

基于像素模型的CT仿真投影快速计算

张顺利^{1,2} 张定华² 程云勇² 张小波¹

(咸阳师范学院图形图像处理研究所 咸阳 712000)¹

(西北工业大学现代设计与集成制造技术教育部重点实验室 西安 710072)²

摘要 提高仿真投影的计算速度一直是CT仿真技术的重要研究内容。针对像素模型的投影计算,提出了一种快速仿真投影生成算法。首先确定射线与像素模型边界相交的起始像素索引及纵向距离,然后通过该纵向距离依次计算射线穿过的像素索引及相交长度,同时累加出投影数据。实验结果表明,该算法大大提高了投影计算速度,与Siddon算法相比取得了约7倍的加速比;图像重建结果也证明了该算法的正确性。

关键词 仿真投影,图像重建,Siddon算法,像素模型

中图分类号 TP391 文献标识码 A

Fast Calculation of CT Simulation Projection Based on Pixel Phantom

ZHANG Shun-li^{1,2} ZHANG Ding-hua² CHENG Yun-yong² ZHANG Xiao-bo¹

(Institute of Graphics and Image Processing, Xianyang Normal University, Xianyang 712000, China)¹

(Key Laboratory of Contemporary Design & Integrated Manufacturing Technology, Ministry of Education, Northwestern Polytechnical University, Xi'an 710072, China)²

Abstract Improving the calculation speed of simulation projection has been the important research of CT (computed tomography) simulation technique. Aiming at the projection calculation of pixel phantom, a fast algorithm of generating simulation projection was proposed. Firstly, the original pixel index and its vertical distance intersected by the ray and the pixel phantom boundary were determined. Then, the other pixel index and intersection length intersected by the ray were calculated in turn using the distance, the projection data was obtained from cumulative sum simultaneously. Experimental result shows that the proposed algorithm can improve the calculation speed of projection greatly, we can get a speedup about 7 times compared with the Siddon algorithm. The reconstructed result also shows the proposed algorithm is correct.

Keywords Simulation projection, Image reconstruction, Siddon algorithm, Pixel phantom

1 引言

CT技术是一种先进的无损检测手段,广泛应用于医学和工业领域。在CT成像系统的设计、参数优化及图像重建算法性能等的研究中,通常采用仿真技术。该技术提供了一个廉价、灵活的虚拟试验环境,可以满足科研与生产的不同需求。目前通过计算机仿真生成投影数据的方法可分为解析法^[1,2]、随机法^[3]和离散法^[4]。解析法的特点是计算精确,但不够灵活,每当选用一种新模型就要重新设计投影计算方法,而且通常只能模拟几种密度均匀的物体。随机法的特点是能够逼真反映X射线成像的物理过程,但其算法设计难度较高,计算时间较长。离散法的特点是灵活性高,可以模拟内部密度不均匀的物体,此外还可以有效地验证图像重建算法的正确性,评估重建图像的质量。

在CT仿真研究中,提高仿真投影计算速度一直是研究

的重要内容。针对离散像素模型,提出了一种快速仿真投影计算算法,并通过实验对文中算法的有效性进行了验证。

2 仿真投影计算

在仿真投影计算中,通常假设X射线为理想的单能量射线,此时投影数据在数值上等于物体密度分布函数沿射线路径的线积分^[5]。下面以二维像素模型为对象,来研究仿真投影的计算。

考虑由 $n \times n$ 个像素组成的离散模型,在每个像素内部的密度值均匀分布,记为 $f(i, j)$, $0 \leq i < n$, $0 \leq j < n$ 。该像素模型的仿真投影数据 $proj$ 可表示为

$$proj = \int_L f(i, j) dl = \sum_{i,j} l(i, j) f(i, j) \quad (1)$$

式中, L 为射线路径, $l(i, j)$ 为射线与像素 (i, j) 的相交长度。可以看出,仿真投影计算的实质是射线与像素的遍历和求交

到稿日期:2010-12-07 返修日期:2011-01-22 本文受陕西省自然科学基金(2009JQ8017),陕西省教育厅专项基金(11JK1030),咸阳师范学院引进人才项目(10XSYK102, 07XSYK266)资助。

