Sensor Manger & Location Overview

Dr. Abraham Campbell
University College Dublin
Abey.campbell@ucd.ie

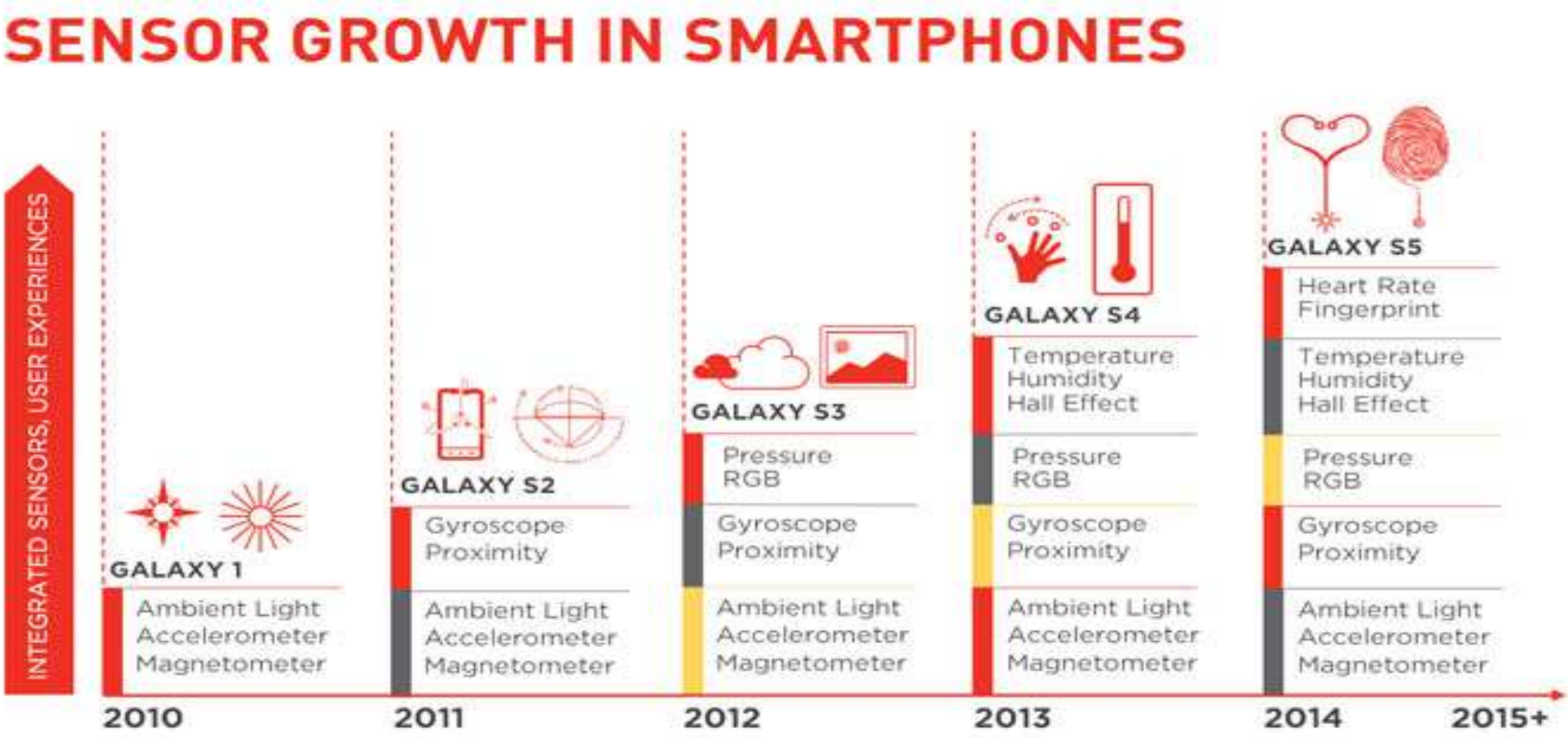
Outline

- Rise of Sensors within Mobile Phone Devices
- Sensor Manager Framework
- How can your location be sensed ?
- Location Framework

