

**毕业设计（论文）**

题 目 基于监控视频的智能灯控系统设计

起讫日期 2019年 03 月 01 日 至 2019年 06 月 01 日

学生姓名 杜羿辰 专业班级 自动化153

所在学院 电气信息工程学院

指导教师 张旭秀 职称 教授

所在单位

年 月 日

摘 要

当今社会科技正在高速发展，各个领域里的科技产品都争相为了利民便民而变得更加自动化和智能化，而现如今灯光控制在各个领域各种环境都会得到极大的应用，不论是生活中校园中各种环境的节日中交通运输中等等各个领域，更加智能化的灯控系统都正在被大范围地利用以更好的为世界带来便利、节能、温馨与我们所需要的气氛。

本文以目前视频监控技术和计算机软件技术为基础，在此基础上利用特定软件编程协助视频监控自动控制灯光系统可以节约人力的优点进行设计。利用Python语言和OpenCv开发平台，开发一套windows视频处理程序，通过OpenCv提供的NumPy库，SciPy库，OpenNI库，SensorKinect库等，运行程序实现对监控视频图像的采集，处理与分析并输出到灯光控制系统以完成无人控制智能灯控系统的设计。

**关键词：**视频监控 OpenCv Python 智能控制

**ABSTRACT**

Nowadays, science and technology are developing at a high speed, and the scientific and technological products in all fields are competing for the convenience of the people and becoming more automatic and intelligent. However, lighting control is widely applied in all kinds of environments, no matter in the festivals in the campus or in the transportation and other fields. The more intelligent lighting control system is being widely used to bring convenience, energy saving, warmth and the atmosphere we need to the world.

This paper is based on the current video monitoring technology and computer software technology, on this basis, using specific software programming to assist video monitoring automatic control lighting system can save the advantages of manpower design. Python language and OpenCv development platform were used to develop a set of Windows video processing program. Through the NumPy library, SciPy library, OpenNI library and SensorKinect library provided by OpenCv, the running program realized the collection of monitoring video images, processing and analysis and output to the light control system to complete the design of the unmanned intelligent light control system.

**Key words：**Video monitoring OpenCv Python Intelligent control

目 录

**[第一章 绪论](#_Toc19581_WPSOffice_Level1)**[1](#_Toc19581_WPSOffice_Level1)

[1.1背景及意义](#_Toc11986_WPSOffice_Level2) [1](#_Toc11986_WPSOffice_Level2)

[1.3研究目标](#_Toc5787_WPSOffice_Level2) [2](#_Toc5787_WPSOffice_Level2)

[1.4研究内容](#_Toc3379_WPSOffice_Level2) [2](#_Toc3379_WPSOffice_Level2)

[1.5研究方法](#_Toc23817_WPSOffice_Level2) [3](#_Toc23817_WPSOffice_Level2)

**[第二章 Python与OpenCV简介](#_Toc11986_WPSOffice_Level1)**[3](#_Toc11986_WPSOffice_Level1)

[2.1 OpenCv应用软件介绍](#_Toc23132_WPSOffice_Level2) [3](#_Toc23132_WPSOffice_Level2)

[2.2 Python软件介绍](#_Toc17616_WPSOffice_Level2) [4](#_Toc17616_WPSOffice_Level2)

[2.2.1 控制语句](#_Toc11986_WPSOffice_Level3) [5](#_Toc11986_WPSOffice_Level3)

[2.2.2 表达式](#_Toc5787_WPSOffice_Level3) [6](#_Toc5787_WPSOffice_Level3)

[2.2.3 数学运算](#_Toc3379_WPSOffice_Level3) [7](#_Toc3379_WPSOffice_Level3)

[2.2.4 函数](#_Toc23817_WPSOffice_Level3) [7](#_Toc23817_WPSOffice_Level3)

[2.2.5 类型](#_Toc23132_WPSOffice_Level3) [8](#_Toc23132_WPSOffice_Level3)

[2.2.6 特点](#_Toc17616_WPSOffice_Level3) [9](#_Toc17616_WPSOffice_Level3)

**[第三章 基于Python语言的动态检测研究](#_Toc5787_WPSOffice_Level1)**[11](#_Toc5787_WPSOffice_Level1)

[4.1 背景分割器](#_Toc9046_WPSOffice_Level2) [11](#_Toc9046_WPSOffice_Level2)

[4.1.1 K-Nearest(KNN)](#_Toc9046_WPSOffice_Level3) [11](#_Toc9046_WPSOffice_Level3)

[4.1.2 Mixture of Gaussians(MOG2)](#_Toc7825_WPSOffice_Level3) [12](#_Toc7825_WPSOffice_Level3)

[4.2 均值漂移和CAMshift](#_Toc7825_WPSOffice_Level2) [12](#_Toc7825_WPSOffice_Level2)

[4.2.1 均值漂移（Meanshift）](#_Toc22315_WPSOffice_Level3) [12](#_Toc22315_WPSOffice_Level3)

[4.2.2 CAMshift](#_Toc8808_WPSOffice_Level3) [13](#_Toc8808_WPSOffice_Level3)

[4.3 卡尔曼滤波器](#_Toc22315_WPSOffice_Level2) [14](#_Toc22315_WPSOffice_Level2)

**[第四章 基于Python语言的图像处理仿真](#_Toc3379_WPSOffice_Level1)**[16](#_Toc3379_WPSOffice_Level1)

[4.1 设计思路与步骤](#_Toc8808_WPSOffice_Level2) [16](#_Toc8808_WPSOffice_Level2)

[4.1.1 图像提取](#_Toc17857_WPSOffice_Level3) [17](#_Toc17857_WPSOffice_Level3)

[4.1.2 灰度图转换与明暗度检测](#_Toc24571_WPSOffice_Level3) [17](#_Toc24571_WPSOffice_Level3)

[4.1.3 动态检测方法介绍](#_Toc2603_WPSOffice_Level3) [19](#_Toc2603_WPSOffice_Level3)

[4.1.4 仿真研究](#_Toc20462_WPSOffice_Level3) [24](#_Toc20462_WPSOffice_Level3)

**[结论](#_Toc23817_WPSOffice_Level1)**[25](#_Toc23817_WPSOffice_Level1)

**[致谢](#_Toc23132_WPSOffice_Level1)**[26](#_Toc23132_WPSOffice_Level1)

**[参考文献](#_Toc17616_WPSOffice_Level1)**[28](#_Toc17616_WPSOffice_Level1)

**[附录](#_Toc9046_WPSOffice_Level1)**[29](#_Toc9046_WPSOffice_Level1)

1. 绪论

1.1背景及意义

当今社会科技正在高速发展，各个领域里的科技产品都争相为了利民便民而变得更加自动化和智能化，而现如今灯光控制在各个领域各种环境都会得到极大的应用，不论是生活中校园中各种环境的节日中交通运输中等等各个领域，更加智能化的灯控系统都正在被大范围地利用以更好的为世界带来便利、节能、温馨与我们所需要的气氛。

电力线网络是智能照明控制的核心。信息传输基于电力线X10协议和电力线载波技术。它可以实现多种先进的控制方式，如电话遥控，无线遥控，电脑控制以及定时控制，再控制亮度。数量和亮度反映了智能场景记忆的特征。

1.2国内外发展情况

由于中国各地供电情况不尽相同致使供电电压存在一定程度的差异，因此在设计各类灯具时，为了使其启动与发光能自适应这一差异，其设计电压一般低于标准线电压380V；往往通过提高输送电压的方式来提高电力系统电能输送的效率，如此操作可能会导致照明灯具实际工作电压偏高。超出额定值的电压会带来三大负面影响：浪费电能与缩短灯具寿命且降低灯具使用效率。

今天，随着全球资源日益稀缺，城市照明仍然占据了大部分资源消耗。这是一个严重的环境问题。为了解决这个问题，智能照明控制系统的研究和开发已经提上议事日程以确保照明。在需求的前提下节约能源和环境保护是一个需要实现的目标。越来越多的城市正在采用节能灯更换来解决当前的问题，但随着城市规模的不断扩大，灯具的使用率不断提高，这是不可能解决的问题。智能控制系统的出现可以解决这个问题，而更多节能灯的组合可以达到第二个节能效果。智能照明控制系统可根据时间或要求控制亮度。还可以实现单灯和多灯控制灯，进一步节省能源。由于LED本身是一种环保产品，它可以保证它不会污染环境。通过数据采集统计和比较，智能照明控制系统可以实现大规模的节电。

