

# 基于直方图着色法的 3D 重建\*

王 瀚, 柏正尧, 安镇宙, 武彩萍, 高 翔, 白宝丹  
(云南大学 信息与电子科学系, 云南 昆明 650091)

**摘要:** 着色是基于图像的三维场景重建的重要步骤. 它通过一致性检测来实现, 即检测一个体素映射到这个体素的所有视角中的相应区域的颜色是否一致. 一致性检测方法主要有标准差法、自适应阈值法、直方图法等. 对直方图一致性检测方法进行了分析并提出相应的算法. 它的一个显著特点是不需要调整参数, 算法相对容易实现. 实验结果表明, 该算法是有效可行的.

**关键词:** 着色法; 体素; 直方图; 一致性检测

中图分类号: TP 391 文献标识码: A 文章编号: 0258- 7971(2007)04- 0339- 05

从一组二维图像进行虚拟场景的真实感 3D 重建是计算机视觉的一个有挑战性的问题. 它成为计算机视觉领域的一个热点, 将使有真实感的三维模型应用于远程呈现(telepresence), 虚拟漫游(virtual walkthrough), 及其它面向图形的问题<sup>[1]</sup>. 到目前为止, 已有许多算法. 这些算法大致可分为 2 类, 一类是主动场景, 它分析反射到物体上的结构光; 另一类是被动场景, 它只要求在自然光下分析多幅二维图像. 被动场景不如主动场景精确, 它是一种要求不高的方法, 特别适用于普通相机<sup>[2]</sup>. 基于被动场景, 现在也有很多算法, 但它们没有解决好阻塞问题或者只有当拍摄图像的相机集中在一起才能工作. 为了克服以上困难, 着色法就产生了. 它是由 Seiz 和 Dyer<sup>[1]</sup>最先提出的. 从此, 许多方法在它们的基础上不断提出. 本文主要研究基于直方图的着色算法.

## 1 一致性(photoconsistency)检测

着色法(Voxel Colorng)是一个通过实践检验的成功算法. 重建场景由一系列有规律的立体网状的体素构成, 可以把体素类比为图像中的像素, 这样就类似像素, 每一个体素与一个颜色相联系. 在重建中, 通过着色法, 使分配给体素的颜色与输入

图像的颜色一致. 它有 3 步: 选择体素; 映射和建立体素与像素的相互关系; 如果颜色一致, 就着色, 如图 1 所示<sup>[3]</sup>.

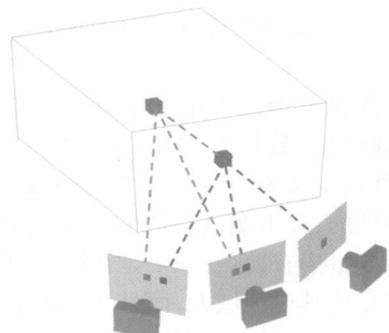


图 1 一致性检测

Fig. 1 Photoconsistency check

算法表示如下:

```

体素空间的每一层{
  每一层上的每一个体素 {
    映射每一个体素到图像
    收集这个体素映射到的一组像素
    通过这组像素比较体素的一致性
    如果(一致性小于阈值)
    { 给这个体素着色 } } }
    
```

\* 收稿日期: 2006- 10- 13

基金项目: 云南大学理(工)科校级科研项目资助(2004021QC); 云南大学中青年骨干教师培养计划资助.

作者简介: 王 瀚(1973- ), 男, 硕士生, 主要从事图像处理与 3D 建模方面的研究.

通讯作者: 柏正尧(1967- ), 男, 博士, 副教授, 主要从事图像处理与模式识别、DSP 应用方面的研究.

