

Design of an Image Feature Extraction and Archiving Module in ROS Environment

Zexi Meng¹, Kuan Jin¹, Yunxi Zhang^{1,2*}

¹School of Automation and Electrical Engineering, Tianjin University of Technology and Education, Tianjin

²Tianjin Key Laboratory of Information Sensing and Intelligent Control, Tianjin University of Technology and Education, Tianjin

Email: zyunx2004@126.com

Received: Jun. 20th, 2019; accepted: Jul. 9th, 2019; published: Jul. 16th, 2019

Abstract

The purpose of this design is to add the feature of "automatic feature extraction-recognition" into the existing color feature recognition system, and construct a feature extraction archiving module with an independent image data transceiver interface and a feature file read-write interface of the specified framework in the ROS (Ubuntu) environment. By integrating the basic functions of Opencv and XmlLib2, we made our own image transceiver interface class and feature file read-write class, and wrote an extraction algorithm based on RGB color model for feature extraction, so that this module can adapt to ROS environment. After several static tests, the extraction and archiving module of this design basically achieves the design goal of sampling and extracting at the same proportion of the designated area, screening features according to the cumulative number, and archiving features in the designated directory.

Keywords

ROS, Color Recognition, Color Extraction, Gray Level Compensation

一种ROS环境下的图像特征提取归档模块设计

孟泽熙¹, 靳宽¹, 张运喜^{1,2*}

¹天津职业技术师范大学自动化与电气工程学院, 天津

²天津职业技术师范大学天津市信息传感与智能控制重点实验室, 天津

Email: zyunx2004@126.com

收稿日期: 2019年6月20日; 录用日期: 2019年7月9日; 发布日期: 2019年7月16日

*通讯作者。

摘要

本设计的目的是为了在现有的颜色特征识别系统中添加“自动提取特征 - 识别”的特性, 在ROS (Ubuntu)环境下构造一个具有独立图像数据收发接口和指定框架的特征文件读写接口的特征提取归档模块。通过集成Opencv、XmlLib2的基本功能函数, 自制图像收发接口类、特征文件读写类, 并为特征提取编写了以RGB颜色模型为基础的提取算法使该模块能够适配ROS环境。在经过数次静态测试后, 本设计的提取归档模块基本达到了对指定区域的等比例抽样提取、按累计数量筛选特征、将特征归档于指定目录下的设计目标。

关键词

ROS, 颜色识别, 颜色提取, 灰度补偿

Copyright © 2019 by author(s) 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. 引言

颜色作为可见对象的一个主要特征, 在物品识别和物体追踪方面应用非常广泛[1] [2] [3] [4]。包含颜色识别的投入实际应用的技术也有很多, 比如: 车辆号牌识别、条码识别、面部信息采集等[5] [6]。其中大多数计数研究采用给定样本的方法, 也有一部分研究仍停留在 Matlab 的平台仿真验证的阶段。

本设计以工学角度出发, 尝试在不优先考虑性能的情况下设计了一个以使用为基本目的的颜色提取器。根据一般设计规律, 本文将设计中面对的所有问题简化为了四个问题: 软件结构、算法实现、结果检验、抗干扰, 并重点说明其中的颜色提取过程和灰度参考值生成的实现和优化。

本设计的设计目的在于: 通过在开环的颜色识别系统中加入具有自评估能力的颜色样本提取器模块, 使“当系统得到不符合要求的颜色样本时能够利用提取器产生颜色提取任务的回滚”这样一个可以基于PID技术的最优样本生成的行为成为可能, 提高实时识别系统的弹性。

2. 原型设计

接下来将说明原型的设计方案(Alpha版)。由于原型的目的在于功能实现, 并没有仔细考虑架构、动态性能、复验结果等问题, 所以其中存在一些不太合理的结构。大部分问题将在之后的第一改进方案(Beta版)中给出解决方案。

