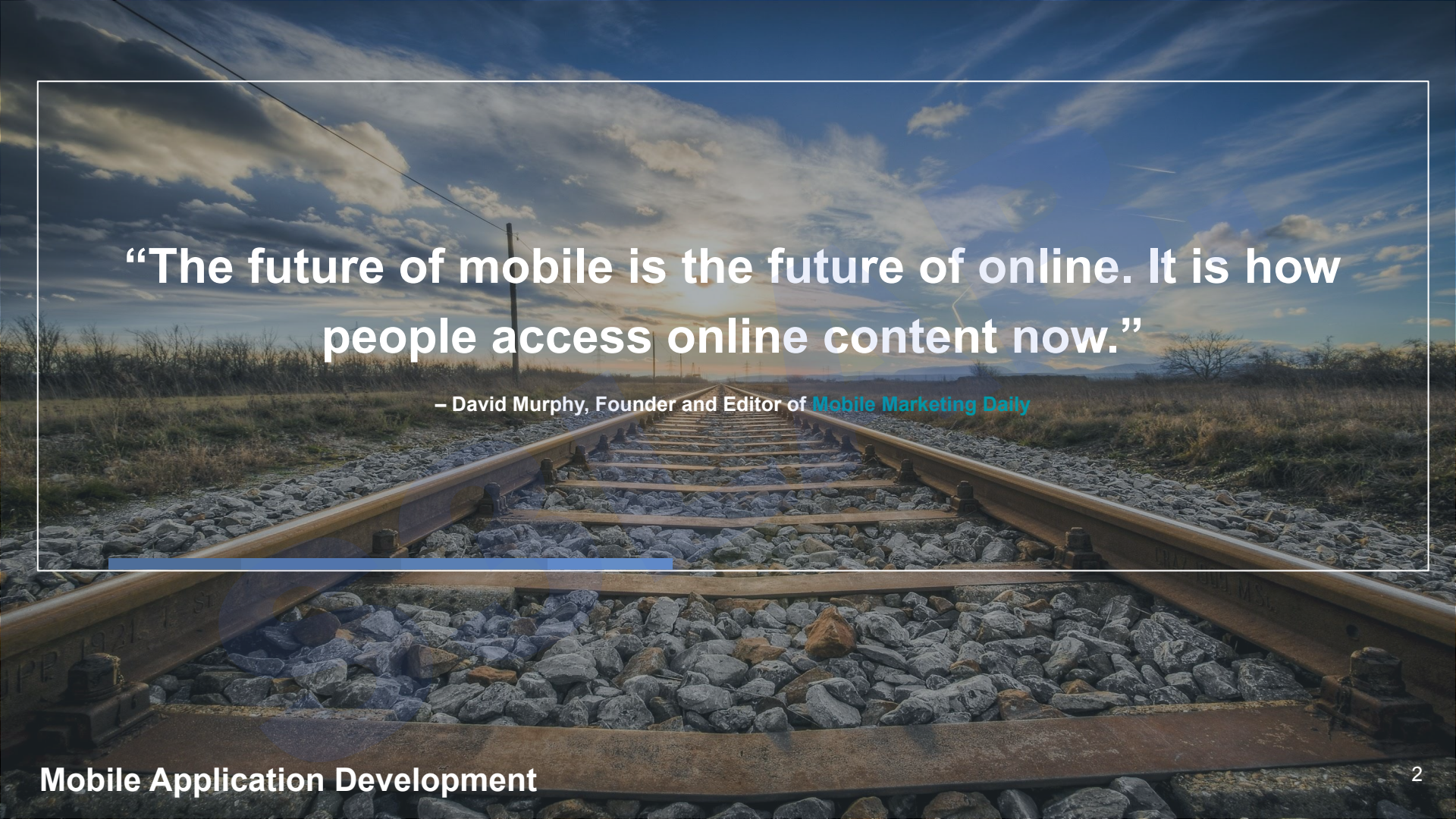
Mobile Application Development

(Sensors)

