



会员

周边

新闻

博文

闪存

赞助商

Chat2DB

代码改变世界



注册

登录

StupidsCat

博客园

首页

新随笔

联系

订阅

管理

公告

昵称: StupidsCat  
园龄: 13年  
粉丝: 71  
关注: 0  
+加关注

<	2025年5月						>
日	一	二	三	四	五	六	
27	28	29	30	1	2	3	
4	5	6	7	8	9	10	
11	12	13	14	15	16	17	
18	19	20	21	22	23	24	
25	26	27	28	29	30	31	
1	2	3	4	5	6	7	

搜索

找我看

常用链接

- 我的随笔
- 我的评论
- 我的参与
- 最新评论
- 我的标签

我的标签

- 皮肤(4)
- 美化(4)
- sql 备份(2)
- QQ(2)
- C#(2)
- 控件(2)
- Textbox(1)
- SQL 监控(1)
- sql 创建数据库(1)
- SQL 监控 死锁 分析(1)
- 更多

随笔分类

- SQL(8)
- 编程(30)
- 网络(5)
- 网络资源(7)

随笔档案

- 2019年9月(1)
- 2017年7月(1)
- 2015年4月(2)

随笔 - 58 文章 - 0 评论 - 49 阅读 - 49万

OpenCvSharp 通过特征点匹配图片

现在的手游基本都是重复操作,一个动作要等好久,结束之后继续另一个动作.很麻烦,所以动起了自己写一个游戏辅助的心思.

这个辅助本身没什么难度,就是通过不断的截图,然后从这个截图中找出预先截好的能代表相应动作的按钮或者触发条件的小图.

找到之后获取该子区域的左上角坐标,然后通过windows API调用鼠标或者键盘做操作就行了.

这里面最难的也就是找图了,因为要精准找图,而且最好能适应不同的分辨率下找图,所以在模板匹配的基础上,就有了SIFT和SURF的特征点找图方式.

在写的过程中查找资料,大都是C++ 或者python的,很少有原生的C#实现,所以我就直接拿来翻译过来了(稍作改动).

SIFT算法

```
public Point2d Point2fToPoint2d(Point2f point) => new
Point2d((double)point.X, (double)point.Y);

public static Bitmap MatchPicBySift(Bitmap imgSrc, Bitmap imgSub)
{
    using (Mat matSrc = imgSrc.ToMat())
    using (Mat matTo = imgSub.ToMat())
    using (Mat matSrcRet = new Mat())
    using (Mat matToRet = new Mat())
    {
        KeyPoint[] keyPointsSrc, keyPointsTo;
        using (var sift = OpenCvSharp.XFeatures2D.SIFT.Create())
        {
            sift.DetectAndCompute(matSrc, null, out keyPointsSrc, out keyPointsTo);
            sift.DetectAndCompute(matTo, null, out keyPointsTo, out keyPointsSrc);
        }
        using (var bfMatcher = new OpenCvSharp.BFMatcher())
        {
            var matches = bfMatcher.KnnMatch(matSrcRet, matToRet, 3);

            var pointsSrc = new List<Point2f>();
            var pointsDst = new List<Point2f>();
            var goodMatches = new List<DMatch>();
            foreach (DMatch[] items in matches.Where(x =>
            {
                if (items[0].Distance < 0.5 * items[1].Distance)
                {
                    pointsSrc.Add(keyPointsSrc[items[0].QueryIdx]);
                    pointsDst.Add(keyPointsTo[items[0].TrainIdx]);
                    goodMatches.Add(items[0]);
                }
            })
            {
                Console.WriteLine($"{keyPointsSrc[items[0].QueryIdx].X} {keyPointsSrc[items[0].QueryIdx].Y} {keyPointsTo[items[0].TrainIdx].X} {keyPointsTo[items[0].TrainIdx].Y}");
            }
        }
    }
}
```

支持MCP的AI原生IDE

2013年11月(1)  
2013年8月(6)  
2013年7月(3)  
2013年6月(1)  
2013年5月(6)  
2013年4月(4)  
2013年3月(4)  
2013年2月(2)  
2013年1月(7)  
2012年12月(1)  
2012年11月(1)  
2012年10月(2)  
更多

