

An Improvement on Workpiece Image Matching via Locality Sensitive Hashing

Xiuli Shao, Zihao Dong, Zihan Li, Jingya Yang

College of Computer and Control Engineering, Nankai University, Tianjin
Email: 641069042@qq.com

Received: May 9th, 2018; accepted: May 23rd, 2018; published: May 30th, 2018

Abstract

The extraction and matching of workpiece contour is one of the most important steps in robot sorting. The traditional contour matching algorithm uses Hausdorff distance, with considerable amount of calculation and difficulty in determining the position and angle of the workpiece. Based on the template matching, the image feature vector is used to match the image in this paper, which reduces the computation and solves the problem of image recognition. We also improve the traditional linear search algorithm by using the Locality Sensitive Hashing algorithm. Moreover, the hash algorithm is improved to reduce the storage space. In the experiment, the algorithm can effectively ignore the problems caused by the translation, zoom and rotation of workpiece during image recognition. What's more, it can improve the efficiency and accuracy of image search to a certain extent.

Keywords

Workpiece Matching, Locality Sensitive Hashing, And/Or Operation

工件图像匹配的局部敏感哈希应用改进

邵秀丽, 董子昊, 李子涵, 杨静雅

南开大学计算机与控制工程学院, 天津
Email: 641069042@qq.com

收稿日期: 2018年5月9日; 录用日期: 2018年5月23日; 发布日期: 2018年5月30日

摘要

工件轮廓的提取与匹配是机器人分拣中最重要的一环。传统的轮廓匹配是采用Hausdorff距离进行模板

匹配,但是计算量过大且要求工件的位置和角度一定。本文在模板匹配的基础上,采用图像特征向量进行图像匹配,降低了计算量并解决了工件的几何变换带来的图像识别问题。改进传统的线性搜索,采用敏感哈希算法匹配图像,并且对哈希算法做出了一定的改进,降低了存储空间。通过实验发现,本文所采取的算法可以有效的忽略在图像识别中工件平移、缩放、旋转所带来的问题,并且在一定程度上提高了图像搜索的效率和准确度。

关键词

工件匹配,局部敏感哈希,与或操作

Copyright © 2018 by authors and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 概述

在自动化工业分拣过程中,图像的匹配是非常重要的的一环,要求一定的实时性和很高的准确率。在目前的工业分拣中,常采用模板匹配的方法。但是工件的平移和旋转等位置的不确定性给模板匹配带来了很大的困难。因此,研究一种可以快速准确进行匹配,忽略位置光线带来的干扰的匹配算法是非常的必要的。

国外 Bourret 采用多尺度算法进行边缘检测和边缘闭合,对获得的边缘特征模拟退火算法极小化分割函数来完成图像的匹配[1]。国内梁勇等人使用特征点匹配的方法进行工件的边缘匹配,但由于工件的几何形状简单,特征点的信息不足,常使得匹配的结果不够准确,且该算法只解决了图像的平移问题[2]。华中科技大学的张学军在图像匹配过程中采用了支持向量机(SVM)的判决功能,解决了匹配只能用单一特征作为判别条件的缺陷。使用最小二乘支持向量机对已有的数据进行分类训练,然后用训练好的向量机检测目标[3]。刘健庄、谢维信等人使用 Hausdorff 距离作为度量图像相似度的标准,使用遗传算法进行检查,该方法取得了一定的效果但是常规遗传算法的局部寻优效果不好,受参数影响非常大且容易产生早熟情况,且 Hausdorff 距离的计算时间较长,降低了工件分拣的效率[4]。中国科技大学与中科院合肥智能机械研究所合作研究了基于机器视觉的瑕疵工件检测系统[5],该系统采用图像处理及模式匹配的方法识别工件,可以通过与模版对比,识别出做工有瑕疵不能在工业生产中使用的工件,并将其分拣出来,该系统在工业生产线上具有很好的实用性。为了避免参数影响和早熟的情况,该算法采用线性搜索的方法,增加了计算的时间。

