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 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

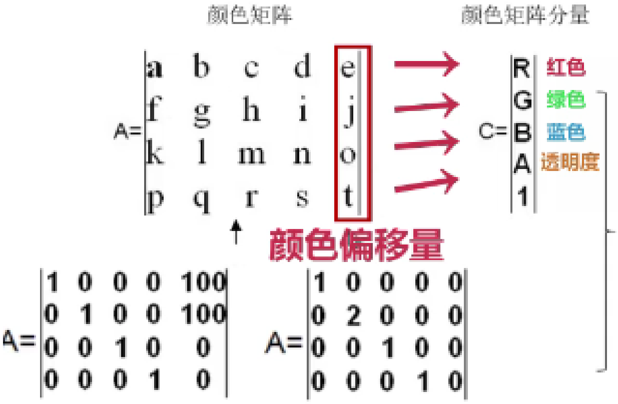
```
package com.flyme.moyu.imageprocess;

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

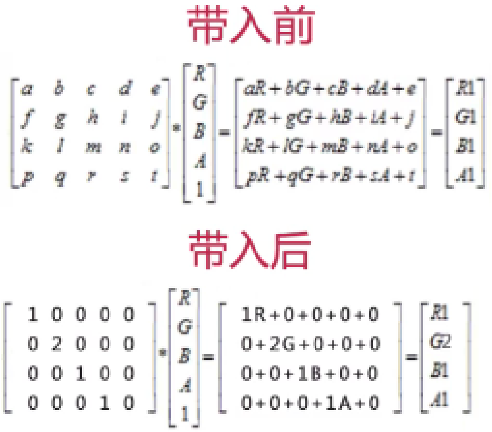
```
package com.flyme.moyu.imageprocess;

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:
                mHun = (i - MID_VALUE) * 1.0f / MID_VALUE * 180;
                break;
            case R.id.seekbarstu:
                mStaurtion = i * 1.0f / MID_VALUE;
                break;
            case R.id.seekbarlun:
                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



$$A=\begin{vmatrix} a & b & c & d & e \\ f & g & h & i & j \\ k & l & m & n & o \\ p & q & r & s & t \end{vmatrix} \qquad C=\begin{vmatrix} R \\ G \\ B \\ A \\ 1 \end{vmatrix}$$

$$A=\begin{vmatrix} 1 & 0 & 0 & 0 & 100 \\ 0 & 1 & 0 & 0 & 100 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{vmatrix} \qquad A=\begin{vmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{vmatrix}$$



$$\begin{bmatrix} a & b & c & d & e \\ f & g & h & i & j \\ k & l & m & n & o \\ p & q & r & s & t \end{bmatrix} * \begin{bmatrix} R \\ G \\ B \\ A \\ 1 \end{bmatrix} = \begin{bmatrix} aR+bG+cB+dA+e \\ fR+gG+hB+iA+j \\ kR+lG+mB+nA+o \\ pR+qG+rB+sA+t \end{bmatrix} = \begin{bmatrix} R1 \\ G1 \\ B1 \\ A1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} * \begin{bmatrix} R \\ G \\ B \\ A \\ 1 \end{bmatrix} = \begin{bmatrix} 1R+0+0+0+0 \\ 0+2G+0+0+0 \\ 0+0+1B+0+0 \\ 0+0+0+1A+0 \end{bmatrix} = \begin{bmatrix} R1 \\ G2 \\ B1 \\ A1 \end{bmatrix}$$