

“中山大学街景” — 基于OPENCV的全景图拼接

李映真

中山大学数学与计算科学学院

数学与应用数学

September 28, 2012

Abstract

本文介绍了“中山大学街景”全景图构建和相关的OpenCV的拼接技术，并且拓展介绍了一些新的拼接技术。我们在中山大学南校区逸仙路取了93个取景点，每个取景点拍摄8个角度的照片，用于拼接全景图。我们参考了谷歌地图的设计方式，预期将在12月前完成开发网页交互界面，使得用户可以进行“虚拟游览”。

1 引言

图像拼接技术就是将数张有重叠部分的图像(可能是不同时间、不同视角或者不同传感器获得的)拼成一幅大型的无缝高分辨率图像的技术。[8] 图像拼接的关键技术包括图像配准和图像融合, 其中前者是后者的基础; 而完成图像拼接还需要在其中进行坐标变换。一般而言图像拼接主要包括以下步骤:

- 图像预处理。包括去噪、边缘提取、直方图处理等预处理、建立图像的匹配模板以及对图像进行某种变换(如傅里叶变换、小波变换等)等操作。
- 图像配准。就是选定适当的匹配策略,找出待拼接图像中的模板或特征点在参考图像中对应的位置,进而确定两幅图像之间的变换关系。
- 建立变换模型。根据模板或者图像特征之间的对应关系,建立好新的参考图像坐标系,并计算出模型中的各参数值。
- 统一坐标变换。事先建立好新的参考图像坐标系后,根据此坐标系建立数学转换模型,将待拼接图像转换到该坐标系中。
- 融合重构。将带拼接图像的重合区域进行融合得到拼接重构的平滑无缝全景图像。

在这其中,关键步骤——图像配准算法的计算量一般非常大,而且图像匹配策略的选择很大程度影响了计算复杂度以及匹配精度。某些算法还需要人工选取初始匹配点,无法适应大数据量图像的融合。

OpenCV, 全称Open Source Computer Vision Library, 是由Intel公司以C语言为基础开发, 提供多种编程语言接口的一个跨平台的开源计算机视觉库, 以BSD许可证授权发行, 可以在商业和研究领域中免费使用。 [6]OpenCV关注实时图像的处理, 实现了图像处理和计算机视觉方面的很多通用算法, 可用于开发图像处理、计算机视觉以及模式识别程序。它的应用包括但不限于图像拼接与分割、人脸、手势识别、人机交互、增强现实等。为了支持一些功能, 它还包括了一些机器学习的技术, 如Boosting、决策树算法、期望最大化算法、贝叶斯分类器、人工神经网络、支持向量机等。自1999年开发以来, OpenCV被广泛应用于日常生活中。谷歌公司开发的谷歌地图、谷歌街景等产品中都使用了部分OpenCV的接口。而且目前OpenCV已经开发了GPU模块, 可以在更高速的向量运算的基础上达到更好的运行效率。

本文主要讨论了在OpenCV中实现图像拼接的具体技术，并且介绍了谷歌街景的核心技术。我们使用OpenCV的库函数在待拼接图像中检测角点,匹配角点，并完成旋转拼接等工作。拼接完成后，我们将其投影到圆柱面上，构造出虚拟的360度全景场景。由于图像拍摄问题，部分图像无法完全匹配。我们将各个取景点按取景顺序编号，并且提供场景切换，使用户体验时有身临其境之感。

以下篇幅的组织如下：第2部分将介绍经典拼接技术以及OpenCV采用的拼接技术；第3部分将介绍模拟街景的过程，包括谷歌街景的算法以及我们的算法；第4部分将介绍我们实验的结果；我们将在第5部分给出对结果的评价以及结论。

2 拼接技术

2.1 经典拼接技术

图像拼接技术是经典的计算机视觉研究方向之一，并且被广泛应用。经典的拼接算法有频域法、灰度级法和特征匹配法等。[7]

频域法是指对图像进行快速傅立叶变换或正/余弦变换，利用频域的平移不变性以及旋转保持性，找出数张图中相似的频域部分，并进行反变换后求出旋转角以及平移量。该方法的优势是匹配精度高，运算速度快，但缺点是需要较大的重合比例，否则容易造成平移量的估计偏差上升。

基于灰度级的方法是将重叠部分的特征点找出，然后通过比较相关系数、差值方差或差值绝对值等指标确定图像配准。关于像素灰度的比较有比值比较法和块比较法，各有优劣。

基于特征的匹配方法是目前较为常见的方法，主要分为特征提取、特征匹配、变换模型构建、图像插值和重采样四个步骤。而其中的特征提取又分为角点提取、边缘提取（一阶、二阶微分算子）、形状匹配和提取（霍夫变换）、主动轮廓提取、区域、纹理描述等。