Rise of Sensors within Smart Phones

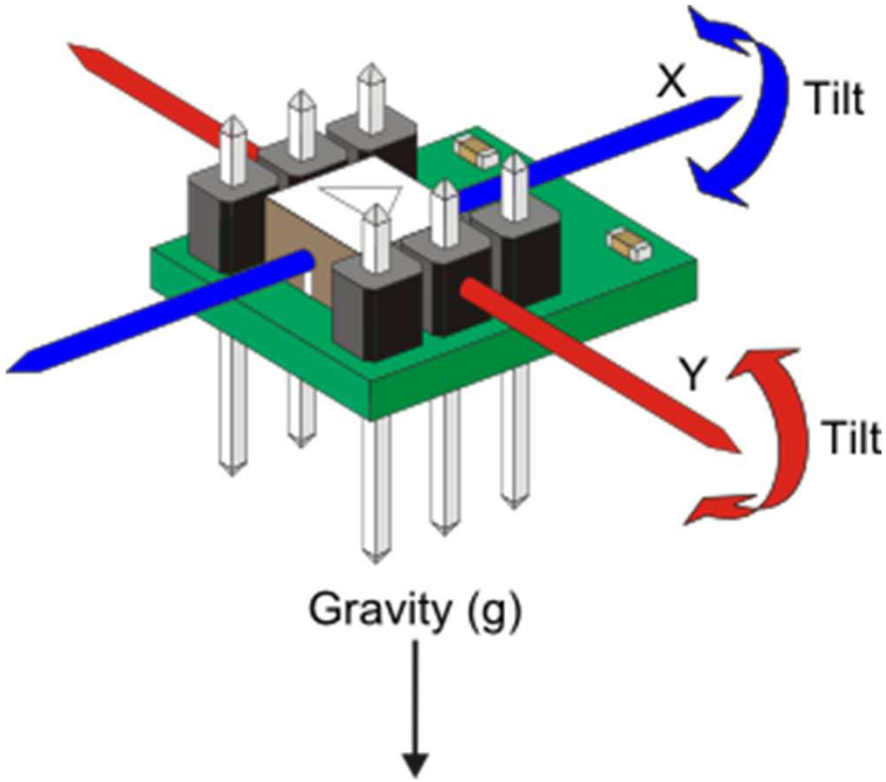
- Why is a Smart Phone different than programming for just a PC.
- **Context:** The phone is with you constantly and knows what's going on around it .
- It achieves this context with the use of sensors.

Case Study: Sensor growth Samsung S series



Accelerometer

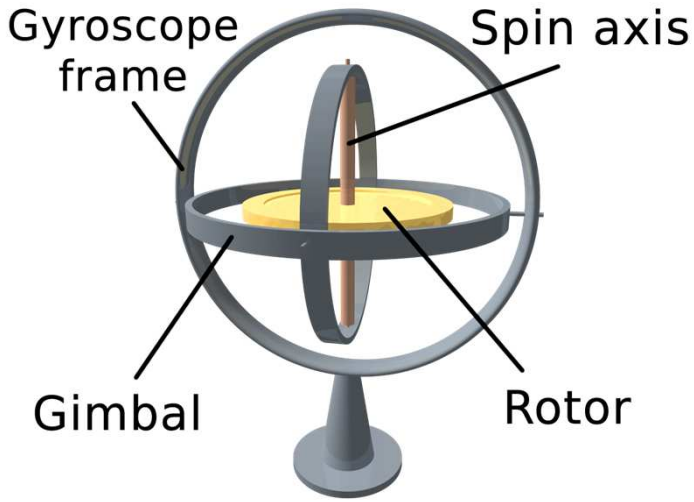
- Gives you the acceleration of the sensor in X,Y,Z axis
- When the phone is not moving , then the only force acting upon it is gravity thus giving you the basic orientation of the phone
- Multiple positions give the same reading so you do not get a complete orientation postion.



