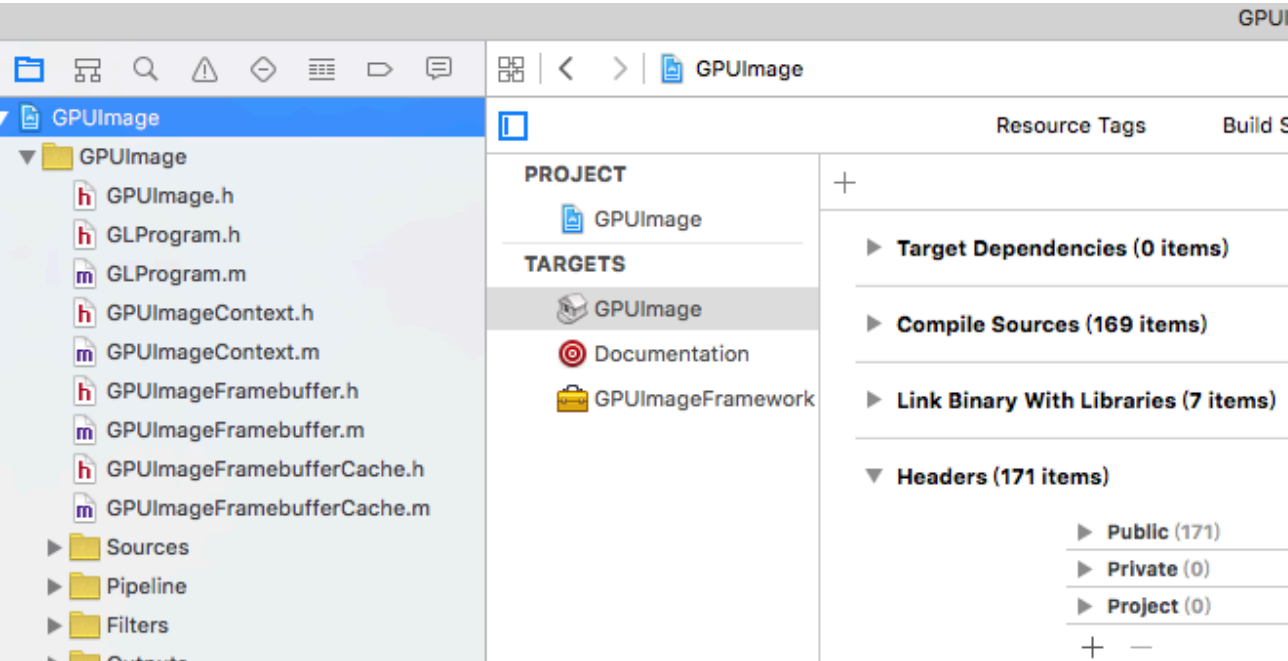


GPUImage的简单使用

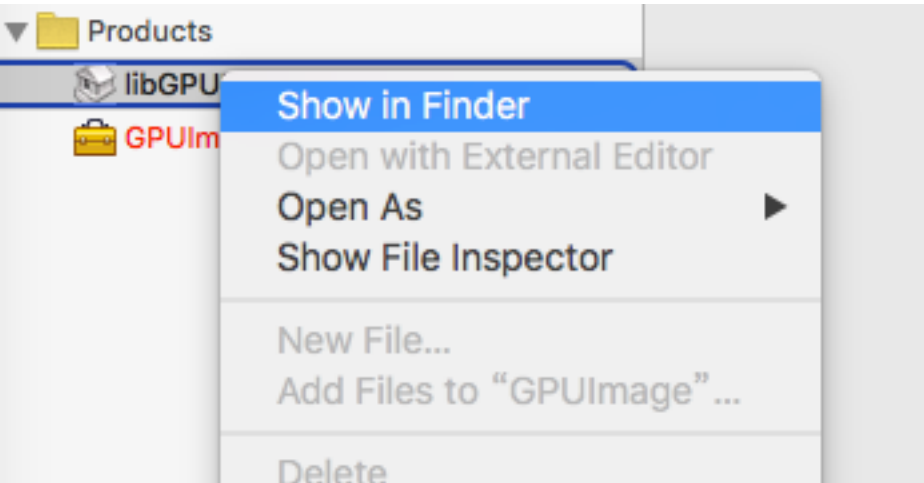
GPUImage 是一个开源的图像处理库，提供了非常多的滤镜效果来加工图片。GPUImage 并不像一般的第三方库可以直接拖入到工程中使用，而是需要先在本地编译，然后将编译后的文件拖入到工程中使用。配置步骤如下：