智能灯光的控制为了能够更加方便地与各种其他相关控制和管理系统的平台实现扩展与联接，故而都是通过软件控制实现我们所需要的功能的：开关、调光、就地控制、区域控制等。故而使用时通过整套较为简易的操作就可以实现对灯光的随意操作与场景切换。达到了无需专业灯光控制人员操作，降低资源利用率，节省成本，多点、分布式控制，便于管理等目的。

智能照明系统能够通过软件控制照明系统，只需要将原来的机械开关更换为智能开关，设计简单，不用复杂的施工。另外，由于它是计算机软件控制，它不仅可以实现像传统控制系统一样的切换功能，还可以实现调光功能，时间等。通过控制系统和管理系统的结合，方便人们使用，没有浪费精力，实现资源的双赢。

1.3研究目标

本课题的研究目标，由于校园对灯光的需求比较大，为了能够更高效的利用灯光系统，我们将条件设置为，当光照充足时，灯光控制系统不启动，当光照不足时，灯光控制系统取决于视频中是否有人，若有人则启动，无人则不启动，这样能够高效的利用电能，人来灯亮人走灯灭，节约能源，增加灯具的使用寿命。

1.4研究内容

主题为基于视频监控的灯光智能控制系统设计，利用OpenCV视频处理软件，处理分析视频监控器拍摄到的视频图，根据设定的不同条件输出不同的结果到最终的灯光控制系统，以达到光线不足人来灯亮人走灯熄的目的。

1.5研究方法

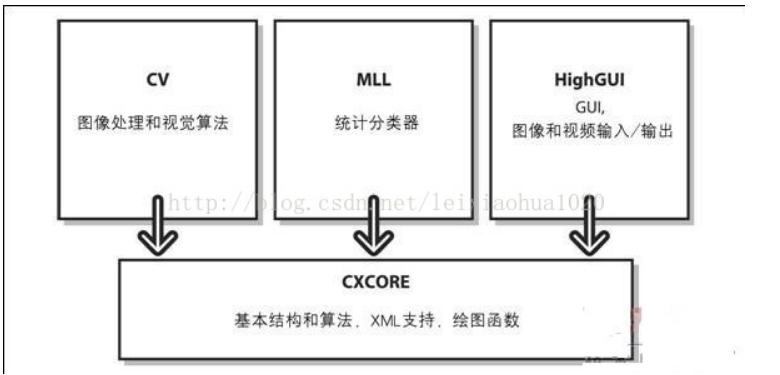
首先，对数字图像处理进行研究，掌握数字图像处理的常用软件，其次对传统的智能控制技术进行了解，并设计照明灯控制系统，同时了解基于视频的运动目标检测和跟踪技术。在灯控系统的基础上增加基于监控视频所获取的信息，提高控制系统的自主性和高效性。

第二章 Python与OpenCV简介

本设计针对现有的视频监控系统设计一个监控视频图像分析的灯光控制系统，该软件应包括视频文件提取，包含调用、读取、采集视频文件与调用摄像头。视频文件中光照强度检测与分析，检测结果既光照强度的强与弱作为输出到灯光控制器中的第一要素。视频中运动物体的检测与反馈，检测结果既视频中是否有人作为输出到灯光控制器的第二要素。两种因素根据其优先级共同决定控制器最终的控制结果。综合以上需求设计程序应包含图像采集处理程序，光强检测程序，动态监测程序及最后的控制程序。下面对程序设计编程软件及原理进行介绍。

2.1 OpenCv应用软件介绍

OpenCV（开源计算机视觉库）是一个基于BSD许可证（开源）的跨平台计算机视觉库，可在最常见的操作系统上运行，例如Linux，Windows，Android和Mac OS。 它简单而有效 - 结构由C函数主导，辅以少量C ++，并与许多语言（如Python，Ruby和MATLAB）接口，以实现广泛的常见图像处理和计算机算法，被应用于各种图像和视频处理工作，具有良好的可移植性，通过OpenCv能很容易地实现一些有前景且功能现今的应用。

OpenCV主要的应用领域非常的广泛，包括：人机互动功能，物体鉴别，人脸检测跟踪，运动轨迹分析与跟踪，计算机视觉等。其主体分为5个模块：

2.2 Python软件介绍

Python的思维基础理念是“便捷”、“明确”、“简单”。Python开发者的理念是“用且尽可能只用一种方法来作为事件的解决手段”。编写Python代码时，若有几种不同的方向，Python编写者大多数情况不会考虑繁琐复杂的语言及语法，而选择简洁明了的且基本没有有歧义的语言及语法。出于设计理念存在极大的不同情况下，Python源代码具有相对于其他同类软件更好的可读性，其功能的强大足以支撑大规模的软件开发，叫做Python格言。在Python[解释器](https://baike.baidu.com/item/%E8%A7%A3%E9%87%8A%E5%99%A8" \t "https://baike.baidu.com/item/Python/_blank)内运行import this可以获得完整的列表。

Python开发人员经常试图避免不必要的操作和优化。对于世界上不重要的部分，速度更快的补丁通常不会集成到Python中。很多人认为Python很慢。然而，根据80/20规则，大多数程序不需要太多的速度。在某些情况下，速度是至关重要的，Python设计人员倾向于使用JIT技术，或者用C/ c++重写程序的这一部分。可用的JIT技术是PyPy。

Python是一种完全面向对象（包括函数、模块、数字和字符串等）的语言。并完全支持继承、重载、派生、多重继承，并有助于增强源代码重用。Python支持重载操作符和动态类型。与传统函数式编程语言Lisp相比，Python对函数式设计的支持有限。两个标准库(functools、itertools)为Haskell和标准ML提供了经过验证的函数式编程工具。

虽然Python可以被松散地归类为“脚本语言”，但它实际上被谷歌广泛用于大型软件开发项目，如Zope、Mnet和BitTorrent。Python的支持者更喜欢将其称为高级动态编程语言，因为“脚本语言”通常指只执行简单编程任务的语言，如shell脚本、VBScript和其他只能处理简单任务，而不能与Python一起使用的编程语言比较的。

设计人员将Python设计为可扩展的。并非所有特性和功能都集成到语言核心中。 Python提供了许多api和工具，使程序员可以轻松地在C，C ++和Cython中编写扩展。 Python编译器本身也可以集成到需要脚本语言的程序中。 因此，许多人也使用Python作为“粘合语言”。使用Python集成和打包用其他语言编写的程序。Google中的许多项目（如Google Engine）都在c ++中编写了最需要的部分，然后在Python或Java / Go中调用相应的模块。 “这很难说，”Python技术手册的作者Alex Martelli说。 但到了2004年，Python已进入谷歌。 谷歌招募了很多Python，但在他们决定使用Python之前，他们的目标是Python。 使用c ++控件硬件时，我们不得不使用Python进行快速开发。

2.2.1 控制语句

if语句，当条件成立时运行语句块。经常与else, elif(相当于else if) 配合使用。

for语句，遍历列表、字符串、字典、集合等[迭代器](https://baike.baidu.com/item/%E8%BF%AD%E4%BB%A3%E5%99%A8" \t "https://baike.baidu.com/item/Python/_blank)，依次处理迭代器中的每个元素。

while语句，当条件为真时，循环运行语句块。

try语句，与except,finally配合使用处理在程序运行中出现的异常情况。

class语句，用于定义[类型](https://baike.baidu.com/item/%E7%B1%BB%E5%9E%8B/6737759" \t "https://baike.baidu.com/item/Python/_blank)。

def语句，用于定义函数和类型的方法。

pass语句，表示此行为空，不运行任何操作。

assert语句，用于测试在程序调试阶段是否满足运行条件。

with语句(在Python 2.6后中定义的语法)，在场景中运行语句块。例如，在运行语句块之前加密它，然后在语句块运行完之后解密它。

yield语句，在迭代器函数中使用的返回一个元素。自Python 2.5发布以来。这个语句变成一个运算符。

raise语句，制造一个错误。

import语句，导入一个模块或包。

from import语句，从包导入模块或从模块导入某个对象。

import as语句，将导入的对象赋值给一个变量。

in语句，判断一个对象是否在一个字符串/列表/元组里。

2.2.2 表达式

主要的算术运算符与C/C++类似。+, -, \*, /, //, \*\*, ~, %分别表示加法或者取正、减法或者取负、乘法、除法、整除、乘方、取补、取余。>>, <<表示右移和左移。&, |, ^表示二进制的AND, OR, XOR运算。>, <, ==, !=, <=, >=用于比较两个表达式的值，分别表示大于、小于、等于、不等于、小于等于、大于等于。在这些运算符里面，~, |, ^, &, <<, >>必须应用于整数。