这个场景被初始化为由许多不透明的体素 (Voxel) 组成, 这些体素颜色不一致就会被去掉, 如果颜色一致就被保留。

一致性是这个算法的关键, 目前有许多改进的方法. 他们虽然有不同点, 但有一个共同的前提, 即是否着色或剪切体素是通过映射体素到图像来决定的. 给定  $N$  幅图像, 假设  $I_1, I_2, \dots, I_{p-1}$  是一系列图像并且体素  $V$  没有被它们阻塞. 这样, 对于每一个体素  $V$ , 一组非空的颜色值  $\pi$  从图像中被提取出来. 如下表示

$$\pi = \bigcup_{j=0}^{p-1} \pi_j = (c_0, c_1, \dots, c_m), \quad (1)$$

其中,  $\pi_j$  是一组像素颜色值从映射到体素  $V$  的图像  $j$  中提取的.  $(c_0, c_1, \dots, c_m)$  是真实的颜色值. 如果得到这组颜色值, 一致性检测可以用如下方法实现<sup>[4]</sup>: 标准差 (standard deviation, seitz and dyer, 1999), 自适应阈值 (Adaptive threshold, slabaugh et al, 2004), 闵可夫斯基距离 (Minkowsky distance, slabaugh et al, 2001), 直方图 (Histogram, slabaugh et al, 2004), 以及颜色缓存 (Colorcaching, chhabra, 2001)<sup>[5]</sup>.

一个体素是否包含在重建场景中是由实验测定它是否映射一致性的颜色. 在先前的一些工作中, 一致性被定义为阈值变量, 换一种说法, 一组颜色是一致性的, 如果颜色变量小于一个阈值.

阈值变量法有缺点: 一个是在图像中纹理区域的颜色变化比其他区域大, 而且一个体素在一幅图像中可见, 它通常从一组像素点可见, 这一组像素点的颜色值是变化的, 因此一个体素映射一组像素不只有一个颜色. 这时要确定一个固定阈值是困难的. 另一个缺点就是早期一致性检测中, 重建质量对阈值非常敏感, 重建须进行多次, 才能找到最佳值, 这是费时的.

与这些方法相比, 直方图有一些特点: ① 在重建纹理表面时具有非常好的鲁棒性; ② 适当地处理 RGB 颜色信息; ③ 不需调整参数值<sup>[6]</sup>. 所以, 我们用直方图来进行立体重建.

## 2 我们的方法

**2.1 摄像机标定** 计算机视觉 (computer vision) 的一个关键点就是从二维图像中获取三维场景的立体信息. 如位置, 形状等立体几何信息. 这是一个从二维到三维的转化过程, 需要确定摄像机成像的

几何位置模型. 通过实验与计算, 得到模型参数的过程就是标定. 摄像机模型参数满足下列矩阵方程

$$z c \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} 1/dx & 0 & u_0 \\ 0 & 1/dy & v_0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & t \\ o^T & 1 \end{bmatrix} \begin{bmatrix} xw \\ yw \\ zw \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha x & 0 & u_0 & 0 \\ 0 & \alpha y & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & t \\ o^T & 1 \end{bmatrix} \begin{bmatrix} xw \\ yw \\ zw \\ 1 \end{bmatrix} = M_1 M_2 X w = M X w, \quad (2)$$

其中,  $u, v$  是以像素为单位的坐标,  $x, y$  是以毫米为单位的坐标. 公式第 1 行右边第 1 个矩阵是 2 个坐标的关系; 第 2 个矩阵是投影透视关系, 第 3 个矩阵是表示旋转平移关系<sup>[7]</sup>.

**2.2 直方图** 为了克服阈值变量的难点, 我们采用直方图法, 提出的颜色一致性检测分为 2 步, 直方图构建和直方图交叠 (histogram intersection)<sup>[6]</sup>.

第 1 步, 提取每一个体素的任何一个可见像素, 对于一幅二维图像, 创建一个颜色直方图, 一组图像映射到一个给定的体素  $V_i$ ,  $t$  是叠代次数,  $i$  是从这一组像素映射的图像的顺序. 直方图法把 RGB 颜色空间分为  $512(8 * 8 * 8)$  个一致的立方体, 或者 RGB 也可被分为小的颜色空间. 每一个小的颜色空间为  $32 * 32 * 32$  个像素宽, 它又称为库 (BIN). 对每个体素而言, 直方图用来保留映射到每一幅二维图象上的颜色库的数量. 每一种颜色被编码为  $(0 \sim 7)$  之间. 例如: 红色用 0 表示  $(0 \sim 31)$ , 用 1 表示  $(32 \sim 63)$  ……

