Beginning iOS 10 Application Development

Locations & Maps

Yanping Zhao

Nov. 2016

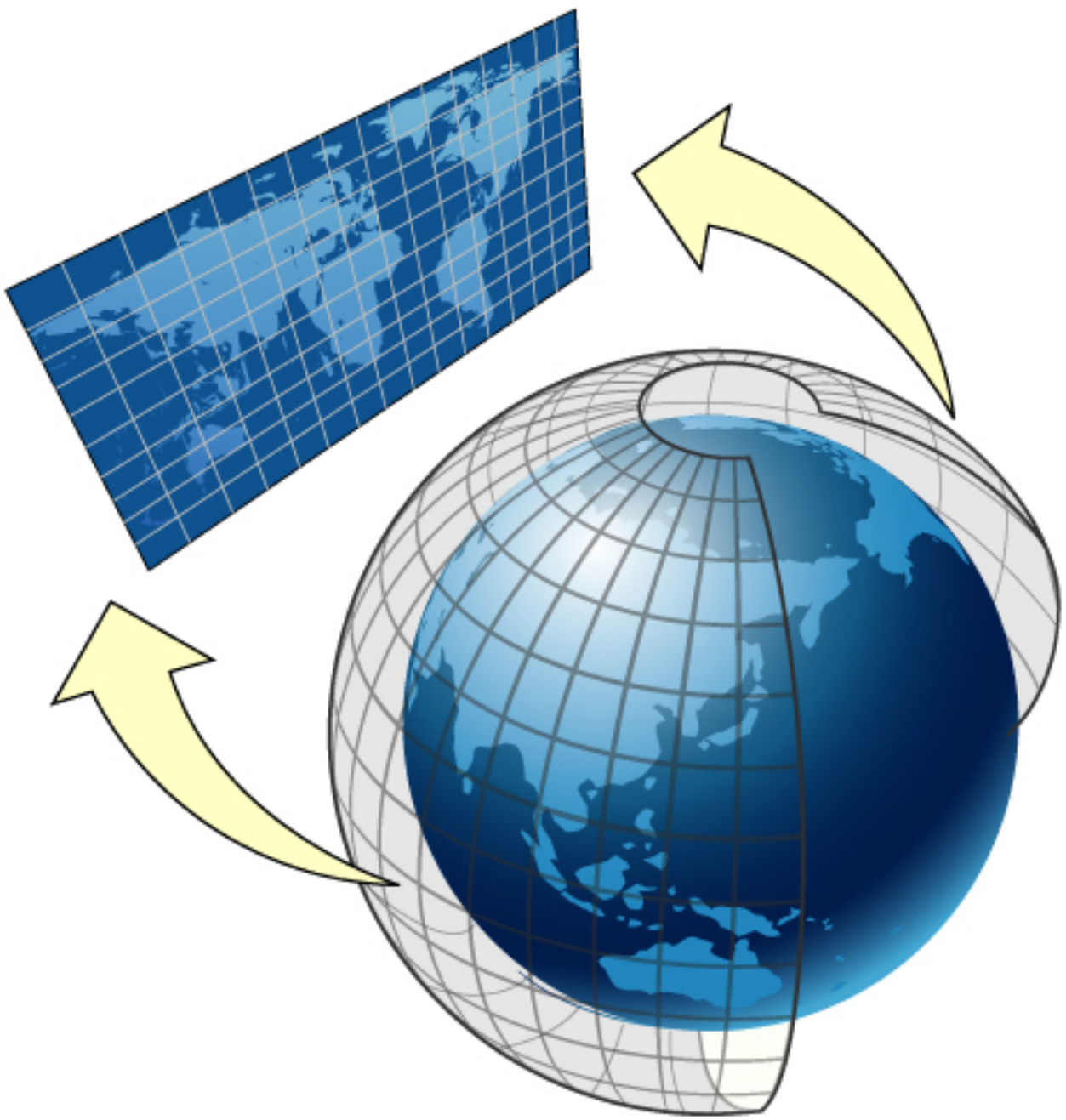
Maps

Embed a Full-Functional Map

- Include the Map Kit framework
- Map types: street, satellite, hybrid
- Zoom/pan the map
- Annotate the map
- Overlay the map
- Search the map
- Calculate directions