本文基于中科大研究的瑕疵工件匹配系统中采用的模板匹配思想基础上提出了一种可忽略工件几何变换的图像识别系统并且改进了线性搜索方法,即利用局部敏感哈希算法对工件图像进行处理,最终可以快速高效的识别匹配工件类型,应用于自动化流水作业中图像处理阶段。在获得了工件图像的特征向量之后,需要提取模板的特征向量进行存储,对采集到的新的工件图像进行迅速的匹配。这里可以使用特征向量之间的欧氏距离作为两个工件的相似性度量,欧氏距离越小则代表两个图片越相似,欧氏距离小于设定的阈值,则代表两工件属于同一类别。而在特征库非常庞大的情况下,为快速有效匹配计算欧氏距离并排序,本文采用局部敏感哈希算法进行相似特征的快速搜索。考虑到本文所使用的特征向量长度仅为 8,因此本文用之前提取的特征向量建立了大小为 $4 * 8$ 的哈希表,即指有四张哈希表,每个表有八个桶。因此,本文主要介绍在提取出特征向量之后进行迅速匹配的方法。

2. 基于局部敏感哈希存储提取的工件图像特征

为方便匹配时可以快速的进行检索,使用哈希算法进行提取的工件图像特征的存储,在获得需要匹配的向量后直接对其进行投影即可找到相似的向量。但是由于传统的哈希算法在投影过后不能保证距离不变,因此需要对传统哈希进行改进,改变其投影的哈希函数,使其投影过后的元素距离不变,也即采用局部敏感哈希算法。为此先给出几个定义:

桶: Hash 又被翻译为“散列”,是把任意长度的输入转变为一个固定长度的输出。采取映射的办法。其中映射方法又被称之为散列算法,输出被称为散列值,即桶。

Hash 桶的性质: 如果两个输入的元素经过映射后所在的桶不同,那么这两个元素一定不相同。而如果映射后在同一桶中,则说明两元素有可能相同但并不是一定相同。

如图 1 所示,传统哈希算法不具有距离不变性,即经过映射以后,原本距离很近的两个元素可能被分在距离较远的两个桶中,也即映射后桶的距离不能代表原输入的两个元素的距离;而 locality-sensitive hashing(局部敏感哈希)简称 LSH,则是一种保距算法[6]。

若定义 $d(x, y)$ 是 x, y 两元素之间的距离,则 LSH 的映射算法则需要满足以下两个条件,其中 d_1, d_2 是根据实际需要规定的值:

- 1) 若 $d(x, y) \leq d_1$, 则两元素被映射到同一个桶中的概率,即 $h(x) = h(y)$ 的概率至少为 p_1 。
- 2) 若 $d(x, y) \leq d_2$, 则两元素被映射到同一个桶中的概率,即 $h(x) = h(y)$ 的概率至多为 p_2 。

其中, d_1 是最大的相似距离,与 d_1 的距离越小则映射到同一桶的概率就越大; d_2 是最小的相似距离,与 d_2 的距离越大则映射到同一桶的概率就越小。

满足上述两个条件的哈希映射函数被称为-sensitive。

3. 工件特征的哈希表的生成

一般在快速寻找元素时会采取哈希算法,但是传统哈希算法并不能解决距离不变的问题,因此本论文采取了保距的局部敏感哈希算法。说明了局部敏感哈希算法的性质,不同的距离度量方法以及其相应的哈希映射函数,然后介绍了哈希表的生成方法和哈希函数族,并提供了在搜索过程中提高容错度与准确度的方法。

本文采用局部敏感哈希算法,建立多个哈希表存放工件模版的特征向量,然后用不同的哈希函数将需要查找的向量投影到不同的表中,再通过对不同桶的与或操作获得最终的相似向量。

图 2 给出了局部敏感哈希原理, $H_{\eta_1 \dots \eta_k}(Q)$ 为其中一个样本, $H_{\eta_1 \dots \eta_k}(N)$ 为另一个样本,将两个样本经 hash 函数转换为 0, 1 的 sim Hash 值,最后经汉明码计算,便可以用汉明码衡量两者的相似程度。

