

文章编号: 1001-0920(2009)10-1568-05

机械臂视觉伺服系统中的高精度实时特征提取

于乃功, 秦永钢, 阮晓钢

(北京工业大学 电子信息与控制工程学院, 北京 100124)

摘要: 以机械臂视觉伺服系统为背景, 立体视觉所需要的特征点为目标, 提出一种基于颜色块的实时特征提取方法. 特征点的定位分为粗定位和精定位两个过程, 粗定位用来跟踪目标颜色块, 精定位在粗定位的基础上, 切出小图像并提取目标轮廓, 利用 $K-L$ 变换对轮廓点进行分类, 并通过最小二乘法对边缘点进行拟合. 根据直线方程计算顶点的亚像素精度坐标, 为下一步的目标位姿测量提供基础. 实验结果验证了该方法的有效性.

关键词: 视觉伺服; 颜色块; 目标跟踪; 特征提取

中图分类号: TP391.41

文献标识码: A

High accurate and real-time feature extraction in visual servoing systems for robotic arms

YU Nai-gong, QIN Yong-gang, RUAN Xiao-gang

(Electronic Information and Control Engineering College, Beijing University of Technology, Beijing 100124, China.
Correspondent: YU Nai-gong, E-mail: yunaigong@bjut.edu.cn)

Abstract: With the background of visual servoing for robotic arms and with the aim of features required by stereo vision, a method for real-time feature extraction based on the block of color is presented. The localization of feature includes coarse location and fine location. The coarse location is used to track the object. On the basis of coarse location, the fine location cuts a small image from the origin image, extracts the contour of the object from the small image and classifies the contour by $K-L$ transform, then fits the edge points by using least square error method. The sub-pixel coordinates of the feature points are calculated by the equations of lines, which provides foundation for the measurement of object's pose and position. The experimental results show the effectiveness of the method.

Key words: Visual servoing; Block of color; Object tracking; Feature extracting

1 引言

机械臂视觉伺服系统在机器人领域具有重要的作用. 传统的工业机械臂系统, 一般采用示教或离线编程的方式对加工任务进行路径规划和运动编程, 加工过程只是简单地重复预先编程设计的动作. 当加工对象或者工作环境发生变化时, 需要重新设计程序. 而利用视觉系统, 可以实时检测加工对象的信息, 并对这些信息应用人工智能的理论和 method 进行处理, 从而提高机器人的智能化水平. 借助机械臂视觉伺服系统不需要预先对机械臂的运动进行编程, 可节约大量的编程时间, 提高生产效率, 进一步提高产品的加工质量. 研究机械臂的实时视觉测量与控制系统, 对于提高机械臂的加工质量和加工效率及拓展机械臂的应用范围具有十分重要的意义^[1].

本文研究的视觉伺服系统是一种固定位置的双目摄像机系统, 如图 1 所示. 双目摄像机安装在机械臂本体外的固定位置, 采用基于位置的控制方式, 根据摄像机采集的图像估计目标的空间位姿, 并作为机械臂关节控制器的输入. 由于摄像机安装在机械臂本体以外的固定位置, 目标物体容易被机械臂遮挡^[2], 本文采取一种跟踪的机制, 可以有效防止目标因遮挡而造成的丢失. 基于位置的控制方式容易受摄像机标定误差和机械臂运动学模型误差的干扰^[3], 对目标特征的坐标精度要求比较高. 本文介绍一种跟踪基础上的高精度目标特征提取方法, 并具有较好的实时性.

2 系统介绍

机械臂视觉伺服系统如图 1 所示. 工作场景中

收稿日期: 2008-10-13; 修回日期: 2009-03-11.

基金项目: 国家 863 计划项目(2007AA04Z226); 北京市教委科技计划项目(KM200810005016).

作者简介: 于乃功(1966—), 男, 山东潍坊人, 教授, 博士, 从事机器人控制、机器视觉等研究; 秦永钢(1984—), 男, 山东临沂人, 硕士, 从事计算机视觉、立体视觉等研究.

