

基于尺度不变特征变换特征的显微图像在线拼接方法^①

孙明竹^② 赵 新 程小燕 卢桂章

(南开大学机器人与信息自动化研究所 天津 300071)

摘 要 为解决显微镜观测视野狭小的问题,提出了一种基于尺度不变特征变换(SIFT)特征的显微图像自动拼接方法。该方法充分考虑了显微图像采集过程以及显微图像自身的特点,通过一系列适用于显微图像拼接的算法,控制单幅图像拼接时间在 1 秒内,进而可以保证显微图像拼接随图像采集在线完成,同时,利用 SIFT 特征匹配准确度高的优点,可达到亚像素级的拼接精度。该方法拓展了显微镜的应用,尤其适用于生物和医学显微图像观测。

关键词 显微图像, 在线图像拼接, SIFT 特征, 图像融合

0 引 言

用显微镜可以看到微小物体,但显微镜存在视野狭小的问题,在一些情况下,大范围感兴趣区域无法显示在同一视野下,这给显微观测和操作造成了极大不便,尤其在生物和医学显微观测。此时,需要用到图像拼接(Image stitch/mosaic)技术。图像拼接是指将存在重叠区域的多幅图像拼成一幅大视野图像,由此增大视野范围或提高观测分辨率。对于非运动观察对象(如贴壁细胞或组织切片),可以利用图像拼接技术将多幅局部显微图像拼接到一起,形成一幅全局视野图,进而指导显微操作。在生物和医学应用中,观测目标一般是活体组织或细胞,长时间暴露在空气中会导致目标活性降低甚至死亡,必须尽可能减少观测时间,因此,在显微图像拼接中,局部图像获取与图像拼接的效率应重点考虑。

图像拼接的第一步(最重要)是得到相邻局部图像的位置匹配关系。按使用图像信息的不同,图像配准方法一般分为基于灰度信息和基于特征两类^[1]。前者直接利用图像重叠区的亮度一致性估计变换参数,由于考虑了重叠区所有像素的信息,因而具有较高的精度,但不适用于图像之间形变较大的情况;基于特征的配准方法根据图像特征之间的几何关系确定配准参数,因此具有较高的可靠性,但其配准精度往往低于前者。从精度上考虑,显微图像

拼接一般采用基于灰度的拼接方法^[2-4],由于很难准确确定重叠区域的搜索范围,此类方法的处理时间都较长,两幅图像拼接约需数秒。如果通过图像金字塔方法^[5, 6]加快搜索步长,虽可减少搜索时间,但从实际拼接效果看,此法有可能错过图像的最佳重叠位置,最终造成错误拼接。基于特征拼接常用的特征包括图像双梯度^[7]、Harris 角点^[8]等,由于缺乏对特征信息的描述,如何快速、正确地完成特征匹配,是这类方法考虑的重点。

近几年来,尺度不变特征变换(scale invariant feature transform, SIFT)特征检测成为国内外特征匹配领域的研究热点。SIFT 特征是一种基于尺度空间的对图像缩放、旋转甚至仿射变换保持不变性的图像局部特征描述算子。在物体识别及全景图拼接^[9]方面, SIFT 特征都表现出良好的性质。利用 SIFT 特征进行显微图像拼接在最近一年已有报道^[10, 11],但所报道的方法均与全景图拼接类似,在拼接过程中过多考虑了相邻图像配对及图像旋转,拼接速度无法达到“随采集随拼接”的在线图像拼接要求。事实上,从分析图像拼接过程可知,显微图像的采集一般由电动载物台辅助完成,载物台在二维平面内沿 X 轴或 Y 轴运动,显微镜下采集的图像基本上没有畸变,因此可以忽略采集过程中微小的尺度变化和旋转。另外,图像采集顺序隐含了相邻图像的信息,可以利用该信息直接得到待拼接图像对,不再需要其他额外处理。本文基于上述思想,在保

① 国家自然科学基金(60875059),教育部新世纪优秀人才支持计划(NCET-07-0464)和教育部高等学校博士学科点专项科研基金(20060055037)资助项目。

② 女,1981 年生,博士;研究方向:图像处理,微操作机器人;联系人, E-mail: sunmz@robot.nankai.edu.cn (收稿日期:2008-11-07)

