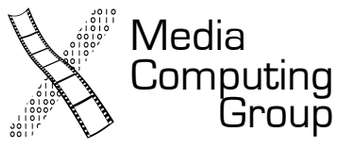


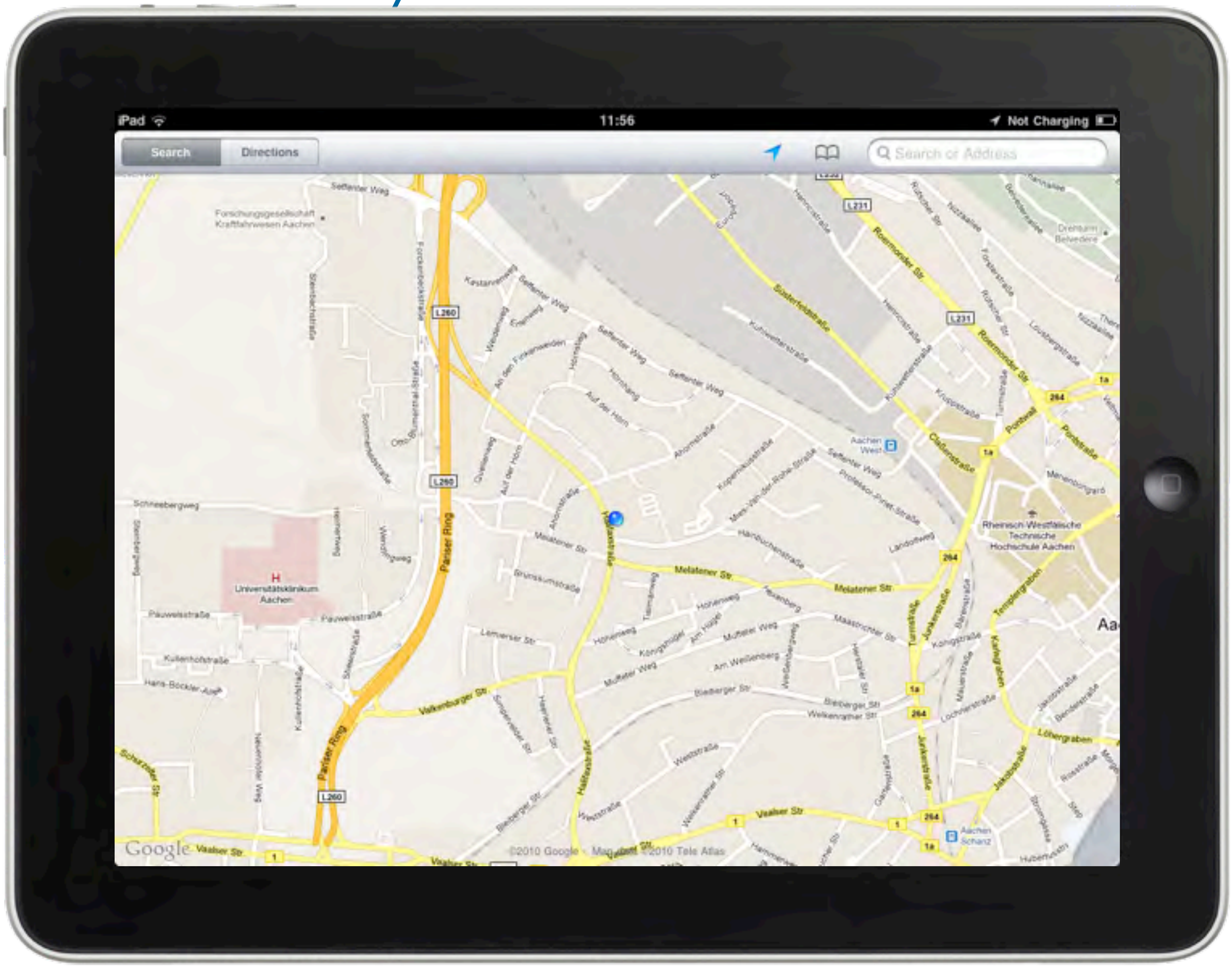
RWTH AACHEN UNIVERSITY

Mobile Application Development
L10: Location & Orientation

Jonathan Diehl (Informatik 10)
Hendrik Thüs (Informatik 9)

Location Tracking

Why Location Information?