有机臂、带有矩形蓝色色标的目标物体及双目立体摄像机,其中摄像机的位置与机械臂基座的位置相对固定.假设世界坐标系为 s_w ,机械臂基座坐标系为 s_r ,摄像机坐标系为 s_c ,目标物体坐标系为 s_o ,色块坐标系为 s_k .根据标定原理,可以标定出 s_c 相对于靶标的转换矩阵,机械臂各关节的转换矩阵是已知的,因此可以标定出 s_r 相对于靶标的转换矩阵,利用矩阵变换可以计算出机械臂基座坐标系 s_r 与摄像机坐标系 s_c 的转换矩阵为 rT_c .色块根据目标特征的需要任意安装在目标物体上,可以借助第 3 方的三维测量工具,如激光测距仪、高精度的双目摄像机等,分别计算目标物体和色块在第 3 方测量工具坐标系中的位姿,利用矩阵变换获得目标物体坐标系与色块坐标系的转换矩阵为 kT_o .

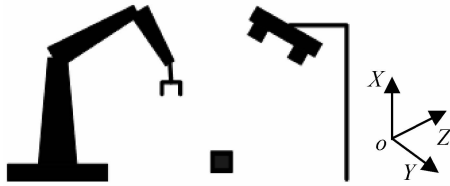


图 1 机械臂视觉伺服系统

如图 1 所示,目标物体在场景中,视觉系统的作用是计算目标物体在摄像机坐标系中的位姿.由于色标的特征比较明显,测量色标在摄像机坐标系中的坐标较为方便.在此系统中,视觉系统的作用是计算色标在摄像机坐标系中的位姿,假设测得色标相对于摄像机坐标系的位姿为 cP_k .

根据上面信息,可以得到目标物体相对于机械臂基座坐标系的位姿为 ${}^rP_o = {}^rT_c \cdot {}^cP_k \cdot {}^kT_o$,其中 rP_o 为机械臂关节控制器的输入.

获取目标物体的位姿最重要的一个步骤是在摄像机采集的图像中找出色标特征点在图像中的坐标,其工作流程如图 2 所示.

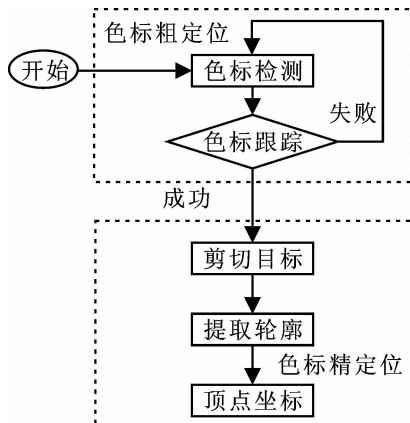


图 2 工作流程图

色标特征点的坐标提取过程分为两个子过程,分别为色标粗定位的过程和色标精定位的过程.

粗定位的具体过程为:在相机采集的原始图像中根据色标的颜色信息,在 HSV 空间中对图像进行分割,由此确定色标的位置;然后跟踪该色标,当跟踪失败时,系统会再次触发色标检测的过程,确定当前色标的位置.

精定位的具体过程为:根据粗定位得出的色标的中心位置和色标长和宽的大小,从原始图像中剪切得到目标图像,并从目标图像入手,检测色标的轮廓;通过 K-L 变换对轮廓点进行分类^[4],确定各个边的轮廓点;通过最小二乘法对各个边的轮廓点进行拟合,分别得出 4 条边的直线方程;根据直线方程计算 4 个顶点的亚像素精度坐标.

3 色标粗定位

从图 2 中可以看出,色标粗定位过程分为色标检测和色标跟踪两个过程.在色标粗定位过程中用到的主要是颜色信息,由于 HSV 能较好地反映人对色彩的感知和鉴别,能够分离颜色、亮度及饱和度信息,颜色分割时不易受到光线影响,鲁棒性较好,本文在 HSV 空间中对色标进行检测和跟踪.