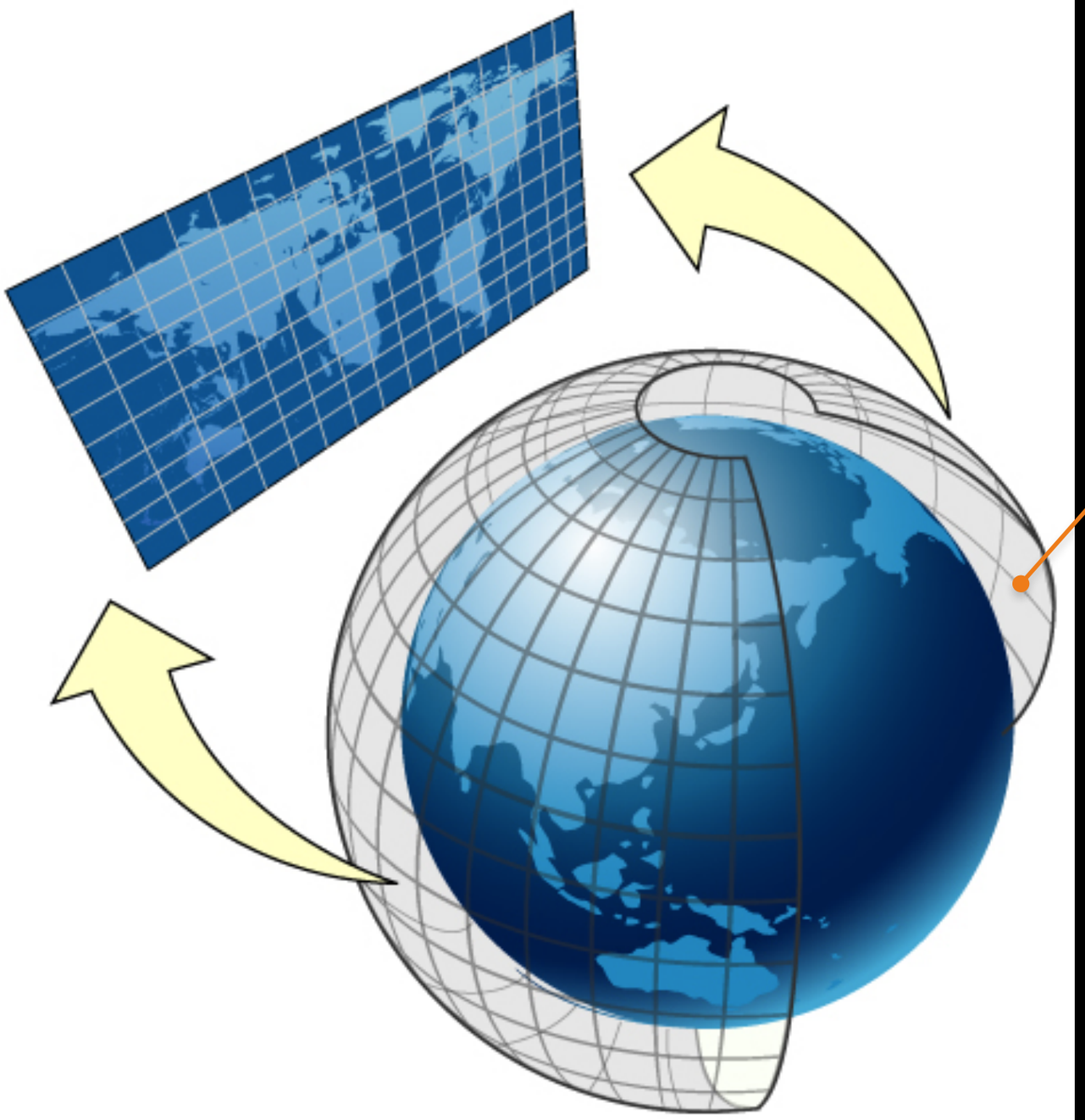
The Map Geometry

Figure 4-1 Mapping spherical data to a flat surface



The Map Geometry

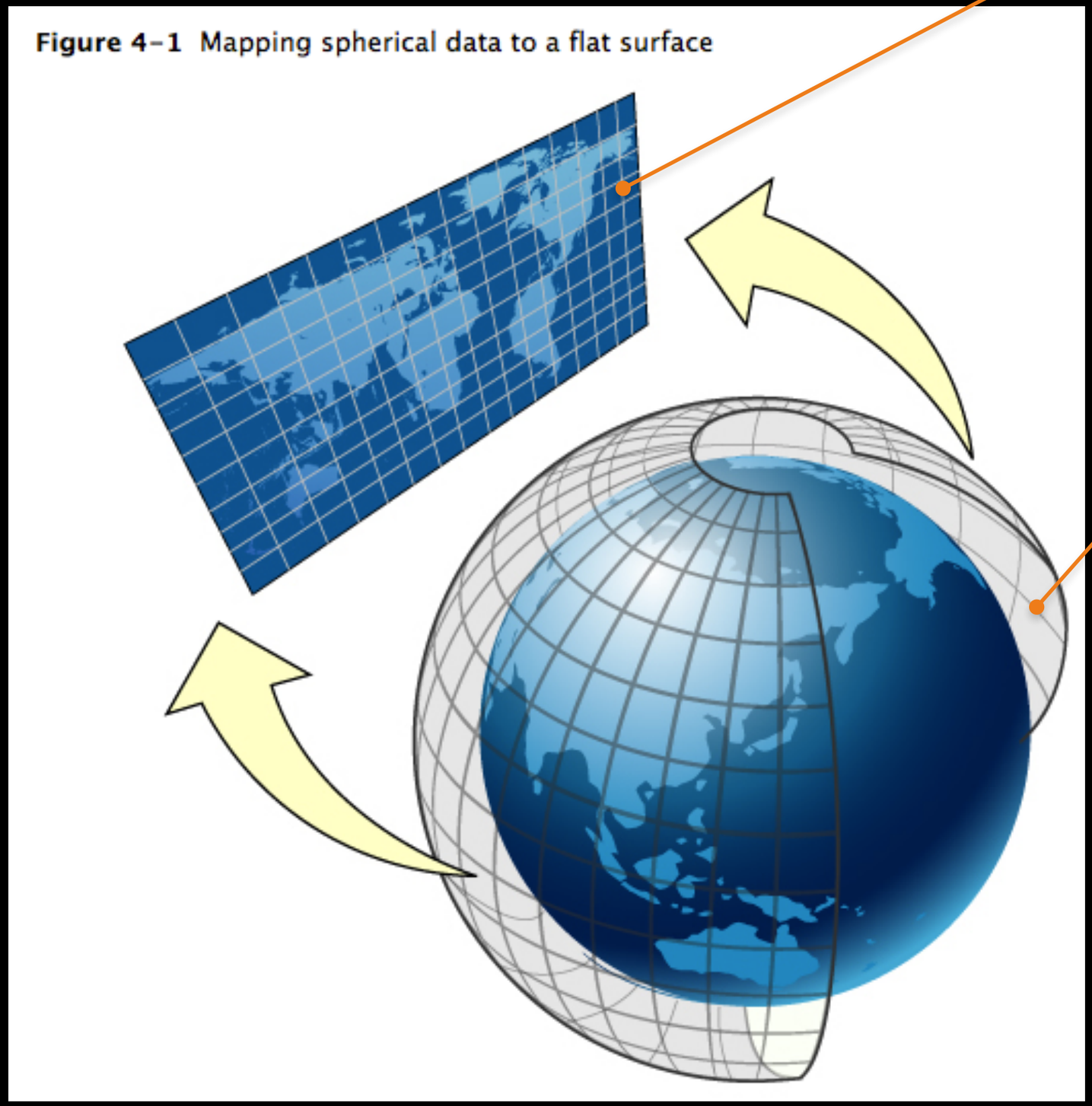
Figure 4-1 Mapping spherical data to a flat surface



Map Coordinate

- CLLocationCoordinate2D
  * latitude
  * longitude
- MKCoordinateSpan
  * latitudeDelta
  * longitudeDelta
- MKCoordinateRegion
  * center
  * span