- (1)：首先下载GPUImage
https://github.com/BradLarson/GPUImage
- (2)：解压后，在framework 目录下，打开 GPUImage.xcodeproj 工程
- (3)：



把Headers文件夹下 Project中的头文件，全部拖到 Public文件夹下面

- (4)：运行该工程
- (5) 点击Products下的 libGPUImage.a ，右键， show in finder



- (6) 把该目录下的libGpuImage.a 和 User文件夹拖到我们的工程下面

公告

昵称： acBool
园龄： 3年8个月
粉丝： 5
关注： 0
+加关注

<		2016年9月						>
日	一	二	三	四	五	六		
28	29	30	31	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	1		
2	3	4	5	6	7	8		

搜索

找找看

谷歌搜索

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

- iOS开发(32)
- 趣味(4)
- Python(2)
- 数据结构(1)
- 数学(1)
- Java(1)

随笔档案

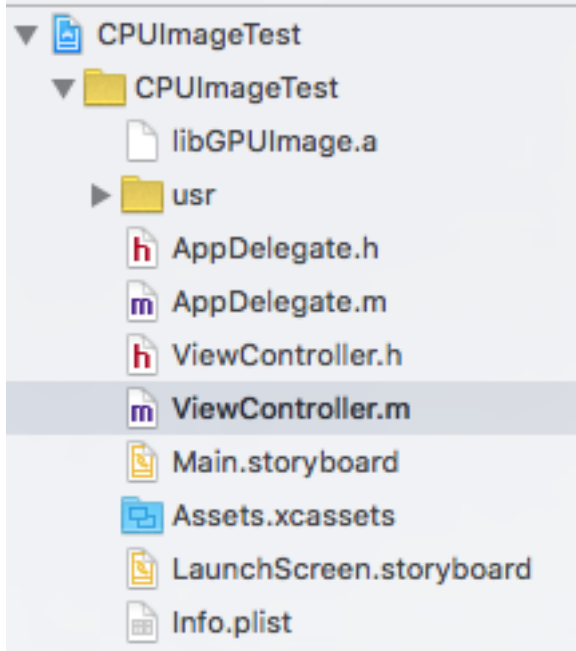
- 2016年8月 (1)
- 2016年7月 (3)
- 2016年6月 (1)
- 2016年5月 (7)
- 2016年2月 (2)
- 2016年1月 (11)
- 2015年12月 (9)
- 2015年10月 (2)
- 2015年9月 (3)

最新评论

阅读排行榜

评论排行榜

推荐排行榜



(7) 然后就可以使用 GPUImage了。一个示例代码：

```
1 - (void)viewDidLoad {
2     [super viewDidLoad];
3
4     UIImage *inputImage = [UIImage imageNamed:@"zzy"];
5     //创建一个亮度的滤镜
6     // GPUImageBrightnessFilter *passthroughFilter =
[[GPUImageBrightnessFilter alloc] init];
7     // passthroughFilter.brightness = -0.5f;
8     // //设置要渲染的区域
9     // [passthroughFilter
forceProcessingAtSize:inputImage.size];
10    // [passthroughFilter useNextFrameForImageCapture];
11    // //获取数据源
12    // GPUImagePicture *stillImageSource =
[[GPUImagePicture alloc] initWithImage:inputImage];
13    // //添加上滤镜
14    // [stillImageSource addTarget:passthroughFilter];
15    // //开始渲染
16    // [stillImageSource processImage];
17    // //获取渲染后的图片
18    // UIImage *newImage = [passthroughFilter
imageFromCurrentFramebuffer];
19    // //加载出来
20    // UIImageView *imageView = [[UIImageView alloc]
initWithImage:newImage];
21    // imageView.frame =
CGRectMake(0,0,inputImage.size.width
,inputImage.size.height);
22    // [self.view addSubview:imageView];
23
24
25    //GPUImageStretchDistortionFilter *disFilter =
[[GPUImageStretchDistortionFilter alloc] init];
26    //GPUImageBulgeDistortionFilter *disFilter =
[[GPUImageBulgeDistortionFilter alloc] init];
27    //GPUImagePinchDistortionFilter *disFilter =
[[GPUImagePinchDistortionFilter alloc] init];
28    //GPUImageGlassSphereFilter *disFilter =
[[GPUImageGlassSphereFilter alloc] init];
29    //GPUImageSphereRefractionFilter *disFilter =
```

```
[[GPUImageSphereRefractionFilter alloc] init];
30      //GPUImageToonFilter *disFilter =
[[GPUImageToonFilter alloc] init];
31      GPUImageVignetteFilter *disFilter =
[[GPUImageVignetteFilter alloc] init];
32      //设置要渲染的区域
33      [disFilter forceProcessingAtSize:inputImage.size];
34      [disFilter useNextFrameForImageCapture];
35      //获取数据源
36      GPUImagePicture *stillImageSource =
[[GPUImagePicture alloc] initWithImage:inputImage];
37      //添加上滤镜
38      [stillImageSource addTarget:disFilter];
39      //开始渲染
40      [stillImageSource processImage];
41      //获取渲染后的图片
42      UIImage *newImage = [disFilter
imageFromCurrentFramebuffer];
43      //加载出来
44      UIImageView *imageView = [[UIImageView alloc]
initWithImage:newImage];
45      imageView.frame =
CGRectMake(50,50,inputImage.size.width
,inputImage.size.height);
46      [self.view addSubview:imageView];
47 }
```



