**《移动平台应用开发》实验报告**

**图像滤波**

**成 绩：**

**学 号：**

**姓 名：**

**班 级：**

**指导教师：**

**报告日期： 年 月 日**

1. **实验内容**

刚获得的图像有很多噪音。这主要由于平时的工作和环境引起的，图像增强是减弱噪音，增强对比度。想得到比较干净清晰的图像并不是容易的事情。为这个目标而为处理[图像所](https://www.baidu.com/s?wd=%E5%9B%BE%E5%83%8F%E6%89%80&tn=SE_PcZhidaonwhc_ngpagmjz&rsv_dl=gh_pc_zhidao" \t "_blank)涉及的操作是设计一个适合、匹配的滤波器和恰当的阈值。常用的有高斯滤波、均值滤波、中值滤波、最小均方差滤波、Gabor滤波。本次实验的主要内容是实现图像滤波，了解关于图像处理的基本内容

1. **实验目的**

使用Swift语言，利用Xcode工具实现一个简易的图像滤波

1. **功能模块描述和设计思路**

故事面板设计：

viewController

label显示滤波类型和大小

Segmented

选择滤波类型

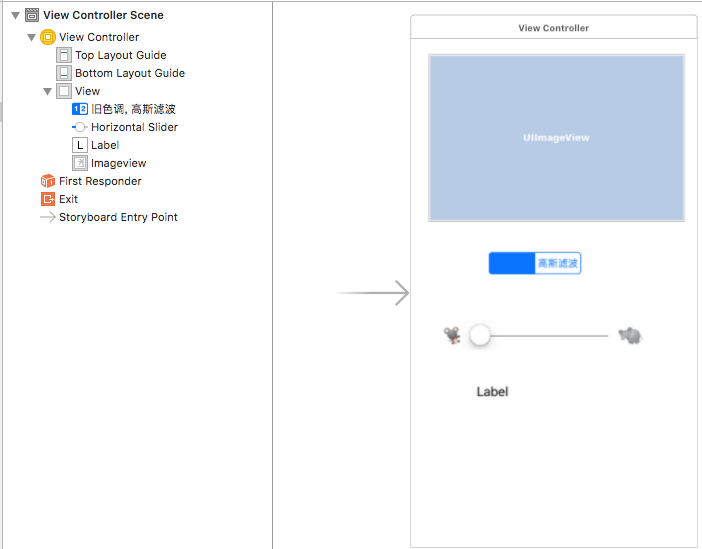
Horizontal Sliser

控制滤波大小

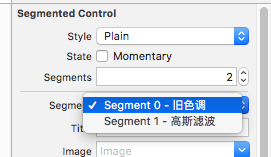
imageView

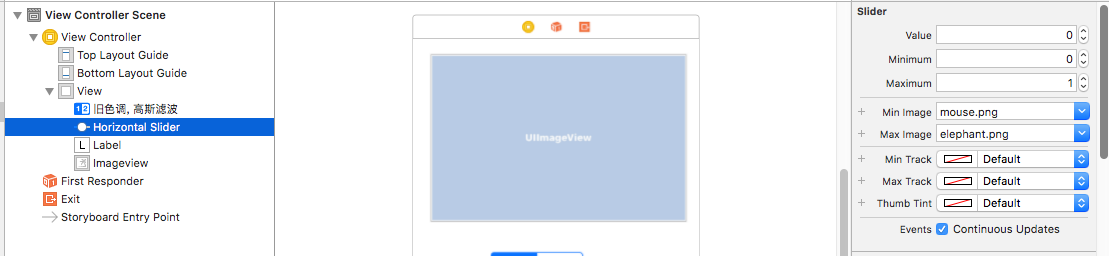
被处理的图片

界面设计截图：

i

1. **具体实现过程**
2. 首先先新建一个Single View Application模版，工程名为Filter,把11 resouce中（三个选项都是第一个，这样资源才导的进去），文件拖入项目回到故事面板，拖拽一个imageView,用于显示待处理的图片，以及处理后的图片，再拖拽一个Horizontal Sliser，用滑动条来控制滤波的大小，在属性检查器里面Min image属性选择图片mouse.png,Max image属性选择elephant.png；再拖入一个Segmented，在segment属性中设为2，一个是进行旧色调滤波用的，一个是进行高斯滤波用的；最后拖拽一个Label，用于显示当前滤波类型，以及滤波大小





2、点击辅助编辑器，进入ViewContorller里面进行编辑代码

首先添加相应

一个是imageView,connection设置为outlet，命名为imageView

还有label标签设置为outlet,命名为label

设置变量image用于后面对图像进行操作

设置int变量flag,用于判断当前滤波类型

class ViewController: UIViewController {

@IBOutlet weak var imageview: UIImageView!

