**基于HSV算法的多监控相机场景的行人分析研究**

**及系统实现研究报告**

电子信息与电气工程学院 陈宇翔 5110309783

指导教师：杨华 马文琪 电子信息与电气工程学院

摘 要

随着我国在近几年经济的飞速增长，越来越多的人涌入城市。而人们为城市的发展贡献自己力量的同时，伴随而来的是大量人口流动带来的社会治安等问题。随着公共场所监视系统的日趋完善，我们可以获得大量的视频影像资料，从而为识别某一个人在不同镜头下的出现创造了可能。这次项目实现了基于多监控相机场景的行人比对技术，而本篇研究报告将主要介绍该技术基于HSV算法的原理和实现。

HSV(Hue, Saturation, Value)是根据颜色的直观特性由A.R.Smith在1978年创建的一种颜色空间，也称六角椎体模型。这个模型中颜色的参数分别是：色调(H)，饱和度(S)和亮度(V)。HSV 以人类更熟悉的方式封装了关于颜色的信息：“这是什么颜色？深浅如何？明暗如何？”，从而在计算机领域的图像处理中得到了极大应用。在这个课题中，我们在前景分割后采用不同的算法来进行不同镜头下的行人比对，其中就包括了使用HSV颜色空间来对不同图片中的同一个行人进行识别。

本篇研究报告的第一部分先从总体上阐述了项目所研究的内容及其现实意义，并介绍了目前常见的一些比对算法。第二部分则简单介绍了HSV算法的原理。在第三部分，我将详细介绍基于HSV算法的行人比对的步骤，并对其中关键的步骤进行深入分析。报告的第四部分将展示对不同镜头下行人进行分割并使用HSV算法进行识别的程序的运行结果。第五部分会对程序的性能进行分析，并提出可能的改进方法。而报告的最后一部分则会对此研究项目进行总结，并对其在未来的实用性进行展望。

关键词： 多监控相机场景，行人比对，HSV，算法实现

**ABSTRACT**

With the fast increase of China’s economy, more and more people flood into cities. Fairly much contributions as they make to the development of the cities, they do cause the problem of social security as a result of the huge flow of such a large population. Luckily, since many supervisory devices are set in public places, we are enabled to recognize a certain people showing up in different monitors. So this project implements the technique of comparing people based on multi-camera scene, and this study report will introduce the implementation of this technique using the algorithm of HSV.

HSV (Hue, Saturation, Value) is a kind of color space established by A.R.Smith in 1978, according to the intuitive features of colors. In this model, the parameters of the colors are: hue (H), saturation (S), and value (V). And HSV encapsulate the information of colors in a method people familiar with, “What is this color? Is it dark or shallow? How is its brightness? ”, and thus it is widely used in image processing. In this project, we use different methods to compare people in multi-camera after the segment of foreground, and of course it contains the method of using HSV color space to distinguish the same person in different graphs.

So in this study report, we will introduce the contents of the project and its realistic significance, as well as some common algorithms of comparing people nowadays in the first part. Then we’ll say something about the basic principles of HSV algorithm in the second part. In the third part, I’ll describe how I compare person by HSV in details and make a deep analysis for the key steps in it. The fourth part will show the results of the program using HSV algorithm to discern people after foreground segment. Then comes the analysis of the performance of the program and its possible improvements in the fifth part. And in the last part we’ll make a conclusion on the project and propose its future work prospects.

**Keywords**：Multi-camera Scene, Pedestrian Comparing, HSV, Algorithm implementation

**1 项目研究内容和现实意义**

1.1 研究背景

随着我国在近几年经济的飞速增长，越来越多的人涌入城市。而人们为城市的发展贡献自己力量的同时，也带来了诸如社会治安不稳定等一系列问题。但是如果使用传统的人工识别目标的方法，不仅会造成效率低下，而且准确率也收到人类本身的主观感觉影响。如果能开发出基于多监控相机场景的行人比对技术，将会对治安的管理以及警方的日常工作造成极大的便利。因此，这种利用前景分割后检测不同镜头下行人的算法，对整个社会的和谐发展有着极为重要的作用和意义。