2.1. 基本设计要求

1) 我们针对的是 kinect2 的体感摄像头传回的数据, 当然这并不意味着我们就不需要读取图像文件来作为提取图像。所以, 根据 ROS 关于话题的特性, 我们需要一个接口类去订阅 kinect2_bridge 发布的图像数据, 由于我们并不需要处理过时的图像, 所以这个缓存池只需要一幅最新的图像即可, 然后通过提取器的查询式接口传回数据。

2) 我们处理对象是八位 BGR 三通道像素。这是 Opencv3 中能够使用的最小的彩色像素的格式。即使如此, 每个像素仍具有 2^{24} 种组合结果。所以我们需要采用可调节的减色算法减少提取结果的复杂度并将减色比例标注在结果文件中。

3) 为了提高提取速度和提取到的特征的纯净度, 我们需要可自主选取地在图像的某个区域内进行提取。同时, 为了防止提取区域过大导致被提取像素呈平方化增长, 我们需要规定最大提取总数, 并且根据提取总数与提取区域的比例自动调节 x 轴 y 轴的提取步距。

4) 为了积累实验数据, 我们需要将每次的识别结果保存下来, 为之后的统计识别结果。所以, 我们需要一个基于 XML 文件技术的可指定路径且以固定格式保存特征文件的读写接口。

2.2. 原型的功能结构示意图

通过如《2.1 基本设计要求》的描述, 我们基本可以得出如下图(图 1 模块功能结构示意图)。图中主要说明了《2.1 基本设计要求》中的 1), 4) 两点以及其二者必要的环境, 关于 2), 3) 两点将在《2.6 特征信息提取流程》说明。

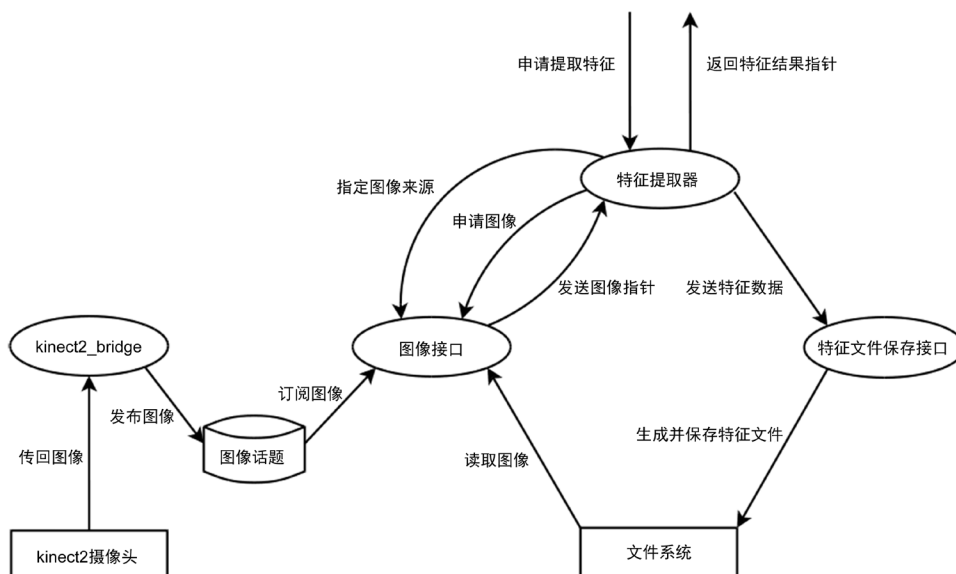


Figure 1. Diagram of functional structure of modules
图 1. 模块功能结构示意图

2.3. 原型的数据结构

2.3.1. 实现

类数据成员的大致结构如下:

1) 必要的数据存储空间和挂载的类结构。如: 图像接口与特征文件保存接口的类指针、图像指针、字符串处理类等。由于图像接口和特征文件保存接口所使用的接口类都是双向的接口类而且原有系统中的调度节点的提取和识别功能并不会同时进行, 所以用指针共用一个实例可以节省内存, 减少接口数量。

2) 结果登记单元与结果参数列表。结果登记单元包括一个像素的特征, 提取范围内的数量, 参考灰度值; 结果参数列表包括一个结果登记单元的 `std::vector` 数组和一组灰度参考值, 以及一个减色比例。

