

基于 GPU 加速的实时 4K 全景视频拼接

卢嘉铭 朱 哲

(清华大学计算机科学与技术系 北京 100084)

摘 要 虚拟现实是近年来热门的新技术,而全景视频摄制是生产虚拟现实内容的一种重要方式。介绍了一个全景视频拼接系统,其能够对六路的 2K 视频实时进行拼接,生成 4K 的输出视频。设计了一个并行化的拼接流程,对整个拼接过程中的不同步骤都进行了并行化处理,并在 GPU 上进行了实现。实验结果表明,该系统能够实时拼接出高质量的 4K 全景视频。

关键词 图像拼接,图像融合,GPU 加速

中图法分类号 TP391.41 文献标识码 A DOI 10.11896/j.issn.1002-137X.2017.08.003

Real-time 4K Panoramic Video Stitching Based on GPU Acceleration

LU Jia-ming ZHU Zhe

(Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China)

Abstract Virtual reality is a popular technique in recent years, and panoramic video capture is an important way of producing the content of virtual reality. In this paper, we proposed a practical system that can stitch 4K panoramic video in real time. The inputs of our system are 6 candidate videos with 2K resolution, and output is a 4K stitched video. With the parallelization of the whole procedure, we can efficiently implement the whole system in GPU. Experimental results show that our system can produce high-quality 4K panoramic video.

Keywords Image stitching, Image blending, GPU acceleration

1 引言

随着计算机技术的进步,人们的视觉体验不断提高。黑白显示器到彩色显示器的过渡,极大地丰富了肉眼所感知的色彩的种类;从平面的观影体验到 3D 的影视作品,细节的丰富程度和逼真程度有了质的提升。如果说传统的 3D 观看是被动式的身临其境,那么虚拟现实则能够提供一种主动式的更具沉浸感的体验;用户的交互和反馈能够在场景中得到实时展现。因此,虚拟现实技术必将会改变很多行业的面貌。在新闻直播中,使用虚拟现实技术之后读者能够更真切地感受到现场的氛围;体育文艺赛事转播使用虚拟现实技术后可以给用户随时实地体验现场的便利;更为广泛的是,在飞行器驾驶培训、复杂机械操作等行业,基于虚拟现实技术可以采用仿真的方式进行培训,大大降低了成本与风险。在视频会议^[18]中,虚拟现实技术能够极大地提升会议质量。

如今,虚拟现实技术呈现爆发式发展。在显示端,索尼公司的 Morpheus、HTC 的 Vive 以及 Oculus 公司的 Oculus Rift 都是比较典型的 VR 显示头盔。在内容生产端,诺基亚公司的 Ozo、谷歌的 Jump 等全景拍摄设备也相继出现。由于眼镜和显示屏幕的距离较近,因此虚拟现实对视频内容的分辨率要求很高。而高分辨率导致了更大的计算量,因此对于全景

视频的生成算法提出了很大的挑战。

全景视频一般由多路视频拼接得到。通常情况下,这些视频由位置固定的若干个相同规格的摄像头拍摄,相邻的摄像头拍摄的视频之间有一定比例的重合,用来进行特征点的匹配,以便进行准确的注册。本文所提系统采用了 6 台摄像机的方案(见后文图 1),其中 5 台 GoPro 相机水平放置,1 台 GoPro 相机竖直向上放置。

设计的系统以 6 路 GoPro 拍摄的视频作为输入,实时地拼接出高清的全景视频。对拼接流程进行了并行化处理,在 GPU 上实现了整个系统。同时,拍摄了各类不同的场景,并在这些场景下测试了系统的性能。

本文第 2 节介绍了全景拼接的相关工作;第 3 节详细介绍了系统的实现细节;第 4 节给出了实验结果;最后总结全文。

2 相关工作

2.1 图像对齐

图像对齐方法可以分为基于像素的对齐方法^[16]和基于特征点的对齐方法^[17]。在基于像素的对齐方法中,通常两幅图像有重叠内容的区域不能相距太远,它们的精确匹配结果可以通过局部使用梯度下降的方法进行搜索得到。虽然基于

