

## xuchenglu的专栏

目录视图

摘要视图

RSS 订阅

## 个人资料



xuchenglu

访问: 12873次  
积分: 265  
等级: 千里之外  
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 ( 2 )

标签: 计算机视觉 tld matlab 源代码

2014-05-18 20:21 1295人阅读 评论(4) 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。

今天继续，下面是开始要生成正负例来训练分类器了，首先：

```
[cpp]
01. // TRAIN DETECTOR =====
02.
03. // Initialize structures
04. tld.imgsize = size(tld.source.im0.input);
05. //为fern准备的训练集
06. tld.X = cell(1,length(tld.source.idx)); //training data for fern
07. tld.Y = cell(1,length(tld.source.idx));
08. %为nearest neighbor准备的训练集
09. tld.pEx = cell(1,length(tld.source.idx)); // training data for NN
10. tld.nEx = cell(1,length(tld.source.idx));
11. //输入:
12. //tld.source.bb: 用户目标标定框
13. //tld.grid: 生成的gridbox信息矩阵
14. //输出:
15. // overlap一维向量，记录GRID中的各个gridbox与用户目标标定框的重叠率
16. overlap = bb_overlap(tld.source.bb,tld.grid);
```

进入bb\_overlap来看一下：

```
[cpp]
01. // Input
02. double *bb1 = mxGetPr(prhs[0]); int M1 = mxGetM(prhs[0]); int N1 = mxGetN(prhs[0]); //4X1
03. double *bb2 = mxGetPr(prhs[1]); int M2 = mxGetM(prhs[1]); int N2 = mxGetN(prhs[1]); //6Xn(n表示
gridbox总数)
04.
05. // Output
06.
07. if (N1 == 0 || N2 == 0) {
08.     N1 = 0; N2 = 0;
09. }
10. plhs[0] = mxCreateDoubleMatrix(N1, N2, mxREAL); //创建输出矩阵，1Xgridbox的数量
11. double *out = mxGetPr(plhs[0]);
12.
13. for (int j = 0; j < N2; j++) { //gridbox的数量
14.     for (int i = 0; i < N1; i++) { //1
15.         *out++ = bb_overlap(bb1 + M1*i, bb2 + M2*j); //计算重叠度
16.     }
17. }

[cpp]
01. double bb_overlap(double *bb1, double *bb2) {
02.
03.     if (bb1[0] > bb2[2]) { return 0.0; } //判断如果两个矩形没有相交部分，重叠度就为0;
04.     if (bb1[1] > bb2[3]) { return 0.0; }
05.     if (bb1[2] < bb2[0]) { return 0.0; }
```

## 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 (-	(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)  
qinchaoht: 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，能把你阅读程序时用的参考文献分享一下吗？我只有Zdenek Kalal的几篇文章，感觉太少了...
- 物体跟踪meanshift详解之相似度qyshr: 文章很好，只是公式写的不太严谨，比如，有些地方应用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qyshr: 顶

```

06.         if (bb1[3] < bb2[1]) { return 0.0; }
07.
08.         double colInt = min(bb1[2], bb2[2]) - max(bb1[0], bb2[0]) + 1; //求相交矩形的宽和高
09.         double rowInt = min(bb1[3], bb2[3]) - max(bb1[1], bb2[1]) + 1;
10.
11.         double intersection = colInt * rowInt; //相交面积
12.         double area1 = (bb1[2]-bb1[0]+1)*(bb1[3]-bb1[1]+1); //分别求两个输入矩形的面积
13.         double area2 = (bb2[2]-bb2[0]+1)*(bb2[3]-bb2[1]+1);
14.         return intersection / (area1 + area2 - intersection); //求重叠率
15.     }

```

再接着

```

[cpp]
01. //输入:
02. //tld.img{1}.input: 输入图像，第一帧
03. //tld.bb(:,1): 用户目标标定框
04. //输出:
05. //tld.target: 目标标定框中特定的图像
06. tld.target = img_patch(tld.img{1}.input,tld.bb(:,1));

