

# 基于 SIFT 和 SURF 图像拼接算法的改进算法

史露<sup>1</sup> 苏刚<sup>2</sup> 韩飞<sup>1</sup>

<sup>1</sup>(武汉大学物理科学与技术学院 湖北 武汉 430072)

<sup>2</sup>(吉林大学计算机科学与技术学院 吉林 长春 130012)

**摘要** 目前,图像拼接技术已经成为数字图像处理、计算机图形学和人工智能等领域的研究热点。介绍基于 SIFT(Scale Invariant Feature Transform)特征点的图像拼接算法和基于 SURF(Speeded Up Robust Features)特征点的图像拼接算法,并且对这两个算法的性能做了比较,给出各自的优缺点。最后,基于这两种算法,提出一种加快图像拼接速度和质量的算法。

**关键词** SIFT 算法 SURF 算法 图像拼接

中图分类号 TP391 文献标识码 A DOI:10.3969/j.issn.1000-386x.2013.06.020

## AN IMPROVED ALGORITHM BASED ON SIFT AND SURF IMAGE STITCHING ALGORITHM

Shi Lu<sup>1</sup> Su Gang<sup>2</sup> Han Fei<sup>1</sup>

<sup>1</sup>(School of Physics and Technology, Wuhan University, Wuhan 430072, Hubei, China)

<sup>2</sup>(College of Computer Science and Technology, Jilin University, Changchun 130012, Jilin, China)

**Abstract** At present, image stitching technology has become the research focus in the fields of digital image processing, computer graphics and artificial intelligence. In this paper we introduce the image stitching algorithms based on SIFT (scale-invariant feature transform) feature points and SURF (Speeded Up Robust Features) feature points respectively, compare the performances of these two algorithms and give their respective strengths and weaknesses. In end of the paper, based on these two algorithms, we propose an algorithm which speeds up the image stitching with better quality.

**Keywords** SIFT algorithm SURF algorithm Image stitching

## 0 引言

图像拼接技术作为图像处理的热点问题吸引了很多国内外的研究者对其进行广泛而深入的研究。但是由于数字图像拼接技术涉及到很多学科的理论和技术,致使数字图像拼接技术仍然处于研究和发展的阶段。近些年来,为了提高数字图像拼接的质量和鲁棒性,国内外研究者进行了大量的基础性和实际开发工作,取得了大量的研究成果<sup>[1]</sup>。

图像拼接中最重要的部分是图像配准,图像配准算法分为两大类<sup>[2]</sup>:一类是基于区域的算法,其中包括基于空间的像素配准算法、基于频域的像素配准算法等;另一类是基于特征的算法,利用图像中的明显特征来计算图像之间的变换。而基于特征的算法由于其较低的复杂度和较高的鲁棒性,成为图像配准的首选,很多学者在这方面进行了研究。文献[3]方法使用基于灰度边缘特征点的拼接方法;文献[4]提出了一种基于角点特征的 KLT 跟踪全景图像拼接算法;文献[5]提出了一种基于像素点颜色变化特征的拼接算法;文献[6]采取基于像素点颜色变化特征和序列相似性检测算法相结合的图像拼接融合方法;文献[7]采用 SIFT 特征点进行图像拼接;文献[8]采用 SURF 特征点进行图像拼接。

本文结合 SIFT 特征点匹配算法和 SURF 特征点匹配算法的优点,提出了一种改进的图像拼接算法,实验结果显示,本文改进的算法实时性较高,可以得到比较好的拼接效果。

## 1 SIFT 算法和 SURF 算法

### 1.1 SIFT 算法

SIFT 的主要思路<sup>[9]</sup>是:首先建立图像的尺度空间表示,然后检测该图像尺度空间特征点,定义特征点主方向,最后生成特征向量描述子。

SIFT 算法检测尺度空间主要采用 DoG 算子,其中 DoG 算子定义如下:

$$\begin{aligned} D(x, y, \sigma) &= [G(x, y, k\sigma) - G(x, y, \sigma)] \cdot I(x, y) \\ &= L(x, y, k\sigma) - L(x, y, \sigma) \end{aligned} \quad (1)$$

