

【泡泡图灵智库】基于CPU的实时6D物体姿态估计 (arXiv)

原创：paopaoslam 泡泡机器人SLAM 今天

泡泡图灵智库，带你精读机器人顶级会议文章

标题：Real-Time 6D Object Pose Estimation on CPU

作者：Yoshinori Konishi, Kosuke Hattori, Manabu Hashimoto

来源：Arxiv

编译：侯延华

审核：谢泽如

欢迎个人转发朋友圈；其他机构或自媒体如需转载，后台留言申请授权

摘要

本文提出了一种基于模板的从RGB-D图像快速估计6D物体姿态的方法。

模板由密集采样产生。

运用了3种主要技术：

PCOF-MOD，(多模态透视累积定向特征)：在混乱背景下，不增加虚警的前提下只放松3D目标姿态微小变化的匹配条件，提高了在混乱背景条件下的鲁棒性。

BPT：基于树的数据结构，分辨率与金字塔相似，降低了2D位置和3D姿态的搜索空间，用于高效匹配大量的模板。

用于从粗到精搜索的**优化内存重组**：低分辨率图像金字塔重组后相邻像素的不同特征对齐，在此基础上运行SIMD模板匹配。

bin picking数据集上的桌面电脑实验结果表明：在准确率和速度方面优于当下包括基于CNN在内的方法。使用3D CAD数据可以在几分钟之内完成模板训练；姿态估计在CPU环境下只需要23fps，几乎是实时的。

研究背景

当前主要有3种估计6D目标姿态的方法：

1 基于模板的：

多种特征可供使用边缘、轮廓等，深度信息可以提高鲁棒性；预先计算的响应映射有助于快速计算匹配的相似度；从粗到精的搜索策略、hash 表、GPU等的运用都可以加快匹配速度。

2 基于局部描述子：

点对特征(PPF)是目前已知最好的描述子，存在多种变种方法。比基于模板的方法慢。

3. 基于学习的：

在应对混乱背景和局部遮挡问题方面比较好，匹配速度快，但是训练需要大量有标记样本和大量的GPU时间。

算法流程

1 构造模板：

PCOF特征对3D姿态引起的外观变化不敏感，从RGB图像抽取的方向梯度--轮廓形状；法线特征--物体的表面形状。

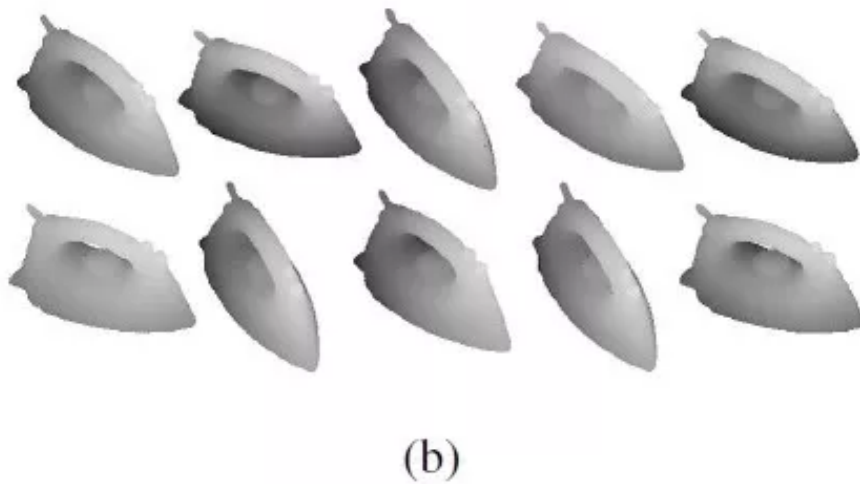
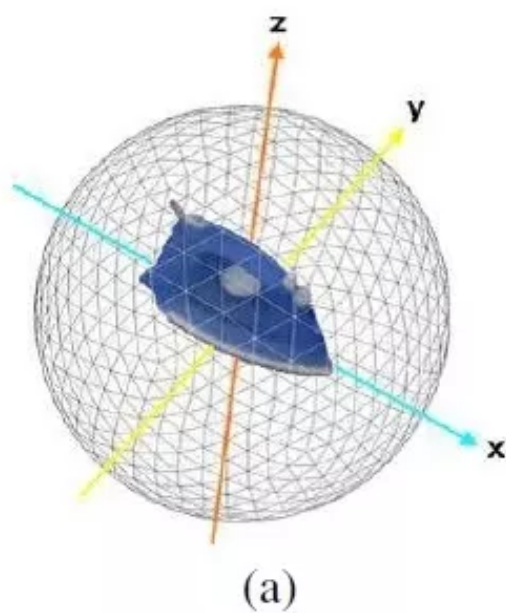