The Map Geometry



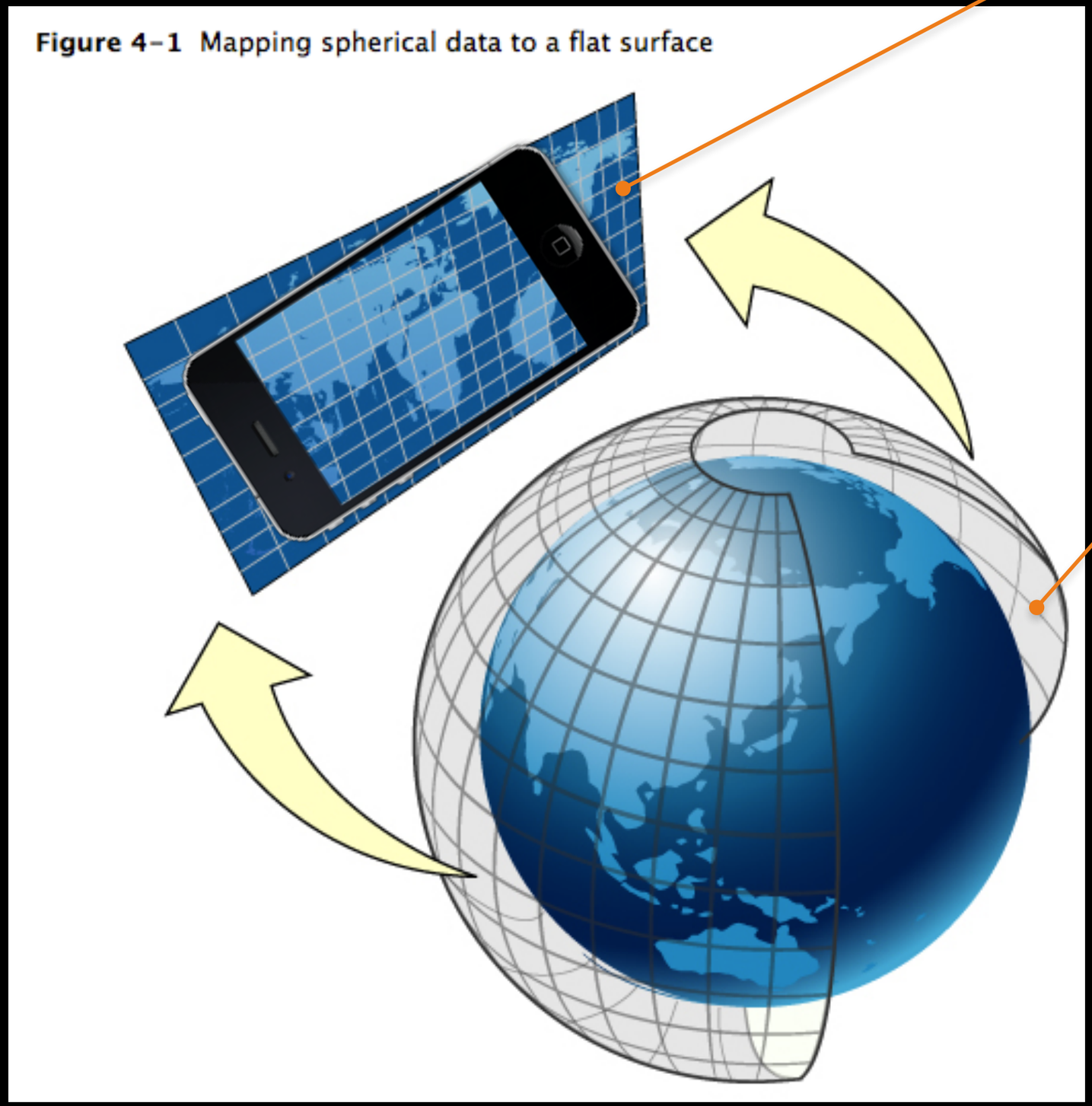
The Mercator Map Projection

- MKMapPoint
  * x
  * y
- MKMapSize
  * width
  * height
- MKMapRect
  * origin
  * size

Map Coordinate

- CLLocationCoordinate2D
  * latitude
  * longitude
- MKCoordinateSpan
  * latitudeDelta
  * longitudeDelta
- MKCoordinateRegion
  * center
  * span

