

基于 OpenCV 的图像碎片拼接

董 乾 黄晓鸣

(北京交通大学理学院数学系,北京 100044)

摘 要 针对二维不规则图像碎片,提出了一种基于 OpenCV 和图像角点的图像碎片拼接算法。该算法实现了碎片轮廓检测、角点提取、角点序列匹配、图像拼接及缺失修复的功能。通过实例测试,可以得到很好的拼接效果。

关键词 OpenCV 角点匹配 图像拼接

中图法分类号 TP391.41; **文献标志码** A

OpenCV (Intel® Open Source Computer Vision Library) 是 Intel 公司面向应用程序开发者开发的计算机视觉库,其中包含大量的函数用来处理计算机视觉领域中常见的问题。OpenCV 具备强大的图像和矩阵运算能力,可以大大减少开发者的编程工作量,有效提高开发效率和程序运行的可靠性。

角点是二维图像亮度变化剧烈的点或图像边缘曲线上曲率极大值的点。这些点在保留图像图形重要特征的同时,可以有效地减少信息的数据量,使其信息的含量很高,有效地提高了计算的速度,有利于图像的可靠匹配。

1 算法流程

本文研究了如何利用图像角点进行图像碎片拼接的问题。首先进行是角点提取的工作。角点提取后会产生角点序列,我们需要的是沿轮廓方向的角点序列。如果得到的角点序列不合这一要求,需要进行重排。然后进行序列的匹配,通过匹配,会得到图像碎片上匹配的角点。这样,就可以对图像碎片拼接进行定位,完成拼接的过程。最后,对得到的拼接后的图像进行缺失修复,用来弥补原始信息的丢失和拼接误差。算法主要流程如

图 1。

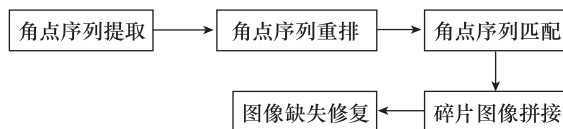


图 1 算法流程图

1.1 图像碎片预处理

若原图为彩色图像,要先进行色彩空间转换,将图像转化为灰度图。对得到的灰度图利用 OpenCV 中函数进行处理,完成拼接。这一过程中,主要应用了 OpenCV 的 CvtColor 函数。

1.2 角点序列提取与重排

角点序列的提取应用了 OpenCV 的 cvGoodFeaturesToTrack 函数。但存在两个问题:

(1)角点序列并不是沿图像的轮廓线存储的,则无法转化为方便判定的序列,如沿轮廓线方向的角序列,或者是相邻角点间的距离序列。

(2)提取的角点序列,有时会包含轮廓外的角点,这样的点会影响到轮廓匹配的判定。

以上两个问题的解决,应用了 OpenCV 的 cvFindContours 函数^[1]。该函数可以得到沿轮廓线的顺时针方向存储的图像轮廓点序列,基于这一特点,给出了解决角点序列存储的算法:

(1)调用 cvFindContours 函数,得到轮廓线的点序列 $Contour = \{Q_1, Q_2, \dots, Q_m\}$;

(2)调用 cvGoodFeaturesToTrack 函数,得到角

点序列 $\text{Corners} = \{P_1, P_2, \dots, P_n\}$;

(3) 从 Contour 序列中依次取出 Q_1, Q_2, \dots, Q_m , 将其与 Corners 序列中的元素作比较。如果对于某一 $Q_i (i=1, 2, \dots, m)$, 存在 $P_j (j=1, 2, \dots, n)$, 使得 $\|P_i - Q_{j\infty}\| \leq 2$, 则将 P_i 放入新的序列 Corners0 中。这样得到的序列 Corners0 满足了序列点沿轮廓线存储的要求; 同时, 将条件适当放宽, 把在轮廓线附近的角点也存储起来, 充分利用图像中的信息, 便于图像匹配的判定。通过测试, 发现找到的角点序列符合存储要求。而且, 通过调节参数, 可以得到尺度不同的角点序列。所以, 可以通过调节参数的设定来优化角点

$$\alpha_i = \arccos \frac{(x_{i+1} - x_i)(x_{i-1} - x_i) + (y_{i+1} - y_i)(y_{i-1} - y_i)}{\sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2} \sqrt{(x_{i-1} - x_i)^2 + (y_{i-1} - y_i)^2}}$$

其中, $\text{Corners0} = \{P_i = (x_i, y_i) | i=1, 2, \dots, n\}$ 。

此时, 由角点生成的角序列的集合:

$$\{\alpha_i | i=1, 2, \dots, n\}.$$

1.3.2 角序列匹配(最长公共子序列算法)

