

结合深度图像和最佳分割阈值迭代的自遮挡检测算法^①

张世辉^② 张煜婕 孔令富

(燕山大学信息科学与工程学院 秦皇岛 066004)

摘 要 针对视觉任务中普遍存在的自遮挡现象和目前视觉系统尚无明确的自遮挡检测算法的现状,提出了一种结合视觉目标深度图像和最佳分割阈值迭代的自遮挡检测算法。在分析图像阈值分割技术的基础上,将分割阈值迭代法的思想引入深度图像领域,并结合使用视觉目标对应的深度差值图像信息,通过求取合适的阈值实现了对自遮挡现象的检测。实验结果表明,该方法能够有效地检测出视觉目标中存在的自遮挡现象并定位自遮挡边界,弥补了目前自遮挡检测领域研究的不足。

关键词 深度图像,深度差值图像,阈值选择,迭代,自遮挡检测

0 引 言

在目标识别、立体匹配、视觉跟踪、场景重建、自动检测等诸多视觉研究邻域,遮挡现象已成为影响识别、匹配、跟踪等各种任务执行效果的主要原因之一。如何准确合理地检测、界定遮挡现象的发生已成为解决各类视觉目标遮挡问题的关键和必要环节。视觉遮挡既是一个几何现象又是一个光学现象。依据遮挡的不同情形,遮挡问题可分为两大类,一类是指不同物体之间存在的相互遮挡,另一类是指同一物体的多个部分之间存在的自遮挡(self-occlusion)。目前,在遮挡检测研究领域,国内外大多都是针对相互遮挡的检测和运动物体跟踪方面的文献^[1-5]。有关静态视觉目标自遮挡现象检测研究方面的文献并不多,仅几篇外文文献中涉及到。代表性的文献如文献[6]论述了利用散焦来处理深度信息恢复过程中的自遮挡,文献[7]采用阈值法将自遮挡区域从经过中值滤波加强的深度图像中分离,但相关文献并未给出具体的检测方法或实验结果。由此可见,国内外的自遮挡现象的研究,特别是以实现静态视觉目标主动、灵活、全面、细致的观测,以便使视觉系统更好地完成识别、重构、建模、定位、对接、装配等功能或作业为目的的有关视觉目标自遮挡问题的相关技术和方法的研究,有待进一步深入进行。鉴于此,本文从 863 课题研究的实际需要出

发,提出了一种结合视觉目标深度图像信息和最佳分割阈值迭代思想的自遮挡检测算法。该算法可以实现对复杂实体存在的自遮挡现象进行检测,且检测结果不受光照强度、纹理变化的影响。实验结果验证了所提算法的可行性和有效性,具有较强的实用性。

1 算法原理

分析视觉目标所对应的深度图像信息可知,图像上点的邻域关系直接反映了目标实体上点的空间邻域关系。具体来说,对于被观测的三维实体上不存在自遮挡现象的地方,在深度图像上所对应的图像点与其邻域点之间的深度值变化是连续的,即图像中相邻的点在实体上也相邻。对于被观测的三维实体上存在自遮挡现象的地方,在深度图像上所对应的图像点与其邻域点之间的深度值变化是不连续的,即深度值分布有明显的阶跃,也就是说,这些点在图像上虽然是相邻的,但在三维实体上并不相邻。故一般情况下,实体上存在自遮挡现象的区域与图像中存在深度不连续现象的区域相对应。基于上述分析,本文通过计算分析了视觉目标深度图像上的点与其一定邻域内各点所对应深度值的差值变化特征,并考虑到此差值如果在某一限定阈值内变化则表明图像在该点处一定邻域范围内的深度是连续的,否则表明在该点处的一定邻域范围内出现了深

^① 863 计划(2006AA04Z212)和河北省自然科学基金(F2007000423, F2010001276)资助项目。

^② 男,1973 年生,博士,副教授;研究方向:智能并联机器人,视觉信息处理;联系人, E-mail: sshhzz@ysu.edu.cn
(收稿日期:2009-10-26)

度不连续现象,从而可以判定三维实体在该处发生了自遮挡现象,具体如下。

1.1 深度差值图像计算

一幅图像是一个平面物体,其亮度或彩色可逐点不同,这种变化在数学上可表示为两个空间变量的函数,称为图像函数。若只考虑黑白图像,则图像函数可表示为 $f(x, y)$ 。如果用 $f(x, y)$ 表示图像平面上的点所对应景物点的景深(也称为深度值),即对应景物点到图像平面的距离,则称为深度图像,记为 $d(x, y)$ 。换句话说,对于深度图像而言,其像素值表示对应景物点到图像平面的距离信息,也即深度信息。