Why Location Information?

Why Location Information?

Why Location Information?

Location Tracking Technologies

- Global Positioning System (GPS)
  - triangulation via satellites
  - requires special hardware
  - works best in open areas (bad in city canyons, impossible underground)
  - precision: ~10 meters (best condition)
  - also includes height (over sea level)

Location Tracking Technologies

- GSM Positioning
  - triangulation via cell-tower positions
  - requires phone and access to cell-tower location database (operator- or community-maintained)
  - works best in open spaces, obstacles (buildings) cause interference
  - precision: depends (~ 30 meters in typical situations)

Location Tracking Technologies

- WiFi Positioning
  - triangulation via wifi hotspots
  - requires wifi and access to wifi location database (usually community-maintained)
  - can be unreliable because wifi hotspots might move
  - works anywhere, where multiple wifi hotspots are visible
  - precision: depends (within room tracking possible)

Heading

- Direction in which the device is pointing
  - relative to magnetic or true north
  - only works when the device is laid down
  - magnetic fields or metal can interfere


iOS Location Tracking

```
   // load the location manager
manager = [[CLLocationManager alloc] init];
manager.delegate = self;

// start location and heading updates
[manager startUpdatingLocation];
[manager startUpdatingHeading];


// delegate method for a changed location
- (void)locationManager:(CLLocationManager *)manager
     didUpdateToLocation:(CLLocation *)newLocation
            fromLocation:(CLLocation *)oldLocation;

// delegate method for a changed heading
- (void)locationManager:(CLLocationManager *)manager
        didUpdateHeading:(CLHeading *)newHeading
```

Android Location Tracking

1. Set permission:

```
android.permission.ACCESS_FINE_LOCATION
```

2. Start updating the GPS/GSM/WiFi-position
3. Get list of location providers
4. Try to get a position from every provider

