鬼影或者图像结构破损之类的伪影。晚期的图像拼接技术, 例如割缝算法[2, 12]和共混算法[4, 17], 可以减弱这些伪影, 不过它们并不能处理显著的错位。

近期的图像的拼接方法利用了空间变化的变形算法来匹配输入图像[13, 27]。虽然空间变化的翘曲算法能比单应性矩阵更好地处理视差, 但它仍然无法很好的处理视差很大的图像。图 1 展示了一个富有挑性的例图, 图中有显著的很大的视差。注意到图 1(a)所示的输入图像中的汽车、树和烟囱的水平空间顺序。在左边的输入图像中, 烟囱在汽车和树的中间。然而在右边的图像中, 树在汽车和烟囱的中间。



(a)输入图像



(b)AutoStitch



(c)APAP[27]



(d)我们的成果

在这个示例中，需要折叠其中一张图像才能与另一张图像匹配，从根本上说，这对变形算法是一个极其困难的任务，因为这个算法不仅不能折叠图像，还会带来不良的变形。如图 1(c)所示。

在本文中，我们提出了一个可容忍视差的图像拼接方法。这个方法的提出基于以下观察：对于图像拼接来说，我们不必要把整个重叠区域完全完美地匹配，相反，我们只需要将重叠区域中存在的那些能连接起来的局部区域进行匹配。我们把这称作**局部拼接**，还

开发出了一种有效的方法用来找出能完成最优拼接的局部定位匹配。我们的局部拼接方法提供了一种同时运用单应性矩阵和保持内容的翘曲这两种方法的混合匹配模式。单应性矩阵算法能保持图像的整体结构，但是不能处理视差。与此相反，保持内容的图像翘曲算法能比单应性矩阵更好地处理视差，却无法像单应性矩阵一样保持图像的整体结构。此外，局部拼接还是更适宜处理那些匹配效果好的常见的大区域。然而，在利用单应性矩阵匹配具有大视差的图像时，匹配区域的尺寸和匹配质量通常是两个相互矛盾的追求目标。我们用单应性矩阵仅仅进行大致的图像匹配，再使用保持内容的图像翘曲来完善之前的匹配，从而解决了上述问题。

首先我们开发了一种有效的随机算法用来搜索用于不精确的局部匹配的单应性矩阵。我们从大致匹配后的图像中寻找合理接缝并使用该接缝的代价来给这个单应性矩阵打分，从而预测某个单应性矩阵能用于局部拼接的合适程度。基于接缝寻找，我们开发出了一种图像切割方法，同时满足几何匹配和保持图像内容，这种方法能仅仅通过大致匹配的图像估算出合理的接缝。一旦我们找到了最优的单应性矩阵，我们就能用这个矩阵来预对准输入图像，然后再用保持内容的图像翘曲来完善这个匹配。

本文的主要贡献是给出了一个能很好地处理大视差图像的具有有效性和鲁棒性的图像拼接方法。我们这个方法的效率体现在局部匹配，保持内容的翘曲算法和接缝分割增强了这个方法的效果，局部匹配还能探索图像的内容以及几何匹配并且找到一个最佳的局部区域从而把图像拼接在一起。正如我们的实验所示，我们的方法能拼接具有显著的大视差的图像。

2. 相关工作

图像拼接已经在计算机视觉和图像领域经过了深入研究，在[22]中可以找到很好的调查。这一节内容只给出一个简要的概述，主要着重于视差处理。

大多数现有的图像拼接方法是在两个输入图像之间估算一个 2D 变换，通常是一个单应性矩阵，然后用这个矩阵来将图像匹配[23, 3]。由于单应性矩阵不能解析处理视差，这些基于单应性的方法只能在输入图像视差较小的时候才能很好地作业。当输入图像具有较大视差时，就会有像鬼影之类的伪影出现。利用由运动估计进行局部变换能用于减弱重影假象[21]。图像合成技术也常被用来减弱伪影，例如切缝技术[12, 2, 7]和共混技术[4, 17]。然而，单独使用这些方法仍然不能处理显著视差。在我们方法的最后一步，也应用了切缝技术和共混技术。近期的双重单应性翘曲法能够拼接有视差的图片，但是它要求场景内容能够在两个平面内建模。

多视角的全景技术能很好地处理视差[20, 28, 16, 6, 19, 1, 26, 18]。这些技术需要三维重建和/或场景的密集采样。由于用户通常提供的是整个全景，这些技术要么费时，要么不能很好地处理只有稀疏集合的输入图像。以下的这个多视角全景技术的思想启发了我们的工作。这个思想是：不需要将输入图像的所有重叠区域完美地匹配。是要我们能在视觉上合理地将它们拼接起来，一个赏心悦目的全景图像就能被创建出来。

