

一种自适应四叉树分块的图像拼接算法

王元炜¹, 郁梅^{1,2*}, 姜浩¹, 邵华¹

(1. 宁波大学 信息科学与工程学院, 浙江 宁波 315211;

2. 南京大学 计算机软件新技术国家重点实验室, 江苏 南京 210023)

摘要: 图像拼接技术是计算机图形学的研究热点, 具有代表性的 As-Projective-As-Possible (APAP) 算法采用均匀分块法来实现图像拼接的局部映射, 没有充分考虑图像不同区域之间的差异性. 为了兼顾图像拼接的精度和效率, 提出了一种基于特征点分布的图像映射算法, 利用四叉树对图像分块来实现图像拼接的局部映射. 首先, 设定四叉树的最大深度, 并根据图像匹配点的总数估计出每个图像块的匹配点阈值; 然后, 利用最大深度和匹配点阈值对图像做自适应四叉树分块; 最后, 对不同的图像块使用移动线性变换估计出单应性矩阵, 依据估计的结果完成图像拼接. 实验结果表明, 提出的算法提高了图像拼接的精度, 其速度远大于 APAP 算法, 并且提升了拼接图像的质量.

关键词: 图像拼接; 图像映射; 投影变换; 四叉树

中图分类号: TN911.73

文献标志码: A

文章编号: 1001-5132 (2018) 04-0052-07

图像拼接是把一组图像拼接成一幅大视场图像的处理过程, 在全景成像、场景理解、摄影测量以及遥感成像等领域得到有效应用^[1]. 图像拼接过程通常包含了预处理、配准、映射和融合等步骤^[2]. 图像映射是根据图像配准的结果, 利用变换矩阵估计图像间的空间映射关系, 从而实现图像的无缝拼接. 按照变换矩阵的作用区域, 图像映射分为全局映射和局部映射两种映射模型. 全局映射指的是整幅图像使用同一个变换矩阵的方法. Brown 等^[3]提出了 AutoStitch 算法, 该算法只使用一个单应性矩阵来完成图像的拼接. 全局映射算法要求图像的重叠区域近似在同一个平面上, 并且拍摄图像时的相机光心近似重合, 只适合拼接视差很小或没有视差的图像.

为了克服全局映射模型的缺陷, 人们提出了局部映射模型, 局部映射是对图像的不同区域使用不同变换矩阵的方法. Gao 等^[4]提出了 DHW (Dual-Homography Warping) 算法, 将图像划分为背景区域和前景区域, 用 2 个单应性矩阵分别对应于

前景区域和背景区域, 提高了图像拼接算法的精度. Lin 等^[5]提出了 SVA (Smoothly Varying Affine) 算法, 用多个仿射变换矩阵实现图像的分块映射, 拼接精度进一步提高. Zaragoza 等^[6]提出了 APAP 算法, 将图像划分成均匀密集的网格块, 使用移动线性变换 (Moving Direct Linear Transformation, MDLT) 来估计每个网格块的单应性矩阵. APAP 算法能够精确地进行拼接, 可以较好地拼接视差较大的图像.

APAP 算法的精度和效率问题也非常值得研究. 本文针对 APAP 算法的均匀分块引起的时间复杂度过高的问题展开研究, 提出了改进算法. 实验表明, 在保证精度的同时, 本文提出的不均匀分块的图像拼接算法要比 APAP 算法的效率高出得多.

1 基于自适应四叉树分块的图像拼接算法

针对 APAP 算法的分块过多导致速度过慢的问题, 提出一种自适应四叉树分块的图像拼接算法, 该算法主要包含图像配准、基于匹配点分布的

收稿日期: 2017-11-13.

宁波大学学报(理工版)网址: <http://journal.lg.nbu.edu.cn/>

基金项目: 国家自然科学基金(61671258); 浙江省自然科学基金(LY15F010005).

