Android中的图像处理

图像分析之ARGB模型（透明度红绿蓝）

- 色调/色相–物体传递的颜色
- 饱和度–颜色的纯度，从0（灰）到100%（饱和）来进行描述（红色，淡红，大红）
- 亮度/明度–颜色的相对明暗程度

系统提供的类

色调：ColorMatrix

```
ColorMatrix hueMatrix=new ColorMatrix();
hueMatrix.setRotate(0,hue);//--R
hueMatrix.setRotate(1,hue);//--G
hueMatrix.setRotate(2,hue);//--B
```

饱和度

```
ColorMatrix saturationMatrix=new ColorMatrix();
saturationMatrix.setSaturation(saturation);
```

亮度

```
ColorMatrix lumMatrix=new ColorMatrix();
lumMatrix.setScale(lum,lum,lum,1);
```

实例演示

xml布局

```xml
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">

    <ImageView
        android:id="@+id/iv_show"
        android:layout_width="300dp"
        android:layout_height="300dp"
        android:layout_centerHorizontal="true"
        android:layout_marginBottom="25dp"
        android:layout_marginTop="25sp" />

    <SeekBar
        android:id="@+id/seekbarHue"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@id/iv_show" />

    <SeekBar
        android:layout_marginTop="10dp"
        android:id="@+id/seekbarstu"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@id/seekbarHue" />

    <SeekBar
        android:layout_marginTop="10dp"
        android:id="@+id/seekbarlun"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@id/seekbarstu" />
</RelativeLayout>
```

工具类ImageHelper.java

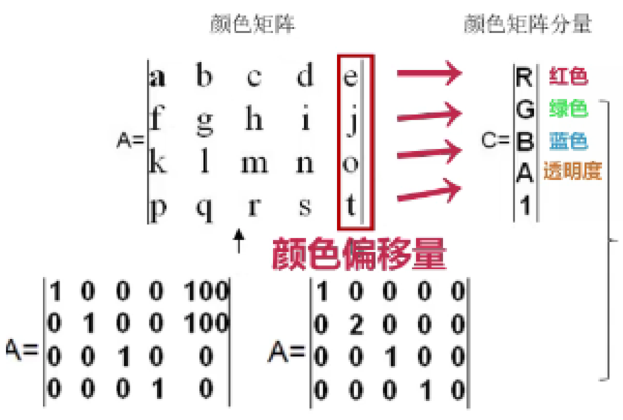
```java
package com.flyme.moyu.imageprocess;

import android.graphics.Bitmap;
import android.graphics.Canvas;
import android.graphics.ColorMatrix;
import android.graphics.ColorMatrixColorFilter;
import android.graphics.Paint;
```

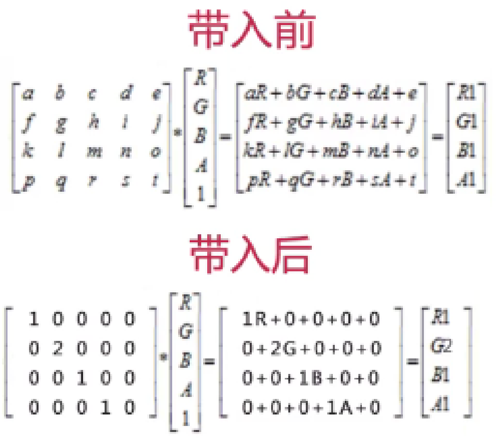
```
/**
 * Created by Administrator on 2015/9/1.
 * 图像处理工具类
 */
public class ImageHelper {
    /**
     * @param bm               :待处理图片
     * @param hue: 色调
     * @param saturation: 饱和度
     * @param lum              : 亮度
     * @return
     */

    public static Bitmap handleImageEffect(Bitmap bm, float hue,
float saturation, float lum) {
       //传递过来的bitmap默认是不可修改的
        Bitmap
bmp=Bitmap.createBitmap(bm.getWidth(),bm.getHeight(),Bitmap.Config
.ARGB_8888);
        Canvas canvas=new Canvas(bmp);
        Paint paint=new Paint(Paint.ANTI_ALIAS_FLAG);

        ColorMatrix hueMatrix=new ColorMatrix();
        hueMatrix.setRotate(0,hue);
        hueMatrix.setRotate(1,hue);
        hueMatrix.setRotate(2, hue);

        ColorMatrix saturationMatrix=new ColorMatrix();
        saturationMatrix.setSaturation(saturation);

        ColorMatrix lumMatrix=new ColorMatrix();
        lumMatrix.setScale(lum,lum,lum,1);

        //将前面设置的三个揉和起来
        ColorMatrix imageMatrix=new ColorMatrix();
        imageMatrix.postConcat(hueMatrix);
        imageMatrix.postConcat(saturationMatrix);
        imageMatrix.postConcat(lumMatrix);

        paint.setColorFilter(new
ColorMatrixColorFilter(imageMatrix));
        canvas.drawBitmap(bm,0,0,paint);

        return bmp;
    }
}
```

