

## xuchenglu的专栏

目录视图

摘要视图

RSS 订阅

## 个人资料



xuchenglu

访问: 12871次  
积分: 265  
等级:  2  
排名: 千里之外

原创: 11篇 转载: 5篇  
译文: 1篇 评论: 11条

## 文章搜索

## 文章分类

python (5)  
opencv (3)  
c++ (2)  
机器视觉 (5)  
android (1)  
数学基础知识 (1)

## 文章存档

2014年11月 (1)  
2014年10月 (1)  
2014年05月 (5)  
2014年04月 (4)  
2014年03月 (6)

## 阅读排行

TLD matlab 源代码阅读 (1933)  
TLD matlab源代码阅读 (1295)  
Android 蓝牙串口服务客 (1138)  
opencv2.4.8参考手册 (1125)  
网页收藏小工具 (一) -> (1022)  
物体跟踪meanshift详解 (785)  
为什么不去读顶级会议上 (670)  
物体跟踪meanshift详解 (658)  
网页收藏小工具 (二) -> (652)

【观点】世界上最好的语言是什么 【知识库】50个精品领域内容，一键直达 晒知识图谱，享技术荣誉

## TLD matlab 源代码阅读 ( 1 )

标签: 计算机视觉 matlab tld

2014-05-17 18:12 1933人阅读 评论(3) 收藏 举报

分类: 机器视觉 (4)

版权声明：本文为博主原创文章，未经博主允许不得转载。

最近在看视觉跟踪方面的论文，ZK博士的TLD算法作为跟踪算法的state-of-the-art，当然不得不去拜读下了，看完论文后虽然对作者整体的思想有了一个大致了解，但是对于很多细节却也还是无从得知，好在作者将自己的算法源代码全部开源，这也造福了我们这些无知者的胃口，虽然网上有几个c++版本的源码，但是matlab版本作为作者的原始版本，拿来作研究也是极好的。

通览源代码，个人感觉精髓之处无非两个函数，即：

tldInit();

tldProcessFrame();

这二者一个是用来跟踪前的初始化工作，后者当然是run-time的跟踪工作。我个人的阅读注释也将从这两个函数展开。好了，闲话少叙，下面来看看tldInit()这个函数：

进去第一个函数是

```
[plain]
01. %输入:
02. %tld.source.bb 用户输入的目标标定框
03. %size(tld.source.im0.input) 输入图像的尺寸
04. %tld.model.min_win 目标标定框的长或宽的最小尺寸
05. %输出:
06. %tld.grid 是一个6Xn(n表示不同有效尺度下共有多少个的网格)的矩阵，前四行组成列向量表示gridbox的4个顶点，5表示索引，6表示横向的分布点数
07. % tld.scales 有效尺度下gridbox的高和宽，为2Xm矩阵 (m表示有效的尺度的个数)
08. [tld.grid tld.scales] = bb_scan(tld.source.bb,size(tld.source.im0.input),tld.model.min_win);
```

再让我们继续进入bb\_scan()函数：

```
[plain]
01. bbw = round(bb_width(bb) .* SCALE);bbw_width(bb)*(0.1615~6.1917) 矩形框变尺度
02. bbh = round(bb_height(bb) .* SCALE);
03. bbSHH = SHIFT * min(bbh,bbH);%取bbH和bbH中最小的元素组成新的矩阵，维数和原来相同 <span style="font-family: Arial, Helvetica, sans-serif;">SHIFT =</span>0.1表示变形后的矩形框的移动步长
04. bbSHW = SHIFT * min(bbh,bbW);
05.
06. bbF = [2 2 imsize(2) imsize(1)]';%[2 2 320 240]
07.
08. bbs = {};
09. sca = [];
10. idx = 1;
11.
12. for i = 1:length(SCALE)
13.     if bbw(i) < MINBB || bbh(i) < MINBB, continue; end
14.
```

## 评论排行

TLD matlab源代码阅读 (2)	(4)
TLD matlab 源代码阅读	(3)
物体跟踪meanshift详解之	(3)
物体跟踪meanshift详解之	(2)
始于足下 c++ references	(0)
始于足下 c++ exceptions	(0)
opencv2.4.8参考手册 (一)	(0)
Python字符编码详解	(0)
协方差	(0)
网页收藏小工具 (三) -I	(0)