第 2 步, 颜色一致性检测. 我们通过比较所有成对  $V_i$  来进行一致性检测  $\forall_{kl} V_k^i \approx V_l^i, k \neq l$ . 判断一个体素是否一致性, 是通过这个体素的所有视角的全部直方图是否相交来判断, 也就是  $\forall_{kl} \text{HIST}(V_k^i) \cap \text{HIST}(V_l^i) \neq \emptyset, k \neq l$  如果只有一对非零的相应的库相同, 就可以认为它们相交<sup>[8]</sup>.

**2.3 交叠的直方图库** 如上所述, 此种颜色一致性的检测有一个缺点. 由于落在库边缘的颜色值不同于其它处的颜色值, 这就会产生误差. 在我们的

简单实现中, 一个像素有红色值 15 落在库的中间, 另一个像素在 15 与 30 之间被认为在同一个库. 假如这 2 个像素在不同的视角上, 它们的直方图相交并被认为是一致性的. 另一方面, 有 2 个像素, 1 个红色值为 31, 另 1 个为 32, 颜色值很近, 但它们被认为在不同的库, 这样 2 个红色值不容易通过一致性检测.

既然库的边界选择是任意的, 颜色一致性应该对它们的准确位置不敏感; 而且颜色值不能准确测量. 例如, 一个相机对同一场景连续拍摄, 会得到稍微不同的颜色; 此外, 光通常随时间变化一点点, 造成颜色漂移. 因此, 由于种种原因, 我们将忽略细微的颜色不同. 为解决这个问题, 通过使用交叠库. 我们在特定的条件下将扩大库, 让交叠的邻近库达到 20%. 一个像素的颜色落在多个交叠库应算在每一个库中. 落在一个库就算在那个库中. 这样就使一致性检测对库边界及小的不准确性不敏感.

**2.4 体素的映射尺寸** 当一个体素在一幅图像中仅仅从一小组像素中可见时, 直方图的相交并不是一个可靠的指标. 因此, 当像素的数目下降到一个规定值(15), 这幅图像并不能完全看见体素. 起初, 这可能是一个不幸的底线. 对于体素的尺寸不足以映射到足够的像素. 实际中, 这不是一个问题. 运行时间正比于体素数目, 这将在一致性检测中限制体素的尺寸. 既然更多的信息在高分辨率图像中对重建有益. 可以采用高分辨率相机<sup>[6]</sup>.

### 3 一致性(photoconsistency)检测算法

#### 3.1 构建直方图 直方图构建算法如下:

对每一个体素, 可定义‘ $i$ ’个直方图, 把从‘ $i$ ’幅图像中的像素映射到体素的颜色收集起来,

放到直方图中相应颜色库中.

#### 3.2 一致性检测 一致性检测算法如下:

对每一个体素, 比较它的所有直方图, 如果其中有一对非零的颜色库交叠, 就认为它是一致性的; 否则, 剪切这个体素.

## 4 实验过程

**4.1 拍摄图像** 拍摄 2 组图像, 1 组拍标定方格板, 标定内部参数, 如图 2 所示.

另 1 组拍摄包括重建物体的标定方格板, 标定外部参数, 如图 3 所示.

**4.2 摄像机标定** 在重建中用从 bouget web site<sup>[9]</sup>下载的标定工具“Bougets camera toolbox”. 用图 2 第 1 组图像进行内部参数的标定, 是公式(2)中的前 2 个矩阵; 用图 3 第 2 组图像进行外部参数的标定, 是公式(2)的第 3 个矩阵.