GPUImage提供的过滤器很多，每个头文件就是一个过滤器，这里贴一个过滤器的简介：



```
1 #import "GLProgram.h"
2
3 // Base classes
4 #import "GPUImageOpenGLContext.h"
5 #import "GPUImageOutput.h"
6 #import "GPUImageView.h"
7 #import "GPUImageVideoCamera.h"
8 #import "GPUImageStillCamera.h"
9 #import "GPUImageMovie.h"
10 #import "GPUImagePicture.h"
11 #import "GPUImageRawDataInput.h"
12 #import "GPUImageRawDataOutput.h"
13 #import "GPUImageMovieWriter.h"
14 #import "GPUImageFilterPipeline.h"
15 #import "GPUImageTextureOutput.h"
16 #import "GPUImageFilterGroup.h"
17 #import "GPUImageTextureInput.h"
18 #import "GPUImageUIElement.h"
19 #import "GPUImageBuffer.h"
20
21 // Filters
22 #import "GPUImageFilter.h"
```

```
23 #import "GPUImageTwoInputFilter.h"
24
25
26 #pragma mark - 调整颜色 Handle Color
27
28 #import "GPUImageBrightnessFilter.h" //亮度
29 #import "GPUImageExposureFilter.h" //曝光
30 #import "GPUImageContrastFilter.h" //对比度
31 #import "GPUImageSaturationFilter.h" //饱和度
32 #import "GPUImageGammaFilter.h" //伽马线
33 #import "GPUImageColorInvertFilter.h" //反色
34 #import "GPUImageSepiaFilter.h" //褐色 (怀旧)
35 #import "GPUImageLevelsFilter.h" //阶
36 #import "GPUImageGrayscaleFilter.h" //灰度
37 #import "GPUImageHistogramFilter.h" //色彩直方图, 显示在图片上
38 #import "GPUImageHistogramGenerator.h" //色彩直方图
39 #import "GPUImageRGBFilter.h" //RGB
40 #import "GPUImageToneCurveFilter.h" //色调曲线
41 #import "GPUImageMonochromeFilter.h" //单色
42 #import "GPUImageOpacityFilter.h" //不透明度
43 #import "GPUImageHighlightShadowFilter.h" //提亮阴影
44 #import "GPUImageFalseColorFilter.h" //色彩替换 (替换亮部和暗部色彩)
45 #import "GPUImageHueFilter.h" //色度
46 #import "GPUImageChromaKeyFilter.h" //色度键
47 #import "GPUImageWhiteBalanceFilter.h" //白平衡
48 #import "GPUImageAverageColor.h" //像素平均色值
49 #import "GPUImageSolidColorGenerator.h" //纯色
50 #import "GPUImageLuminosity.h" //亮度平均
51 #import "GPUImageAverageLuminanceThresholdFilter.h" //像素色值亮度平均, 图像黑白 (有类似漫画效果)
52
53 #import "GPUImageLookupFilter.h" //lookup 色彩调整
54 #import "GPUImageAmatorkaFilter.h" //Amatorka lookup
```

```
55 #import "GPUImageMissEtikateFilter.h"
//MissEtikate lookup

56 #import "GPUImageSoftEleganceFilter.h"
//SoftElegance lookup

57
58
59
60
61 #pragma mark - 图像处理 Handle Image
62
63 #import "GPUImageCrosshairGenerator.h" //十字
64 #import "GPUImageLineGenerator.h" //线条
65
66 #import "GPUImageTransformFilter.h" //形状变化
67 #import "GPUImageCropFilter.h" //裁剪
68 #import "GPUImageSharpenFilter.h" //锐化
69 #import "GPUImageUnsharpMaskFilter.h" //反遮罩锐化
70