## 推荐文章

- \* 2016 年最受欢迎的编程语言是什么？
- \* Chromium扩展（Extension）的页面（Page）加载过程分析
- \* Android Studio 2.2 来啦
- \* 手把手教你做音乐播放器（二）技术原理与框架设计
- \* JVM 性能调优实战之：使用阿里开源工具 TProfiler 在海量业务代码中精确定位性能代码

## 最新评论

- TLD matlab源代码阅读 (2)  
qinchaohit: 博主，有带有注释的完整matlab源代码吗？可以麻烦发一份给我吗？邮箱：289414090@qq.c...
- TLD matlab 源代码阅读 (1)  
yanbohaha: 楼主你好，能不能把注释之后的代码发给我，谢谢 1394903938@qq.com
- 物体跟踪meanshift详解之密度估  
you\_and\_007: 楼主，我有个问题想请教下，在公式：Ps(Yu) = (s(b(x0)-Yu) + s(b(x1)...
- TLD matlab源代码阅读 (2)  
xia316104: 最近在学习TLD，有源代码MATLAB版的，能否给我也发份，邮箱 973496253@qq.com
- TLD matlab 源代码阅读 (1)  
qq\_20981151: 作者matlab版本的程序需要怎么配置才能运行
- TLD matlab源代码阅读 (2)  
我的小和尚: 给博主点个赞，帮助很大呀，能否也给我发一份带有注释的matlab源代码，邮箱 1033032785@...
- TLD matlab源代码阅读 (2)  
caiyuzhu001: 博主，能否给我发一份带有注释的matlab源代码！感谢 邮箱：657967280@qq.com...
- TLD matlab 源代码阅读 (1)  
handsomewangggg: 您好，把你阅读程序时用的参考文献分享一下吗？我这只只有Zdenek Kalal的几篇文章，感觉太少了...
- 物体跟踪meanshift详解之相似度  
qykshr: 文章很好，只是公式写的不太严谨，比如，有些地方应该用Ps代替Psn. 还有如果r(s)的定义是(1)...
- 物体跟踪meanshift详解之密度估  
qykshr: 顶

```

15.     left  = round(bbF(1):bbSHW(i):bbF(3)-bbW(i)-1);%2开头，0.1* min(bbH(i),bbW(i))为步长，320- bbW
(i)-1为结尾
16.     top   = round(bbF(2):bbSHH(i):bbF(4)-bbH(i)-1);%2开头，0.1* min(bbH(i),bbH(i))为步长，240- bbH
(i)-1为结尾
17.
18.     grid  = ntuples(top,left);%重复功能，维度为2 X length(top)*length(left);grid的每个列向量成为网格
顶点的坐标点
19.     if isempty(grid), continue; end
20.
21.     bbs(end+1) = [grid(2,:); ...%省略号为续行，bbs前四行的列向量构成一个矩形网格的四个顶点
22.                 grid(1,:); ...
23.                 grid(2,:)+bbW(i)-1; ...
24.                 grid(1,:)+bbH(i)-1; ...
25.                 idx*ones(1,size(grid,2));%坐标点对应的编号，从1开始
26.                 length(left)*ones(1,size(grid,2));];%记录横向分布多少个点？
27.     sca   = [sca [bbH(i); bbW(i)]]; %在原来SCA的基础上增加[bbH(i); bbW(i)]列向量，该列向量表示某个尺度因
子下的gridbox的高和宽
28.     idx  = idx + 1;
29. end
30. bb_out = [];
31. for i = 1:length(bbs)
32.     bb_out = [bb_out bbs{i}];
33. end

```

总之，该函数功能就是扫描输入图像得到很多的gridbox。

下面来到了产生特征的函数

```

[plain]
01. %输入：
02. %tld.model.num_trees: 有10棵分类树
03. %tld.model.num_features: 每棵树里有13个特性，这里是会有13个点对的比较
04. %输出：
05. %tld.features: 4*13X10的矩阵，为fern分类器选取的随机点对，4表示一个点对的四个坐标值
06. tld.features = tldGenerateFeatures(tld.model.num_trees,tld.model.num_features,1);