**4.3 三维重建** 3D 重建算法采用 C++ 编程实现, 实验结果如图 4 和图 5 所示. 并把这个算法与标准差(standard deviation, seitz and dyer, 1999)的结果(图 6)进行比较.



图 2 内部参数标定板

Fig. 2 Intrinsic calibration board

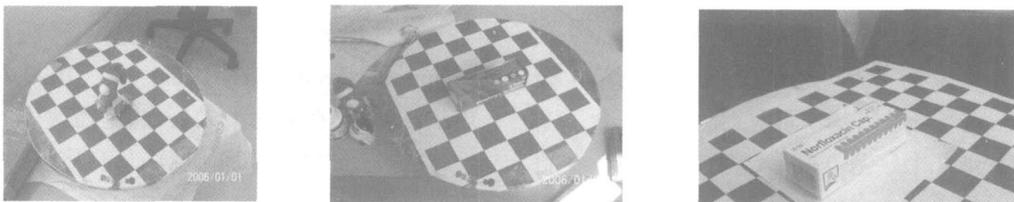


图 3 外部参数标定板

Fig. 3 Extrinsic calibration board

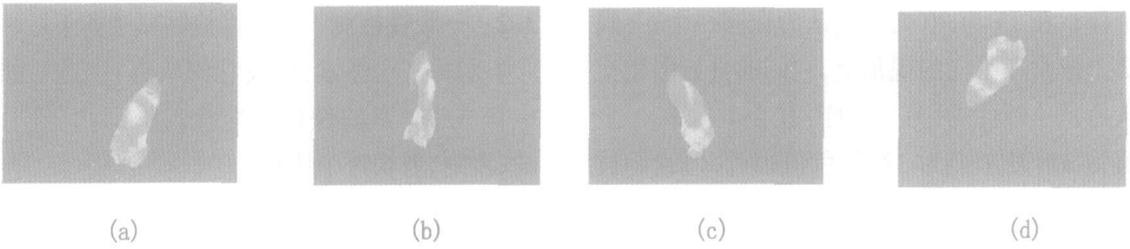


图 4 一个布娃娃的 3D 重建图

Fig. 4 3D reconstruction image of a doll

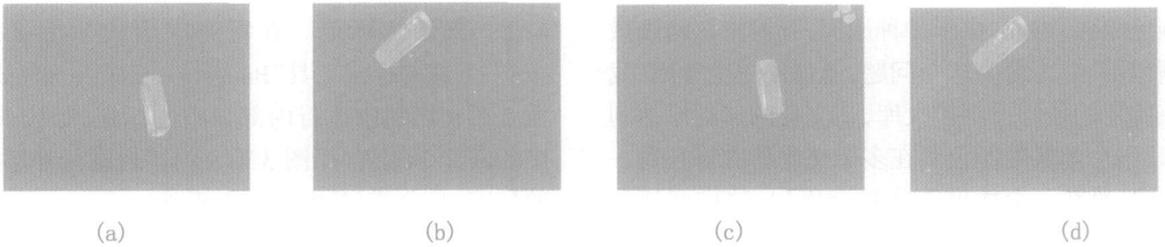


图 5 乒乓盒子的 3D 重建图

Fig. 5 3D reconstruction image of a table tennis box

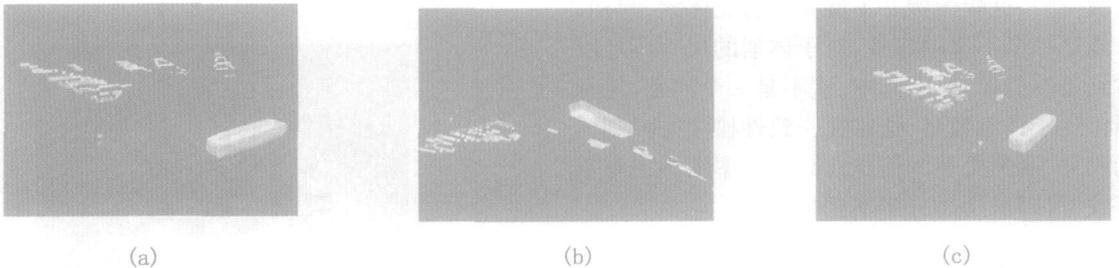


图 6 一个盒子的 3D 重建图

Fig. 6 3D reconstruction image of a box

## 5 结 论

第 1 组图像中布娃娃的各个侧面基本成形, 这是一个纹理形状都特别复杂的物体. 特别是图 4 (b) 中, 把帽子的复杂形状基本表现出来.