到稿日期:2016-06-22 返修日期:2016-08-17

卢嘉铭(1995-),男,博士生,主要研究方向为计算机图形学、计算机视觉;朱哲(1988-),男,博士生,主要研究方向为计算机图形学、计算机视觉,E-mail:ajex1988@gmail.com。

像素的对齐方法能够使用区域中所有像素的信息,但它受图像噪声和物体遮挡的影响比较严重,且当两幅图像在初始位置的差别很大时很难找到精确的对齐位置。而随着特征点提取技术^[19-20]的成熟,基于特征点的图像对齐方法变得更为鲁棒,能够很好地处理两幅图像之间不同程度的噪声和遮挡。

2.2 图像融合

图像融合是图像处理和图像编辑中的一个基础操作,在图像补全^[9-10]以及图像的物体编辑^[11]中都有重要的作用。多尺度图像融合^[5]是一种经典的图像融合算法,它通过对图像在不同尺度下进行拉普拉斯算子的操作,构建出图像拉普拉斯金字塔。金字塔的每一层可以看作是图像在不同频率下的分解结果。通过在金字塔的每一层进行线性加权融合,将所有层相加得到最终的融合结果。不过,该方法构建的拉普拉斯金字塔只是近似的,因此在数学上无法保证最优的融合效果。一个数学上更完美的解决方案是将图像融合定义为求解泊松方程的问题。在文献^[2]中,Perez 等人受热力学的启发,给出了图像融合的泊松方程的定义,并且通过求解一个大型的稀疏线性方程组得到最终的结果。不过该方法有两个明显的缺陷:1)在融合边界不够光滑时,泊松融合会产生明显的瑕疵;2)由于涉及了大规模线性方程组的求解,因此融合速度很慢。为了克服泊松融合在速度上的缺陷,Agarwala^[3]注意到如果在融合区域内,计算未融合时和融合后的像素差,最后的像素差的偏移场是分块平滑的,因此区域内的偏移场可以通过四叉树进行近似。使用四叉树的表示方法可以大大减少线性方程组中未知数的个数,因此可以对泊松融合进行加速。为了进一步对泊松融合进行加速,Szeliski 等人^[6]注意到如果给每一幅待融合图像都关联一个偏移场,这个偏移场用低维的样条表示,那么偏移场在整个图像的定义域都是光滑的。低维的样条可以进一步减少方程组中未知数的个数。由于求解线性方程组是一个非常耗时的过程,因此为了避免求解线性方程组,Farbman 等人^[4]通过均值坐标(MVC)的方法来对光滑的偏移场进行插值。如果待融合区域的形状固定,均值坐标可以预先进行计算。该方法的另外一个优势是可以完全并行地实现。由于 MVC 方法仍然是泊松方法的近似,因此它也会产生与泊松方法类似的瑕疵。

为了消除融合结果中的瑕疵,Jia 等人^[7]首先采用优化求解的思路在初始边界的基础上优化出一个更为光滑的边界,然后在优化后的边界上进行泊松融合。另一个避免瑕疵的思路是待融合区域计算出一个权重图,通过这个权重图使瑕疵出现在纹理比较丰富的区域,这些区域不容易被人眼察觉^[8]。然而,这些方法都非常耗时,并且不能保证最终的融合结果完全没有瑕疵。

更为详尽的图像融合算法的比较可以参见文献^[15]。

近些年也有一些相关的视频拼接研究成果,Perazzi F. 等人^[23]研究了基于非固定相机阵列的视频拼接。Lin K. 等人^[24]基于手持摄像机的输入,通过重构 3D 场景等方法进行无缝的视频拼接。

3 拼接系统

一般的全景拼接流程是先对输入的多路图像进行对齐,

使得输入的多路图像相互匹配并对应到全景图的相应位置,再对这些对齐后的图像进行合并。目前的对齐图像工具有 Hugin、PanoTools 等,合并图像工具有 Enblend、Enfuse 等。