证拼接精度的基础上,将拼接效率作为核心,提出了一种适用于大范围显微图像的实时自动拼接方法。该方法将单幅图像拼接时间控制在 1 秒以内,因此可以随图像采集在线完成拼接,图像采集结束时即可观察到大视野显微图像。

1 显微图像采集

图像采集由本单位自行研制的微操作机器人系统完成,此系统采用层次化和模块化的结构,分为上位机部分和下位机部分,如图 1 所示。

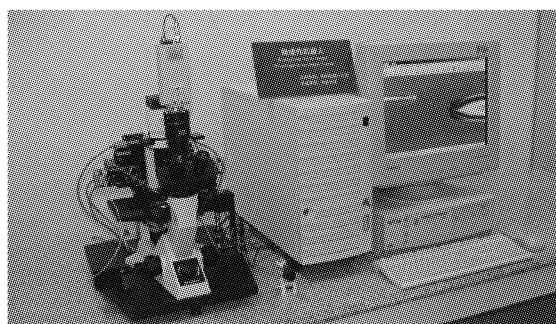


图 1 微操作机器人系统

系统配备 Olympus CK40 显微镜,采集图像时使用的 CCD 及图像采集卡分别为 Panasonic wv-CP470 和 Daheng CG400。通过显微镜标定可知,在 $4\times$ 物镜下,单个像素对应 $1.1\mu\text{m}$ 。微操作机器人系统通过电动方式驱动显微镜载物台,载物台可沿 X 方向和 Y 方向运动,移动精度为 $2\mu\text{m}$,运动范围为 $10\text{mm}\times 10\text{mm}$,最快的移动速度为 $1500\mu\text{m/s}$ 。

在采集待拼接显微图像序列时,首先将目标置于改造后的载物台上,并大致确定目标的分布区域。一般将该区域设置得大一些,以防止目标丢失。之后,通过平面扫描方式获取目标分布区域的局部显微图像。扫描过程中,选取目标区域的某个顶点作为扫描的起始点和终点,即扫描结束后平台又回到起始位置。整个扫描区域被分割成大小等同于当前屏幕视野的若干个单元格,各单元格之间要有一定程度的重叠,以保证图像信息不丢失,便于后续的图像拼接。为了尽量减少采集图像的数量,通常将重叠区设为图像大小的 20% 左右。在本文中,电动载物台以折线的方式运动,每次拍摄完都把相邻单元格移入显微视野,载物台运动路径如图 2 所示。

由上述运动路径可知,当确定 X 方向与 Y 方向扫描图像的数量后,可以通过图像序列的编号判断

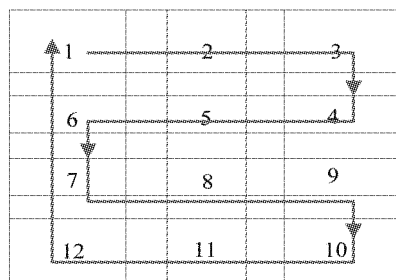


图 2 电动载物台运动路径示意图

图像的邻接情况。例如图 2 中,编号为 5 的图像,其相邻图像只可能为 1、2、3、4、6、7、8、9 号图像;同时,实际处理中,由于 6-9 号图像后于 5 号图像采集,因此,在拼接 5 号图像时,只需考察其与 1-4 号图像的匹配情况。这样,在图像拼接过程中,只需将当前处理图像与先于它采集的可能邻接的图像进行特征匹配,从而大大减少了参与匹配的图像数量。

2 基于 SIFT 特征的图像拼接方法

2.1 SIFT 特征检测及匹配

检测待拼接图像中的特征是图像拼接的第一步,本文使用的图像特征为 SIFT 特征,文献[12]给出了特征检测的过程:

- (1) 获取图像尺度空间,并在尺度空间进行极值检测,以初步确定特征点位置和所在尺度。
- (2) 通过函数插值精确确定特征点位置及尺度,去除低对比度的特征点和不稳定的边缘响应点。
- (3) 利用特征点邻域像素的梯度大小及方向确定该点的主方向,以实现算子对方向的无关性。
- (4) 由特征点邻域像素的梯度信息生成 128 维特征向量,此时 SIFT 特征向量已经去除了尺度变化、旋转等几何变形的影响,再将特征向量归一化,则可进一步去除光照变化的影响。