提取到的工件特征是一个 1×8 的向量,随机选取其中的六位,并设定其阈值 T_1, \dots, T_6 ,若某位的值大于该位阈值,则该位哈希值取 1,否则取 0。如工件特征向量为 (101, 200, 403, 529, 98, 398, 268, 127), 哈希函数选取其中的第 1 到 6 位,阈值设定为 (150, 300, 300, 450, 100, 350), 则其哈希值为 001101。选取不同的位数,设定不同的阈值以生成多个哈希表以便提高准确度降低存储空间。

图 3 给出了工件图像进处理的哈希表的生成的几个主要步骤:

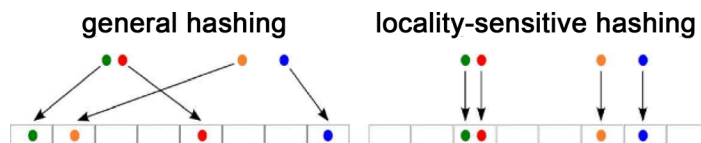


Figure 1. Comparison of local sensitive Hash and traditional Hash
图 1. 局部敏感哈希与传统哈希比较

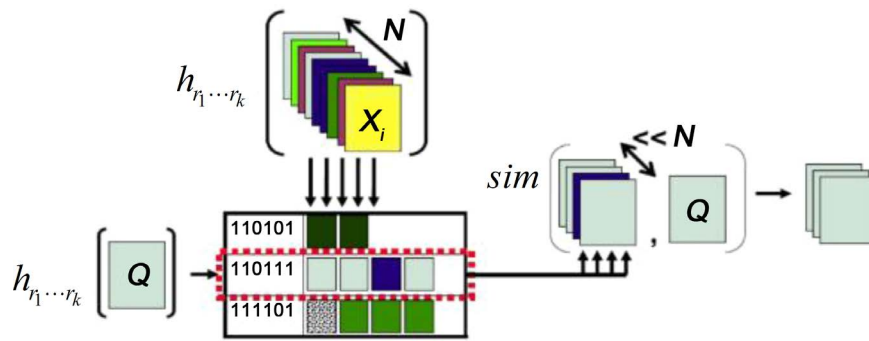


Figure 2. The schematic diagram of LSH
图 2. 局部敏感哈希原理图



Figure 3. The generation process of Hash table
图 3. 哈希表生成过程

其中主要包括保距投影、哈希函数族映射和哈希表。下面分别介绍相应的工作内容。

3.1. 保距投影

本文为了减少存储空间，对哈希函数做了一定的改进。首先对传统的欧式空间进行投影，将其转变为方便比较计算的汉明空间且要求投影过后距离不变。其投影的具体实现方法如下：

对于任意一点 $p \in P$ ， $p = \{x_1, x_2, x_3, \dots, x_n\}$ ， P 为 d 维的原始空间，其投影函数为：

$$V(p) = \text{Unary}_c(x_1)\text{Unary}_c(x_2)\text{Unary}_c(x_3)\dots\text{Unary}_c(x_n) \tag{1}$$

表示一个由 x 个 1， $c-x$ 个 0 构成的序列。为了减少存储空间，本文定义 $\text{Unary}_c(x)$ 是指当 $x > t$ 时，其指为 1，反之为 0。其中 t 时根据特征库范围设定的阈值。在投影过后，对于任意两点 $p, q \in P$ ， $d(p, q) = dH(v(p), v(q))$ ， dH 表示在汉明空间下的两点之间的汉明距离。

汉明距离小的两个向量，在经过汉明距离的哈希映射函数映射过后，映射到同一个桶中的概率更大。而为了增强搜索能力，增大汉明距较小的两个向量的冲突概率，使用多个哈希表。

3.2. 哈希函数和哈希表的形成

接下来需要对获得的汉明空间下的向量进行映射，同样需要保证映射过后在同一桶中的元素其原始数据也是相近的。而且为了增加冲突率提高搜索效率使用多个表相应的也就需要使用多个哈希函数。本文使用的具体哈希函数如下：

对于 d 维空间，定义 $I = \{1, 2, 3, \dots, d\}$ ，取 L 个 I 的子集，子集的长度相同均为 i ，分别记为 $I_1, I_2, I_3, \dots, I_n$ ，定义 $p|I$ 是 p 在 I 上的投影，也即 I 中每个数值作为位置索引，取 p 对应位置的值并串联起来。

