

JS检测PNG图片是否有透明背景、抠图等相关处理

这篇文章发布于 2018年05月1日, 星期二, 01:33, 归类于 [Canvas相关](#)。阅读 14904 次, 今日 11 次 6 条评论

by zhangxinxu from <http://www.zhangxinxu.com/wordpress/?p=7510>

本文可全文转载, 但需要保留原作者和出处, 摘要引流则随意。

一、JS检测PNG图片是否有透明背景

用户上传图片, 如果是PNG图片, 有时候我们希望这张PNG图片背景不要透明的, 例如:

1. 后端保存图片时候常常会转成JPG格式的, 此时, 如果后端同学不加以判断和处理, 则透明区域会变成黑色, 影响体验;

例如, 下面canvas将PNG图强制转换成JPG后的对比效果:



您可以狠狠地点击这里: [PNG透明背景转JPG后变黑demo](#)

2. 轮播广告的后台, 为了显示质量, 后台往往是直接存图, 保留原格式。这样就有这样一个问题, 如果广告图是上下层叠定位的, 此时如果某个广告图编辑传的是部分区域透明度的图, 前端又没有考虑到这种场景则显示就会出现問題, 透出了下面一张图糟糕的体验问题;

有时候, 我们又希望上传的PNG图片带透明的, 例如:

1. 某平台可以自定义菜单, 允许用户编辑文字和上传图标, 很显然, 这个图标需要是透明背景的, 否则可能UI会很糟糕;
2. QQ邮箱域名邮箱可以自定义logo和主题, 如果上传的域名logo图片背景不是透明的, 主题背景颜色也不一样, 同样会出现比较糟糕的UI, 如下截图:



以上这些场景的处理通常这样的:

- PNG背景透明区域变黑色 -> 后端黑色用白色代替;

- 广告图镂空 -> CSS给 `` 元素设置个背景色:

```
img { background-color: #fff; }
```

但, 如果图片源用在多个地方, 则此方法很难罩住所有场景。

- 上传图标需透明 -> 忽略, 交给用户
- 上传logo需透明 -> 忽略, 交给用户

实际上, 以上所有的处理都可以在上传图片的时候由前端一次性解决。

分两步:

- 前端获取上传图片信息 (包括图片类型, 图片数据), IE10+
- JS检测图片是否有透明背景, IE9+

我们先看demo, 您可以狠狠地点击这里: [png图片是否含有透明像素JS检测demo](#)

如果是不含透明色的PNG图片, 则会提示不含透明; 如果是, 则提示含透明, 如下截图:



检测原理是借助canvas的 `getImageData()` 方法, 关于此方法具体API和使用, 可以参见“[canvas getImageData与任意字符图形点、线动效](#)”这篇文章“像素点信息获取”这里的详细介绍。

一句话概括就是 `getImageData()` 方法可以获取canvas画布上每一个像素点的颜色信息, 于是, 我们只要把上传的图片绘制在canvas上, 然后检测有没有透明的像素点信息即可。

相关检测JS代码如下 (完整代码参见[demo页面](#)):

```
// 图片绘制在画布上
context.drawImage(img, 0, 0);
// 获取图片像素信息
var imageData = context.getImageData(0, 0, width, height).data;
// 检测有没有透明数据
var isAlphaBackground = false;
for (var index = 3; index < imageData.length; index += 4) {
    if (imageData[index] !== 255) {
        isAlphaBackground = true;
        break;
    }
}
// isAlphaBackground就是最后石头有透明或半透明背景色的结果
```

上面 `imageData` 就是我们获取的图片像素信息数组, 是个一维数组, 类型为 `Uint8ClampedArray`, 也就是数组中所有的值范围都是0~255, 数组信息这样:

```
[R,G,B,A,R,G,B,A,R,G,B,A,R,G,B,A,R,G,B,A,R,G,B,A,...]
```

因此, 我们只要判断数组中的A是否全部都是255就可以判断有没有透明信息了, 一个 `for` 循环就搞定

了。

如果图片尺寸很大

demo页面旨在演示，如果图片尺寸很大，按照demo页面的处理，可能会比较耗时间，我们可以在把canvas画布进行尺寸限制，也就是图片尺寸压缩，这个不会影响对是否有透明背景的判断，具体操作和代码可以参见这篇文章：“[canvas实现图片前端JS压缩并上传](#)”。