完成特征检测后,逐一比较两幅图像中各个特征点的特征向量,如果满足一定的相似性条件,则认为找到一个匹配点对。考虑到处理速度及匹配效果,本文使用欧式距离判别特征点的相似程度。

在匹配过程中,取图像 1 的某个特征点,依次与图像 2 中的所有特征点进行比较,找出与该特征欧式距离最近的两个特征点。用最近距离除以次近距离得到一个比例值,如果该值小于某个设定阈值,则接受这个匹配点对。实验中,将阈值设为 0.7,可以实现较好的匹配效果。

2.2 图像变换估计

对于普通数码照片而言,由于观测位置和角度不同,两幅待拼接图像间一般会存在平移、旋转、缩放、变形等变化,这些变化通常可由仿射变换描述:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \mathbf{M} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (1)$$

其中, $[x' \ y' \ 1]^T$ 和 $[x \ y \ 1]^T$ 分别为同一特征在不同图像中的齐次坐标, $\mathbf{M}$ 是仿射变换矩阵。

通过对显微图像采集过程的分析可知,由于显微物镜垂直放置,在采集过程中不存在观测角度的变化,同时,控制电动载物台在二维平面内做平移运动,可以忽略采集过程中微小的尺度变化和旋转。这样,对于显微图像拼接,上述变换可简化为

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & \Delta x \\ 0 & 1 & \Delta y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} + \begin{bmatrix} \Delta x \\ \Delta y \\ 0 \end{bmatrix} \quad (2)$$

也即待拼接图像之间只存在平移变化,图像偏移量用 $\Delta x, \Delta y$ 表示,文献[11]通过实验分析得出了相同结论。

图像间偏移可由同一特征点在不同图像中的位置变化直接得出,为了提高拼接精度,可以将多个特征点偏移量取平均作为图像最终的偏移量。然而,这种方法对于特征错误匹配的情况十分敏感,如果不去除错误匹配点,就有可能造成拼接失败。

在图像拼接中,通常采用随机采样一致性算法排除错误匹配点(又称为外点, outliers)^[9-11],考虑到实现效率,本文借助模式识别中的聚类思想,将特征沿 X, Y 方向的偏移量作为分类依据进行聚类,同时,控制聚类半径,取特征最多的一个类别,只使用这些特征计算图像偏移量。由于 SIFT 特征对于显微图像比较稳定,不会产生过多的错误匹配点。从实验结果看,上述方法基本可以排除所有的错误匹配,而且具有较高的处理速度。

2.3 图像位置确定

按照图像采集和特征匹配的策略,当前待拼接图像只与先于它采集的、可能邻接的图像进行特征匹配,这样,设定第一幅采集图像在拼接后全局图像中的位置为 $(0,0)$,可由相邻图像间的变换关系求得后续图像位置。设待拼接图像编号为 j ,根据其相邻图像 i 得到的图像位置为

$$x_j^i = x_i + \Delta x_j^i, \quad y_j^i = y_i + \Delta y_j^i \quad (3)$$

其中, (x_i, y_i) 是已知的图像 i 在全局图像中的位置, $(\Delta x_j^i, \Delta y_j^i)$ 是图像 j 相对于图像 i 的偏移量。

一般来说,待拼接图像不可能只有一幅相邻图像(如图2所示),这样,由公式(3)会计算出多个图像位置。在这些位置不一致的情况下,本文通过特征数加权计算当前待拼接图像在全局图像中的最终位置。计算公式如下:

$$x_j = \sum_i n_{ij} x_j^i / \sum_i n_{ij}, \quad y_j = \sum_i n_{ij} y_j^i / \sum_i n_{ij} \quad (\text{图像 } i, j \text{ 邻接}) \quad (4)$$

其中, n_{ij} 为特征聚类后用于计算图像 i, j 间偏移量的特征数目。

2.4 图像融合

在线显微图像自动拼接过程中,随着图像采集,在等待电动载物台运动的同时,上位机软件执行上述步骤;当待拼接区域采图完毕后,即可同时获得各幅局部显微图像在全局图像中的位置。对于全局图像来说,图像上每一个点有可能对应到多幅局部显微图像的不同点,此时需要进行图像融合,以消除拼接缝隙,实现平滑连接,本文采用简化的多波段融合技术实现图像融合。