```

进入img\_patch,这个函数比较庞大，先看其中用到的一部分：

```

[cpp]
01. //如果4个坐标值都是整数
02. if sum(abs(round(bb)-bb))==0
03.     L = max([1 bb(1)]);
04.     T = max([1 bb(2)]);
05.     R = min([size(img,2) bb(3)]);
06.     B = min([size(img,1) bb(4)]);
07.     patch = img(T:B,L:R); //在不超过画面尺寸和小于1x1的情况下，取出BB框出的画面
08.
09.     % Sub-pixel accuracy
10. else
11.
12.     cp = 0.5 * [bb(1)+bb(3); bb(2)+bb(4)]-1; //bbox的中心坐标 center point
13.     %[1 0 -cp(1)]
14.     %[0 1 -cp(2)]
15.     %[0 0 1 ]
16.     H = [1 0 -cp(1); 0 1 -cp(2); 0 0 1];
17.
18.     bbw = bb(3,:)-bb(1,:); //宽
19.     bbH = bb(4,:)-bb(2,:); //高
20.     if bbw <= 0 || bbH <= 0
21.         patch = [];
22.         return;
23.     end
24.     box = [-bbw/2 bbw/2 -bbH/2 bbH/2];
25.
26.     if size(img,3) == 3 //如果图像有三个通道，即判断图片是否为真彩色
27.         for i = 1:3
28.             P = warp(img(:, :, i), inv(H), box);
29.             patch(:, :, i) = uint8(P);
30.         end
31.     else
32.         patch = warp(img, inv(H), box); //inv(H)=[1 0 cp(1); 0 1 cp(2); 0 0 1]; 平移变换
33.         patch = uint8(patch);
34.     end
35.
36. end

```

上面的函数功能就是对BB区域的图像提取，但是有针对坐标为整数和小数的处理，这里应该只用到整数部分，但至于小数坐标的处理跟踪了一下代码，发现是对图像作了平移的仿射变换，但是至于为什么要这么做，我也不理解，感觉直接舍去小数部分问题应该也不大吧（个人理解，没有看懂）。

好了下面开始产生正训练样本了：

```

[cpp]
01. //输入:
02. //overlap: 一维行向量，记录GRID中的各个gridbox与用户目标标定框的重叠率
03. //tld.p_par_init: opt.p_par_init= struct
04. ('num_closest',10,'num_warps',20,'noise',5,'angle',20,'shift',0.02,'scale',0.02);
//输出:

```

```

05. //pX:10 X length(idxP)*20 (length(idxP)<=10, 20为'num_warps',20)的矩阵列向量表示一个gridbox的10棵
    树上的13位有效的code
06. //pEx: 225X1的列向量,各元素值为原像素值减去像素均值
07. //bbP: 最靠近BBOX的的gridbox,列向量表示该box的4个顶点
08. [pX,pEx,bbP] = tldGeneratePositiveData(tld,overlap,tld.img{1},tld.p_par_init);
09. pY = ones(1,size(pX,2));%1 X length(idxP)*20