71 #import "GPUImageFastBlurFilter.h" //模糊
72 #import "GPUImageGaussianBlurFilter.h" //高斯模糊
73 #import "GPUImageGaussianSelectiveBlurFilter.h" //高斯模糊，选择部分清晰
74 #import "GPUImageBoxBlurFilter.h" //盒状模糊
75 #import "GPUImageTiltShiftFilter.h" //条纹模糊，中间清晰，上下两端模糊
76 #import "GPUImageMedianFilter.h" //中值，有种稍微模糊边缘的效果
77 #import "GPUImageBilateralFilter.h" //双边模糊
78 #import "GPUImageErosionFilter.h" //侵蚀边缘模糊，变黑白
79 #import "GPUImageRGBErosionFilter.h" //RGB侵蚀边缘模糊，有色彩
80 #import "GPUImageDilationFilter.h" //扩展边缘模糊，变黑白
81 #import "GPUImageRGBDilationFilter.h" //RGB扩展边缘模糊，有色彩
82 #import "GPUImageOpeningFilter.h" //黑白色调模糊
83 #import "GPUImageRGBOpeningFilter.h" //彩色模糊
84 #import "GPUImageClosingFilter.h" //黑白色调模糊，暗色会被提亮
85 #import "GPUImageRGBClosingFilter.h" //彩色模糊，暗色会被提亮
86 #import "GPUImageLanczosResamplingFilter.h" //Lanczos重取样，模糊效果
87 #import "GPUImageNonMaximumSuppressionFilter.h" //非最大抑制，只显示亮度最高的像素，其他为黑
```

```
88 #import
"GPUImageThresholdedNonMaximumSuppressionFilter.h" //与上相比，像素丢失更多

89

90 #import "GPUImageSobelEdgeDetectionFilter.h"
//Sobel边缘检测算法(白边，黑内容，有点漫画的反色效果)

91 #import "GPUImageCannyEdgeDetectionFilter.h"
//Canny边缘检测算法(比上更强烈的黑白对比度)

92 #import "GPUImageThresholdEdgeDetectionFilter.h" //阈值边缘检测(效果与上差别不大)

93 #import "GPUImagePrewittEdgeDetectionFilter.h" //普瑞维特(Prewitt)边缘检测(效果与Sobel差不多，貌似更平滑)

94 #import "GPUImageXYDerivativeFilter.h"
//XYDerivative边缘检测，画面以蓝色为主，绿色为边缘，带彩色

95 #import "GPUImageHarrisCornerDetectionFilter.h"
//Harris角点检测，会有绿色小十字显示在图片角点处

96 #import "GPUImageNobleCornerDetectionFilter.h"
//Noble角点检测，检测点更多

97 #import "GPUImageShiTomasiFeatureDetectionFilter.h"
//ShiTomasi角点检测，与上差别不大

98 #import "GPUImageMotionDetector.h" //动作检测

99 #import "GPUImageHoughTransformLineDetector.h" //线条检测

100 #import
"GPUImageParallelCoordinateLineTransformFilter.h" //平行线检测

101

102 #import "GPUImageLocalBinaryPatternFilter.h" //图像黑白化，并有大量噪点

103

104 #import "GPUImageLowPassFilter.h" //用于图像加亮

105 #import "GPUImageHighPassFilter.h" //图像低于某值时显示为黑

106

107

108 #pragma mark - 视觉效果 Visual Effect

109

110 #import "GPUImageSketchFilter.h" //素描

111 #import "GPUImageThresholdSketchFilter.h" //阈值素描，形成有噪点的素描

112 #import "GPUImageToonFilter.h" //卡通效果(黑色粗线描边)

113 #import "GPUImageSmoothToonFilter.h" //相比上面的效果更细腻，上面是粗旷的画风

114 #import "GPUImageKuwaharaFilter.h" //桑原(Kuwahara)滤波，水粉画的模糊效果；处理时间比较长，慎用

115

116 #import "GPUImageMosaicFilter.h" //黑白马赛克

117 #import "GPUImagePixellateFilter.h" //像素化

118 #import "GPUImagePolarPixellateFilter.h" //同心圆像素化