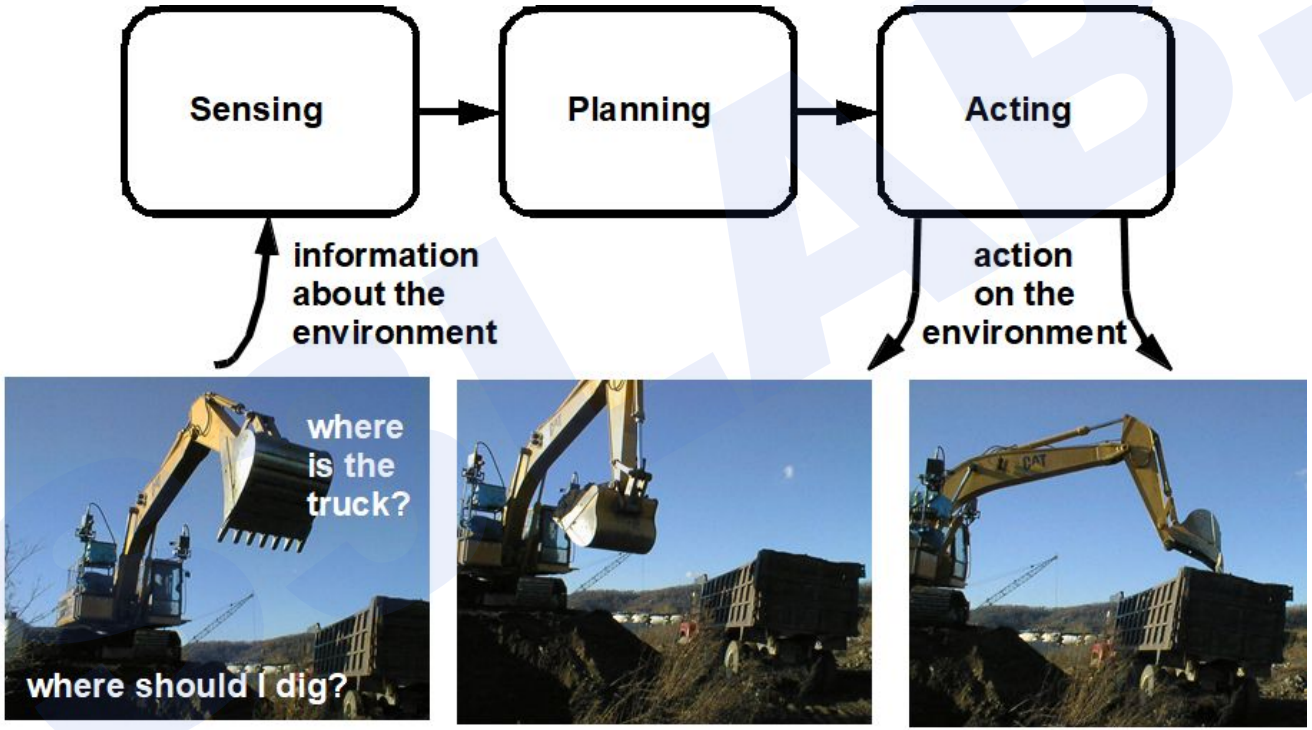
Instructor: Thanh Binh Nguyen

February 1st, 2020

S3Lab
Smart Software System Laboratory


“The future of mobile is the future of online. It is how people access online content now.”
– David Murphy, Founder and Editor of Mobile Marketing Daily

Sensors


Sensing
Planning
Acting
information about the environment
action on the environment
where is the truck?
where should I dig?

Sensors

- **Sensing**: Collect information about the world
- **Sensor** - an electrical/mechanical/chemical device that maps an environmental attribute to a quantitative measurement
  - attribute mixtures - often no one to one map
  - hidden state in environment
- Each sensor is based on a **transduction principle** - conversion of energy from one form to another
- Also known as **transducers**

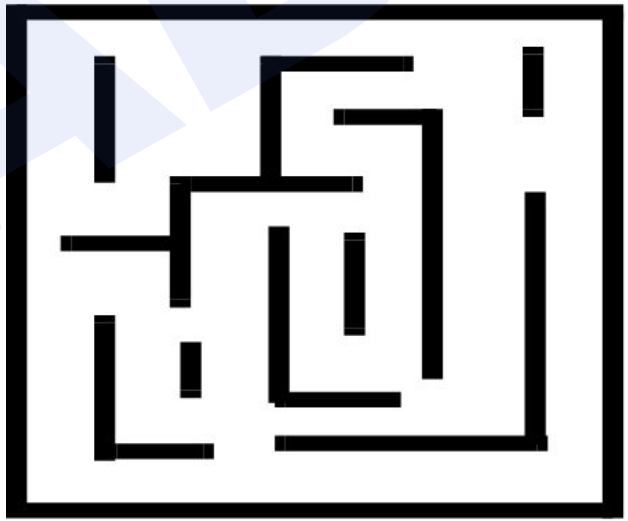
Sensors

Examples

Will I hit anything?

obstacle detection

Where am I?



localization

Sensors

Examples

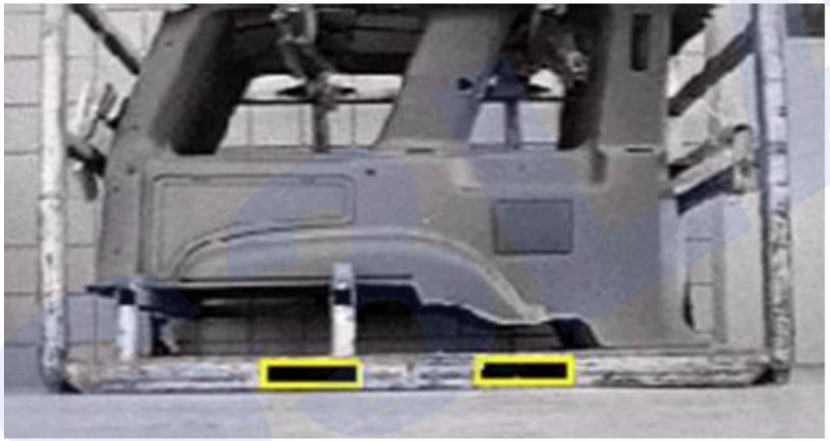
Where is the cropline?