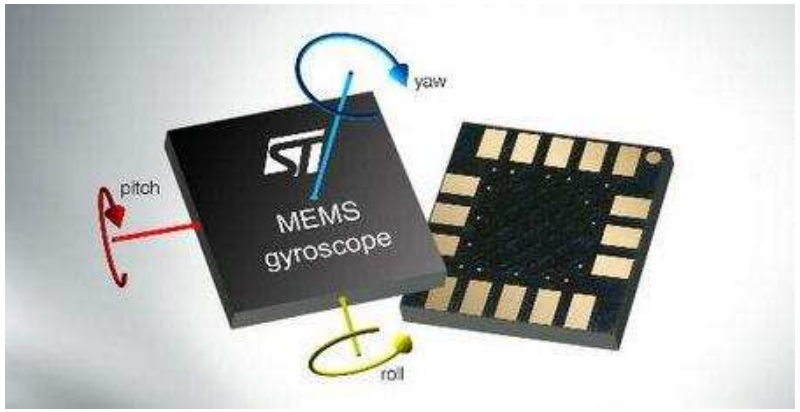
Gyroscope

- To get a complete orientation position of a phone would require a Gyroscope.
- It can tell you with extreme accuracy what angle it started and ended at but can't tell what orientation it was at the beginning.

Traditional Gyroscope



Just another Chip in your phone but with similar principles



Ambient Light Sensor & Proximity Detection

- Sensors within the phone checking the light and potential reducing or increasing the screen brightness .
- Proximity detection, when you move the phone to your ear, remember it’s a touch screen .
  - Why does your ear not press any buttons on your phone.
  - The phone switches off the touch screen when it detects that it is about to be covered.
  - Different phones implement these sensors differently but from the programmers perspective , Android abstracts out any details.