3) 参数列表单元与参数列表队列。参数列表单元包括提取范围, 提取比例, 灰度补偿, 特征的操作参数, 以及备用参数列表的索引; 参数列表队列包括一个外部设置的 `std::vector` 数组, 一个预设的 `std::vector` 数组, 还有一个指针索引用来记录当前使用的参数列表。

4) 其他运行参数。

我们需要注意的是: 参数列表本身是一组必要参数的集合, 这样就导致如果要修改其中一个参数就

需要重新输入一组参数。第一，这导致了修改参数的繁琐与过大的开销；第二，由于这样参数结构对只修改其中一部分参数。

我们主要讨论的问题其一将集中在：参数列表单元过大对局部修改兼容性不足导致参数接口过于复杂以及参数列表队列导致的过量冗余。

2.3.2. 优缺点

这样的数据结构的唯一优点是：这个数据结构支持了整个提取过程把功能实现了。

而它的缺点也极为明显：由于我们在《2.1 基本设计要求》提出了提取总量不变，提取密度随提取范围变化而变化的要求。由于我们采用了最方便的矩形区域作为提取范围的基本形状，因此为了提取某个非矩形的目标区域，使用者可能会选择多次局部提取来防止提取到不符合要求的区域，如下图(图 2 多次提取的示意图)所示。深色部分为需要提取的区域。

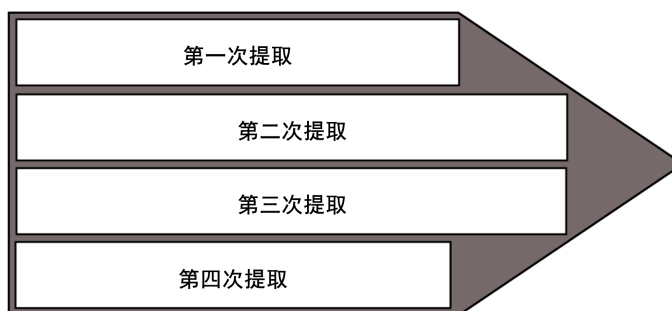


Figure 2. Schematic diagram of multiple extractions

图 2. 多次提取的示意图

这就要求设计里参数列表中各组数据的接口非常灵活且独立。但我们在设计原型的数据结构时将参数列表整体视为了一个单元，如果我们需要有一个参数列表中的某部分的时候，模块会把整个参数列表加载进来。为了在这基础之上制作足够灵活的接口，我们不得不在既有但不兼容的结构之上重新构造另一组更灵活但是功能完全相同的结构来将原来的参数列表的局部屏蔽掉，使用通过接口获得的数据。也就导致了如下图(图 3 原型设计的参数索引结构示意图)所示的参数索引结构，提取过程的每个子过程由一个函数表示，每个函数具有如图 3 所示的结构，通过在进入工序后，先加载参数，然后再执行算法函数。在提取过程中，存在两个子过程需要具有这样的结构：1) 提取特征函数，这个函数在取回图像数据后执行，通过读取参数列表中的提取范围和提取密度进行均匀地(等步距)抽样提取。2) 按照提取比例筛选特征函数，这个函数是对提取到的特征的累计结果降序排序，根据参数列表中设定好的提取比例提取满足条件的一组特征。在如图 2 多次提取的示意图中所示的多次提取过程中，每次提取的范围大小、位置是存在不同的；而如果得到的特征过多或过少，使用者则需要调整提取过程的提取比例，因此从使用角度来讲这样的结构满足了使用要求，但这样随意堆砌的结构让整个模块付出了加载参数部分的结构过于复杂的代价。

2.4. 特征信息提取流程

特征提取流程如下图所示(图 4 特征信息提取流程图所示)。当颜色特征识别系统的调度环节调用了本设计的特征提取模块的处理函数后，特征提取模块开始进行一次提取。提取模块先通过调用图像收发接口类，取回 kinect2 传回的图像数据，然后开始检测运行参数是否需要刷新，如果需要刷新，则对需要刷新的部分重新计算，至此提取过程的需要的数据样本和参数设定准备完毕。接下来根据运行参数中的提