1.2 应用前景

多监控相机场景的行人比对技术有着广泛的应用前景，通常可以用在：

（1）管理客流量较大的交通场所。比如：寻找火车站、地铁站、公交车站及这些车内的犯罪嫌疑人，从而可以更加高效的锁定警方正在监视的人，并能获得其在不同地点的动态。继而可以大大减少负责在视频中寻找嫌疑人的民警工作量，并且对这些场所的恶性行为造成了一定震慑，有利于交通枢纽处的治安管理。

（2）寻找走失的人。寻找走失亲属的人可以将亲属的一张与走失时比较相似的照片提供给警方，从而使得在多个监控摄像中寻找这个人的难度大大减小，并且提高了寻找时的效率和准确度。

（3）在各种场合实时监控人群中的可疑人员，及时发现突发情况。这样可以减少便衣警察的数量和出警次数，并能实现各个监控点的无缝衔接。

1.3 研究内容

本课题的研究中，我们对不同相机拍下的场景中的同一个行人进行了识别。我们利用从不同相机处获得的视频进行行人的检测。先对这些不同地点拍摄的视频进行前景分割，以去除背景噪声的干扰，然后将获得的前景图片使用不同的算法利用OpenCv进行图像之间的比对。通过比较，我们就可以获得与提供的目标较为相近的行人的图片，从而完成对多监控相机下的行人比对。[1]

在这篇报告中，主要是完成利用HSV算法进行的检测。但是实际上可以进行检测的算法还有很多。比如我们还可以利用RGB颜色空间来对图片进行检测。采用这种编码方法，每种颜色都可用三个变量来表示-红色绿色以及蓝色的强度。只要将每张图片的RGB三维矩阵画出，并加以比对既可以得到我们想要的相近图片。RGB颜色空间是图像处理中最基本、最常用、面向硬件的颜色空间。我们采集到的彩色图像，一般就是被分成R、G、B的成分加以保存的。因此，RGB颜色空间在处理图像时显得比较直观。但是RGB算法也有一个不足，就是它对光线的明暗程度比较敏感，并且不像HSV那样可以让人对颜色有个直观的感觉。我们也可以使用SIFT算法进行图片检测。SIFT，即尺度不变特征转换(Scale-invariant feature transform ,SIFT)，是用于图像处理领域的一种描述子。这种描述具有尺度不变性，可在图像中检测出关键点。SIFT算法大致有四个步骤：尺度空间极值检测，关键点定位，方向确定和关键点描述。在每个关键点的领域内测量图像局部的梯度。最终用一个特征向量来表达。SIFT算法的优点是具有较强的匹配能力和鲁棒性，但其缺点是维数高实时性差，即需要经过相当数量的计算，因此速度较慢。我们还可以采用的是HOG算法。HOG描述器最重要的思想是：在一幅图像中，局部目标的表象和形状能够被梯度或边缘的方向密度分布很好地描述。具体的实现方法是：首先将图像分成小的连通区域，我们把它叫细胞单元。然后采集细胞单元中各像素 点的梯度的或边缘的方向直方图。最后把这些直方图组合起来就可以构成特征描述器。与其他的特征描述方法相比，HOG有很多优点，比如它的位置和方向空间的量化一定程度上可以抑制平移和旋转带来的影响，并且由于采取在局部区域归一化直方图，可以部分抵消光照变化带来的影响。使得图像所需要的表征数据的维度降低了。不过，HOG也有缺点：它的描述子生成过程冗长，导致速度慢，实时性差；很难处理遮挡问题。同时由于梯度的性质，该描述子对噪点相当敏感。当然，我们还可以采用LBP, Gabor等方法从纹理的角度对两幅图像的相似度进行比较。