Autonomous harvesting

Sensors

Examples

Where are the forkholes?



Autonomous material handling

Sensors

Examples

Sensors

Type

- **Active**
  - send signal into environment and measure interaction of signal environment
  - e.g. radar, sonar
- **Passive**
  - record signals already present in environment
  - e.g. video cameras

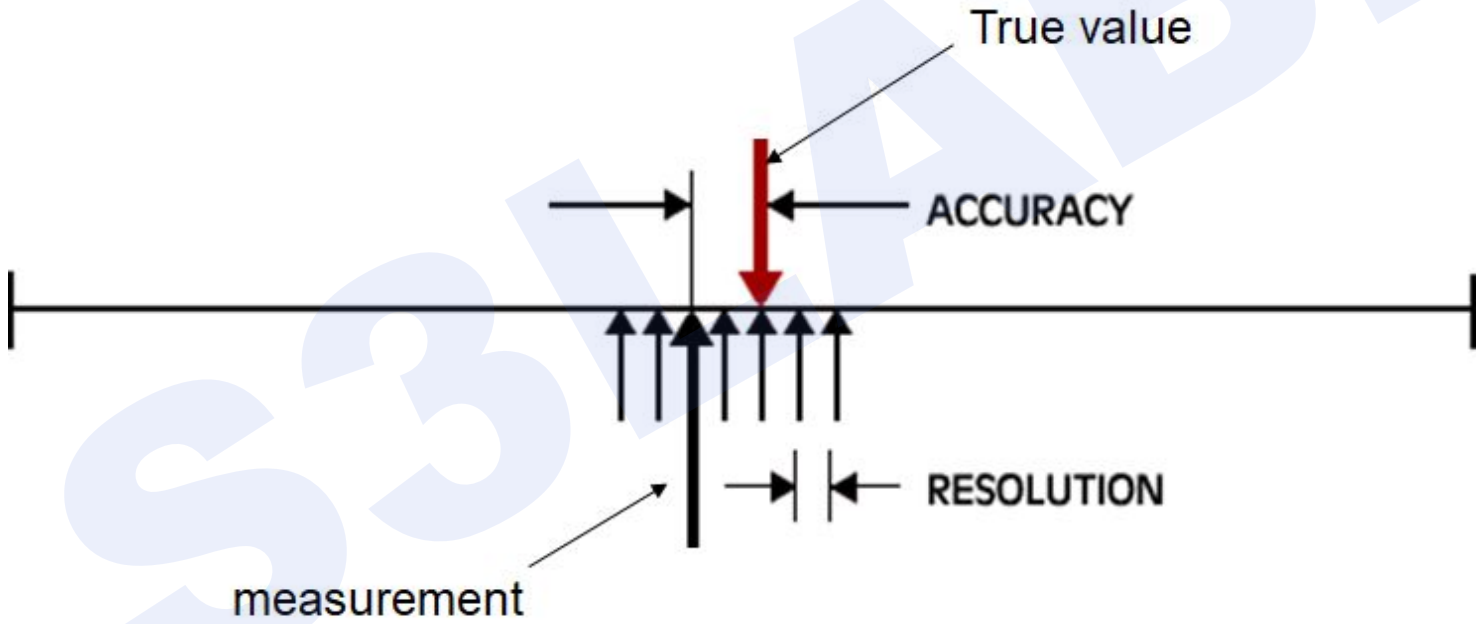
Sensors

Specifications

- **Accuracy**: error between the result of a measurement and the true value being measured.
- **Resolution**: the smallest increment of measure that a device can make.
- **Sensitivity**: the ratio between the change in the output signal to a small change in input physical signal. Slope of the input-output fit line.
- **Repeatability/Precision**: the ability of the sensor to output the same value for the same input over a number of trials