取范围和提取密度两组参数对取回的图像进行指定范围内的均匀的抽样提取，提取到的特征为包含数量的 RGB 颜色特征，在“统计相同特征的数量”以前，所有特征的数量均为 1。此时，模块提取到的样本中存在大量的重复样本，因此需要通过接下来的“将提取结果按色号排序”和“统计相同特征的数量”以产生颜色特征唯一的统计结果。之后再按统计数量降序排序，按提取比例取出符合要求的特征数据，最终生成一个新的特征文件并把提取结果保存至其中。

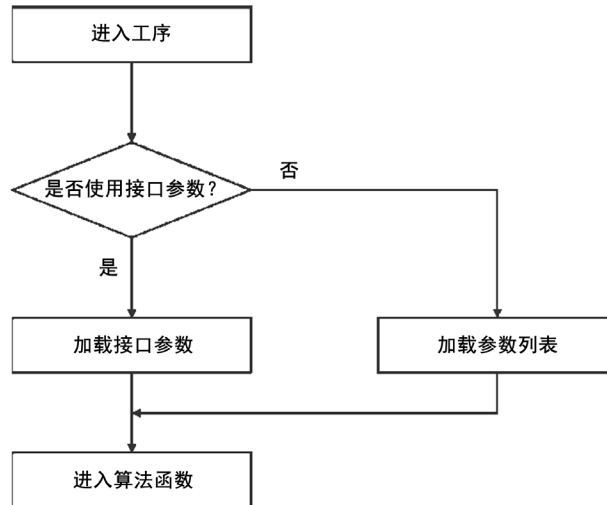


Figure 3. Parametric index schematic of prototype design

图 3. 原型设计的参数索引结构示意图

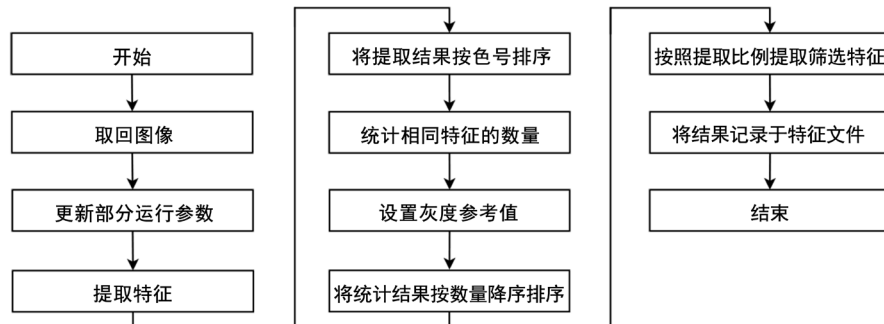


Figure 4. Functional structure diagram of module

图 4. 模块功能结构示意图

由于原型设计并不考虑计算性能，所以这里使用的排序方法均是最常用的“冒泡排序法”。如果考虑计算性能的话，采用可以适用于多线程开发的排序方法更为合适，比如“奇偶排序法”。

在图 4 模块功能结构示意图中的“设置灰度参考值”这一工序放在“统计相同特征的数量”之后的原因在于：不论是采用 for 循环还是采用 std::vector 迭代器的方法去设置灰度参考值，其处理的数据量越少性能越好。在“统计相同特征的数量”之后得到的统计结果的数据量是不会多于“提取特征”之后的未处理结果的数据量的。因此，“设置灰度参考值”放在“统计相同特征的数量”之后更为合适。