2.2 OpenCV采用的拼接技术

OpenCV从2.4版本开始提供Stitcher封装类，包括图像特征提取、特征匹配、相机校准、图像变形，图像拼接等部分。一般命令行操作只需要用户输入所需拼接的图像以及设定适当的参数，系统会自动进行以上步骤并输出拼接后的图

像。下面我们讨论了Stitcher类的关键函数、接口以及其采用的算法。[2]

2.2.1 cv::estimateTransform

该接口通过调用类内matchImages函数实现了多幅图片拼接时查找特征和匹配特征的功能，并通过调用类内estimateCameraParams函数对相机位置进行了校准。

matchImages函数依次将多幅读取的图像的大小进行标准化，并通过调用features_Finder函数寻找特征点，最后调用features_Matcher函数进行特征的匹配。matchImages函数筛选出输入图片流中可以构成全景图的图像，并输出以向量记录的匹配特征信息供下一步拼接调用。features_Finder函数的关键是检测出图像的关键点 (keypoint)，OpenCV提供了若干种方法，比如GFTT (good features to track,简单地计算了像素灰度级差，默认使用Sobel算子，可选Harris角点检测)、SIFT (scale-invariant feature transform，提取对尺度缩放、旋转、亮度变化无关的特征向量) 以及它们的金字塔算法等。这些方法各有优劣，如GFTT和FAST提取的特征点多，但是准确率相对较低；而SURF (Speeded Up Robust Feature,)MSER、ORB等提取了较精确的少量特征点。SIFT和SUFT方法计算复杂度相对较大，但SIFT提取的特征点在刚体运动、缩放、环境影响等情况下较为稳定，而ORB方法还提供了检测特征向量方向的功能。[5] features_Matcher函数则是将features_Finder函数的输出进行双向比对，找到每两张图之间相似的特征点进行对齐。

estimateCameraParams函数则是对所有输入的图片进行统计，统计出所用相机拍摄焦距的中位数，以及提供波段修正 (wave correction)功能（水平方向/垂直方向修正）。

2.2.2 cv::composePanorama

该接口实现了全景图的拼接，它首先修正了各图片的大小并且建立了图像蒙版 (image mask)，完成了原图以及蒙版的图像卷绕 (image warping)，之后确定了图像之间的缝合位置，并且对图片进行了曝光补偿处理，最后将处理好的图像融合，输出结果。融合区域的确定主要是通过比较每张图的四角的重合位置并且给出相应位置的子蒙版，而图像卷绕部分则提供了平面（矩形）卷绕、球面卷绕和柱面卷绕的选择。如果用户希望之后构建的是投影到球面上的全景图，则可以选择使用球面卷绕（投影）。

2.3 近年的研究

近年来学界关于图像拼接技术的研究主要是集中在特征选取和图像融合的改进和应用上。在特征选取算法上，由于SURF、SIFT等算法已经相对成熟，所以在特征选取方面的研究主要是集中在其应用上，如手机上的拼接软件 [4]；而在图像融合方面的研究主要是关于对不同类型的图像的缝合区域的选择。

3 模拟街景

3.1 关于谷歌街景

谷歌街景，是谷歌地图于2007年推出的一项特色服务。[3] 它是由专用街景车进行拍摄，然后将拼接后的全景图投影到球面上，并放在谷歌地图里供用户使用。目前为止谷歌已经提供了美国，法国，西班牙，意大利，荷兰，英国，澳大利亚，新西兰，日本等9个国家的135个城市的街景服务。尽管该服务引发诸多关于隐私保护争议，但其较高的仿真度极大地提高了用户体验。

3.1.1 图像采样

现在谷歌公司使用的最新街景车可以拍摄15个角度的照片，用于合成全景图。街景车同时还配备了记录拍摄位置的感应器以及定位用的GPS，以及抽取3D资料用以判断相机到建筑物距离的镭射装置等。工作人员综合考虑气候、温度、地形、光照等自然因素以及城市规划等社会因素决定拍摄时间和拍摄路线，从而获得清晰的待处理图像。

3.1.2 图像拼接

街景车每经过一个拍摄点，便同时拍摄了15张有重叠部分的照片，保证可以进行图像融合操作。每张照片的实际拍摄为止都在谷歌地图上被标示出来，并且标示了拍摄角度。谷歌构建的拼接算法中根据相机的几何位置信息，计算出哪些地方应该进行拼接。