根据上面的角度计算方法, 分别计算两块图像碎片的角点序列生成的角序列记为:

$$a = \{a_i | i=1, 2, \dots, m\} \text{ 和 } b = \{b_j | j=1, 2, \dots, n\},$$

其中, m 和 n 分别是角序列中包含的角度值的数量。由于角度的计算顺序是按照角点存储的顺序进行的, 所以得到的角序列与角点序列的顺序是一致的, 为轮廓的顺时针方向。在进行两块图像碎片的拼接过程中, 易知在拼接匹配点处角度值应该是相等的。即如果 $|a_i - b_j| < \varepsilon$, 其中 ε 是容许误差, 则认为第一条轮廓线的第 i 个点与第二条轮廓线的第 j 个点匹配的。这样可以把 a 和 b 看作两个字符串, 求得这两个字符串的最长公共子串。但是需要注意, 在两图像碎片匹配时, 两个角序列的方向必须是相反的。这就要求 a 和 b 是反序的, 所以应在试配之前先对其中的一个序列进行反序处理, 例如对 b 序列进行反序处理得到序列 b' 。然后, 再将序列 a 和 b' 看作两个字符串, 求得这两个字符串的最长公共子串。

寻找字符串 a 和 b' 的最长公共子串, 需要确定三个量: 最长公共子串的长度 l , 最长公共子串在串 a 中的起始位置 a_{i_0} , 最长公共子串在串 b' 中的起始

检测。

1.3 角点序列匹配

1.3.1 角序列的计算

对角点序列的匹配是通过角点的一些特征作为角点匹配的判定条件实现的, 本文选取角点处的角度作为判定角点匹配的条件^[2]。显然, 对于两块碎片的匹配处, 以角点为顶点的角度应该相等(考虑不超过 180° 的角)。因此, 可以从角点序列生成角点的角序列, 然后对角序列进行匹配, 实质上就完成了对角点序列的匹配。

角序列的生成根据向量内积法则, 可得

起点 b'_{i_0} 。因此, 可以先固定公共子串在 a 中的位置 a_i , 然后分别以 b' 中的每个元素分别为公共子串的起点, 计算公共子串的长度。这样, 可以得到以 a_i 为起点的最长公共子串 (b' 中对应子串的起始位置也可以确定)。然后, 再从这些公共子串中取出最长的子串即得到最长公共子串。

为了减少冗余的运算, 依据模式匹配 KMP 算法的思想^[3], 在以 a_i 为起始位置寻找公共子串时, 做到使 b' 的指针不回退。即如果现在找到了公共子串 $\{a^i, a^{i+1}, \dots, a^{i+k}\}$ 和 $\{b'_j, b'_{j+1}, \dots, b'_{j+k}\}$, 而且发现 $a_{i+k+1} \neq b'_{j+k+1}$, 则在下次寻找的过程中, 跳过 b'_{j+1} 直接将 b'_{j+k+1} 作为起始位置。其他的地方不做改动。

通过对两个碎片角序列的匹配, 可以找到匹配的角, 根据一一对应关系, 可以确定匹配的角点。

1.4 碎片图像拼接

假定以图像碎片 F 为当前图像碎片, 将图像碎片 F' 与之拼合。在拼接过程中, 主要涉及到图像的旋转与平移。假定 F 中的点 $A(x_A, y_A), B(x_B, y_B)$ 分别对应 F' 中的点 $A'(x'_A, y'_A), B'(x'_B, y'_B)$, 基本的思路就是先对 F' 进行旋转, 然后再平移, 将两幅图拼到一起。算法如下:

(1) 对 F' 进行一次坐标变换将坐标原点变换到 A' ;

(2) 确定旋转角,通过计算向量 \overrightarrow{AB} 与向量 $\overrightarrow{A'B'}$ 之间的夹角 θ ,对图像 F' 以原点 A' 为旋转中心旋转角度 θ ;

(3) 在此基础上,计算点 A' 到 A 的平移量,得到图像 F' 到图像 F 的平移向量 Δ ,作平移变换。

通过以上步骤,可以得到图像 F' 到图像 F 的坐标变换公式:

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x' - x'_A \\ y' - y'_A \end{pmatrix} + \begin{pmatrix} x_A \\ y_A \end{pmatrix}。$$

其中, $\theta = \tan^{-1}\left(\frac{y_A - y_B}{x_A - x_B}\right) - \tan^{-1}\left(\frac{y'_A - y'_B}{x'_A - x'_B}\right)$ 。

1.5 图像缺失修复

两块碎片完成拼接后,往往会在拼接处留下一条细缝。出现这种情况,主要是在旋转角度和平移距离的计算上存在误差,加上坐标变换时进行了近似处理。细缝的出现,对于接下来对这张拼接后碎片的处理是一个难题,影响到轮廓检测和角点检测。因此,需要对得到的图像进行进一步的处理。