在所设计的系统中,为了方便采集视频,使用 6 个 GoPro 相机构建了一套视频采集装置(见图 1),其中 5 个环绕于水平圆周上,另外 1 个竖直放置于顶部。对相机设置适当的分辨率可以使相邻相机拍摄的图像之间有合适的重叠部分。对于多路图像,使用 PTGui 工具(基于 PanoTools)计算对齐参数。通过获得的参数对齐图像后,计算相邻图像之间的拼缝,使用 Multi-Band Blending 方法对这些图像进行合并。图 2 展示了多路图像对齐到全景图对应位置的结果和拼缝位置。

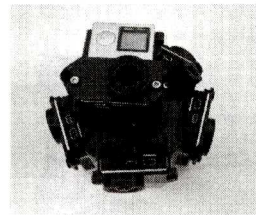


图 1 系统使用的视频采集装置

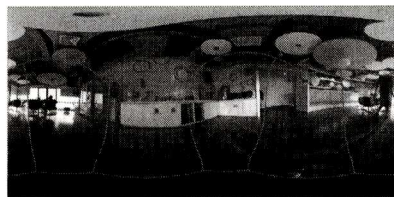


图 2 拼接的全景图示例及接缝

在输入的多路视频中,多路摄像机的相对位置是固定的,因此可以预先采样一些输入图像,预先获得对齐参数并通过这些参数预计算映射表。视频流的每一帧图像不需要再次对齐,只需要查询映射表并插值就可以得到对齐后的图像,再对这些图像进行融合。对每帧图像的插值和融合可以通过 GPU 进行加速,从而实现实时拼接。

3.1 图像对齐

输入预采样的多路图像,使用 PTGui 工具得到 PTStitcher 脚本。PTStitcher 脚本中包含各路图像的图像对齐参数。通过这些参数对每帧图像进行变换即可得到对齐后的图像。下面将详细描述该变换包含的步骤和参数。

3.1.1 畸变校正

已知光学中心成像点 P_{center} ,对于图像中的每个像素 P_{dist} ,畸变的矢径 $R_{dist} = P_{dist} - P_{center}$ 。假定校正后的矢径为 R_{corr} ,可以采用 4 次曲线对畸变进行拟合。函数如下式,其中 a, b 和 c 为前一步获得的脚本中的校正参数。

$$R_{dist} = aR_{corr}^4 + bR_{corr}^3 + cR_{corr}^2 + R_{corr}$$

3.1.2 透视变换

从脚本中可以获得对应图像 3 个轴向的旋转角 y, r 和 p ,通过透视变换将图像对齐到全景图平面上。

3.1.3 投影变换

从脚本中获得输入图像和全景图像类型,进行相应投影,例如,输入图像为矩形平面,全景图为球面。

图 3 描述了上述变换过程。对上述过程取逆则可以获得

全景图像素和对应的源图像坐标位置。

考虑到变换后得到的源图像坐标可能不位于像素点上,需要进行插值。实验采用3次多项式插值的方法,通过该坐标周围的 4×4 像素点进行插值来获得该位置对应的颜色值。

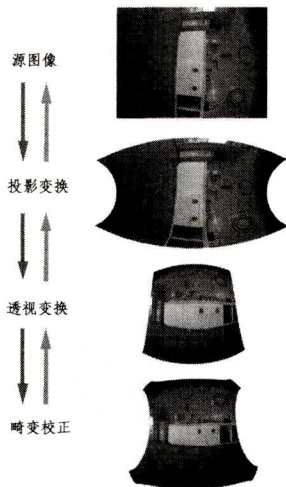


图3

图3示出了系统中逆向的图像对齐过程。深色箭头方向为逆向过程,由全景图像坐标得到对应的源图像坐标。浅色箭头方向为正向过程,由源图像得到对齐后的图像。

3.2 计算接缝

多路视频之间有重叠,需通过计算接缝进行融合。由于视频拼接中每帧计算接缝的开销过大,因此采用预先计算的简单接缝。简单接缝需要满足条件:

- 1) 较为平滑;
- 2) 与两幅重叠图像的距离相近。

Nearest Feature Transform 算法^[22]和 Graphcut^[21]算法均不满足条件1)。为了满足上述两个条件,采用如下方法。对于条件1),假定图像对齐后的图像块边界上分布有均匀且密度相等的电荷,像素点 P 相对于图像块 S 的势能函数 E_P 可以衡量像素点 P 到图像块 S 的距离。对于图像块 S ,对应的边界为 C ,有:

$$E_P = - \frac{1}{\sum_{P_i \in C} \|P_i - P\|}$$

为了满足条件2),取接缝为两幅图像块的重叠区域中势能值相等的点。图4描述了两块重叠的图像区域以及计算得到的对应接缝。

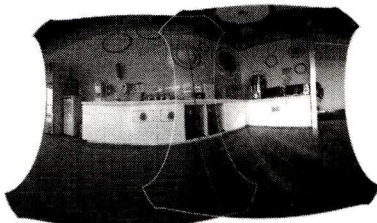


图4 箭头所指线条为对重叠的图像块计算的接缝

对于没有近距离的运动物体的场景,使用预计算的简单接缝可以取得较好的效果。如果场景中的运动物体距离摄像头较近,在物体经过接缝时,可能出现较明显的瑕疵。

3.3 多尺度图像融合方法

图像对齐后采用多尺度融合方法进行融合。该方法的基本思路:对不同的图像进行融合,融合区域的大小与进行融合的图像特征频率有关。例如,对于大片天空等具有低频特征的图像,这些大块区域的颜色变化易被肉眼察觉,融合过渡区域应该较大;而对于如车辆、树木这种具有高频特征的图像融合过渡区域应该较小,否则会造成重影。

3.3.1 拉普拉斯金字塔

为了将图像特征中的低频和低频信息分别提取出来,需要对每一路图像构建拉普拉斯金字塔。构建拉普拉斯金字塔即是对图像特征重复地进行低通滤波,将高频信息保留在当前层,将低频信息传递到下一层。对于最终得到的金字塔,其顶层包含较多的点,代表高频、小片区域的特征信息;底层包含较少的点,代表低频、大片区域的特征信息。

对于某个 N 层金字塔,假定 G_0 为输入的原图。对图像的降采样操作称为 REDUCE 操作,升采样操作称为 EXPAND 操作。对于 $0 < l < N$,计算图像对应的高斯金字塔:

$$G_{l,0} = REDUCE[G_{l-1,0}]$$

对于 $k > 0$,记:

$$G_{l,k} = EXPAND[G_{l,k-1}]$$

对于 $0 < l < N$,计算图像对应的拉普拉斯金字塔:

$$L_{l,0} = G_{l,0} - EXPAND[G_{l+1,0}] = G_{l,0} - G_{l+1,1}$$

3.3.2 高斯金字塔融合图像

每个待融合的图像可以构建对应的拉普拉斯金字塔,对不同金字塔的相同层进行融合。对于金字塔的不同层,可以通过计算每层对应的 alpha 通道值来控制融合过渡区域的大小。金字塔底层对应的 alpha 通道为计算的接缝所确定的掩膜 M_0 ,对于 $0 < l < N$,对应的高斯金字塔为:

$$M_{l,0} = REDUCE[M_{l-1,0}]$$

对于不同的图像,存在拉普拉斯金字塔 L_K 和对应的 alpha 通道 M_K 。融合后的金字塔为 L_S 。

$$L_{S,l}(i,j) = \frac{\sum_{K=0}^N L_{K,l}(i,j) \times M_{K,l}(i,j)}{\sum_{K=0}^N M_{K,l}(i,j)}$$

