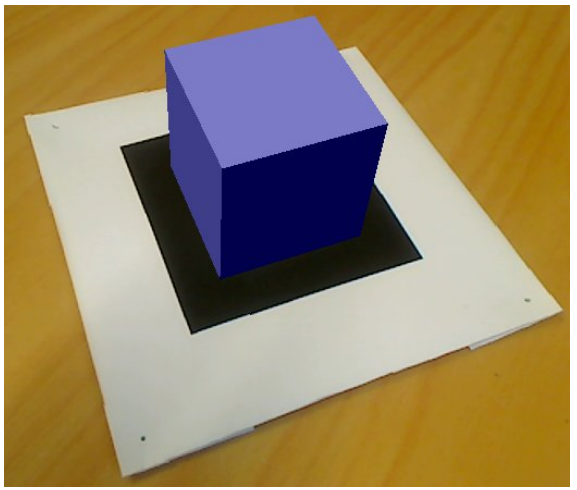
Why camera?

- Kind of light sensor :)
- Take pictures, record video
- Background image for AR
  - Layar
- Input for computer vision/image processing
  - Pose estimation
  - Heartbeat
  - Spectrometer



touchqode

Outline

- Why is camera important
- Camera API
- Basic usage
- Camera in emulator
- Image formats

touchqode

Camera API

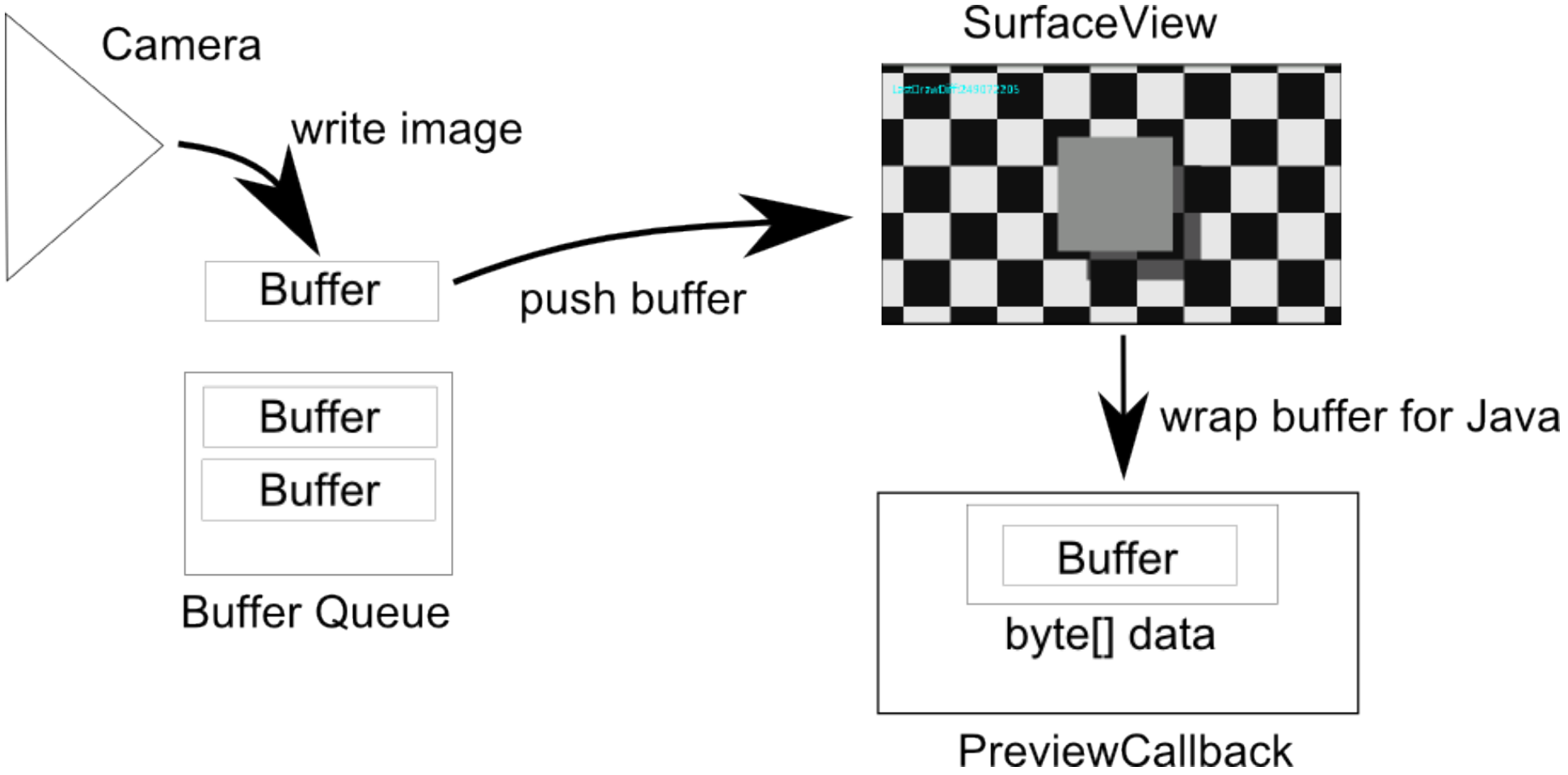
- Permissions

```
<uses-permission android:name="android.permission.CAMERA" />
<uses-feature android:name="android.hardware.camera" />
```