图2 (a)3D CAD电熨斗, 坐标轴及采样视角
(b)围绕某一坐标轴随机视角的深度图像样本

1.1 4个参数(x , y , optical轴, 到目标距离)均匀分布随机采样的视角渲染得到深度图像。

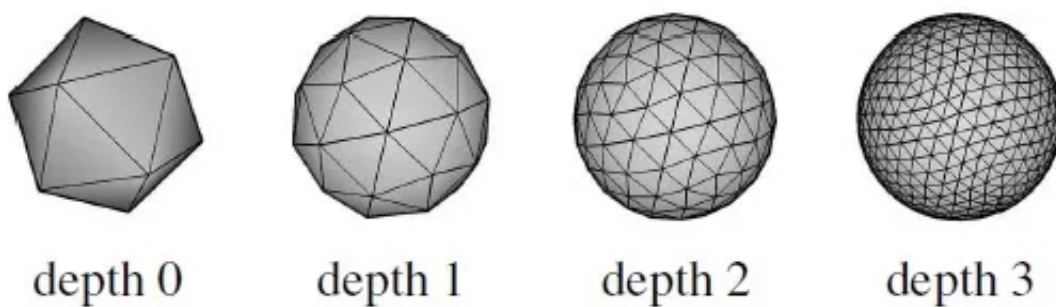


图4 从二十面体开始递归分解为
1280面 (642 节点)正多面体

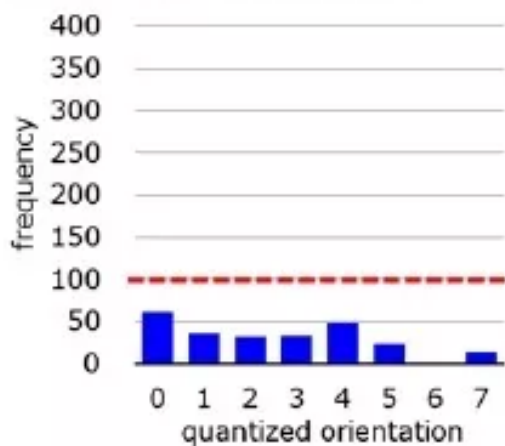
这些视角相当于正多面体的节点。从最左的正二十面体开始递归二分每个边，最终得到1280正多面体。

1.2 对上面的深度图像Sobel算子对轮廓计算梯度向量，通过临近点拟合平面计算法线向量。

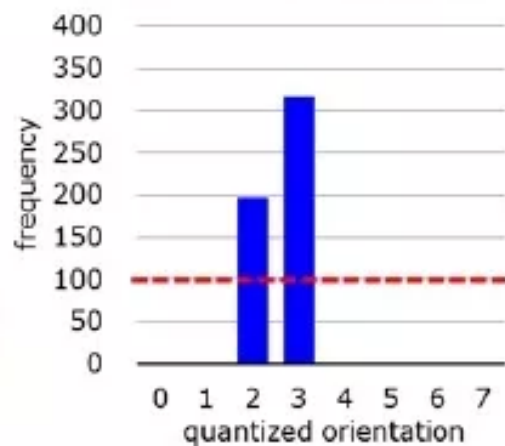
1.3 计算每个像素的梯度、法线的方向8-bins直方图和反应深度的权值--计算每个像素的梯度和法线方向的分布(频率)。