### 阅读排行榜

1. Oracle\_Database\_11g\_标准版\_企业版\_\_下载地址\_详细列表(226551)
2. C# 给程序加日志功能。(27724)
3. 注册edu邮箱的办法(20838)
4. C# 给自己写的软件，加注册码功能。(19938)
5. C# 多线程、控制线程数提高循环输出效率(19116)

### 评论排行榜

1. C# 给自己写的软件，加注册码功能。(9)
2. OpenCvSharp 通过特征点匹配图片(8)
3. 获取本地 有线 正在使用的网卡信息(6)
4. C# 多线程控制 通讯 和切换(5)
5. 极简单的方式序列化sqlalchemy结果集为JSON(4)

### 推荐排行榜

1. OpenCvSharp 通过特征点匹配图片(7)
2. Oracle\_Database\_11g\_标准版\_企业版\_\_下载地址\_详细列表(7)
3. C# 使用栈遍历磁盘文件和目录，非传统递归方法。(3)
4. C# 多线程控制 通讯 和切换(3)
5. 通过C#发送自定义的html格式邮件(3)

### 最新评论

1. Re:C# 给自己写的软件，加注册码功能。  
@桦仔 可以分享一下demon吗? ...  
--i\_dume
2. Re:C# 给自己写的软件，加注册码功能。  
@i\_dume @桦仔 您好！实现该功能了吗？实现了...  
--桦仔
3. Re:C# 给自己写的软件，加注册码功能。  
@桦仔 您好！实现该功能了吗? ...

```
    }  
}  
  
var outMat = new Mat();  
  
// 算法RANSAC对匹配的结果做过滤  
var pSrc = pointsSrc.ConvertAll(Point2fToPoint2d);  
var pDst = pointsDst.ConvertAll(Point2fToPoint2d);  
var outMask = new Mat();  
// 如果原始的匹配结果为空，则跳过滤步骤  
if (pSrc.Count > 0 && pDst.Count > 0)  
    Cv2.FindHomography(pSrc, pDst, HomographyMethod.RANSAC, outMat, outMask);  
// 如果通过RANSAC处理后的匹配点大于10个，才应用过滤。  
if (outMask.Rows > 10)  
{  
    byte[] maskBytes = new byte[outMask.Rows * outMask.Cols];  
    outMask.GetArray(0, 0, maskBytes);  
    Cv2.DrawMatches(matSrc, keyPointsSrc, matDst, keyPointsDst, outMask, outMat);  
}  
else  
    Cv2.DrawMatches(matSrc, keyPointsSrc, matDst, keyPointsDst, outMask, outMat);  
return OpenCvSharp.Extensions.BitmapConverter.ToBitmap(outMat);  
}
```



### SURF算法



```
public static Point2d Point2fToPoint2d(Point2f point) => new  
Point2d((double)point.X, (double)point.Y);  
  
public static Bitmap MatchPicBySurf(Bitmap imgSrc, Bitmap imgSub,  
    {  
        using (Mat matSrc = imgSrc.ToMat())  
        using (Mat matTo = imgSub.ToMat())  
        using (Mat matSrcRet = new Mat())  
        using (Mat matToRet = new Mat())  
        {  
            KeyPoint[] keyPointsSrc, keyPointsTo;  
            using (var surf = OpenCvSharp.XFeatures2D.SURF.Create(1000))  
            {  
                surf.DetectAndCompute(matSrc, null, out keyPointsSrc, out matSrcRet);  
                surf.DetectAndCompute(matTo, null, out keyPointsTo, out matToRet);  
            }  
  
            using (var flnMatcher = new OpenCvSharp.FlannBasedMatcher())  
            {  
                var matches = flnMatcher.Match(matSrcRet, matToRet);  
                //求最小最大距离  
                double minDistance = 1000; //反向逼近  
                double maxDistance = 0;  
                for (int i = 0; i < matches.Length; i++)  
                {  
                    double distance = matches[i].Distance;  
                    if (distance < minDistance)  
                        minDistance = distance;  
                    if (distance > maxDistance)  
                        maxDistance = distance;  
                }  
            }  
        }  
    }  
}
```

支持MCP的AI原生IDE

--i\_dume

## 4. Re:OpenCvSharp 通过特征点匹配图片

有用么 这些算法 原图抠出来可以对起来 不

是原图都没用

不知道是自己的问题还是

--hongkong77

## 5. Re:C# 给自己写的软件, 加注册码功

能。

这个注册码没有时间限制?