- import android.hardware.Camera;
- Open/release
  - Constructor is private => Camera.open()
- Where is display?
  - Create your own extending SurfaceView
  - Add to activity using setContentView

touchqode

Image pipeline



touchqode

Typical usage

```
public class ARCameraPreview extends SurfaceView implements SurfaceHolder.Callback {
    SurfaceHolder mHolder;
    Camera mCamera;

public ARCameraPreview(Context context) {
    super(context);

    mHolder = getHolder();

    mHolder.addCallback(this);

    mHolder.setType(
                SurfaceHolder.SURFACE_TYPE_PUSH_BUFFERS
            );
}
```

get instance of SurfaceHolder

register for callback
on surface create and destroy

set buffer type to
SURFACE_TYPE_PUSH_BUFFERS

touchqode

Typical usage (contd.)

called once the surface was fully initialized

```
public void surfaceCreated(SurfaceHolder holder) {
    mCamera = Camera.open();

    try {
        mCamera.setPreviewDisplay(holder);
    } catch (IOException exception) {
        mCamera.release();
        mCamera = null;
    }
}
```

get instance of camera

set fully initialized surface to camera

touchqode

Typical usage (contd.)

called when surface size changes

```
public void surfaceChanged(SurfaceHolder holder, int format,
        int width, int height) {

    mCamera.stopPreview();
    Camera.Parameters parameters = mCamera.getParameters();
    parameters.setPreviewSize(width, height);
    mCamera.setParameters(parameters);

    mCamera.startPreview();
}
```

stop preview just to be sure

set width&height
for the camera picture

tell camera to start
sending frames to surface

touchqode

Typical usage (contd.)

called when surface is destroyed

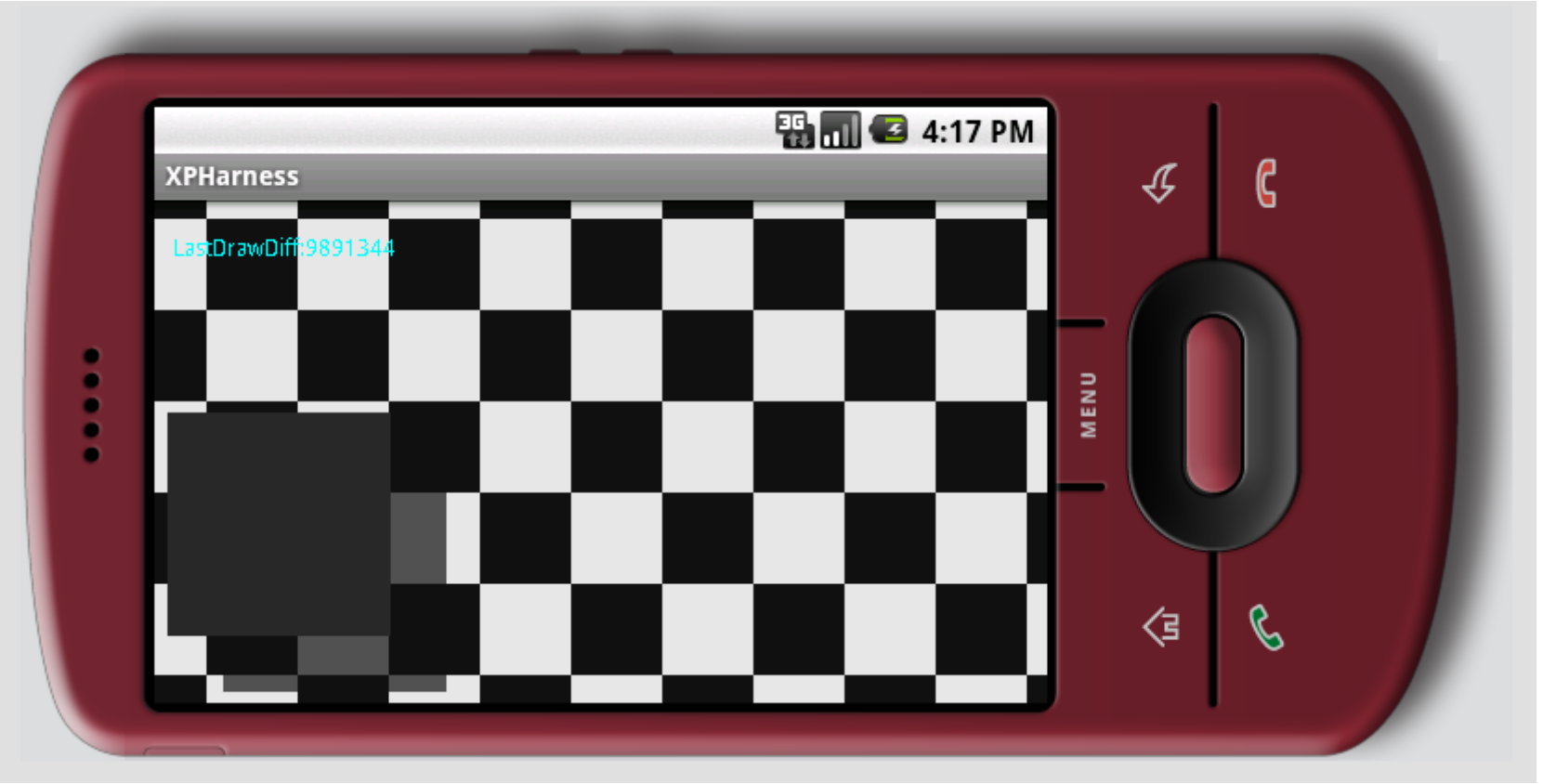
```
public void surfaceDestroyed(
        SurfaceHolder holder) {

    mCamera.stopPreview();
    mCamera.release();
    mCamera = null;
}
```

