Output streams and cropping

Output streams

Unlike the old camera subsystem, which has 3-4 different ways of producing data from the camera (ANativeWindow-based preview operations, preview callbacks, video callbacks, and takePicture callbacks), the new subsystem operates solely on the ANativeWindow-based pipeline for all resolutions and output formats. Multiple such streams can be configured at once, to send a single frame to many targets such as the GPU, the video encoder, RenderScript, or app-visible buffers (RAW Bayer, processed YUV buffers, or JPEG-encoded buffers).

As an optimization, these output streams must be configured ahead of time, and only a limited number may exist at once. This allows for pre-allocation of memory buffers and configuration of the camera hardware, so that when requests are submitted with multiple or varying output pipelines listed, there won't be delays or latency in fulfilling the request.

To support backwards compatibility with the current camera API, at least 3 simultaneous YUV output streams must be supported, plus one JPEG stream. This is required for video snapshot support with the application also receiving YUV buffers:

- One stream to the GPU/SurfaceView (opaque YUV format) for preview
- One stream to the video encoder (opaque YUV format) for recording
- One stream to the application (known YUV format) for preview frame callbacks
- One stream to the application (JPEG) for video snapshots.

The exact requirements are still being defined since the corresponding API isn't yet finalized.

Cropping

Cropping of the full pixel array (for digital zoom and other use cases where a smaller FOV is desirable) is communicated through the ANDROID_SCALER_CROP_REGION setting. This is a per-request setting, and can change on a per-request basis, which is critical for implementing smooth digital zoom.

The region is defined as a rectangle (x, y, width, height), with (x, y) describing the top-left corner of the rectangle. The rectangle is defined on the coordinate system of the sensor active pixel array, with (0,0) being the top-left pixel of the active pixel array. Therefore, the width and height cannot be larger than the dimensions reported in the ANDROID_SENSOR_ACTIVE_PIXEL_ARRAY static info field. The minimum allowed width and height are reported by the HAL through the ANDROID_SCALER_MAX_DIGITAL_ZOOM static info field, which describes the maximum supported zoom factor. Therefore, the minimum crop region width and height are:

```
{width, height} =
 { floor(ANDROID_SENSOR_ACTIVE_PIXEL_ARRAY[0] /
     ANDROID_SCALER_MAX_DIGITAL_ZOOM),
   floor(ANDROID_SENSOR_ACTIVE_PIXEL_ARRAY[1] /
     ANDROID_SCALER_MAX_DIGITAL_ZOOM) }
```

If the crop region needs to fulfill specific requirements (for example, it needs to start on even coordinates, and its width/height needs to be even), the HAL must do the necessary rounding and write out the final crop region used in the output result metadata. Similarly, if the HAL implements video stabilization, it must adjust the result crop region to describe the region actually included in the output after video stabilization is applied. In general, a

camera-using application must be able to determine the field of view it is receiving based on the crop region, the dimensions of the image sensor, and the lens focal length.

Since the crop region applies to all streams, which may have different aspect ratios than the crop region, the exact sensor region used for each stream may be smaller than the crop region. Specifically, each stream should maintain square pixels and its aspect ratio by minimally further cropping the defined crop region. If the stream's aspect ratio is wider than the crop region, the stream should be further cropped vertically, and if the stream's aspect ratio is narrower than the crop region, the stream should be further cropped horizontally.

In all cases, the stream crop must be centered within the full crop region, and each stream is only either cropped horizontally or vertical relative to the full crop region, never both.

For example, if two streams are defined, a 640x480 stream (4:3 aspect), and a 1280x720 stream (16:9 aspect), below demonstrates the expected output regions for each stream for a few sample crop regions, on a hypothetical 3 MP (2000 x 1500 pixel array) sensor.

Crop region: (500, 375, 1000, 750) (4:3 aspect ratio)
640x480 stream crop: (500, 375, 1000, 750) (equal to crop region)
1280x720 stream crop: (500, 469, 1000, 562)