为了判断深度图像 $d(x, y)$ 上存在的深度不连续现象,在进行自遮挡检测之前需要对图像进行预处理。首先对深度图像上的每一点与其对应 4-邻域内各点的深度值进行差值计算,然后将其平均值 $f'(x, y)$ 作为该点处新的像素值,从而得到深度差值图像 $d'(x, y)$ 。 $f'(x, y)$ 的计算公式为

$$f'(x, y) = \frac{|f(x, y) - f(x \pm 1, y)| + |f(x, y) - f(x, y \pm 1)|}{4} \quad (1)$$

1.2 最佳分割阈值选取

一般来讲,阈值分割方法是针对灰度图像而言的,根据图像中所要提取的目标物与其背景在灰度特性上的差异,把图像视为具有不同灰度级的区域的组合,通过选取阈值,将目标区域从它们的背景中分离出来。当用阈值来分割目标与背景时,如果某点的灰度值小于阈值则将其构成目标中的一点,如果大于阈值的话就构成背景中的一点。最佳分割阈值迭代^[8-10]选取阈值的思想是:首先选择一个估计值作为阈值的初始值,然后依据此阈值对图像进行分割,产生子图像并根据子图像的特征来选取新的阈值,再用新的阈值分割图像,经过一定次数的循环,使错误分割的图像像素点降到最少。

本文将最佳分割阈值迭代思想引入深度图像处理领域,目的是针对深度差值图像 $d'(x, y)$ 进行最佳阈值选取,从而根据迭代得出的阈值 T_d 来检测深度图像 $d(x, y)$ 中是否存在自遮挡现象。将最佳分割阈值迭代思想应用于处理深度差值图像 $d'(x, y)$ 进行阈值迭代选取的最佳分割阈值迭代算法(optimal segmentation threshold iteration algorithm, OSTIA)的具体描述如下:

输入:深度差值图像 $d'(x, y)$

输出:适合深度差值图像 $d'(x, y)$ 的阈值 T_d

步骤 1:令初始阈值 $T_d = \frac{(f'_{\max} + f'_{\min})}{2}$, 其中,

$f'_{\max}$ 和 $f'_{\min}$ 分别表示深度差值图像 $d'(x, y)$ 中的最大和最小像素值。

步骤 2:根据阈值 T_d 将深度差值图像 $d'(x, y)$ 上的点分割为两组不同的像素点集 A 和 B , 分别求出这两组像素点集的平均像素值 f'_A 和 f'_B :

$$f'_A = \frac{\sum_{f(i,j) < T_d} f'(i, j) \times N(i, j)}{\sum_{f(i,j) < T_d} N(i, j)} \quad (2)$$

$$f'_B = \frac{\sum_{f(i,j) > T_d} f'(i, j) \times N(i, j)}{\sum_{f(i,j) > T_d} N(i, j)} \quad (3)$$

式中, $f'(i, j)$ 为深度差值图像 $d'(x, y)$ 上点 (i, j) 处的像素值, $N(i, j)$ 是点 (i, j) 处的权重系数,一般取 1.0。

步骤 3:求出新阈值 $T_{d+1} = \frac{f'_A + f'_B}{2}$ 。

步骤 4:若 T_{d+1} 与 T_d 相等,则 T_d 即为所求阈值;否则令 $T_d = T_{d+1}$ 并转步骤 2 进行迭代计算,最终可以计算得出深度差值图像 $d'(x, y)$ 所对应的阈值 T_d 。

1.3 自遮挡检测算法

结合上述阈值迭代选取算法,自遮挡检测算法(self-occlusion detection algorithm, SDA)的具体描述如下:

输入:视觉目标深度图像 $d(x, y)$

输出:视觉目标自遮挡检测结果

步骤 1:遍历视觉目标所对应深度图像 $d(x, y)$ 上各点,分别计算得出点 (x, y) 与其对应 4-邻域内各点的深度差值绝对值并结合(1)式获得深度差值图像 $d'(x, y)$ 。