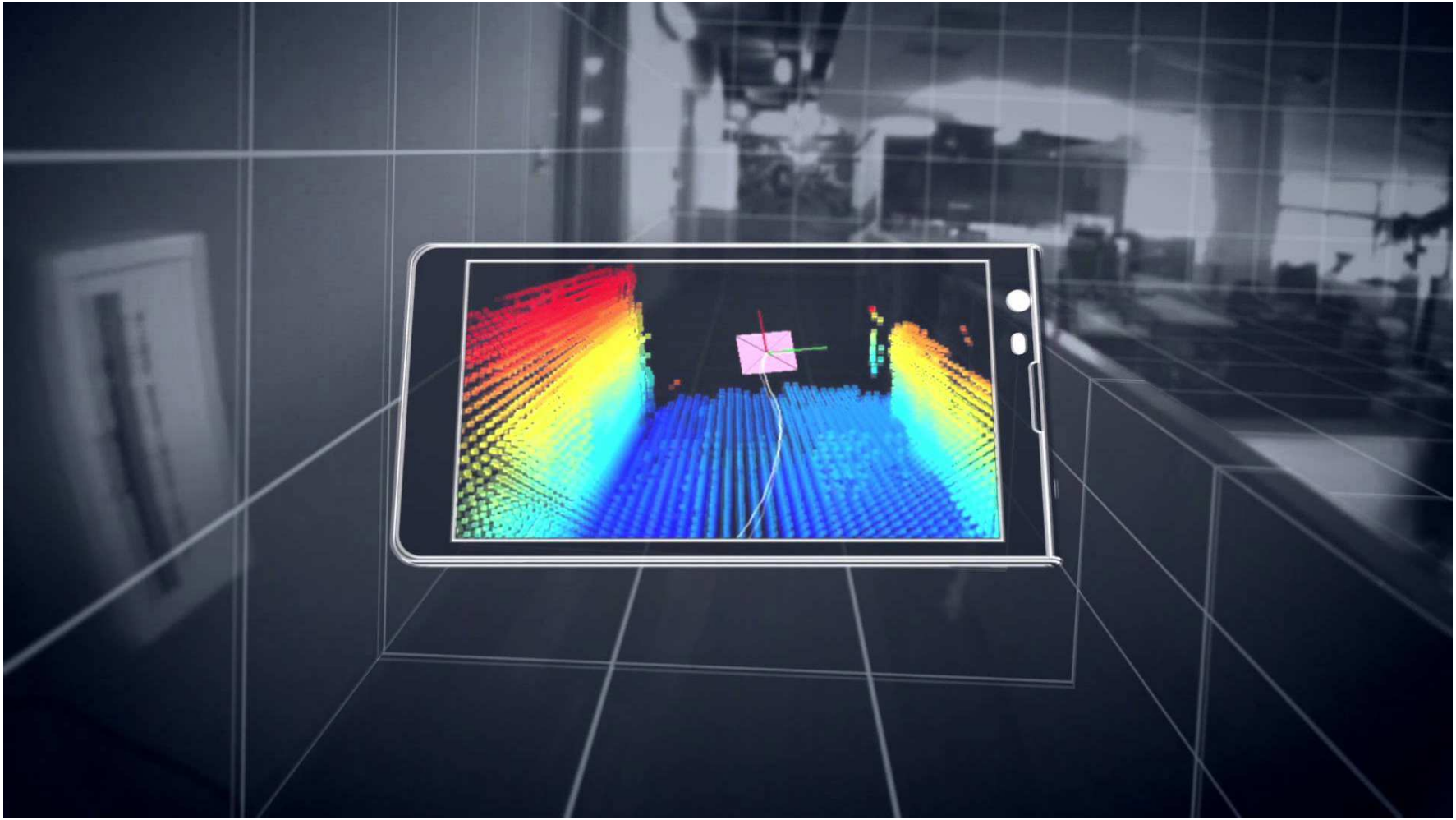
Pressure / Temperature / Humidity

- Some mobile phone devices will even include environmental sensors to detect the outside world.
- Each of these sensors is developed into a solid state chip that allows previously complex environment sensors to be mass procudedc.
- In most cases though the accuracy and more important calibration is not close to commercially available environmental sensors.
- Once calibrated though these sensors could provide a sensor web where all our phones could through crowd sourcing provide all our weather prediction needs.