The Map Geometry



The Mercator Map Projection

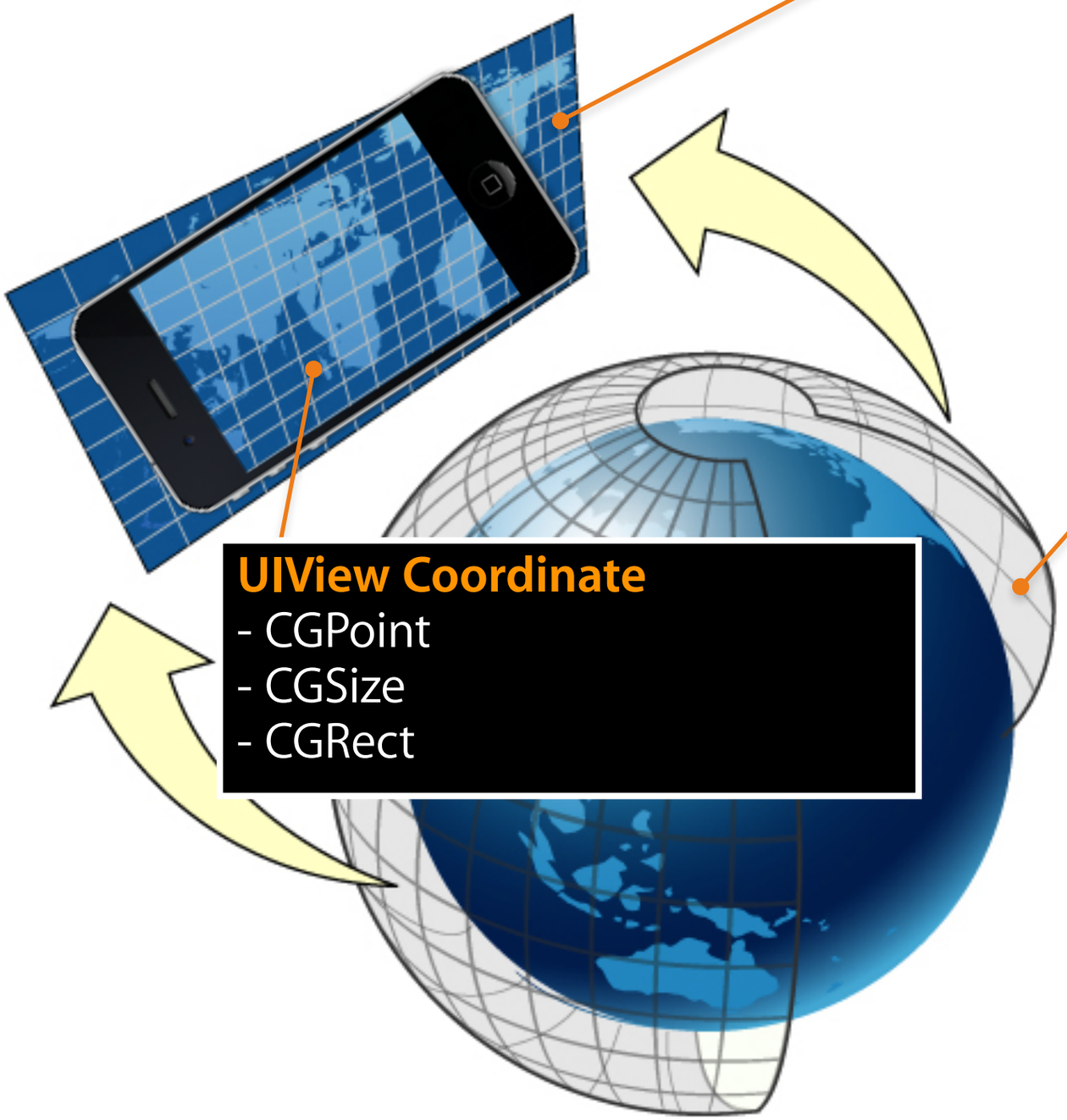
- MKMapPoint
  * x
  * y
- MKMapSize
  * width
  * height
- MKMapRect
  * origin
  * size

Map Coordinate

- CLLocationCoordinate2D
  * latitude
  * longitude
- MKCoordinateSpan
  * latitudeDelta
  * longitudeDelta
- MKCoordinateRegion
  * center
  * span

The Map Geometry

Figure 4–1 Mapping spherical data to a flat surface



The Mercator Map Projection
- MKMapPoint
  * x
  * y
- MKMapSize
  * width
  * height
- MKMapRect
  * origin
  * size

Map Coordinate
- CLLocationCoordinate2D
  * latitude
  * longitude
- MKCoordinateSpan
  * latitudeDelta
  * longitudeDelta
- MKCoordinateRegion
  * center
  * span