Python使用and, or, not表示逻辑运算。

is, is not用于比较两个变量是否是同一个对象。in, not in用于判断一个对象是否属于另外一个对象。

Python支持“列表推导式”（list comprehension），比如计算0-9的平方和:

|  |  |
| --- | --- |
| 1  2 | >>> sum(x **\*** x **for** x **in** range(10))  285 |

Python使用lambda表示匿名函数。匿名函数体只能是表达式。比如：

|  |  |
| --- | --- |
| 1  2  3 | >>> add**=lambda** x, y : x **+** y  >>> add(3,2)  5 |

Python使用y if cond else x表示条件表达式。这意味着当cond为真时表达式的值为y，当cond为真时表达式的值为x。Java中的c++和Cond Y: x

Python区分列表和元组。列表是1 2 3，元组是1 2 3。您可以更改列表中的元素，而不是元组。在某些情况下，可以省略元组上的括号。元组对赋值语句有特殊的处理。因此，你可以同时分配多个变量，例如:

|  |  |
| --- | --- |
| 1 | >>> x, y**=**1,2 # 同时给x,y赋值，最终结果：x=1, y=2 |

特别地，可以使用以下这种形式来交换两个变量的值：

|  |  |
| --- | --- |
| 1 | >>> x, y**=**y, x #最终结果：y=1, x=2 |

Python使用单引号和双引号来表示字符串。 与Perl，Unix Shell，Ruby或Groovy不同，这两个符号用于相同的目的。 通常，如果双引号出现在字符串中，则单引号用于表示字符串; 如果没有，则使用双引号。 如果没有，请选择您喜欢的字符串中出现的\（反斜杠），并将其解释为特殊字符，例如\换行。 将r添加到表达式意味着Python不会解释字符串中出现的含义。 这通常用于编写正则表达式或Windows文件路径。

Python支持列表切割(list slices)，可以取得完整列表的一部分。支持切割操作的类型有str, bytes, list, tuple等。它的语法是...[left:right]或者...[left:right:stride]。假定nums变量的值是[1, 3, 5, 7, 8, 13, 20]，那么下面几个语句为真：

nums[2:5] == [5, 7, 8] 从下标为2的元素切割到下标为5的元素，但不包含下标为5的元素。

nums[1:] == [3, 5, 7, 8, 13, 20] 切割到最后一个元素。

nums[:-3] == [1, 3, 5, 7] 从最开始的元素一直切割到倒数第3个元素。

nums[:] == [1, 3, 5, 7, 8, 13, 20] 返回所有元素。改变新的列表不会影响到nums。

nums[1:5:2] == [3, 7] 从下标为1的元素切割到下标为5的元素但不包含下标为5的元素，且步长为2。

2.2.3 数学运算

Python使用与C、Java类似的运算符，支持整数与浮点数的数学运算。同时还支持复数运算与无穷位数（实际受限于计算机的能力）的整数运算。除了求绝对值函数abs()外，大多数数学函数处于math和cmath模块内。前者用于实数运算，而后者用于复数运算。使用时需要先导入它们，比如：

>>> import math

>>> print(math.sin(math.pi/2))

1.0

fractions模块用于支持分数运算；decimal模块用于支持高精度的浮点数运算。

Python定义求余运行a % b的值处于开区间[0, b)内，如果b是负数，开区间变为(b, 0]。这是一个很常见的定义方式。不过其实它依赖于整除的定义。为了让方程式：b \* (a // b) + a % b = a恒真，整除运行需要向负无穷小方向取值。比如7 // 3的结果是2，而(-7) // 3的结果却是-3。这个算法与其它很多编程语言不一样，需要注意，它们的整除运算会向0的方向取值。

Python允许像数学的常用写法那样连着写两个比较运行符。比如a < b < c与a < b and b < c等价。C++的结果与Python不一样，首先它会先计算a < b，根据两者的大小获得0或者1两个值之一，然后再与c进行比较。

2.2.4 函数

Python的函数支持递归、默认参数值、可变参数，但不支持函数重载。为了增强代码的可读性，可以在函数后书写“文档字符串”(Documentation Strings，或者简称docstrings)，用于解释函数的作用、参数的类型与意义、返回值类型与取值范围等。可以使用内置函数help()打印出函数的使用帮助。比如：

>>> def randint(a, b):

... "Return random integer in range [a, b], including both end points."...

>>> help(randint)

Help on function randint in module \_\_main\_\_:

randint(a, b)

Return random integer inrange[a, b], including both end points.

2.2.5 类型

Python使用动态系统。在编译时，Python不检查对象是否具有被调用的方法或属性，但是直到运行时才检查。因此，操作对象时可能会显示异常。但是，尽管Python使用动态类型系统，但它也是强类型的。Python禁止未明确定义的操作，例如数字和字符串。

就像很多面向对象语言，Python允许程序员定义类型。 构造一个对象只需要调用该类型就像它是一个函数一样。例如，对于之前定义的Fish类型，请使用Fish（）。类型本身也是一种特殊类型的对象（类型本身也是一个类型对象），这种设计可以使得对类型进行反射编程。

Python包括多种数据类型。相比Java和C ++，这些数据类型能够减少代码的长度。以下列表简要描述了对于Python 3.x来说，Python内置的数据类型：



除了各种数据类型，Python语言还用类型来表示函数、模块、类型本身、对象的方法、编译后的Python代码、运行时信息等等。因此，Python具备很强的动态性。

2.2.6 特点

优点：

简单性：Python是一种表达简单想法的语言。阅读一个好的Python程序就像阅读英语。它允许您专注于解决问题而不是理解语言本身。

易于学习：Python非常易于使用，因为Python有非常简单的文档[6]。

快速：底层Python是用C语言编写的。许多标准和第三方库也是用C语言编写的，速度非常快。 [3]

免费，开源：Python是FLOSS之一。用户可以自由地分发软件的副本，阅读其源代码，修改它，并将其中的一些用于新的免费软件。 FLOSS基于与团队分享知识的概念。

高级语言：在Python中编写程序时，您不必担心低级细节，例如如何管理程序使用的内存。

可移植性：由于其开源特性（已被修改为在不同平台上工作），Python已被移植到许多平台。这些平台包括Linux，Windows，FreeBSD，Macintosh，Solaris，OS / 2，Amiga，AROS，AS / 400，BeOS，OS / 390，z / OS，Palm OS，QNX，VMS，Psion，Acom RISC OS，VxWorks，基于Android的PlayStation，Sharp Zaurus，Windows CE，PocketPC，Symbian和Google Android平台。

解释性：C和C++是常见的编程语言，两者都可以用来编写相关程序，编写完成之后经过保存，借助编译器可以完成转换，应用在用户使用的计算机之中。

一般的程序在编写完成之后存在硬盘内的文件夹中，运行的时候要用到连接器，它的作用是进行程序的复制转移，把程序导入内存之中。Python这种语言运行起来比较简单，不需要经过一般的编译步骤，用户可以直接使用最初编写的代码，从而完成对程序的运行。

上一段提到Python这种语言使用起来简单方便，具体的来讲主要在于这种语言有相应的解释器，它能起到转换源代码的作用，先把最初的代码变成字节码，字节码再经过翻译成为计算机可以识别的语言形式，从而实现计算机对程序的运行。

[面向对象](https://baike.baidu.com/item/%E9%9D%A2%E5%90%91%E5%AF%B9%E8%B1%A1" \t "https://baike.baidu.com/item/Python/_blank)：我们知道编程可以分为面向过程以及面向对象的编程，这两者是常见的编程方式。在前一种方式中，经常用到过程以及函数来设计程序，而在后一种编程方式中，经常用到数据或者功能来设计程序。Python这种语言的优势之一就在于这两种编程方式都可以使用Python语言。