```

这个函数也是比较大的,但是还要耐心的往下看啊

```

[cpp]
01. pX = [];
02. pEx = [];
03.
04. // Get closest bbox
05. [~,idxP] = max(overlap);%表示行不管,只取列,整个表达式表示最大overlap<span style="font-
    family: Arial, Helvetica, sans-serif;">所对应的列,一维</span>
06. bbP0 = tld.grid(1:4,idxP);%1~4表示矩阵的4个顶点分布在四行,此取最靠近BBOX的的gridbox
07.
08. // Get overlapping bboxes
09. idxP = find(overlap > 0.6);%返回overlap > 0.6所对应的列索引
10. if length(idxP) > p_par.num_closest;%如果overlap > 0.6的gridbox数大于10
11.     [~,sIdx] = sort(overlap(idxP),'descend'); //降序排序
12.     idxP = idxP(sIdx(1:p_par.num_closest));%取前p_par.num_closest个最大重叠度的bboxes所在的列
13. end
14. bbP = tld.grid(:,idxP);%取出10个最大重叠度的gridboxes
15. if isempty(bbP), return; end
16.
17. % Get hull
18. bbH = bb_hull(bbP);%得到能包围所有bbp中boxes<span style="font-family: Arial, Helvetica, sans-
    serif;">的最小矩形</span>
19. cols = bbH(1):bbH(3);
20. rows = bbH(2):bbH(4);
21.
22. im1 = im0;
23. //返回一个225x1(pEx)的列向量,各元素值为原像素值减去像素均值
24. pEx = tldGetPattern(im1,bbP0,tld.model.patchsize);%
25. if tld.model.fliplr
26.     pEx = [pEx tldGetPattern(im1,bbP0,tld.model.patchsize,1)];
27. end
28. //返回20个正例
29. for i = 1:p_par.num_warps;%p_par.num_warps=20
30.     if i > 1
31.         randomize = rand; % Sets the internal randomizer to the same state
32.         %patch_input = img_patch(im0.input,bbH,randomize,p_par);
33.         //返回将画面进行仿射变换后的patch
34.         patch_blur = img_patch(im0.blur,bbH,randomize,p_par);%bbH包围所有bbp中bboxes的最小矩形
35.         //这个很重要,保证在C调用里的偏移的起始地址可以是一样的
36.         im1.blur(rows,cols) = patch_blur;%把仿射变换后的图像放到原图像对应的位置(<span style="font-
            family: Arial, Helvetica, sans-serif;">能包围所有bbp中boxes</span><span style="font-
            family: Arial, Helvetica, sans-serif;">的最小矩形</span>)
37.         %im1.input(rows,cols) = patch_input;
38.     end
39.
40.     // Measures on blurred image
41.     //单次返回10Xlength(idxP)的矩阵,列向量表示一个gridbox的10棵树上的13位code,
42.     //最后返回10Xlength(idxP)*20的矩阵
43.     pX = [pX fern(5,im1,idxP,0)];%idxP : overlap > 0.6所对应的列索引
44.
45.     // Measures on input image
46.     %pEx(:,i) = tldGetPattern(im1,bbP0,tld.model.patchsize);
47.     %pEx = [pEx tldGetPattern(im1,tld.grid(1:4,idxP),tld.model.patchsize)];
48.
49. end

```

当然这个函数是不能这么草草了事的,还有三大函数需要进一步细看:

#### 1.tldGetPattern()

```

[cpp]
01. nBB = size(bb,2);%得到bbp0(最靠近BBOX的gridbox)的列,值为1
02. pattern = zeros(prod(patchsize),nBB);%15*15 X 1 矩阵,返回矩阵
03. if ~exist('flip','var')
04.     flip= 0;
05. end
06.
07. // for every bounding box

```

```

08. for i = 1:nBB//1
09.
10.     // sample patch
11.     patch = img_patch(img.input,bb(:,i));//取出对应框中的图像
12.
13.     // flip if needed
14.     if flip
15.         patch = fliplr(patch);
16.     end
17.
18.     // normalize size to 'patchsize' and nomalize intensities to ZMUV
19.     //返回一个225x1的列向量,各元素值为原像素值减去像素均值
20.     pattern(:,i) = tldPatch2Pattern(patch,patchsize);//patch压缩变换到patchsize大小,然后将各个元素
    减去元素均值
21. end

```

切入到tldPatch2Pattern看一眼:

```

[cpp]
01. patch = imresize(patch,patchsize); // 'bilinear' is faster
02. pattern = double(patch(:));//此时变成225X1的矩阵
03. pattern = pattern - mean(pattern);//mean(pattern)求各列向量的均值

```

2.img\_patch()(4个传参)

```

[cpp]
01. rand('state',randomize);
02. randn('state',randomize);
03. // 'noise',5,'angle',20,'shift',0.02,'scale',0.02;
04. NOISE = p_par.noise;
05. ANGLE = p_par.angle;
06. SCALE = p_par.scale;
07. SHIFT = p_par.shift;
08.
09. cp = bb_center(bb)-1;//HULL矩形的中心
10. Sh1 = [1 0 -cp(1); 0 1 -cp(2); 0 0 1];
11.
12. sca = 1-SCALE*(rand-0.5);%0.99~1.01
13. //[0.99~1.01      ]
14. //[      0.99~1.01      ]
15. //[      1      ]
16. Sca = diag([sca sca 1]);
17.
18. ang = 2*pi/360*ANGLE*(rand-0.5);//-10 ~ 10度 实际为弧度
19. ca = cos(ang);
20. sa = sin(ang);
21. Ang = [ca, -sa; sa, ca];
22. Ang(end+1,end+1) = 1;
23.
24. shR = SHIFT*bb_height(bb)*(rand-0.5);//-0.01~1.01*bb_height(bb)
25. shC = SHIFT*bb_width(bb)*(rand-0.5);//-0.01~1.01*bb_width(bb)
26. Sh2 = [1 0 shC; 0 1 shR; 0 0 1];
27.
28. bbw = bb_width(bb)-1;
29. bbh = bb_height(bb)-1;
30. box = [-bbw/2 bbw/2 -bbh/2 bbh/2];
31.
32. H = Sh2*Ang*Sca*Sh1;
33. bbsize = bb_size(bb);
34. patch = uint8(warp(img,inv(H),box) + NOISE*randn(bbsize(1),bbsize(2)));//给图像造成5的高斯噪声