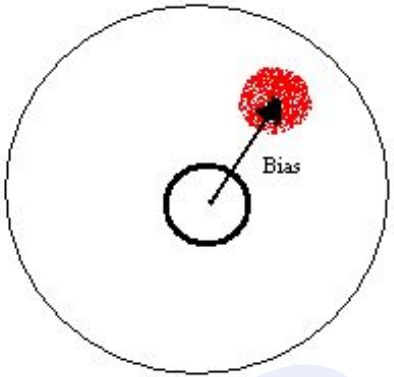
Sensors

Specifications

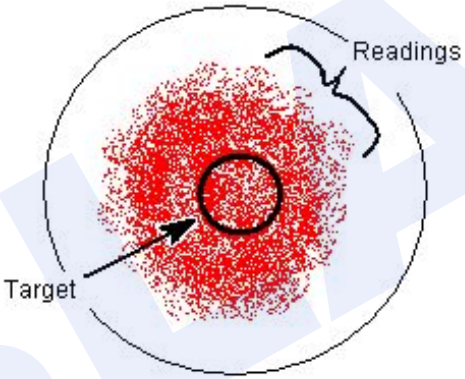


Sensors

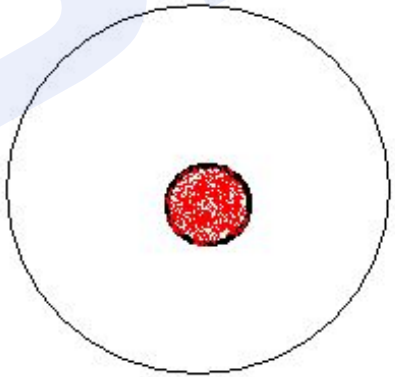
Specifications



Precision without accuracy



Accuracy without precision

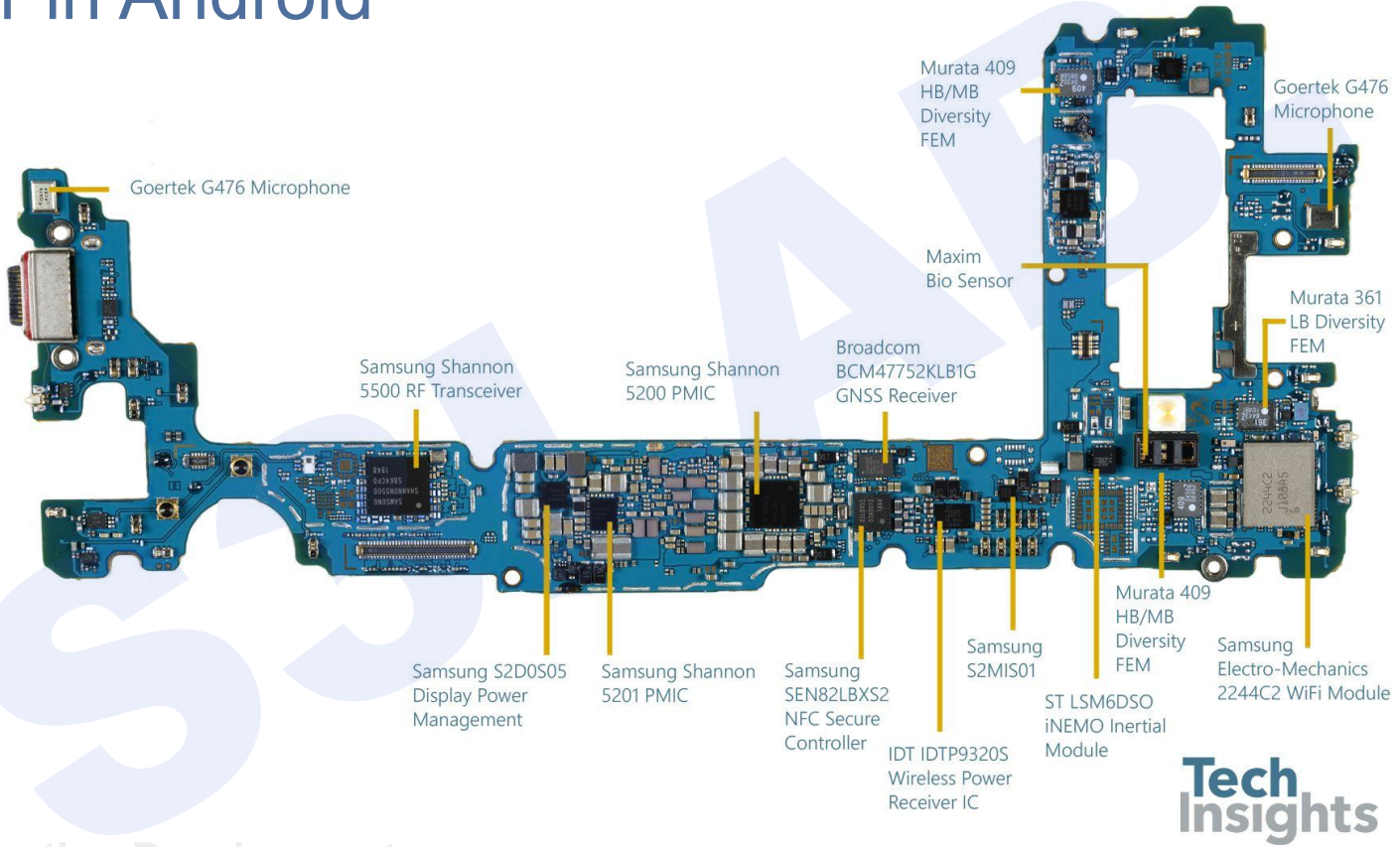


Precision and accuracy

Sensor in Android

- In Android devices, there are various built-in sensors that can be used to measure the orientation, motions, and various other kinds of environmental conditions:
  - **Hardware Sensors**: Hardware sensors are physical components that are present in Android devices. They can directly measure various properties like field strength, acceleration, etc according to the types of the sensors and after measuring the environment properties they can send the data to Software Sensors.
  - **Software Sensors:** Software sensors also know as virtual sensors are those sensors that take the help of one or more Hardware sensors and based on the data collected by various Hardware sensors, they can derive some result.