119 #import "GPUImageCrosshatchFilter.h" //交叉线阴影，形成黑白网状画面
```



```
120 #import "GPUImageColorPackingFilter.h" //色
彩丢失, 模糊 (类似监控摄像效果)

121

122 #import "GPUImageVignetteFilter.h" //晕
影, 形成黑色圆形边缘, 突出中间图像的效果

123 #import "GPUImageSwirlFilter.h" //漩
涡, 中间形成卷曲的画面

124 #import "GPUImageBulgeDistortionFilter.h" //凸
起失真, 鱼眼效果

125 #import "GPUImagePinchDistortionFilter.h" //收
缩失真, 凹面镜

126 #import "GPUImageStretchDistortionFilter.h" //伸
展失真, 哈哈镜

127 #import "GPUImageGlassSphereFilter.h" //水
晶球效果

128 #import "GPUImageSphereRefractionFilter.h" //球
形折射, 图形倒立

129

130 #import "GPUImagePosterizeFilter.h" //色
调分离, 形成噪点效果

131 #import "GPUImageCGAColorspaceFilter.h"
//CGA色彩滤镜, 形成黑、浅蓝、紫色块的画面

132 #import "GPUImagePerlinNoiseFilter.h" //柏
林噪点, 花边噪点

133 #import "GPUImage3x3ConvolutionFilter.h"
//3x3卷积, 高亮大色块变黑, 加亮边缘、线条等

134 #import "GPUImageEmbossFilter.h" //浮
雕效果, 带有点3d的感觉

135 #import "GPUImagePolkaDotFilter.h" //像
素圆点花样

136 #import "GPUImageHalftoneFilter.h" //点
染, 图像黑白化, 由黑点构成原图的大致图形

137

138

139 #pragma mark - 混合模式 Blend

140

141 #import "GPUImageMultiplyBlendFilter.h" //通
常用于创建阴影和深度效果

142 #import "GPUImageNormalBlendFilter.h" //正
常

143 #import "GPUImageAlphaBlendFilter.h" //透
明混合, 通常用于在背景上应用前景的透明度

144 #import "GPUImageDissolveBlendFilter.h" //溶
解

145 #import "GPUImageOverlayBlendFilter.h" //叠
加, 通常用于创建阴影效果

146 #import "GPUImageDarkenBlendFilter.h" //加
深混合, 通常用于重叠类型

147 #import "GPUImageLightenBlendFilter.h" //减
淡混合, 通常用于重叠类型

148 #import "GPUImageSourceOverBlendFilter.h" //源
混合

149 #import "GPUImageColorBurnBlendFilter.h" //色
彩加深混合

150 #import "GPUImageColorDodgeBlendFilter.h" //色
彩减淡混合

151 #import "GPUImageScreenBlendFilter.h" //屏
幕包裹, 通常用于创建亮点和镜头眩光
```

```
152 #import "GPUImageExclusionBlendFilter.h" //排除混合
153 #import "GPUImageDifferenceBlendFilter.h" //差异混合,通常用于创建更多变动的颜色
154 #import "GPUImageSubtractBlendFilter.h" //差值混合,通常用于创建两个图像之间的动画变暗模糊效果
155 #import "GPUImageHardLightBlendFilter.h" //强光混合,通常用于创建阴影效果
156 #import "GPUImageSoftLightBlendFilter.h" //柔光混合
157 #import "GPUImageChromaKeyBlendFilter.h" //色度键混合
158 #import "GPUImageMaskFilter.h" //遮罩混合
159 #import "GPUImageHazeFilter.h" //朦胧加暗
160 #import "GPUImageLuminanceThresholdFilter.h" //亮度阈值
161 #import "GPUImageAdaptiveThresholdFilter.h" //自适应阈值
162 #import "GPUImageAddBlendFilter.h" //通常用于创建两个图像之间的动画变亮模糊效果
163 #import "GPUImageDivideBlendFilter.h" //通常用于创建两个图像之间的动画变暗模糊效果
164
165
166 #pragma mark - 尚不清楚
167 #import "GPUImageJFAVoronoiFilter.h"
168 #import "GPUImageVoronoiConsumerFilter.h"
```