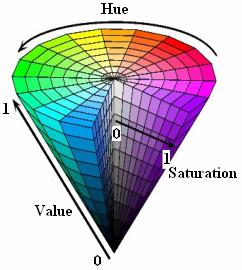
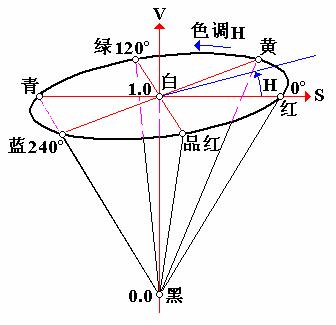
在此课题中，我使用的是HSV方法来进行行人的检测。具体的实现及原理将在下面的几个小节进行详细的表述和分析。

**2 HSV算法简介**

2.1 HSV算法原理

这个模型中颜色的参数分别是：色彩（色调）（H），纯度（饱和度）（S），明度（亮度）（V）。

HSV模型的三维表示从RGB立方体演化而来。设想从RGB沿立方体对角线的白色顶点向黑色顶点观察，就可以看到立方体的六边形外形。六边形边界表示色彩，水平轴表示纯度，明度沿垂直轴测量。H参数表示色彩信息，即所处的光谱颜色的位置。该参数用一角度量来表示，红、绿、蓝分别相隔120度。互补色分别相差180度。纯度S为一比例值，范围从0到1，它表示成所选颜色的纯度和该颜色最大的纯度之间的比率。S=0时，只有灰度。V表示色彩的明亮程度，范围从0到1。它和光强度之间并没有直接的联系。[2]



**Figure 2-1 HSV颜色空间模型 1** **Figure 2-2 HSV颜色空间模型2**

1. 从 RGB 到 HSV 的转换[3]  
   设 (r, g, b) 分别是一个颜色的红、绿和蓝坐标，它们的值是在 0 到 1 之间的实数。设 max 等价于 r, g 和 b 中的最大者。设 min 等于这些值中的最小者。要找到在 HSL 空间中的 (h, s, l) 值，这里的 h ∈ [0, 360）是角度的色相角，而 s, l ∈ [0,1] 是饱和度和亮度，计算为：

（2-1）

（2-2）

（2-3）

**(2) 从 HSV 到 RGB 的转换**

类似的，给定在 HSV 中 (h, s, v) 值定义的一个颜色，带有如上的 h，和分别表示饱和度和明度的 s 和 v 变化于 0 到 1 之间，在 RGB 空间中对应的 (r, g, b) 三原色可以计算为：

mod 6 （2-4）

（2-5）

（2-6）

（2-7）

（2-8）

对于每个颜色向量 (r, g, b),

（2-9）

HSV对用户来说是一种直观的颜色模型。我们可以从一种纯色彩开始，即指定色彩角H，并让 V=S=1，然后我们可以通过向其中加入黑色和白色来得到我们需要的颜色。增加黑色可以减小V而S不变，同样增加白色可以减小S而V不变。例如，要得到深蓝色，V=0.4 S=1 H=240度。要得到淡蓝色，V=1 S=0.4 H=240度。

2.2 HSV算法实现思路

由于HSV中的亮度（明度）V表示了整幅图片的明暗程度，因此，如果要进行前景分割的话，只需要将HSV三个属性中的V属性去掉即可，这样就避免了由于背景是黑色而导致的对整幅图片的干扰问题。而由于去掉V属性可以极大程度上减轻行人比对时光线的明暗对画面的干扰，所以我的这个算法中也只使用了HS两个属性。这样的话，事实上，由于背景的干扰已经不明显了，因此这个程序也可以不进行前景分割了，只需要将要比对的图片输入即可。这样，通过将图像化为二维的直方图矩阵并进行比较，选出与原图最为贴近的几幅图像即可。[4]

2.3 选用HSV算法的原因

由于我们要进行对各种算法在行人比对方面的研究，因此我们需要尝试很多的方法。我选择HSV方法的原因是因为HSV是一种比较直观的模型，选用HSV算法能比较直观地从色调、饱和度、亮度三个方面对一张图像进行分解。同时我们也可以通过对亮度的取舍来避免背景的明暗对比对结果的影响。