可扩展性：在一些程序设计之中，可能根据我们的实际需要要对编程方式做一些调整，比如说要加快代码的运行速度，有些代码我们需要保密等等，这种情况下我们就可以把这部分的程序用C编写，然后在Python程序中使用它们。

可嵌入性：Python这种语言有着很强的兼容性，它能够和C或者C++完美契合，实际应用中，我们可以在C中导入Python，赋予程序一些其他的功能。

丰富的库：在Python这种语言中库的种类非常多，常见的有标准库、图像库等等。每种库都有自己的作用，其中最重要的就是标准库了，Python语言的标准库功能很强大，一些编程中需要的工作都可以由它完成，比如像文档生成，电子邮件以及网页浏览器等等。

代码的规范性：Python语言编写出来的代码在排列上整齐规律，方便编程人员查阅，同时前文提到代码需求转化。

### 缺点：

单行语句和[命令行](https://baike.baidu.com/item/%E5%91%BD%E4%BB%A4%E8%A1%8C" \t "https://baike.baidu.com/item/Python/_blank)输出问题：程序在通常情况下不能在编写在同一行中，要分行进行，而其他语言在这点上就没有过多限制。

独特的语法：代码书写时语法的特殊性决定了这种语言入门起来比较困难，辨别语句与语句之间的关系时采用的是缩进方法，关系辨别比其他语言有难度，这使得很多刚接触到这门语言的用户很多时候看不懂代码，给编程带来困难。

运行速度慢：这里是指与C和C++相比。

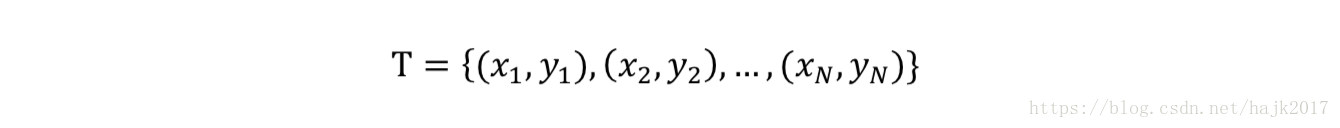
第三章 基于Python语言的动态检测技术

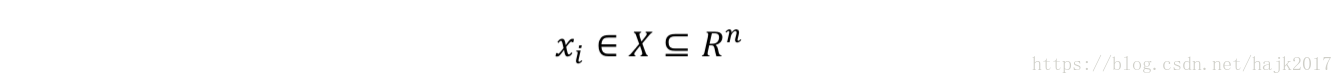
4.1 背景分割器

4.1.1 K-Nearest(KNN)

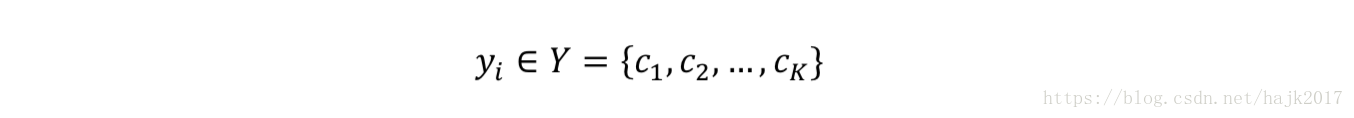
KNN分类算法的分类预测过程简单易懂：在实际中如果我们要对变量x进行预测，就在x周围选取k个集合，这些集合中包含的是最靠近x的向量，那么预测的结果就是x是k个样本中类别数最多的那一类。

输入：训练数据集





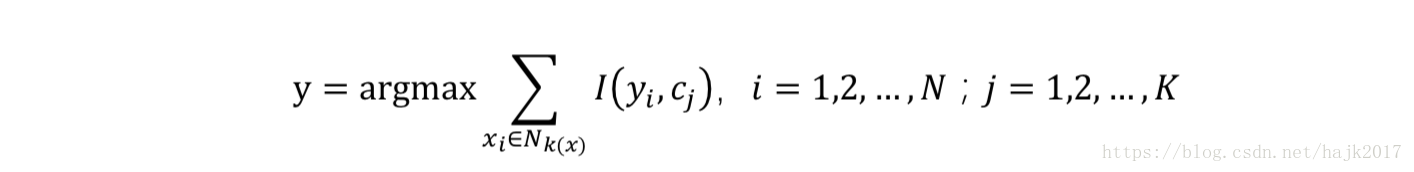
为N维的实例特征向量

 输出：预测实例X所属类别Y

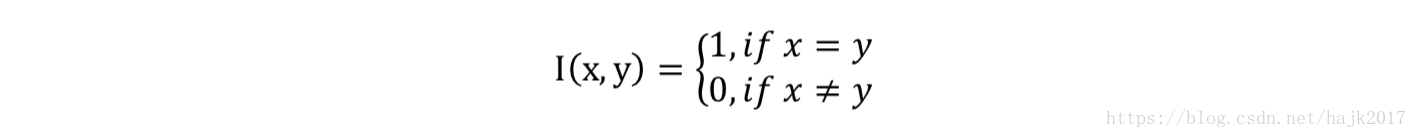
算法步骤：

1. 选取出k个样本点，这些样本点与x之间的距离最小，把这些样本放进一个集合，记这个集合为N\_k(x)；

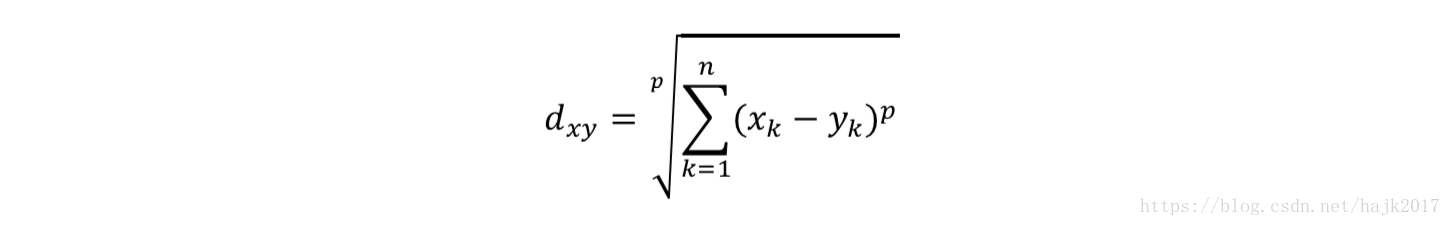
2. 下面的式子是对x进行预测分类的基本原则，多采用多数投票的方式来进行：



上式中I为指示函数：



闵可夫斯基距离本身不是一种距离，而是一类距离的定义。在一个多维空间中，假设这个空间的维度是n，空间中有两个点x和y，两点的坐标分别为(x1,x2,…,xn)以及(y1,y2,…,yn)，那么要计算两点之间的闵可夫斯基距离，就会用到下面的公式：



