

【CodeLab】HarmonyOS 基于图像模块实现图库图片的四种常见操作

本期我们将为大家带来最常见的图像业务开发。

官网链接：

<https://developer.harmonyos.com/cn/docs/documentation/doc-guides/cookbook-image-processing-0000001089174695>

在 HarmonyOS 中，由图像模块支持图像业务的开发，常见功能如图像解码、图像编码、基本的位图操作、图像编辑等。当然，开发者也可以通过接口组合来实现更复杂的图像处理逻辑。

在这个教程中，我们将以图库图片中旋转、剪裁、缩放、镜像四种常见操作为例，给大家介绍 HarmonyOS 图像编解码的相关开发指导。

在 HUAWEI DevEco Studio 创建一个 Phone 的 Empty Feature Ability(Java)模板工程，我们将在这个模板中完成操作。

我们需要先准备一张 1024*768 的 JPG 格式图片作为示例，放在我们创建的手机工程项目中的 entry/src/main/resources/base/media 目录下。



1、将图片转换为 PixelMap 对象

为了便于后续图片显示及处理，我们需要对图片进行解码处理。图像解码就是将所支持格式的存档图片解码成统一的 PixelMap 图像。当前 HarmonyOS 支持的图片格式包括 JPEG、PNG、GIF、HEIF、WebP、BMP。

这个例子中我们提供的是 `getPixelMapFromResource` 函数，可以将已经存放在相关目录下的图片资源转换为 PixelMap 图像，其中入参为图片的资源 ID，

示例代码如下：

```
private PixelMap getPixelMapFromResource(int resourceId) {  
  
    InputStream inputStream = null;
```

来源：HarmonyOS 开发者微信号 <https://mp.weixin.qq.com/s/bd2T4KUBNShmWJf7w3G0Cw>

```

try {

    // 创建图像数据源 ImageSource 对象

    inputStream                                     =
getContext().getResourceManager().getResource(resourceId);

    ImageSource.SourceOptions srcOpts = new ImageSource.SourceOptions();

    srcOpts.formatHint = "image/jpg";

    ImageSource imageSource = ImageSource.create(inputStream, srcOpts);


    // 设置图片参数

    ImageSource.DecodingOptions decodingOptions = new
ImageSource.DecodingOptions();

    return imageSource.createPixelMap(decodingOptions);

} catch (IOException e) {

    HiLog.info(LABEL_LOG, "IOException");

} catch (NotExistException e) {

    HiLog.info(LABEL_LOG, "NotExistException");

} finally {

    if (inputStream != null) {

        try {

            inputStream.close();

        } catch (IOException e) {

            HiLog.info(LABEL_LOG, "inputStream IOException");

```

```

        }

    }

}

return null;

}

```

2、图片旋转、缩放、裁剪参数设置

这个步骤中，我们将对图片进行旋转，缩放，裁剪处理。本 Codelab 中的图像处理是基于 1024*768 的图像尺寸，实现如下功能：

旋转：点击一次旋转按钮将进行 90 度旋转；

缩放：点击缩放按钮将按 2:1 比例进行缩放，再次点击还原；

裁剪：点击一次裁剪按钮将在保证宽度不变的情况下对高度进行 400 像素的剪裁；

相关参数设置如下所示：

```

// 设置图片参数

ImageSource.DecodingOptions decodingOptions = new
ImageSource.DecodingOptions();

// 旋转

decodingOptions.rotateDegrees = 90 * whirlCount;

// 缩放

decodingOptions.desiredSize = new Size(isScale ? 512 : 0, isScale ? 384 : 0);

// 剪裁

```

```
decodingOptions.desiredRegion = new Rect(0, 0, isCorp ? 1024 : 0, isCorp ? 400 : 0);
```

开发者也可根据实际需求对参数进行修改。

3、图片镜像操作处理

除了旋转、缩放、裁剪等操作，常见的图片操作处理还有镜像。所谓图片镜像，就是对图片以纵坐标为轴制作对称图片。本例采用对图像 Canvas 画布的镜像操作实现图片的镜像显示。

示例代码如下所示：

```
private void mirrorImage(PixelMap pixelMap) {  
  
    scaleX = -scaleX;  
  
    image.addDrawTask(  
  
        new Component.DrawTask() {  
  
            @Override  
  
            public void onDraw(Component component, Canvas canvas) {  
  
                if (isMirror) {  
  
                    isMirror = false;  
  
                    PixelMapHolder pmh = new PixelMapHolder(pixelMap);  
  
                    canvas.scale(  
  
                        scaleX,  
  
                        1.0f,  
  
                        (float) pixelMap.getImageInfo().size.width / 2,
```

```
                (float) pixelMap.getImageInfo().size.height / 2);

        canvas.drawPixelMapHolder(

            pmh,

            0,

            0,

            new Paint());

    }

}

});

}
```

4、创建界面基础组件

这个部分主要针对我们需要展示的界面文本和按钮进行创建,我们需要创建的内容有界面标题文本——“HarmonyOS 图像开发”,四个对应不同图片处理的按钮——“旋转”、“裁剪”、“缩放”、“镜像”。

标题文本创建:

```
<Text

    ohos:height="match_content"

    ohos:width="match_content"

    ohos:layout_alignment="horizontal_center"

    ohos:text="HarmonyOS 图像开发"
```

```
ohos:text_size="80"

ohos:top_margin="40vp"

/>
```

<DirectionalLayout

```
xmlns:ohos="http://schemas.huawei.com/res/ohos"

ohos:height="match_content"

ohos:width="match_content"

ohos:layout_alignment="horizontal_center"

ohos:orientation="horizontal"

ohos:top_margin="20vp">
```

“旋转” 按钮绘制:

<Button

```
ohos:id="$+id:whirl_image"

ohos:height="match_content"

ohos:width="match_content"

ohos:background_element="$graphic:background_button"

ohos:padding="12vp"

ohos:right_margin="5vp"

ohos:text="旋转"
```

```
        ohos:text_size="20vp"

        ohos:top_margin="10vp">

</Button>
```

“裁剪” 按钮绘制：

```
<Button

        ohos:id="$+id:crop_image"

        ohos:height="match_content"

        ohos:width="match_content"

        ohos:background_element="$graphic:background_button"

        ohos:left_margin="5vp"

        ohos:padding="12vp"

        ohos:text="剪裁"

        ohos:text_size="20vp"

        ohos:top_margin="10vp">

</Button>
```

“缩放” 按钮绘制：

```
<Button

        ohos:id="$+id:scale_image"

        ohos:height="match_content"

        ohos:width="match_content"
```



```
        ohos:background_element="$graphic:background_button"

        ohos:left_margin="5vp"

        ohos:padding="12vp"

        ohos:text="缩放"

        ohos:text_size="20vp"

        ohos:top_margin="10vp">

</Button>
```

“镜像” 按钮绘制：

```
<Button

        ohos:id="$+id:mirror_image"

        ohos:height="match_content"

        ohos:width="match_content"

        ohos:background_element="$graphic:background_button"

        ohos:left_margin="5vp"

        ohos:padding="12vp"

        ohos:text="镜像"

        ohos:text_size="20vp"

        ohos:top_margin="10vp"/>

</Button>
```

图片位置：

<Image

ohos:id="\$+id:image"

ohos:height="match_content"

ohos:width="match_content"

ohos:image_src="\$media:shanghai.jpg"

ohos:layout_alignment="horizontal_center"

ohos:top_margin="20vp">

</Image

最终实现如下效果：



需要注意的是：以上代码仅 demo 演示参考使用，产品化的代码需要考虑数据校验和国际
化。