**3 程序实现步骤**

3.1 OpenCV简介

本课题中我们主要是使用OpenCv来对图像进行分析的。OpenCV于1999年由Intel建立，如今由Willow Garage提供支持。OpenCV是一个基于（开源）发行的跨平台计算机视觉库，可以运行在Linux、Windows和Mac OS操作系统上。它轻量级而且高效——由一系列 C 函数和少量 C++ 类构成，同时提供了Python、Ruby、MATLAB等语言的接口，实现了图像处理和计算机视觉方面的很多通用算法。而我们的这个课题中主要是通过使用OpenCv与C++相结合来得到最终的实验结果。[5]

3.2 程序实现的基本步骤[6]

（1）首先我们将所得到的视频进行前景分割，并使用截图程序不断地得到行人的前景图片

（2）然后我们将得到的行人图片放缩至合适大小，并使用HSV算法加以比对

（3）将与原图接近的图片编号输出并与实际情况对比

3.3 前景分割及截图的实现过程分析

这里所需的大部分函数都基于OpenCv中的示例程序，因此对于示例程序中的一些常见函数不做过多解释。同时，限于篇幅，下面给出的程序只是整个程序中比较重要的部分。

构建矩阵：

BackgroundSubtractorMOG2 bg\_model;

Mat img, fgmask, fgimg;

//构建前景图像的矩阵

char s[50];size\_t frameNum = 0;

for(;;){

cap >> img;

if( img.empty() ) break;

if( fgimg.empty() )

fgimg.create(img.size(), img.type());

//即更新model中的内容

bg\_model(img, fgmask, update\_bg\_model ? -1 : 0);

fgimg = Scalar::all(0);

img.copyTo(fgimg, fgmask);

Mat bgimg;

bg\_model.getBackgroundImage(bgimg);

sprintf(s, "D:\\lab\\exp.jpg", frameNum++);

//在D盘的lab文件夹先新建一个exp.jpg图片，将此时的前景图片存在exp.jpg中

imwrite(s, fgimg);

Mat newimg; FILE\* f = 0; char \_filename[1024];

newimg = imread(s);

if( newimg.data ) strcpy(\_filename, s);

else{

f = fopen(argv[1], "rt");

if(!f){

fprintf( stderr, "ERROR: the specified file could not be loaded\n");

return -1;

}

}

上述程序大部分基于OpenCv中前景分割的示例函数，我在上面做了一下修改使它可以截取当前的前景图片并保存起来。这样使用上述程序即可以使用HOG算法得到每个人去除了背景以后的图像。

3.4 HSV算法的实现过程分析

HSV算法是把两张图片中的H和S分量计算出来，然后将其转化为二维矩阵，并使用OpenCv中的comparehist函数进行比较，最后得出与示例图像最相近的结果。而在这个算法的实现过程中，我把原图片和目标图片分成上下两部分分别进行比较，然后将代表相关度的变量排序并输出结果。程序关键部分实现如下：

//储存基准图像和另外两张对比图像的矩阵( RGB 和 HSV )

Mat src\_base, hsv\_base;Mat src\_test1, hsv\_test1;

Mat hsv\_half\_down;Mat hsv\_half\_up;Mat hsvbase\_half\_down;Mat hsvbase\_half\_up;

/// 对hue通道使用50个bin,对saturatoin通道使用60个bin

int h\_bins = 50; int s\_bins = 60;

int histSize[] = { h\_bins, s\_bins };

// hue的取值范围从0到256, saturation取值范围从0到180

float h\_ranges[] = { 0, 256 };float s\_ranges[] = { 0, 180 };

const float\* ranges[] = { h\_ranges, s\_ranges };// 使用第0和第1通道

int channels[] = { 0, 1 };/// 直方图矩阵，二维（其中将整幅图像分为两个半身图像）

MatND hist\_base;MatND hist\_test1;MatND hist\_half\_down;

MatND hist\_half\_up;MatND histbase\_half\_down;MatND histbase\_half\_up;