MKMapView

- A self-contained interface for presenting map data in an app

- Should never be sub-classed

- Embed it as-is

- Be created in storyboard or programmatically as any other views

The MKMapViewDelegate Protocol

- Map-related update messages

- Request annotation/overlay views and manage interactions with those views

Set the Visible Portion

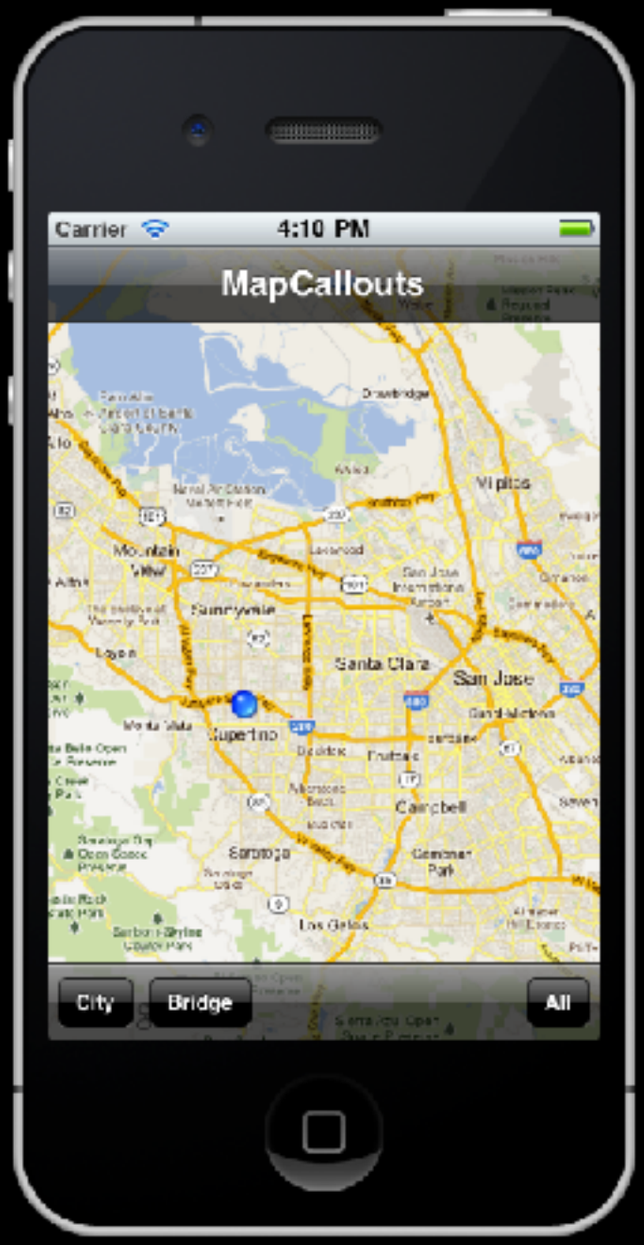
```
//set the center of the map
let centerLocation =
   CLLocationCoordinate2DMake(37.328230, -122.025922);

//set the area
let region =
MKCoordinateRegionMakeWithDistance(centerLocation, 1000,
1000)

self.mapView.setRegion(self.mapView.regionThatFits(region),
animated: true)
```

The Current Location

self.myMapView.showsUserLocation = YES

- The map view uses Core Location to find the user's location and add an annotation to the map