最近几周我们进行了一个相关的观测，发现拟合最好的单应性矩阵并不一定能进行最佳的图像拼接[9]。他们估算出了一组单应性矩阵，每个矩阵代表一个平面结构，运用这些单应性矩阵创建多个拼接结果，然后找出拼接质量最高的拼接结果。这个方法能成功地处理一些图像的视差，还促进了我们的工作；

然而，这种方法很慢，而且还需要创建大量的拼接结果并且进行评估。更重要的是，有时候在那些代表不同平面结构的单应性矩阵中，没有一个能在视觉上地进行合理拼接。相比之下，我们的方法能在不创建拼接结果的情况下对重叠质量进行评估，而且还更有效。此外，我们的方法继承了内容保持的图像翘曲，在单应性矩阵估算是，还放松了对图像匹配程度的要求，因此能提供更多的备选匹配方案。而且在搜索匹配区域时，同时考虑到了图像内容和几何匹配，我们的方法能更得到更适合图像拼接匹配方案。

近日，空间变化的翘曲算法已经被扩展到图像拼接领域。林等人开发了平滑变化的仿射拼接方法来处理视差[13]。萨拉戈萨等人开发出一种用来计算尽可能投影的翘曲，目的是进行全局投影同时使局部非投影偏差能计算出视差[27]。在处理有视差的图像时，这些方法表现出了优良性，这对基于单应性矩阵的方法来说是很困难的。然而，它们还是不能处理具有大视差的图像，如图 1 所示。我们的方法还运用了空间变换翘曲算法的变形算法，但仅仅用它来匹配输入图像的局部重叠区域。

3. 可容忍视差的图像拼接

我们的方法运用了一种常见的图像拼接路径。具体地讲，首先我们将输入图像匹配，然后用图像切割算法找出一个接缝把匹配后的图像拼接起来[12]，最后运用多波段混合算法创建最后的拼接结果[4]。我们的成果是一种新型的图像匹配方法，用这种方法可以匹配图片，匹配后的图片可以进行合理的图像拼接。

我们观察到不必要将图像的重叠区域全部都完美匹配。事实上，对于具有大的视差的图像，如果有可能的话，完美地匹配它们是很困难的。我们的目标是在能找到可拼接接缝的局部区域，将图像匹配。我们

采用随机算法来寻找好的定位点。具体来说，首先我们检测 SIFT（尺度不变特征转换）特征点并在两张图像间进行匹配[15]。由于我们的目标是估算出特征分布密集的局部区域的匹配效果，下一步，我们随机选择一个种子特征点并把它周围的特征点分成一组，用来估算它的匹配效果。我们估算这个匹配的拼接质量，如果这个匹配确实质量足够好，我们就停止；否则我们重复进行匹配估算和质量评估。接下来我们先讨论这个算法的部分关键部分，然后提供一个详细的算法描述。

3.1. 匹配模型选择

第一个问题就是要使用哪种匹配模式。两种受欢迎的选择是：全景二维转换法（常见的有单应性矩阵）和空间变换的翘曲法，例如保持内容的翘曲[14, 24]。大多数现有的方法是运用二维全景转换法来匹配两张图像。二维全景转换法有一个很重要的优点是它对图像进行全局性的扭曲变换，从而避免了不良的局部变形。例如单应性矩阵可以保留线条，而转换法能够保留物体的整体外形。但是它们处理视差都过于死板。对于图像拼接来说，我们认为没必要完全地将整个重叠区域都进行匹配，而这种方法仍然优先选择尽可能大的共同区域对图像进行匹配。然而，对于具有大视差的图像，二维转换法甚至单应性矩阵通常都只能对图像的很小的局部区域进行匹配。相反的是，保持内容的翘曲算法更灵活，也能更好地匹配图像，但是它常常会引入不良的局部变形。

我们的解决方案是将这两种匹配模型相结合，从而能很好地对具有大面积重叠区域的图像进行匹配，同时失真最小。给定一个种子特征点，我们的方法逐步将它的相邻特征点分组用于配合二维转换（即默认单应性矩阵）。这里我们用了一个较大的适合度阈值，

目的是获得具有尽可能多的特征点集合，虽然这使得单应性矩阵不能准确地与这些特征对应匹配。由于保持内容的图像翘曲很适合对局部变换细化而不引入显著的失真，因此降低单应性矩阵的适合度能在稍后进行的保持内容的图像翘曲时弥补。

3.2. 匹配质量评估

评价上述复杂匹配的拼接质量的一个直接的方法是先使用单应性矩阵进行图像翘曲和运用保持内容的翘曲算法。我们可以通过对比翘曲后的图像和参考图像来检验两张图像能匹配的效果有多么好。然而，这种方法不能可靠地预测能否在重叠区域找到一个很好的接缝。此外，这种方法没有考虑图像内容的拼接效果。在拼接时，突出的图像特征例如边缘，应该很好地匹配对齐，然而像天空之类的图像区域就不需要完美地匹配对齐。最后，由于无论何时当我们在随机算法内对匹配进行评估时，这个方法都要运行保持内容的翘曲，因此它很慢。