--老翁钓大鱼

```

        maxDistance = distance;
    }
    if (distance < minDistance)
    {
        minDistance = distance;
    }
}
Console.WriteLine($"max distance : {maxDistance}");
Console.WriteLine($"min distance : {minDistance}");

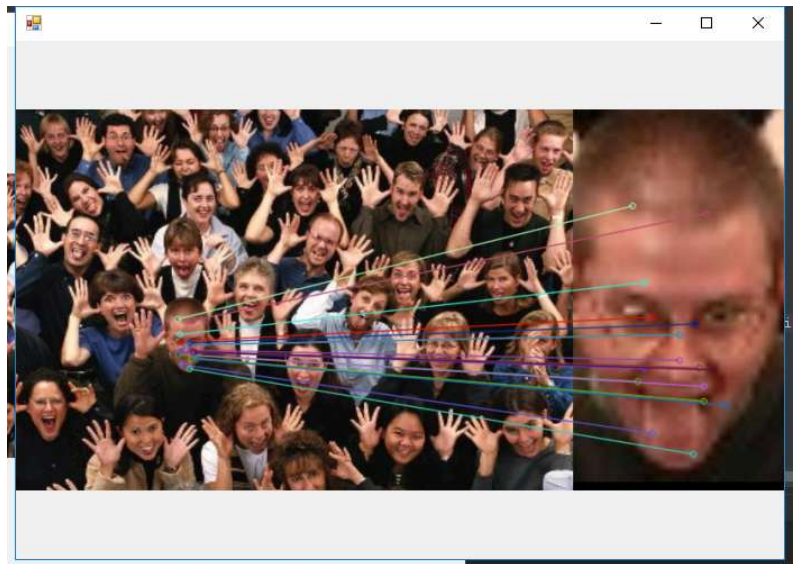
var pointsSrc = new List<Point2f>();
var pointsDst = new List<Point2f>();
//筛选较好的匹配点
var goodMatches = new List<DMatch>();
for (int i = 0; i < matSrcRet.Rows; i++)
{
    double distance = matches[i].Distance;
    if (distance < Math.Max(minDistance * 2, 0.01))
    {
        pointsSrc.Add(keyPointsSrc[matches[i].Idx1]);
        pointsDst.Add(keyPointsDst[matches[i].Idx2]);
        //距离小于范围的压入新的DMatch
        goodMatches.Add(matches[i]);
    }
}

var outMat = new Mat();

// 算法RANSAC对匹配的结果做过滤
var pSrc = pointsSrc.ConvertAll(Point2fToPoint2d);
var pDst = pointsDst.ConvertAll(Point2fToPoint2d);
var outMask = new Mat();
// 如果原始的匹配结果为空, 则跳过过滤步骤
if (pSrc.Count > 0 && pDst.Count > 0)
    Cv2.FindHomography(pSrc, pDst, HomographyMethod.RANSAC, outMask);
// 如果通过RANSAC处理后的匹配点大于10个, 才应用过滤。
if (outMask.Rows > 10)
{
    byte[] maskBytes = new byte[outMask.Rows * outMask.Cols];
    outMask.GetArray(0, 0, maskBytes);
    Cv2.DrawMatches(matSrc, keyPointsSrc, matDst, keyPointsDst, outMask, outMat);
}
else
    Cv2.DrawMatches(matSrc, keyPointsSrc, matDst, keyPointsDst, new Mat(), outMat);
return OpenCvSharp.Extensions.BitmapConverter.ToBitmap(outMat);
}
}
}

