

# 基于角点变换矩阵的图像拼接

曹红杏<sup>1,2</sup> 柳稼航<sup>1</sup> 阮萍<sup>1</sup>(中国科学院西安光学精密机械研究所<sup>1</sup>, 西安 710119; 中国科学院研究生院<sup>2</sup>, 北京 100048)

**摘要** 图像拼接有广泛的应用领域。基于角点的图像拼接主要有角点提取、角点匹配、变换矩阵的计算和图像融合四步。先分析了平面图形的两像面的变换关系,然后以此为依据,根据匹配角点对,计算变换矩阵。最终将两幅图融合。实验结果证明,该方法实用可行。

**关键词** 图像拼接 角点 双线性插值

**中图法分类号** TP391.41; **文献标识码** A

图像拼接在航天、医学、工业等诸多领域都有广泛应用。而基于特征点(尤指角点)的图像拼接方法是最受关注的一类拼接方法。角点是图像中边界曲线上具有曲率极大值的点,或者说在该点灰度值会发生剧烈的变化<sup>[1]</sup>。它不易受光照、旋转等变换而发生改变,具有较强的稳定性,而且少量的角点不仅能反映出图像的重要信息,还有利于提高图像拼接速度。因此,许多人在从事这方面的研

究<sup>[2-4]</sup>。本文分析了平面图形的两像面之间的变换关系,并以此为依据,根据匹配角点对,计算出变换矩阵,最后将两幅图进行融合。

## 1 平面图形的两像面之间的变换关系

图1是两个CCD相机从不同角度进行拍摄的示意图。两个像平面也就是CCD像面, $Z$ 轴和 $Z'$ 轴是光轴方向。空间一点 $P$ 在两个像面上分别成

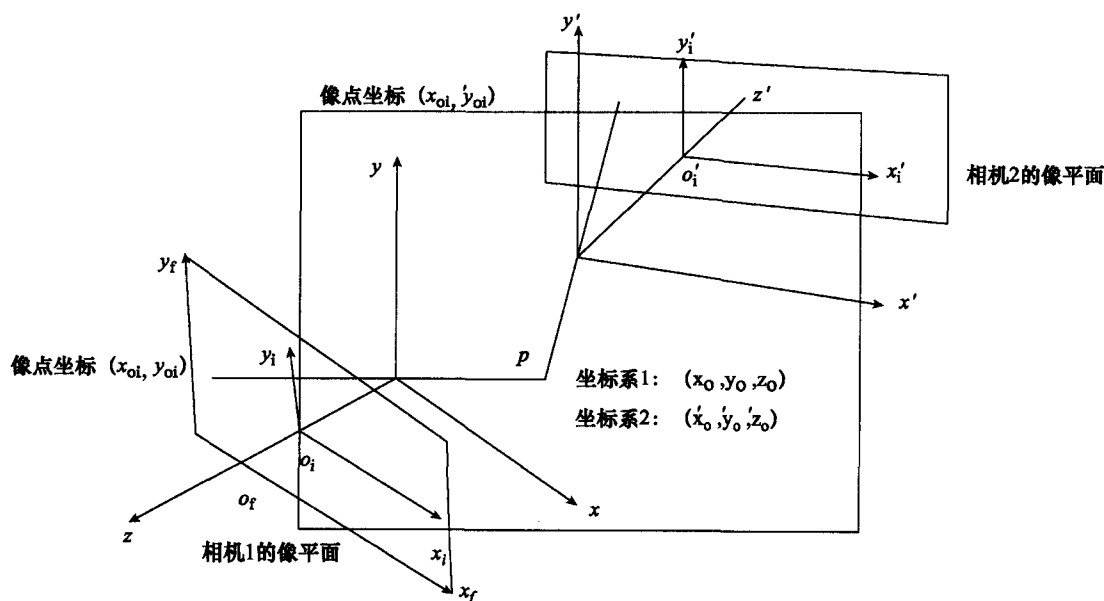


图1 两相机成像示意图

像为  $(x_{oi}, y_{oi})$  和  $(x'_{oi}, y'_{oi})$ 。相机成像过程可理想化为小孔成像,可用矩阵运算表示。例如,像点  $(x_{oi}, y_{oi})$  与 P 点之间满足关系:

$$\begin{pmatrix} x_{oi} \\ y_{oi} \\ 1 \end{pmatrix} = \begin{pmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_o \\ y_o \\ z_o \end{pmatrix}, \text{ 即: } x_{oi} = \frac{x_o f}{z_o}, y_{oi} = \frac{y_o f}{z_o} \quad (1)$$