显示PrimaryColorActivity.java

```java
package com.flyme.moyu.imageprocess;

import android.app.Activity;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.os.Bundle;
import android.widget.ImageView;
import android.widget.SeekBar;

import butterknife.Bind;
import butterknife.ButterKnife;

/**
 * Created by Administrator on 2015/9/1.
 */
public class PrimaryColorActivity extends Activity implements
SeekBar.OnSeekBarChangeListener {

    @Bind(R.id.iv_show)
    ImageView showImageView;
    @Bind(R.id.seekbarHue)
    SeekBar hueSeekBar;
    @Bind(R.id.seekbarstu)
    SeekBar stuSeekBar;
    @Bind(R.id.seekbarlun)
    SeekBar lunSeekBar;
    public static int MAX_VALUE = 255;
    public static int MID_VALUE = 127;
    public float mHun, mStaurtion, mLum;
    private Bitmap bitmap;


    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_primary_color);
        ButterKnife.bind(this);

        bitmap = BitmapFactory.decodeResource(getResources(),
R.drawable.a);
        hueSeekBar.setOnSeekBarChangeListener(this);
        stuSeekBar.setOnSeekBarChangeListener(this);
        lunSeekBar.setOnSeekBarChangeListener(this);


        hueSeekBar.setMax(MAX_VALUE);
        stuSeekBar.setMax(MAX_VALUE);
        lunSeekBar.setMax(MAX_VALUE);
        hueSeekBar.setProgress(MID_VALUE);
        stuSeekBar.setProgress(MID_VALUE);
        lunSeekBar.setProgress(MID_VALUE);

        showImageView.setImageBitmap(bitmap);
    }

    @Override
    public void onProgressChanged(SeekBar seekBar, int i, boolean
b) {
        switch (seekBar.getId()) {
            case R.id.seekbarHue:
            //下公式是根据经验得出的
                mHun = (i - MID_VALUE) * 1.0f / MID_VALUE * 180;
                break;
            case R.id.seekbarstu:
            //下公式是根据经验得出的
                mStaurtion = i * 1.0f / MID_VALUE;
                break;
            case R.id.seekbarlun:
            //下公式是根据经验得出的
                mLum = i * 1.0f / MID_VALUE;
                break;
        }


showImageView.setImageBitmap(ImageHelper.handleImageEffect(bitmap,
mHun,mStaurtion,mLum));
    }

    @Override
    public void onStartTrackingTouch(SeekBar seekBar) {

    }

    @Override
    public void onStopTrackingTouch(SeekBar seekBar) {

    }
}
```