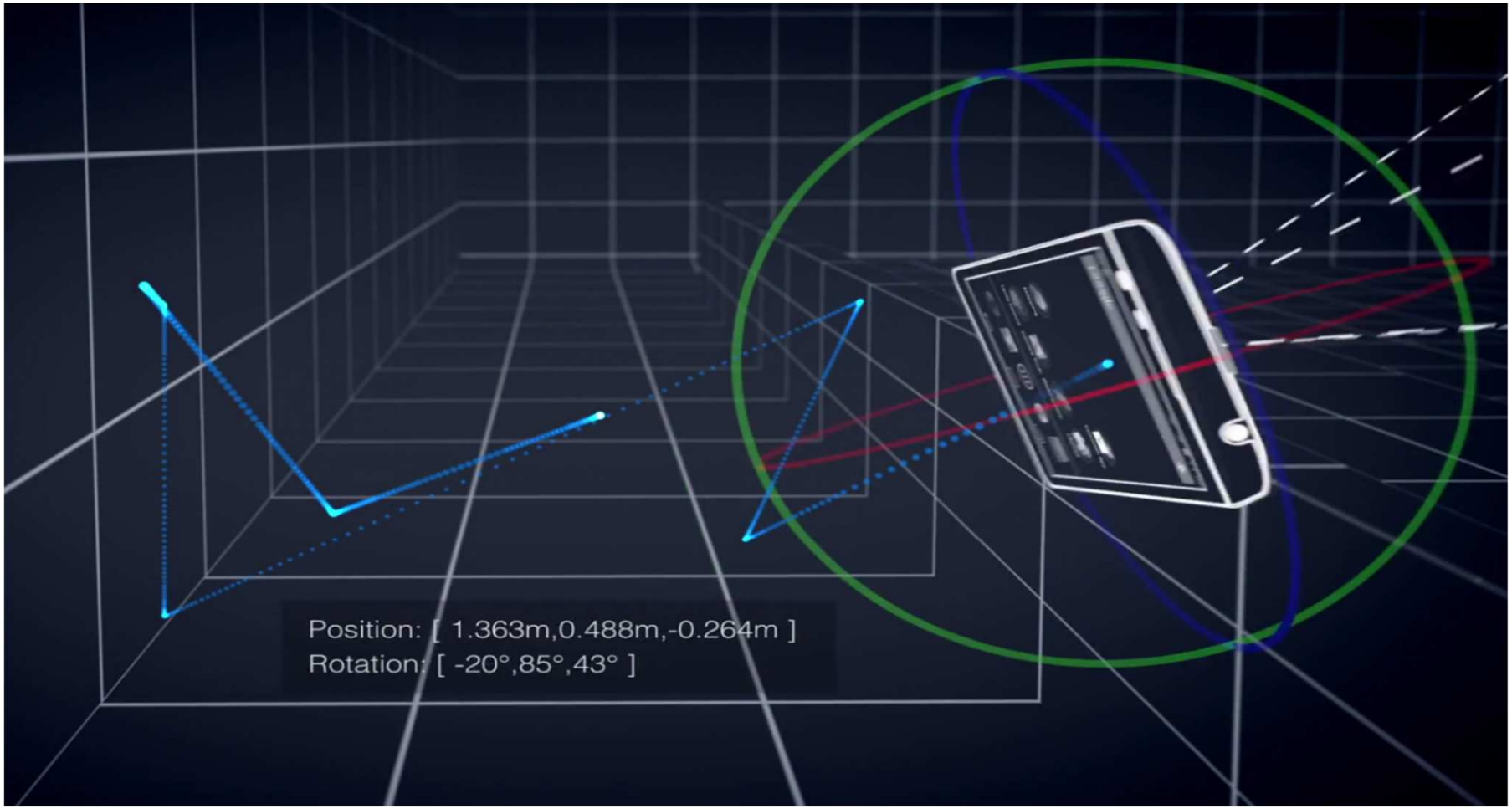
Magnetometer

- Many mobile devices provide a Magnetometer to act as a digital compass.
- This can be achieved in multiple ways such as
  - Halls effect based Magnetometer
  - Magnetoresistive devices
- Due to large effects of metal in buildings, these sensors can be very inaccurate inside.
- A side effect of this effect has been used to actual map buildings as they can have a unique magnetic finger print.

Future Sensors :
Project Tango: More Sensors to come



Mapping the world: Combing Sensors



Sensor Manger Framework

- Gives a fixed interface to the majority of sensor within an Android phone.
- Not all Android phones will have all of the potential sensors but you can assume sensors like an accelerometer for most standard phones.
- Remember the microphone and camera are consider other sensors too but fall outside of this framework.
- The phone discovers its location using the location framework which will discuss later on.

Sensors list (Basic)

Sensor	Type	Description	Common Uses
TYPE_ACCELEROMETER	Hardware	Measures the acceleration force in m/s^2 that is applied to a device on all three physical axes (x, y, and z), including the force of gravity.	Motion detection (shake, tilt, etc.).
TYPE_AMBIENT_TEMPERATURE	Hardware	Measures the ambient room temperature in degrees Celsius (°C). See note below.	Monitoring air temperatures.
TYPE_GRAVITY	Software or Hardware	Measures the force of gravity in m/s^2 that is applied to a device on all three physical axes (x, y, z).	Motion detection (shake, tilt, etc.).
TYPE_GYROSCOPE	Hardware	Measures a device's rate of rotation in rad/s around each of the three physical axes (x, y, and z).	Rotation detection (spin, turn, etc.).
TYPE_LIGHT	Hardware	Measures the ambient light level (illumination) in lx.	Controlling screen brightness.