```

以上的代码注释就少了,因为全都是关于仿射变换的,具体可以参看仿射变换,大体就是作者在论文中提到的(shift+~1%, scale +~1%, in-plane rotation +~10度)用来提高训练样本的多样性。

3.fern () (第一个传参为5,获得模式)

```

[cpp]
01. unsigned char *input = (unsigned char*) mxGetPr(mxGetField(prhs[1],0,"input"));
02. unsigned char *blur = (unsigned char*) mxGetPr(mxGetField(prhs[1],0,"blur"));//获得仿射变换后的
    patch
03.
04. //if (mxGetM(prhs[1])!=iHEIGHT) { mexPrintf("fern: wrong input image.\n"); return; }
05.
06. // bbox indexes
07. double *idx = mxGetPr(prhs[2]);//bbp所对应的列索引

```

```

08. int numIdx = mxGetM(prhs[2]) * mxGetN(prhs[2]); //1 X (<=10)
09.
10. // minimal variance
11. double minVar = *mxGetPr(prhs[3]); //minVar=0
12. if (minVar > 0) {
13.     img(input, IIMG, iHEIGHT, iWIDTH); //返回IIMG, 是图像进行矩形积分后的结果 (运行不到这)
14.     img2(input, IIMG2, iHEIGHT, iWIDTH); //返回IIMG, 是图像进行矩形平方积分后的结果 (运行不到这)
15. }
16.
17. // output patterns
18. //创建输出矩阵: 10X (<=10)
19. plhs[0] = mxCreateDoubleMatrix(nTREES, numIdx, mxREAL);
20. double *patt = mxGetPr(plhs[0]);
21. //创建输出矩阵: 1 X (<=10)
22. plhs[1] = mxCreateDoubleMatrix(1, numIdx, mxREAL);
23. double *status = mxGetPr(plhs[1]);
24.
25. for (int j = 0; j < numIdx; j++) { // (<=10)
26.
27.     if (minVar > 0) {
28.         double bboxvar = bbox_var_offset(IIMG, IIMG2, BBOX+j*BBOX_STEP); //BBOX保存网格数据索引等数据
        (运行不到这)
29.         //E (p^2) -E^2 (p)
30.         if (bboxvar < minVar) { continue; } (运行不到这)
31.     }
32.     status[j] = 1;
33.     double *tPatt = patt + j*nTREES;
34.     for (int i = 0; i < nTREES; i++) { //10
35.         //返回对应gridbox及对应树的13位有效的像素比较码
36.         tPatt[i] = (double) measure_tree_offset(blur, idx[j]-1, i); //idx:bbp
37.     }
38. }
39. return;

```

进入measure\_tree\_offset

```

[cpp] C
01. int index = 0;
02. int *bbox = BBOX + idx_bbox*BBOX_STEP; //BBOX存储gridbox的索引等信息BBOX_STEP=7(因为grid的行为6)
03. //OFF + bbox[5],该表达式表示该gridbox的特征点信息在OFF的偏移,bbox[5]表示图像横向上多少个网格点

```

```

[cpp] C
01. //OFF = create_offsets(s,x); //记录各个特征点在各种尺度下box中的具体位置
02. int *off = OFF + bbox[5] + idx_tree*2*nFEAT; //OFF存储特征点在各个尺度框下的分布位置等
03. for (int i=0; i<nFEAT; i++) { //13
04.     index<<=1;
05.     //off[0]为特征点的x坐标, off[1]为特征点的y坐标, bbox[0]为该gridbox在图画中的位置
06.     int fp0 = img[off[0]+bbox[0]];
07.     int fp1 = img[off[1]+bbox[0]];
08.     if (fp0>fp1) { index |= 1; } //两个像素点比较并置位相应CODE
09.     off += 2; //移到下一个点对
10. }
11. return index;