第 2 组图像是完成乒乓球盒子的重建. 第 1 幅盒子图 5(a) 中, 条形码基本可见. 最后一个盒子图 5(d) 中乒乓球几个字基本成形, 盒子的一个空的圆圈清晰可见 (在本文的电子版中可以放大重建图像). 本文中, 用于重建的图片分辨率为  $1656 \times 1242$ , 如果用更高分辨率的相机, 效果会更好, 但重建时间会更长. 算法在时间上太长, 这是需改进的地方. 另外, 在直方图交叠, 如果继续研究, 应该

得到更好的效果的算法.

第 3 组图像图 6 是用标准差 (standard deviation, seitz and dyer, 1999) 做出来的. 它的噪声很多, 形状不逼真, 物体不够清楚.

通过比较以上 3 组图, 这个算法比标准差 (standard deviation, seitz and dyer, 1999) 有了一些进步. 但算法在时间上太长, 这是需改进的地方. 另外, 在直方图交叠, 如果继续研究, 应该得到更好的效果的算法.

## 参考文献:

- [1] SEITZ S, DYER C. Photorealistic scenereconstruction By voxel coloring [C]//Proceeding of 1997 Computer

- vision and Pattern Recognition conference, Puerto, Puerto Rico, USA, 1997: 1 067 1 073.
- [2] Yi Young Youl, Kim hyo sung, Ye Soo Young, et al. Searching the optimal threshold for voxel coloring in 3Dreconstruction[EB/OL]. [2006- 10- 18]. <http://pixel.otago.ac.nz/ipapers/34.pdf>.
- [3] DYER C. Volumetric scene reconstruction from multiple views[EB/OL]. [2006- 10- 18]. <http://www.cs.wisc.edu/~dyer>.
- [4] OZUN O, Ulas Yilmaz, ATALAY V. Comparison of photoconsistency measures used in voxel coloring[EB/OL]. [2006- 10- 18]. <http://www.ipf.tuwien.ac.at/isprs/wgiii/BenCos2005/papers>.
- [5] CHHABRA V. Reconstructing specular objects with image based rendering using color caching. master's thesis, Worcester polytechnic institute, 2001[EB/OL]. [2006- 10- 18]. <http://www.wpi.edu/Pubs/ETD/Available/etd0427101-111305/unrestricted/chhabra.pdf>.
- [6] Stevens Mark R, CULBERTSON B, MALZBENDER T. A histogram \_based color consistency test for voxel coloring[EB/OL]. [2006- 10- 18]. <http://www.hpl.hp.com/research/mms/publications/vision/>.
- [7] 马颂德, 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, 1998.
- [8] SLABAUGH G G, CULBERTSON W B, MALZBENDER T, et al. Methods for volumetric reconstruction of visual scenes[J]. International Journal of Computer Vision, 2004, 57(3): 179-199.
- [9] Jean Yves Bouguet.[EB/OL]. [2006- 10- 18]. [Http://www.vision.caltech.edu/bougetj/calib\\_doc/](http://www.vision.caltech.edu/bougetj/calib_doc/).

### 3D reconstruction based on coloring using histogram

WANG Han, BAI Zhengyao, AN Zherzhou,

WU Caiping, GAO Xiang, BAI Baodan

(Department of Information and Electronic Science, Yunnan University, Kunming 650091, China)

**Abstract:** Coloring is an important step of 3D reconstruction based on images. It needs photo consistency checking, i. e. if a voxel is projecting to all regions of consistent color in all views of the voxel. The major methods of photo consistency checking use standard deviation, adaptive threshold, histogram, etc. The histogram based method of photo consistency checking is analyzed and a consistency checking algorithm is proposed. One of the important characteristics of the method is that its parameters does not need to be regulated, thus the algorithm can be easily implemented. The experimental results show that the algorithm is practical and effective.

**Key words:** coloring; voxel; photo consistency; histogram