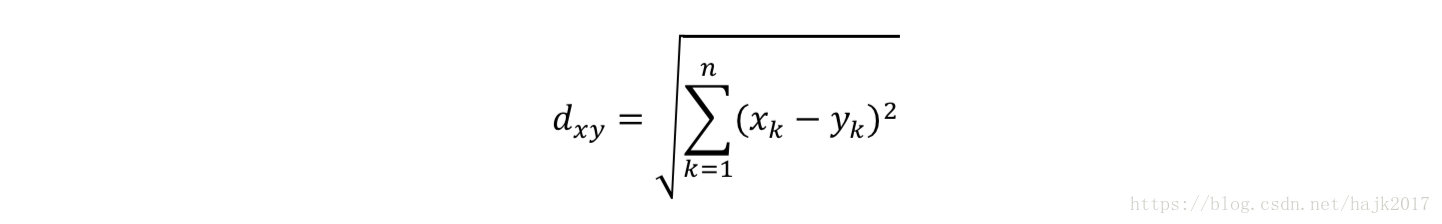
其中，p是一个可变参数：

当p=1时，被称为曼哈顿距离；

当p=2时，被称为欧氏距离；

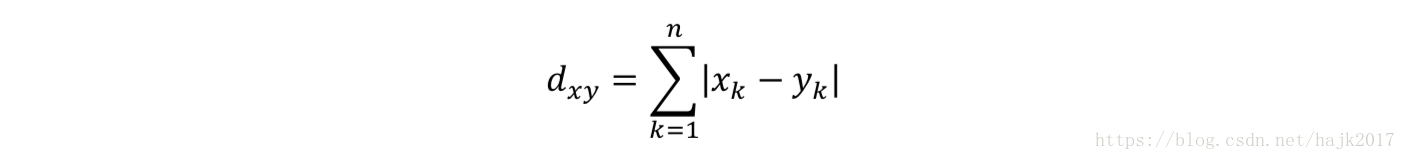


欧氏距离可写作：



欧氏距离的计算理解起来非常简单，我们可以根据欧式空间点之间的距离导出，目前欧式距离在实际中应用的比较广泛。

通过上面的有关定义和公式，可以计算出两点之间的曼哈顿距离：



KNN的局限性

KNN有着非常明显的优点和缺点：

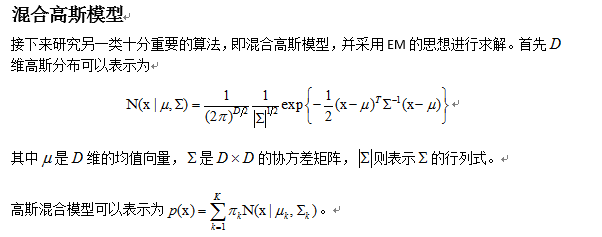
优点：预测结果可信度比较高，外界因素对整个过程造成的影响不大，预测分类步骤较少。

缺点：计算比较复杂。

适用数据范围：数值型和标称型

4.1.2 Mixture of Gaussians(MOG2)

在MOG方法的基础上经过改进和完善，推出了MOG2法，即为高斯混合模型分离算法。



4.2 均值漂移和CAMshift

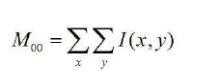
4.2.1 均值漂移（Meanshift）

思想：利用概率密度的梯度爬升寻找局部最优。

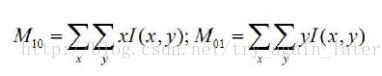
算法过程：

（1）搜索窗的选取；

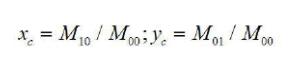
（2）计算零阶距：



计算一阶距：



计算搜索窗的质心



调整搜索窗大小：宽度为

20171124150126356

在上述过程中存在一个固定的临界值，这个值可以控制上面的过程。计算把搜索窗的中心移动到质心经过的距离，若距离大于临界值，重复上面的（2）（3）（4）三步；若它们之间的距离小于这个临界值，这时就停止计算。如果经过循环，距离始终大于这一临界值，那么循环将会在设定的最大次数时停止。

4.2.2 CAMshift

这种定位方法主要是利用了目标的颜色来实现的。我们把物体的颜色分布直方图转化为概率图，在概率图中可以看出颜色的分布情况，然后把搜索窗初始化，让其恢复到默认的大小和位置，每一帧图像中搜索窗的位置都会由上一帧所确定，从而定位出当前图像的目标中心。

在用这种方法来跟踪目标时，对当前输入的图像进行计算，上一帧图像经过计算得到的结果作为下一帧计算时的初始输入数据，不停地进行这个过程，目标就会被追踪。

算法过程为：

(1).初始化搜索窗

(2).把颜色分布直方图转化为颜色分布概率图,也就是反向投影的过程

(3).借助meanshift进行计算，不断更新搜索窗的大小和位置。

(4).（3）中的结果输入下一帧图像的计算中，完了之后在进行步骤（2）。反向投影：

（1）经过研究发现光照条件发生轻微的改变之后，RGB颜色空间就会发生很大的变化，从而对追踪带来困难，所以我们需要进行相关的转换，得到HSV空间。

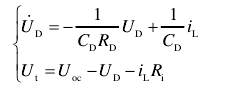
（2）做出颜色分布直方图。

（3）对由（2）得来的颜色分布直方图进行相关的变换，用概率替代直方图中的具体值，得到一个灰度图像，这个图像就是颜色分布概率图。

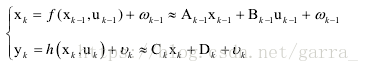
4.3 卡尔曼滤波器

在传统的卡尔曼模式的基础上进行了扩展和创新，推出了卡尔曼滤波算法，这种算法种类比较多，常见的有自适应、线性以及扩展卡尔曼等等。在这些种类中。每一种都有适用于特定的条件之下，比如线性卡尔曼一般情况下就只用于线性系统之中，其他的不满足要求的系统要经过处理，变成符合条件的系统然后在进行相关一些操作。在遇到一些需要展开的方程时，就要用到扩展卡尔曼来进行处理。

建立锂离子电池的一阶RC等效电路模型（即Thevenin模型），下面给出了它的状态方程以及观测方式：



对上述方程进行线性离散化：



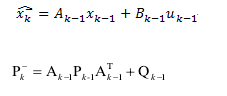
20180627085338526

这种计算方法的基本原理是：在k-1时刻输入i、矩阵A、矩阵B，利用这三个参数来对k时刻的一些相关数据记性估计，主要包括先验估计值以及协方差矩阵。随后根据公式对新息误差以及卡尔曼增益进行计算，最后就可以利用前几步中的结论对k时刻进行估计，不断的重复找到一个最合理的值。

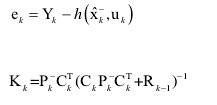
计算步骤如下

（1）初始化 x0，P0，Q0，R0

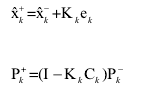
（2）利用k-1时刻进行先验估计：



（3）根据（2）中计算结果对新息和卡尔曼增益进行计算：



（4）由上面的结果，修正估计值：



1. 过程（1）到（4）是在估计过程中从k-1到k时刻系统内部进行的相关操作，这个过程不停地进行，最后经过比较寻优找到最合理的估计值。

我们用上面的方法来对电池的SOC进行估算理论上是可行的，但是在实际应用中还会遇到各种各样的难题：

1. 一般情况下，我们通过构建相关的电池模型来确定非线性电池系统中的一些参数，比如说电池系统的系统矩阵A,B,C,D，这些参数很大程度上会决定最后的估计值，所以要保证这些参数的准确性。现阶段为了解决这一问题常借助线性最小二乘法来进行，最小二乘法的相关特点就决定了估计值存在的一些问题：数据之间的发散程度比较大、估计值与实际值偏离过多、参数失去物理意义等等。 （2）在卡尔曼算法中，Q，R的值会对最终的估计结果造成影响，所以需要在最初设定Q和R时选定一个合理的值，但是实际中操作有一定的困难，因为实际条件与这种算法计算时的理论条件有比较大的出入，实际中观测噪声其实是会影响到测量噪声的。

（3）把这种算法应用到实际中，需要在把它导入单机片之中，在这一过程中可能会影响到上面的各个式子，最终影响对参数的估计。

第四章 基于Python语言的图像处理仿真

4.1 设计思路与步骤

首先将本次设计过程分解为几个部分：图像提取部分、图像灰度图转化与亮度判断分析部分、图像中动态分析部分与输出。整个程序运行过程如下：

结束

有

无

暗

亮

输出0

输出1

图像中的动态检测

开始

灰度图转换与明暗亮度检测判断

图像读取与图片截取

4.1.1 图像提取

利用软件编写程序，设置好视频相关路径，将准备好的相关监控视频读入，并设定间隔25帧从视频中截取一次图片，将视频截取为多个图片并存入指定路径等待进一步处理，相关程序见附录。

4.1.2 灰度图转换与明暗度检测

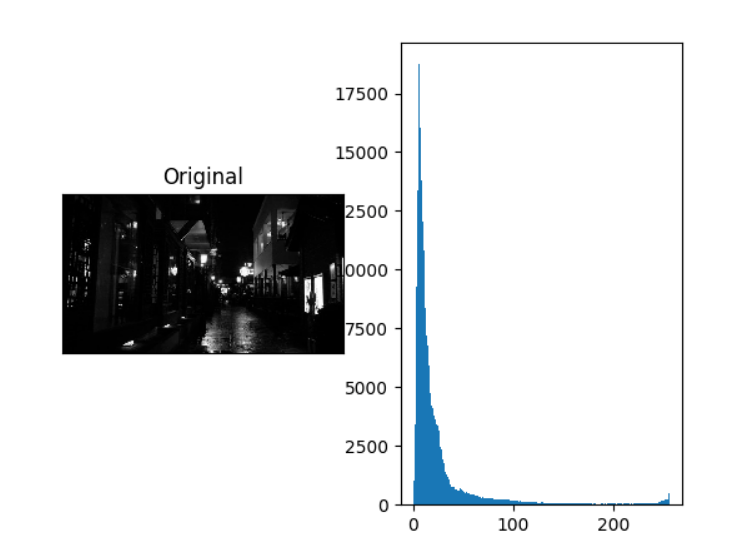
将上一步截取的图片打包提取并将其转换为灰度图，通过灰度直方图判断明暗度，举例如下：