张顺利(1973—),男,博士后,副教授,主要研究方向为工业CT、计算机图形图像处理, E-mail: slmmzhang@sina.com; 张定华(1958—),男,博士,教授,博士生导师,主要研究方向为CAD/CAM、工业CT和计算机图形图像处理等; 程云勇(1976—),男,博士后,副教授,主要研究方向为工业CT、计算机图形图像处理; 张小波(1979—),男,博士生,主要研究方向为图像处理。

问题,而求交算法的优劣直接关系到投影的计算效率。

Siddon 算法^[6]是一种典型的射线与像素的遍历和求交算法,其原理如图 1 所示。

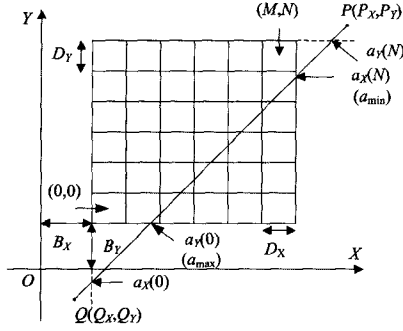


图 1 Siddon 算法示意图

在 Siddon 算法中,使用参数 α 来指示射线上的点 (x, y) 到起点 P 的距离,其 x, y 坐标可表示为

$$\begin{aligned} x(\alpha) &= P_x + \alpha(Q_x - P_x) \\ y(\alpha) &= P_y + \alpha(Q_y - P_y) \end{aligned} \quad (2)$$

式中, P_x, P_y, Q_x 和 Q_y 分别表示起点 P 和终点 Q 的坐标分量。反过来,当已知射线上一个点的坐标分量 x 和 y 时,也可计算出对应的参数 α 。

算法的第 1 步是计算射线穿过图 1 所示像素区域的入点和出点的 α 值。定义像素模型中系列水平网格线与垂直网格线相交的交点 α 值为

$$\begin{aligned} \alpha_x(I) &= (I \cdot D_x + B_x - P_x) / (Q_x - P_x) \\ \alpha_y(J) &= (J \cdot D_y + B_y - P_y) / (Q_y - P_y) \end{aligned} \quad (3)$$

于是得到

$$\begin{aligned} \alpha_{\min} &= \max\{\min\{\alpha_x(M), \alpha_x(0)\}, \min\{\alpha_y(N), \alpha_y(0)\}\} \\ \alpha_{\max} &= \min\{\max\{\alpha_x(M), \alpha_x(0)\}, \max\{\alpha_y(N), \alpha_y(0)\}\} \end{aligned} \quad (4)$$

式中, $\alpha_{\min}$ 和 $\alpha_{\max}$ 分别为射线穿过像素模型的入点和出点的 α 值。对介于 $\alpha_{\min}$ 和 $\alpha_{\max}$ 之间的其他点的 α 值,可以分为一系列垂直网格线与射线相交的 α_x 值和一系列水平网格线与射线相交的 α_y 值。然后计算入点 $\alpha_{\min}$ 所在像素的网格线索引 $i_{\min}$ 和 $j_{\min}$ 以及出点 $\alpha_{\max}$ 所在像素的网格线索引 $i_{\max}$ 和 $j_{\max}$ 。再计算其他点的 α 值,并且保存在两个数组 $\{\alpha_x\}$ 和 $\{\alpha_y\}$ 中。其中 $\{\alpha_x\} = \{\alpha_x(i_{\min}), \alpha_x(i_{\min}+1), \dots, \alpha_x(i_{\max})\}$, $\{\alpha_y\} = \{\alpha_y(j_{\min}), \alpha_y(j_{\min}+1), \dots, \alpha_y(j_{\max})\}$ 。最后将这两个数组排序,合并为一个有序数组 $\{\alpha\} = \{\alpha(0), \alpha(1), \dots, \alpha(m)\}$ 。经过上述步骤之后,计算射线穿过的像素索引及长度,像素的水平和垂直索引为

$$\begin{aligned} i &= \lfloor P_x + \alpha_{mid}(Q_x - P_x) - B_x / D_x \rfloor \\ j &= \lfloor P_y + \alpha_{mid}(Q_y - P_y) - B_y / D_y \rfloor \end{aligned} \quad (5)$$

式中, $\alpha_{mid} = [a(h) + a(h-1)]/2, 0 \leq h \leq m$ 。相应的相交长度为 $l(i, j) = (a(h) - a(h-1)) \cdot |PQ|$, 由式(1)便可计算出投影数据。