我们解决上述问题的方法如下。首先，基于图像边缘而不是整个图像，我们检验匹配的质量。然后，我们只评估单应性矩阵作用下的拼接。以下事实可以表明这种简化具有合理性：只要全景翘曲需要轻微的调整，内容感知的翘曲就非常有效。但这也带来了一个挑战：在我们的方法中，单应性矩阵被设计成稀疏矩阵因而不准确匹配两张图像。接下来，我们需要对这个匹配算法将大概匹配后的图像进行无缝拼接的效果做出预测。我们应对这一挑战的方法是从大致匹配后的图像中找到一个合理的接缝然后利用这个接缝的代价给这个匹配得分。

我们首先对输入图像向下取样，从而提高速度并且容忍小偏差。然后，我们使用 Canny 边缘检测方法计算输入图像的边缘映射[5]。边缘映射是通过低通

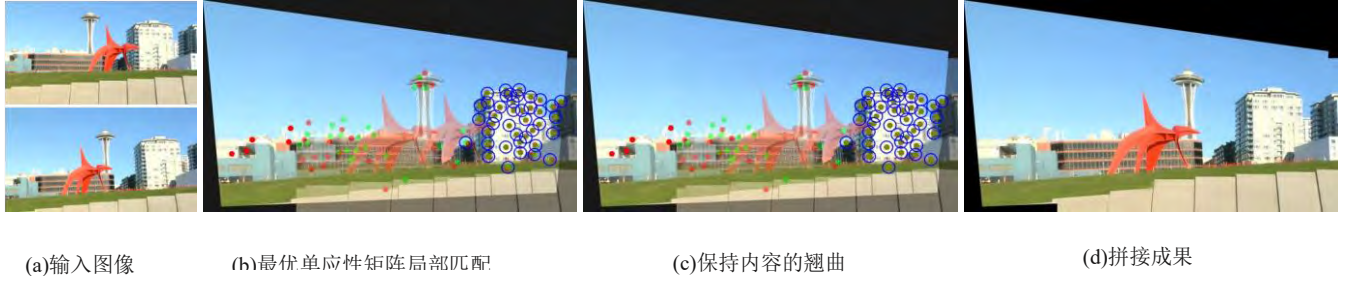


图 2. 拼接路径。请放大本图以便更好地检查(b)和(c)中的匹配结果。给定具有大视差的输入图像(a), 如 3.2 节所示, 本文介绍的方法先估算了一个能大致局部地对齐图(b)并可进行合理最优拼接的单应性矩阵。在(b)和(c)中, 我们仅用强度平均化来混合对齐后的图像来说明匹配结果。在翘曲后的图像中, 红色和灰色的点分别是翘曲后的图像和参考图像的 SIFT 特征点。当两个特征点对齐时, 就会变成橄榄绿。只有包含篮圈在内的特征点子集会被采集到, 用于稀疏地适配单个应性矩阵。然后本方法利用保持内容的翘曲来局部精炼这个匹配(3), 最后, 用缝切割和多频段共混得到最终的拼接结果(d)。

滤波得到的, 因而能容忍小偏差。我们计算翘曲的边缘映射与基准图像的边缘映射之间的差值, 得到差值映射 E_d 。一个合理的接缝应该避免遗失差值映射大的像素, 以获得无缝拼接结果。我们延伸图片切缝发现了一个找到合理接缝的方法。简单地说, 我们认为在重叠区域, 每个像素点都可以看作是一个图片节点。我们定义了两个相邻节点 s 和 t 之间的边缘代价如下:

$$e(s, t) = f_c(s) |E_d(s)| + f_c(t) |E_d(t)| \quad (1)$$

式中我们用一个匹配度置信函数 $f_c(s)$ 来加权边缘代价。计算 $f_c(s)$ 以进一步解释单应性矩阵只能大致匹配两张图像以及保持内容的翘曲能用来完善这个匹配的事实。具体来说, 如果一个局部区域具有 SIFT 特征点, 匹配很可能可以被保持内容的翘曲算法完善, 因而, 只用单应性矩阵造成的错位能够被抵消。根据 SIFT 的分布, 为了消除错位, 我们计算 $f_c(s)$ 如下:

$$f_c(s) = \frac{1}{\sum_{p_i} g(\|P_s - P_i\|) + \delta} \quad (2)$$

其中 P_i 是一个 SIFT 特征点的位置, P_s 是像素 s 的位置。 g 是一个高斯函数, 它用来传递 SIFT 特征对其周围的影像。 δ 是一个默认值 0.01 的小常数。