其中  $L(x, y, \sigma) = G(x, y, \sigma) \cdot I(x, y)$ ,  $G(x, y, \sigma)$  为尺度可变高斯函数,  $I(x, y)$  为图像位于  $(x, y)$  的像素值。

DoG 函数在尺度空间的 Taylor 展开式的二级近似为:

$$D(X) = D + \frac{\partial D^T}{\partial X} X + \frac{1}{2} X^T \frac{\partial^2 D}{\partial X^2} X \quad (2)$$

其极值点为:  $\hat{X} = (x, y, \sigma)^T$ , 式(2)对图像的行、列及尺度三个量进行了修正, 得到修正值:

$$\hat{X} = - \frac{\partial D^T}{\partial X} \left( \frac{\partial^2 D}{\partial X^2} \right)^{-1} \quad (3)$$

将修正后的值带入 Taylor 展开式, 去除不稳定的极值点 (Lowe 的试验显示取值小于 0.04 的极值点均可抛弃)。求出精确关键点后, 就可以生成用于匹配的描述向量。利用关键点邻域像素的梯度方向分布特性, 为每个关键点指定方向参数方向, 从而使描述子对图像旋转具有不变性。

## 1.2 SURF 算法

SURF 算法由 Bay 等人<sup>[10]</sup>提出, 它主要分成 3 部分: 1) 特征点提取, 即在积分图像的基础上, 利用方框滤波近似代替二阶高斯滤波计算待选特征点及其周围点的 Hessian 值, 如果最大, 则为特征点。2) 特征点描述, 即在特征点周围小区域上计算 Harr 小波, 并计算其 4 种和以构成特征描述。3) 利用特征点描述向量进行配准。在一定程度上 SURF 算法是对 SIFT 算法的改进, 它也是一种尺度、旋转不变性的特征点探测器。其最大的特点就是快, 并且在快的基础上还能保持性能, 能满足实时性的拼接要求。

SURF 算法主要部分是 Hessian 矩阵的计算。Hessian 矩阵定义如下:

$$H(x, \sigma) = \begin{bmatrix} L_{xx}(x, \sigma) & L_{xy}(x, \sigma) \\ L_{yx}(x, \sigma) & L_{yy}(x, \sigma) \end{bmatrix} \quad (4)$$

然后计算其 Hessian 矩阵的判别式, 根据判别式取值正负, 来判别该点是否为极值点。因为 Hessian 矩阵的判别式计算复杂度较高, 所以利用方框滤波近似代替二阶高斯滤波计算待选特征点及其周围点的 Hessian 值, 这样可以利用积分图进行快速计算, 得到近似的判别值:

$$\det(H_{approx}) = D_{xx}D_{yy} - (\omega D_{xy})^2 \quad (5)$$

## 2 本文改进的算法

因为传统的 SIFT 算法能找到大量的特征点, 并且通过 RANSAC 能得到较准确的透视矩阵, 所以拼接效果很好, 但是它有一个显著的缺点, 即使采用 BBF 算法来加快搜索匹配速度, 它的复杂度依然很高。传统的 SURF 算法只能检测到少量的特征点, 还降低了特征点维数, 采用积分图方式能加快速度, 虽然算法复杂度低, 但是其拼接效果有时并不好。上述两个算法的运算时间中检测特征点的过程占据了大部分的时间, 此外 SIFT 的 BBF 搜索匹配算法比 SURF 算法要快, 而且 SIFT 中的 RANSAC 算法能得到较好的矩阵。