由上述过程可以看出, Siddon 算法在合并数组 $\{\alpha_x\}$ 和 $\{\alpha_y\}$ 中需要进行大量的比较运算, 计算像素索引又涉及大量的乘法、除法以及取整运算, 因而大大降低了该算法的效率。为此, 本文提出一种高效的射线与像素的遍历和求交算法来提高仿真投影的计算效率。

3 快速算法

以像素模型的中心为原点建立平面坐标系, 像素边长为 δ , 其左下角顶点坐标为 (x_0, y_0) , 像素索引 (i, j) 的安排如图 2 所示。射线的方程为 $y = kx + b$, 不失一般性, 我们仅考虑斜率 k 在 $(0, 1]$ 时的情形。

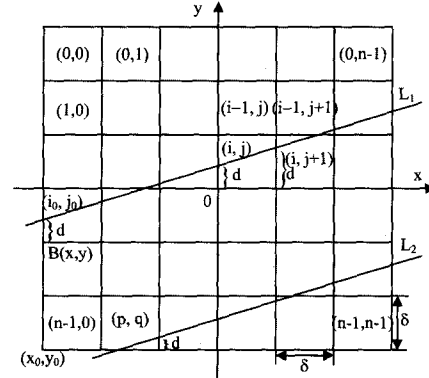


图 2 快速算法示意图

首先确定射线与模型边界相交的起始像素索引。在斜率 $0 \leq k \leq 1$ 时, 射线如果与像素模型相交, 则必然与其左边界和底边界之一相交。

如果射线与像素模型的左边界相交, 即满足 $kx_0 + b \geq y_0$, 如直线 L_1 所示。则起始像素 (i_0, j_0) 的索引计算为

$$i_0 = \lfloor [-y_0 - (kx_0 + b)] / \delta \rfloor, \quad j_0 = 0 \quad (6)$$

此时射线与像素 (i_0, j_0) 左下角顶点的纵向距离 d 可表示为

$$d = kx_0 + b + y_0 + (i_0 + 1)\delta \quad (7)$$

如果射线与像素模型的底边界相交, 即满足 $kx_0 + b < y_0$, 如直线 L_2 所示。则像素 (p, q) 的索引计算为

$$\begin{aligned} p &= n - 1 \\ q &= \lfloor [(y_0 - b) / k - x_0] / \delta \rfloor \end{aligned} \quad (8)$$

此时射线与像素 (p, q) 的相交长度为

$$l(p, q) = (x_0 + (q + 1)\delta - \frac{y_0 - b}{k}) \sqrt{k^2 + 1} \quad (9)$$

然后将像素 $(p, q + 1)$ 作为起始像素, 射线与该像素左下角顶点之间的纵向距离 d 为

$$d = kx_0 + (q + 1)\delta k - y_0 + b \quad (10)$$

接下来根据起始像素索引和纵向距离 d 来依次确定与射线相交的像素索引及相交长度。假如目前进行到了像素 (i, j) , 射线与其左下角顶点的纵向距离为 d , 则与其右下角顶点的纵向距离 $d = d + k\delta$ 。考虑以下 3 种情形:

如果 $d > \delta$, 则射线依次穿过像素 (i, j) 和像素 $(i - 1, j)$, 其相交长度分别为

$$\begin{aligned} l(i, j) &= (k\delta + \delta - d) \sqrt{k^2 + 1} / k \\ l(i - 1, j) &= \delta \sqrt{k^2 + 1} - l(i, j) \end{aligned} \quad (11)$$

下一个要计算的像素为 $(i - 1, j + 1)$, 相应的 $d = d - \delta$ 。

如果 $d < \delta$, 则射线穿过像素 (i, j) , 其相交长度为

$$l(i, j) = \delta \sqrt{k^2 + 1} \quad (12)$$

下一个要计算的像素为 $(i, j + 1)$ 。

如果 $d = \delta$, 则射线穿过像素 (i, j) , 其相交长度为

$$l(i, j) = \delta \sqrt{k^2 + 1} \quad (13)$$

下一个要计算的像素为 $(i - 1, j + 1)$, 相应的 $d = 0$ 。

注意在遍历过程中, 一旦遇到某个像素不在像素模型的

区域之内,则停止计算。下面给出射线斜率 k 在 $(0,1]$ 时的算法描述。

输入:射线的端点 $P(x_1, y_1), Q(x_2, y_2)$ 。

输出:仿真投影数据 $proj$ 。

Step1 初始化。计算射线方程 $y=kx+b$ 的基本参数,其中 $k=(y_2-y_1)/(x_2-x_1), b=y_1-kx_1$ 。置投影初值 $proj$ 为 0。