基于在等式 1 中定义的边缘代价, 接缝搜索问题能够和图片切割问题一样, 被公式化进而解决[12]。

只要我们获得这个接缝, 我们就能运用这个接缝的相关成本代价对匹配质量进行评分。

3.2.1. 单应性矩阵筛查

虽然有些单应性矩阵可以做到无缝拼接, 但是他们有时会严重扭曲图像, 并导致视觉上令人不快的拼接结果。我们要检测这样的矩阵并且在评估它们的匹配质量之前将它们舍弃。我们通过计算单应性矩阵 H 与它的最佳拟合相似变换之间的偏离程度来测量对图像 I 使用矩阵 H 的透视变形。记 C_i 为输入图像的四个顶点之一, \bar{C}_i 是经 H 转换得来的对应点。我们找到最佳拟合相似变换 \hat{H}_s 如下:

$$\hat{H}_s = \arg \min_{H_s} \sum C_i \| H_s C_i - \bar{C}_i \|^2, \text{ where } H_s = \begin{bmatrix} a & -b & c \\ b & a & d \end{bmatrix} \quad (3)$$

只要我们得到 \hat{H}_s , 我们把经 \square 矩阵和 \hat{H}_s 矩阵变换得到的顶点之间的距离求和。图像尺寸归一化后, 如果求得的距离和比阈值 (默认值 0.01) 大, 我们就舍弃这个单应性矩阵。

3.3 匹配算法摘要

现在我们来描述为拼接而评估匹配优劣的随机算法。

1. 在输入图像间检测和匹配 SIFT 特征并且为输入图像估算边缘映射[5]。

2. 随机采集一个种子节点，依次空间上最邻近它的点集合在一起，直到采集的特征集不再能适配于预先定义了阈值的单应性矩阵。我们为每个特征点设置一个计数值，用于判定在迭代过程中它被采集的次数。当一个特征点被采集时，我们将它的计数值加一。在每一次迭代中，为了保证被采集的种子节点是有价值的，特征点不能作为种子节点被采集过，而且它的计数值大小要比所有特征点的平均计数值低。

3. 步骤 2 中评价最拟合单应性矩阵的质量运用了步骤 3.2 所描述的算法。如果这个单应性矩阵满足预定义的质量阈值，转步骤 4。否则，如果平均计数值太低，转步骤 2；如果不是，选择迭代过程中估计的最佳单应性并转步骤 4。

4. 使用最佳的单应性矩阵预先调整图像，用在特征点集导向下的保留内容的翘曲算法进行细化调整，如 3.3 节所述。

图 2 展示了该方法的流程。给定输入图像(a)，我们的方法首先找到如图(b)所示的一个合理的局部单应性矩阵和能与之相对应的特征点子集，我们用蓝色的圆圈说明选择的部分特征点。注意到单应性并没有完全地匹配这些特征点。然后我们用内容保留的翘曲算法去细化对齐结果。如(c)中所示，那些采集到的特征点对现在能够很好地匹配。我们的方法最终将匹配好的图像合成在了一起(d)。

3.3.1. 保持内容的翘曲

各种保持内容的翘曲方法已经在很多应用中使用，诸如视频稳定行和图像[14]视频重定向[25, 24]。尽管保持内容的翘曲不总能单独用在对齐图像的整个

重叠区域，但是它适合用于一些局部微调。因此，我们用它进一步对齐经单应性矩阵向参考图像翘曲后的成果，如图 2(b)和(c)。

我们用 I , \bar{I} 和 \hat{I} 分别表示输入图像、预翘曲结果和最终翘曲结果。我们把输入图像 I 分为 $m \times n$ 个相同的网格。 I , \bar{I} 和 \hat{I} 的顶点分别用 V_i , \bar{V}_i 和 \hat{V}_i 表示。然后我们把图像翘曲问题转化为网格翘曲问题，其中 \hat{V}_i 是未知量， \bar{V}_i 从预翘曲可以知道。这个网格翘曲问题可以被定义为一个优化问题，其目的是避免明显错位的情况下将 \bar{I} 和参考图像对齐。下文我们详细描述能量项。

局域匹配项。 图像 I 和 \bar{I} 中的特征点应该移动以与其参考图像的相应位置相匹配，以便它们可以很好地对准。由于特征点 P_j 通常不与任何网格顶点重合，我们找到包含 P_j 的网格单元。我们用 \bar{P}_j 表示在 \bar{I} 中 P_j 相应的点，用在图像 \bar{I} 中相应网格的 4 个顶点的线性组合得到 P_j 。这个线性相关系数用逆双线性插值法来计算 [10]。结合输出图像 \hat{I} 中的顶点，这些系数用来计算 \hat{P}_j 。然后，我们可以定义匹配项如下。

$$E_p = \sum_{j=1}^n \left\| \sum \alpha_{j, k} \hat{V}_{j, k} - \bar{P}_j \right\|^2, \quad (4)$$