```

还是进入一看究竟：

```

[plain]
01. SHI = 1/5;
02. SCA = 1;
03. OFF = SHI;
04.
05. x = repmat(ntuples(0:SHI:1,0:SHI:1),2,1);%ntuples生成网格点坐标2X36矩阵，repmat表示将生成的矩阵作为
初始化元素(还是矩阵)
06. %生成一个2X1的大矩阵，这里是4X36(6*6)矩阵
07. x = [x x + SHI/2];%前者4X36矩阵，后者矩阵偏移0.1
08. k = size(x,2);%k = 36*2
09. %rand(1,k)随机生成都1Xk维随机分布矩阵，范围0~1
10. r = x; r(3,:) = r(3,:) + (SCA*rand(1,k)+OFF);%x的第3行加上随机的1X72矩阵(0~1)再加上0.2
11. l = x; l(3,:) = l(3,:) - (SCA*rand(1,k)+OFF);%x的第3行减去随机的1X72矩阵(0~1)再减去0.2
12. t = x; t(4,:) = t(4,:) - (SCA*rand(1,k)+OFF);%x的第4行减去随机的1X72矩阵(0~1)再减去0.2
13. b = x; b(4,:) = b(4,:) + (SCA*rand(1,k)+OFF);%x的第4行加上随机的1X72矩阵(0~1)再加上0.2
14.
15. x = [r l t b];
16.
17. idx = all(x([1 2],:) < 1 & x([1 2],:) > 0,1);%idx和X维数相同，如果满足条件相应元素为1，否则为0
18. x = x(:,idx);%挑选满足条件的所有X的列
19. x(x > 1) = 1;%x元素中大于1的则用1代替此元素
20. x(x < 0) = 0;%x元素中小于0的则用0代替此元素
21.
22. numF = size(x,2);%看看现在X还有多少列
23.
24. x = x(:,randperm(numF));%randperm(numF)从1~numF的数字序列随机打乱，整个表达式是指把X的列打乱
25. x = x(:,1:nFEAT*nTREES);%取x的前130列，x目前为4x130矩阵
26. x = reshape(x,4*nFEAT,nTREES);%把X塑造成4*13 X 10 的矩阵

```

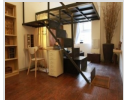
整个函数就是生成10\*13个特征点对，坐标用小数表示，表示在box的相对位置，至于为什么随机我也感觉挺困惑的，求解释啊！！

下面再来到fern(0)：没什么好说的，来看下面的吧：

```

[plain]
01. %输入：
02. %tld.source.im0.input: 输入的图像

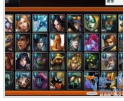
```



青年公寓租房



酒店式公寓月租



lol美服代练



澳大利亚移民条

```

03. %tld.grid: 6Xn(n表示不同有效尺度下共有多少个的网格)的矩阵, gridbox信息
04. %tld.features: 4*13X10的矩阵, 为fern分类器选取的随机点对, 4表示一个点对的四个坐标值
05. %tld.scales: 有效尺度下gridbox的高和宽, 为2Xm矩阵 (m表示有效的尺度的个数)
06. %输出:
07. fern(1,tld.source.im0.input,tld.grid,tld.features,tld.scales); % allocate structures

```

进入CPP的fern

```

[cpp]
01. iHEIGHT = mxGetM(prhs[1]); //240
02. iWIDTH = mxGetN(prhs[1]); //320
03. nTREES = mxGetN(mxGetField(prhs[3],0,"x")); //<span style="font-family: Arial, Helvetica, sans-serif;">10</span>
04. nFEAT = mxGetM(mxGetField(prhs[3],0,"x")) / 4; // feature has 2 points: x1,y1,x2,y2 13
05. thrN = 0.5 * nTREES; //等于5
06. nSCALE = mxGetN(prhs[4]); //获得共有多少种尺度的BOX,即有效尺度的个数
07.
08. IIMG = (double*) malloc(iHEIGHT*iWIDTH*sizeof(double)); //积分图像变量准备
09. IIMG2 = (double*) malloc(iHEIGHT*iWIDTH*sizeof(double)); //平方积分图像变量准备
10.
11. // BBOX
12. mBBOX = mxGetM(prhs[2]); //等于6
13. nBBOX = mxGetN(prhs[2]); //等于在各个有效尺度下有多少个网格
14. BBOX = create_offsets_bbox(mxGetPr(prhs[2])); //创建保存网格数据索引等数据
15. //prhs[3]是4*13X10的矩阵
16. //matlab 代码中有f.x = x; f.type = 'forest'; 见到下面就不怪了
17. double *x = mxGetPr(mxGetField(prhs[3],0,"x")); //获得特征点的指针
18. double *s = mxGetPr(prhs[4]); //各种尺度的BOX的尺寸
19. OFF = create_offsets(s,x); //记录各个特征点对在各种尺度下box中的具体位置
20.
21. for (int i = 0; i < nTREES; i++) {
22.     WEIGHT.push_back(vector<double>(pow(2.0,nBIT*nFEAT), 0)); //nBIT=1,权重分配权2^13
23.     nP.push_back(vector<int>(pow(2.0,nBIT*nFEAT), 0)); //nBIT=1,nP分配2^13
24.     nN.push_back(vector<int>(pow(2.0,nBIT*nFEAT), 0)); //nBIT=1,nN分配2^13
25. }
26. //static vector<vector<double>> WEIGHT;
27. //static vector<vector<int>> nP;
28. //static vector<vector<int>> nN; 下面见怪不怪, 10X 2^13的容器
29. for (int i = 0; i < nTREES; i++) {
30.     for (int j = 0; j < WEIGHT[i].size(); j++) {
31.         WEIGHT[i].at(j) = 0;
32.         nP[i].at(j) = 0;
33.         nN[i].at(j) = 0;
34.     }
35. }