stop preview

important!
camera is shared resource

touchqode

Demo



touchqode

Webcam in emulator

- Don't do that
- Emulator
  - Infinite sample loop (box over checkers)
  - Custom SocketCamera
    - Video input from webcam is sent to server on emulator
    - Really slow
  - Only for quick prototype
- Real phone is much better

touchqode

Processing camera input

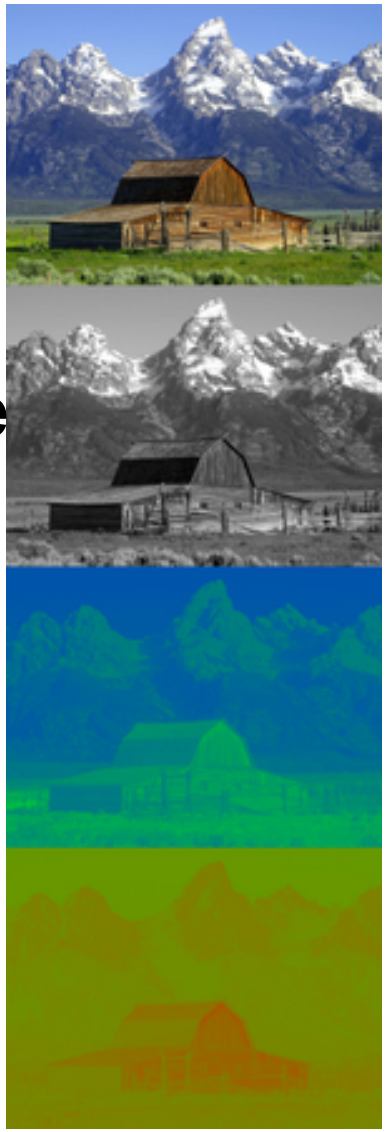
- Before Froyo (Android 1.0-2.1)
  - setPreviewCallback → onPreviewFrame

```
mCamera.setPreviewCallback(new Camera.PreviewCallback() {
  @Override
  public void onPreviewFrame(byte[] data, Camera camera) {
      // do something with data
  }
});
```

- Froyo and beyond (Android 2.2)
  - Create&reuse your own buffer - addCallbackBuffer
  - setPreviewCallback**WithBuffer** → onPreviewFrame
  - After processing data from buffer add it back to queue using addCallbackBuffer

What is in byte[] data?

- Image byte by byte
- PixelFormat – YUV/NV21, RGB565
  - Luckily many algorithms work in grayscale
- YUV (YCbCr_420_SP/NV21)



Single Frame YUV420:

Position in byte stream:

What is in byte[] data? (contd.)

- YUV (YCbCr_420_SP/NV21)
  - Default format (on older versions the only available)
  - Full greyscale (people are more sensitive)
    - First width*height bytes
  - Subsampled blue and red (chroma)
  - Use greyscale - first width*height bytes – and don't bother with the rest

What is in byte[] data? (contd.)