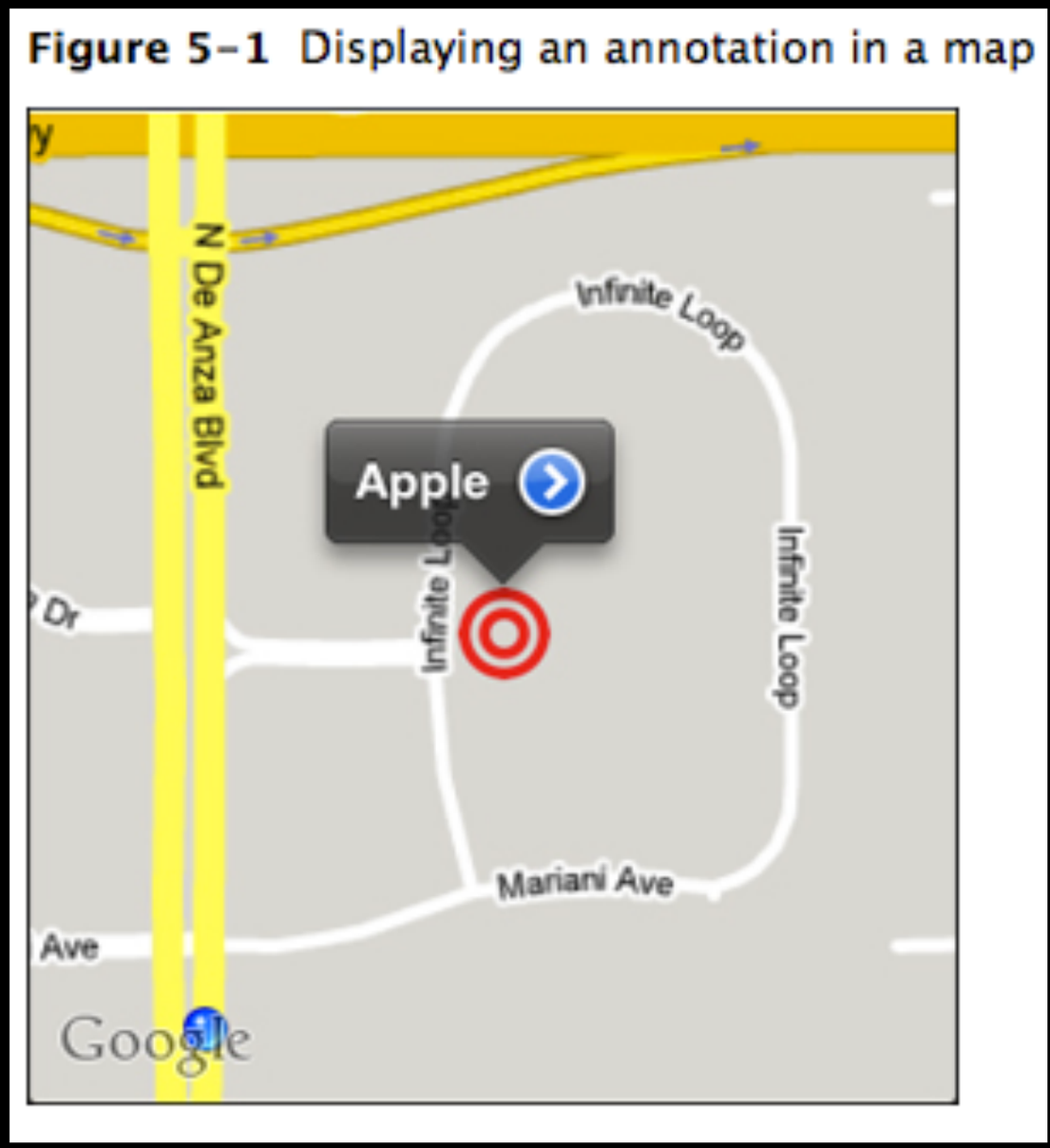
- No updates in the background



Annotate the Map

- Display content over the map that can be defined by a single coordinate point

- Samples: current location, a specific address, a single point of interest



An Annotation

- An annotation object – an object that confirms to the MKAnnotation protocol