第一作者: 王元炜(1993-), 男, 湖北十堰人, 在读硕士研究生, 主要研究方向: 多媒体信号处理与通信. E-mail: wyw1038194231@qq.com

*通信作者: 郁梅(1968-), 女, 江苏无锡人, 博导/教授, 主要研究方向: 多媒体信号处理与通信. E-mail: yumei2@126.com

目标图像四叉树分块、目标图像映射和插值、图像融合等步骤。

图像配准包含特征点提取和特征点匹配 2 个步骤。特征提取算法种类繁多, 已有的特征提取算法不少于 50 种^[7], 其中 SIFT 算法^[8]和 MSER 算法表现出极好的性能^[9]。SIFT 算法具有鲁棒性好, 缩放、旋转、平移不变性以及特征点丰富等特点。在特征点提取后, 需要进行特征点匹配, 这一过程容易产生误配点, 可以通过随机抽样一致性算法^[10]来剔除错误的匹配点对, 该算法鲁棒性好, 在提高精度的同时减少了匹配点对的损失。

在图像映射的过程中, 为了提高精度, APAP 算法将图像分割成密集且均匀的矩形块(默认分成 100×100 个块), 对每个矩形块分别进行单应性矩阵估计, 而单应性矩阵估计复杂且耗时, 这在很大程度上影响了图像拼接的效率。其改进思路就是尽可能地保留精度的同时减少图像的分块。

单应性矩阵的估计是根据特征匹配点对来进行的。当匹配点数量多的时候, 为了提高精度, 有必要对图像密集分块; 反之, 当图像的匹配点很少的时候, 即使密集分块也不能提高精度。此外, 匹配点密集的区域一般纹理复杂、细节信息丰富, 匹配点稀疏的区域一般纹理简单、细节信息较少, 应该对特征点密集的区域做精确的单应性估计, 而对特征点稀疏的区域做较粗略的估计。因此, 图像分块可以依据匹配点的密度来确定, 在匹配点密度大的区域密集分块, 密度小的区域稀疏分块。

采用四叉树能够实现上述的图像分块思想, 为此, 本文所提出算法描述如下: 首先设定一个四叉树的最大深度 D_e 限制四叉树的最大划分深度; 然后设定每个分块内部的匹配点阈值 T_h , 如果当前块内部的匹配点大于阈值 T_h , 就对当前块进行四叉树划分, 否则就不再划分, 认为当前块已经划分完成; 根据这个划分规则对图像进行四叉树递归划分, 直到所有图像块内部的匹配点数量小于阈值 T_h 或者该分块对应的四叉树深度达到 D_e 时停止划分, 完成图像的分块操作, 算法流程如图 1 所示。一幅已知图像的分块是由 T_h 和 D_e 确定的, 对匹配点分布比较密集的区域, 由于每个图像块内部的匹配点不超过 T_h , 该区域的分块就会比较密集; 而在某些匹配点过于密集的区域, 仅仅靠

T_h 来自适应划分图像, 又会导致该区域的分块过密, 因此引入参数 D_e 来限制图像的最大划分深度, 防止图像的某些区域分块过密。

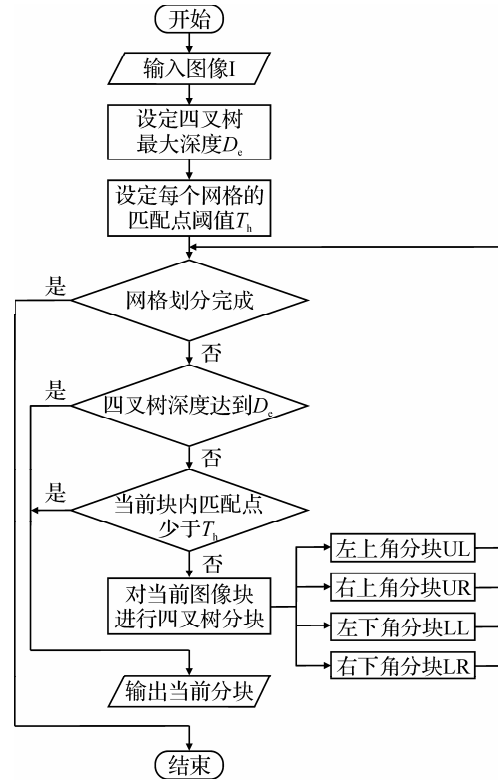


图1 图像自适应四叉树分块的算法流程

若采用一棵深度是 D_e 的四叉树对图像进行分块, 则这棵树最多可划分的图像块数 S 为:

$$S = 4^{D_e-1}. \quad (1)$$

D_e 只影响极少数的小图像块, 所以 D_e 的取值只要在一定范围即可, 一般 D_e 取 8~10 都能很好满足要求。每个分块的匹配点阈值 T_h 可以根据匹配点数量来设置, 当匹配点数量很多的时候, 阈值 T_h 应该适当取较大值, 反之当匹配点数量很少的时候, 阈值 T_h 应该取较小值。研究表明, 对于匹配点数在几百到几千之间的普通图像来说, T_h 可以设置为 1~20, 具体可由实验确定。

图像的空间几何坐标变换是建立一幅图像坐标 (x, y) 与另一幅图像坐标 (x', y') 间的变换关系^[11]。常用的变换有相似变换、仿射变换和投影变换。为了提高精度, 这里选用投影变换来完成图像映射。对目标图像每一个分块使用一个投影变换矩阵, 将目标图像映射到参考图像平面上去, 以便于后续的图像融合。

假设坐标 $\mathbf{x}' = [x', y']^T$ 和坐标 $\mathbf{x} = [x, y]^T$ 分别是参考图像 I' 和目标图像 I 的一对匹配点, 这两个点之间的单应性关系可表示为:

$$\tilde{\mathbf{x}}' \sim \mathbf{H}\tilde{\mathbf{x}}, \quad (2)$$

$$\mathbf{H} = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix}, \quad (3)$$

式中, $\tilde{\mathbf{x}}$ 和 $\tilde{\mathbf{x}}'$ 分别为 $\mathbf{x}$ 和 $\mathbf{x}'$ 对应的齐次坐标, $\mathbf{H}$ 是 3×3 的单应性矩阵, $\sim$ 表示向量按比例相等. 将单应性矩阵 $\mathbf{H}$ 表示成式(3)的形式后, 非齐次坐标下的匹配点 $\mathbf{x}$ 和 $\mathbf{x}'$ 之间的对应关系表示成:

$$\begin{aligned} x' &= \frac{h_{11}x + h_{12}y + h_{13}}{h_{31}x + h_{32}y + h_{33}}, \\ y' &= \frac{h_{21}x + h_{22}y + h_{23}}{h_{31}x + h_{32}y + h_{33}}, \end{aligned} \quad (4)$$

式中, x', y', x, y 是坐标 $\mathbf{x}' = [x', y']^T$ 和坐标 $\mathbf{x} = [x, y]^T$ 的坐标分量.

将式(2)变形成 $0_{3 \times 1} = \tilde{\mathbf{x}}' \times \mathbf{H}\tilde{\mathbf{x}}$, 计算得到:

$$\begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & -x-y-1 & xy' & yy' & y' \\ x & y & 1 & 0 & 0 & 0 & -xx'-yx'-x' \\ -xy'-yy'-y' & xx' & yx' & x' & 0 & 0 & 0 \end{pmatrix} \mathbf{h}, \quad (5)$$

式中,

$$\mathbf{h} = (h_{11}, h_{12}, h_{13}, h_{21}, h_{22}, h_{23}, h_{31}, h_{32}, h_{33})^T. \quad (6)$$

在估计单应性矩阵 $\mathbf{H}$ 的时候, 可以尽可能利用更多的匹配点信息来减少误差. 式(5)中等式右边 3×9 的系数矩阵中只有 2 行是独立的, 选择前 2 行组成独立系数矩阵 $\mathbf{A}_i$, 将所有的匹配点考虑进来, 就可以组成一个 $2N \times 9$ 的系数矩阵 $\mathbf{A}$, 根据最小二乘法的思想, 对 $\mathbf{h}$ 的求解可以表示成:

$$\hat{\mathbf{h}} = \arg \min_{\mathbf{h}} \sum_{i=1}^N \|\mathbf{A}_i \mathbf{h}\|^2 = \|\mathbf{A} \mathbf{h}\|^2, \quad (7)$$