2.5. 灰度参考值

2.5.1. 实现

为了克服由于环境光明暗导致的特征值的整体偏移，我们需要在被提取的范围内给出灰度参考值用

以之后的识别系统的校准。为了提供人工识别的接口，这里采用的是一个心理学灰度计算公式，如公式(1)所示。公式(1)放大 100 倍后的结果如公式(2)所示。

$$V_{\text{grey}} = B \times 0.114 + G \times 0.587 + R \times 0.299 \quad (1)$$

$$V_{\text{grey}} = B \times 114 + G \times 587 + R \times 299 \quad (2)$$

由于在计算机中位运算速度为整数乘法运算的四倍，所以我们将公式(2)换算为二进制公式。在计算机中，位运算是一种二元运算符，其符号为： $x \ll y$ ，其数学意义为 $x \times 2^y$ 。如下公式(3)为直接位运算化后的结果。因为在灰度补偿过程中仅仅需要比例关系，所以将常数分母约掉，并用 K_{grey} 替换 V_{grey} 。

$$K_{\text{grey}} = B \ll 7 + G \ll 9 + R \ll 8 \quad (3)$$

公式(2)与公式(3)中各系数进行对比后有如下表所示(表 1 公式(2)与公式(3)的系数对比)。

Table 1. Coefficient comparison between formula (2) and formula (3)

表 1. 公式(2)与公式(3)的系数对比

	总和	B 通道	G 通道	R 通道
公式(2)系数	1000	114	587	299
对应比例	100.00%	11.40%	58.70%	29.90%
公式(3)系数	896	128	512	256
对应比例	100.00%	14.29%	57.14%	28.57%
公式(3)关于公式(2)系数的偏差	0.00%	+2.89%	-1.56%	-1.33%

由于我们希望的偏差平均在 1% 左右，来更好地适配识别补偿算法，所以我们提出了公式(4)来优化偏差。具体做法是将 B 通道值向 G、R 两通道的中值拟合，在数据容量允许之内找到一个比较合适的结果。

$$K_{\text{grey}} = (B \ll 7) \times 55 + (G \ll 15) + (R \ll 14) \quad (4)$$

其偏差如下表(表 2 公式(4)关于公式(2)的系数对比)。

Table 2. Coefficient comparison of formula (4) with respect to formula (2)

表 2. 公式(4)关于公式(2)的系数对比

	总和	B 通道	G 通道	R 通道
公式(4)系数	896	128	512	256
对应比例	100.00%	14.29%	57.14%	28.57%
公式(3)关于公式(2)系数的偏差	0.00%	+1.13%	-0.39%	-0.74%

2.5.2. 局限

由于在《2.5.1 实现》我们选择是对同一类环境做的灰度补偿公式，这就意味着：我们可以在自然光环境做明暗的灰度补偿，而对于由一种色光主导的环境向另一种色光主导的环境的转换是做不到的，因为在归一化 RGB 颜色模型下，不同色光主导的颜色特征其特征因子完全不同，如果要强制转换的话将导致数据转换前后的差异非常之大。

3. 原型提取结果样本

结果的文件结构见图 5 样本文件结构。每一组样本文件包含三个组成部分：1) 提取记录图像。如图 6

提取记录图像细节所示,其黄框内部分(包含红色像素点部分)为被提取部分,红色像素为提取器提取过的像素,在提取后被提取的位置被重置为了红色。2) RGB 三通道谱。这组通道谱只是测试时用来观察提取结果是否覆盖到了提取范围内的主要特征。3) 特征结果文件。这个是提取模块在一次提取后最终结果的存档文件,其在浏览器中打开的效果如图 7 特征文件格式所示,用浏览器打开的有点在于及时 XML 文件没有任何格式符,在浏览器中打开也依旧会保持 XML 原有的树状数据格式。文件中的“src”为第一级目录,代表下级目录的数据来源于资源数据;“color”为第二级目录,代表彩色图像,由于 Kinect2 为体感摄像头,其中不仅有彩色图像,还有代表距离的深度图像,因此在结果文件中对这两种图像的数据加以区分,防止混淆;total 代表特征总数, grey_h、grey_l、grey_a 是灰度参考值, div_bit 代表减色位数。以“spec_1”为例的为第三级目录,是文件的最后一级目录,用来保存颜色特征,其中的 ch_B、ch_G、ch_R 为颜色特征的对应 BGR 三通道值。