解决这一问题应用了数学形态学操作^[4],先进行一次膨胀将细缝消除,再进行一次腐蚀还原图像外轮廓。方法的实现利用了 OpenCV 中的 cvDilate 函数(形态学膨胀)和 cvErode 函数(形态学腐蚀)。通过测试,发现数学经形态学处理后,细缝部分基本消除。但是,图像轮廓变的相对模糊。因此,需要对所得的图像进行增强处理。利用 OpenCV 中的 cvEqualizeHist 函数,进行图像增强处理,得到符合要求的拼接结果。

2 应用实例

为了验证算法的有效性,给出一个两块图像碎片拼接的实例。

图 2 给出了原始的图像碎片,直观上两幅图显然可以拼接为一幅图。对两幅碎片图像进行预处理,得到二值化的图像(图 3)。在二值化的图像上进行角点检测(结果为图 4)与轮廓检测,同时完成对角点序列的提取与存储。根据角点序列计算出角序列,得到最长公共子序列,从而确定图像拼接

的参数,得到拼接结果为图 5。从图中可以看出,拼接效果是相当好的,只是在两碎片之间有细小的缝隙,拼接位置是吻合的。进行图像修复后,从图 6 可以看出,细缝得到了较好的消除,已经不会影响下次对该图像的处理。最后进行一次图像增强操作,结果为图 7。



图 2 原始的碎片图像

图 3 预处理后的图像

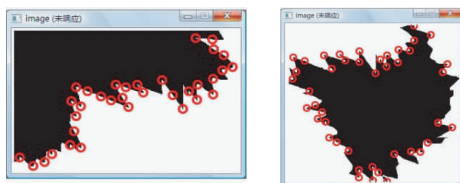


图 4 角点检测结果



图 5 图像拼接结果

图 6 图像修复结果



图 7 图像增强结果

3 结论

利用 OpenCV 中图像处理函数,解决了图像碎片拼接过程中关键的角点序列的提取与重排以及图像碎片拼接的修补,实现了二维不规则图像碎片的拼接算法。代码具有较高的执行效率和运行可靠性,可作为图像碎片拼接系统的内核。

参 考 文 献

- 1 谢凤英,赵丹培. Visual C++ 数字图像处理. 北京:电子工业出版社,2008
- 2 周 丰,黄晓鸣. 基于角序列的二维碎片轮廓匹配算法. 科学技术与工程,2007;7(15):3757—3760
- 3 严蔚敏,吴为民. 数据结构(C语言版). 北京:清华大学出版社,1997
- 4 陈盛勇,刘 盛. 基于 OpenCV 的计算机视觉技术实现. 北京:科学出版社,2008

The Stitching Algorithm of Image Fragments Based on OpenCV

DONG Qian, HUANG Xiao-ming

(Department of Mathematics of Science Institute, Beijing Jiaotong University, Beijing 100044, P. R. China)

[Abstract] Based on OpenCV and image corners, a stitching algorithm of two-dimensional irregular image fragments is proposed. This algorithm carried out the functions of fragment contour detection, corner extraction, corner sequence matching, image mosaic and missing image repair. According to test cases, outstanding stitching effect could be achieved.

[Key words] OpenCV corner matching image mosaic

(上接第 5424 页)

- 5 胡忠义,胡荣强. 基于 ARM 处理器 S3C2410 的视频播放器的实现. 电子元件应用,2009;11(3):10—13
- 6 吴 骏,肖志涛. 基于 S3C2410A 的嵌入式网络通信系统设计与实现计算机应用研究,2008;25(6):1897—1899
- 7 许生模,余 敏. 嵌入式 QT 的内核分析与优化. 微计算机信息,2007;23(10):59—61
- 8 Chen Zujue, Zhang Zhixiong, Zhang Jianjiang. Design and implementation of video player system based on embedded system and Qt/E. Visual Information Engineering, 2008. VIE 2008. 5th International Conference on. July 29 2008 – Aug. 1 2008;468—472

Design and Implementation of Multimedia Terminal

ZHAO Hong, YIN Lei, CAO Jie, SHEN Chao

(School of Computer and Communication, Lanzhou University of Technology Lanzhou 730050, P. R. China)

[Abstract] At present, with commonly used embedded multimedia terminal cost performance is not high, and development is difficult and other problems, thus an embedded multimedia terminal based on SAMSUNG S3C2410 microprocessor chip and using QT/Embedded for program development tools has been designed. This embedded multimedia terminal use ARM9 processing and OSS(open source software) QT/Embedded, which can effectively improve cost performance. It features prompt, high effectiveness and inexpensive. The result shows that the scheme of the multimedia terminal achieved the design target for the terminal working is stable and design is rational.

[Key words] multimedia terminal ARM Linux QT/embedded FFmpeg