/// 装载两张背景环境不同的图

for (i = 0; i < count1 - 3; ++i) {

path1 = "D:\\VIPeR\\img1\\" + (string)result1[i+2];

src\_base = imread(path1); //读入文件夹1的文件

for (j = 0; j < count2 - 3; ++j) {

path2 = "D:\\VIPeR\\img2\\" + string(result2[j+2]);

src\_test1 = imread(path2);

cvtColor(src\_base, hsv\_base, CV\_BGR2HSV ); /// 转换到 HSV

cvtColor(src\_test1, hsv\_test1, CV\_BGR2HSV );

hsv\_half\_down = hsv\_test1( Range( hsv\_test1.rows/2, hsv\_test1.rows - 1 ), Range( 0, hsv\_test1.cols - 1 ) ); //半身基准图像的二维直方图（只列出一组，其余略）

calcHist( &hsv\_half\_down, 1, channels, Mat(), hist\_half\_down, 2, histSize, ranges, true, false );

normalize( hist\_half\_down, hist\_half\_down, 0, 1, NORM\_MINMAX, -1, Mat() );

///使用**Bhattacharyya 距离**对比标准将基准图像(hist\_base)的直方图与其余各直方图进行对比，这里列出一组，其余略

double base\_half = compareHist( histbase\_half\_down, hist\_half\_down, CV\_COMP\_BHATTACHARYYA);

double base\_half2 = compareHist( histbase\_half\_up, hist\_half\_up, CV\_COMP\_BHATTACHARYYA);

half\_relativity[j] = base\_half+base\_half2;

half\_ordered\_relativity[j] = base\_half+base\_half2;

}}

使用上述程序，我们即可以得到对应于每一张图片的前5个最接近的结果，从而完成对多相机监控的行人分析研究。其中，我们使用了HSV算法中的H和S两个元素构建出二维的向量矩阵，并且在最后使用OpenCv的自带函数Comparehist中的**Bhattacharyya 距离比较出了各个图片之间的相似度。**

**4 程序的运行结果**

本次实验中，我的HSV程序的运行结果大体上还是比较令人满意的。在随机选取的27组图片中，匹配的结果如下表所示。（使用**Bhattacharyya 距离在对图片进行比较时，得到的相关值越大，说明两幅图片越相似，同时第一组的001。Jpg与第二组的001.jpg相对应，以此类推）。**

**表 4-1 不同角度的同一个人的图像比对**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 组号\排名 | 1 | 2 | 3 | 4 | 5 |
| 000.jpg | 000.jpg | 012.jpg | 021.jpg | 017.jpg | 010.jpg |
| 001.jpg | 021.jpg | 003.jpg | 017.jpg | 012.jpg | 019.jpg |
| 002.jpg | 009.jpg | 012.jpg | 022.jpg | 000.jpg | 017.jpg |
| 003.jpg | 021.jpg | 003.jpg | 020.jpg | 006.jpg | 017.jpg |
| 004.jpg | 021.jpg | 019.jpg | 020.jpg | 017.jpg | 003.jpg |
| 005.jpg | 007.jpg | 023.jpg | 002.jpg | 000.jpg | 009.jpg |
| 006.jpg | 021.jpg | 006.jpg | 003.jpg | 017.jpg | 012.jpg |
| 007.jpg | 007.jpg | 023.jpg | 017.jpg | 010.jpg | 021.jpg |
| 008.jpg | 022.jpg | 008.jpg | 023.jpg | 012.jpg | 018.jpg |
| 009.jpg | 009.jpg | 012.jpg | 022.jpg | 021.jpg | 017.jpg |
| 010.jpg | 010.jpg | 021.jpg | 012.jpg | 022.jpg | 017.jpg |
| 011.jpg | 000.jpg | 023.jpg | 002.jpg | 007.jpg | 009.jpg |
| 012.jpg | 021.jpg | 012.jpg | 019.jpg | 022.jpg | 017.jpg |
| 013.jpg | 021.jpg | 019.jpg | 017.jpg | 023.jpg | 020.jpg |
| 014.jpg | 009.jpg | 012.jpg | 017.jpg | 019.jpg | 000.jpg |
| 015.jpg | 023.jpg | 022.jpg | 019.jpg | 009.jpg | 021.jpg |
| 016.jpg | 022.jpg | 000.jpg | 010.jpg | 012.jpg | 017.jpg |
| 017.jpg | 021.jpg | 017.jpg | 019.jpg | 003.jpg | 012.jpg |
| 018.jpg | 023.jpg | 022.jpg | 018.jpg | 010.jpg | 025.jpg |
| 019.jpg | 021.jpg | 020.jpg | 019.jpg | 012.jpg | 003.jpg |
| 020.jpg | 021.jpg | 020.jpg | 003.jpg | 019.jpg | 017.jpg |
| 021.jpg | 021.jpg | 003.jpg | 020.jpg | 017.jpg | 019.jpg |
| 022.jpg | 022.jpg | 023.jpg | 008.jpg | 018.jpg | 019.jpg |
| 023.jpg | 023.jpg | 021.jpg | 009.jpg | 022.jpg | 019.jpg |
| 024.jpg | 024.jpg | 023.jpg | 000.jpg | 012.jpg | 017.jpg |
| 025.jpg | 025.jpg | 022.jpg | 023.jpg | 018.jpg | 008.jpg |
| 026.jpg | 019.jpg | 021.jpg | 020.jpg | 017.jpg | 026.jpg |