Step2 确定射线与模型边界相交的起始像素索引及相应的纵向距离 d 。如果 $kx_0+b \geq y_0$, 则 $i=floor((-y_0-(kx_0+b))/\delta), j=0, d=kx_0+b+y_0+(i+1)\delta$; 否则 $i=n-1, j=floor(((y_0-b)/k-x_0)/\delta), length=sqrt(k^2+1) \times (x_0+(j+1)\delta-(y_0-b)/k), proj=proj+length \times f(i, j), j=j+1, d=(kx_0+(j+1)k\delta-y_0+b)$ 。

Step3 $d=d+k\delta$ 。

Step4 如果 $d > \delta$, 且满足 $i \geq 0, j < n$, 则 $length=(k\delta+\delta-d) \times sqrt(k^2+1)/k, proj=proj+length \times f(i, j), i=i-1$; 否则结束。如果 $i \geq 0$, 则 $length=sqrt(k^2+1) \times \delta-length, proj=proj+length \times f(i, j), j=j+1, d=d-\delta$, 转 Step3; 否则结束。如果 $d < \delta$, 且满足 $i \geq 0, j < n$, 则 $length=sqrt(k^2+1) \times \delta, proj=proj+length \times f(i, j), j=j+1$, 转 Step3; 否则结束。如果 $d=\delta$, 且满足 $i \geq 0, j < n$, 则 $length=sqrt(k^2+1) \times \delta, proj=proj+length \times f(i, j), i=i-1, j=j+1$, 转 Step3; 否则结束。

当射线为水平或垂直时,计算过程更简单。以水平射线为例,参照上述方法计算起始像素索引 (i, j) , 则投影值的计算过程为

for($j=0; j < n; j++$)

$proj=proj+\delta \times f(i, j)$

对于斜率为其他值的情形,可参考上述算法来处理。需要指出的是,本算法在执行过程中,对于每一条射线, $k\delta$ 和 $sqrt(k^2+1) \times \delta$ 的值都不变,因此可以作为常量。同 Siddon 算法相比,本文算法不需开辟额外的内存空间,投影数据在遍历求交的同时累加得到,也避免了复杂的排序操作。

4 实验结果与分析

为了验证文中所提算法的运算效率,选用标准 Shepp-Logan 头模型^[7],将其离散化后的图像作为像素模型,如图 3(a)所示。

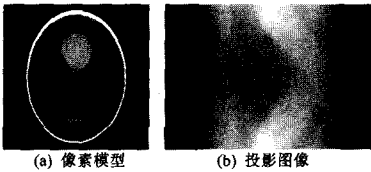


图 3 Shepp-Logan 模型及其投影图像

实验中,对 Shepp-Logan 头模型按不同分辨率离散化,分别采用 Siddon 算法和本文算法来计算不同分辨率下像素模型的仿真投影数据,投影视角均为 360,每个投影下的射线数和图像分辨率相同。图 3(b)为该模型 512×512 分辨率下的投影图像。测试计算机配置为 Intel 酷睿 I5/760 2.80GHz 四核 CPU,内存为 4G,操作系统为 Microsoft Windows XP,开发工具采用 VC++6.0。

表 1 给出了几种不同分辨率下,Siddon 算法和本文算法

计算仿真投影的时间对比。

表 1 仿真投影计算时间比较(单位:s)

投影算法	n×n 像素模型,360 个投影视角,n 条射线			
	n=128	n=256	n=512	n=1024
Siddon 算法	0.956	3.813	15.298	61.241
本文算法	0.137	0.542	2.165	8.653

由表 1 可以看出,本文算法较 Siddon 算法,取得了约 7 倍的加速比。其主要原因在于本文算法中,除前两步涉及到一些乘、除运算外,后面大部分计算为比较和加、减运算,因而运算效率大大提高。

为了验证本文算法的正确性,基于本文算法生成的投影数据,利用滤波反投影图像重建算法对其进行重建。重建图像的精度评估一般有两种方法:一种是采用 Profile 曲线法来比较原始模型和重建图像对应的灰度值。具体方法为:选择原始模型和重建图像同一行或列的像素,在直角坐标系中分别做出这一行或列的灰度变化曲线,比较原始模型和重建图像灰度曲线的接近程度,就可以直观地对重建图像的质量做出判断。图 4(a)为重建图像,图 4(b)为原始模型及重建图像中间一行的 Profile 曲线图。