其中 n 是从匹配优化步骤（第 3.3 节）中采集的特征集的大小， $\alpha_{j, k}$ 是双线性组合系数， $\hat{V}_{j, k}$ 是一个包含 \hat{P}_j 网格单元的顶点， \bar{P}_j 是参考图像中相应的特征点。

全局匹配项。 以上的匹配项只是直接约束已选定特征点的重叠区域的翘曲。对于其他的区域，保持内容的翘曲往往会使它们失真。因为预翘曲成果 \bar{I} 已经提供了一个很好的近似值，我们的方法促进没有特征点的区域尽可能的接近预翘曲结果。因此，我们定义了下面这个全局匹配项，

$$E_g = \sum_i \tau_i \left\| \hat{V}_i - \bar{V}_i \right\|^2, \quad (5)$$

其中， \widehat{V}_i 和 \overline{V}_i 是在保持内容的翘曲结果和预翘曲结果中相应的顶点， τ_i 是一个二元值，如果在 V_i 的相邻点中没有特征点，我们令 τ_i 为 1，否则为 0。对局部对齐， τ_i 的应用提供了相当的灵活性。

平滑项。 为了在翘曲时能进一步减小局部失真，我们令预翘曲结果中的每个网格单元做相似变换。我们用 [11] 中的二元能量方程为相似变换约束编码。特别地，考虑一个三角形 $\triangle \overline{V}_1 \overline{V}_2 \overline{V}_3$ 。它的顶点 \overline{V}_1 可以由另外两个顶点表示如下：

$$\overline{V}_1 = \overline{V}_2 + \mu(\overline{V}_3 - \overline{V}_2) + \nu R(\overline{V}_3 - \overline{V}_2), \quad R = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}, \quad (6)$$

其中， μ 和 ν 是在由 \overline{V}_2 和 \overline{V}_3 定义的局部坐标系中 \overline{V}_1 的坐标。如果这个三角形做一个相似变换，在局部坐标系中它的坐标不会改变。因此，这个相似变换的公式被定义如下：

$$E_s(\widehat{V}_i) = \omega_s \left\| \widehat{V}_i - \left(\widehat{V}_2 + \mu(\widehat{V}_3 - \widehat{V}_2) + \nu R(\widehat{V}_3 - \widehat{V}_2) \right) \right\|^2, \quad (7)$$

其中 μ 和 ν 由等式 6 计算得出。我们对所有顶点的 $E_s(\widehat{V}_i)$ 求和，获得完全平滑的能量项 E_s 。这里的 ω_s 用 [14] 中的方法衡量三角形 $\triangle \overline{V}_1 \overline{V}_2 \overline{V}_3$ 的显著度值。与那些显著区域相比，我们利用这个显著度权重给比较不明显的区域分配更多的失真。

最优化。 我们结合以上的三个能量项得到如下的能量最小化问题。

$$E = E_p + \alpha E_g + \beta E_s, \quad (8)$$

其中 α 和 β 是每一项的加权值，默认值分别为 0.01 和 0.001。上述的最小化问题是二次的，而且被一个标准的稀疏线性解法解决了。得到输出网格后我们用纹理映射来呈现最终的结果。

4. 实验



(a) 输入图像



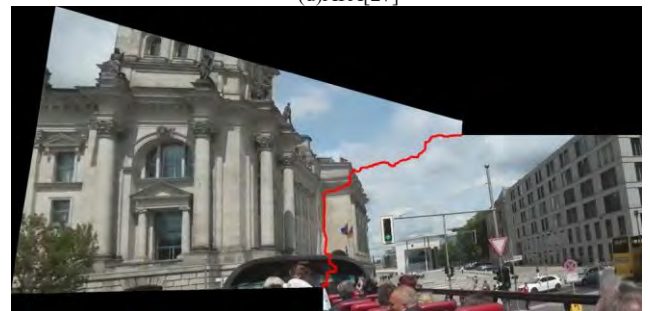
(b) AutoStitch



(c) SEAM[9]



(d) APA[27]