本文结合 SIFT 算法和 SURF 算法各自的优点, 提出了如下改进算法: 在求取特征点过程采用 SURF 算法, 由于每幅图像的独立性, 采用并行计算来提高速度, 在实验中采用多进程算法。对得到的 SURF 特征点采用 SIFT 的 BBF 算法进行快速匹配搜索, 最后采用 SIFT 的 RANSAC 算法求透视变换矩阵并进行拼接, 算法的主要步骤如下:

- 1) 采用并行算法求出每个图像的 SURF 角点;
- 2) 对每幅图像的特征点建立 KD 树来加快搜索匹配;
- 3) 利用 RANSAC 和 DLT 算法求出两幅图像间的匹配点数

量和转换矩阵;

4) 利用转换矩阵就可以将两幅图像映射成一幅全景图像。

本算法在 Visual C++ 2005 上利用 OpenCV 库编程实现, 算法的实现过程中求两幅图像的匹配点数量和。

转换矩阵的核心代码在下面给出:

```
int stich_image::matchHTrix (correspond &cor)
{
    clock_t st = clock();
    int count = 0;
    int nn = cor.match_point.size();
    int maxInlier = 0;
    vector< pair< Ipoint*, Ipoint* >> inlier;
    CvMat * Hmat = NULL;
    srand(time(NULL));
    while(count <= RANSACCOUNT)
    {
        int number = 0;
        vector< int > dex = randomRac(nn);
        vector< pair< Ipoint*, Ipoint* >> inlier1;
        CvMat * H = NULL;
        if(! inlier1.empty())
            inlier1.clear();
        CvPoint2D64f point1[4];
        CvPoint2D64f point2[4];
        for (int i=0; i<4; i++)
        {
            point1[i] = cvPoint2D64f(cor.match_point[dex[i]].first -
                >x, cor.match_point[dex[i]].first - >y);
            point2[i] = cvPoint2D64f(cor.match_point[dex[i]].second -
                >x, cor.match_point[dex[i]].second - >y);
        }
        H = lsq_homog(point1, point2, 4); //计算变换矩阵
        cvSave("D:\outMat.txt", H);
        for (int i=0; i<nn; i++)
        {
            double err;
            CvPoint2D64f pt1 = cvPoint2D64f(cor.match_point[i].first -
                >x, cor.match_point[i].first - >y);
            CvPoint2D64f pt2 = cvPoint2D64f(cor.match_point[i].second -
                >x, cor.match_point[i].second - >y);
            err = homog_xfer_err(pt1, pt2, H);
            //计算在变换矩阵下一对匹配点误差
            if (err < RANSACERROCTRL)
                //如果误差小的话说明这个点是内点, 一致点
            {
                inlier1.push_back(cor.match_point[i]);
                number++; //计算一致点数量
            }
        }
        if(number > maxInlier)
        {
            maxInlier = number;
            inlier = inlier1;
            Hmat = H;
        }
        count++;
    }
}
```

```

}
cor.mat = Hmat;
cor.inlier = inlier;
cor.inlier_number = maxInlier;
return maxInlier;
cout << "the " << m_dex << " image and " << cor.dex << " image
match mat time: " << clock() - st << endl;
}

```

### 3 实验结果及分析

算法实验环境为:AMD Athlon(tm) 64 X2 Dual Core Processor 3600 + 2000MHz CPU,1GB 内存,Windows XP 32 位操作系统。编程工具为 Visual Studio 2005。实验中所用的三张原始待处理图像是从网络上下载的 SIFT 角点检测算法的一组测试图片,图片大小为均为  $568 \times 758$ 。

#### 3.1 SIFT 算法拼接结果

图 1 所示的三张图像即为待处理的图像。通过 SIFT 算法检测图 1 中三张图像的尺度空间特征点,如图 2 所示。其中图 (a) 共找到 2948 个特征点,花费时间约 11.5s;图 (b) 共找到 3707 个特征点,花费时间约 12.8s;图 (c) 共找到 3251 个特征点,花费时间约 12s。图 3 是 SIFT 算法拼接得到的全景图像,实验中对三幅经过排序的图像进行拼接总共花费了约 52.7s 的时间。