Sensor in Android



Categories

- **Motion Sensors**:

  Measure acceleration forces and rotational forces along three axes -> **accelerometers**, **gravity** sensors, **gyroscopes**, and rotational vector sensors.

- **Environmental Sensors:**

  Measure various environmental parameters -> **ambient** air temperature and **pressure**, illumination, and humidity -> **barometers**, **photometers**, and **thermometers**.

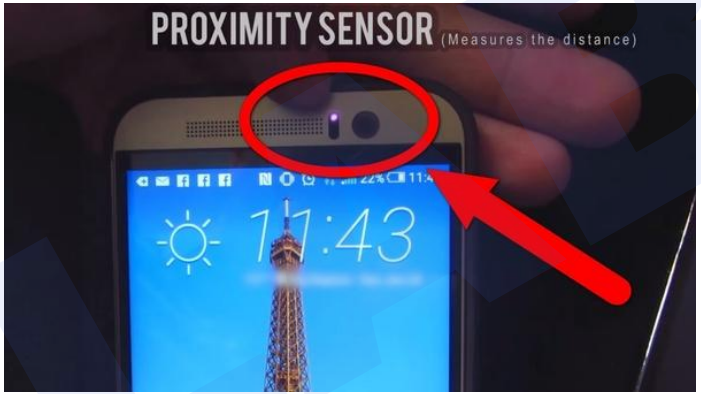
- **Orientation and Position Sensors:** Measure the physical position of a device: orientation sensors and magnetometers.

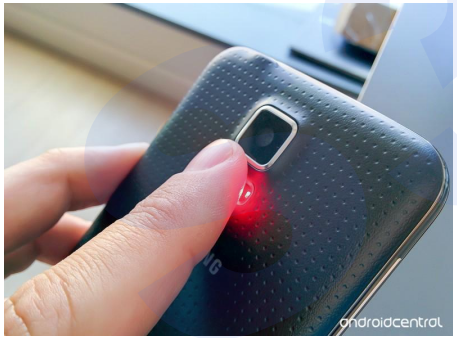
Categories

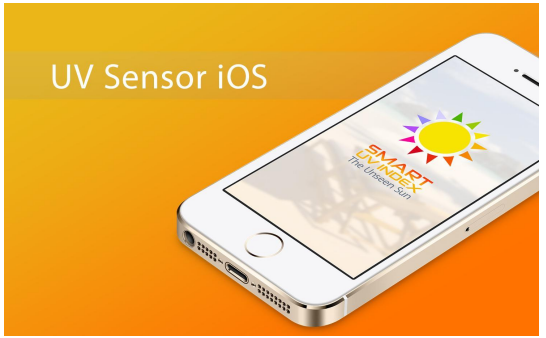

Proximity sensor
Ambient Light Sensor
CMOS Imager sensors
GPS sensor
Accelerometer
Humidity sensor
Microphone
Temperature sensor
Pressure sensor
Fingerprint sensor
Magnetometer
Gyroscope
Touch sensor