图1 较暗光照条件及灰度直方图

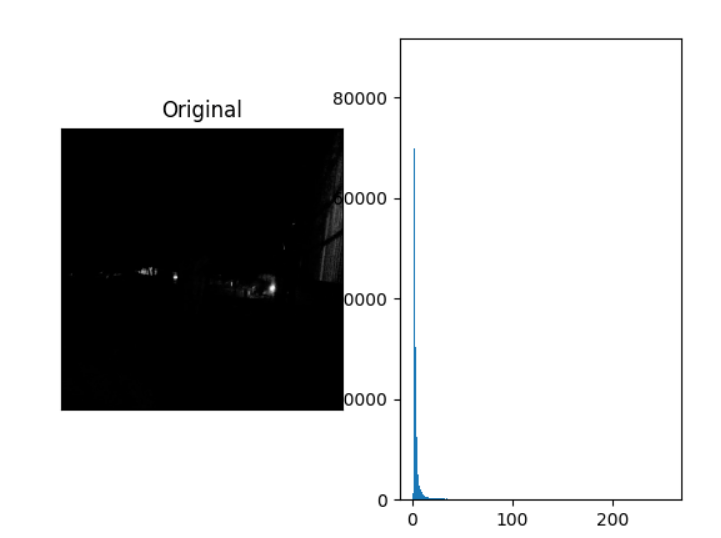


图2 极暗光照条件及灰度直方图

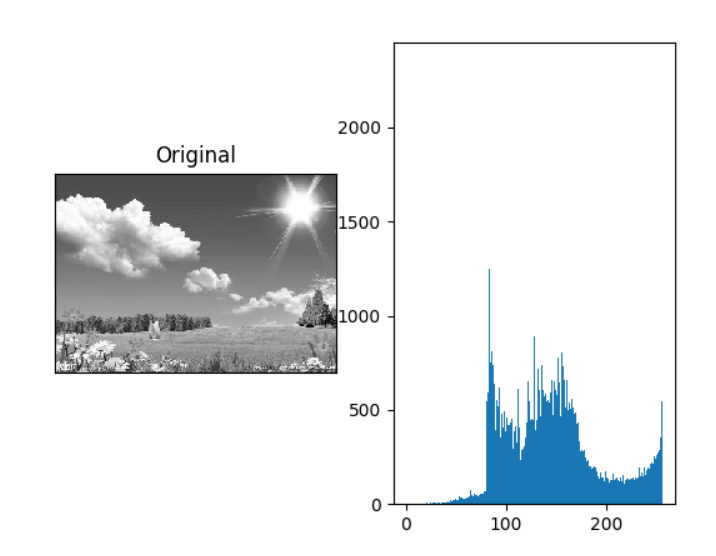


图3 明亮光照条件及灰度直方图

如上3幅图，图1为较暗环境的灰度图与灰度直方图，图2为极暗环境的灰度图与灰度直方图，图3为较亮环境的灰度图与灰度直方图，判断图片的亮暗,只需要统计偏暗的像素个数,用像素个数比上总个数，结果用p表示，至于p大于多少即判断为暗,则可以由自己设置，灰度转换与明暗检测分析判断程序见附录。检测完若光照强度为强则直接输出0，若光照强度为弱则进行下一步——动态检测。

4.1.3 动态检测方法介绍

目标跟踪:为了跟踪视频中的所有目标，首先要完成的任务是识别视频帧中那些可能包含移动目标的区域。

有很多实现视频目标跟踪的方法，根据需要跟踪的对象不同可以有很多跟踪方法。如果我们要对图像中所有移动的目标进行跟踪时，就要重视帧与帧之间的差异，要根据实际情况选择不同的方法来进行跟踪，达成我们的目的。

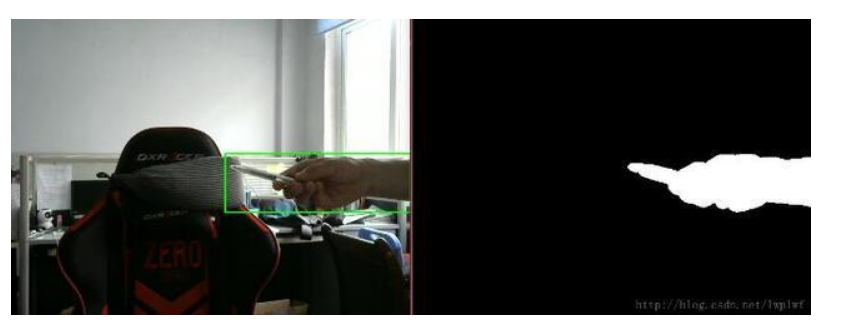
（1）基本运动检测

打开摄像头获取图像，选取最开始的一帧当做背景。以后读取的每一帧图像都会用来跟背景作比较，计算它们之间的差异。在开始之前，需要对帧进行预处理，首先要做的就是将帧转换为灰阶，并进行模糊处理。

在完成对帧的灰度转换和平滑后，就可以计算与背景帧的差异（背景帧也需要进行灰度转换和平滑），用一个图来表示，这种图叫差分图。借助临界值构造一个黑白图像，完成对其的膨胀过程。

侵蚀和膨胀也可用作噪声滤波器，这和上面采用的模糊技术很像。侵蚀和膨胀可以通过调用cv2.morphologyEx 函数来获得，为了达到透明的目的，这两部都需要进行。之后剩下要做的就是在计算出的差分图里面进行搜索，提取并显示出白色的斑点，也可以根据临界值对一些矩形区域进行筛选，把大于这一值的矩形显示出来。这一过程中对外界条件的要求不是过分严格，一些性能比较好的摄像头可以不设置阈值。

该技术最明显的问题是：整个过程中要基于最开始选取的背景，若是环境条件不稳定的话，会影响最后的结果，比如光强波动大，这种方法就显得相当不灵活。举例如下：



（2）背景分割器

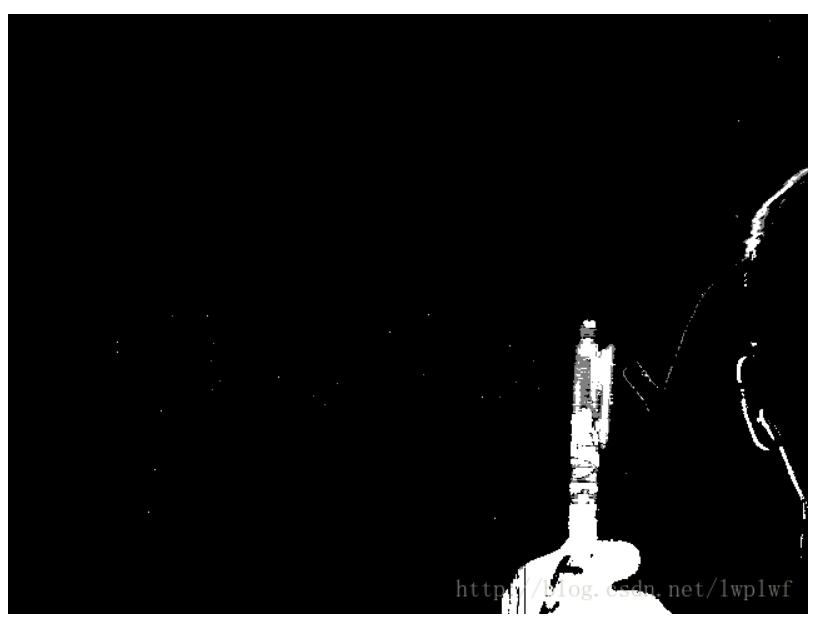
OpenCV常用来进行背景的分割，用它来分割背景操作简便，分割效果好，OpenCV除了常用的分割背景的用途之外还有其他一些重要的作用，比如说应用在检测背景的过程中，提高检测的效率，还可以起到保存文件的作用。