对于 L 个子集，建立 L 个哈希表。对于 $j \in [1, L]$ ，记 $H_j(p) = p|I_j$ 。对汉明空间中的点采用该映射函数映射到不同的桶中。哈希函数族以及其映射结果如表 1 所示。

4. 工件的图像匹配

在搜索时，可以迅速找出哈希表中与当前工件图像的特征向量欧式距离最短的向量，进而得到工件的类型，对工件进行快速的分类。

Table 1. Hash function and Hash table
表 1. 哈希函数与映射后的哈希表

Sub-Hash table	Hash function	Hash families
T_1	$H_1(p) = p I_1$	$H_1(p_1)H_1(p_2)\cdots H_1(p_i)$
T_2	$H_2(p) = p I_2$	$H_2(p_1)H_2(p_2)\cdots H_2(p_i)$
T_3	$H_3(p) = p I_3$	$H_3(p_1)H_3(p_2)\cdots H_3(p_i)$
...
T_L	$H_L(p) = p I_L$	$H_L(p_1)H_L(p_2)\cdots H_L(p_i)$

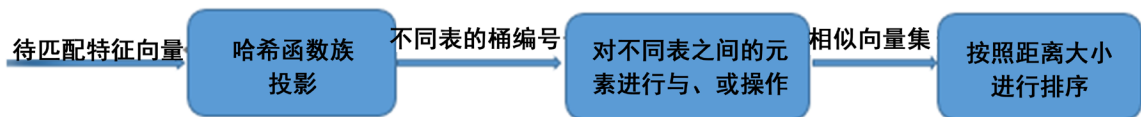


Figure 4. The flow chart of image matching
图 4. 图像匹配流程图

图像匹配过程即寻找与该特征向量距离最为相近的向量，然后获得其图像信息。具体的匹配步骤如图 4 所示。

4.1. 与或操作

在经过哈希函数族进行映射以后，会获得几个表中桶的编号，根据桶中的元素数量，以及为了保证投影过后元素的邻近概率和错误率，本文采取与、或操作进行平衡，具体做法如下：

1) 与操作

从哈希函数族中挑出 k 个哈希映射函数，当且仅当 k 个 $H_i(p) = H_i(q)$ 均成立时，才会被投影到相同的桶内。这种操作大幅度降低了分配在同一桶中但向量不邻近的概率，但是一定程度上降低了找到邻近数据的概率。

2) 或操作

从哈希函数族中挑出 k 个哈希映射函数，只要在 k 个 $H_i(p) = H_i(q)$ 中有一个成立，就会被投影到相同的桶内。这种操作一定程度上提高了找到邻近数据的概率，但是也在一定程度上提高了分配在同一桶中但向量不邻近的概率

4.2. 桶内排序

桶内排序即对距离进行排序。距离是用来度量两个向量之间相似度的参数，距离越大则相似度越低，距离越小则相似度越高。而向量的距离之间有不同的表示方法，具体有以下几种：

1) jaccard 距离

jaccard 相似度是指两集合交集的元素个数比上两集合并集的元素个数，具体公式如下：

$$\text{jaccard distance} = 1 - \frac{|A \cap B|}{|A \cup B|} \quad (2)$$

2) 夹角距离

夹角距离指两个向量之间的夹角越小则表示其距离越近。也可以理解为使用了一个随机的超平面对原始的数据空间进行划分，而每个数据在被投影后都会落到超平面的某一侧，认为投影后落在同侧的数据有很大可能是相邻的，也即原始数据之间的夹角很小。

3) 欧几里得距离

欧几里得距离又称欧氏距离，是常用的衡量 D 维空间内两个点之间的距离的方式，其表达式为：

$$\begin{aligned} A &= [x_1, x_2, x_3, \dots, x_n] \\ B &= [y_1, y_2, y_3, \dots, y_n] \\ \text{distance} &= \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + (x_3 - y_3)^2 + \dots + (x_n - y_n)^2} \end{aligned} \quad (3)$$