Categories

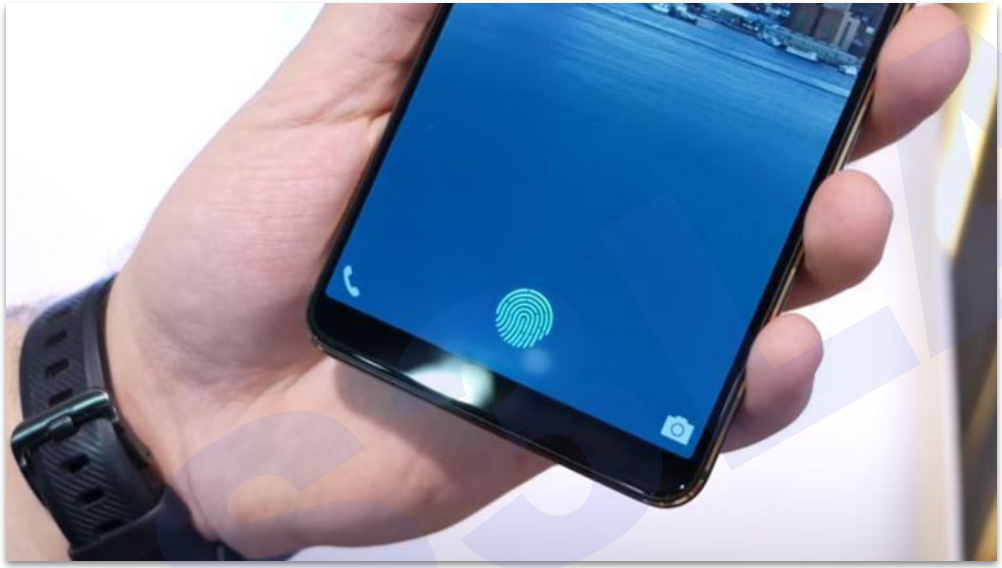

Sensors in
Mobile Device
Ambient
Light
techie
prashant
[Hindi]
Settings
Display & Brightness
BRIGHTNESS
Auto-Brightness
Night Shift
10:00 PM to 7:00 AM
Auto-Lock
5 Minutes


PROXIMITY SENSOR (Measures the distance)
11:43


androidcentral


UV Sensor iOS
SMART
UV INDEX

Categories



Android Sensor Framework

- It lists all the available sensors on the device
- It determines the capabilities of each sensor, such as its maximum range, manufacturer, power requirements, and resolution.
- It can acquire raw sensor data and define the minimum rate at which you acquire sensor data.
- Register and unregister sensor event listeners that monitor sensor changes.

Android Sensor Framework

Android.hardware

- **SensorManager**: This is used to get access to various sensors present in the device to use it according to need.
- **Sensor**: This class is used to create an instance of a specific sensor.
- **SensorEvent**: This class is used to find the details of the sensor events.
- **SensorEventListener**: This interface can be used to trigger or perform some action when there is a change in the sensor values.

Android Sensor Framework

List of sensors

- **TYPE_ACCELEROMETER**

  Type: Hardware

  Computes the **acceleration** in m/s2 applied on all three axes (x, y and z), including the force of gravity.

- **TYPE_AMBIENT_TEMPERATURE**

  Type: Hardware

  Monitors the **temperature** of the surroundings in degrees Celsius.

- **TYPE_GRAVITY**

  Type: Software or Hardware

  Computes the **gravitational** force in m/s2 applied on all three axes (x, y and z).

Android Sensor Framework

List of sensors

- **TYPE_GYROSCOPE**

  Type: Hardware

  Computes the rate of **rotation** in rad/s around each of the three axes (x, y and z).

- **TYPE_LIGHT**

  Type: Hardware

  Evaluates the **light** around a surrounding in lx units.

- **TYPE_LINEAR_ACCELERATION**

  Type: Software or Hardware

  Computes the acceleration force in m/s2 applied on all three axes (x, y and z), excluding the force of gravity.

Android Sensor Framework

List of sensors

- **TYPE_MAGNETIC_FIELD**

  Type: Hardware

  Computes the geomagnetic field for all three axes in tesla (μT).

- **TYPE_ORIENTATION**

  Type: Software

  Computes the degree of rotation around all three axes.

- **TYPE_PRESSURE**

  Type: Hardware

  Computes the air pressure in hPa or mbar.

Android Sensor Framework

List of sensors

- **TYPE_PROXIMITY**

  Type: Hardware

  Computes the proximity of the device's screen to an object in centimeters.

- **TYPE_RELATIVE_HUMIDITY**

  Type: Hardware

  Computes the humidity of the surrounding air as a percentage (%).

- **TYPE_ROTATION_VECTOR**

  Type: Software or Hardware

  Computes the orientation of a device by the device's rotation vector.

Android Sensor Framework

List of sensors

- **TYPE_TEMPERATURE**

  Type: Hardware

  Monitors the temperature of the surroundings in degrees Celsius. In API 14, the **TYPE_AMBIENT_TEMPERATURE** sensor replaced this sensor.

Android Sensor Framework

List of sensors

Sensor	Android 4.0 (API Level 14)	Android 2.3 (API Level 9)	Android 2.2 (API Level 8)	Android 1.5 (API Level 3)
TYPE_ACCELEROMETER	Yes	Yes	Yes	Yes
TYPE_AMBIENT_TEMPERATURE	Yes	n/a	n/a	n/a
TYPE_GRAVITY	Yes	Yes	n/a	n/a
TYPE_GYROSCOPE	Yes	Yes	n/a[1]	n/a[1]
TYPE_LIGHT	Yes	Yes	Yes	Yes
TYPE_LINEAR_ACCELERATION	Yes	Yes	n/a	n/a
TYPE_MAGNETIC_FIELD	Yes	Yes	Yes	Yes
TYPE_ORIENTATION	Yes[2]	Yes[2]	Yes[2]	Yes
TYPE_PRESSURE	Yes	Yes	n/a[1]	n/a[1]
TYPE_PROXIMITY	Yes	Yes	Yes	Yes
TYPE_RELATIVE_HUMIDITY	Yes	n/a	n/a	n/a
TYPE_ROTATION_VECTOR	Yes	Yes	n/a	n/a
TYPE_TEMPERATURE	Yes[2]	Yes	Yes	Yes

