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：图像追踪 | | |
| 实验内容：   1. 连通域 2. 实现图像的快速连通域算法，并将不同区域用不同颜色表示 3. 快速连通域算法：从左到右从上到下搜索，对于点m，其左侧相邻点像素值为p，类别为L(p)，右侧相邻点像素值为q，类别为L(q)，那么存在以下几种情形：当三个点像素值相等时，p和q属于同一类别，则m与p、q同类；当p、q不同类时，选取其中一类的为m的类，并将另一类归并到该类下；当m只与某一点像素值相等时，则归为该类；当m与两点像素都不相等时，新建类别 4. 并查集：最后使用并查集将各个类别分类到最终类别上 5. 随机选取每类别的颜色，并给图像上色 6. 选取图像处理结果如下：   ../../a.png ../../a0.png  e. 代码如下： *//6.1.1 分区域* **int** num=0; **struct** color{  **int** n;  **int** b;  **int** g;  **int** r; }c[1000];  **for**(**int** i=0;i<1000;i++){  c[i].b=0;  c[i].g=0;  c[i].r=0; }  c[0].n=0; c[0].b=random()%255; c[0].g=random()%255; c[0].r=random()%255;   **for**(**int** i=1;i<h;i++){  **for**(**int** j=1;j<w;j++){  **if**((**int**)input.at<uchar>(i,j)==(**int**)input.at<uchar>(i-1,j)||(**int**)input.at<uchar>(i,j)==(**int**)input.at<uchar>(i,j-1)){  **if**((**int**)input.at<uchar>(i-1,j)==(**int**)input.at<uchar>(i,j-1)){  **if**(c[t[i-1][j]].n==c[t[i][j-1]].n){  t[i][j]=t[i][j-1];  }**else**{  t[i][j]=t[i][j-1];  **if**(c[t[i-1][j]].n>c[t[i][j-1]].n){  c[t[i-1][j]].n=c[t[i][j-1]].n;  c[t[i-1][j]].b=c[t[i][j-1]].b;  c[t[i-1][j]].g=c[t[i][j-1]].g;  c[t[i-1][j]].r=c[t[i][j-1]].r;  }**else if**(c[t[i-1][j]].n<=c[t[i][j-1]].n){  c[t[i][j-1]].n=c[t[i-1][j]].n;  c[t[i][j-1]].b=c[t[i-1][j]].b;  c[t[i][j-1]].g=c[t[i-1][j]].g;  c[t[i][j-1]].r=c[t[i-1][j]].r;  }   }  }**else if**((**int**)input.at<uchar>(i,j)==(**int**)input.at<uchar>(i-1,j)){  t[i][j]=t[i-1][j];  }**else if**((**int**)input.at<uchar>(i,j)==(**int**)input.at<uchar>(i,j-1)){  t[i][j]=t[i][j-1];  }  }**else**{  num++;  c[num].n=num;  c[num].b=(random()\*2)%255;  c[num].g=(random()\*3)%255;  c[num].r=(random()\*4)%255;  t[i][j]=num;  }  } }  **for**(**int** i=0;i<h;i++){  **for**(**int** j=0;j<w;j++){  output.at<Vec3b>(i,j)[0]=c[t[i][j]].b;  output.at<Vec3b>(i,j)[1]=c[t[i][j]].g;  output.at<Vec3b>(i,j)[2]=c[t[i][j]].r;  } }  imshow(**"out"**,output); imwrite(**"/Users/apple/Desktop/a0.png"**,output); cvWaitKey(0);   1. 对二值图像删除较小的区域，保留最大的一个 2. 基于并查集统计各个类别的个数 3. 记录像素最大的类别（在部分问题中需要设置阈值去除较小的区域） 4. 删除其他类别 5. 实验结果如下：   ../../a1.png   1. 代码：   *//6.1.2 最大区域* num++; **int** q[num]; **for**(**int** i=0;i<num;i++){  q[i]=0; }  **for**(**int** i=0;i<h;i++){  **for**(**int** j=0;j<w;j++){  q[c[t[i][j]].n]++;  } } **int** max=0; **int** index=0; **for**(**int** i=1;i<num;i++){  *//cout<<q[i]<<endl;* **if**(q[i]>max){  max=q[i];  index=i;  } }  cout<<num<<endl; *//cout<<index<<endl;* **for**(**int** i=0;i<h;i++){  **for**(**int** j=0;j<w;j++){  **if**(c[t[i][j]].n==index){  out1.at<uchar>(i,j)=255;  }**else**{  out1.at<uchar>(i,j)=0;  }  } } imshow(**"out1"**,out1); imwrite(**"/Users/apple/Desktop/a1.png"**,out1); waitKey(0);   1. 距离场变换 2. 距离场是计算非零点到零点到最短路，一般可以用于计算图像的骨骼 3. 距离常用的定义如下：      1. 图像的距离场可以用o（n）复杂度算法近似计算 2. 本实验测试了opencv的distancetransform函数，使用了不同的距离进行计算（l1，l2，l12等等） 3. 最终得到的图像像素值为小数，需要标准化才可以显示 4. 几种处理如下：   ../../a.png ../../ak.png  ../../dd.jpg ../../dk.png  ../../timg.jpg ../../tk.png  彩色rgb图像处理结果与二值化的阈值有关   1. 代码如下：   Mat input=imread(**"/Users/apple/Desktop/dd.jpg"**); cvtColor(input,input,***CV\_BGR2GRAY***); threshold(input,input,128,255,***THRESH\_BINARY***); Mat output;  distanceTransform(input,output,***CV\_DIST\_L12***,***CV\_DIST\_MASK\_PRECISE***); normalize(output,output,0,1,cv::***NORM\_MINMAX***); imshow(**"out"**,output); *//imwrite("/Users/apple/Desktop/a1.png",output);* waitKey(0); | | |
| 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. 在实验过程中，发现了在倾斜的部分出现了延长的线，后来直接使用并查集后的结果解决 2. distancetransform的过程中发现总是显示原图，后将图像normal之后就可以显示正常了 | | |
| 结论分析与体会： 本次实验实现了快速连通域算法，使用了并查集方法，并了解了opencv的distancetransform的使用，对距离场的效果有了一定的了解 | | |