1.4 最后获取主方向。

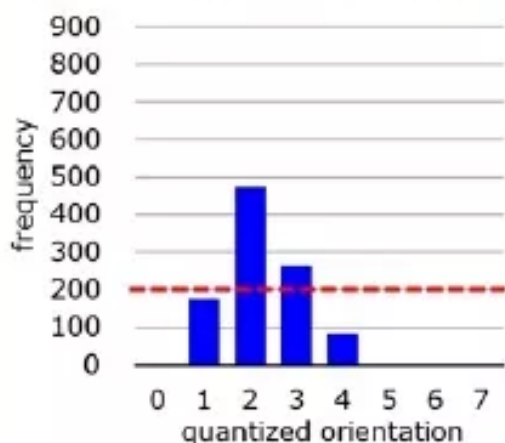
(1) ori: 00000000, w: 0



(2) ori: 00110000, w: 316.8



(3) ori: 00110000, w: 472.6



(4) ori: 00100000, w: 763.4

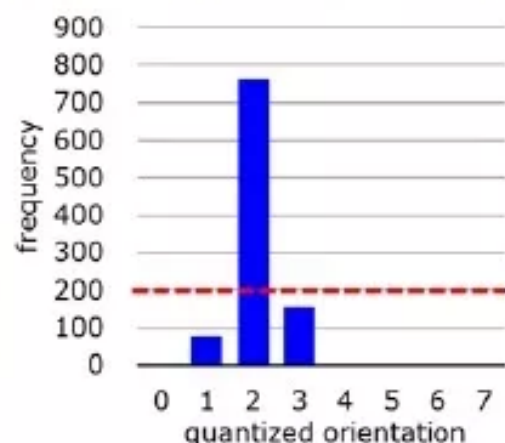


图3 直方图、量化方向特征(ori)及权值(w)

得到两种模板

$$T : \{x_i, y_i, ori_i, w_i | i = 1, \dots, n\}.$$

输入图像在像素(x,y)与模板的相似度计算:

$$score(x, y) = \frac{\sum_{i=1}^n \delta_k(ori_{(x+x_i, y+y_i)}^I \in ori_i^T)}{\sum_{i=1}^n w_i}$$

$$\delta_i(ori^I \in ori^T) = \begin{cases} w_i & \text{if } ori^I \wedge ori^T > 0, \\ 0 & \text{otherwise.} \end{cases}$$

2 .BPT: Balanced Pose Tree伪代码

BPT 是一个 B-trees 组(depth 0,1,2,3)每个parent 节点有12-16个子节点。

Algorithm 1 Building balanced pose tree

Input: Orientation histograms H_3 at level 3

Output: Templates T_i ($i = 0, \dots, d - 1$)

for $i \leftarrow d - 1$ to 0 **do**

$P_i \leftarrow$ parent viewpoints of i th level in BPT

for each parent viewpoint $P_{i,j}$ **do**

$C_{i+1,j} \leftarrow$ child viewpoints of $P_{i,j}$

$H'_{i+1,j} \leftarrow$ add histograms of child viewpoints

$H_{i+1} \in C_{i+1,j}$ at each pixel

$H''_{i+1,j} \leftarrow$ normalize histograms $H'_{i+1,j}$

$H_{i,j} \leftarrow$ add histograms of nearby 2×2 px of $H''_{i+1,j}$

$H'_{i,j} \leftarrow$ normalize histograms $H_{i,j}$

$T_{i,j} \leftarrow$ thresholding $H'_{i,j}$ and extracting new binary features and weights

end for

end for

3.内存重组以便于从粗到精的搜索

0	1	2	3	4	5	6
10	11	12	13	14	15	16
20	21	22	23	24	25	26
30	31	32	33	34	35	36
40	41	42	43	44	45	46
50	51	52	53	54	55	56
60	61	62	63	64	65	66

