第二次作业

191250128 孙钰昇

gitlab：http://49.235.227.136:8099/root/WebHomework

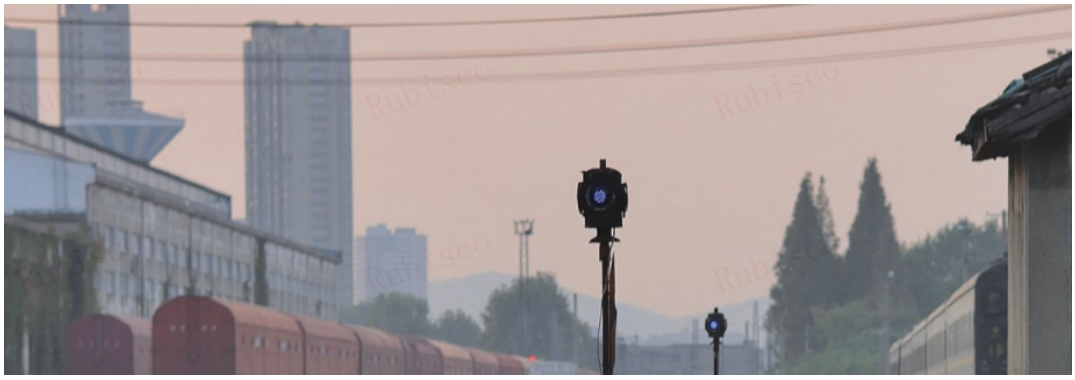
> 前端水印方案实现图像水印，分别实现可见水印和不可见数字水印。可自行选择实现方案
>
> 作业预览网址点这里
>
> 注，服务器可能拒绝登陆和注册时的post请求这时需要重新输入对应网址
>
> - 主页：点这里

说明

在第一次作业的基础上，采用js重写了二级页面。并将二级页面从上至下展示为：

进入二级页面的方法是在主页index中点选图片，而不是直接访问Subpage.html（或者手动传入参数也行`subpage.html?3.jpg`）

- 大图（点击跳转原图）
- 有可见水印的大图
  - 水印如图所示，是许多倾斜的红色Rubisco



- 隐写水印
  - 注：直接在文件中打开可能因为没有域名被浏览器视为跨域，无法正常绘制。可以在webstorm中打开或本地localhost运行或访问点这里
- 有隐写水印的大图

可见水印：Canvas+背景图标

实现方法

用Canvas绘制base64的图片和文字，并将canvs转换出来的地址URL作为图片div的背景。为原图片设置一点透明度。即可显示出背景的水印

关键代码

js

```
function createWaterMark() {
    const angle = -20;
    //设置旋转角度
```

```
    const txt = 'Rubisco'
    //设置文本
    const canvas = document.createElement('canvas')
    const ctx = canvas.getContext('2d');
    ctx.clearRect(0, 0, 180, 100);
    ctx.globalAlpha = 0.9;
    //设置透明度
    ctx.font = `32px serif`
    ctx.fillStyle = "red"
    //设置字体字号颜色
    ctx.rotate(Math.PI / 180 * angle);
    ctx.fillText(txt, 0, 50);
    return canvas.toDataURL();
}
var div = document.getElementById('visiblemask')
//找到对应盒子
div.innerHTML = "<img src="+link+" alt="+link+"} />"
div.style.backgroundImage = "url("+createWaterMark()+")"
//绘制
```

css

```
div#visiblemask{
    background-repeat: repeat;
    width: 80%;
    height: 60%;
    margin-top: 5%;
    margin-left: 10%;
    pointer-events: none;
    //取消事件
}
div#visiblemask img{
    width: 100%;
    height: 100%;
    margin: 0;
    opacity: 0.95;
    //设置透明
}
```

隐式水印

实现方法

在Canvas绘制图片的过程中，将水印图片的像素藏在原图片中。

- 加密：即分析两个图片在同一个点的信息，如果某一点有水印图片的数据，就把这一点的某一个选定通道分量设为奇数。相应地，没有水印图片数据的点在该分量被设为偶数。
- 解密：分析图片的每一个像素点，在已知哪个通道分量被加密的前提下，对该通道分量每个奇数像素点设为255，每个偶数像素点设为0，清除其他通道分量后，有图片信息的点会被置为一个颜色，其他像素点为黑色

关键代码

js

```
function invisibleMask(img){
    var invisibleMask =
document.getElementById('invisiblemask').getContext("2d")
    var unmask = document.getElementById('unmask').getContext('2d')
    //分别获得水印和加水印的图片对应的画布
    var imgdata
    var textData
    var processData = function (originalData,bit){
        //解密：需要传入通道分量，否则不能正确解密
        var data = originalData.data
        for(var i=0;i<data.length;i++){
            if(i%4===bit){
                //对目的通道的操作
                if(data[i] % 2 === 0){
                    //偶数代表这里不藏信息
                    data[i] = 0;
                } else {
                    //奇数代表这里藏着信息
                    data[i] = 255;
                }
            }
            else if(i%4===3){
                //不能把透明度分量关掉
            }
            else {
                data[i]=0;
                //可以选择关掉其他分量来保证水印看得清楚。也可以选择不关，这样会在变色的图片
上显示水印
            }
        }
        unmask.putImageData(imgdata,0,0)
    }
    var merge_data = function (bit) {
        //加密：传入目的通道
        var oData = imgdata.data
        var newData = textData
        //就是将这两个信息混合
        for(var i=0;i<oData.length;i++){
            if(i%4===bit){
                //对目标通道
                if(newData[i+(3-bit)]===0&&(oData[i]%2===1)){
                    //newData[i+(3-bit)]===0表示这里没有信息
                    //oData[i]%2===1表示这里本来是奇数
                    //按照方法中所述这里应该是偶数，所以加1（255应该减1防溢出）
                    if(oData[i]===255){
                        oData[i]--;
                    }
```

```
                    else {
                        oData[i]++;
                    }
                }
                else if(newData[i+(3-bit)]!==0&&(oData[i]%2===0)){
                    //有信息的像素，如果是偶数就加一
                    oData[i]++;
                }
            }
        }
        invisibleMask.putImageData(imgdata,0,0)
        return imgdata
    }
    var setSize = function (id,width,height){
        document.getElementById(id).setAttribute("width",width)
        document.getElementById(id).setAttribute("height",height)
    }
    img.onload = function () {
        var bit = Math.floor(Math.random() * 3)//这里使用随机通道写水印，对应即可得到到
a通道
        invisibleMask.font = '30px Microsoft Yahei'
        invisibleMask.fillText('Rubisco的水印'+bit,30,30)
        textData = invisibleMask.getImageData(0,0,img.width,img.height).data
        var width = img.width
        var height = img.height
        setSize('invisiblemask',width,height)
        setSize('unmask',width,height)
        setSize('visiblemask',width,height)
        //设置画布大小
        invisibleMask.drawImage(img,0,0,width,height)
        imgdata = invisibleMask.getImageData(0,0,width,height)
        processData(merge_data(bit),bit)
    }
}
```