式中, $\mathbf{h}$ 是经过归一化后的单位向量; N 表示匹配点的总对数; $\mathbf{A}_i$ 表示第 i 对匹配点对应的独立系数矩阵. 可以通过奇异值分解来完成对 $\mathbf{h}$ 的计算, $\mathbf{A}$ 的最小奇异值所对应的右奇异向量就是所求的结果. 将向量 $\hat{\mathbf{h}}$ 的各个元素按照式(3)的排列顺序就得到了单应性矩阵 $\mathbf{H}$ 的估计值.

实现目标图像局部映射的时候, 可以将目标图像划分成多个图像块, 对每个图像块利用上述估计方法来计算该图像块对应的单应性矩阵. 但

是, 研究表明, 单独对每个图像块求取单应性矩阵将会使拼接图像的整体视觉效果变得很差.

为了兼顾图像映射的精度和拼接图像的整体视觉效果, 采用 MDLT 算法^[12]来实现图像映射. MDLT 算法的思想类似于移动最小二乘法(Moving Least-Squares, MLS)^[13], 将整幅图像的所有匹配点都考虑进来, 根据当前图像块的位置, 对图像的所有匹配点分配相应权重, 从而完成该图像块的单应性矩阵估计. 匹配点权重是由匹配点到图像块中心点之间的距离决定的, 距离越小权重越大, Zaragoza 等^[12]提出的 MDLT 算法选用高斯函数来计算权重, 其计算方法如下:

$$\omega_i^j = \max \left(\exp \left\{ -\frac{\|\mathbf{x}_* - \mathbf{x}_i\|^2}{\sigma^2} \right\}, \gamma \right), \quad (8)$$

式中, ω_i^j 表示第 i 对匹配点在当前图像块的权重; $\mathbf{x}_*$ 表示当前图像块的中心坐标; $\mathbf{x}_i$ 表示目标图像的第 i 对匹配点坐标; σ 为尺度因子; γ 为最小权重值, 用以防止某些远离当前图像块的匹配点权重过小.

Lin 等^[14]提出了另外一种计算权重的方法, 使用学生 t -分布函数来代替高斯分布函数, 因为学生 t -分布函数比高斯分布函数更加平滑, 可以省略掉最小权重值 γ , 从而具有更好的鲁棒性. 计算式^[14]如下:

$$\omega_i^j = \left(1 + \frac{\|\mathbf{x}_* - \mathbf{x}_i\|^2}{\nu} \right)^{-\frac{\nu+1}{2}}, \quad (9)$$

式中, ν 是自由度. 与高斯函数类似, 应该在上式中加入尺度因子 σ :

$$\omega_i^j = \left(1 + \frac{\|\mathbf{x}_* - \mathbf{x}_i\|^2}{\nu \sigma^2} \right)^{-\frac{\nu+1}{2}}. \quad (10)$$

学生 t -分布函数与高斯分布函数非常相近, 实际实验中, 两种计算方法所得到的实验结果相差不大, 但是高斯分布函数的计算速度要比学生 t -分布函数的计算速度快很多, 这里仍然采用高斯函数来计算权重.

图像映射分为正向映射和反向映射, 为了防止映射后的图像出现空洞, 实际经常采用反向映射的方法^[11]. 在图像反向映射的过程中, 利用单应性矩阵计算出的图像位置可能不是整数, 必须利

用其周围整数像素位置通过插值技术得到. 常用的插值算法有最邻近插值法、双线性插值法和双三次插值法, 这里使用最邻近插值法.

目标图像完成投影变换后, 与参考图像近似在同一个平面上, 但与参考图像之间仍然存在明暗差异、拼接缝、鬼影等问题, 这就需要图像融合技术. 本文主要研究的是图像拼接中的图像映射问题, 为了便于跟 APAP 等算法做比较分析, 这里直接采用简单的直接平均法^[15]做图像的融合.

