

JS实现照片图片变成黑白线条线稿

这篇文章发布于 2018年06月3日, 星期日, 23:36, 归类于 [JS实例](#)。阅读 10864 次, 今日 9 次 [3 条评论](#)

by zhangxinxu from <http://www.zhangxinxu.com/wordpress/?p=7635>

本文可全文转载, 但需要同时保留原作者和出处。

一、彩色图变成漫画线条效果预览

效果如下, 人物摄影照片:



扁平化的绘制图形:



若想自己体验, 您可以狠狠的点击[这里](#): [JS图片变成线条图片demo](#)

点击demo页面中这个“更换图片”按钮, 可以选择本地图片, 体验线图效果:



二、图片变线稿实现的原理

采用canny算法对图像的边缘进行检测。Canny边缘检测算子是John F. Canny于 1986 年开发出来的一个多级边缘检测算法。截止2014年8月, Canny发表的该篇论文, 已被引用19000余次。

具体该算法实现原理可以参见此[百科](#)。

三、图像的边缘检测在web中的具体实现

关于图像处理的很多经典算法，已经有很多前辈在JS中实现了。例如，人脸识别(Face recognition)，光流(Optical flow)，角点检测等。

几个比较有名的项目JS见下表（附带人脸检测一些数据）：

Chrome 40 / FF 35	
Detections per Second	
Detections	Seconds
js-objectdetect	
17.5 / 16.9	
50 / 50	
2.86 / 2.96	
jsfeat ¹	
9.4 / 6.3	
30 / 30	
3.18 / 4.75	
tracking.js	
7.7 / 8.97	
48 / 48	
6.24 / 5.35	
Beyond Reality Face	
7.4 / 1.7	
41 / 41	
5.50 / 23.98	
CCV ²	
2.2 / 4.4	
8 / 8	
2.22 / 1.80	

js-objectdetect
tracking.js
jsfeat ¹
Beyond Reality Face ²
CCV ³

¹也包括auduno/clmtrackr, camgaze.js。基于老版本的js-objectdetect。

²也包括jquery.facedetection, neave/face-detection, wesbos/HTML5-Face-Detection, auduno/headtrackr。

不同的JS项目对于图像处理的侧重点有所不同。

对于本文展示的黑白线稿效果，我使用的是jsfeat。

jsfeat项目地址: <https://github.com/inspirit/jsfeat>

其文档是几个项目中我觉得做得最好的,上手要最容易。

jsfeat.js中内置了canny边缘检测处理,我们可以直接使用,但也没到一行代码就出结果那么简单。

大致使用步骤和实现原理如下:

1. 引用jsfeat.js, 如下:

```
<script src="./jsfeat-min.js"></script>
```

2. 借助canvas读取图片像素信息:

```
context.drawImage(img, 0, 0, width, height);  
var imageData = context.getImageData(0, 0, width, height);
```

3. 对图片像素信息进行边缘查找算法处理。实现要分为3步: 1. 灰度 2. 高斯模糊 3. canny边缘检测。这样才能获得结果较好的图像边缘信息,这些边缘就是我们需要的图片中的重要线条。

```
// 用来记录存储处理的图片数据对象  
var img_u8 = new jsfeat.matrix_t(width, height, jsfeat.U8C1_t);  
// 灰度  
jsfeat.imgproc.grayscale(imageData.data, width, height, img_u8);  
// 高斯模糊  
jsfeat.imgproc.gaussian_blur(img_u8, img_u8, 6, 0);  
// canny计算  
jsfeat.imgproc.canny(img_u8, img_u8, 20, 50);
```

4. `img_u8` 中的数据此时就是图像边缘数据,然后改变 `imageData` 数据并写入canvas,最终的效果就在web上呈现了。

如果我对原理不感兴趣,只想实现效果?