通常使用的是 RGB 颜色空间,因此需要将 RGB 颜色转换到 HSV 空间,其转换公式为

$$H = \begin{cases} 60 \times \frac{G-B}{\max - \min} + 0, & \max = R, G \geq B; \\ 60 \times \frac{G-B}{\max - \min} + 360, & \max = R, G < B; \\ 60 \times \frac{B-R}{\max - \min} + 120, & \max = G; \\ 60 \times \frac{G-R}{\max - \min} + 240, & \max = B. \end{cases} \quad (1)$$

其中: $\max = \max(R, G, B)$, $H \in [0, 360]$.

3.1 色标检测

色标检测为色标跟踪提供色标的初始位置.色标检测根据图像 H 值的大小进行阈值分割,有

$$I(x, y) = \begin{cases} 255, & H > T_{\min}, H < T_{\max}; \\ 0, & \text{otherwise.} \end{cases} \quad (2)$$

其中: $I(x, y)$ 表示图像中位于点 (x, y) 处的像素值, $T_{\min}$ 和 $T_{\max}$ 表示阈值的范围.计算二值图像的重心,将该点作为目标跟踪的起始位置.

3.2 色标跟踪

本文采取的目标跟踪算法是基于连续自适应平均值迁移算法(Camshift 算法)^[5].Camshift 算法是指基于连续图像颜色动态变化的概率分布获得的一种有效的目标跟踪算法,具有鲁棒性和实时性好等优点.

Camshift 算法的核心是均值迁移算法

(Meanshift 算法), 该算法是一种迭代算法, 即先计算出当前点的偏移均值, 移动该点 to 其偏移均值; 然后以此为新的起始点, 继续移动, 直到满足一定的条件结束. 具体的算法过程如下:

Step1: 设定一个搜索窗口的大小;

Step2: 确定搜索窗口的初始位置;

Step3: 计算搜索窗口的质心;

Step4: 将新的搜索窗口的中心置于 Step3 得出的质心位置;

Step5: 重复 Step3 和 Step4, 直至收敛.

在 Step3 中, 对于颜色概率分布图像, 搜索窗的质心计算过程如下:

$$M_{00} = \sum_x \sum_y I(x, y); \quad (3)$$

$$M_{10} = \sum_x \sum_y xI(x, y),$$

$$M_{01} = \sum_x \sum_y yI(x, y); \quad (4)$$

$$x_c = M_{10}/M_{00}, y_c = M_{01}/M_{00}. \quad (5)$$

其中: M_{00} 为零阶矩, M_{10} 和 M_{01} 分别为 X 和 Y 的一阶矩, x_c 和 y_c 为搜索区域的质心, $I(x, y)$ 为图像中位于点 (x, y) 处的像素值, x 和 y 在整个搜索窗口范围内取值.

Camshift 算法根据需要寻找的目标颜色, 得出图像的颜色概率分布图, 然后在其搜索窗内使用 Meanshift 算法, 其算法流程如下:

Step1: 选择初始搜索窗的位置;

Step2: 使用 Meanshift 算法进行迭代并保存窗口零阶矩的大小;

Step3: 设置搜索窗的大小为 Step2 中零阶矩的一个函数;

Step4: 重复 Step2 和 Step3, 直至收敛.

当均值平移算法收敛时, 计算跟踪色块的位置以及长和宽的大小, 位置即为 (x_c, y_c) , 长和宽为

$$M_{20} = \sum_x \sum_y x^2 I(x, y),$$

$$M_{02} = \sum_x \sum_y y^2 I(x, y); \quad (6)$$

$$a = M_{20}/M_{00} - x_c^2, c = M_{02}/M_{00} - y_c^2, \\ b = 2(M_{11}/M_{00} - x_c y_c); \quad (7)$$

$$l = \sqrt{((a+c) + \sqrt{b^2 + (a-c)^2})/2}; \quad (8)$$

$$w = \sqrt{((a+c) - \sqrt{b^2 + (a-c)^2})/2}. \quad (9)$$

4 色标精定位

精定位的过程主要利用色标的轮廓或者边缘信息, 提取矩形色标的顶点坐标. 结合区域分割得出的轮廓和边缘检测的轮廓, 得到一个精度高的轮廓信息; 然后对各条边的点进行直线拟合, 利用直线方程

计算得到亚像素精度的顶点坐标.