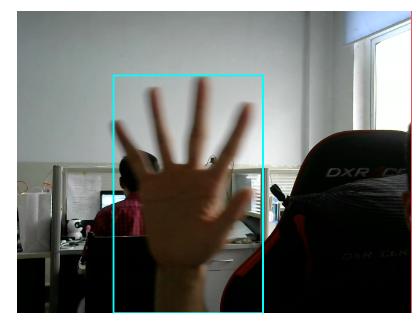
常见的三种分割器是：以及

Geometric Multigid(GMG),他们对应的算法用来计算背景分割。

BackgroundSubtractor 的基本特征：一是其专门用于视频分析，对视频的每一帧进行逐一分析。二是可以用于对阴影的计算，计算出图像中的阴影，这对视频分析的精确性影响很大，消除阴影后我们可以着重分析实际从而很少发生“合并”不需要对象的情况。

由于分割器的精确性和阴影检测能力，即使相邻对象没有在一起进行检测，运动检测的结果也是相当准确的。举例如下：







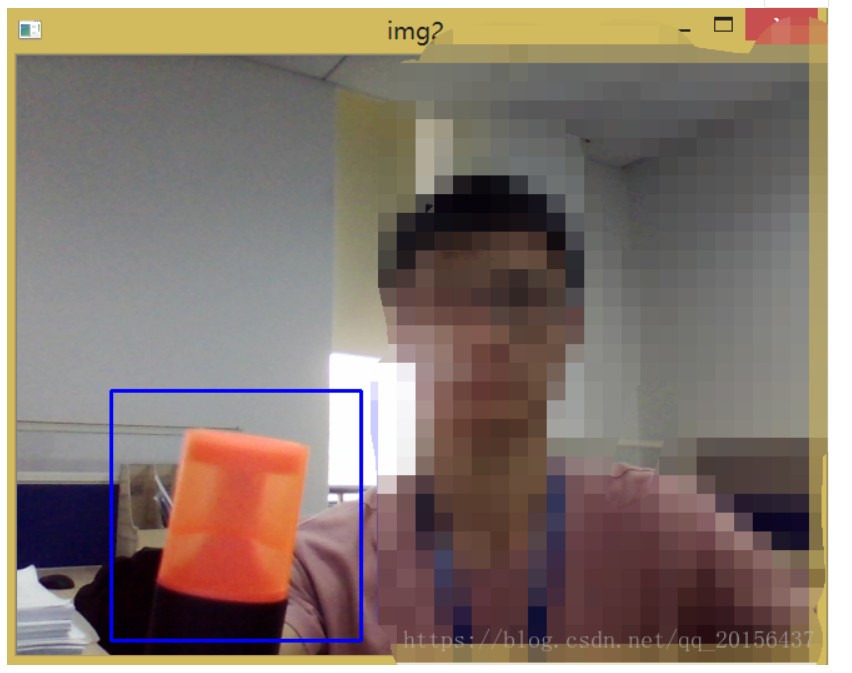
（3）均值漂移和CAMshift

均值漂移主要的原理是求出样本最大密度，图像每换一帧就重新进行一次计算，从而实现对目标的跟踪。均值漂移算法中，目标的移动方向是固定的，计算过程的不断进行，知道与原始中心匹配，或者在连续迭代计算后中心保持不变，这一最后匹配称为收敛。

出了理论之外，均值漂移在跟踪视频中需求区域时非常有用，例如如果预先不知道所要跟踪的区域，就必须采用这种方法，开发程序设定条件，使应用能动态地开始跟踪（和停止跟踪）视频的某些区域。

在计算机上运行均值漂移指令代码，会观察到均值漂移窗口是怎样搜索指定颜色范围的；如果没有找到，就只能看到窗口抖动。如果有指定范围的目标进入窗口，该窗口就会开始跟踪这个目标。

CAMshift 增加了均值漂移的复杂性，其与直接采用均值漂移的区别在于：在调用CAMshift后，会根据具体的旋转来绘制矩阵，这种旋转会与被跟踪对象一起旋转。



（4）卡尔曼滤波器

当含有噪声的数据输入到卡尔曼滤波器之中，滤波器会对这些数据进行递归，同时经过比较寻优，找到一个符合实际的合理估计值。

这种算法在计算时主要有两个过程：其一是预测阶段，这步中根据当前点的参数来预测目标的新位置；其二是更新阶段，这一步中会对目标位置的相关信息进行保存，同时修正协方差，方便下一步计算。

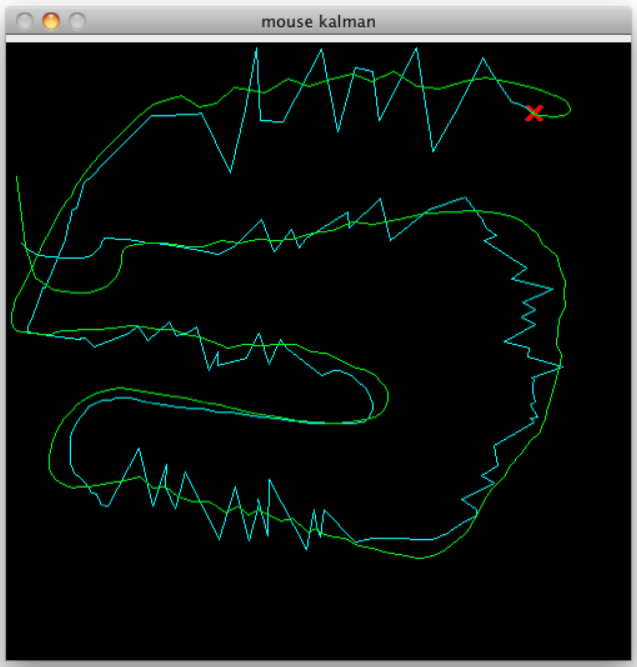
卡尔曼滤波器类的构造函数有如下可选参数：

dynamParams:该参数表示状态维度。

MeasureParams:该参数表示测量的维度。

ControlParams:该参数表示控制的维度。

Vector.type:该参数表示所创建的矩阵类型。



4.1.4 仿真研究

本次设计使用了背景分割方法进行动态检测，检测结果根据情况不同输出不同结果，检测程序见附录。

结论

本设计通过OpenCv视频处理平台，利用Python语言设计出基于视频监控的灯控系统。该程序视频提取功能、图像灰度转换亮度判断功能和动态检测功能。在现实生活中，该程序可以用于校园教室、会议厅等场景的灯光控制。该程序在下列情况中得以应用：

1. 当光照强度非常充裕时，监控摄像头对室内的实时情况进行录像，录像文件通过程序提取功能进入程序进行处理，光强足够条件下，程序会直接结束，不给灯光控制系统发出任何指令。
2. 当光照强度不足时，程序对监控摄像头提取的视频文件进行与之前相同的处理步骤，光照条件不足的情况下，程序会从光强检测步骤进入到下一步动态检测步骤，若图像中未检测到动态物体，则程序直接结束。
3. 同理当光照强度不足时，程序执行与上面相同的视频提取与亮度检测步骤，光照强度不足，程序进行动态检测步骤，若检测出有物体的动态反应则进行输出，对灯光控制器发出指令，使其控制灯光开启。

当然该软程序仍存在一些不足，首先是该程序运用的动态检测方式较为敏感，易受到干扰，其次，灰度值是由人为输入数据进行判定的，不够简洁且准确度具有一定的偏差。该软件还可将录制视频的处理改成实时监控处理控制，这样可以省略视频录制截取的步骤，可以很大程度上节约存储空间。未来该程序的完善可以从这几方面进行改进与升级。

致谢

截止今日，我的毕业论文撰写工作终于要告一段落了，回想起从今年三月份到现在几个月之中的点点滴滴，心中不禁感慨万千。有迷茫，最开始选题时知识储备的不足，让我在敲定题目这一块花了不少时间；有坚持，这个过程并不是一帆风顺的，需要进行大量枯燥、千篇一律的工作，还会有各种各样的困难，不过贵在持之以恒；有收获，查阅大量的资料，增强了专业素养，扩展了自己的知识面。

在导师的帮助下，今年三月初，我定下了我的论文题目：基于监控视频的智能灯控系统。敲定论文题目之后，我便开始查阅大量的相关文献，了解这一块的相关研究状况，并对前人的一些研究成果进行分析，对我所研究的对象有了一定的了解。在张旭秀导师细心的指导下，终于使我对自己现在的工作学习方向和方法有了掌握。