3.3.3 重构结果

将拉普拉斯金字塔进行叠加可以得到最终结果:

$$G_{S,0} = \sum_{l=0}^N L_{l,l}$$

图5示出了某个拼接过程中,拉普拉斯金字塔和对应的高斯金字塔中部分层次的内容以及最终拼接结果。

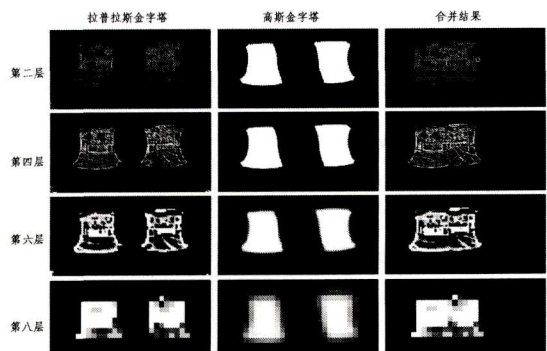


图5 融合过程中金字塔部分层次中的内容

3.4 实时化

通过预计算和并行化的方法,实现实时视频拼接。

3.4.1 预计算映射表

在一段多路视频中,多路摄像机的相对位置不发生变化。对于某个场景,可以预先进行采样得到多路图像,预计算对应的对齐参数,并通过对齐参数预计算全景图对应源图像坐标的映射关系,将其存储于映射表中。输入多路视频的每帧图像时不需要重新对齐,只需要查表即可得到对应坐标进而进行对齐,进而减少了每帧图像的映射操作开销。

3.4.2 预计算接缝

通过预计算的对齐参数可以得到多路视频对应到全景图中的图像块。对这些图像块预计算接缝得到每路图像的 alpha 通道值及对应的高斯金字塔,减少了每帧图像的合并操作开销。

3.4.3 并行化的融合操作

经过上述的预计算之后,对每帧图像需要进行的操作为:

1)通过对全景图的每个像素查表得到对应的源图像位置,插值得到颜色值;

2)对每路图像构建拉普拉斯金字塔;

3)对多路拉普拉斯金字塔进行融合;

4)对融合后的金字塔进行叠加得到全景图。

上述操作的每一步中像素之间是互相独立的,均可以使用 GPU 进行加速。步骤 2)、4)中存在对图像的升采样和降采样操作,为了方便升、降采样操作的并行,可以将全景图扩展为 $2^N \times 2^N$ 大小,其中 N 为正整数。为了不改变原图包含的信息,对全景图的空白部分采用镜像反射的方法进行扩展。

4 实验

4.1 数据集

实验使用的采集视频设备如图 1 所示。为了对系统进行测试分析,采集了各种环境下的多组多路视频作为数据集。每路视频的帧速率为 30fps,分辨率为 1920×1440 。

4.2 实验结果

实验中,将本文的实现与 OpenCV 中的 Multi-Band-Blender 类进行比较。由于拼接算法的速度只与全景图的大小有关,因此测试了在同一场景下生成不同大小全景图的 3 种实现的时间开销。实验中使用的 CPU 为 Intel Xeon E5-2620 2.0GHz,内存为 32GB, GPU 为 NVIDIA GeForce GTX 970,显存大小为 4GB,内存与显存的交换速率为 4GB/s。拼接系统的实现采用 C++ 语言, GPU 部分使用 CUDA。

比较的 OpenCV 版本为 2.4.11。记录的时间开销不包括初始化以及预计算开销。实验结果如表 1—表 3 所列。表中的交换时间表示 GPU 版本中内存与显存之间的数据交换开销。在实际的系统中,数据交换开销可以通过硬件方法来减少或消除,如直接将输入视频流从设备传输到显存中。图 6—图 8 示出了一些拼接结果。