2 实验结果与分析

为了说明所提出算法的有效性, 将其与经典全局映射算法和 APAP 算法^[6]进行比较. 实验硬件设施为台式电脑, CPU 为 Intel i3 3.4 GHz, 内存 4 GB, 操作系统为 Windows 7, 编程开发平台为 MATLAB R2014a 和 Visual Studio 2013. 为了保证实验的客观性, 这里的全局映射算法和 APAP 算法都使用 APAP 算法^[6]提供的代码做测试. 实验分为定性分析和定量分析 2 个部分: 定性分析以 APAP 算法提供的 railtracks 图像对为例, 对比分析 3 种算法的主观视觉效果; 定量分析以 APAP、DHW^[4]以及 AANAP^[14]中的几组图像为例, 主要对比 3 种图像映射算法的误差和运行时间.

图 2 是对 railtracks 图像对^[6]使用全局映射算法拼接的效果, 区域 a 和区域 b 分别对应于已拼接图像的黑色边框和白色边框内的局部放大图像.



图 2 全局拼接算法效果

图 3 和图 4 的图像放置方法与图 2 相同. 图 3 是 APAP 算法的拼接效果, 网格分块采用 APAP 算法的默认分块 100×100 , 即 10 000 块. 图 4 是本文算法的拼接效果, 其中 D_c 为 8, T_h 为 9, 图像的分块总数为 835 块.



图 3 APAP 算法拼接效果



图 4 本文算法的拼接效果

图 2 中, 区域 a 和区域 b 都出现了很严重的重影现象, 视觉效果很差, 这是因为全局映射算法的精度很低, 导致在拼接图像的重叠区域出现了图像的错位. 图 3 中, 局部区域 a 和局部区域 b 明显比图 2 的效果好得多, 这是因为 APAP 算法使用了局部映射模型, 使映射模型的精度得到了有效提

升,有效消除了映射误差所引起的鬼影现象,但是仔细观察可以发现图3的区域a和区域b中,物体的轮廓部分出现了比较明显的模糊,这说明APAP算法的精度还不够高,仍然存在误差所引起的鬼影现象.图4中,区域a和区域b的主观效果与图像3基本相同,但是图4中物体的边缘模糊程度要比图3弱,说明本文提出的图像映射模型的精度要比APAP算法高.

图5是原始的两幅待拼接图像的配准图,上面是参考图像,下面为目标图像,其中黑色小圆环表示成功匹配的特征点.可以发现参考图像和目标图像的匹配点分布是很不均匀的,多数待拼接图像的匹配点都有这样的分布规律.



(a)参考图像



(b)目标图像

图5 原始图像的特征提取和特征点匹配效果图

图6是APAP算法中的均匀分块与本文提出的基于匹配点密度的自适应四叉树分块算法的示意图.为了便于观察,这里的APAP算法把图像划分成 16×16 的均匀矩形块,本文算法采用最大深度为8,匹配点阈值为20的参数对图像分块.从示意

图可以看出,基于特征点密度的四叉树分块是不均匀分块,左边分块密集,右边分块稀疏,这与图5的目标图像的匹配点密度是完全对应的,即匹配点密度大的区域密集分块,匹配点密度小的区域稀疏分块.



(a)APAP 算法的图像分块示意



(b)本文算法的图像分块示意

图6 APAP算法与本文自适应四叉树的分块效果对比

每幅目标图像的分块情况只取决于四叉树最大深度 D_e 和每个分块的匹配点阈值 T_h 两个参数.这里直接将 D_e 设定为8,实验发现,人眼很难看出不同 T_h 拼接出的图像的差别,所以这里对匹配点使用均方根误差(Root Mean Squared Error, RMSE)来定量分析 T_h 的取值情况. RMSE的计算如下:

$$E_{\text{rms}}(f) = \sqrt{\frac{1}{N} \sum_{i=1}^N \|f(x_i) - x'_i\|^2}, \quad (11)$$

式中, f 表示变换模型; x_i 表示目标图像中的匹配点; x'_i 表示 x_i 在参考图像中的对应匹配点; $f(x_i)$ 表示 x_i 经过变换模型 f 变换后的点,是 x'_i 的估计值; N 是匹配点对的总数.