背景透明与否的实际处理

如果站在解决实际问题的角度，判断是否背景透明并不一定要得到。

1. 对于不需要透明背景的处理，我们可以直接使用canvas补一个背景色，无论图片是不是透明的都不需要判断。操作如下：

1. canvas绘制一个底色，通常是白色：

```
context.fillStyle = '#ffffff';
context.fillRect(0, 0, canvas.width, canvas.height);
```

2. 绘制图片：

```
context.drawImage(img, 0, 0);
```

3. 上传canvas图片（转换成base64后者blob数据都可以），`canvas.toDataURL(file.type)` 或者 `canvas.toBlob(callback, file.type)`。

2. 如果是需要PNG背景图片，则我们就可以先进行前端检测，然后：

- 简单的处理是直接提示用户，“您上传图片为实色背景，最终效果可能不好看，确认继续？”
- 更好的处理则是发现用户上传图片不是透明PNG图，则立即弹出个小窗口，在线背景去色就好了！于是自然引出了下一部分内容，如何实现图片的简易抠图（背景透明）效果？

二、实现一个简易的图片背景去色效果

图片去色网上很多在线的工具，有些还很厉害，例如[clippingmagic](#)，但，这东西是别人的，且我们不需要这么复杂交互，就算代码爬过来也没什么用，还是自己写写功能更实在。

好在，如果只是想实现一个不太复杂的图片背景色去色效果，还是挺容易的。

实现的关键依然是 `getImageData()` 方法，去色，然后把相似颜色变成透明颜色就可以了。

您可以狠狠地点击这里：[借助canvas实现图片背景色去色demo](#)

例如，我们点击左边图片的白色区域，然后设置容差值是20，点击“执行去色”按钮，结果如下截图：



如果我们选择紫色，则效果是：



实现原理

关键2个步骤，一是取色，二是去色。

1. 取色的实现

根据点击坐标，取当前点击位置像素点颜色值信息即可，代码如下：

```
canvas.addEventListener('click', function (event) {
```

```

var rect = this.getBoundingClientRect();
var x = event.clientX - rect.left;
var y = event.clientY - rect.top;
// rgbaPicker就是点击像素点的颜色信息
rgbaPicker = context.getImageData(x, y, 1, 1).data;
});

```

2. 去色的实现

去色则是替换 `getImageData()` 的一些像素点色值为透明即可，这个不难，难的是如何判别两种色值是相似的呢？

这里有个简易的颜色色值相似度计算算法：

```
similar = Math.sqrt((r2 - r1) * (r2 - r1) + (g2 - g1) * (g2 - g1) + (b2 - b1) * (b2 - b1))
```

其中，`r2`，`r1` 表示RGB中的红色red，`g2`，`g1` 表示RGB中的绿色green，`b2`，`b1` 表示RGB中的蓝色blue。

OK，有了相似度算法，下面就是简单的色值替换处理了：

```

// 像素点色值
var rgba = rgbaPicker;
// 容差大小
var tolerance = eleTolerance.value;
// 基于原始图片数据处理,
// 其中:
// imgData是左图像素信息,
// imgDataResult是右图处理后的像素信息
for (var index = 0; index < imgData.data.length; index += 4) {
    var r = imgData.data[index];
    var g = imgData.data[index + 1];
    var b = imgData.data[index + 2];

    if (Math.sqrt(
        (r - rgba[0]) * (r - rgba[0]) +
        (g - rgba[1]) * (g - rgba[1]) +
        (b - rgba[2]) * (b - rgba[2])) <= tolerance
    ) {
        imgDataResult.data[index] = 0;
        imgDataResult.data[index + 1] = 0;
        imgDataResult.data[index + 2] = 0;
        imgDataResult.data[index + 3] = 0;
    } else {
        imgDataResult.data[index] = r;
        imgDataResult.data[index + 1] = g;
        imgDataResult.data[index + 2] = b;
        imgDataResult.data[index + 3] = imgData.data[index + 3];
    }
}
// put数据
contextResult.putImageData(imgDataResult, 0, 0);

```

点睛代码就是最后的 `putImageData()`，可以让画布使用新数据进行图像呈现，就可以看到去色后的效果了。

以上JS为片段截取，如果看不透彻，可以前往[demo页面](#)查看完整的源代码，未压缩，就在页面上，简洁

明了无干扰，非常适合学习。

代码采用MIT协议，可以随意复制，商业亦可，但需要保留代码顶部的作者和原出处版权说明。所有代码都是自己键盘一个一个敲出来的，尊重辛勤劳动，尊重代码使用授权协议，共建和谐社会。如果发现都是删掉协议说明的拿来主义，我想我会考虑使用付费。