(1)式中,  $(x_o, y_o, z_o)$  为 P 点在相机 1 的坐标系下的坐标,  $f$  为相机 1 的焦距。

同样地,相机 2 的像点与 P 点(相机 2 坐标系下的坐标)也存在这种变换关系。而两个相机的坐标系之间也存在一个矩阵变换,经过推导,可以得出:当拍摄的是平面图形(或者景深相对于物距足够小时),两个像面对应像点满足如下变换关系:

$$\begin{pmatrix} x'_{oi} \\ y'_{oi} \\ 1 \end{pmatrix} = \begin{pmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & 1 \end{pmatrix} \begin{pmatrix} x_{oi} \\ y_{oi} \\ 1 \end{pmatrix} \quad (2)$$

这种投影变换比一般平移,旋转,缩放,仿射等变换,有更广的适用范围。平移对应于矩阵中的  $m_{11} = m_{22} = 1, m_{12} = m_{21} = m_{31} = m_{32} = 0$  的情况,  $m_{13}, m_{23}$  为平移量;旋转对应于  $m_{13}, m_{23}, m_{31}, m_{32}$  为零,左上角是一个旋转矩阵的情况;而仿射变换则是  $m_{31} = m_{32} = 0$  的情况。由于本文的变换矩阵有更强的适应性,因此对拍摄拼接图像相机的运动的限制减少了。

根据式(2)可知,若要将不同角度拍摄的平面物体(或者景深相对物距较小的物体)的两幅(下文简称这两幅图像为图像 1,图像 2)图像拼接在一起,则需要根据 4 对匹配点,得到变换矩阵中的  $m_{11} - m_{32}$  等共 8 个参数,将图像 1 投影到图像 2 平面上,并与之融合。

## 2 变换矩阵的计算

由(2)式可得:

$$x'_{oi} = \frac{m_{11}x_{oi} + m_{12}y_{oi} + m_{13}}{m_{31}x_{oi} + m_{32}y_{oi} + 1} \quad (3)$$

$$y'_{oi} = \frac{m_{21}x_{oi} + m_{22}y_{oi} + m_{23}}{m_{31}x_{oi} + m_{32}y_{oi} + 1} \quad (4)$$

$$m_{11}x_{oi} + m_{12}y_{oi} + m_{13}, -m_{31}x_{oi}x'_{oi} - m_{32}y_{oi}x'_{oi} = x'_{oi} \quad (5)$$

$$m_{21}x_{oi} + m_{22}y_{oi} + m_{23} - m_{31}x_{oi}y'_{oi} - m_{32}y_{oi}y'_{oi} = y'_{oi} \quad (6)$$

一对匹配点(即式中的  $(x_{oi}, y_{oi})$  和  $(x'_{oi}, y'_{oi})$  都为已知数)可建立 2 个方程,4 对匹配点(不存在三点共线)就可得到如下(7)所示方程组,解方程即得  $m_{11} - m_{32}$  等共 8 个参数。

$$\begin{pmatrix} x_{oi}^1 & y_{oi}^1 & 1 & 0 & 0 & 0 & -x_{oi}^1 x_{oi}^{1'} & -y_{oi}^1 x_{oi}^{1'} \\ 0 & 0 & 0 & x_{oi}^1 & y_{oi}^1 & 1 & -x_{oi}^1 y_{oi}^{1'} & -y_{oi}^1 y_{oi}^{1'} \\ x_{oi}^2 & y_{oi}^2 & 1 & 0 & 0 & 0 & -x_{oi}^2 x_{oi}^{2'} & -y_{oi}^2 x_{oi}^{2'} \\ 0 & 0 & 0 & x_{oi}^2 & y_{oi}^2 & 1 & -x_{oi}^2 y_{oi}^{2'} & -y_{oi}^2 y_{oi}^{2'} \\ x_{oi}^3 & y_{oi}^3 & 1 & 0 & 0 & 0 & -x_{oi}^3 x_{oi}^{3'} & -y_{oi}^3 x_{oi}^{3'} \\ 0 & 0 & 0 & x_{oi}^3 & y_{oi}^3 & 1 & -x_{oi}^3 y_{oi}^{3'} & -y_{oi}^3 y_{oi}^{3'} \\ x_{oi}^4 & y_{oi}^4 & 1 & 0 & 0 & 0 & -x_{oi}^4 x_{oi}^{4'} & -y_{oi}^4 x_{oi}^{4'} \\ 0 & 0 & 0 & x_{oi}^4 & y_{oi}^4 & 1 & -x_{oi}^4 y_{oi}^{4'} & -y_{oi}^4 y_{oi}^{4'} \end{pmatrix} \times \begin{pmatrix} m_{11} \\ m_{12} \\ m_{13} \\ m_{21} \\ m_{22} \\ m_{23} \\ m_{31} \\ m_{32} \end{pmatrix} = \begin{pmatrix} x_{oi}^{1'} \\ y_{oi}^{1'} \\ x_{oi}^{2'} \\ y_{oi}^{2'} \\ x_{oi}^{3'} \\ y_{oi}^{3'} \\ x_{oi}^{4'} \\ y_{oi}^{4'} \end{pmatrix} \quad (7)$$