Gradient orientation (RGB)

0	1	2	3	4	5	6
10	11	12	13	14	15	16
20	21	22	23	24	25	26
30	31	32	33	34	35	36
40	41	42	43	44	45	46
50	51	52	53	54	55	56
60	61	62	63	64	65	66

Normal orientation (Depth)



Memory re-arrangement

0	0	10	10	20	20	30	30	1	1	11	11	21	21	31	31	2	2	12	12	22
10	10	20	20	30	30	40	40	11	11	21	21	31	31	41	41	12	12	22	22	32
20	20	30	30	40	40	50	50	21	21	31	31	41	41	51	51	22	22	32	32	42
30	30	40	40	50	50	60	60	31	31	41	41	51	51	61	61	32	32	42	42	52

Feature map for SIMD instructions

图5 内存重排示意图

匹配的过程是对模板穷举的，SIMD指令可以加速相似度计算。重排后各种level都可以利用SIMD加速。

4.姿态估计与调整

4.1从深度图像计算梯度向量、法向量

4.2遍历**BPT**特征根节点，计算相似度(特征金字塔).相似度高于阈值的作为(pose and position)备选。备选区的低层金字塔特征进一步匹配BPT低层模板，最后BPT底层模板的匹配底层金字塔特征获得姿态和位置估计。

4.3 ICP算法做进一步调整。



主要结果

	PPF	Ours
Bolt	0.754	0.944
Connector	0.758	0.982
Holder	0.922	0.983
Nut	0.858	0.980
Pipe	0.879	0.986
SheetMetal	0.700	0.867
Mean	0.812	0.957

Table 4. F1 scores on bin-picking dataset.

	PPF	w/o MemRea	w/ MemRea
Bolt	3334.1	230.6	107.2
Connector	2382.4	209.7	89.2
Holder	1343.1	99.6	42.9
Nut	4387.3	214.3	97.0
Pipe	1501.5	100.1	41.6
SheetMetal	3121.3	183.9	77.9
Mean	2678.3	173.0	76.0

Table 5. Processing time (ms) with and without the memory rearrangement (MemRea) on bin-picking dataset.

	position (mm)			rotation (deg)		
	X	Y	Z	X	Y	Z
Bolt	0.392	0.312	0.421	1.968	0.687	0.722
Connector	0.443	0.391	0.448	1.372	1.302	0.991
Holder	0.685	0.647	0.406	0.532	0.742	1.029
Nut	0.384	0.274	0.378	0.841	0.765	1.486
Pipe	0.354	0.254	0.344	0.557	0.432	0.564
SheetMetal	0.662	0.615	0.397	0.671	0.571	0.945
Mean	0.487	0.415	0.399	0.990	0.750	0.956

Table 6. Mean absolute errors on bin-picking dataset.