如图1所示是谷歌街景车R7系统的采样摄像头，共计15个镜头，提供了转角360度，最大俯仰角差 X 度的全景图。[1]。我们定义摄像头之间的夹角为 α ，镜头的视场角为 β ，则两个镜头间重复的部分的视场角则为 $\beta - \alpha$ 。



Figure 1: 谷歌的15镜头摄像机，其中1个摄制正上方角度，另外14个分为两组分别摄制偏上方和偏下方的全景图片。

3.1.3 图像球极投影

谷歌街景在拼接全景图后，将图像进行球极投影，并根据用户的操作（如前进、旋转等）实时计算出投影后应显示的图像。如图2所示，设待投影点的坐标为 (x, y) ，全景图长度为 a ，宽度为 b ，则该点投影到球极坐标(图中X点坐标)为

$$(x, y) \rightarrow (r, \frac{360x}{a}, \frac{180y}{b}),$$

其中 $r = \frac{a}{2\pi}$ 。用户看到的并不是真正的球极投影图像，而是将球面贴图再次投影到平面上，则此时在新的投影面上该点（X'）的投影坐标为

$$(r, \frac{360x}{a}, \frac{180y}{b}) \rightarrow (r \sin \frac{360x}{a}, r \cos \frac{180y}{b}),$$

所以在谷歌街景的投影中的投影关系为

$$(x, y) \rightarrow (\frac{a}{2\pi} \sin \frac{360x}{a}, \frac{a}{2\pi} \cos \frac{180y}{b}).$$

3.2 我们的方法

3.2.1 图像采样

由于缺少类似谷歌街景车的设备，所以我们使用了佳能EOS 600D单反相机，统一设置了曝光时1/320sec，f/3.5，ISO640以及焦长18mm，拍摄了93个取

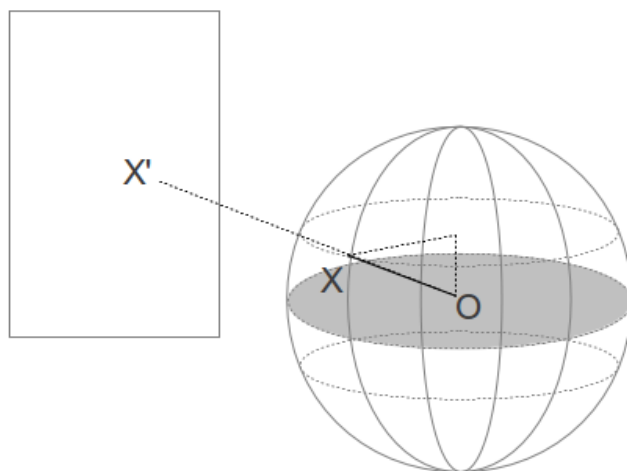


Figure 2: 球面投影法

样点784张候选图片，每个取样点拍摄了8个角度的照片。我们并没有严格控制拍摄的角度，而且对景深的控制也相较粗糙。进一步，和谷歌街景车取样不同，我们每个取样点的8张照片并不是同时拍摄的。这一不同带来的直接结果是可能在拼接后的图像中会出现两个相同的移动物体。另外由于拍摄时间的限制，在不同取样点的照片的光照条件不同，有些照片出现了较大的光强变化以及对比度下降等问题。

3.2.2 图像拼接

由于我们并未像谷歌街景车一样固定相机角度，我们无法采用根据几何位置计算拼接部分的算法。于是我们使用了OpenCV的开源拼接算法，对图像进行特征检测、特征匹配、图像投影旋转等操作。由于并未严格固定相机的仰角，我们还对拼接后的全景图进行了旋转修正和剪裁修正。

3.2.3 图像投影

由于图片采样的方式不同，我们不同于谷歌街景而采用了柱面投影法。如图3所示：设视角为 θ ，圆柱的半径为 r ，则我们已知拼接好的全景图长度并设为 l ，以及成功拼接的图片张数 n ，则有