Android Sensor Framework

List of sensors -> ex. Get list all of sensor

- Make new app.
- In onCreate event of activity
  - SensorManager sensorManager = (SensorManager)
    getSystemService(Context.SENSOR_SERVICE);
    List<Sensor> list = sensorManager.getSensorList(Sensor.TYPE_ALL);
    for (Sensor s : list) {
        Log.e("SENSORS","name: "+s.getName());
    }

Android Sensor Framework

List of sensors -> ex. Get list of specific sensor

```
private SensorManager sensorManager;
...
sensorManager = (SensorManager) getSystemService(Context.SENSOR_SERVICE);
if (sensorManager.getDefaultSensor(Sensor.TYPE_MAGNETIC_FIELD) != null){
    // Success! There's a magnetometer.
} else {
    // Failure! No magnetometer.
}
```

Android Sensor Framework

List of sensors -> ex. Monitor raw data

```java
public class SensorActivity extends Activity implements SensorEventListener {
    private SensorManager sensorManager;
    private Sensor mLight;

    @Override
    public final void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        sensorManager = (SensorManager) getSystemService(Context.SENSOR_SERVICE);
        mLight = sensorManager.getDefaultSensor(Sensor.TYPE_LIGHT);
    }

    @Override
    public final void onAccuracyChanged(Sensor sensor, int accuracy) {
        // Do something here if sensor accuracy changes.
    }

    @Override
    public final void onSensorChanged(SensorEvent event) {
        // The light sensor returns a single value.
        // Many sensors return 3 values, one for each axis.
        float lux = event.values[0];
        // Do something with this sensor value.
    }

    @Override
    protected void onResume() {
        super.onResume();
        sensorManager.registerListener(this, mLight, SensorManager.SENSOR_DELAY_NORMAL);
    }

    @Override
    protected void onPause() {
        super.onPause();
        sensorManager.unregisterListener(this);
    }
}
```

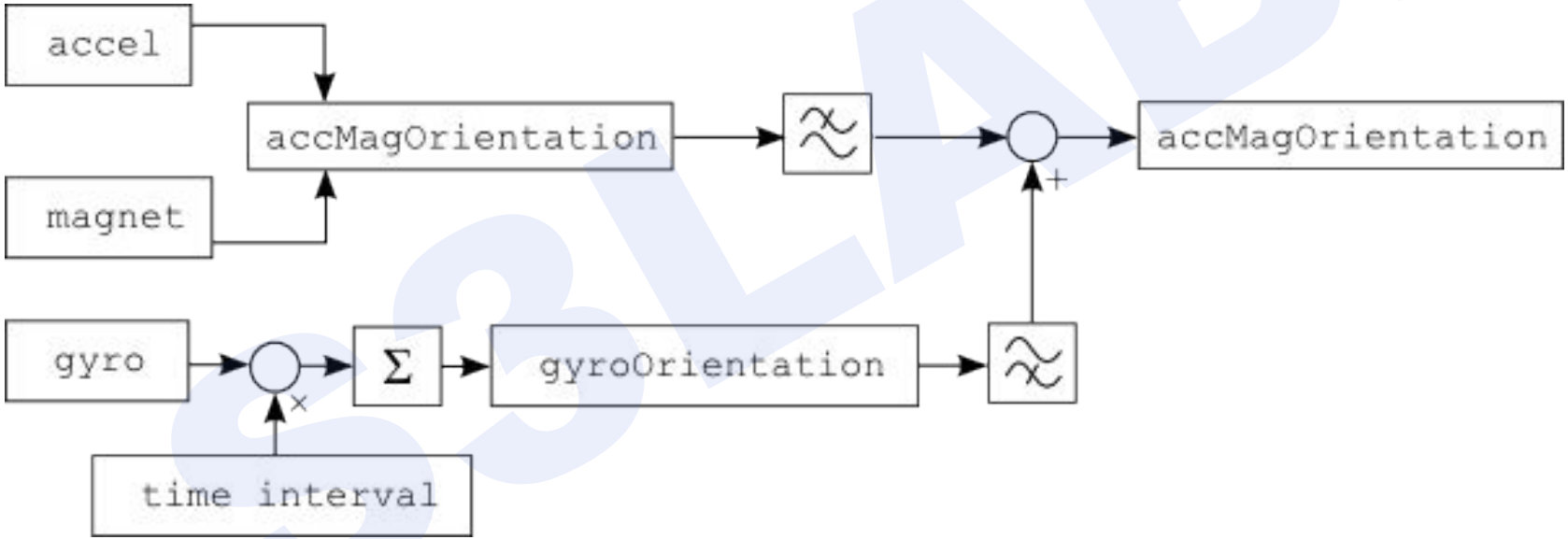
Android Sensor Framework

Sensor Fusion - Combining sensor data

- This method of using data from two or more sensors to get a more accurate result is known as Sensor Fusion.
  - Typically, you can develop a compass just with a **magnetometer**
  - However, you can combine a **magnetometer** with an **accelerometer**