(e) 我们的包含接缝的成果

图3. 不同拼接方法之间的对比

我们用本文的方法对一系列具有很大大视差很有挑战性的图像进行了实验。我们将我们的方法与最先进的

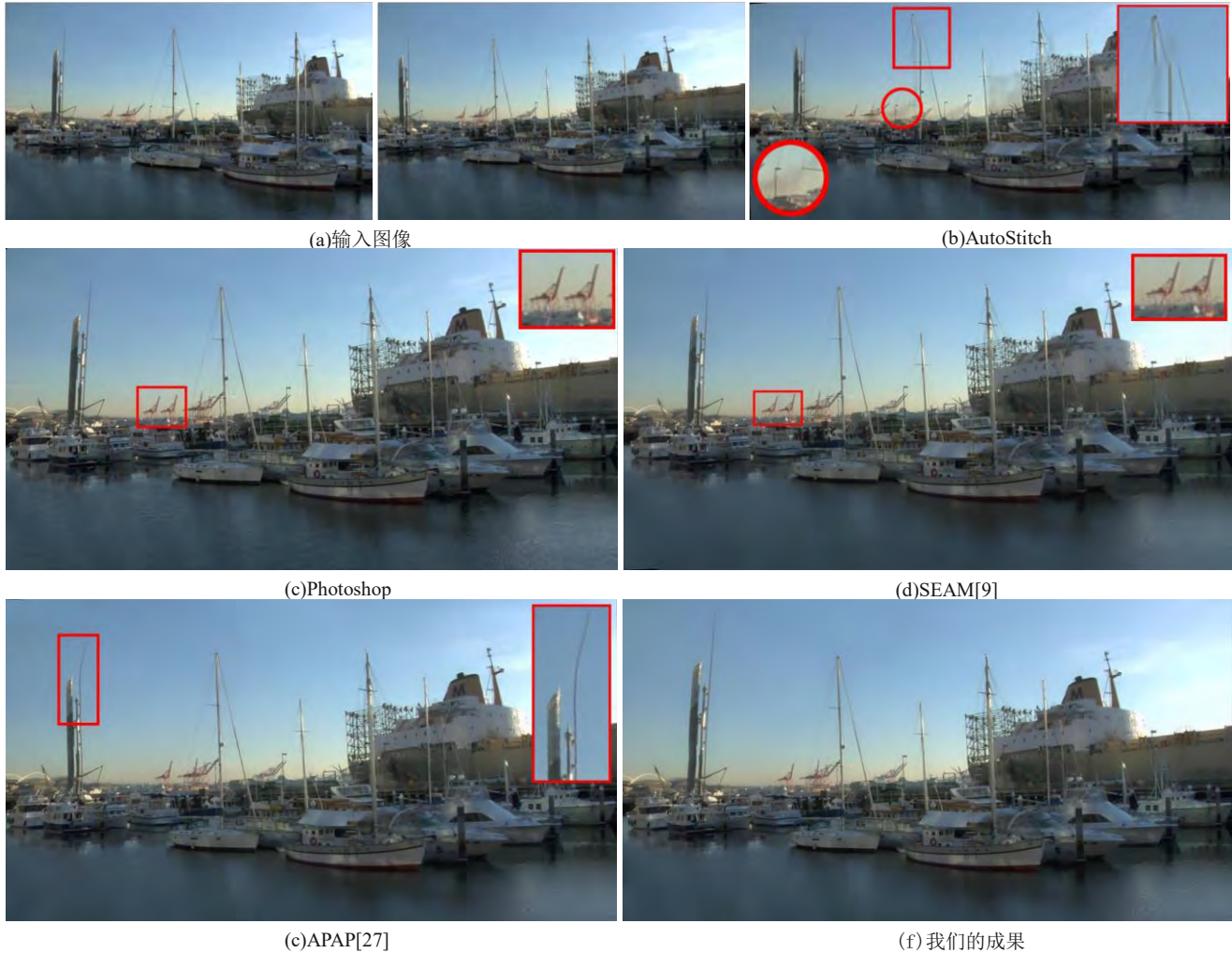
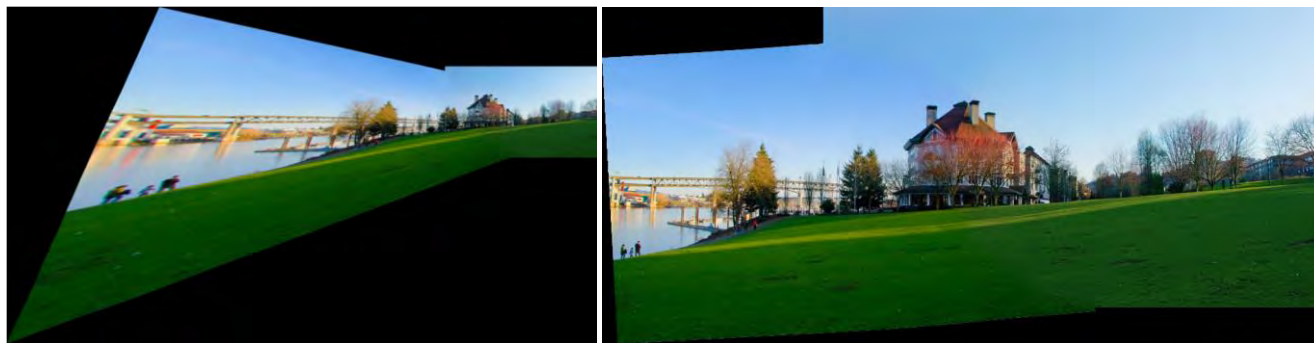