4.1 剪切目标图像

根据图像跟踪模块得出的目标的中心位置和目標长和宽的大小, 确定所要剪切的图像的大小和位置, 为保证能将目标完整切割出来, 本文切割一个以 (x_c, y_c) 为中心, 边长为 $3\sqrt{(l/2)^2 + (w/2)^2}$ 的正方形图片, 如图 3 所示. 下面对切割出来的目标图像进行处理.



图 3 目标图像

4.2 提取轮廓

在 HSV 空间中, 根据式(2) 对目标图像进行阈值分割, 处理后二值图像的一个全局特征是目标(即蓝色色标)所在区域是图像中最大的白色区域. 对此二值图像进行区域标记, 寻找所有独立的白色区域中面积最大的区域, 即为蓝色色块所在的区域, 消除此区域之外的所有独立目标.

计算目标图像中目标的外轮廓线, 然后使用 Canny 算法对图 3 的目标图像进行处理, 得到边缘检测图. 以二值图像中白色区域的轮廓点所在的坐标为起始点, 在 Canny 边缘检测图中寻找最近的边缘点, 对所有的轮廓点依次进行处理, 即可从 Canny 边缘检测图中摘出蓝色色标准确的边缘线, 如图 4 中轮廓线所示.

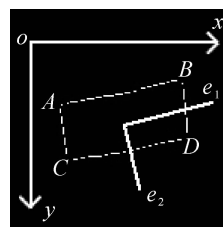


图 4 边缘检测

4.3 确定顶点坐标

从 Canny 边缘检测图中得出了色标的准确边缘信息, 但是这些边缘线是不连续的, 尤其是在顶点的位置. 先将轮廓图上的点区分为各条边上的点, 然后对各个边的点进行直线拟合, 根据直线方程计算顶点的坐标.

如图 4 所示, 在图像平面建立坐标系 xoy , x 轴沿水平方向, y 轴沿竖直方向. 将边缘点的图像坐标看作是一个二维向量, 以 x 和 y 的坐标值作为向量

的两个分量, 即

$$p_i = (x_i, y_i)^T, i = 1, 2, \dots, n.$$

令 $p_0 = (\bar{x}, \bar{y})^T$, 其中: $\bar{x} = Ex, \bar{y} = Ey, p_0$ 为图像重心点.

对于各条边缘线上的点, 其 x 和 y 坐标值间存在明显的线性相关性, 即 y 坐标值随着 x 标值的增加相应地增加或者减少. 通过 $K-L$ 变换^[7], 可以消除 x 和 y 坐标值之间的线性相关性. e_1 和 e_2 分别代表向量 $p_i (i = 1, 2, \dots, n)$ 的协方差矩阵特征向量的方向, 向量的方向分别与四方形轮廓的长边和短边的方向接近. 做 $K-L$ 变换有 $p'_i = T(p_i - p_0)$, 其中 $T = (e_1, e_2)^T$. 变换结果如图 5 所示.

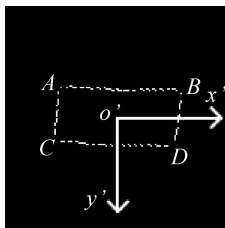


图 5 $K-L$ 变换

$K-L$ 变换建立了一个新的坐标系 $x'o'y'$, 原点在 o' , 坐标轴 x' 和 y' 的方向分别与四边形轮廓的长边和短边的方向基本相同. 根据下式寻找 4 个顶点 A, B, C, D , 并由此判断各条边的点:

$$\begin{aligned} A: \max(-x'_i - y'_i), B: \max(x'_i - y'_i), \\ C: \max(-x'_i + y'_i), D: \max(x'_i + y'_i). \end{aligned} \quad (10)$$

根据 4 条边 AB, BC, CD, AD 的点分别进行最小二乘法直线拟合, 得出 4 条直线方程为 $l_k: a_kx + b_ky + c_k = 0, k = 1, 2, 3, 4$.