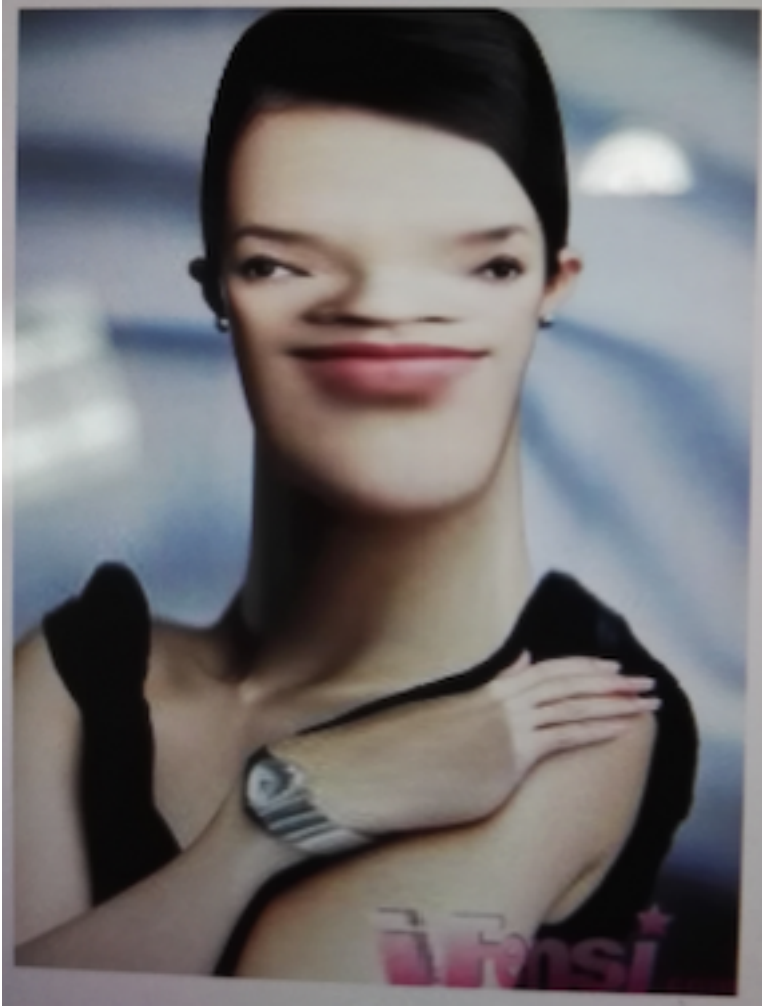



PS：GPUImage 功能非常强大，可以将照片做成类似于哈哈镜的效果，效果图如下：

原图：



处理后的图：



标签: [iOS开发](#)

[好文要顶](#)[关注我](#)[收藏该文](#)

[acBool](#)

[关注 - 0](#)[粉丝 - 5](#)

[+加关注](#)

0

推荐

0

反对

« 上一篇: [OC中 self.view.frame.size.height = 100; 不能通过编译的原因](#)
» 下一篇: [iOS开发中KVC、KVO简介](#)
posted @ 2015-12-28 21:14 acBool 阅读(794) 评论(0) 编辑 收藏
[刷新评论](#) [刷新页面](#) [返回顶部](#)

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问网站首页](#)。

- 【推荐】50万行VC++源码：大型组态工控、电力仿真CAD与GIS源码库
- 【活动】优达学城正式发布“无人驾驶车工程师”课程
- 【推荐】移动直播百强八成都在用融云即时通讯云
- 【推荐】别再闷头写代码！找对工具，事半功倍，全能开发工具包用起来

掘金浏览器插件
e.xitu.io

热门技术内容聚合

- 最新IT新闻：
- 三星平板冒烟致航班紧急迫降 官方称外力所致
 - 任正非：华为三十年大限快到了，想不死就得新生
 - 美团点评买牌照花了4~5亿？并不是！卖家是这么说的...
 - 苹果又获得几项新专利 与地图和键盘等有关
 - 百度美团谈崩了？顺丰或成下一个买家？
- » [更多新闻...](#)

最新知识库文章：

- 陈皓：什么是工程师文化？
 - 没那么难，谈CSS的设计模式
 - 程序猿媳妇儿注意事项
 - 可是姑娘，你为什么要编程呢？
 - 知其所以然（以算法学习为例）
- » 更多知识库文章...