山东大学 计算机科学与技术 学院

计算机视觉 课程实验报告

|  |  |  |
| --- | --- | --- |
| 学号：201600301304 | 姓名：贾乘兴 | 班级：人工智能16 |
| 实验题目：基于直方图的目标追踪 | | |
| 实验内容：  一．图像直方图  1.图像直方图是对图像中各个灰度或者rgb数值的像素点个数的统计  2.在统计直方图后，可将直方图视作图像的分布密度函数，我们可以求出图像的像素点的分布函数，再对其进行归一化，得到最终的直方图，直方图可用于直方图均衡化，是图像处理中常见的方法  二．目标追踪  1.基于直方图的目标追踪问题，我们可以匹配两张图的直方图的相似度来判断其相似性，直方图可以很好的反应图像的特征，直方图计算并归一化的代码如下：  **for**(**int** j=y0;j<y0+dy;j++){  **for**(**int** i=x0;i<x0+dx;i++){  **for**(**int** c=0;c<3;c++){  t[(**int**)frame.at<Vec3b>(j,i)[c]][c]++;  }  } } **for**(**int** j=0;j<256;j++){  **for**(**int** c=0;c<3;c++){  d[j][c]=(**float**)(t[j][c])/sum;  } } **for**(**int** j=1;j<256;j++){  **for**(**int** c=0;c<3;c++){  d[j][c]+=d[j-1][c];  } } rectangle(frame,cvPoint(x0,y0),cvPoint(x0+dx,y0+dy),cvScalar(0,255,0),2);  2.本次试验选用了车辆视频，取视频中20-45帧，将20帧图像的目标标记好框，作为第一张图像，之后的匹配将基于该图像进行  3.得到第t张图后，查找下一张图最相似的框内图像，将其选定为下一个图像框，框内为追踪的目标  4.将图像连续显示，并储存中间结果  三．追踪方法  1.快速选取下一个候选图像的方法：基于运动速度的估计，在给定的上一个位置下，确定以原方形为中心的九宫格区域，在这个区域内，如果全部遍历仍然具有很大的时间代价，所以采用随机的方法，部分代码如下： **while**(1-cos>alpha&&pnum<500){  ddx=rand()%(dx/2)-dx/4;  ddy=rand()%(dy/2)-dy/4;  x1=x0+ddx;  y1=y0+ddy;  *// new h* **if**(x1>=160&&y1>=0&&x1<480-dx&&y1<480-dy)  //compute p  //compute cos  //compare  }  2.相似度测量使用两张图像直方图展成向量的余弦相似度，在余弦相似度接近1时相似度较大，余弦相似度计算如下：    代码如下：  **float** sumx=0; **float** sum1=0; **float** sum2=0; **for**(**int** j=1;j<256;j++){  **for**(**int** c=0;c<3;c++){  sumx+=d[j][c]\*d1[j][c];  sum1+=d[j][c]\*d[j][c];  sum2+=d1[j][c]\*d1[j][c];  } } *//cout<<sumx<<" "<<sum1<<" "<<sum2<<endl;* **if**(sum2!=0){  cos0=abs(sumx/sqrt(sum1\*sum2)); }**else**{  cos0=0; }  四．实验结果与分析  本实验每五帧截图一次，结果如下：  ../../20.png  20帧  ../../25.png  25帧  ../../30.png  30帧  ../../35.png  35帧  ../../40.png  40帧  ../../45.png  45帧  可见框的位置相对来说还算准确  五．实验改进  1.该图像可见车在变大，考虑使用可变化的候选框（归一化下直方图便于计算）  2.随机方法加快了速度，但是也导致一定情况下会出现取样不好的问题，所以可以基于运动估计进行采样  3.实验中对于附近的与原框重叠较大的候选框，可以使用一定的加速方法优化  六．实验代码  #include **<iostream>** #include **<opencv2/opencv.hpp>** #include **<cmath>** #include **<string>  using namespace** std; **using namespace** cv; **int** main(){  VideoCapture cap(**"/Users/apple/Desktop/1.avi"**);  **if**(!cap.isOpened()){  **return** -1;  }  **int** x,y,x0,y0,dx,dy;  **int** num=1;  Mat frame0;  cap >> frame0;  x=frame0.cols;  y=frame0.rows;  **int** t[256][3];  **float** d[256][3];  **for**(**int** j=0;j<256;j++){  **for**(**int** c=0;c<3;c++){  t[j][c] = 0;  d[j][c] = 0;  }  }  **float** alpha=0.001;  **int** sum=0;  *//(0,160)-->(0+470,160+320)* **for**(;;){  Mat frame;  cap >> frame;  num++;  **if**(num==20){  x0=350;  y0=20;  dx=110;  dy=120;  sum=dx\*dy;  *// h* **for**(**int** j=y0;j<y0+dy;j++){  **for**(**int** i=x0;i<x0+dx;i++){  **for**(**int** c=0;c<3;c++){  