考虑到算法的时间复杂度以及精确度，本文选择欧几里得距离作为向量之间相似度的度量。

5. 主要函数的实现

本文哈希表的生成涉及到以下几个主要函数：

1) lsh 函数，用户调用该函数，并传入哈希表中表的个数 l 和桶的个数 k 以及原始数据空间的维数 d 和原始数据 varagin 便可以得到映射过后的哈希表

2) lshfunc 函数，创造随机的哈希函数族，在 lsh 函数中被调用。

3) lshprep 函数，创建 l 个哈希表，被给每个哈希表设置其哈希函数。

4) lshins 函数，在 lsh 函数中被调用，将原始数据通过上述 lshfunc 函数获得的哈希函数族映射到哈希表中。且为了减少占用的内存，这里表中并不存储特征向量，只存储特征向量在向量库中的索引值。

5) lshlookup 函数是局部敏感哈希的查询代码，其中传入参数 temp_T 是当前工件的特征向量，Patches 是模板特征库，T1 是已经建立好的哈希表，函数可以返回与 temp_T 投影在同一桶中的特征向量的个数，以及其在特征库中的索引。

6) lsh(l, k, d, x, varagin) 函数调用 lshfunc 生成函数族，然后使用 lshprep 创建哈希表，调用 lshins 进行投影，其核心代码较为简单，这里不做赘述。

7) lshfunc 函数，按照用户传入的桶的个数随机选取几个维度，给这些维度设置一个阈值 t，hash 函数为 $x(d) \geq t$ 。

其中，lshins 函数作为映射函数非常重要，传入参数为 lshprep 已创建好的哈希表和原始数据，由于函数只给数据创建索引但不存储，因此原始数据不能删除以便后期追溯。该函数的流程图如图 5 所示。

6. 实验测试

6.1. 图像匹配

本文算法的实现是基于 MATLAB 进行开发的。本文共采集了 100 种工件的模板，建立了 $4 * 8$ 的哈希表，并对这些工件做出一些几何变换如平移、缩放、旋转等，对新产生的工件进行搜索匹配，观察是否能准确的对工件进行分类。

本文采用 canny 算法提取工件检索数据库(如图 6)中的每个工件的轮廓，然后选择其中一种工件，利用图像的仿色变换得到处理后的扭曲工件边缘(如图 7 所示)；将扭曲的工件边缘图像重新放到数据库中。图 6 右下角的图片就是提取的钥匙轮廓。

最后利用 LSH 算法在数据库检索选择的图像，其中 LSH 算法测试的核心部分如下：

```
1 #set pictures index by lsh
2 lsh.index(np.array(training).flatten().tolist())
3 #query the target image
4 result = lsh.query(np.array(test1).flatten().tolist())
```

LSH 检索器得出的检索结果如图 8 所示。即选择图 7 仿射变换后的工件轮廓为测试集，通过 LSH 算

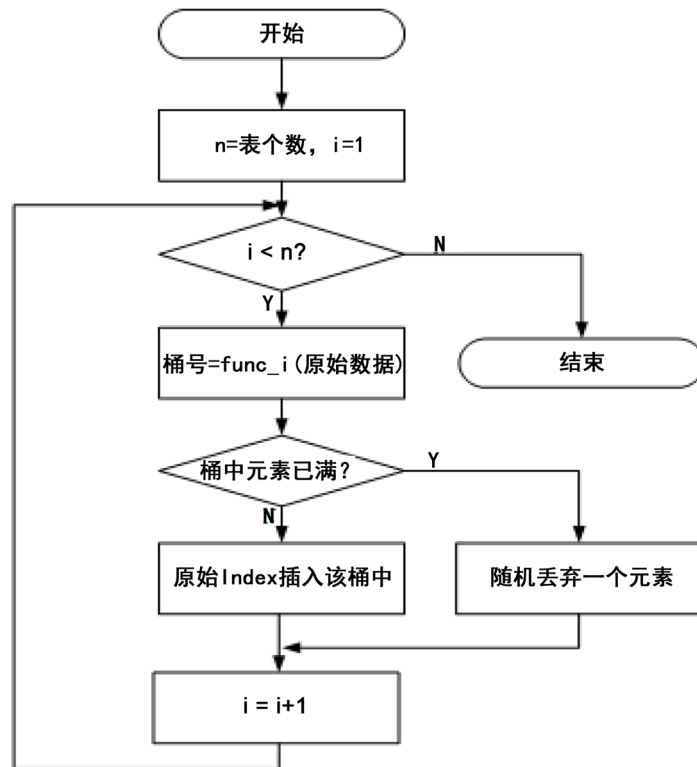


Figure 5. The flow chart of lshins function.