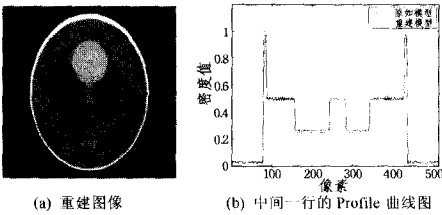


图 4 重建结果

另一种方法是采用原始模型与重建图像的 3 个距离评价参数^[8]:归一化均方距离判据 d ,归一化平均绝对距离判据 r 和最坏情况距离判据 e 。其中, d 反映了少数点出现较大偏差的情况, r 反映了多数点出现小的偏差的情况, e 反映了原始图像和重建图像之间的最大密度偏差。表 2 给出了不同分辨率下的重建误差。

表 2 重建误差

评价参数	n×n 图像			
	n=128	n=256	n=512	n=1024
d	0.293828	0.236093	0.208314	0.228175
r	0.203841	0.162885	0.146866	0.181876
e	0.288478	0.209181	0.217142	0.226177

由图 4 和表 2 的结果可以看出,本文算法生成的仿真投影数据较好地重建了原始模型,因此文中投影生成算法是正确的。

结束语 本文针对离散像素模型提出了一种快速仿真投影生成算法,实验结果验证了本文方法的有效性和准确性。该方法为图像重建算法研究提供了有力的工具。近年来,锥束 CT 重建理论逐渐成为研究热点^[9],因此研究基于离散像素模型的快速投影生成算法是下一步要解决的问题。此外,本文仅考虑了理想单能量射线的投影计算,实际的射线为多能谱的,且存在 Compton 散射和射束硬化等物理现象,如何更逼真地对实际投影进行仿真也是今后的研究内容。

参考文献

[1] Tang Shao-jie, Yu Heng-yong, Yan Hao, et al. X-ray projection

simulation based on physical imaging model [J]. Journal of X-Ray Science and Technology, 2006, 14(3): 177-189

- [2] 孙丰荣, 刘泽, 李艳玲, 等. 基于模型的 CT 三维医学图像重建仿真[J]. 系统仿真学报, 2006, 18(3): 781-784
- [3] 闫浩, 牟轩沁, 罗涛, 等. 锥束 X 射线 CT 投影的蒙特卡罗仿真[J]. 西安交通大学学报, 2008, 42(4): 414-417
- [4] 秦中元, 牟轩沁, 王平, 等. 一种通用的 X 射线锥形束投影生成算法[J]. 西安交通大学学报, 2002, 36(2): 160-164
- [5] 黄魁东, 张定华, 毛海鹏, 等. 理想材料零件的锥束 CT 投影仿真

[J]. 计算机辅助设计与图形学学报, 2007, 19(2): 157-162

- [6] Siddon R L. Fast calculation of the exact radiological path for three-dimensional CT array [J]. Med Phys, 1985, 12: 252-255
- [7] Shepp L A, Logan B F. The Fourier reconstruction of a head section [J]. IEEE Trans Nucl Sci, 1974, NS-21: 21-43
- [8] 庄天戈. CT 原理与算法[M]. 上海: 上海交通大学出版社, 1992
- [9] 张顺利, 张定华, 李明君, 等. 基于最小圆柱区域的锥束 CT 快速图像重建[J]. 计算机科学, 2010, 37(5): 257-260

(上接第 279 页)

表 2 是引入限制因子的 Harris 角点和 Harris 角点的邻域彩色相关匹配结果比较。

表 2 Harris 角点和引入限制因子后角点的邻域彩色匹配

	Harris 角点匹配	引入限制因子后角点匹配
匹配点对	1695	762
正确匹配点对	1448	709
匹配正确率	85.43%	93.04%

从表 2 的结果看, 引入限制因子后, 特征点匹配的精度明显提高, 由于角点的数量明显减少, 因此能够有效地提高立体匹配效率。

使用斜率一致法剔除误匹配点对的效果如图 7 所示。对比图 7(a) 和 (b) 可以看出一些错误明显的匹配点对被剔除, 这样后期 RANSAC 算法的迭代次数可能减少, 也为 RANSAC 算法估计正确的基础矩阵提供了一定的保证。