(7)式中  $x_{oi}^m, y_{oi}^m$  为图像 1 上第  $m$  点的横、纵坐标,  $x_{oi}^{m'}, y_{oi}^{m'}$  为图像 2 上第  $m$  点的横、纵坐标。

方程组求解采用将系数矩阵 LU 分解的方法,最终计算出 8 个参数。

## 3 图像的变换合成

理论上,得到  $m_{11} - m_{32}$  8 个参数后,可按式(2)将图像 1 上的点投影到图像 2 的平面上,投影区域为  $S$ 。将  $S$  与图像 2 合并即得到拼接图像。但如果这样,图像 1 上的整数点坐标将投影为小数坐标,用这些疏密不均的小数坐标点的灰度值来重构区域  $S$  的整数点坐标较复杂。本文先根据变换矩阵,计算出图像 1 在图像 2 的平面上的投影区域  $S$ ,将拼接

图像分成 A, B, C, D 四部分, 对四部分的灰度值分别进行计算。具体如下:

A. 在投影区  $S$  内, 但不落在图像 2 内的点。

1) 采用后向映射法<sup>[5]</sup>, 按式(8), 式(9)依次计算区域  $S$  内的所有整数点坐标  $(x', y')$  对应图像 1 的坐标点  $(x, y)$ 。

$$x = \frac{bf - ce}{bd - ae} \quad (8) \quad y = \frac{cd - af}{bd - ae} \quad (9)$$

其中  $a = m_{31}x' - m_{11}$ ,  $b = m_{32}x' - m_{12}$ ,  $c = m_{13} - x'$ ,  $d = m_{31}y' - m_{21}$ ,  $e = m_{32}y' - m_{22}$ ,  $f = m_{23} - y'$ 。

2) 按(10)式进行双线性插值计算图像 1  $(x, y)$  点灰度值, 将其作为  $(x', y')$  的灰度值。

$$I_1(x, y) = u_2(u_1 I_1([x] + 1, [y] + 1) + (1 - u_1) I_1([x], [y] + 1)) + (1 - u_2)(u_1 I_1([x] + 1, [y]) + (1 - u_1) I_1([x], [y])) \quad (10)$$

其中  $[ ]$  为取整符号;  $u_1 = x - [x]$ ;  $u_2 = y - [y]$ ;  $I_1(m, n)$  表示图像 1 在  $(m, n)$  点的灰度值。

B. 在区域  $S$  和图像 2 的重叠区域, 且落在图像 2 边缘 10 个像素(过渡区域的宽度)范围内的点, 将该点对应于图像 1 上的点的灰度值(计算方法同 A 类点)与图像 2 上的点的灰度值进行加权平滑, 计算公式为:  $I_3(x', y') = I_1(x, y)w + I_2(x', y')(1 - w)$ ,  $1 > w > 0$ ,  $w$  为权, 与该点距离边缘的远近有关。使用加权平滑, 可以使拼接过渡比较自然。过渡区域的宽度可以根据实际情况调整。

C. 在图像 2 内, 且不属于 B 类的点, 直接将其灰度值直接拷入拼接图。

D. 即不在区域  $S$  内, 又不落在图像 2 内的点, 将其灰度设置为 0。

## 4 实验结果

本次试验的待拼接图像由 kodak Z650 相机手持拍摄图片, 经程序转换后的 256 灰度图。使用 VC++ 6.0 实现图像变换矩阵的计算和拼接。

图 3 中绿色框的中心即为使用 SIFT 算法提取得到的匹配角点。从拼接结果图 3 来看, 对于两幅视角有一定差距, 且有一定缩放的图像, 程序拼接结果较自然, 无明显拼接缝。为了进一步提高拼接质量, 今后还需在提高特征点定位精度、改善特征点匹配算法和使用基于小波的图像融合等方面进行探索。