其他说明

这里的去色效果是最简单的去色效果了，其实标准的去色功能，还需要一个“连续”还是“不连续”的勾选功能。

但是，对于图标和logo这些去色需求，当前的整体去色应该足够，“连续”功能有兴趣的小伙伴可以自己尝试实现下，考验连续区域算法。

三、结束语

本文第一段“检测透明”，和第二段“去色变透明”本来计划是两篇文章的，因为以前很多文章好多好东西都整在一个文章里，结果很容易就湮没了，无人问津，甚是可惜；加上文章篇幅变得很长，大家又都很忙，没有那么多耐心通读长文，也很容易错过稀缺优质内容。

但是写着写着还是控制不住自己的手，还是合在了一起，其实我也不明白我的执念在哪里！虽然分开总体阅读类肯定比一篇要多，搜索效果也更好，PV数据也会好看些，这个月发文文章数量也会上去，但是，一想到要分开变成两篇也不错的文章，我就心里难受，难道是跟洁癖一样的心理疾病——“文章大而全合体综合征”，无法放弃去完美品质文章的追求。

忽略眼前利益的极致追求，或许是自己印刻在骨子里的特质吧，也是自己能够成长到现在这个样子的原因所在。

感谢阅读，欢迎交流！

《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](#)。



« [文章文档标题自动生成导航jQuery小插件titleNav.js](#)

[炫酷H5中序列图片视频化播放的高性能实现](#) »

猜你喜欢

- 瞎折腾：把JS,CSS任意文本文件加密成一张图片
- IE6下png背景不透明问题的综合拓展
- 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
- 小tips:使用canvas在前端实现图片水印合成
- SVG <foreignObject>简介与截图等应用
- JS实现照片图片变成黑白线条线稿
- CSS背景色镂空技术实际应用及进阶

- CSS镂空图片transition过渡初加载背景色块问题解决
- CSS ::backdrop伪元素是干嘛用的?
- HTML5 file API加canvas实现图片前端JS压缩并上传

分享到:         1

标签: Blob, getImageData, putImageData, toDataURL, 相似度, 算法, 背景透明, 透明, 颜色

发表评论（目前6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. edison说道:

2018年11月1日 18:01

我想请求一下,我上传png的图片到canvas当中,有透明底的图片会自动变成黑色背景,如果我想保留透明底怎么处理?

<https://segmentfault.com/q/1010000014137605>

跟这个问题相似↑

[回复](#)



2. abaddonpoet说道:

2018年08月17日 10:41

所以。。一直有个刚需。。有没有办法能去掉白色底,并且边缘的颜色也可以计算出去除白色以后的半透明值。。。。。。然后再叠加上期望的背景色,保存成png8。。。这就是真·抠图了。。。。

[回复](#)



3. 农业微信群大全说道:

2018年05月15日 08:48

是的 要脱颖而出总是要有不同寻常的特质 你的坚持让人佩服

[回复](#)



4. vbyzc说道:

2018年05月6日 18:33

太好的文章了,张兄,有一个问题,我搞了个demo,然后我想用张兄的方法检测是否有透明的部分,先用php遍历出png图片,但是文章中是用file表单选中后,然后赋值: file = event.target.files[0];我把这个file打印出来,是个对象,多个元素包含这个文件的几个信息。

但我用php的话,,可以把每张png的路径和文件名输出到页面中,然后在js中,怎么改成以路径文件名来得到一样的file对象?



回复

5. micstone说道:

2018年05月1日 08:47

为劳动者点赞

回复



农业摆渡人说道:

2018年05月17日 13:45

有收获才是好!

回复



最新文章

- » 常见的CSS图形绘制合集
- » 粉丝群第1期CSS小测点评与答疑
- » 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 (179)
- » 粉丝群第1期CSS小测点评与答疑 (113)
- » 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 (73)
- » 写给自己看的display: flex布局教程 (70)
- » 分享三个纯CSS实现26个英文字母的案例 (70)

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

- 瞎折腾: 把JS,CSS任意文本文件加密成一张图片
- IE6下png背景不透明问题的综合拓展
- 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
- 小tips:使用canvas在前端实现图片水印合成
- SVG <foreignObject>简介与截图等应用
- JS实现照片图片变成黑白线条线稿

- CSS背景色镂空技术实际应用及进阶
- CSS镂空图片transition过渡初加载背景色块问题解决
- CSS ::backdrop伪元素是干嘛用的?
- HTML5 file API加canvas实现图片前端JS压缩并上传