Sensors list (Basic)

Sensor	Type	Description	Common Uses
TYPE_LINEAR_ACCELERATION	Software or Hardware	Measures the acceleration force in m/s^2 that is applied to a device on all three physical axes (x, y, and z), excluding the force of gravity.	Monitoring acceleration along a single axis.
TYPE_MAGNETIC_FIELD	Hardware	Measures the ambient geomagnetic field for all three physical axes (x, y, z) in µT.	Creating a compass.
TYPE_PRESSURE	Hardware	Measures the ambient air pressure in hPa or mbar.	Monitoring air pressure changes.
TYPE_PROXIMITY	Hardware	Measures the proximity of an object in cm relative to the view screen of a device. This sensor is typically used to determine whether a handset is being held up to a person's ear.	Phone position during a call.
TYPE_RELATIVE_HUMIDITY	Hardware	Measures the relative ambient humidity in percent (%).	Monitoring dewpoint, absolute, and relative humidity.
TYPE_ROTATION_VECTOR	Software or Hardware	Measures the orientation of a device by providing the three elements of the	Motion detection and rotation detection.

Sensor Framework

- SensorManger
  - This class gives you an instance of the sensor service
- Sensor
  - This class gives you an instance of a sensor so you can poll it for its abilities.
- SensorEvent
  - This class gives you information about a single sensor event and detailed information about when and how it was gathered.
- SensorEventListener
  - Interface to create the necessary call-backs to grab SensorEvents

Finding out what sensors a devices has

- Once you assign your SensorManger object , you can check its default sensors to see if it has the sensor you want.

```
private SensorManager mSensorManager;
...
mSensorManager = (SensorManager)
getSystemService(Context.SENSOR_SERVICE);

if (mSensorManager.getDefaultSensor(Sensor.TYPE_MAGNETIC_FIELD) != null){
  // Success! There's a magnetometer.
  }
else {
  // Failure! No magnetometer.
  }
```

Sensor Events

- To monitor sensor changes, two call back methods are used using the SensorEventListener interface
  - Sensor reports a new value
    - **onSensorChanged()**
  - Sensors accuracy changes
    - **onAccuracyChanged()**
- For this course we will concentrate on onSensorChanged but remember accuracy of a sensor can be just if not more important than the value given back by the device.

Sensor (Light) example

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

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

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

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

Sensor (Light) example (cont.)

```
 @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();
    mSensorManager.registerListener(this, mLight, SensorManager.SENSOR_DELAY_NORMAL);
  }
  @Override
  protected void onPause() {
    super.onPause();
    mSensorManager.unregisterListener(this);
  }
}
```

How can your location be sensed ?

