DESCRIPTION an...@gmail.com created issue #1 May 28, 1

Device: Nexus 9
API: Camera2
Version: 5.0.1
Build: LRX22C

When extracting YUV_420_888 data from the Nexus 9 camera stream, I only get acceptable data at two resolutions (2048 x 1536 and 1280 x 720) although getOutputSize() indicates I should hav available resolutions, including full 8MP resolution.

I am taking the Y component to create monochrome images. When I use any other resolution other than above, the image turns out corrupt. See attached .bmp.

deleted
0 B

Links (1)

" ... image you've attached looks like you haven't correctly accounted for the stride of the image buffer (https://developer.android.com/reference/android/media/Image.Plane.html#getRowStride())."

COMMENTS All comments

et...@google.com <et...@google.com> #2 May 31, 2(

Do you have the code for extracting the Y channel available for inspection?
The sample image you've attached looks like you haven't correctly accounted for the stride of the image buffer (https://developer.android.com/reference/android/media/Image.Plane.html#g

an...@gmail.com <an...@gmail.com> #3 Jun 1, 2(

The code is:

```
ByteBuffer buffer = mImage.getPlanes()[0].getBuffer();
final byte[] bytes = new byte[buffer.remaining()];
buffer.get(bytes);
```

I had thought that the first plane of YUV was the Y-component which was passed to JNI to save as a monochrome BMP through byte[] bytes. mImage is obtained through ImageReader.acqui

an...@gmail.com <an...@gmail.com> #4 Jun 1, 2(

The same code gives me the attached image here at 2048 x 1536. How does row stride play into this? Since the Y component is 1 byte per pixel, then my resolution width should tell me the r correct?

deleted
0 B

an...@gmail.com <an...@gmail.com> #5 Jun 1, 2(

You were 100% correct. By taking account the rowStride I was able to successfully save the data. Thanks!

et...@google.com <et...@google.com> #6 Jun 5, 2(

Status: Won't Fix (Intended Behavior)

No problem, glad it was a simple-to-fix issue.
For more context, while the stride does confuse a lot of people, it's necessary to ensure that hardware blocks can efficiently write the image data (often there are restrictions on stride due to mapping works in the hardware processing blocks).