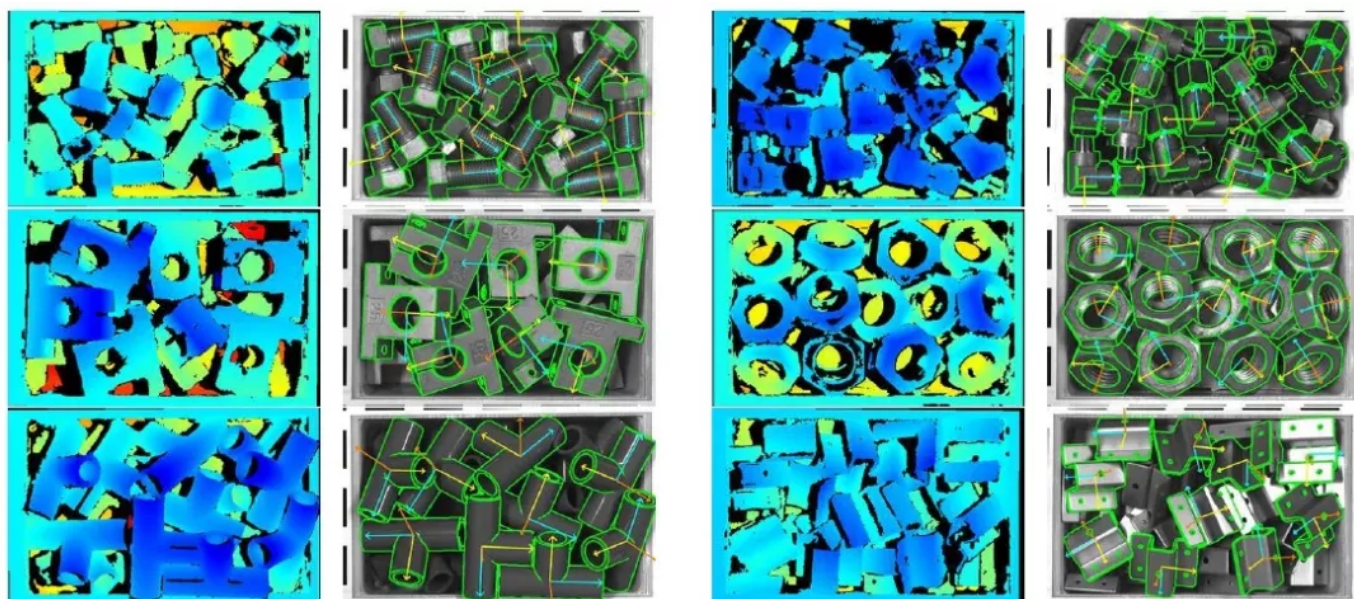


图7. 示例bin-picking dataset.

上: 螺栓及连接件.

中: 支撑和螺母.

Abstract

We propose a fast and accurate 6D object pose estimation from a RGB-D image. Our proposed method is template matching based and consists of three main technical components, PCOF-MOD (multimodal PCOF), balanced pose tree (BPT) and optimum memory rearrangement for a coarse-to-fine search. Our model templates on densely sampled viewpoints and PCOF-MOD which explicitly handles a certain range of 3D object pose improve the robustness against background clutters. BPT which is an efficient tree-based data structures for a large number of templates and template matching on rearranged feature maps where nearby features are linearly aligned accelerate the pose estimation. The experimental evaluation on tabletop and binpicking dataset showed that our method achieved higher accuracy and faster speed in comparison with state-of-the-art techniques including recent CNN based approaches. Moreover, our model templates can be trained only from 3D CAD in a few minutes and the pose estimation run in near real-time (23 fps) on CPU. These features are suitable for any real applications.

如果你对本文感兴趣，想要下载完整文章进行阅读，可以关注【[泡泡机器人SLAM](#)】公众号。

点击[阅读原文](#)，即可获取本文下载链接。

欢迎来到泡泡论坛，这里有大牛为你解答关于SLAM的任何疑惑。

有想问的问题，或者想刷帖回答问题，泡泡论坛欢迎你！

泡泡网站：www.paopaorobot.org

泡泡论坛：<http://paopaorobot.org/bbs/>

泡泡机器人 SLAM

分享 • 深度 • 卓越 • 前沿

SLAM前沿技术领跑者，与你一同成长

Knowledge Worth Spreading



公
开
课



原
创



新
闻



好
书
一
起
读



访
谈



翻
译

泡泡机器人SLAM的原创内容均由泡泡机器人的成员花费大量心血制作而成，希望大家珍惜我们的劳动成果，转载请务必注明出自【泡泡机器人SLAM】微信公众号，否则侵权必究！同时，我们也欢迎各位转载到自己的朋友圈，让更多的人能进入到SLAM这个领域中，让我们共同为推进中国的SLAM事业而努力！

商业合作及转载请联系liufuqiang_robot@hotmail.com

[阅读原文](#)