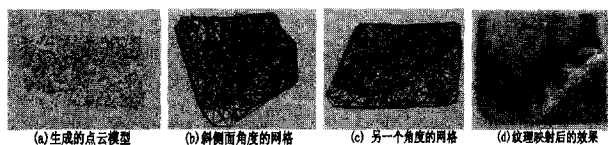
(a) 斜率一致法剔除前 (b) 斜率一致法剔除后

图 7 斜率一致法剔除误匹配点的效果图

运用 RANSAC 算法估计的基础矩阵 F 如下

$$F = \begin{bmatrix} 0.000005 & 0.000010 & -0.026527 \\ -0.000010 & 0.000000 & 0.006872 \\ 0.024594 & -0.007761 & 1.000000 \end{bmatrix}$$

在估计基础矩阵的同时进一步剔除误匹配点对, 并利用估计的基础矩阵引导未匹配点进行匹配, 得到最终的匹配点对。为了验证提出的匹配算法在三维重建中的有效性, 我们把本文算法得到的匹配点对利用 SFM 重建算法^[9,10]进行重建, 重建的效果如图 8 所示。



(a) 生成的点云模型 (b) 斜侧面角度的网格 (c) 另一个角度的网格 (d) 纹理映射后的效果

图 8 三维重建效果图

其中, 图 8(a) 是重建后生成的点云效果图, 图 8(b) 是重建后的斜侧面角度的网格效果, 图 8(c) 是另一个角度的网格效果, 图 8(d) 是纹理映射后的效果图。可以看出, 本文提出的立体匹配算法能够很好地恢复物体的结构, 从而证明了它的有效性。

结束语 本文针对未标定彩色图像三维重建的立体匹配不仅要求匹配精度较高, 而且要求提取到的特征点能够反映物体的外貌特征。首先引入限制因子消除 Harris 角点聚集的现象, 同时利用基于抛物线插值算法使角点达到亚像素级来提高匹配精度, 并运用精度较高的邻域彩色相关匹配法对其进行匹配。为了对基础矩阵估计精度提供保证, 针对我们的相机平移旋转较小的情况, 提出了斜率一致法以剔除误差比较大的误匹配点对; 并且采用 RANSAC 算法估计基本矩阵, 同时进一步剔除误匹配点对。最后, 为了让未匹配的可能反映物体外貌结构的点得以匹配, 我们使用基础矩阵引导匹配。实验证明, 本文的算法能够很好地恢复物体的结构。另外本文算法中虽然邻域彩色相关匹配法使匹配效率有所降低, 但是前期消除 Harris 角点聚集现象的同时能有效地提高匹配效率。针对单相机采集的未标定图像, 如何使算法适用于相机的旋转和平移较大的情况, 是今后进一步研究的内容。

参考文献

- [1] 马颂得, 张正友. 计算机视觉—计算理论与算法基础[M]. 北京: 科学出版社, 2000: 78-80
- [2] Schmid C, Mohr R, Bauckhage C. Evaluation of interest point detectors[J]. International Journal of Computer Vision, 2000, 37(2): 151-172
- [3] Harris C, Stephens M. A Combined Corner and Edge Detector [C] // Fourth Alvey Vision Conference, 1988: 147-151
- [4] Karna D K, Agarwal S, Nikam S. Normalized Cross-correlation Based Fingerprint Matching[C] // Proceedings of the 2008 Fifth International Conference on Computer Graphics, Imaging and Visualisation, 2008: 229-232
- [5] Mthlmanm K, et al. Calculating Dense Disparity Maps from Color Stereo Images; an Efficient Implementation [J]. International Journal of Computer Vision, 2002, 47(1): 79-88
- [6] 周围. 基于单目视觉的运动估计与结构重建[M]. 杭州: 浙江大学, 2007
- [7] 郭荣鑫, 杨邦成, 等. 亚像素位移插值计算方法的比较分析[J]. 昆明理工大学学报: 理工版, 2005, 30(2): 55-59
- [8] 雷鸣, 张广军. 基于互相关的图像匹配亚像素定位[J]. 光电工程, 2008, 35(5): 108-113
- [9] Marques M, Costeira J. Estimating 3D shape from degenerate sequences with missing data [J]. Computer Vision and Image Understanding, 2009, 113(3): 261-272
- [10] Dell'acqua P, Sarti P, Tubaro S. 3D Motion from structures of points, lines and planes [J]. Image and Vision Computing, 2008, 26(4): 529-549