多波段融合的主要思想是将图像在频域内进行划分,低波段图像在较大空间进行融合,高波段图像在较小空间融合,以减弱直接融合造成的高频细节信息丢失。在融合过程中,该方法认为从图像中心到边缘,图像灰度值越来越不可靠,因此,给予每幅局部图像一个权值矩阵:

$$W(x, y) = w(x)w(y) \quad (5)$$

其中, (x, y) 是一幅局部图像沿 X, Y 方向坐标, $w(x)$ 自中心至边缘,从 1 到 0 均匀变化。实际上,该权值矩阵给出了由于像素点在局部图像中所处位置不同,其亮度值的可靠性度量。

之后,对于全局图像中的每一个点,根据其在各幅局部图像中的位置,也即该点灰度值的“可靠度”,确定局部图像在最低波段的融合权值矩阵:

$$W_{\max}^i(x, y) = \begin{cases} 1 & \text{如果 } i = \arg\max_j W^j(x, y) \\ 0 & \text{其他} \end{cases} \quad (6)$$

其中, j 为所有与图像 i 有邻接关系的图像编号。

接下来,设图像 i 的亮度为 $I^i(x, y)$, 初始化 $I_0^i = I^i, W_0^i = W_{\max}^i$, 用 $B_{k\sigma}^i(x, y)$ 表示图像 i 在 $[(k-1)\sigma, k\sigma]$ 波段的图像亮度, $W_{k\sigma}^i(x, y)$ 为该波段的融合权值矩阵,则其迭代计算公式分别为

$$B_{k\sigma}^i(x, y) = I_{(k-1)\sigma}^i(x, y) - I_{k\sigma}^i(x, y) \quad (7)$$

$$I_{k\sigma}^i(x, y) = I_{(k-1)\sigma}^i(x, y) * g_{\sigma'}(x, y)$$

$$W_{k\sigma}^i(x, y) = W_{(k-1)\sigma}^i(x, y) * g_{\sigma'}(x, y) \quad (8)$$

其中, $k \geq 1$, $g_{\sigma'}(x, y)$ 是方差为 σ' 的高斯函数, $*$ 代表卷积。迭代过程中, σ' 由下式定义:

$$\sigma' = \sqrt{k^2 \sigma^2 - (k-1)^2 \sigma^2} = \sqrt{2k-1} \sigma \quad (9)$$

由此,可以得到在 $[(k-1)\sigma, k\sigma]$ 波段的图像融合结果:

$$I_{k\sigma}(X, Y) = \sum_i (B_{k\sigma}^i(X - x_i, X - y_i) W_{k\sigma}^i(X - x_i, X - y_i)) / \sum_i W_{k\sigma}^i(X - x_i, X - y_i) \quad (10)$$

此时, (X, Y) 代表全局图像沿 X, Y 方向坐标, (x_i, y_i) 是在 2.3 中求得的图像位置。最终,将各波段的融合结果加到一起,得到整幅图像的融合结果。

实际处理中,可根据成像情况选择适当的分波段个数及波段带宽。由于在计算高波段亮度图像和融合权值矩阵时,需要使用卷积运算,当分波段过多,或处理图像尺寸巨大时,需要较长的融合时间。在文献[9]中,使用上述方法融合一幅 8908×2552 图像(分为 3 个波段),共耗时 15min,这显然无法满足实时显微图像拼接的需要。另一方面,对于显微图像来说,通过 CCD 采集到的图像是显微镜视野的中心部分,图像中心和周围的光照变化较小,而且图像边缘也不存在明显的变形,因此,本文采用简化的多波段融合方法,将待拼接图像作为一个波段处理,这样,不需要进行卷积运算,对于千万像素的全局图像,可将图像融合时间控制在秒级。

总结上述各步骤,图 3 给出了在线显微图像拼

接的整体流程,除最后的图像融合外,其他步骤均可在图像采集过程中完成。

3 实验结果

本文实验在个人计算机系统(CPU: AMD Athlon 64 × 2 Dual Core 4800 + 2.51GHz; 内存: DDRII 667 2GB)中完成,实验程序运行在 Visual Studio 2005 平台上。为了实现实时显微图像拼接,文中首先分析了 SIFT 特征提取算法中各参数的取值,之后给出了图像拼接结果,最后将本文方法与多种图像拼接方法进行了对比。