$$l = \frac{45n}{360} \times 2\pi r$$

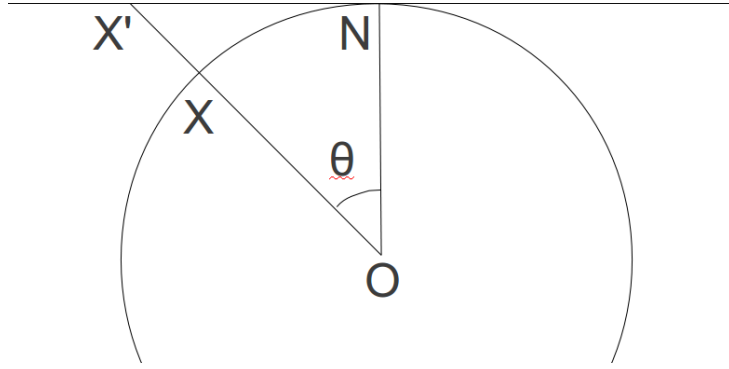


Figure 3: 柱面投影法

于是可得

$$\bar{ON} = r = \frac{4l}{n\pi}$$

从图中可以得到，在原图中以 N 点为原点作数轴，可得点 X 的坐标为

$$x = NX = \frac{\theta}{360} \times 2\pi r = \frac{\theta l}{45n}$$

X 点投影到柱面上，平面显示为 X' 点的坐标为（以 N 点为原点）

$$x' = r \tan \theta = \frac{4l}{n\pi} \tan \theta$$

于是 X 和 X' 的关系可以表示为

$$x' = \frac{4l}{n\pi} \tan \frac{45nx}{l}$$

4 实验与评价

我们拍摄了784张照片，使用OpenCV函数库中的函数进行了特征检测、特征匹配以及全景图拼接，并使用matlab进行柱面投影。由于手工拍摄相片存在曝光、角度等问题，拼接中存在匹配率较低、需要剪裁修正和旋转修正等问题。

4.1 特征检测与特征匹配

我们分别测试了SURF方法和GFTT方法，并在GFTT方法中使用了Harris角点检测得到前50个特征值最大的特征点，结果如图4：



(a) SURF-1



(b) SURF-2



(c) GFTT-1



(d) GFTT-2

Figure 4: 使用SURF/GFTT方法进行特征检测。图4(a)和4(b)是SURF算法输出的检测结果，图4(c)和4(d)是带Harris角点检测的GFTT算法输出的检测结果。图中白色圆圈的中心点既是特征点。



Figure 5: 全景拼接图，匹配置信度0.65



Figure 6: 匹配失败的角度3图像

由图中可知，该两幅图主要的特征点集中在树枝上，GFTT相较SURF算法运行速度快，但是从匹配度上看，尽管匹配点数相近，GFTT的匹配率还是略低于SURF算法。

4.1.1 全景图拼接

将该系列图片进行全景图拼接（经过旋转、裁剪修正）结果如图5：

我们发现，原本8张图像只有7张图像被成功拼接，我们考虑匹配置信度是否过大（初始设定0.65），于是我们降低了置信度(0.4)得到新的全景图依旧只有7张图像，于是我们检查了丢失的图像（角度3），发现特征点无法匹配（即图像完全不重合）。这说明图像采样是全景图拼接的决定性因素，重合率越高匹配成功率越高，相应的匹配置信度也更大。

4.1.2 柱面投影

我们将上一步得到的全景图按照柱面投影算法进行投影，由于图像丢失，我们要修正图像柱面投影所占的角度。由于只成功拼接了7张图片，我们先定位首张及末张图片中心位置，再计算出对应的轴角：设两中心距离为1，则每单位轴角（度）对应的图像长度为 $\frac{4}{7\pi}$ 。我们提供了单张图片的虚拟街景游览，可以查

看柱面投影的效果。

5 结论

本文讨论了全景图像拼接的步骤，并且讨论了google街景的拼接方式和OpenCV中实现拼接的函数和算法，并具体针对采样图像进行了实验比较。我们完成了93个取景点的所有图片的拼接工作，由于采样问题不少取景点的8角度全景图并没有完全拼接成功，我们需要针对每个取景点成功拼接的角度个数重新计算柱面投影关系。

References

- [1] D. A. et al. Google street view: Capturing the world at street level. *IEEE Computer*, 43:32–38, 2010.
- [2] fossies.org. cv::stitcher class reference. http://fossies.org/dox/OpenCV-2.4.2/classcv_1_1Stitcher.html.
- [3] Google. Cars, trikes, and more - street view - google maps. <http://maps.google.com/intl/en/help/maps/streetview/learn/cars-trikes-and-more.html>.
- [4] H. I. Koo and N. I. Cho. Feature-based image registration algorithm for image stitching applications on mobile devices. *Consumer Electronics, IEEE Transactions on*, 57(3):1303–1310, august 2011.
- [5] C. V. Talks. Comparison of the opencv’s feature detection algorithms ii. <http://computer-vision-talks.com/2011/07/comparison-of-the-opencvs-feature-detection-algorithms-ii/>.
- [6] Wikipedia. Opencv. <http://en.wikipedia.org/wiki/OpenCV>.
- [7] 徐正光等. 图像拼接方法探讨. *微计算机信息*, 22(10-3):255–257, 2006.
- [8] 百度百科. 图像拼接技术. <http://baike.baidu.com/view/2935395.htm>.