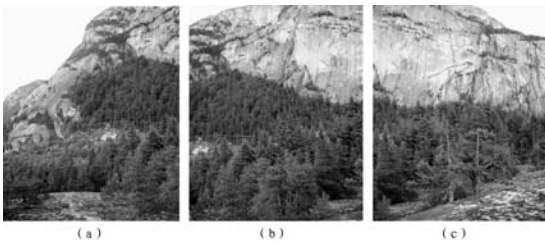


图 1 待处理的 3 张原图

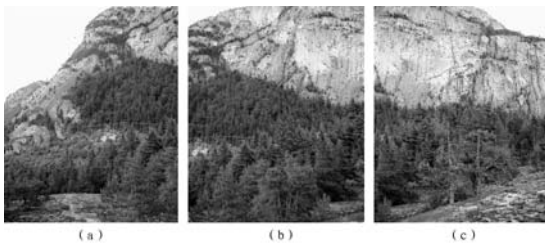


图 2 通过 SIFT 算法检测得到的 SIFT 特征点

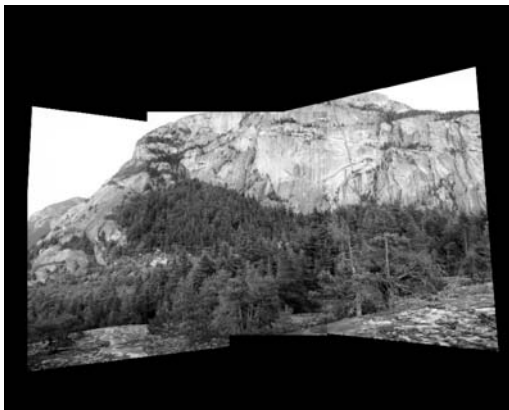


图 3 采用 SIFT 算法进行拼接得到的全景图

#### 3.2 SURF 算法拼接结果

采用 SURF 算法对图 1 中的 3 张原图检测尺度空间特征点,检测结果如图 4 所示。其中 (a) 共检测到 803 个特征点,花费时间约 5.8s,(b) 共检测到 874 个特征点,共花费约 5.8s,(c) 共检测到 849 个特征点,花费时间约 5.8s。由图 4 中 3 幅图像经过匹配等步骤得到最终的拼接图像,如图 5 所示。对 3 幅图像计算特征点总共花费的时间约为 17.4s,匹配时间约 2.5s,加上矩阵计算,整个完整的拼接过程大约花费约 25s。