```

看完上面, 真的有点累啊, 算了, 把负例也看下好了, 简单看了下, 代码不算太多:

```

[cpp] C
01. // Correct initial bbox
02. tld.bb(:,1) = bbP(1:4,:); //最靠近BBOX的的gridbox
03.
04. // Variance threshold
05. tld.var = var(pEx(:,1)) / 2; //var计算方差,这里即求各个数平方和的平均数
06. // disp(['Variance : ' num2str(tld.var)]);
07.
08. // Generate Negative Examples
09. //nx:patch variance 挑出合适的patches, 并提取fern特征赋给nx,
10. //nEx返回一个225x100(nEx)的矩阵,列向量各元素值为原像素值减去像素均值,100为num_patches
11. //输入:
12. //overlap: 一维行向量, 记录GRID中的各个gridbox与用户目标标定框的重叠率
13. //输出:
14. //nx: patch variance 挑出合适的patches, 并提取fern特征赋给nx
15. //nEx: 一个225x100(nEx)的矩阵,列向量各元素值为原像素值减去像素均值,100为num_patches
16. [nX,nEx] = tldGenerateNegativeData(tld,overlap,tld.img{1});

```

再进

```
[cpp]
01. // Measure patterns on all bboxes that are far from initial bbox
02. //opt.n_par = struct('overlap',0.2,'num_patches',100);
03. idxN = find(overlap<tld.n_par.overlap); //overlap < 0.2
04. [nX,status] = fern(5,img,idxN,tld.var/2); //此函数通过patch variance剔除一批，剩下的进入fern特征码提取
05. idxN = idxN(status==1); // bboxes far and with big variance,注意C++代码中的status[j] = 1;一句
06. nX = nX(:,status==1); //选出进入第二级分类器的负样本
07.
08. // Randomly select 'num_patches' bboxes and measure patches
09. idx = randvalues(1:length(idxN),tld.n_par.num_patches); // 'num_patches',100应该是随机取出100个gridbox
10. bb = tld.grid(:,idxN(idx));
11. nEx = tldGetPattern(img,bb,tld.model.patchsize); //不复注解
```

再进入fern(5,...)因为有tld.var/2，执行稍有不同，请参见上面就行。

好了，至此已经为分类器的训练产生了可用的正例和负例了。

顶 踩  
2 0

[上一篇 TLD matlab 源代码阅读（1）](#)

[下一篇 为什么不去读顶级会议上的论文？适应于机器学习、计算机视觉和人工智能](#)

#### 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（1）](#) 2014-05-17 阅读 1934
- [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

#### 猜你在找

- 《C语言/C++学习指南》数据库篇(MySQL& sqlite)
- MAC OS 108 环境下运行TLD算法MATLAB版
- 顾荣：开源大数据存储系统Alluxio（原Tachyon）的原理
- matlab2013a vs2013 opencv248 编译TLD
- C++语言基础
- MATLAB2010a+OpenCV231+VS2010运行TLD
- Visual Studio 2015开发C++程序的基本使用
- 运行matlab版的TLD算法
- Swift与Objective-C\C\C++混合编程
- TLD的Matlab源码学习



Download Now

#### 查看评论

4楼 [qinchaohit](#) 2016-01-05 21:17发表



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.com  
非常感谢

3楼 [xia316104](#) 2015-04-20 10:50发表



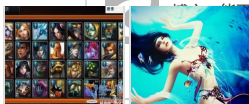
最近在学习TLD，有源代码MATLAB版的，能否给我也发份，邮箱973496253@qq.com

2楼 [我的小和尚](#) 2015-04-12 14:47发表



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@qq.com  
非常感谢了

1楼 [caiyuzhu001](#) 2015-03-16 21:55发表



lol美服代练

少女网页游戏



网页游戏



澳大利亚移民条

给我发一份带有注释的matlab源代码！感谢

067280@qq.com

或[注册]

个人观点，不代表CSDN网站的观点或立场

[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 