由上表可见，实验的结果大体上还算比较准确，除了中间的几组以外，剩余的图片基本上都与各自的原图片完成了匹配。而未完成匹配的几组，据我分析可能是由于目标图片与原图片还是受到了一定程度的光照影响而使得自身的S（饱和度）属性受到了一定程度上的干扰。同时也由于有的图片的HSV矩阵较为均匀，从而完成了与很多其他矩阵的匹配。虽然两幅图片并不是十分相似，但是可能它们的矩阵构造碰巧满足了HSV中**Bhattacharyya 距离的条件。比如第一组的002.jpg图片和第二组的002.jpg, 009.jpg, 012.jpg，如下：**

****   

**Figure 4-1 Figure 4-2 Figure 4-3 Figure 4-4**

**第一组的002.jpg 第二组的002.jpg 第二组的009.jpg 第二组的012.jpg**

所以从上面的这组图中可以看出，被判定为相似的图像还是有其相似之处的，所以我们勉强可以接受实验中存在的误差。

**5 对程序的分析和改进**

OpenCv的[compareHist](http://opencv.willowgarage.com/documentation/cpp/imgproc_histograms.html?#compareHist) 执行了具体的直方图对比的任务。该函数提供了4种对比标准来计算相似度：

1. **Correlation ( CV\_COMP\_CORREL )**

（5-1）

其中

（5-2）

N是直方图中bin的数目。

1. **Chi-Square ( CV\_COMP\_CHISQR )**

（5-3）

1. **Intersection ( CV\_COMP\_INTERSECT )**

（5-4）

1. **Bhattacharyya 距离( CV\_COMP\_BHATTACHARYYA )**

（5-5）

而我在这个程序中则主要使用了第四个对比标准，其他三个标准我也进行过尝试，效果和第四个方法的效果大致相同。因此，我在想，如果我们可以找到一个算法，将上面四种属性完美地结合起来，应该可以获得更好的比对结果。同时，在这个HSV算法的实现中，我们还要注意许多问题。比如，虽然使用HS两个属性时对背景的要求并不是很严格，但是最好还是能把前景分割出来，排除背景噪声的干扰。同时，我们还要注意的是，在比对两个镜头下的图像时，最好可以把要比对的图像变换成同样大小，以排除图像的尺寸对结果的干扰。

至于性能方面，这个算法所实现的程序的运行时间还是比较快的，唯一需要注意的环节就是在从前景视频中获得每一帧的图片时，我们最好关闭前台显示出的bgimg, fgimg等视频窗口，从而使程序可以快速地完成截图的任务。

**6 总结和展望**

科技在发展，时代在进步，我们正处于新一轮科技革命的浪潮中，因此未来的世界中，人类的工作量必将被大大减少，取而代之的是越来越多的计算机来完成我们的很多任务。而基于多监控相机场景的行人比对技术可以配合未来完善的人脸识别系统来维护公共场所的治安，或者是帮助人们寻找自己失散的亲人。所以，这种技术的开发可以在未来进一步解放用于治安的人力资源，对整个社会的发展也具有极其深远的意义。