根据直线方程便可计算得到顶点 A, B, C, D 的坐标, 分别为

$$\begin{aligned} x_{i,j} &= \frac{c_j b_i - c_i b_j}{a_j b_i - a_i b_j}, y_{i,j} = \frac{a_j c_i - a_i c_j}{a_i b_j - a_j b_i}, \\ i &= 1, 2, 3, 4, j = 1, 2, 3, 4. \end{aligned} \quad (11)$$

5 实验结果

利用上述方法, 对立体摄像机中左摄像机拍摄的色标图像进行跟踪和特征点提取的实验. 图 6(a) 为左摄像机原始采集的图像, 图像的原始大小为 $640 \text{ pixel} \times 480 \text{ pixel}$, 色标检测对图像进行阈值处理之后, 图片的重心位置为 $[367, 240]$. 目标作一个大致的矩形运动, 跟踪得出的目标的中心轨迹如图 6(b) 所示, 跟踪算法可以使目标锁定在当前物体, 而不受场景中同颜色物体影响. 实验表明, 当场景中出现同样颜色物体以及对目标形成遮挡时, 不影响目标的跟踪效果. 对色标进行跟踪之后切割出来的图像的大小为 $143 \text{ pixel} \times 144 \text{ pixel}$, 如图 6(c) 所示. 图 6(d) 为对图像进行阈值处理之后的效果图. 图 6(e) 为对二值图像进行区域标记去除噪音之后的效果. 图 6(f) 为根据区域检测得出的轮廓图, 此轮廓图精度不高. 图 6(g) 为对目标图像进行 Canny 边缘检测之后的效果图. 图 6(h) 为根据图 6(f) 和图 6(g) 寻找出来的边缘轮廓图. 图 6(i) 为对图 6(h) 的边缘点进行 $K-L$ 变换之后的效果. 图 6(j) 为曲线拟合得出的直线.

计算所得的 4 条直线方程如图 6(k) 所示, 分别为

$$\begin{aligned} l_{AD}: y - 66.5556 &= 36.4017(x - 34.5556), \\ l_{CD}: y - 92.3803 &= 0.1329(x - 70.8082), \\ l_{BC}: y - 77.5263 &= -49.7464(x - 109.868), \\ l_{AB}: y - 52.6667 &= 0.1277(x - 74.7179). \end{aligned}$$

计算所得的 4 个顶点坐标为

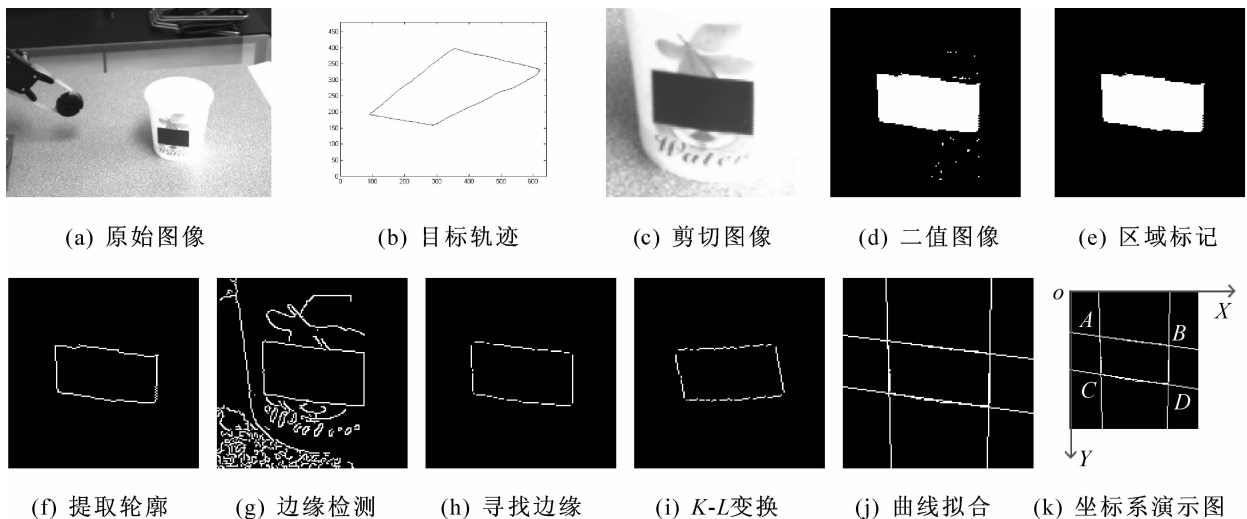


图 6 实验结果

$A(34.0313, 47.4718)$, $B(110.2769, 57.2069)$,
 $C(35.1344, 87.6259)$, $D(109.4662, 97.5339)$.