表 1 4096×2048 分辨率下 3 种实现的时间开销/ms

实现	交换时间	计算时间	总时间
Ours(GPU)	89	44	133
OpenCV(GPU)	—	—	3800
OpenCV(CPU)	—	—	4700

表 2 2048×1024 分辨率下 3 种实现的时间开销/ms

实现	交换时间	计算时间	总时间
Ours(GPU)	48	14	62
OpenCV(GPU)	—	—	335
OpenCV(CPU)	—	—	280

表 3 1024×512 分辨率下 3 种实现的时间开销/ms

实现	交换时间	计算时间	总时间
Ours(GPU)	40	6	46
OpenCV(GPU)	—	—	140
OpenCV(CPU)	—	—	75



图 6 户外光照充足环境下的拼接结果



图 7 户外夜晚环境下的拼接结果



图 8 室内场景下的拼接结果

4.3 实验分析

从表 1—表 3 中可以看出,设计的系统极大地减少了计算时间开销。计算过程中的主要时间开销为内存与显存之间的数据交换开销。使用更高数据交换速率的硬件或通过硬件手段将数据流直接载入到显存中可以减少数据交换开销,从而实现实时视频拼接。在一些分辨率下,OpenCV GPU 版本的时间开销反而比 CPU 版本的大,原因是在图像分辨率较小时,显存与内存之间的数据交换开销相对较大,反而降低了效率。

结束语 本文实现了一个实时的 4K 全景拼接系统。根据相机位置固定这一先验条件,预计算出了一个拼接的模板,然后采用并行化后的多尺度图像融合方法进行融合。在 GPU 上实现了整个拼接和融合的流程,在 4K 分辨率的输出下能够达到实时的效果。

(下转第 26 页)

- context learning for visual tracking[C]//European Conference on Computer Vision. 2012:716-729.
- [13] LUO H L, SHAN S Y, KONG F S. Mean Shift Tracking Based on Ensemble Multiple Instance Learning[J]. Journal of Computer-Aided Design and Computer Graphics, 2015, 27(2): 226-237. (in Chinese)
罗会兰, 单顺勇, 孔繁胜. 基于集成多示例学习的 Mean Shift 跟踪算法[J]. 计算机辅助设计与图形学学报, 2015, 27(2): 226-237.
- [14] MA L, CHANG F L, QIAO Y Z. Target Tracking Based on Mean Shift Algorithm and Particle Filtering Algorithm[J]. Pattern Recognition and Artificial Intelligence, 2006, 19(6): 787-793. (in Chinese)
马丽, 常发亮, 乔谊正. 基于均值漂移算法和粒子滤波算法的目标跟踪[J]. 模式识别与人工智能, 2006, 19(6): 787-793.
- [15] LIU Q, HOU J H, MOU H J, et al. Object detection and tracking combining generative and discriminative model[J]. Journal of Image and Graphics, 2013, 18(10): 1293-1301. (in Chinese)
刘倩, 侯建华, 牟海军, 等. 联合生成与判别模型的目标检测与跟踪[J]. 中国图象图形学报, 2013, 18(10): 1293-1301.
- [16] MA B, SHEN J, LIU Y, et al. Visual tracking using strong classifier and structural local sparse descriptors[J]. IEEE Trans. on Multimedia, 2015, 17(10): 1818-1828.
- [17] MA B, HU H, SHEN J, et al. Linearization to Nonlinear Learning for Visual Tracking[C]//IEEE ICCV. 2015:4400-4407.

(上接第 21 页)