步骤 2:采用最佳分割阈值迭代算法 OSTIA 对得到的深度差值图像 $d'(x, y)$ 进行迭代计算,得出阈值 T_d 。

步骤 3:选取图像 $d'(x, y)$ 上任一点 (x, y) 作为起始点。

步骤 4:对该点进行判断:如果 $f'(x, y) \geq T_d$, 则令 $(x, y) \in S$, 并将该点处的像素值记为 1;否则将该点处的像素值记为 -1。其中 S 为与自遮挡点具有相同像素坐标的点所组成的集合。

步骤 5:如果图像 $d'(x, y)$ 上存在像素值非 -1 或 1 的点,则选取其中任一点重复步骤 3 和步骤 4;

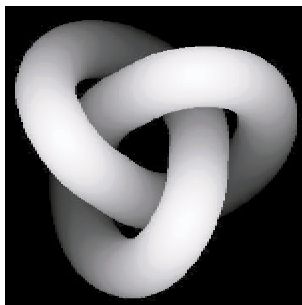
否则检测结束,得出点集 S 。如果集合 S 不为空,则表明深度图像 $d(x, y)$ 中存在自遮挡现象;如果集合 S 为空,则表明深度图像 $d(x, y)$ 中不存在自遮挡现象。

此处需要说明的是,由于深度图像 $d(x, y)$ 上点的像素坐标与深度差值图像 $d'(x, y)$ 上的点一一对应,故扫描集合 S 中各点的像素坐标便可以在深度图像 $d(x, y)$ 上定位得出自遮挡点,进而可获得自遮挡边界。

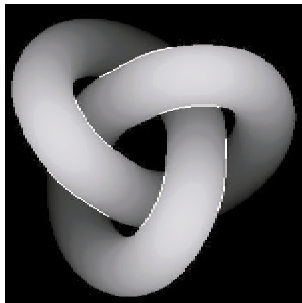
2 实验及分析

实验用的图像选自 <http://range.informatik.uni-stuttgart.de/docs/html> 上 Stuttgart Range Image Database 中大小为 400×400 的深度图像,利用 Matlab 进行编程实现自遮挡检测算法的验证工作。由于目前未查到包含可与之进行比较的有关静态视觉目标自遮挡检测方法及其实验结果的文献,所以,此处仅给出本文的实验方案和结果。

实验用 Knot 的原始深度图像如图 1(a) 所示。采用上述提出的自遮挡检测算法,经分析计算后选取深度阈值 $T_d = 3.7933$ 得到的实验结果如图 1(b) 所示。



(a) Knot 的深度图像



(b) Knot 的自遮挡检测结果

图 1 Knot 的深度图像和自遮挡检测结果

图 1(b) 中白(点)线部分是检测得到的所有自遮挡点以及生成的自遮挡边界,即 Knot 中的 6 处自

遮挡现象均被进行了正确的检测和标记,故本文方法可以正确有效地检测出深度图像所对应的视觉目标中存在的自遮挡现象,而且通过分析得出的自遮挡点可以进一步定位自遮挡边界。

为了进一步说明本文方法的可行性和有效性,图 2 和图 3 分别给出了 Bunny、Pitt-plane 的深度图像和自遮挡检测结果。实验过程中, Bunny 和 Pitt-plane 的阈值经计算分别取 $T_d = 4.2316$ 和 $T_d = 1.2015$ 。

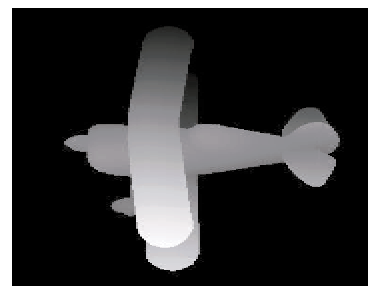


(a) Bunny 的深度图像

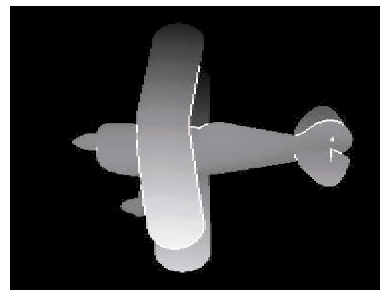


(b) Bunny 的自遮挡检测结果

图 2 Bunny 的深度图像和自遮挡检测结果



(a) Pitt-plane 的深度图像



(b) Pitt-plane 的自遮挡检测结果

图 3 Pitt-plane 的深度图像和自遮挡检测结果

由图 2 可知,本文方法同样能够正确地检测和定位出 Bunny 头部、耳朵、躯干和尾部的所有自遮挡现象(以白点线表示)。进一步对照图 2(a)和图 2(b)可知:图 2(b)中 Bunny 躯干中下方处虽然存在深度值不连续趋势,但这些区域的点并未被检测为自遮挡点,这正是深度阈值所起的作用,而图 2(a)中的对应区域也的确未发生自遮挡现象。