查阅资料的过程是枯燥的，在三月份，大量的时间都用在搜集相关资料这项工作中。把查找的资料和相关知识进行记录，并进行分析和研究，提出自己的看法。

3月底，资料已经查找完毕了，我开始着手整个程序的运行步骤设计工作。在设计过程中由于我之前从未接触过OpenCv视频处理平台，因此开始的时候我一窍不通。后来通过张旭秀老师的耐心指导，闫涵学长的认真帮助和我自己的努力学习终于完成了整个程序运行步骤与结构的设计，这一过程使我受益匪浅。当然其中也有部分功能虽然经过不懈努力仍未实现，然而我认为这不算什么，努力的过程是最重要的。

4月初，我开始着手软件的编写工作，这一过程更是不容易，也是多亏了张旭秀老师和闫涵学长的帮助最终成功编写完成。时至今日，我的论文已经完成，回想这几个月的时间真的是永生难忘。这几个月的时间里我学到了如下内容：

1. Python软件与OpenCv平台的安装与环境变量改写方式。
2. Python软件的相关函数库及其大致功能。
3. Python软件的编程语言与相关语言的阅读能力。

还收获了更多的是无法用语言表述的学习方法和师生情谊。我一定会带着这份收获走好毕业以后的每一步，为老师争光为学校添彩。

逝者如斯夫，时间如白驹过隙，四年的大学生活转瞬即逝，又到了一个夏天，又是一个说再见的季节。此时我的脑海中不禁浮现出四年前的那个夏天，我穿着半袖，拉着似乎装满了憧憬与梦想的沉甸甸的箱子，踏进了这个校门，开始了我们高中三年来日思夜想的大学生活。四年中，或平平淡淡，或收获满满，无论怎般终究是成长。在这里首先我得感谢我的父母，是你们给了我生命，含辛茹苦默默无闻供我上学。其次感谢我的指导老师，感谢在这段时间来的栽培，迷茫时的解疑答惑，困境时的悉心指导，从一开始的选题到论文的撰写，再到最后的修改以及答辩的指导，都给了我莫大的帮助。最后还得感谢我的师兄在这段时间对我的帮助，在论文开题的初期他亲自带领我制定程序执行步骤设计计划，教我学习Python语言，并为我提供了相关的很多学习资源，在这里我要真诚的道一声感谢。感谢的人太多，难以一一言表，我只希望自己通过以后更加努力的学习和工作来回报这些在我成长路上提供帮助的人。

参考文献

[1][基于OpenCV-Python的图像分割技术的设计与应用研究](http://kns.cnki.net/kcms/detail/detail.aspx?filename=TXWL201819079&dbcode=CJFQ&dbname=CJFDTEMP&v=" \t "http://kns.cnki.net/kcms/detail/frame/kcmstarget)[J]. 陈之尧.中国新通信. 2018(19)

[2][基于OpenCV-Python的图像分割技术的设计和应用](http://kns.cnki.net/kcms/detail/detail.aspx?filename=ELEW201803065&dbcode=CJFQ&dbname=CJFD2018&v=" \t "http://kns.cnki.net/kcms/detail/frame/kcmstarget)[J]. 周宇.电子世界. 2018(03)

[3][Python语言在科学算法中的优势](http://kns.cnki.net/kcms/detail/detail.aspx?filename=XXDL201904025&dbcode=CJFQ&dbname=CJFD2019&v=" \t "http://kns.cnki.net/KCMS/detail/kcmstarget)[J]. 刘瑞.信息与电脑(理论版). 2019(04)

[4][利用Python对自然语言进行简单处理](http://kns.cnki.net/kcms/detail/detail.aspx?filename=XDSM201907086&dbcode=CJFQ&dbname=CJFD2019&v=" \t "http://kns.cnki.net/KCMS/detail/kcmstarget)[J]. 刘家岐.现代商贸工业. 2019(07)

[5][面向非计算机专业学生的Python教学内容设计](http://kns.cnki.net/kcms/detail/detail.aspx?filename=TZXB201803010&dbcode=CJFQ&dbname=CJFD2018&v=" \t "http://kns.cnki.net/KCMS/detail/kcmstarget)[J]. 闯跃龙,郭文平.台州学院学报. 2018(03)

[6][Python在科学计算中的应用](http://kns.cnki.net/kcms/detail/detail.aspx?filename=SZJT201611051&dbcode=CJFQ&dbname=CJFD2016&v=" \t "http://kns.cnki.net/KCMS/detail/kcmstarget)[J]. 姚建盛,李淑梅.数字技术与应用. 2016(11)

[7][Python与R语言联合应用的实现](http://kns.cnki.net/kcms/detail/detail.aspx?filename=ZGWT201702054&dbcode=CJFQ&dbname=CJFD2017&v=" \t "http://kns.cnki.net/KCMS/detail/kcmstarget)[J]. 平凯珂,陈平雁.中国卫生统计. 2017(02)

[8][Python语言的探讨](http://kns.cnki.net/kcms/detail/detail.aspx?filename=TXWL201707084&dbcode=CJFQ&dbname=CJFD2017&v=" \t "http://kns.cnki.net/KCMS/detail/kcmstarget)[J]. 史梦楚.中国新通信. 2017(07)

[9][数字图像处理技术及其应用](http://kns.cnki.net/kcms/detail/detail.aspx?filename=DZRU201718070&dbcode=CJFQ&dbname=CJFD2017&v=" \t "http://kns.cnki.net/kcms/detail/frame/kcmstarget)[J]. 姚希.电子技术与软件工程. 2017(18)

[10][关于图像处理技术现状及发展的分析](http://kns.cnki.net/kcms/detail/detail.aspx?filename=ZXLJ201825040&dbcode=CJFQ&dbname=CJFD2018&v=" \t "http://kns.cnki.net/KCMS/detail/kcmstarget)[J]. 安丛姝.科技资讯. 2018(25)

[11][探究图像处理的关键技术](http://kns.cnki.net/kcms/detail/detail.aspx?filename=WXXJ201811165&dbcode=CJFQ&dbname=CJFD2018&v=" \t "http://kns.cnki.net/KCMS/detail/kcmstarget)[J]. 康春艳.计算机产品与流通. 2018(11)

[12][图像处理中颜色模式的探究](http://kns.cnki.net/kcms/detail/detail.aspx?filename=DNBC201901046&dbcode=CJFQ&dbname=CJFD2019&v=" \t "http://kns.cnki.net/KCMS/detail/kcmstarget)[J]. 肖彬.电脑编程技巧与维护. 2019(01)

[13][计算机图像处理技术的特点与应用研究](http://kns.cnki.net/kcms/detail/detail.aspx?filename=CXJL201903067&dbcode=CJFQ&dbname=CJFD2019&v=" \t "http://kns.cnki.net/KCMS/detail/kcmstarget)[J]. 刘晓杰.信息记录材料. 2019(03)

[14][计算机技术在图像处理中的应用分析](http://kns.cnki.net/kcms/detail/detail.aspx?filename=CAIZ201816202&dbcode=CJFQ&dbname=CJFD2018&v=" \t "http://kns.cnki.net/KCMS/detail/kcmstarget)[J]. 张璋.才智. 2018(16)

[15][计算机图像处理技术的发展及未来展望](http://kns.cnki.net/kcms/detail/detail.aspx?filename=DNMI201807043&dbcode=CJFQ&dbname=CJFD2018&v=" \t "http://kns.cnki.net/KCMS/detail/kcmstarget)[J]. 杨怀义.电脑迷. 2018(07)

[16][计算机图像处理技术的探析](http://kns.cnki.net/kcms/detail/detail.aspx?filename=DNMI201806121&dbcode=CJFQ&dbname=CJFD2018&v=" \t "http://kns.cnki.net/KCMS/detail/kcmstarget)[J]. 许友文.电脑迷. 2018(06)

[17][计算机图像处理技术的发展趋势](http://kns.cnki.net/kcms/detail/detail.aspx?filename=XXDL201618074&dbcode=CJFQ&dbname=CJFD2016&v=" \t "http://kns.cnki.net/KCMS/detail/kcmstarget)[J]. 柳彦合.信息与电脑(理论版). 2016(18)

附录