在这次研究中我从对OpenCv在C++上的实现一知半解，一直到现在可以比较熟练地使用OpenCv中的一些函数并构建自己的算法，其间我也付出了很多的努力。当然，随着我对OpenCv的了解进一步加深，我对项目的内容和意义也有了深入的了解，同时也对目前较为主流的一些行人比对算法有了一定的认识。这次研究中，我从OpenCv的示例程序入手，通过改写示例程序的部分代码来一步步提升自己对OpenCv的掌握，然后通过阅读关于HSV算法的一些文献，在OpenCv与C++结合的平台上实现了HSV算法，并将其用其在行人比对上，使之可以完成本次课题的要求。最后我也对HSV算法的优缺点进行了深入思考，研究了在不同光线照射以及不同对比标准对算法最终结果的影响。

这次我对基于HSV算法的多监控相机场景的行人比对技术进行了研究和实现，取得了一定的成果，但从一个研究者的角度来看，这个算法还有很多可以改进的地方，这也是我在未来可以进一步进行探索的一些问题：

1. HSV算法的结果与光线的强弱还是有很大的关系，因此，可以通过综合利用RGB,SIFT算法以及关于纹理的算法等来增强HSV算法的鲁棒性。
2. 目前所实现的HSV算法还只使用了**Bhattacharyya 距离这一种对比标准，因此在两张图片的比对上还是存在一些局限性的。如果能将四种比对标准结合起来的话，应该可以获得更加令人满意的结果。**
3. **同时，这个程序目前还是要先从两个视频中截取图像，然后对图像进行静态比较，这样的话没有办法进一步增强监控的实时性。所以，我认为可以进一步通过对程序的完善，使之具有在播放视频的同时动态比对行人的功能，从而增强行人比对的实时性。**

**7 致谢**

回顾做prp项目的这段时间，我感到受益匪浅，感触良多。

在做prp项目的这几个月中，虽然我在使用OpenCv方面遇到了很多困难，也造成了比较大的压力，但是从总体上来讲，我还是获得了非常多的收获。

首先，我要向杨华老师表示感谢，感谢杨老师给了我参加这次prp项目的机会以及杨老师一个学期以来对我们提供的指导和鼓励。同时，杨老师认真负责的精神和对她平易近人的态度也是我将来学习的楷模。然后，我还要特别感谢一下马文琪学长。他每个星期都会耐心地在实验室为我解答各种问题。无论我使用OpenCv时出现的问题，还是我在进行研究时所要确定的方向，都有马学长对我的帮助和关心。在他的帮助下，我对科学研究有了更深入的了解，也养成了认真勤恳的良好习惯。

其次，我还要感谢小组的另外三位成员——程栋彬学长，顾婕昱同学和罗若天同学。在整个实验的过程中，我了解到了与同学合作的重要性。尤其是与罗若天同学的深入合作，更让我领悟到了大家一起完成一个项目所带来的益处。正是由于上述三位同学的配合，我们的项目才能顺利完成。

这是我第一次正式进入实验室进行科研型的学习，这将是我的一段及其宝贵的经历。再次感谢杨华老师和马文琪学长！感谢程栋彬学长，顾婕昱同学和罗若天同学！

**8 参考文献**

[1] Gianfranco Doretto • Thomas Sebastian •Peter Tu • Jens Rittscher ：Appearance-based person reidentification in camera networks

[2] 百度百科，HSV颜色空间

http://baike.baidu.com/view/541362.htm#sub8445478

[3] Wikipedia，HSL和HSV色彩空间

https://zh.wikipedia.org/wiki/HSL%E5%92%8CHSV%E8%89%B2%E5%BD%A9%E7%A9%BA%E9%97%B4

[4]CSDN: 颜色空间RGB与HSV(HSL)的转换http://blog.csdn.net/idfaya/article/details/6770414

[5] References for OpenCv http://opencv.org/

[6] Jianxin Wu，Christopher Geyer，James M. Rehg ：Real-Time Human Detection Using Contour Cues