以railtracks图像对^[6]为例进行实验分析,经过

特征提取和匹配后, 该组图像总共有 2 753 对匹配点. 图 7 是反映 RMSE 与 T_h 关系的折线图, 虚线表示的是 APAP 算法的 RMSE 值(为了提高数据的可信度, 所有测量结果都是重复 50 次实验的平均值). 显然, 当 T_h 小于 11 时, 本文算法的 RMSE 值小于 APAP 算法的 RMSE 值. 进一步地, T_h 在区间[9,12] 时 RMSE 的斜率最大, 在其他区间的斜率相对较小, 为了兼顾图像拼接的效率和精度, 取 T_h 值为 9.

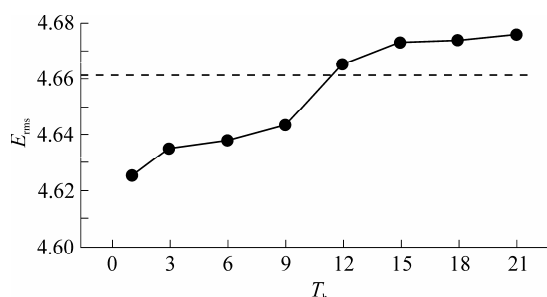


图 7 图像 railtracks 的 RMSE 与 T_h 的关系

经过对大量不同的测试图像反复实验后发现, 这一规律具有普遍性, 即 RMSE 斜率最大的区域集中在 T_h 的某一段区间上. 根据实验归纳, T_h 的取值近似满足下式时, 图像拼接可以较好地兼顾效率和精度问题:

$$T_h = \begin{cases} 1, & S_p < 150, \\ \left\lfloor \frac{S_p}{300} \right\rfloor, & \text{其他,} \end{cases} \quad (12)$$

式中, S_p 表示匹配点的总数; $\lfloor x \rfloor$ 表示对 x 四舍五入取整.

表 1 给出了上述 3 种算法的实验数据.

表 1 3 种算法的均方根误差与运行时间的比较

算法	railtracks ^[6]			temple ^[4]		
	RMSE	分块数/块	时间/s	RMSE	分块数/块	时间/s
全局映射	14.319	-	0.630	7.454	-	0.075
APAP	4.661	10 000	6.077	2.496	10 000	1.737
本文	4.644	835	1.446	2.476	1 663	0.443

算法	carpark ^[4]			roundabout ^[14]		
	RMSE	分块数/块	时间/s	RMSE	分块数/块	时间/s
全局映射	5.005	-	0.086	4.985	-	0.292
APAP	2.141	10 000	1.537	3.726	10 000	2.129
本文	2.139	1 138	0.317	3.695	985	0.634

从表 1 的多对测试图像测出数据可以看出, 本文所提算法的 RMSE 比 APAP 算法稍小, 在一定程

度上提高了拼接算法的精度. 所提出的算法只需要对图像做很少的分块就可以达到 APAP 算法中分为 10 000 个图像块的拼接精度, 效率远大于 APAP 算法.

3 结论

APAP 算法是局部映射模型的一个代表性算法, 在图像拼接上表现出较好的性能. 就 APAP 算法的复杂度和精度问题, 本文提出一种既能保持 APAP 算法的精度, 同时又能大幅度提高算法效率的改进算法. 其主要思路是根据图像匹配点的分布, 设计了一种自适应四叉树图像分块的算法, 对每个图像块使用 MDLT 算法来估计图像块的单应性矩阵, 利用单应性矩阵实现图像映射后, 再进行图像融合. 研究表明, 所提出算法在保证精度的同时能够大幅度减少原有 APAP 算法的图像分块数量, 有效地提高了图像拼接的效率. 拼接算法中的图像形变以及视差容忍度问题有待更深入的研究.