图 5. lshins 函数流程图



Figure 6. The workpiece dataset

图 6. 工件检索数据库

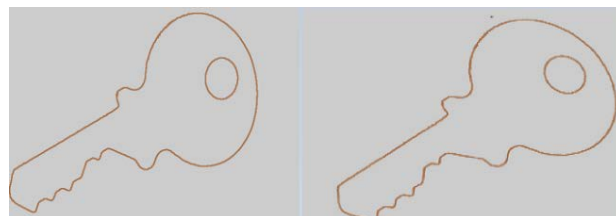


Figure 7. The Edge of the original workpiece and the one after affine transformation.

图 7. 原始工件轮廓与仿射变换后的工件轮廓

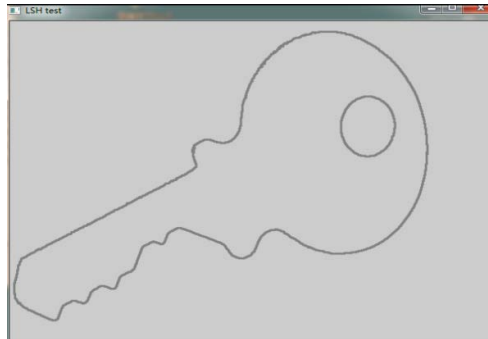


Figure 8. The retrieval result of LSH
图 8. LSH 得到的检索结果

法得到检索集内的原始工件轮廓。

为了更进一步地检验改进的 LSH 检索器的检索效果,本文采用了 Cross-Correlation 的相关性计算方法,计算公式如下:

$$r = \frac{\sum_i [(x(i) - mx) * (y(i) - my)]}{\sqrt{\sum_i (x(i) - mx)^2} \sqrt{\sum_i (y(i) - my)^2}} \quad (4)$$

其中, mx 为工件轮廓 x 坐标和的平均值, my 为工件轮廓 y 坐标和的平均值。

通过 9 次轮廓迭代检测和相关度检测实验,可以得出改进后的 LSH 检索器的检索工件相关度为 93.6%,测试结果如图 9 所示。

6.2. 特征存储

在系统运行之前,先调用 SAVE 函数采集 100 张图片的特征向量,每个工件图像提取其 36 个不同方向上的特征向量,生成一个大小为 $3600 * 8$ 的模板特征库,之后运行系统。在系统界面生成过程中,预先加载了特征向量库并且对其进行投影映射生成 $4 * 8$ 的哈希表以便后期使用。在 openbutton_Callback 函数中,调用文件系统打开图像文件,由用户选择需要匹配的工件图像,显示该图像并提取图像的特征向量。在 searchbutton_Callback 函数中,对该特征向量进行映射,然后提取出在同一桶内的特征向量,计算出其欧氏距离最短的一个,然后显示该图片。系统的整体运作流程如图 10 所示。

7. 实验结果

采用工业生产中不同形状的钥匙以及齿轮作为特征库,对工件图像进行匹配分类,当工件发生位移时、旋转时和缩放时进行匹配。实验表明,本系统可以有效的解决图像的几何变换,对于平移、旋转、缩放过后的工件图像均有较好的识别效果,且搜索速度大幅提升。而对于模板中没有的工件,则在搜索中可能不能找出其视觉上相似的图片。在实际工件流水线中,需要耗时的步骤有提取当前工件图像的特征向量和搜索图像,由于边缘方向直方图的计算量较大,特征向量提取约需要 1 ms,搜索时间仅需 0.05 ms,能够有效应用于实际生产当中。