图 4.不同拼接方法之间的对比

方法做了对比，包括Photoshop， AutoStitch， 逼近投影拼接 (APAP) [27] 以及我们的接缝导向的拼接 (SEAM) 的实现[9]。对于APAP，我们用作者开源的代码。由于代码只能对齐图像，我们运用了曾经在本文的方法中使用过的接缝切割和多波段混合算法在这个APAP对齐结果中，从而得到最终的拼接结果。由于篇幅限制，这篇论文只是展示了一些具有代表性的、进行了部分截取的拼接结果。想得到更多的没有被裁切的结果和更多的中间结果，请参考项目网站¹。

图3(a)展示了两幅具有显著视差的输入图像。Photoshop没能成功地产生任何结果，AutoStitch不

能很好地用全局二维转换对齐两幅图像，因此拼接结果出现了鬼影，如图3(b)中用红色圆圈标出的部分。交通灯在最终结果中重复出现。SEAM方法没有找到一处由单应性矩阵代表的可用来进行无缝拼接的局部平面，并且如图3(c)所示，交通灯重复出现。用APAP创建产生了一种合理的拼接结果，如图3(d)所示。然而，当APAP试图在整个重叠区域对齐两幅图像时，它扭曲了图片的主要布局，比如被红色矩形标出的柱子。我们的方法通过一种允许最佳拼接的方式局部对齐输入图像，以此来处理这种具有挑战性的例子，如图3(e)所示。我们用红线标出了拼接的缝隙。



(a)单应性矩阵

(b)相似转换

图 5.单应性矩阵和相似转换对比。我们的方法在初始匹配选择二维全景转换时具有灵活性。有时候为了减少失真，我们会用常用的单应性矩阵代替相似转换。

图4(a)展示了另一个挑战性的例子。两幅输入图像有很大的视差，而且没有能让它们重复区域很好地对齐的全局转换。如图4(b)所示，AutoStitch结果产生了显著的鬼影假象。虽然共混方法可以减轻错位现象，但它会造成严重的模糊伪影，如红色圆圈所示。Photoshop和SEAM都使红圈内的部分重复出现，如图4(c)和(d)所示。APAP会使直线弯曲，如图4(e)所示。图(f)中，我们的结果中没有产生这些伪影。

4.1. 讨论

正如3.2节中描述的那样，我们的方法只需要局部地对齐输入图像，稀疏与单应性矩阵匹配。因此，相对于单应性矩阵来说，有时我们会使用更具约束性的全局转换来消除单应性矩阵中的额透视畸变。图5(a)展示了一个在算法中用单应性矩阵进行初始对齐的拼接结果，出现了显著的透视畸变的拼接结果。一旦我们用相似变换取代单应性矩阵做初始对齐，拼接结果中失真现象就会减轻，如图5(b)所示。

通过计算用我们的单应性矩阵和最佳拟合矩阵得到的转换图像的角位置之间的距离，我们也测试了出用我们的方法得到的单应性矩阵和最佳拟合矩阵之间有多大的差别。在我们课题网站中，有超过75%的共享示例的平均角位置间距大于36像素（给定图像宽

1000像素）。中等距离大约是60个像素点。这证实了我们的方法比最佳拟合方法更好地利用了单应性矩阵。

我们的方法适用于一系列具有大的视差的图例，包括报道在最近的APAP论文[27]中的所有的例子。与此同时，我们也发现了一些失败的例子。如在项目网站中所展示的那样。一个例子是输入图像有很大的视差并充分突出了结构。对于拼接，图像必须能对齐，从而至少存在一个重合区域，可以从中找到一个良好的接缝。在有巨大视差的图像中，通常没有这样可以被对其的局部区域。我们的方法探究了这样一个现象：不突出的区域往往不需要完全对齐，并且在搜索一个优良的局部区域匹配时，我们的方法应用到了这一点。但是如果一个图像有巨大的视差并且全都是突出结构，我们的方法可能无法很好地适用，因为不存在不突出的区域。

本方法采用了常规的图像拼接途径，其主要的创新之处在于它对齐图像的步骤使得最优拼接得以实现。包括最优局部单应性估计和保持内容的翘曲，在配置了英特尔i7的cpu和8GB的存储器的台式机上，这个步骤通常需要20~40秒对齐两幅宽度1000像素的图像。其他所有步骤都有现成的图像拼接技术共享。

5.结论

本文呈现了一种容忍视差的图像拼接方法。我们观察到，具有显著视差的图像通常不能对整个重叠区域很好地对齐并且不产生类似过度折叠的伪影，而且事实上也不必要为了图像拼接完全对齐这些图像的所有重叠区域。然后我们设计了一种能够局部匹配输入图像的方法，这种方法可以进行最佳拼接。我们设计出一个高效的算法，这种算法能在不进行实际拼接的情况下估算出一个匹配结果能进行无缝拼接的可能性的。我们对具有挑战性的拼接任务的实验展示出了这个方法的有效性。

致谢。图3中的图像是在知识共享署名许可下从youtube用户Lars Troen手中得到的。这项工作由美国国家科学基金会的资金IIS-1321119，CNS-1205746，和CNS-1218589支持

参考文献：

[1] A. Agarwala, M. Agrawala, M. Cohen, D. Salesin, and R. Szeliski. Photographing long scenes with multi-viewpoint panoramas. *ACM Trans. Graph.*, 25(3):853–861, 2006.