@IBOutlet weak var label: UILabel!

var image:UIImage!

var flag=0

override func viewDidLoad() {

super.viewDidLoad()

// Do any additional setup after loading the view, typically from a nib.

image=UIImage(named: "SkyDrive340.png")

imageview.image=image

label.text=""

}

@IBAction func changevalue(\_ sender: Any) {

let seg=sender as! UISegmentedControl

if seg.selectedSegmentIndex==0

{

flag=0

}

else

{

flag=1

}

}

@IBAction func sliderchange(\_ sender: Any) {

let slider=sender as! UISlider

var value:Float=slider.value

if flag==0

{

filterSpiaTone(value)

}

else

{

filterGaus(value)

}

}

func filterSpiaTone(\_ value:Float)

{

let context=CIContext(options: nil)

let cimage=CIImage(cgImage: self.image.cgImage!)

if let sepiaTone=CIFilter(name: "CISepiaTone")

{

sepiaTone.setValue(cimage, forKey: "inputImage")

let text=String(format:"旧色调 %f",value)

self.label.text=text

sepiaTone.setValue(value, forKey: "inputIntensity")

let result=sepiaTone.value(forKey: "outputImage") as! CIImage

let imageRef=context.createCGImage(result, from: CGRect(x: 0, y: 0, width:self.imageview.image!.size.width, height: self.imageview.image!.size.height))

let image=UIImage(cgImage: imageRef!)

self.imageview.image=image

}

flag=0

}

func filterGaus(\_ value:Float)

{

let context=CIContext(options: nil)

let cimage=CIImage(cgImage: self.image.cgImage!)

if let Guss=CIFilter(name: "CIGaussianBlur")

{

Guss.setValue(cimage, forKey: "inputImage")

let text=String(format:"GUss Radius %f",value\*10)

NSLog(text)

self.label.text=text

Guss.setValue(value, forKey: "inputRadius")

let result=Guss.value(forKey: "outputImage") as! CIImage

let imageRef=context.createCGImage(result, from: CGRect(x: 0, y: 0, width:self.imageview.image!.size.width, height: self.imageview.image!.size.height))

let image=UIImage(cgImage: imageRef!)

self.imageview.image=image

}

flag=1

}

override func didReceiveMemoryWarning() {

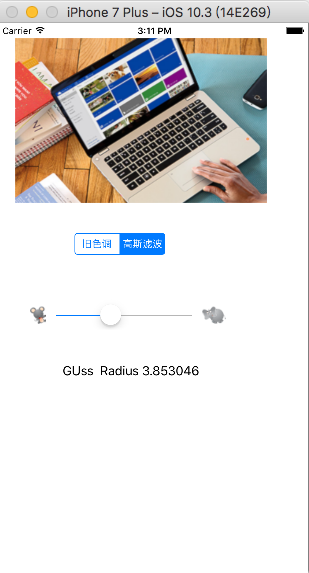
super.didReceiveMemoryWarning()

// Dispose of any resources that can be recreated.

}

}

1. **运行结果**

1. **总结和展望**

这一次的实验是按照书上的步骤来的，容易操作，对于页面的设计也比较少，主要是代码的编写，尤其是对于滤波类型的替换，通过对于页面的点击操作，用Label来显示当前滤波类型和大小。对于以娱乐为主的iOS，Android等移动设备来讲，图片的处理变得越来越收用户重视，所以理解有关图像的操作是十分必要的，这一次的实验算是带我们了解了一点有关图像处理的内容，但是知道的并且学会的内容不多，还是需要课后的大量练习