目标图像放大 7 倍, 根据上面的结果, 查看 A 点的坐标在图像中的位置. 如图 7 中圈点所示, 可以看出, 通过直线拟合, 能够得到准确的特征点坐标.

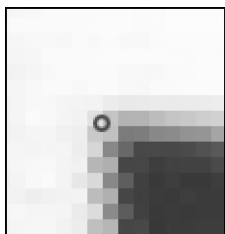


图 7 结果演示图

6 结 论

本文针对机械臂的视觉伺服系统, 提出了一种实时的目标跟踪和特征提取方法. 通过为目标物体添加色标, 分别利用色标的颜色信息和边缘信息, 实现目标的粗定位和精定位. 粗定位是一个检测和跟踪的过程, 使用 Camshift 跟踪算法可以实时跟踪目标, 同时又可以避免同场景中颜色相近物体的干扰, 提高了目标定位的鲁棒性. 精定位结合区域检测得出轮廓和 Canny 边缘检测图, 并得出目标的精确轮廓. 最小二乘法对各条边的边缘点进行直线拟合, 通过直线方程得出顶点的亚像素精度坐标.

本文在大图像上使用了实时性高的 Camshift 算法, 既跟踪了目标又提高了运算速度; 然后对小图片进行边缘和顶点的计算, 既节省了计算量, 又避免了噪音的影响. 实验结果表明, 这是一种有效的目标跟踪和特征提取方法, 此方法为立体摄像机计算目标的位姿, 实现机械臂的视觉伺服奠定了基础.

参考文献 (References)

- [1] 徐德, 谭民, 李原. 机器人视觉测量与控制[M]. 北京: 国防工业出版社, 2008.
 (Xu D, Tan M, Li Y. Visual measurement and control for robots [M]. Beijing: National Defense Industry Press, 2008.)
- [2] 赵清杰, 连广宇, 孙增圻. 机器人视觉伺服综述[J]. 控制与决策, 2001, 16(6): 849-853.
 (Zhao Q J, Lian G Y, Sun Z Q. Survey of robot visual servoing[J]. Control and Decision, 2001, 16(6): 849-853.)
- [3] 王麟琨, 徐德, 谭民. 机器人视觉伺服研究进展[J]. 机器人, 2004, 26(3): 277-282.
 (Wang L K, Xu D, Tan M. Survey of research on robotic visual servoing[J]. Robot, 2004, 26(3): 277-282.)
- [4] 沈扬, 徐德, 谭民. 仿人形机器人火炬传递中的高精度目标特征提取[J]. 传感技术学报, 2005, 18(4): 822-827.
 (Shen Y, Xu D, Tan M. Study on high accurate feature extraction in the process of transferring touch with humanoid robots [J]. Chinese J of Sensors and Actuators, 2005, 18(4): 822-827.)
- [5] Gary R Bradski, Santa Clara. Computer vision face tracking for use in a perceptual user interface [J]. Intelligent Technology J, 1998, 2(2): 214-219.
- [6] 阮秋琦. 数字图像处理[M]. 北京: 电子工业出版社, 2003.
 (Ruan Q Q. Digital image processing [M]. Beijing: Publishing House of Electronics Industry, 2003.)
- [7] 章毓晋. 图像工程(上册)——图像处理[M]. 北京: 清华大学出版社, 2006.
 (Zhang Y J. Image engineering — Image processing [M]. Beijing: Tsinghua University Press, 2006.)

(上接第 1567 页)

- [6] Vijayan T, Kumaran M. Inventory models with a mixture of backorders and lost sales under fuzzy cost[J]. European J of Operational Research, 2008, 189(1): 105-119.
- [7] Zeng L. Expected value method for fuzzy multiple attribute decision making [J]. Tsinghua Science and Technology, 2006, 11(1): 102-106.