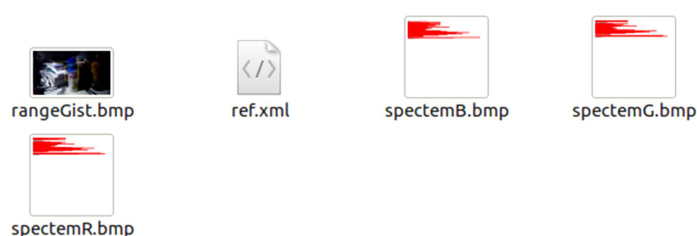


Figure 5. Sample file structure

图 5. 样本文件结构

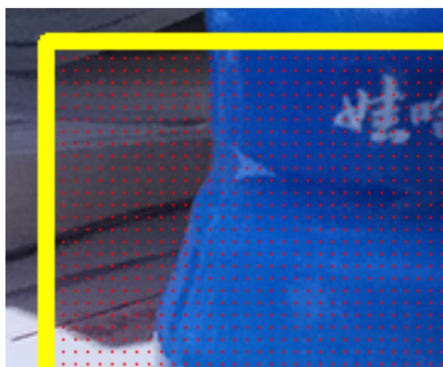


Figure 6. Extraction of recording image details

图 6. 提取记录图像细节

```

- <src>
  - <color total="80" grey_h="211455" grey_l="11775" grey_a="75384" div_bit="2">
    <spec_0 ch_B="236" ch_G="220" ch_R="212"/>
    <spec_1 ch_B="236" ch_G="212" ch_R="204"/>
    <spec_2 ch_B="172" ch_G="148" ch_R="140"/>
    <spec_3 ch_B="188" ch_G="164" ch_R="160"/>
  
```

Figure 7. Format of feature file

图 7. 特征文件格式

4. 总结

本文主要是基于 ROS 环境设计了一款图像特征提取器模块,从工学角度出发,在不优先考虑性能的情况下设计了以实用为基本目的的颜色提取器。根据一般设计规律,本文先给出软件结构,进而进行算

法实现、结果检验等问题的分析，并着重阐述了颜色提取过程和灰度参考值生成的实现和优化。本文所设计的颜色提取器对于实际应用中颜色辨识问题具有很好的实用价值。

基金项目

天津市高等学校科技发展基金资助项目(项目编号: JWK1614)。

参考文献

- [1] 陈倩, 陶彬娇, 等. 主颜色提取算法及其在服装图像检索中的应用[J]. 华南师范大学学报(自然科学版), 2019, 51(1): 111-119.
- [2] 焦圣喜, 刘永权, 冀辰宇. 基于颜色特征提取方法的汽车仪表指针识别研究[J]. 工业控制计算机, 2013, 26(12): 18-19.
- [3] 李莎, 钱建平, 等. 不同颜色模型识别成熟苹果图像的比较分析[J]. 计算机工程与设计, 2015, 36(6): 1545-1550.
- [4] 周建平. 基于 HSV 颜色空间的水下运动目标提取[J]. 电脑知识与技术, 2018, 14(21): 230-232.
- [5] 王克奇, 王业琴, 等. 板材图像识别中颜色特征参数的提取[J]. 东北林业大学学报, 2006, 34(3): 104-105.
- [6] 缪盾. 融合颜色信息的尺度特征提取及匹配算法[J]. 江西科学, 2018, 36(2): 319-323.

知网检索的两种方式:

1. 打开知网首页: <http://cnki.net/>, 点击页面中“外文资源总库 CNKI SCHOLAR”, 跳转至: <http://scholar.cnki.net/new>, 搜索框内直接输入文章标题, 即可查询;
或点击“高级检索”, 下拉列表框选择: [ISSN], 输入期刊 ISSN: 2325-6753, 即可查询。
2. 通过知网首页 <http://cnki.net/> 顶部“旧版入口”进入知网旧版: <http://www.cnki.net/old/>, 左侧选择“国际文献总库”进入, 搜索框直接输入文章标题, 即可查询。

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

期刊邮箱: jisp@hanspub.org