同样,对比图 3(a)和图 3(b)可知,在 Pitt-plane 的头、机翼、尾部等多处均存在自遮挡现象,图 3(b)中白(点)线部分是检测得到的所有自遮挡点以及生成的自遮挡边界。该实验结果同样表明所提方法对深度图像中是否存在自遮挡能够进行有效检测及定位。

3 结论

(1)自遮挡现象是许多视觉任务不可避免的问题,本文将阈值迭代思想应用于深度图像,提出一种结合深度图像和最佳分割阈值迭代的自遮挡检测算法,并以 Knot、Bunny 和 Pitt-plane 为例进行了实验,实验表明该算法对自遮挡现象能够得出较满意的检测结果。

(2)由于该算法仅利用视觉目标深度图像信息和仅使用了一次阈值法,故该算法具有操作简单、方便可行的特点,在目前国内外尚无更好自遮挡检测算法的情况下,不失为一种可选方法。

(3)所提算法弥补了自遮挡领域研究的不足,在视觉系统主动界定、规避自遮挡、和完成相关视觉任务方面具有理论意义和实用价值。

参考文献

- [1] Ito K, Sakane S. Robust view-based visual tracking with detection of occlusions. In: Proceedings of the IEEE International Conference on Robotics and Automation, Seoul, Korea, 2001. 1207-1213
- [2] Wu Y, Yu T, Hua G. Tracking appearances with occlusions. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Madison, Wisconsin, USA, 2003. 789-795
- [3] Tao H, Sawhney H, Kumar R. Object tracking with Bayesian estimation of dynamic layer representations. *IEEE Transactions on Pattern Analysis And Machine Intelligence*, 2002, 24(1): 75-89
- [4] Gentile C, Camps O. Segmentation for robust tracking in the presence of severe occlusion. *IEEE Transactions on Image Processing*, 2004, 13(2): 166-178
- [5] 周妍,胡波,张建秋.基于粒子滤波器和风险决策跟踪遮挡目标的方法.电子学报,2007,35(2):350-353
- [6] Bhasin S, Chaudhuri S. Depth from defocus in presence of partial self-occlusion. In: Proceedings of the 8th IEEE International Conference on Computer Vision, Vancouver, Canada, 2001. 488-493
- [7] Park J C, Kim S M, Lee K H. 3D mesh construction from depth images with occlusion. In: Lecture Notes in Computer Science. Berlin: Springer Berlin/ Heidelberg, 2006. 770-778
- [8] 徐平,邵定宏,魏楹.最佳阈值分割和轮廓提取技术及其应用.计算机工程与设计,2009,30(2):437-439
- [9] Rhee F C H, Shin Y S. A fast numerical method for finding the optimal threshold for image segmentation. In: Proceedings of the IEEE International Conference on Fuzzy Systems, St. Louis, MO, US, 2003. 984-989
- [10] Batenburg K J, Sijbers J. Optimal threshold selection for tomogram segmentation by projection distance minimization. *IEEE Transactions on Medical Imaging*, 2009, 28(5): 676-686

Self-occlusion detection algorithm combining depth image and optimal segmentation threshold iteration

Zhang Shihui, Zhang Yujie, Kong Lingfu

(College of Information Science and Engineering, Yanshan University, Qinhuangdao 066004)

Abstract

Aiming at the familiar self-occlusion phenomenon in vision tasks and considering the actuality that there are no specific self-occlusion detecting algorithms, for vision systems, the paper proposes a self-occlusion detection algorithm combining depth image and optimal segmentation threshold iteration. At first, the segmentation threshold iteration method is applied to depth image by analyzing the threshold segmentation technology. Then, the self-occlusion detection is realized by obtaining a proper threshold and using the depth difference image information of the vision object. The experimental results show that the proposed algorithm can detect the self-occlusion phenomenon and obtain the occlusion boundary effectively, so the deficiency in self-occlusion detection domain is compensated.

Key words: depth image, depth difference image, threshold selection, iteration, self-occlusion detection