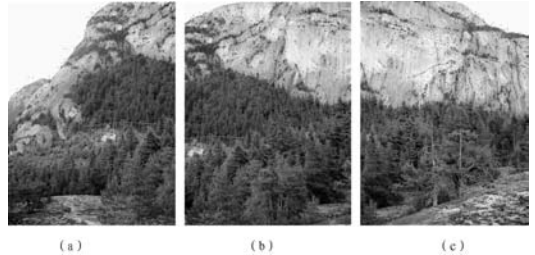


图 4 采用 SURF 算法检测得到的 SURF 特征点

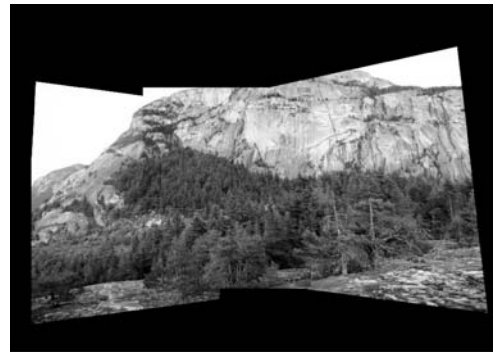


图 5 采用 SURF 算法进行拼接得到的全景图像

#### 3.3 本文算法拼接结果

在本文改进算法中,对 3 幅图像求特征点总共需要 11.6s,匹配时间约 1.2s,整个拼接过程所花费的时间约为 16s,图 6 为本文算法拼接得到的全景图像。从表 1 可以看出本文改进算法所花费的时间比 SIFT 算法和 SURF 算法都少,因此本文改进的算法具有较高的实时性。通过比较本文改进算法的拼接全景图和 SIFT 拼接全景图、SURF 拼接全景图,分别如图 6、图 3、图 5 所示,可以看出本文改进的算法相较于 SURF 算法拼接效果更好,与 SIFT 算法的效果没有太大差别,能得到较好的拼接效果。

表 1 各个拼接算法所花费的时间

SIFT 算法	SURF 算法	本文改进算法
52.7s	25s	16s



图 6 本文改进算法拼接得到的全景图

由表 8 得出与演员名字相关性高的前 3 个分面推荐:10-发行日期,11-评分,14-电影风格。

假如选择编号为 10 的分面,即发行日期,得到的精炼结果集按发行日期统计如表 9 所示。

表 9 按照发行日期统计得到的查询结果

电影名称	发行日期	$n_i(p)$
Da hong deng long gao gao gua	1991	1
Daihao meizhoubao	1989	1
Hong gao liang	1987	1
Huozhe	1994	1
Ju Dou	1990	1
Man cheng jin dai huang jin jia	2006	1
Qiu Jud a guan si	1992	1
Yao a yao yao dao waipo qiao	1995	1

选择发行日期值,就可以得到某一年份发行的影片信息。

为进一步说明本文方法的有效性,对第 2 个话题即与“周星驰”有关的影片进行了 XML 分面推荐。

输入“周星驰”后得到初步 xml 文档结果集,根据关键词“周星驰”所在的分面统计情况得出编号为 16 的分面覆盖率最大,选择编号为 16 的分面,即演员名称进行推荐。然后在得到的结果集中计算各个分面与演员名称的相关性  $xsd_{ij}$ ,得出与演员名称相关性高的前 5 个分面分别为:3-国家,4-语言,5-颜色,11-评分,14-风格;假如选择编号为 14 的分面,即风格,得到精炼结果集中风格的覆盖率值,推荐覆盖率值大的前 3 个影片风格:Comedy, Action, Drama;假如选择风格 = Comedy(喜剧),进一步得到精炼结果集,计算所有分面与风格的相关性  $xsd_{ij}$ ,得出与风格相关性高的分面有:3-国家,4-语言,11-评分;12-导演;13-编剧;如选择编号为 12 的分面即导演,得到精炼结果集中覆盖率最大的前 3 个导演分别是:周星驰,李力持,王晶。假如选择导演 = 李力持,进一步得到精炼结果集,计算所有分面与导演相关性  $xsd_{ij}$ ,得出与导演相关性高的分面有:1-运行时间,10-发行日期,11-评分,13-编剧,21-制片人。如选择编号为 10 的分面即发行日期,得到精炼结果集,如表 10 所示。

表 10 按照发行日期统计得到的查询结果

电影名称	发行日期	$n_i(p)$
Gwok chaan Ling Ling Chat ,Poh waai ji wong	(1994)	2
Hei kek ji wong	(1999)	1
Hung wan yat tew long	(1998)	1
Qing sheng	(1991)	1
Sik san	(1996)	1
Tang Bohu dian Qiuxiang	(1993)	1

由表 10 就可以查询某一年份发行的影片了。

实验结果表明,统计的方法在前几步分面推荐过程中推荐效果很好,但是在后续的推荐过程中就很难推荐出有效的分面了,而考虑 XML 分面之间的相关性就能够达到更好的效果。

## 4 结 语

本文对 XML 分面搜索中的分面推荐方法进行了研究,根据 XML 文档结构的语义特点,提出了 XML 分面的相关性定义,并将 XML 分面的相关性和传统的覆盖率方法结合进行 XML 分面推荐,实验结果表明本文方法有效地提高了 XML 分面的推荐效果。

