android 图片的浏览、缩放、拖动和自动居中示例

android OpenGL 知识汇总
http://www.eoeandroid.com/thread-203299-1-1.html

检测手机wifi的状态
http://www.eoeandroid.com/thread-203307-1-1.html

GridView + ViewFlipper布局界面，模仿“机锋市场”
http://www.eoeandroid.com/thread-203252-1-1.html

直接上代码吧！文中写了比较详细的注释。

```
/** * 图片浏览、缩放、拖动、自动居中 */
public class ImageTouch extends Activity implements OnTouchListener {
    Matrix matrix = new Matrix();
    Matrix savedMatrix = new Matrix();
    DisplayMetrics dm;
    ImageView imgView;
    Bitmap bitmap;
    float minScaleR;// 最小缩放比例
    static final float MAX_SCALE = 4f;// 最大缩放比例
    static final int NONE = 0;// 初始状态
    static final int DRAG = 1;// 拖动
    static final int ZOOM = 2;// 缩放
    int mode = NONE;
    PointF prev = new PointF();
    PointF mid = new PointF();
    float dist = 1f;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        imgView = (ImageView) findViewById(R.id.imag);// 获取控件
        // bitmap = BitmapFactory.decodeResource(getResources(),
        // this.getIntent()
        // .getExtras().getInt("IMG"));// 获取图片资源
        bitmap = BitmapFactory.decodeResource(getResources(), R.drawable.image);// 获取图片资源
        imgView.setImageBitmap(bitmap);// 填充控件
        imgView.setOnTouchListener(this);// 设置触屏监听
        dm = new DisplayMetrics();
        getWindowManager().getDefaultDisplay().getMetrics(dm);// 获取分辨率
        minZoom();
        center();
        imgView.setImageMatrix(matrix);
    }

    /** * 触屏监听 */
    public boolean onTouch(View v, MotionEvent event) {
        switch (event.getAction() & MotionEvent.ACTION_MASK) { // 主点按下
        case MotionEvent.ACTION_DOWN:
            savedMatrix.set(matrix);
            prev.set(event.getX(), event.getY());
            mode = DRAG;
            break; // 副点按下
        case MotionEvent.ACTION_POINTER_DOWN:
            dist = spacing(event); // 如果连续两点距离大于10，则判定为多点模式
            if (spacing(event) > 10f) {
                savedMatrix.set(matrix);
                midPoint(mid, event);
                mode = ZOOM;
            }
            break;
        case MotionEvent.ACTION_UP:
        case MotionEvent.ACTION_POINTER_UP:
            mode = NONE;
        break;
    case MotionEvent.ACTION_MOVE:
        if (mode == DRAG) {
            matrix.set(savedMatrix);
            matrix.postTranslate(event.getX() - prev.x, event.getY()
                    - prev.y);
        } else if (mode == ZOOM) {
            float newDist = spacing(event);
            if (newDist > 10f) {
                matrix.set(savedMatrix);
                float tScale = newDist / dist;
                matrix.postScale(tScale, tScale, mid.x, mid.y);
            }
        }
        break;
    }
    imgView.setImageMatrix(matrix);
    CheckView();
    return true;
}

/** * 限制最大最小缩放比例，自动居中 */
private void CheckView() {
    float p[] = new float[9];
    matrix.getValues(p);
    if (mode == ZOOM) {
        if (p[0] < minScaleR) {
            matrix.setScale(minScaleR, minScaleR);
        }
        if (p[0] > MAX_SCALE) {
            matrix.set(savedMatrix);
        }
    }
    center();
}

/** * 最小缩放比例，最大为100% */
private void minZoom() {
    minScaleR = Math.min(
            (float) dm.widthPixels / (float) bitmap.getWidth(),
            (float) dm.heightPixels / (float) bitmap.getHeight());
    if (minScaleR < 1.0) {
        matrix.postScale(minScaleR, minScaleR);
    }
}

private void center() {
    center(true, true);
}

/** * 横向、纵向居中 */
protected void center(boolean horizontal, boolean vertical) {
    Matrix m = new Matrix();
    m.set(matrix);
    RectF rect = new RectF(0, 0, bitmap.getWidth(), bitmap.getHeight());
    m.mapRect(rect);
    float height = rect.height();
    float width = rect.width();
    float deltaX = 0, deltaY = 0;
    if (vertical) {
        // 图片小于屏幕大小，则居中显示。大于屏幕，上方留空则往上移，下放留空则往下移
        int screenHeight = dm.heightPixels;
        if (height < screenHeight) {
            deltaY = (screenHeight - height) / 2 - rect.top;
        } else if (rect.top > 0) {
            deltaY = -rect.top;
        } else if (rect.bottom < screenHeight) {
            deltaY = imgView.getHeight() - rect.bottom;
        }
    }
```

```
        if (horizontal) {
            int screenWidth = dm.widthPixels;
            if (width < screenWidth) {
                deltaX = (screenWidth - width) / 2 - rect.left;
            } else if (rect.left > 0) {
                deltaX = -rect.left;
            } else if (rect.right < screenWidth) {
                deltaX = screenWidth - rect.right;
            }
        }
        matrix.postTranslate(deltaX, deltaY);
    }

    /** * 两点的距离 */
    private float spacing(MotionEvent event) {
        float x = event.getX(0) - event.getX(1);
        float y = event.getY(0) - event.getY(1);
        return FloatMath.sqrt(x * x + y * y);
    }

    /** * 两点的中点 */
    private void midPoint(PointF point, MotionEvent event) {
        float x = event.getX(0) + event.getX(1);
        float y = event.getY(0) + event.getY(1);
        point.set(x / 2, y / 2);
    }
}
```

main.xml

```
<?xml version="1.0" encoding="utf-8"?>
    <FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
        android:layout_width="fill_parent" android:layout_height="fill_parent"
        android:layout_gravity="center">
        <ImageView android:id="@+id/imag" android:layout_width="fill_parent"
            android:layout_height="fill_parent" android:layout_gravity="center"
            android:scaleType="matrix">
        </ImageView>
    </FrameLayout>
```