8. 结论

本文基于边缘方向直方图作为工件特征向量,采取局部敏感哈希算法匹配图像,相对于传统的哈希,此算法既可以迅速的找到目标所在的位置,又保证了投影过后位置相邻的元素其原始数据也非常相近。正好满足图像匹配的需要,而且耗时很短。


```

segments[il.size() 5
segments[il.size() 12
segments[il.size() 9
segments[il.size() 27
9 - 3.1
51 - 19.4
56 - 2.5
73 - 3.6
101 - 11
135 - 12.7
162 - 7.1
168 - 2.5
186 - 7
segments[il.size() 42
segments[il.size() 5
segments[il.size() 17
segments[il.size() 28
segments[il.size() 34
segments[il.size() 27
segments[il.size() 6
segments[il.size() 18
segments[il.size() 23
0.936019

```

Figure 9. The correlation of real workpiece edge map and the one retrieved by LSH method
图 9. 测试工件轮廓与 LSH 检索轮廓的相关度测试

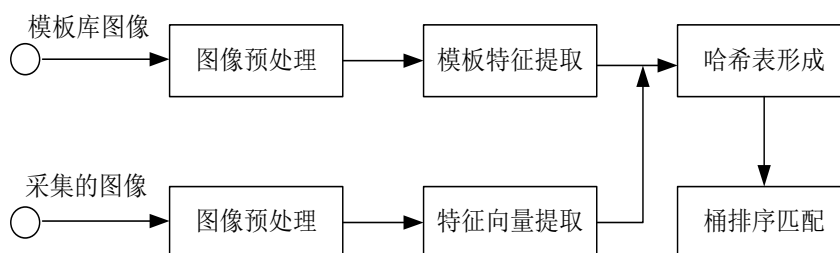


Figure 10. The overall flow chart of this retrieval system
图 10. 系统整体流程图

本文有效解决了在实际操作中工件位置变化和光线不同等情况给识别带来的困难，提高了搜索的速度，但是还存在一些不足，比如：在局部敏感哈希算法中对模板进行投影时，如果一个桶中的元素数已经满了，本文采取了随机抛弃一个元素的做法，对结果造成了一定的误差。可以尝试采取二级哈希表的做法，也即在桶满了以后，在桶中占用一个元素的位置另建一个桶。此外，本文仅考虑了对图像匹配速度要求不是很高但对精度要求非常高的情况，对于另外一种情况，即需要较快的识别速度，且只需要对工件大类进行分拣的情况，可以使用边缘特征点运行机器学习算法，这种方法可以大大提高匹配速度，且在数据比较多时，精准度也会有所提高。

基金项目

16ZXHLGX00250、15ZXHLGX00360、15ZXHLGX00380、15ZXDSGX00090。

参考文献

- [1] Bourret, P. and Cabon, B. (1995) Neural Approach for Satellite Image Registration and Pairing Segmented Areas. *SPIE*, 2576, 22.
- [2] 梁勇, 祝明波, 杨汝良. 一种基于 Hausdorff 度量的多传感器图像配准方法[J]. 遥感技术与应用, 2006, 21(5): 473-476.
- [3] 张学军. 基于支持向量机的红外图像匹配研究[D]. 华中科技大学, 2007.

- [4] 刘健庄, 谢维信, 高新波, 等. 基于 Hausdorff 距离和遗传算法的物体匹配方法[J]. 电子学报, 1996.
- [5] 贺雅琴. 自动物料分拣机器人系统的关键技术研究[D]. 华南理工大学, 2011.
- [6] 凌康, 武港山. 基于位置敏感哈希的相似性搜索技术研究[D]. 南京大学, 2012.

知网检索的两种方式:

1. 打开知网页面 <http://kns.cnki.net/kns/brief/result.aspx?dbPrefix=WWJD>
下拉列表框选择: [ISSN], 输入期刊 ISSN: 2161-8801, 即可查询
2. 打开知网首页 <http://cnki.net/>
左侧“国际文献总库”进入, 输入文章标题, 即可查询

投稿请点击: <http://www.hanspub.org/Submission.aspx>

期刊邮箱: csa@hanspub.org