如果大家只想在自己项目中快速有效果,也是可以的,进入demo页面,左侧可以看到源代码,里面有个名为 `fnCannyEdge()` 的方法,就是图片变线稿的关键。

于是,我们的实现步骤变成下面这样:

1. 引用jsfeat.js, 如下:

```
<script src="./jsfeat-min.js"></script>
```

2. 粘贴 `fnCannyEdge()` 的方法代码;
3. 把图片DOM对象作为参数传进去,例如:

```
fnCannyEdge(img, canvas);
```

效果即达成!

fnCannyEdge语法和API

语法如下:

```
fnCannyEdge(source, canvas, options);
```

其中:

source

表示需要转换的资源，可以是图片 `` DOM元素，也可以是视频 `<video>` DOM元素。此参数必须。

canvas

表示需要呈现最终转换结果的canvas元素。此参数必须。

options

可选参数。具体如下表：

API名称
默认值
释义
blur_radius
2
模糊半径大小
low_threshold
20
表示边缘检测的低阈值
high_threshold
50
表示边缘检测的高阈值

`fnCannyEdge()` 中还包含了一段512 * 512最大尺寸的限制（见下图），因为demo页面选择图片可能很大，实际开发不一定用得到，到时候自行判断要不要删掉。



四、结束语

照片变成黑白简单线条有什么用呢？

1. 手绘风格化

例如，弄一套手绘风格组件，则可以把DOM元素变成图片，然后，再使用本文提供的方法转换下。关于DOM元素如何变成图片，可以参见我这篇文章：[“SVG <foreignObject>简介与截图等应用”](#)。

2. 直接实物变漫画

手绘一张漫画很辛苦了，看看日本那些漫画家，一周画几十张就累死累活。如果可以借助图形处理技术，想画个罗天大醮的龙虎山，直接拿张风景照，擦，瞬间变成线条，关键轮廓就出来，如果是富坚义博，直接就可以给编辑了，从此再也不怕断更了。

3. 减小图片等资源尺寸

有一种图片加载策略是先加载小图，然后大图。其实可以创新下，先加载线稿图，再加载大图。

4. 含沙射影、规避版权

某文章要点评批人，又不好意思直接攻击，可以放一张线稿图。亦或者有些图片有版权，直接用不太好，可以试试线稿，你就说是你亲手绘制的，人家一看这绘画水平，一定会相信的。

5. 没有画技也能图片讲故事

如题。写了部小说，想弄些插画，自己不会画，好的插画师也请不起，学生党又不靠谱。怎么办，自己拍个照，弄个线稿示意示意，聊胜于无，而且前后风格都统一，又有特色，说不定反而大受欢迎，就像简单单纯杨超越，个性自信王菊一样，说不准的。

最后，补充一句：以上作用皆是自己脑洞大开，胡言乱语，大家千万别当真，我自己一条都没用过，纯粹是“每篇文章不随便扯点什么就难受的病”犯了。

以上~

感谢阅读！

《CSS世界》签名版独家发售，包邮，可指定寄语，点击显示购买码

(本篇完) // 想要打赏? 点击[这里](#)。有话要说? 点击[这里](#)。



« 小tips: 点击页面出现富强、民主这类文字动画效果

突破本地离线存储5M限制的JS库localforage简介 »

猜你喜欢

- 时鲜技术: 图像的像素化处理
- 分享一个即插即用的私藏缓动动画JS小算法
- 开源移动端元素拖拽惯性弹动以及下拉加载两个JS
- JS检测PNG图片是否有透明背景、抠图等相关处理
- 图片旋转+剪裁js插件(兼容各浏览器)
- 基于canvas画布的两个炫酷效果展示
- 小tip: 使用CSS将图片转换成模糊(毛玻璃)效果
- 小tips: CSS或JS实现gif动态图片的停止与播放
- 解决canvas图片getImageData,toDataURL跨域问题
- IE6下png背景不透明问题的综合拓展
- 小tips: 使用JS检测用户是否安装某font-family字体

分享到: [+](#) [QQ](#) [微信](#) [微博](#) [贴吧](#) [收藏](#) [0](#)

标签: canvas, jsfeat, 人脸识别, 图像处理, 图片, 算法, 线条, 边缘检测

发表评论 (目前3条评论)