```

来具体看看create\_offset\_bbox和creater\_offset两个函数

```

[cpp]
01. int *offsets = (int*) malloc(BBOX_STEP*nBBOX*sizeof(int)); //7*nBBOX*sizeof(int)
02. int *off = offsets;
03.
04. for (int i = 0; i < nBBOX; i++) { //nBBOX表示所有的网格数
05.     double *bb = bb0+mBBOX*i; //偏移到下一个网格的属性的向量
06.     //sub2idx(row,col,height) ((int) (floor((row)+0.5) + floor((col)+0.5)*(height)))
07.     //floor不大于
08.     //bb[0]:left bb[1]:top bb[2]:right bb[3]:bottom iHEIGHT:240 bb[4]索引从1开始 bb[5]表示
    number of left-right bboxes
09.     //以下记录索引是从左到右, 从上到下方式
10.     *off++ = sub2idx(bb[1]-1,bb[0]-1,iHEIGHT); //左上顶点索引
11.     *off++ = sub2idx(bb[3]-1,bb[0]-1,iHEIGHT); //左下顶点索引
12.     *off++ = sub2idx(bb[1]-1,bb[2]-1,iHEIGHT); //右上顶点索引
13.     *off++ = sub2idx(bb[3]-1,bb[2]-1,iHEIGHT); //右下顶点索引
14.     *off++ = (int) ((bb[2]-bb[0])*(bb[3]-bb[1])); //记录当前网格的大小
15.     *off++ = (int) (bb[4]-1)*2*nFEAT*nTREES; // pointer to features for this scale
16.     *off++ = bb[5]; // number of left-right bboxes, will be used for searching neighbours
17. }
18. return offsets;

```

```

[cpp]
01. int *offsets = (int*) malloc(nSCALE*nTREES*nFEAT*2*sizeof(int)); //
02. int *off = offsets;
03.

```

```
04. for (int k = 0; k < nSCALE; k++){//共有多少种尺度的BOX
05.     double *scale = scale0+2*k;//scale0第一种尺度的尺寸信息，*2表示列向量有2维，即高和宽，此表达式表示
        偏移到下一个box的尺寸向量
06.     for (int i = 0; i < nTREES; i++) { //10
07.         for (int j = 0; j < nFEAT; j++) { //13
08.             //x0 4*13 X 10 的矩阵
09.             double *x = x0 +4*j + (4*nFEAT)*i;//4*j因为每个feature是一个4维列向量，(4*13)*i即下一
        棵树
10.             //sub2idx(row,col,height) ((int) (floor((row)+0.5) + floor((col)+0.5)*
        (height)))
11.             //scale[1]宽，scale[0]高,x[0]x坐标，x[1]y坐标
12.             *off++ = sub2idx((scale[0]-1)*x[1],(scale[1]-1)*x[0],iHEIGHT);//记录第一个点在该尺度
        BOX的具体位置，并转化为索引
13.             *off++ = sub2idx((scale[0]-1)*x[3],(scale[1]-1)*x[2],iHEIGHT);//记录第二个点在该尺度
        BOX的具体位置，并转化为索引
14.         }
15.     }
16. }
```

好了，至此特征点的选取和gridbox的初始化已经完成了。

顶 踩  
2 0

上一篇 [计算机视觉、模式识别、机器学习常用牛人主页链接](#)

下一篇 [TLD matlab源代码阅读（2）](#)