3.1 SIFT 特征检测参数分析

实验中可以发现,在上述图像拼接流程中, SIFT 特征检测与匹配是耗时最多的步骤,按照文献[12]给出的参数设置,处理一幅 576×768 的血细胞切片显微图像,可以得到一万多个 SIFT 特征,在实验计算机上,处理时间超过 20s。分析 SIFT 特征检测算法可知,由于需要对每个特征计算 128 维的特征向量, SIFT 特征点个数直接影响着总体特征检测时间;同时,当特征点数较多时,也会加大特征匹配的时间。因此,通过设置特征检测参数,从而得到适当数量的 SIFT 特征,对于保证图像拼接的实时性至关重要。经实验可知,对于本文使用的血细胞显微图像(576×768),将单幅图像检测出的特征数控制在 100 个左右,是一种较为合适的选择。

在 SIFT 算法的众多参数中,控制检测特征数量的参数主要包括尺度空间层数、特征对比度阈值、边缘响应点阈值以及计算图像金字塔之前是否增倍图像的参数等[12],为了在减少特征数量的同时,尽可能保持特征的各异性,本文使用原始图像计算图像金字塔(不增倍图像),并提高特征对比度阈值,其他参数保持不变。对于不同观测对象,其显微图像存在一定差别,因此,需要在图像拼接前,针对不同目标单独确定特征对比度阈值,通过实验分析,比较合理的选择为 0.07 ~ 0.1(原始值为 0.03)。

对于血细胞显微图像,将 SIFT 特征对比度阈值设为 0.1,可在显微图像中检测出 80 ~ 200 个 SIFT 特征,特征检测时间约为 400 ~ 600ms。图 4 显示了两幅图像进行特征匹配情况,连线表示匹配上的特征。图中共得到 7 个特征点对,没有发生匹配错误,特征匹配时间约为 50ms。

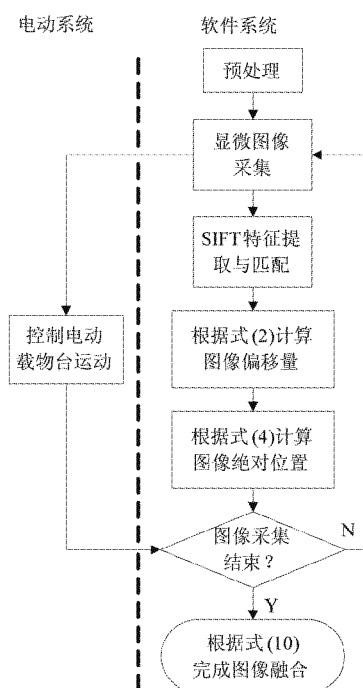


图 3 图像拼接流程

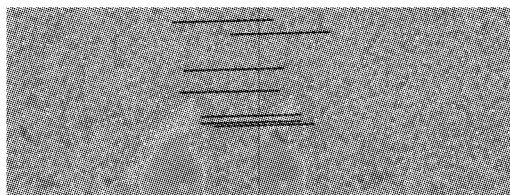


图4 SIFT 特征匹配

3.2 实时显微图像拼接

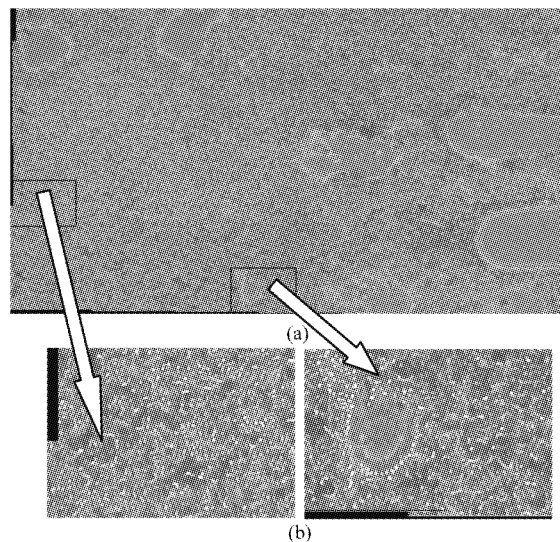
实验中,通过 $4\times$ 显微物镜观察血细胞切片,显微视野的观测面积为 $0.8\text{mm}\times 0.6\text{mm}$ 。在拼接过程中,共采集 30 幅显微图像,拼接结果如图 5(a)所示,此时,实际的观测范围为 $3\text{mm}\times 2.4\text{mm}$,已基本满足生物实验的要求。