名称 (必须)

邮件地址(不会被公开) (必须)

网站

提交评论

1. **kyomic**说道：

2018年06月29日 17:39

人脸识别项目列举的好，顺道参考下。不过我很喜欢富坚义博

[回复](#)



2. **zhz0115**说道：

2018年06月6日 11:22

人家一看这绘画水平，一定会相信的。哈哈哈哈哈~

[回复](#)



3. **长江长江我是黄河**说道：

2018年06月5日 10:53

好多沙发

[回复](#)



最新文章

- » [常见的CSS图形绘制合集](#)
- » [粉丝群第1期CSS小测评与答疑](#)
- » [分享三个纯CSS实现26个英文字母的案例](#)
- » [小tips: 纯CSS实现打字动画效果](#)
- » [CSS/CSS3 box-decoration-break属性简介](#)
- » [CSS :placeholder-shown伪类实现Material Design占位符交互效果](#)
- » [从天猫某活动视频不必要的3次请求说起](#)
- » [CSS vector-effect与SVG stroke描边缩放](#)
- » [CSS ::backdrop伪元素是干嘛用的?](#)
- » [周知: CSS -webkit-伪元素选择器不再导致整行无效](#)

今日热门

- » [常见的CSS图形绘制合集](#) ⁽¹⁷⁸⁾
- » [粉丝群第1期CSS小测评与答疑](#) ⁽¹¹²⁾
- » [未来必热: SVG Sprite技术介绍](#) ⁽¹¹¹⁾
- » [HTML5终极备忘大全 \(图片版+文字版\)](#) ⁽⁸⁵⁾
- » [让所有浏览器支持HTML5 video视频标签](#) ⁽⁸³⁾
- » [Selectivizr-让IE6~8支持CSS3伪类和属性选择器](#) ⁽⁸⁰⁾
- » [CSS3下的147个颜色名称及对应颜色值](#) ⁽⁷⁸⁾
- » [小tips: 纯CSS实现打字动画效果](#) ⁽⁷²⁾
- » [写给自己看的display: flex布局教程](#) ⁽⁶⁹⁾
- » [分享三个纯CSS实现26个英文字母的案例](#) ⁽⁶⁹⁾

今年热议

- » [《CSS世界》女主角诚寻靠谱一起奋斗之人](#) ⁽⁷⁶⁾
- » [不借助Echarts等图形框架原生JS快速实现折线图效果](#) ⁽⁶⁴⁾
- » [看, for..in和for..of在那里吵架!](#) ⁽⁶⁰⁾
- » [是时候好好安利下LuLu UI框架了!](#) ⁽⁴⁷⁾
- » [原来浏览器原生支持JS Base64编码解码](#) ⁽³⁵⁾
- » [妙法攻略: 渐变虚框及边框滚动动画的纯CSS实现](#) ⁽³³⁾
- » [炫酷H5中序列图片视频化播放的高性能实现](#) ⁽³¹⁾
- » [CSS scroll-behavior和JS scrollIntoView让页面滚动平滑](#) ⁽³⁰⁾
- » [windows系统下批量删除OS X系统.DS_Store文件](#) ⁽²⁶⁾
- » [写给自己看的display: flex布局教程](#) ⁽²⁶⁾

猜你喜欢

- 时鲜技术：图像的像素化处理
- 分享一个即插即用的私藏缓动动画JS小算法
- 开源移动端元素拖拽惯性弹动以及下拉加载两个JS
- JS检测PNG图片是否有透明背景、抠图等相关处理
- 图片旋转+剪裁js插件(兼容各浏览器)
- 基于canvas画布的两个炫酷效果展示
- 小tip: 使用CSS将图片转换成模糊(毛玻璃)效果
- 小tips: CSS或JS实现gif动态图片的停止与播放
- 解决canvas图片getImageData,toDataURL跨域问题
- IE6下png背景不透明问题的综合拓展
- 小tips: 使用JS检测用户是否安装某font-family字体