### 我的同类文章

#### 机器视觉（4）

- [为什么不去读顶级会议上的...](#) 2014-05-19 阅读 670
- [TLD matlab源代码阅读（2）](#) 2014-05-18 阅读 1295
- [计算机视觉、模式识别、机...](#) 2014-05-10 阅读 590
- [计算机视觉、机器学习相关...](#) 2014-05-10 阅读 428

### 参考知识库

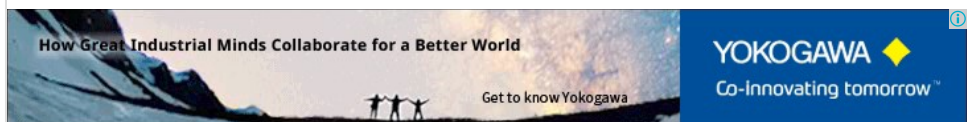


#### 算法与数据结构知识库

904 关注 | 2080 收录

### 猜你在找

- |                                      |                                    |
|--------------------------------------|------------------------------------|
| C语言系列之 递归算法示例与 Windows 趣味小项目         | 他人总结的TLD视觉跟踪算法源代码文章原理等非常好的         |
| 《C语言/C++学习指南》加密解密篇（安全相关算法）           | 运行matlab版的TLD算法                    |
| Windows Server 2012 DHCP Server 管理   | MAC OX 108 环境下运行TLD算法MATLAB版       |
| iOS移动开发从入门到精通(Xcode7 & Swift2)       | matlab2013a vs2013 opencv248 编译TLD |
| 精通iOS移动开发(Xcode7&Swift2;) 初识Xcode7.0 | MATLAB2010a+OpenCV231+VS2010运行TLD  |



#### 查看评论

3楼 [yanbohaha](#) 2015-10-22 19:17发表



楼主你好，能不能把注释之后的代码发给我，谢谢1394903938@qq.com

2楼 [qq\\_20981151](#) 2015-04-17 09:06发表

作者matlab版本的程序需要怎么配置才能运行



1楼 [handsomewangggg](#) 2014-12-29 09:38发表



您好，能把您阅读程序时用的参考文献分享一下吗？我这只有Zdenek Kalal的几篇文章，感觉太少了，理解不了整个实现过程

您还没有登录,请[\[登录\]](#)或[\[注册\]](#)

\* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

#### 核心技术类目

[全部主题](#) [Hadoop](#) [AWS](#) [移动游戏](#) [Java](#) [Android](#) [iOS](#) [Swift](#) [智能硬件](#) [Docker](#) [OpenStack](#)  
[VPN](#) [Spark](#) [ERP](#) [IE10](#) [Eclipse](#) [CRM](#) [JavaScript](#) [数据库](#) [Ubuntu](#) [NFC](#) [WAP](#) [jQuery](#)  
[BI](#) [HTML5](#) [Spring](#) [Apache](#) [.NET](#) [API](#) [HTML](#) [SDK](#) [IIS](#) [Fedora](#) [XML](#) [LBS](#) [Unity](#)  
[Splashtop](#) [UML](#) [components](#) [Windows Mobile](#) [Rails](#) [QEMU](#) [KDE](#) [Cassandra](#) [CloudStack](#) [FTC](#)  
[coremail](#) [OPhone](#) [CouchBase](#) [云计算](#) [iOS6](#) [Rackspace](#) [Web App](#) [SpringSide](#) [Maemo](#)  
[Compuware](#) [大数据](#) [aptech](#) [Perl](#) [Tornado](#) [Ruby](#) [Hibernate](#) [ThinkPHP](#) [HBase](#) [Pure](#) [Solr](#)  
[Angular](#) [Cloud Foundry](#) [Redis](#) [Scala](#) [Django](#) [Bootstrap](#)

[公司简介](#) | [招贤纳士](#) | [广告服务](#) | [银行汇款帐号](#) | [联系方式](#) | [版权声明](#) | [法律顾问](#) | [问题报告](#) | [合作伙伴](#) | [论坛反馈](#)

[网站客服](#) [杂志客服](#) [微博客服](#) [webmaster@csdn.net](#) 400-600-2320 | 北京创新乐知信息技术有限公司 版权所有 | 江苏知之为计算机有限公司 |

江苏乐知网络技术有限公司

京 ICP 证 09002463 号 | Copyright © 1999-2016, CSDN.NET, All Rights Reserved 