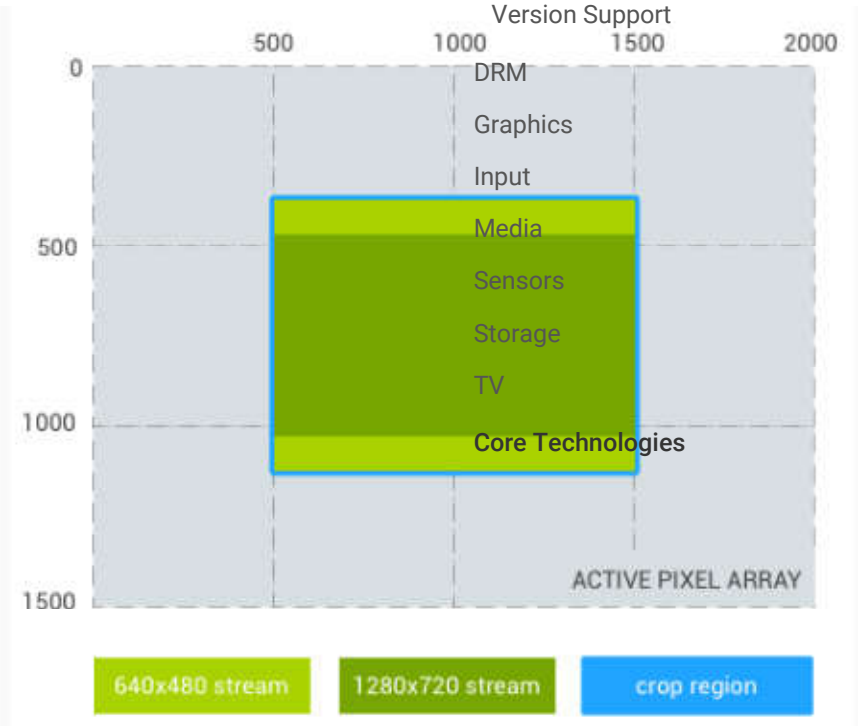

Figure 1. 4:3 aspect ratio

Crop region: (500, 375, 1333, 750) (16:9 aspect ratio)
640x480 stream crop: (666, 375, 1000, 750)
1280x720 stream crop: (500, 375, 1333, 750) (equal to crop region)

Interfaces
Accessories
Audio
Bluetooth
Camera
Camera HAL3
HAL Subsystem
Metadata and Controls
3A Modes and State
Output and Cropping
Errors and Streams
Request Creation
Version Support
DRM
Graphics
Input
Media
Sensors
Storage
TV
Core Technologies

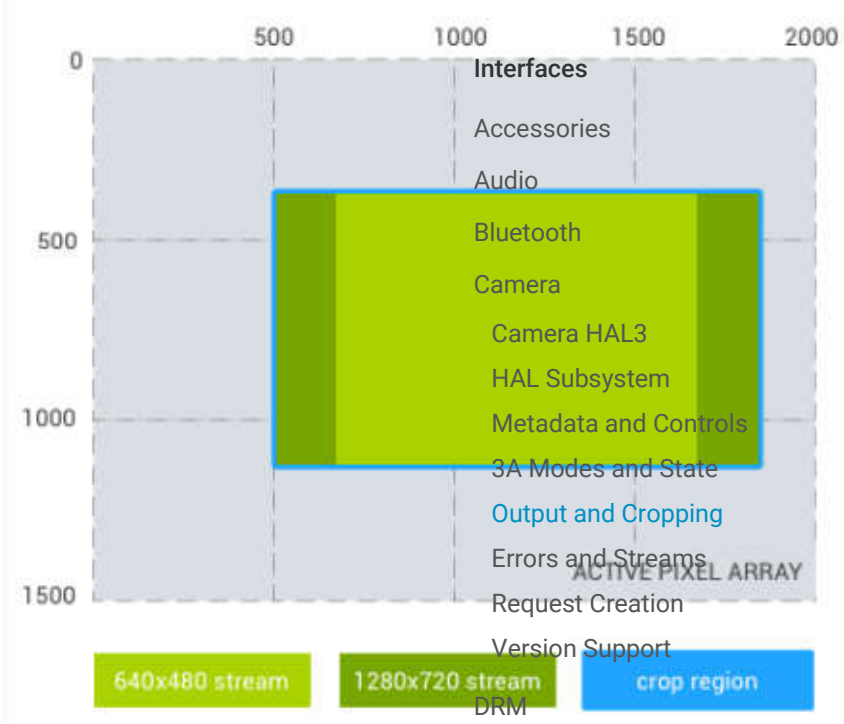

Figure 2. 16:9 aspect ratio

Crop region: (500, 375, 750, 750) (1:1 aspect ratio)
640x480 stream crop: (500, 469, 750, 562)
1280x720 stream crop: (500, 543, 750, 414)