```





### 模板匹配

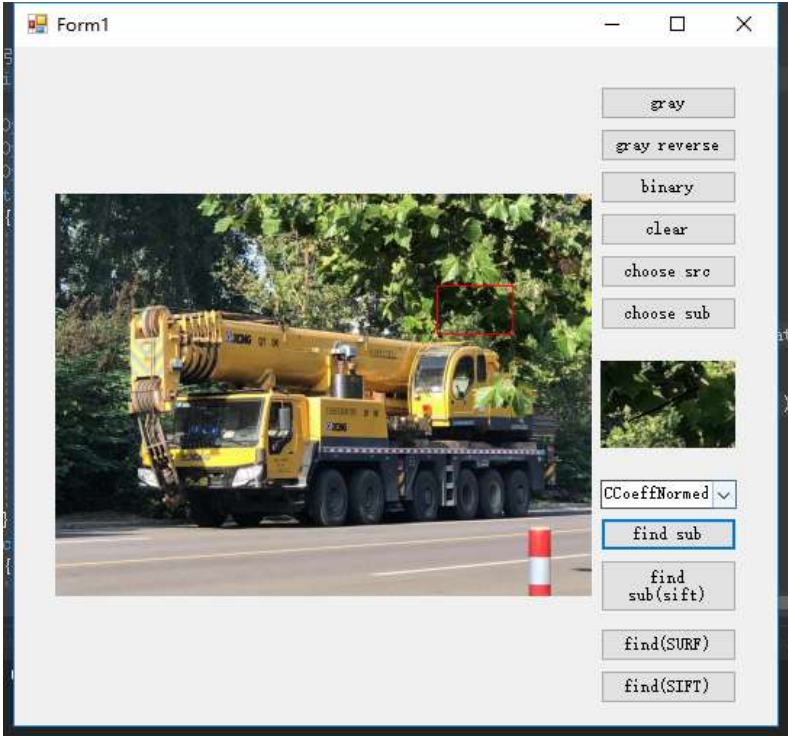


```
public static System.Drawing.Point FindPicFromImage(Bitmap imgSrc
{
    OpenCvSharp.Mat srcMat = null;
    OpenCvSharp.Mat dstMat = null;
    OpenCvSharp.OutputArray outArray = null;
    try
    {
        srcMat = imgSrc.ToMat();
        dstMat = imgSub.ToMat();
        outArray = OpenCvSharp.OutputArray.Create(srcMat);

        OpenCvSharp.Cv2.MatchTemplate(srcMat, dstMat, outArray,
            double minValue, maxValue;
        OpenCvSharp.Point location, point;
        OpenCvSharp.Cv2.MinMaxLoc(OpenCvSharp.InputArray.Create(outArray),
            Console.WriteLine(maxValue);
        if (maxValue >= threshold)
            return new System.Drawing.Point(point.X, point.Y);
        return System.Drawing.Point.Empty;
    }
    catch(Exception ex)
    {
        return System.Drawing.Point.Empty;
    }
    finally
    {
        if (srcMat != null)
            srcMat.Dispose();
        if (dstMat != null)
            dstMat.Dispose();
        if (outArray != null)
            outArray.Dispose();
    }
}
```



支持MCP的AI原生IDE



分类: [编程](#)

标签: [opencv](#), [找图](#), [匹配](#), [C#](#)



StupidsCat  
粉丝 - 71 关注 - 0

7

0

[+加关注](#)

[升级成为会员](#)

« 上一篇: [极简单的方式序列化sqlalchemy结果集为JSON](#)

posted @ 2019-09-03 15:01 StupidsCat 阅读(9667) 评论(8) 收藏 举报

[刷新页面](#) [返回顶部](#)

登录后才能查看或发表评论, 立即 [登录](#) 或者 [逛逛](#) 博客园首页

【推荐】新一代 Linux 服务器运维管理面板 1Panel V2 开放公测!

【推荐】100%开源! 大型工业跨平台软件C++源码提供, 建模, 组态!

【推荐】国内首个AI IDE, 深度理解中文开发场景, 立即下载体验Trae

【推荐】Flutter适配HarmonyOS 5知识地图, 实战解析+高频避坑指南

【推荐】轻量又高性能的 SSH 工具 IShell: AI 加持, 快人一步



支持MCP的AI原生IDE

**编辑推荐：**

- 聊一聊 C# NativeAOT 多平台下的函数导出
- 单一职责原则的思维：为什么你的代码总在"牵一发而动全身"
- 用纯 C# 打造零依赖的 .ico 图标生成器
- golang unsafe遇上字符串拼接优化导致的bug
- 浅谈.NET微服务架构的演变

**阅读排行：**

- 《HelloGitHub》第 110 期
- 重磅！SpringBoot4发布，11项重大变更全解析！
- DeepSeek R1再进化：这次更新让它直接对标Claude 4
- 后台服务器开发领域，还有什么值得爬的山
- 聊一聊 C# NativeAOT 多平台下的函数导出

博客园 © 2004-2025



浙公网安备 33010602011771号 浙ICP备2021040463号-3

支持MCP的AI原生IDE