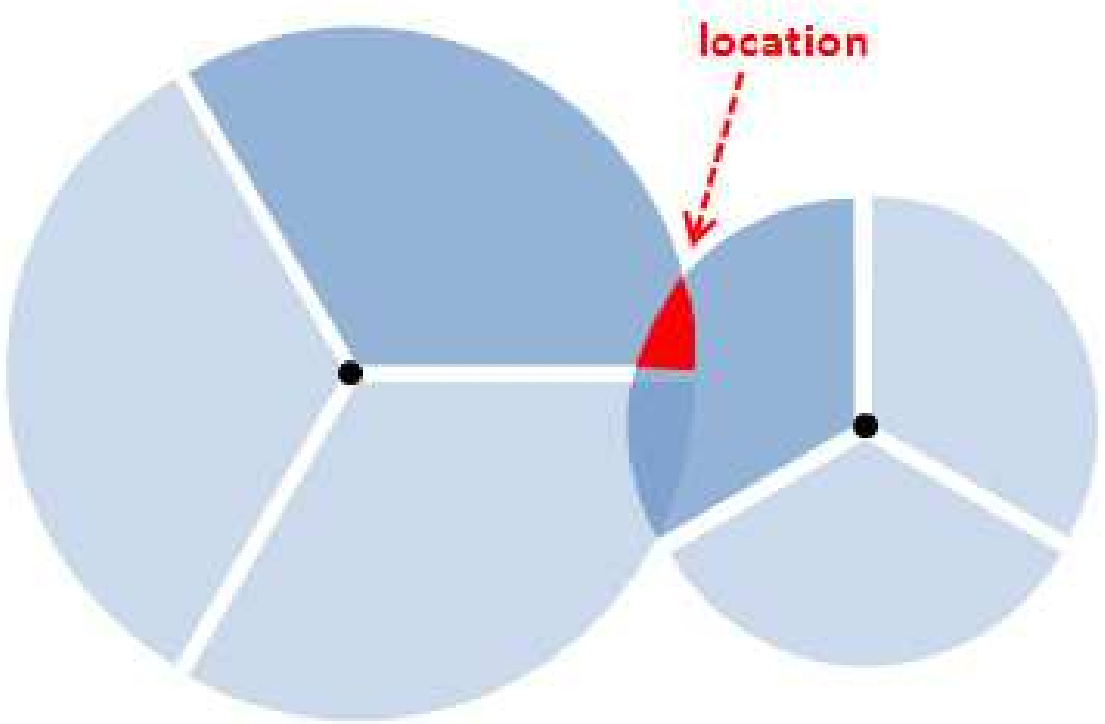
- GPS
- Network towers
- Wifi networks
- Tracking based on combinations of sensors

GPS

- Global Positioning System
- Space-based navigation system
- The receiver needs 4 satellites within line of sight(Radio , not optical)
- Run by US since 1970’s
- Alternatives such as EU’s Galileo (2019) and China’s COMPASS

NETWORK towers

- Your mobile carrier has listed the location of each of its cell phone towers
- Each tower can listen to your phone signal.
- Using the towers position, your position can be triangulated.
- Normally used in conjunction with GPS for Assisted GPS



Cell phone detected within a certain distance of two cell towers with directional antennae.

WIFI and other Tracking

- Similar to Cell phone towers , your network provider knows where you access your WiFi from an can guess your location.
- This is not very accurate but it can predicate where in an area you are.
- Other tracing is possible by combing all of this location information.

Location Manager

- Location Manager will use what ever the current location settings are on the phone.
- This will differ if the settings are for

`ACCESS_COARSE_LOCATION` or `ACCESS_FINE_LOCATION`

- By creating a listener object from the location Manager, your app can react to changes in the users movement.

- Alternatively you can simple just ask for the user last known location.
- Remember this is just a few examples , please examine the Android Developer doc's for more examples.

Location Manager

```
// Acquire a reference to the system Location Manager
LocationManager locationManager = (LocationManager)
this.getSystemService(Context.LOCATION_SERVICE);

// Define a listener that responds to location updates
LocationListener locationListener = new LocationListener() {
    public void onLocationChanged(Location location) {
      // Called when a new location is found by the network location provider.
      makeUseOfNewLocation(location);
    }

    public void onStatusChanged(String provider, int status, Bundle extras) {}

    public void onProviderEnabled(String provider) {}

    public void onProviderDisabled(String provider) {}
  };

// Register the listener with the Location Manager to receive location updates
locationManager.requestLocationUpdates(LocationManager.NETWORK_PROVIDER, 0, 0,
locationListener);
```

Requesting User Permissions / Normal flow

```
<manifest ... >
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
    ...
</manifest>
```