未来计划对立体双目的全景视频进行实时拼接, 同时利用双目信息优化拼缝的处理。

参 考 文 献

- [1] SZELISKI R. Image alignment and stitching: A tutorial[J]. Foundations & Trends in Computer Graphics & Vision, 2004, 2(1): 1-104.
- [2] PEREZ P, GANGNET M, BLAKE A. Poisson image editing[J]. ACM Transactions on Graphics, 2003, 22(3): 313-318.
- [3] AGARWALA A. Efficient gradient-domain compositing using quad-trees[J]. ACM Transactions on Graphics, 2007, 26(3): 94.
- [4] FARBMAN Z, HOER G, LIPMAN Y, et al. Coordinates for instant image cloning[J]. ACM Transactions on Graphics, 2009, 28(3): 67.
- [5] BURT P J, ADELSON E H. A multiresolution spline with application to image mosaics[J]. ACM Transactions on Graphics, 1983, 2(4): 217-236.
- [6] SZELISKI R, UYTENDAELE M, STEEDLY D. Fast poisson blending using multi-splines[C]//International Conference on Computational Photography (ICCP 11). IEEE, 2011: 1-8.
- [7] JIA J, SUN J, TANG C K, et al. Drag-and-drop pasting[J]. ACM Transactions on Graphics, 2006, 25(3): 631-637.
- [8] TAO M W, JOHNSON M K, PARIS S. Error-tolerant image compositing[J]. International Journal of Computer Vision, 2013, 103(2): 178-189.
- [9] ZHU Z, HUANG H Z, TAN Z P, et al. Faithful Completion of Images of Scenic Landmarks Using Internet Images[J]. IEEE Transactions on Visualization and Computer Graphics, 2015, 22(8): 1945-1958.
- [10] ZHU Z, MARTIN R, HU S M. Panorama Completion for Street Views[J]. Computational Visual Media, 2015, 1(1): 49-57.
- [11] CHEN T, ZHU Z, SHARMI A, et al. 3-Sweep: Extracting Editable Objects from a Single Photo[J]. ACM Transactions on Graphics (TOG), Siggraph Asia, 2013, 32(6): 1-10.
- [12] BATRA V, KILGARD M, KUMAR H, et al. Accelerating Vector Graphics Rendering using the Graphics Hardware Pipeline[J]. ACM Trans. on Graph., 2015, 34(4): 146.
- [13] LOOP C, BLINN J. Resolution independent curve rendering using programmable graphics hardware[J]. ACM Trans. on Graph., 2005, 24(3): 1000-1009.
- [14] HUANG M C, LIU F, WU E H. A GPU-based matting Laplacian solver for high resolution image matting[J]. The Visual Computer, 2010, 26(6): 943-950.
- [15] ZHU Z, LU J M, ZHANG S H, et al. A Comparative Study of Blending Algorithms for Realtime Panoramic Video Stitching[J]. arXiv preprint, arXiv: 1606. 00103, 2016.
- [16] IRANI M, ANANDAN P. About direct methods[C]//International Workshop on Vision Algorithms. Springer, Kerkyra, Greece, 1999: 267-277.
- [17] BROWN M, SZELISKI R, WINDER S. Multi-imagematching using multi-scale oriented patches[C]//IEEE Conference on Computer Vision and Pattern Recognition (CVPR ' 2005). 2005: 510-517.
- [18] ZHU Z, MARTIN R R, PEPPERELL R, et al. 3D Modeling and Motion Parallax for Improved Video Conferencing[J]. Computational Visual Media, 2016, 2(2): 131-142.
- [19] LOWE D G. Distinctive image features from scale-invariant keypoints[J]. International Journal of Computer Vision, 2004, 60(2): 91-110.
- [20] MATAS J, CHUM O, URBAN M, et al. Robust wide baseline stereo from maximally stable extremal regions[J]. Image and Vision Computing, 2004, 22(10): 761-767.
- [21] KWATRA V, SCHÖDL A, ESSA I, et al. Graphcut textures: image and video synthesis using graph cuts[J]. ACM Trans. on Graph., 2003, 22(3): 277-286.
- [22] ALSUWAIYEL M, GAVRILOVA M. On the Distance Transform of Binary Images[J]. Methods in Molecular Biology, 2000, 469(4): 87-102.
- [23] PERAZZI F, SORKINE-HORNUNG A, ZIMMER H, et al. Panoramic Video from Unstructured Camera Arrays[C]//Computer Graphics Forum (Proceedings of Eurographics 2015). 2015: 57-68.
- [24] LIN K, LIU S, CHEONG L, et al. Seamless Video Stitching with Hand-held Camera Inputs[J]. Computer Graphics Forum, 2016, 35(2): 479-487.