在图像拼接时,保存每幅图像的特征提取结果,用于所有与其相邻的图像拼接中,这样,按照拼接策略,每采集一幅图像,只需提取该图像的特征,并与先于它采集的邻接图像进行特征匹配并拼接。当一幅新采集的图像加入拼接结果时,从特征提取到确定图像准确位置的时间,可控制在 1s 之内,其中,SIFT 特征检测及匹配耗时约 $400\sim 600\text{ms}$,后续图像拼接工作耗时 $50\sim 100\text{ms}$ 。在图像采集过程中,一般设置微操作机器人电动载物台的运动速度为 $200\mu\text{m}/\text{s}$,对于 $4\times$ 物镜,电动载物台的移动步长约为 $600\mu\text{m}$,因此,整个运动过程大约需要 3s 时间,在等待电动载物台运动时,足够完成上述拼接过程。

最后,在采图完成后,进行图像融合,即可得到最终的图像拼接结果,融合上述 30 幅显微图像总共耗时 $3\sim 4\text{s}$ 。由于图像只与其有限邻接图像进行匹配,拼接速度仅与图像拼接数量相关,而且大部分图像拼接工作可随图像采集同时进行,因此可以保证图像拼接在线完成。

图像拼接精度由 SIFT 特征检测精度保证,通过 SIFT 算法可以得到亚像素级的特征点位置,因此,基于 SIFT 特征的图像拼接方法具有亚像素级的拼

接精度,图像拼接细节如图 5(b)所示。



(a) 整体拼接结果 (b) 拼接细节展示

图5 图像拼接结果

3.3 拼接方法对比

为了验证拼接方法的有效性,本文将文中方法与基于 SIFT 特征的原始方法^[11]、基于灰度的拼接方法^[1]进行了对比,拼接对比结果如图 6 所示。由于上述几种方法都能准确得到图像间的偏移距离,因此拼接结果类似,均可满足显微图像拼接的要求。从而拼接效率上看,基于 SIFT 特征的原始拼接方法在逐幅图像匹配与基于公式(1)获取图像变换关系上耗费了较多时间,实际上,这些处理都是不需要的;基于灰度的拼接方法由于需要进行逐点灰度值对比,因而耗时较长,虽然使用图像金字塔^[6]可以减少参与对比的数据量,但由于无法有效确定搜索范围,会造成拼接结果不稳定。仍以 3.2 节中 30 幅显微图像拼接为例,各种拼接方法所需时间如表 1 所示。从对比结果可以看出,本文方法在保证拼接效果的基础上,减少了处理时间,最终,显微图像拼接可以伴随图像采集在线完成。

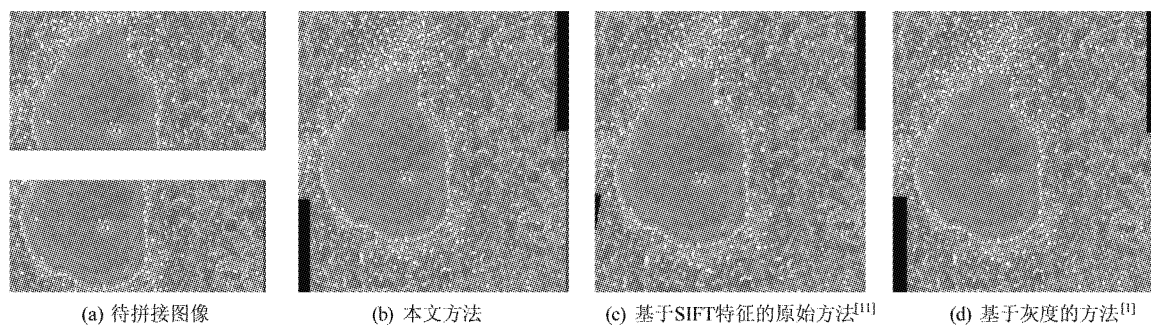


图6 图像拼接结果对比

表 1 图像拼接效率对比

图像拼接方法	所需时间(s)
本文方法	20.51
基于 SIFT 特征的原始方法 ^[11]	76.74
基于灰度的方法(直接) ^[1]	363.5
基于灰度的方法(金字塔) ^[6]	58.84(不稳定)