## 5 本文的创新点

本文首先分析两图像之间的变换为依据, 采用的图像变换矩阵有较广的适应范围, 比文献[6—8]中提出的仿射变换的范围有所的扩展。本文在图像融合过程中, 使用后向映射, 由双线性插值得到小数点灰度值的方法, 并采用加权融合, 使重叠区域过渡较为自然。从试验结果看, 效果较好。

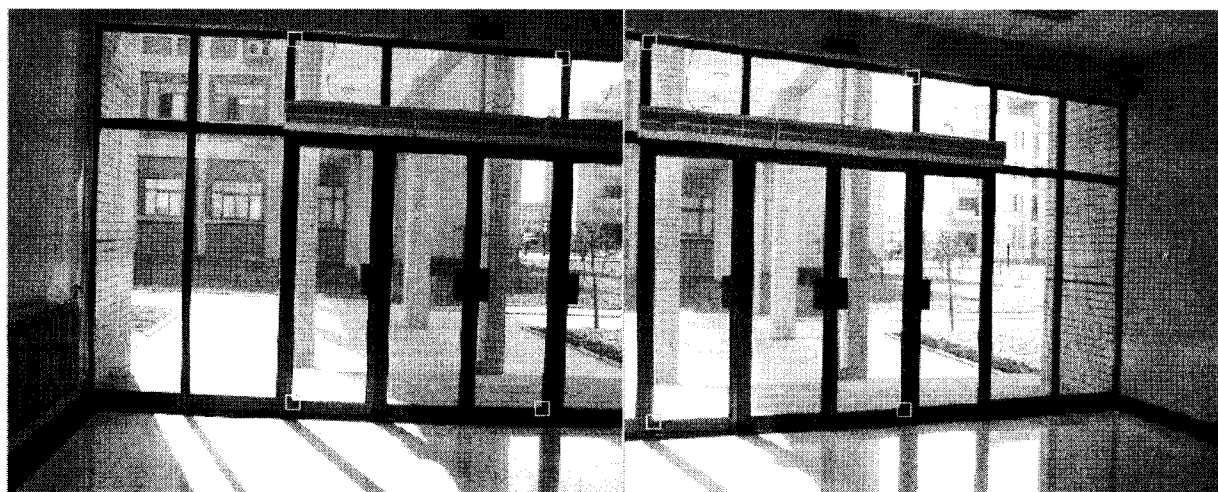


图 2 待拼接图



图3 拼接结果

## 参 考 文 献

- 1 李玉山. 数字视觉视频技术. 西安: 西安电子科技大学出版社, 2006
- 2 陈 静. 图像配准特征点提取算法研究. 南京理工大学, 2006
- 3 汪华琴. 基于特征点匹配的图像拼接方法研究. 华中师范大学, 2007
- 4 陈利军. 图像角点检测和匹配算法得研究. 西安电子科技大学,

2005

- 5 李晓娟. 图像拼接技术. 西安电子科技大学, 2006
- 6 郝明非, 张建秋, 胡 波. 基于快速鲁棒相关和粒子滤波器的图像拼接方法. 信息与电子工程, 2006; 4(3): 191—195
- 7 刘德连, 张建奇. 基于3点匹配的图像拼接算法. 计算机工程. 2006, 第32卷第13期: 203—205
- 8 贺 彬, 王国宇. 不同视点海底图像拼接算法. 微计算机信息, 2005; 12-3: 152—154

## Stitching Image Based on Transform Matrix of Corners

CAO Hong-xing<sup>1,2</sup>, RUAN Ping<sup>1</sup>, LIU Jia-hang<sup>1</sup>(Xi'an Institute of optics and Precision Mechanism of CAS<sup>1</sup>, Xi'an 710068, P. R. China;Graduate University of Chinese Academy of Sciences<sup>2</sup>, Beijing 100048, P. R. China)

[Abstract] Image stitching has been widely used in many fields. Image stitching based on feature point is composed of four steps: finding corners, matching corners, computing transform matrix and mixing image. It is introduced that the practicable way to finish the last two steps. The experiment result also is given. According to experiment results, this method is practicable.

[Key words] feature point    mixing image    approximation by double linear