# 使用C++读写二进制数据文件

以下示例演示了使用C++读写二进制数据文件的方法。

以依次以一定半径在图像上绘制圆，将圆上点保存到向量中，最终将各半径及对应的圆上点保存到一个map<int, vector<Point>>类型的map中。由于数据量大，需要将其写入磁盘，下次需要用时直接读取即可。产生圆数据的代码如下：

void HandGestureRecognizer::genCirclePoints()

{

Point center(250, 250);

for (int r = minRadio; r < maxRadio; r++)

{

Mat tempMat(Size(1100, 1100), CV\_8UC1, Scalar(0));

vector<Point> pts;

circle(tempMat, center, r, Scalar(255), 1);

for (int i = 0; i < 1100; i++)

{

uchar \*pData = tempMat.ptr<uchar>(i);

for (int j = 0; j < 1100; j++)

{

if (pData[j] == 255)

{

pts.push\_back(Point(j - 250, i - 250));

}

}

}

circlePoints.insert(map<int, vector<Point>>::value\_type(r, pts));

}

}

将圆上数据以二进制文件的方式保存到磁盘，文件格式如下：

最小半径(int) 最大半径(int) 总序列个数（int)

半径序号(int) 元素个数（int) 元素1（Point) 元素2（Point)...

半径序号(int) 元素个数（int) 元素1（Point) 元素2（Point)...

…

注意，在文件中数据是连续存储的，没有分行，在这里分行是为了结构明确

bool HandGestureRecognizer::saveCirclePoints(string filename)

{

//最小半径(int) 最大半径(int) 总序列个数（int) 半径序号(int) 元素个数（int) 元素1（Point) 元素2（Point)... 半径序号(int) 元素个数（int) 元素1（Point) 元素2（Point)...

ofstream outfile(filename, ios::binary);

int minRadio = MINRADIO;

int maxRadio = MAXRADIO;

if (!outfile)

{

cout << "faild to save circle point file!" << endl;

return false;

}

//写入最小、最大半径

outfile.write((char\*)&minRadio, sizeof(int));

outfile.write((char\*)&maxRadio, sizeof(int));

//写入元素个数

int count = circlePoints.size();

outfile.write((char\*)&count, sizeof(int));

for (map<int, vector<Point>>::iterator it = circlePoints.begin(); it != circlePoints.end(); it++)

{

vector<Point> pts = it->second;

//写入半径序号

int r = it->first;

outfile.write((char\*)&r, sizeof(int));

//写入元素个数

int n = pts.size();

outfile.write((char\*)&n, sizeof(int));

//写入元素序列

for (int i = 0; i < n; i++)

{

Point pt = pts[i];

outfile.write((char\*)&pt, sizeof(Point));

}

}

outfile.close();

cout << "save circle point ok!" << endl;

return true;

}

读取文件：

bool HandGestureRecognizer::readCirclePoints(string filename)

{

//最小半径(int) 最大半径(int) 总序列个数（int) 半径序号(int) 元素个数（int) 元素1（Point) 元素2（Point)... 半径序号(int) 元素个数（int) 元素1（Point) 元素2（Point)...

ifstream infile(filename, ios::binary);

if (!infile)

{

cout << "faild to read circle point file!" << endl;

return false;

}

//读入最小最大半径

infile.read((char\*)&minRadio, sizeof(int));

infile.read((char\*)&maxRadio, sizeof(int));

//读入总序列个数

int count;

infile.read((char\*)&count, sizeof(int));

for (int i = 0; i < count; i++)

{

int r;

//半径序号

infile.read((char\*)&r, sizeof(int));

//元素个数

int n;

infile.read((char\*)&n, sizeof(int));

vector<Point> pts;

for (int j = 0; j < n; j++)

{

Point pt;

infile.read((char\*)&pt, sizeof(Point));

pts.push\_back(pt);

}

circlePoints.insert(map<int, vector<Point>>::value\_type(r,pts));

}

infile.close();

cout << "read circle point file ok!" << endl;

return true;

}