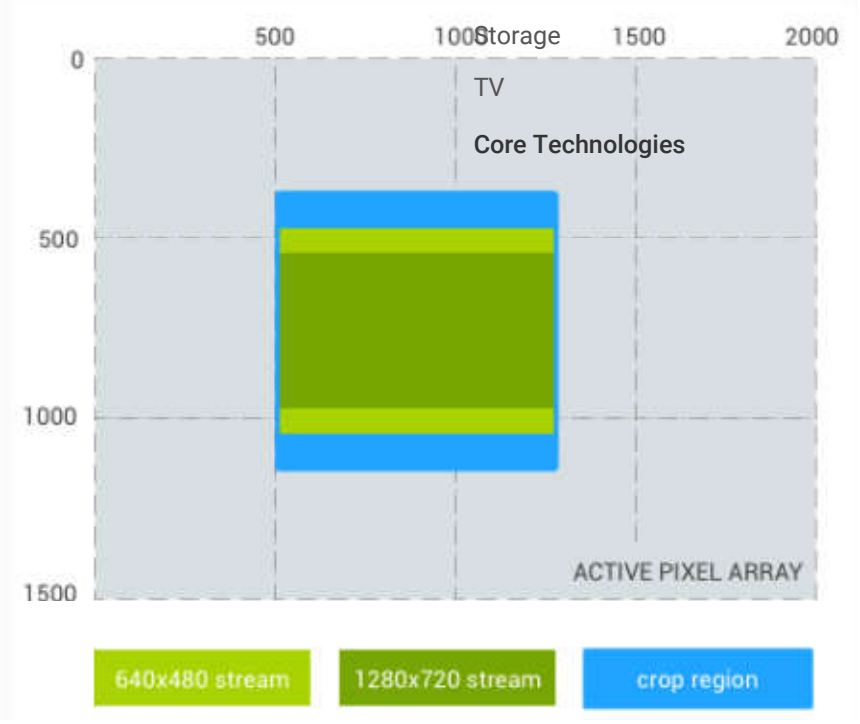

Figure 3. 1:1 aspect ratio

And a final example, a 1024x1024 square aspect ratio stream instead of the 480p stream:
Crop region: (500, 375, 1000, 750) (4:3 aspect ratio)
1024x1024 stream crop: (625, 375, 750, 750)
1280x720 stream crop: (500, 469, 1000, 562)

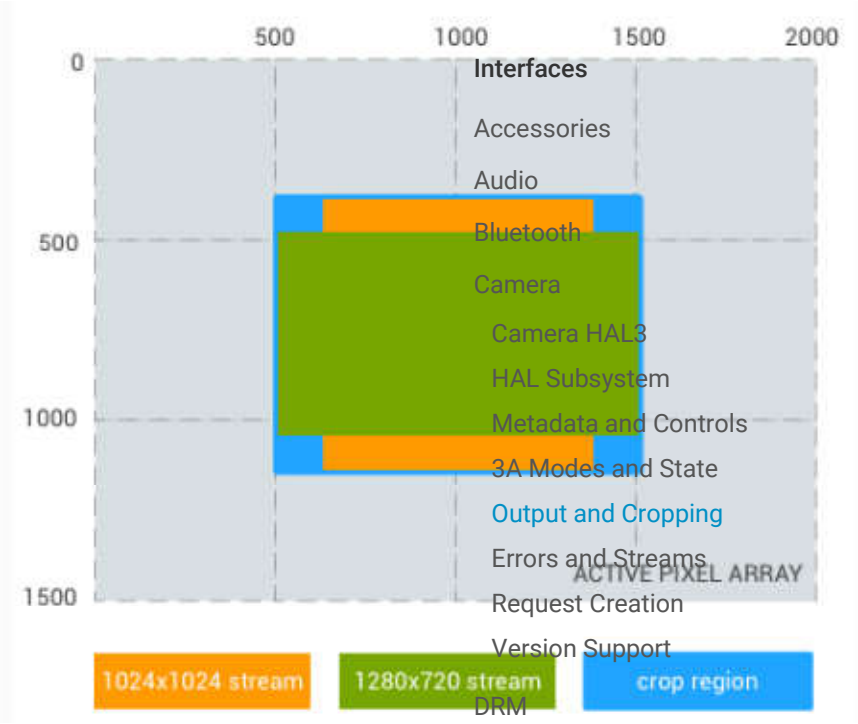

Figure 4. 4:3 aspect ratio, square

Reprocessing

Additional support for raw image files is provided by reprocessing support for RAW Bayer data. This support allows the camera pipeline to process a previously captured RAW buffer and metadata (an entire frame that was recorded previously), to produce a new rendered YUV or JPEG output.