t[(**int**)frame.at<Vec3b>(j,i)[c]][c]++;  }  }  }  **for**(**int** j=0;j<256;j++){  **for**(**int** c=0;c<3;c++){  d[j][c]=(**float**)(t[j][c])/sum;  }  }  **for**(**int** j=1;j<256;j++){  **for**(**int** c=0;c<3;c++){  d[j][c]+=d[j-1][c];  }  }  rectangle(frame,cvPoint(x0,y0),cvPoint(x0+dx,y0+dy),cvScalar(0,255,0),2);  imshow(**"out"**,frame);  imwrite(**"/Users/apple/Desktop/20.png"**,frame);  }**else if**(num>20&&num<=45){  **int** t1[256][3];  **float** d1[256][3];  **for**(**int** j=0;j<256;j++){  **for**(**int** c=0;c<3;c++){  t1[j][c] = 0;  d1[j][c] = 0;  }  }  **float** cos=0;  **float** cos0=0;  **int** pnum=0;  **int** ddx=0;  **int** ddy=0;  **int** x1=160;  **int** y1=0;  **int** xm=x0;  **int** ym=y0;  **float** tm[256][3];  **float** dm[256][3];  **while**(1-cos>alpha&&pnum<500){  ddx=rand()%(dx/2)-dx/4;  ddy=rand()%(dy/2)-dy/4;  x1=x0+ddx;  y1=y0+ddy;  *// new h* **if**(x1>=160&&y1>=0&&x1<480-dx&&y1<480-dy){  pnum++;  **for**(**int** j=0;j<256;j++){  **for**(**int** c=0;c<3;c++){  t1[j][c]=0;  }  }  **for**(**int** j=y1;j<y1+dy;j++){  **for**(**int** i=x1;i<x1+dx;i++){  **for**(**int** c=0;c<3;c++){  t1[(**int**)frame.at<Vec3b>(j,i)[c]][c]++;  }  }  }  **for**(**int** j=0;j<256;j++){  **for**(**int** c=0;c<3;c++){  d1[j][c]=(**float**)(t1[j][c])/sum;  }  }  **for**(**int** j=1;j<256;j++){  **for**(**int** c=0;c<3;c++){  d1[j][c]+=d1[j-1][c];  }  }  *// compute cos* **float** sumx=0;  **float** sum1=0;  **float** sum2=0;  **for**(**int** j=1;j<256;j++){  **for**(**int** c=0;c<3;c++){  sumx+=d[j][c]\*d1[j][c];  sum1+=d[j][c]\*d[j][c];  sum2+=d1[j][c]\*d1[j][c];  }  }  *//cout<<sumx<<" "<<sum1<<" "<<sum2<<endl;* **if**(sum2!=0){  cos0=abs(sumx/sqrt(sum1\*sum2));  }**else**{  cos0=0;  }  *//cout<<cos0<<endl;* **if**(cos0>cos){  cos=cos0;  xm=x1;  ym=y1;  *// update* **for**(**int** j=0;j<256;j++){  **for**(**int** c=0;c<3;c++){  dm[j][c]=d1[j][c];  tm[j][c]=t1[j][c];  }  }  }  }  }  cout<<pnum<<endl;  cout<<ddx<<**" "**<<ddy<<endl;  *//cout<<xm<<' '<<ym<<endl;* x0=xm;  y0=ym;  *// update* **for**(**int** j=0;j<256;j++){  **for**(**int** c=0;c<3;c++){  d[j][c]=dm[j][c];  t[j][c]=tm[j][c];  }  }  rectangle(frame,cvPoint(x0,y0),cvPoint(x0+dx,y0+dy),cvScalar(0,255,0),2);  imshow(**"out"**,frame);  **if**(num%5==0){   stringstream sname;  string name1=**"/Users/apple/Desktop/"**;  string name2=**".png"**;  sname<<name1<<num<<name2<<endl;  imwrite(sname.str(),frame);  }  }  **if**(waitKey(30)>=0){  **break**;  }  } } | | |
| 实验过程中遇到和解决的问题：  （记录实验过程中遇到的问题，以及解决过程和实验结果。可以适当配以关键代码辅助说明，但不要大段贴代码。）   1. 相似度阈值设定较大，导致目标追踪失败，设定合适的阈值，通过实验的方法 2. 车辆在增大，初始框一定程度上设定合适 | | |
| 结论分析与体会：通过本次实验，了解了直方图的使用，同时，在解决问题的过程中有了更多的想法，并一定程度上进行了实现 | | |