Geocoding & Maps

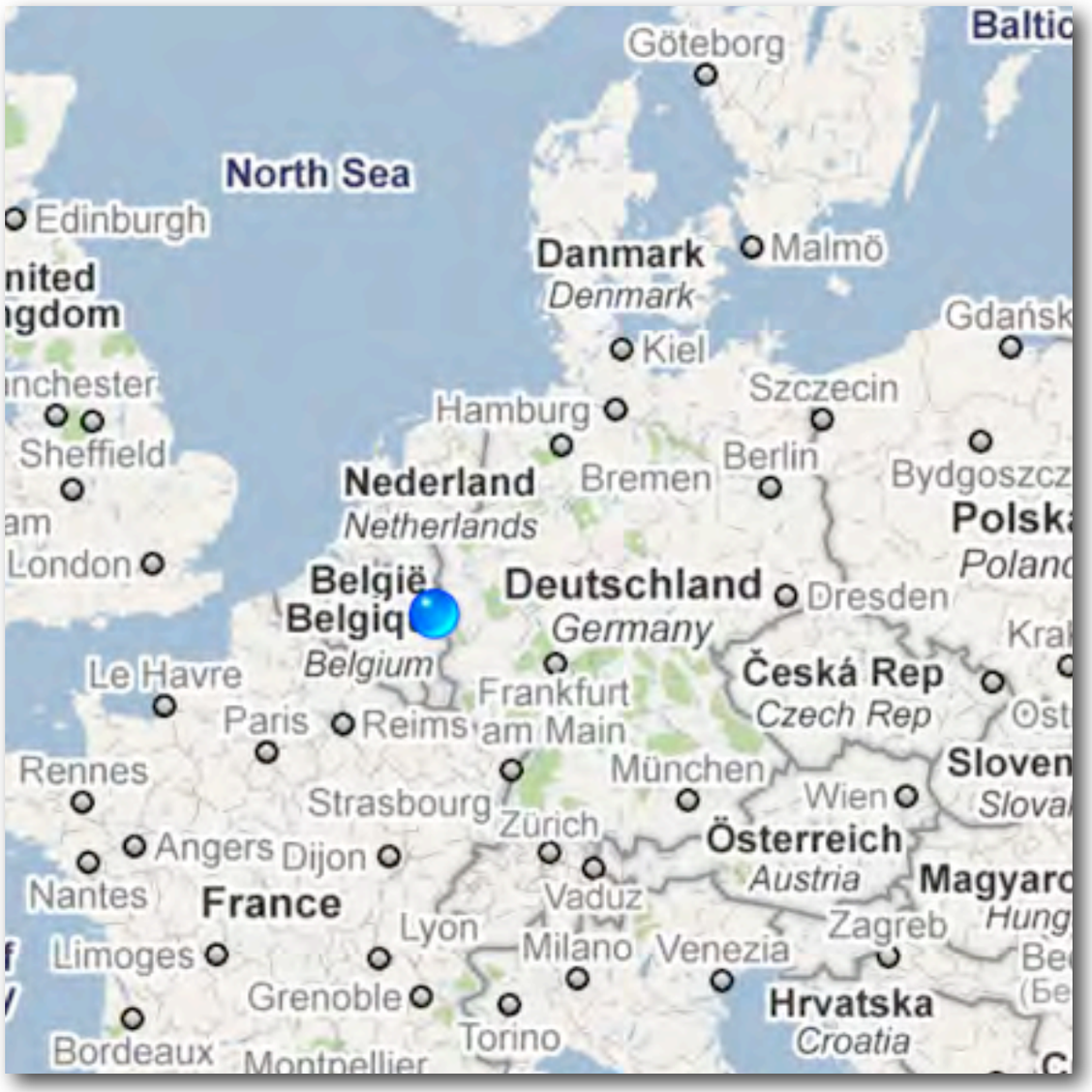
Geocoding

- Get the geographical location from a street address
- Google API:
  - http://code.google.com/apis/maps/documentation/geocoding

Reverse Geocoding

- Extract location information from a geographical location
  - Street address
  - Landmark
  - ...
- Included in the Google API (see previous slide)
  - Built into iOS: MKReverseGeocoder

Maps


Göteborg
North Sea
Edinburgh
Danmark
Malmö
Denmark
Kiel
Gdańsk
Hamburg
Szczecin
Berlin
Bremen
Nederland
Netherlands
London
Deutschland
Dresden
Germany
Belgium
Česká Rep
Czech Rep
Le Havre
Frankfurt
am Main
Paris
Reims
München
Rennes
Strasbourg
Wien
Zürich
Österreich
Austria
Angers
Dijon
Nantes
France
Vaduz
Lyon
Zagreb
Milano
Venezia
Limoges
Grenoble
Hrvatska
Croatia
Torino
Bordeaux

Maps on iOS: MapKit

- MKMapView
  - interactive (scrollable, zoomable) map view
  - can display user location
- Annotations
  - Pins: MKAnnotation, MKAnnotationView
  - Custom shapes: MKCircle, MKPolyline, ...
  - Overlays: MKOverlay

MapActivity on Android

- Create a new project using the Google APIs
- Add maps library to the application-tag of Manifest.xml:

```
<uses-library android:name="com.google.android.maps" />
```

- Allow Internet-access

MapView on Android

```
// Embed a MapView
<com.google.android.maps.MapView
    android:id="@+id/mapview"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:clickable="true"
    android:apiKey=„Google Maps API Key"
/>

// Display the Map
MapView mapView = (MapView) findViewById(R.id.mapview);
mapView.setBuiltInZoomControls(true);
```