[2] A. Agarwala, M. Dontcheva, M. Agrawala, S. Drucker, A. Colburn, B. Curless, D. Salesin, and M. Cohen. Interactive digital photomontage. *ACM Trans. Graph.*, 23(3):294–302, 2004.

[3] M. Brown and D. G. Lowe. Automatic panoramic image stitching using invariant features. *Int. J. Comput. Vision*, 74(1):59–73, 2007.

[4] P. J. Burt and E. H. Adelson. A multiresolution spline with application to image mosaics. *ACM Transactions on Graphics*, 2(4):217–236, 1983.

[5] J. Canny. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 8(6):679–698, 1986.

[6] F. Dornaika and R. Chung. Mosaicking images with parallax. *Signal Processing: Image Communication*, 19(8):771–786, 2004.

[7] A. Eden, M. Uyttendaele, and R. Szeliski. Seamless image stitching of scenes with large motions and exposure differences. In *IEEE CVPR*, pages 2498–2505, 2006.

[8] J. Gao, S. J. Kim, and M. S. Brown. Constructing image panoramas using dual-homography warping. In *IEEE CVPR*, pages 49–56, 2011.

[9] J. Gao, Y. Li, T.-J. Chin, and M. S. Brown. Seam-driven image stitching. In *Eurographics 2013*, pages 45–48, 2013.

[10] P. S. Heckbert. Fundamentals of texture mapping and image warping. Master's thesis, Citeseer, 1989.

[11] T. Igarashi, T. Moscovich, and J. F. Hughes. As-rigid-as-possible shape manipulation. *ACM Transactions on Graphics*, 24(3):1134–1141, 2005.

[12] V. Kwatra, A. Schödl, I. Essa, G. Turk, and A. Bobick. Graphcut textures: image and video synthesis using graph cuts. *ACM Trans. Graph.*, 22(3):277–286, 2003.

[13] W.-Y. Lin, S. Liu, Y. Matsushita, T.-T. Ng, and L.-F. Cheong. Smoothly varying affine stitching. In *IEEE CVPR*, pages 345–352, 2011.

[14] F. Liu, M. Gleicher, H. Jin, and A. Agarwala. Contentpreserving warps for 3d video stabilization. *ACM Transactions on Graphics*, 28(3):44:1–44:9, 2009.

[15] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, 2004.

[16] S. Peleg, B. Rousso, A. Rav-Acha, and A. Zomet. Mosaicing on adaptive manifolds. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(10):1144–1154, 2000.

[17] P. Pérez, M. Gangnet, and A. Blake. Poisson image editing. *ACM Trans. Graph.*, 22(3):313–318, 2003.

[18] A. Rav-Acha, G. Engel, and S. Peleg. Minimal aspect distortion (mad) mosaicing of long scenes. *Int. J. Comput. Vision*, 78(2-3):187–206, 2008.

[19] A. Roman and H. P. Lensch. Automatic multiperspective images. In *Eurographics Symposium on Rendering*, pages 83–92, 2006.

[20] S. M. Seitz and J. Kim. Multiperspective imaging. *IEEE Computer Graphics and Applications*, 23(6):16–19, 2003.

[21] H.-Y. Shum and R. Szeliski. Construction and refinement of panoramic mosaics with global and local alignment. In *IEEE ICCV*, pages 953–956, 1998.

[22] R. Szeliski. Image alignment and stitching: a tutorial. *Found. Trends. Comput. Graph. Vis.*, 2(1):1–104, 2006.

[23] R. Szeliski and H.-Y. Shum. Creating full view panoramic image mosaics and environment maps. In *ACM SIGGRAPH*, pages 251–258, 1997.

[24] Y.-S. Wang, C.-L. Tai, O. Sorkine, and T.-Y. Lee. Optimized scale-and-stretch for image resizing. *ACM Trans. Graph.*, 27(5):118:1–118:8, 2008.

[25] L. Wolf, M. Guttman, and D. Cohen-Or. Non-homogeneous content-driven video-retargeting. In *IEEE ICCV*, 2007.

[26] J. Yu and L. McMillan. A framework for multiperspective rendering. In *EGSR*, pages 61–68, 2004.

[27] J. Zaragoza, T.-J. Chin, M. S. Brown, and D. Suter. Asprojective-as-possible image stitching with moving DLT. In *IEEE CVPR*, 2013.

[28] A. Zomet, D. Feldman, S. Peleg, and D. Weinshall. Mosaicing new views: the crossed-slits projection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(6):741–754, 2003.