Android Sensor Framework

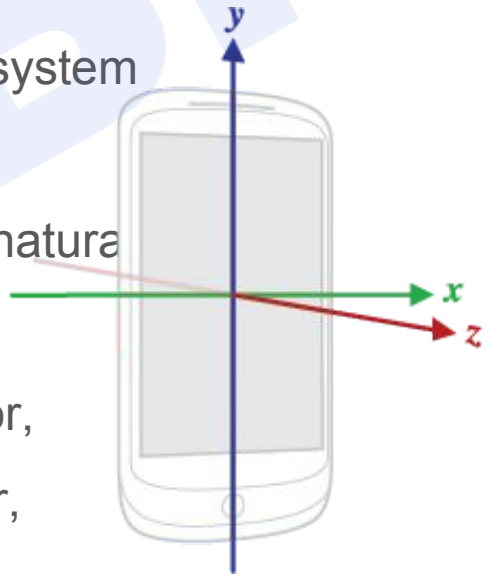
Sensor Fusion - Combining sensor data



Android Sensor Framework

Sensor Coordinate System

- Sensor framework uses a standard 3-axis coordinate system to express data values.
- the sensor coordinate system is always based on the natura orientation of a device (portrait or landscape)
- This coordinate system is used by: Acceleration sensor, Gravity sensor, Gyroscope, Linear acceleration sensor, Geomagnetic field sensor



Android Sensor Framework

Using Google Play Filter

- Only the device which has a specific Sensor can see the application on google play.
- In manifest file:
  - <uses-feature android:name="android.hardware.sensor.accelerometer" android:required="true" />
  - users will see your application on Google Play only if their device has an accelerometer.

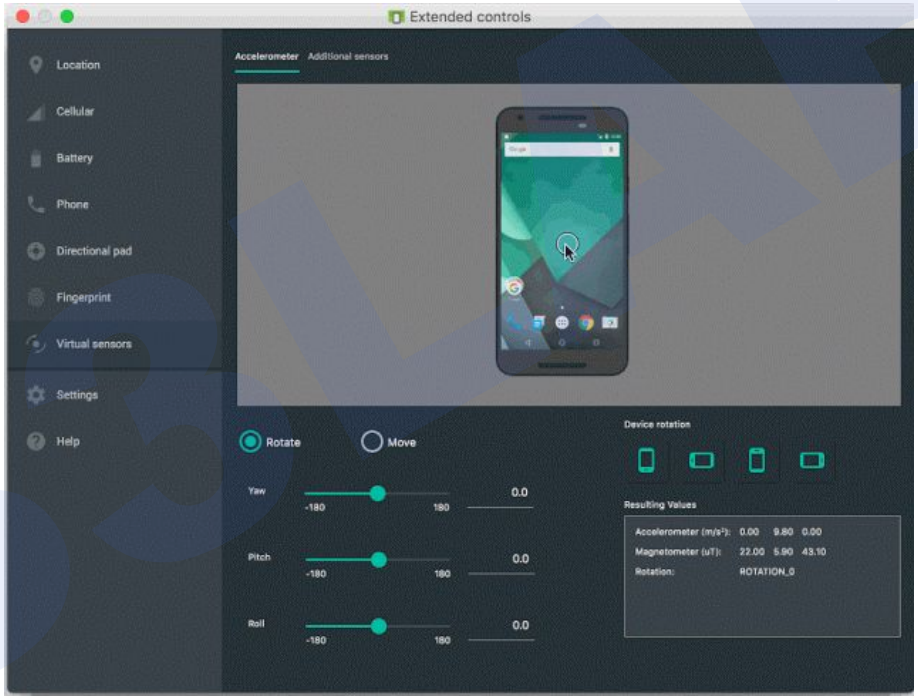
Android Sensor Framework

Virtual Sensor in Emulator

- https://storage.googleapis.com/androiddevelopers/videos/studio-emulator-overview_2-2.mp4

Android Sensor Framework

Virtual Sensor in Emulator



Homeworks

- https://programmerworld.co/android/how-to-create-your-own-compass-android-app-using-magnetic-field-and-accelerometer-sensors-in-android-studio-complete-source-code/
- Run the examples in materials folder and carefully look to understand source code

Q & A

Thank you for listening

“Coming together is a beginning;
Keeping together is progress;
Working together is success.”
- HENRY FORD