## 参 考 文 献

- [1] 王莉,高仲利.基于分面导航理论的 RDF 数据的持久化研究[J].计算机工程与应用,2010,46(9):130-133.
- [2] 陈波.基于开源全文检索系统 Solr 的 OPAC 分面浏览[J].应用实践,2007,11:72-75.
- [3] Sacco, Giovanni M. Dynamic taxonomies and guided searches[J]. Journal of the American Society for Information Science and Technology 2006, 57(6):792-797, 2006.
- [4] S ebastien Ferre. Agile Browsing of a Document Collection with Dynamic Taxonomies [C]//19th International Conference on Database and Expert Systems Application. DOI 10.1109/DEXA.2008.28.
- [5] Sacco, Giovanni M. Uniform access to multimedia information bases through dynamic taxonomies [C]//IEEE 6th Int. Symp. On Multimedia Software Engineering, (ISMSE'04), 2004:320-328.
- [6] Niu Nan, Anas Mahmoud, Yang Xiaoyong. Faceted navigation for software exploration[C]//19th IEEE International Conference on Program Comprehension. IEEE,2011.
- [7] Jonathan Koren, Zhang Yi, Liu Xue. Personalized Interactive Faceted Search[C]//WWW 2008, April 2-125, 2008, Beijing, China.
- [8] Oren E, Delbru R, Decker S. Extending Faceted Navigation for RDF Data [C]//The SemanticWeb-ISWC 2006, LNCS 4273, 2006:559-572.

(上接第 74 页)

## 4 结 语

在基于 SIFT 算法和 SURF 算法的基础上,本文提出了一种改进算法,通过实验对比可以看出,本文的改进算法能克服 SIFT 和 SURF 的缺点,提高 SIFT 和 SURF 算法的运算速度,并且拼接效果也比较好,尤其是速度方面,具有很好的性能。拼接过程采用 RANSAC 算法求解最优匹配矩阵,有效减少了错拼。由于无人机对算法处理的实时性要求非常高,因此本文改进的算法可以应用于无人机地图全景拼接等方面。本文还没有进行图像拼接的其他步骤,如摄像机误差纠正和图像融合,这是以后工作的重点所在。

## 参 考 文 献

- [1] 邵向鑫.数字图像拼接核心算法研究[D].长春:吉林大学,2010.
- [2] 钟力,胡晓峰.重叠图像拼接算法[J].中国图象图形学报,1998,3(5):367-370.
- [3] 殷润民,李伯虎,柴旭东.自适应柱状全景图拼接[J].中国图象图形学报,2008,13(6):1191-1196.
- [4] 胡社教,葛西旺,陈宗海.基于角点特征的 KLT 跟踪全景图像拼接算法[J].系统仿真学报,2007,12(8):1742-1745.
- [5] 潘华伟,邹北骥.一种圆柱形全景图生成新算法及其实现[J].计算机工程与科学,2003,25(6):13-16.
- [6] 彭红星,邹湘军,徐东风,等.改进的全景图拼接融合算法[J].计算机工程与应用,2010,46(25):195-198.
- [7] Lowe D G. Object recognition from local scale-invariant features [C]//Proceeding of the seventh International Conference on Computer Vision (ICCV'99). Amsterdam:[s. n.],1999:1150-1157.
- [8] 张锐娟,张建奇,杨翠.基于 SURF 的图像配准方法研究[J].红外与激光工程,2008,38(1):160-165.
- [9] 周军太,龙永红.一种改进 SURF 算法的图像配准[J].湖南工业大学学报,2011,25(2):95-99.
- [10] 吴建波,赵建民,朱信忠,等.基于一种 SIFT 优化算法的图像检测[J].微型电脑应用,2011,27(5):4-7.