See *http://code.google.com/android/add-ons/google-apis/mapkey.html*

Device Orientation

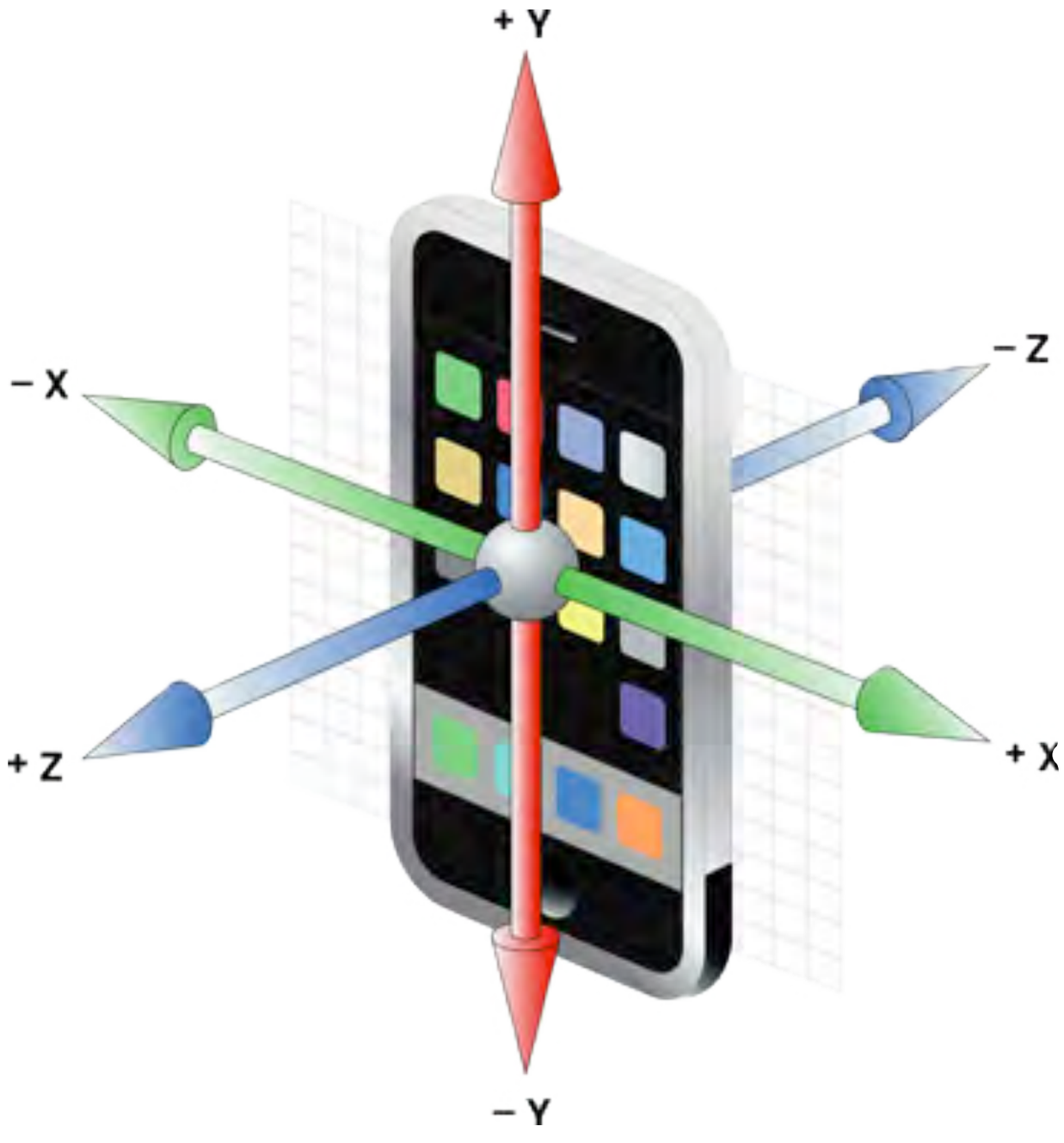
Accelerometer vs. Gyroscope

- Accelerometer
  - Measures proper acceleration
  - relative to free fall
  - 1.0 = 1G (earth's acceleration)
- Gyroscope
  - Measures rotation