4 结 论

本文提出了一种基于 SIFT 特征的显微图像自动拼接方法。该方法在考虑显微图像采集及图像自身特点的基础上,对 SIFT 特征提取方法、图像变换矩阵、图像融合方法进行了不同程度的简化,使其更适合于实时显微图像拼接,进而,图像拼接可以伴随图像采集在线完成。实验结果表明,用此方法处理单幅待拼接显微图像的时间小于 1s,同时,基于 SIFT 特征自身的优点,拼接过程可自动完成,拼接精度可达亚像素级。

在线完成显微图像拼接意味着,显微镜观测可更多地考虑观测内容,而不受图像处理时间的限制。这样,可以用显微镜观察更大的视野范围,进一步拓展显微镜的应用。例如,在活体基因导入实验中,观测目标分布于厘米级范围内,借助上述图像拼接方法,可在很短时间内确定众多待操作细胞的准确位置,指导显微操作的进行。

在 4×物镜下,使用本文方法可顺利完成图像拼接。然而,对于高倍数显微物镜,特别是 40×以上的物镜,随着观测细节的不断增多,视野中的空白区域也越来越多,有可能出现 SIFT 特征匹配失败的情况,此时就需要借助基于灰度的拼接方法进行补救;另一方面,SIFT 特征具有尺度不变的特性,是否可以将 SIFT 特征在不同倍率显微图像之间进行匹配,进而利用低倍率显微图像得到高倍率图像间的

位置关系,由此指导图像拼接,这是我们下一步的工作。

参考文献

- [1] Zitova B, Flusser J. Image registration methods: A survey. *Image and Vision Computing*, 2003, 21(11): 977-1000
- [2] Capek M, Krekule I. Alignment of adjacent picture frames captured by a CLSM. *IEEE Transactions on Information Technology in Biomedicine*, 1999, 3(2): 119-124
- [3] Rankov V, Locke R J, Edens R J, et al. An algorithm for image stitching and blending. In: *Proceedings of SPIE, Three-Dimensional and Multidimensional Microscopy: Image Acquisition and Processing XII*, San Jose, USA, 2005. 190-199
- [4] 苗立刚, 岳永娟, 彭思龙. 基于 2D 格状图的显微图像拼接. *计算机工程*, 2007, 33(12): 49-51
- [5] Adelson E H, Anderson C H, Bergen J R, et al. Pyramid method in image processing. *RCA Engineer*, 1984, 29(6): 33-41
- [6] 聂生东, 司京玉. 医学显微图像自动拼接的方法研究. *中国生物医学工程学报*, 2005, 24(2): 173-178
- [7] 聂斌, 刘照军, 韩忠东等. 医学显微图像的自动拼接方法. *第四军医大学学报*, 2005, 26(6): 20
- [8] Sun C M, Beare R, Hilsenstein V, et al. Mosaicing of Microscope Images. In: *Proceedings of the Digital Imaging Computing: Techniques and Applications*, Cairns, Egypt, 2005. 50-55
- [9] Brown M, Lowe D G. Automatic Panoramic Image Stitching using Invariant Features. *International Journal of Computer Vision*, 2007, 74(1): 59-73
- [10] Fan X, Xia S R. Feature Based Automatic Stitching of Microscopic Images. In: *Proceedings of International Conference on Intelligent Computing*, Qingdao, China, 2007. 791-800
- [11] 汤井田, 王凯, 肖嘉莹. 基于 SIFT 特征检测的医学显微图像自动拼接. *计算机工程与应用*, 2007, 43(35): 243-244
- [12] Lowe D G. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 2004, 60(2): 91-110

On-line microscopic image stitching based on SIFT feature

Sun Mingzhu, Zhao Xin, Cheng Xiaoyan, Lu Guizhang

(Institute of Robotics & Automatic Information System, Nankai University, Tianjin 300071)

Abstract

To solve the problem of narrow field of view in microscope, this paper proposes a new method of microscopic image stitching based on scale invariant feature transform (SIFT) features. This approach fully considers the properties of image acquisition and microscopic images and uses a series of algorithms suitable for microscopic image stitching, so it can limit its single image stitching time to 1 second. This means the on-line microscopic image stitching. For the high precision of the SIFT feature based matching, this approach can give the sub-pixel stitching accuracy.

Key words: microscopic image, on-line image stitching, SIFT feature, image blending