Android图像–矩阵变换





实现思想

设置颜色矩阵的分量以及偏移量

示例代码

ColorMatrixActivity.java

```java
package com.flyme.moyu.imageprocess;

import android.app.Activity;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.graphics.Canvas;
import android.graphics.ColorMatrix;
import android.graphics.ColorMatrixColorFilter;
import android.graphics.Paint;
import android.os.Bundle;
import android.util.Log;
import android.view.Gravity;
import android.widget.EditText;
import android.widget.GridLayout;
import android.widget.ImageView;


import butterknife.Bind;
import butterknife.ButterKnife;
import butterknife.OnClick;

/**
 * Created by Administrator on 2015/9/1.
 * 矩阵变换来进行图像处理
 */
public class ColorMatrixActivity extends Activity {

    @Bind(R.id.iv_color_matrix)
    ImageView mImageView;
    @Bind(R.id.gl_matrix)
    GridLayout mGridLayout;

    private Bitmap bitmap;
    private int mEtWidth, mEtHeight;
    private EditText[] editTexts = new EditText[20];
    private float[] mColorMetrix = new float[20];

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.color_matrix);
        Log.i("tag","加载界面");
        ButterKnife.bind(this);

        bitmap = BitmapFactory.decodeResource(getResources(),
R.drawable.a);

        mImageView.setImageBitmap(bitmap);
        Log.i("tag","iv已显示");

        //此处用来获取mGridlayout的宽高
        mGridLayout.post(new Runnable() {
            @Override
            public void run() {
                mEtWidth = mGridLayout.getWidth() / 5;
                mEtHeight = mGridLayout.getHeight() / 4;
                addEts();
                initMatrix();
            }
        });
    }

    public void getMatrix() {
        for (int i = 0; i < 20; i++) {
            mColorMetrix[i] =
Float.valueOf(editTexts[i].getText().toString());
        }
    }

    public void setImageMatrix() {
        Bitmap bmp = Bitmap.createBitmap(bitmap.getWidth(),
                bitmap.getHeight(), Bitmap.Config.ARGB_8888);
        ColorMatrix colorMatrix = new ColorMatrix();
        colorMatrix.set(mColorMetrix);

        Canvas canvas = new Canvas(bmp);
        Paint paint = new Paint(Paint.ANTI_ALIAS_FLAG);
        paint.setColorFilter(new
ColorMatrixColorFilter(colorMatrix));
```

```
            canvas.drawBitmap(bitmap, 0, 0, paint);
            mImageView.setImageBitmap(bmp);
        }

        @OnClick(R.id.btn_change)
        public void btnChange() {
            getMatrix();//获取矩阵的值
            setImageMatrix();//设置矩阵的值
        }

        @OnClick(R.id.btn_reset)
        public void btnReset() {

            initMatrix();//初始化矩阵
            getMatrix();
            setImageMatrix();
        }


        //初始化EditView
        private void addEts() {
            for (int i = 0; i < 20; i++) {
                EditText editText = new EditText(this);
                editText.setGravity(Gravity.CENTER);
                editTexts[i] = editText;
                mGridLayout.addView(editText, mEtWidth, mEtHeight);
            }
        }

        //初始化矩阵
        private void initMatrix() {
            for (int i = 0; i < 20; i++) {
                if (i % 6 == 0) {
                    editTexts[i].setText(String.valueOf(1));
                } else {
                    editTexts[i].setText(String.valueOf(0));

                }
            }
        }
    }
```

Android图像处理–像素点分析

- **像素点阵**：图像经过放大后会呈现一个个点阵，每个点就是一个像素点，通过空控制RGB的颜色比，就可以显示出不同的颜色。

- **实现思想**：先分离每一个像素点的rgba,对RGBA经过算法处理后，再赋值给像素点。

示例代码
主要方法

```
    //像素点阵
       public static Bitmap handleImageNegative(Bitmap bm) {
           int width = bm.getWidth();
           int height = bm.getHeight();
           int color;
           int r, g, b, a;
           Bitmap bmp = Bitmap.createBitmap(width, height,
   Bitmap.Config.ARGB_8888);

           //保存像素点数组到图像
           int[] oldPx = new int[width * height];

           int[] newPx = new int[width * height];//用来保存变换后的像素点
           bm.getPixels(oldPx, 0, width, 0, 0, width, height);

           for (int i = 0; i < width * height; i++) {
               //取出某一像素点的RGBA
               color = oldPx[i];
               r = Color.red(color);
               g = Color.green(color);
               b = Color.blue(color);
               a = Color.alpha(color);

               //算法处理
               r = 255 - r;
               g = 255 - g;
               b = 255 - b;

               if (r > 255) {
                   r = 255;
               } else if (r < 0) {
                   r = 0;
               }
               if (b > 255) {
                   b = 255;
               } else if (r < 0) {
                   b = 0;
               }
               if (g > 255) {
                   g = 255;
               } else if (r < 0) {
                   g = 0;
               }
```

```
            newPx[i] = Color.argb(a, r, g, b);//将经过算法变换的argb
重新合成

        }
        bmp.setPixels(newPx, 0, width, 0, 0, width, height);
        return bmp;
    }
```

Demo

下载地址