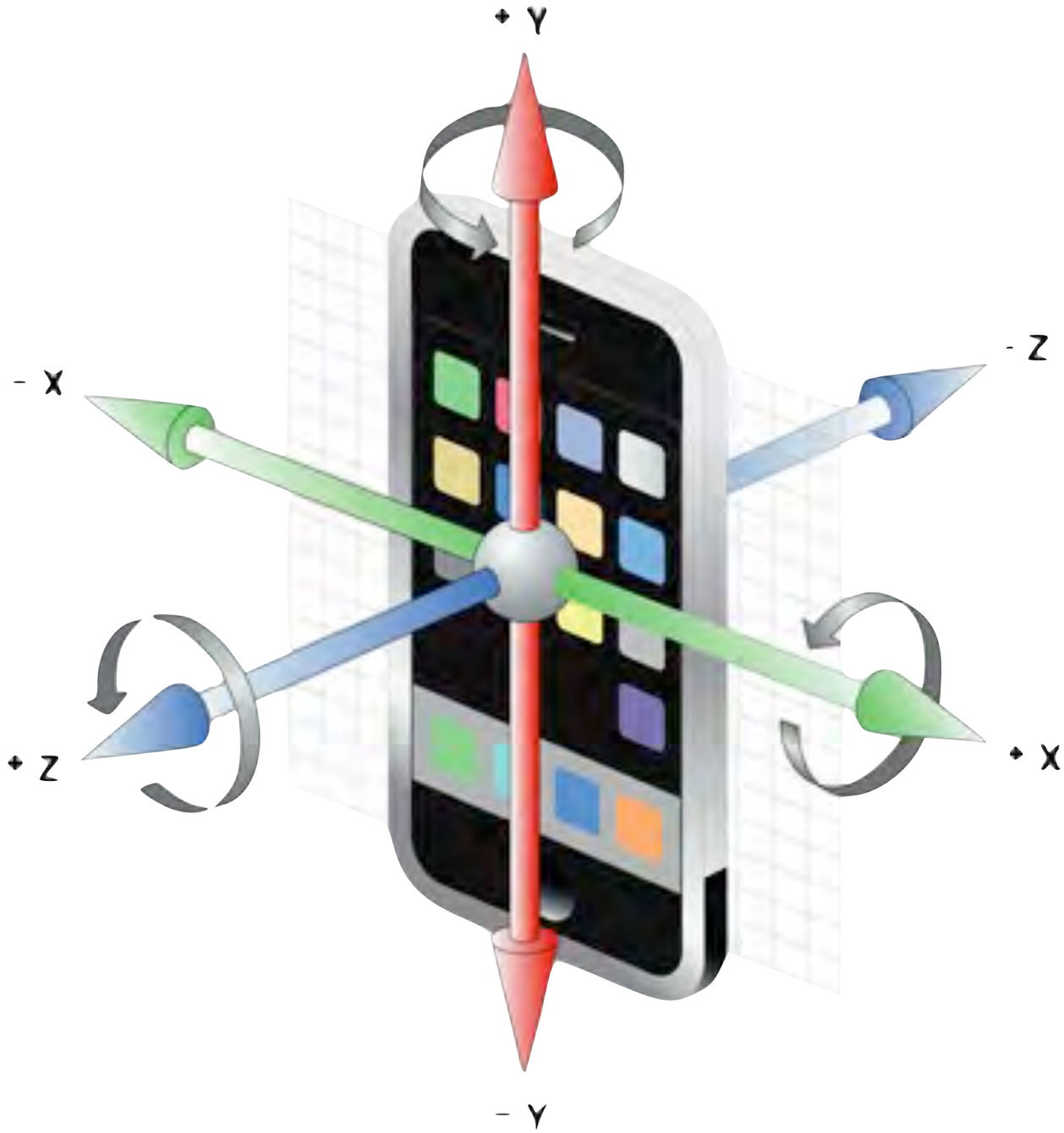
Update Frequency

10–20	Orientation detection
30–60	Real-time input (e.g., games)
70–100	high-frequency motion (e.g., hitting or shaking the device quickly)

