

yuv2bitamp

在 Android 中，当使用相机（例如 `Camera2 API`）捕获图像时，图像通常以 `YUV_420_888` 格式传输。将 `YUV_420_888` 格式的图像转换为 `Bitmap` 需要将 YUV 格式转换为 RGB 格式。

你可以通过以下步骤实现转换：

1. 获取 `YUV_420_888` 的三个平面数据

`YUV_420_888` 格式有三个平面，分别对应 Y、U、V 分量。你需要将它们从 `Image` 对象中提取出来。

```
import android.graphics.Bitmap
import android.graphics.ImageFormat
import android.media.Image
import java.nio.ByteBuffer

fun imageToBitmap(image: Image): Bitmap? {
    val planes = image.planes
    val width = image.width
    val height = image.height

    // YUV 数据的缓冲区
    val yBuffer: ByteBuffer = planes[0].buffer // Y
    val uBuffer: ByteBuffer = planes[1].buffer // U
    val vBuffer: ByteBuffer = planes[2].buffer // V

    val ySize = yBuffer.remaining()
    val uSize = uBuffer.remaining()
    val vSize = vBuffer.remaining()

    // YUV 数据存储为字节数组
    val nv21 = ByteArray(ySize + uSize + vSize)

    // 将缓冲区中的数据复制到字节数组中
    yBuffer.get(nv21, 0, ySize)
    vBuffer.get(nv21, ySize, vSize)
    uBuffer.get(nv21, ySize + vSize, uSize)

    // 转换为 NV21 格式的字节数组
    return nv21ToBitmap(nv21, width, height)
}
```

2. 将 NV21 转换为 Bitmap

Android 提供了将 `NV21` 格式（一个与 `YUV_420_888` 非常类似的格式）转换为 `Bitmap` 的工具。

你可以使用 `YuvImage` 来将 `NV21` 数据转换为 `Bitmap`。

```
import android.graphics.Bitmap
import android.graphics.BitmapFactory
import android.graphics.ImageFormat
import android.graphics.YuvImage
import java.io.ByteArrayOutputStream

fun nv21ToBitmap(nv21: ByteArray, width: Int, height: Int): Bitmap? {
    val yuvImage = YuvImage(nv21, ImageFormat.NV21, width, height, null)
    val out = ByteArrayOutputStream()
    yuvImage.compressToJpeg(android.graphics.Rect(0, 0, width, height),
100, out)
    val imageBytes: ByteArray = out.toByteArray()
    return BitmapFactory.decodeByteArray(imageBytes, 0, imageBytes.size)
}
```

3. 示例调用

通过 `Camera2 API` 获取的 `Image` 可以通过上述方法转换为 `Bitmap`：

```
val bitmap = imageToBitmap(image)
```

总结

- `YUV_420_888` 格式包含 Y、U、V 三个平面。
- 你可以将其转换为 `NV21` 格式的字节数组。
- 使用 `YuvImage` 将 `NV21` 数据转换为 `Bitmap`。

这种方法简单直接，适用于从 `Camera2 API` 的 `Image` 获取图像并将其转化为 `Bitmap` 的场景。