参考文献:

- [1] Xiang T, Xia G S, Zhang L, et al. Locally warping-based image stitching by imposing line constraints[C]. 23rd International Conference on Pattern Recognition (ICPR), 2016:4178-4183.
- [2] 谷雨, 周阳, 任刚, 等. 结合最佳缝合线和多分辨率融合的图像拼接[J]. 中国图象图形学报, 2017, 22(6):842-851.
- [3] Brown M, Lowe D G. Automatic panoramic image stitching using invariant features[J]. International Journal of Computer Vision, 2007, 74(1):59-73.
- [4] Gao J, Kim S J, Brown M S. Constructing image panoramas using dual-homography warping[C]. IEEE Conference on Computer Vision and Pattern Recognition, 2011:49-56.
- [5] Lin W Y, Liu S, Matsushita Y, et al. Smoothly varying affine stitching[C]. IEEE Conference on Computer Vision and Pattern Recognition, 2011:345-352.
- [6] Zaragoza J, Chin T J, Brown M S, et al. As-Projective-As-Possible image stitching with moving DLT[C]. IEEE Conference on Computer Vision and Pattern Recognition, 2013:2339-2346.
- [7] Li Y, Wang S, Tian Q, et al. A survey of recent advances in visual feature detection[J]. Neurocomputing, 2015, 149 (PB):736-751.

- [8] Lowe D G. Distinctive image features from scale-invariant keypoints[J]. International Journal of Computer Vision, 2004, 60(2):91-110.
- [9] Salahat E, Qasaimeh M. Recent advances in features extraction and description algorithms: A comprehensive survey[C]. 2017 IEEE International Conference on Industrial Technology (ICIT), 2017:1059-1063.
- [10] Fischler M A, Bolles R C. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography[J]. Communications of the ACM, 1981, 24(6):381-395.
- [11] 冯宇平. 图像快速配准与自动拼接技术研究[D]. 长春: 中国科学院长春光学精密机械与物理研究所, 2010.
- [12] Zaragoza J, Chin T J, Tran Q H, et al. As-projective-as-possible image stitching with moving DLT[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2014, 36(7):1285-1298.
- [13] 曾清红, 卢德唐. 基于移动最小二乘法的曲线曲面拟合[J]. 工程图学学报, 2004(1):84-89.
- [14] Lin C C, Pankanti S U, Ramamurthy K N, et al. Adaptive as-natural-as-possible image stitching[C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015:1155-1163.
- [15] 白瑞广. 全景视频系统若干关键技术研究[D]. 湘潭: 湘潭大学, 2014.

An image stitching algorithm via adaptive quadtree segmentation

WANG Yuan-wei¹, YU Mei^{1,2*}, JIANG Hao¹, SHAO Hua¹

(1.Faculty of Information Science and Engineering, Ningbo University, Ningbo 315211, China;

2.State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China)

Abstract: Image stitching is a hot topic in computer graphics research, and the representative As-Projective-As-Possible (APAP) stitching algorithm segments image into uniform blocks to achieve local image warping, which does not take into full account the differences between varying image regions. In order to achieve the tradeoff between the accuracy and the efficiency of image stitching, an image warping algorithm based on the feature points distribution is proposed, which uses quadtree to segment image into blocks to achieve local image warping for image stitching. Firstly, the maximum depth of the quadtree is set, and the matching threshold of each image block is estimated according to the number of image matching points. Then, the maximum depth and the matching point threshold is used to adaptively segment the image into blocks by quadtree. Finally, homography matrices of different image blocks are estimated by Moving Direct Linear Transformation algorithm to complete the stitching. The experimental results show that the proposed algorithm notably improve the accuracy of image stitching, and the speed is much faster than the APAP algorithm, and the quality of the stitched image is ameliorated accordingly.

Key words: image stitching; image warping; projective transformation; quadtree

(责任编辑 韩 超)