Acceleration



Gyroscope



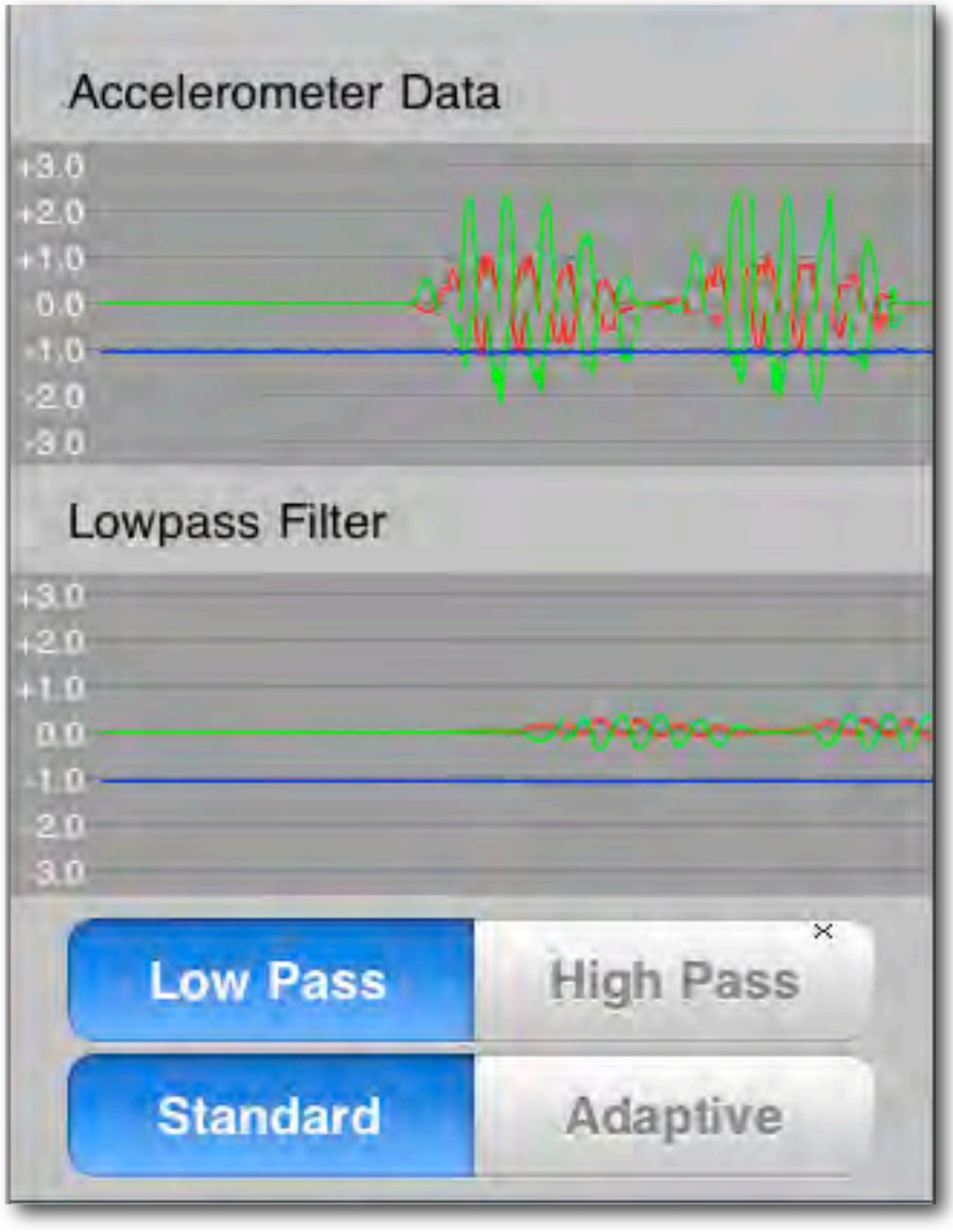
Filtering Data

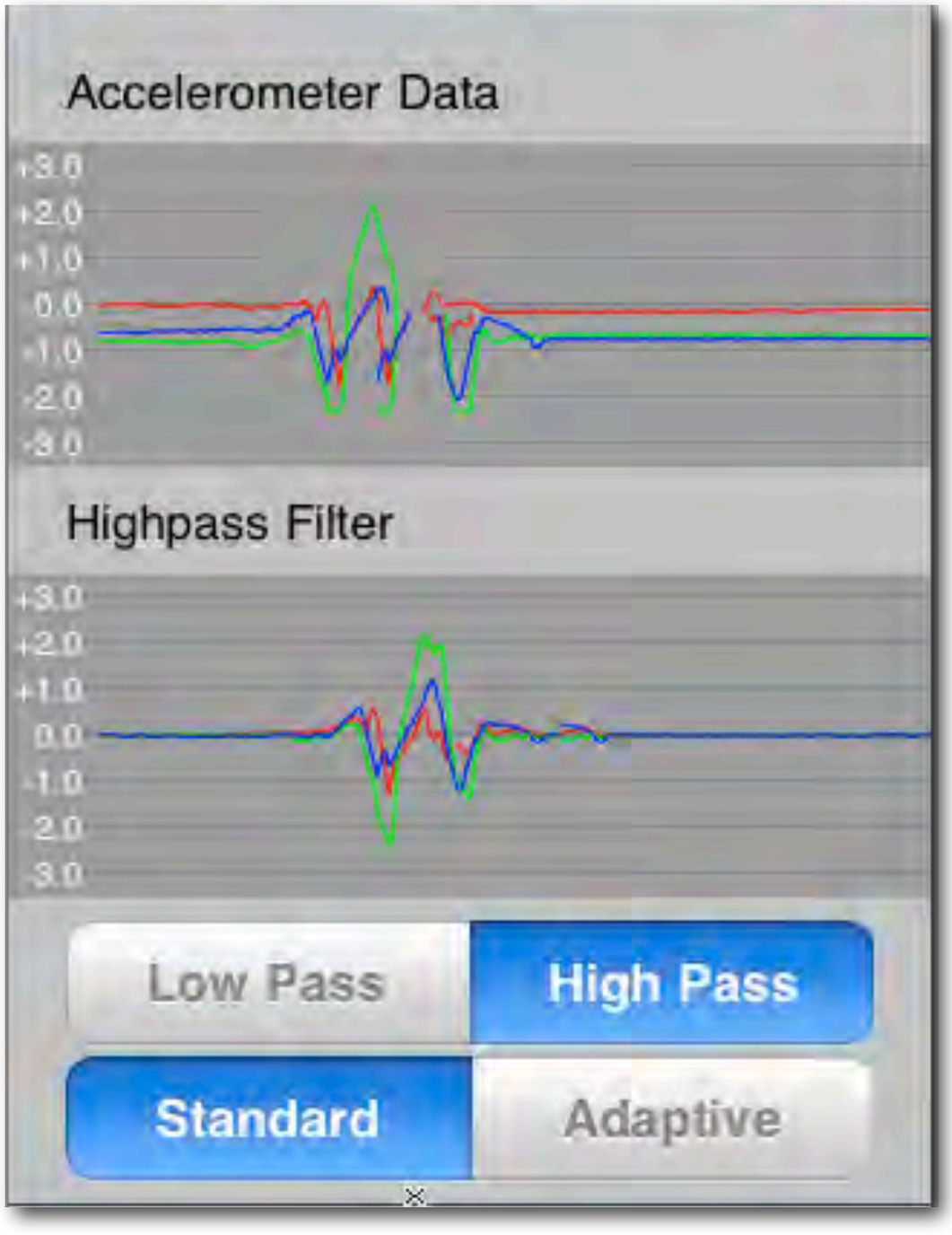
- Low-pass filter
  - pass low-frequency, cut off high-frequency signals
  - detect orientation changes
  - reduces jittering
- High-pass filter
  - pass high-frequency, cut off low-frequency signals
  - detect jittering
  - returns relative value

Low-Pass / High-Pass Filter

```
// low-pass filter
CGFloat lowpassFilter(CGFloat value, CGFloat filterFactor) {
   static CGFloat lowpassValue;
   lowpassValue = alpha * value + (1.0-alpha) * lowpassValue;
   return lowpassValue;
}

// high-pass filter
CGFloat highpassFilter(CGFloat value, CGFloat filterFactor) {
   static CGFloat prevValue, highpassValue;
   highpassValue = alpha * (highpassValue + value - prevValue);
   prevValue = value;
   return highpassValue;
}
```


Accelerometer Data
+3.0
+2.0
+1.0
0.0
-1.0
-2.0
-3.0
Lowpass Filter
+3.0
+2.0
+1.0
0.0
-1.0
2.0
3.0
Low Pass
High Pass
Standard
Adaptive


Accelerometer Data
+3.0
+2.0
+1.0
0.0
-1.0
-2.0
-3.0
Highpass Filter
+3.0
+2.0
+1.0
0.0
-1.0
2.0
3.0
Low Pass
High Pass
Standard
Adaptive

Orientation on iOS: Core Motion

- Core Motion Manager
  - start accelerometer / gyroscope updates
  - must implement own timer!
    - represented as CMAcceleration, CMRotation
- Bonus
  - *attitude*: orientation relative to a given reference
  - *gravity*: acceleration caused by gravity
  - *user acceleration*: acceleration caused by the user

Orientation on Android: Sensors

```
// The accuracy of this event
public int accuracy

// The sensor that generated this event
public Sensor sensor

// The time in nanosecond at which the event happened
public long timestamp

/* The length and contents of the values array depends on
which sensor type is being monitored (see also SensorEvent
for a definition of the coordinate system used) */
public final float[] values
```

More Sensors...

- Accelerometer
- Orientation
- Gyroscope
- Magnetic Field
- Light
- Proximity
- Gravity
- Linear Acceleration
- Rotation Vector