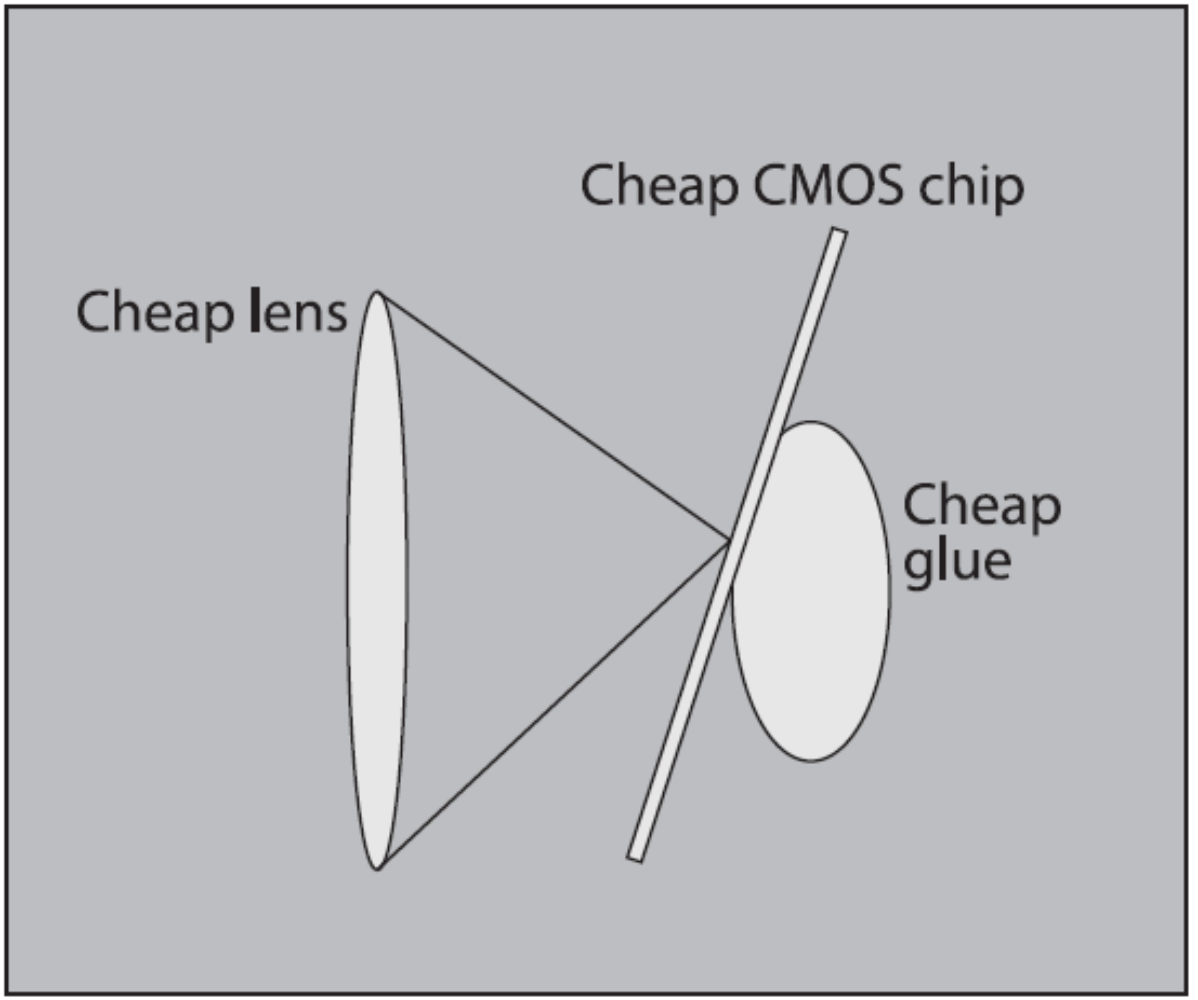
- RGB565 – 16 bits – 2 bytes per pixel

5 6 5

- RRRRRGGG|GGGBBBBB

data[2n] data[2n+1]

What is camera actually?



touchqode

Issues

- Slow – 15 ps
- Blurred at low light
- Narrower angle of view
- Rolling shutter (no global shutter) – skewed image

touchqode

Issues (contd.)

- Getting image from sensor takes time – smaller time slot for calculations
  - Caution - onPreviewFrame called too many times when you do much processing => setOneShotPreviewCallback
- Distortion

More fun with camera

- Do something with computer vision
  - We worked with OpenCV on android
    - Works but be careful about performance
  - Edge detection
  - Feature tracking
  - Face detection
- Motion detection
- Daylight sensor
- Whatever you can think of :)

touchqode

Use the source, Luke!

- http://android.git.kernel.org
  is your friend

- Camera:
  - [platform/frameworks/base.git]
    /core/java/android/hardware
    /Camera.java
  - [platform/frameworks/base.git]
    /jni/android_hardware_Camera.cpp

touchqode