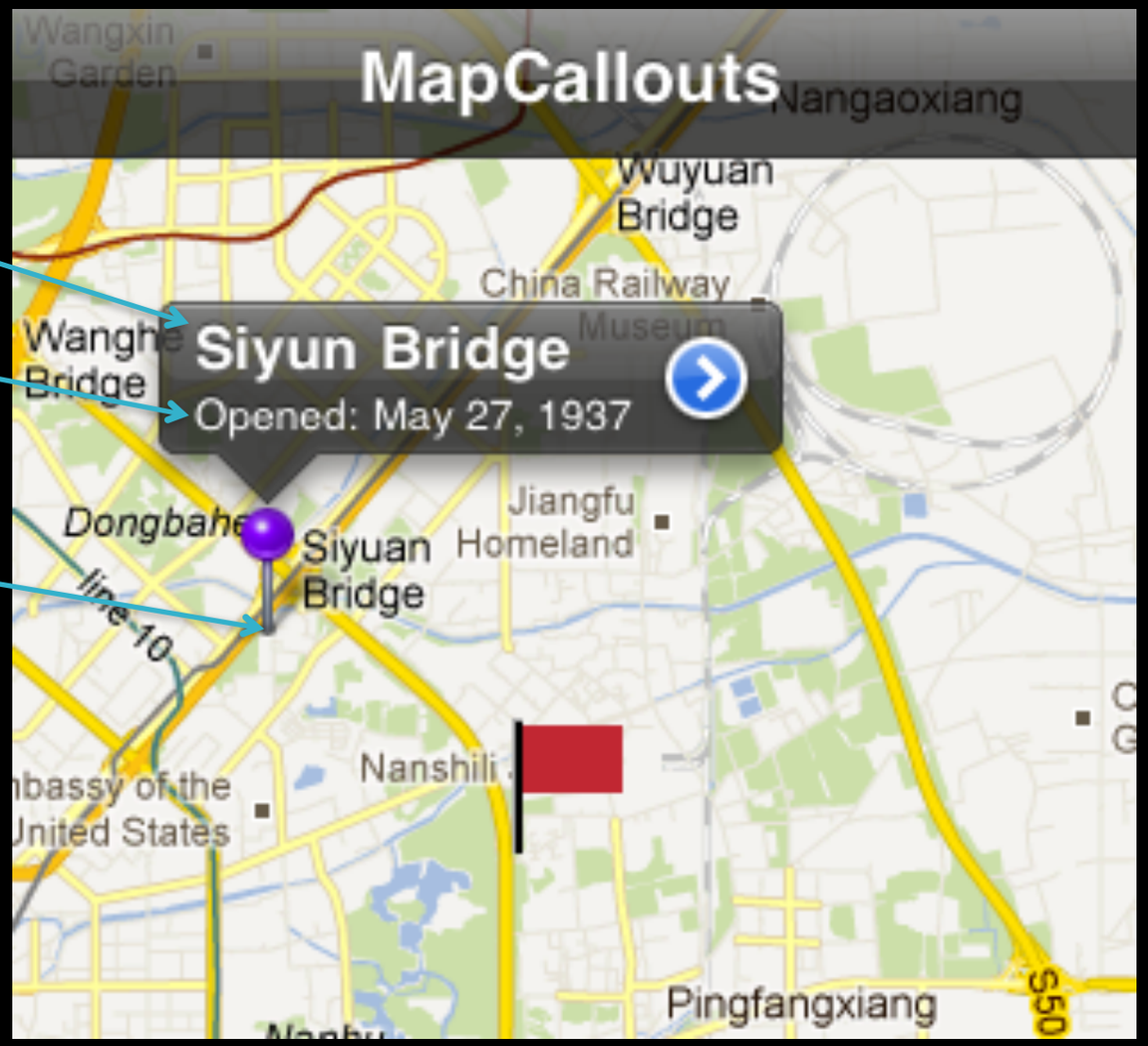
- An Annotation view – the view (derived from the MKAnnotationView class) used to draw the visual representation of the annotation

The MKAnnotation Protocol

- (NSString*)title;

- (NSString*)subtitle;

-(CLLocationCoordinate2D) coordinate;



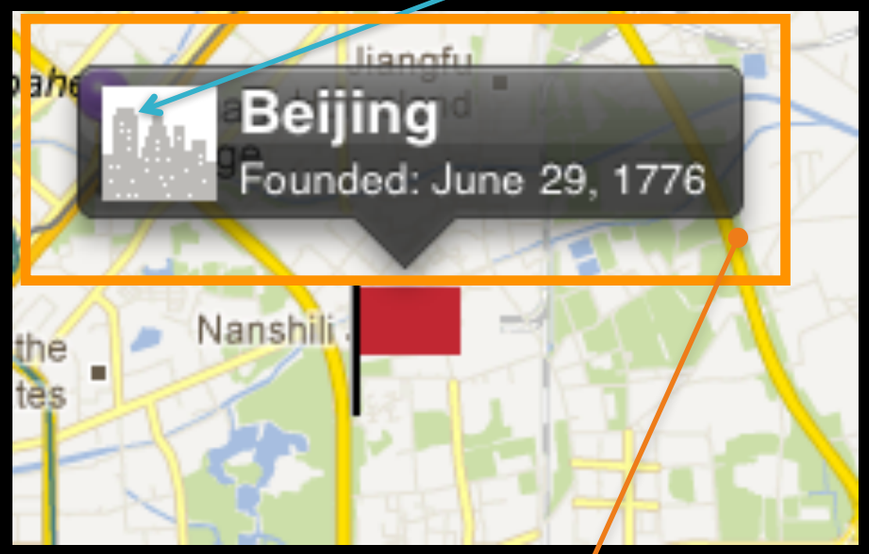
MKAnnotationView

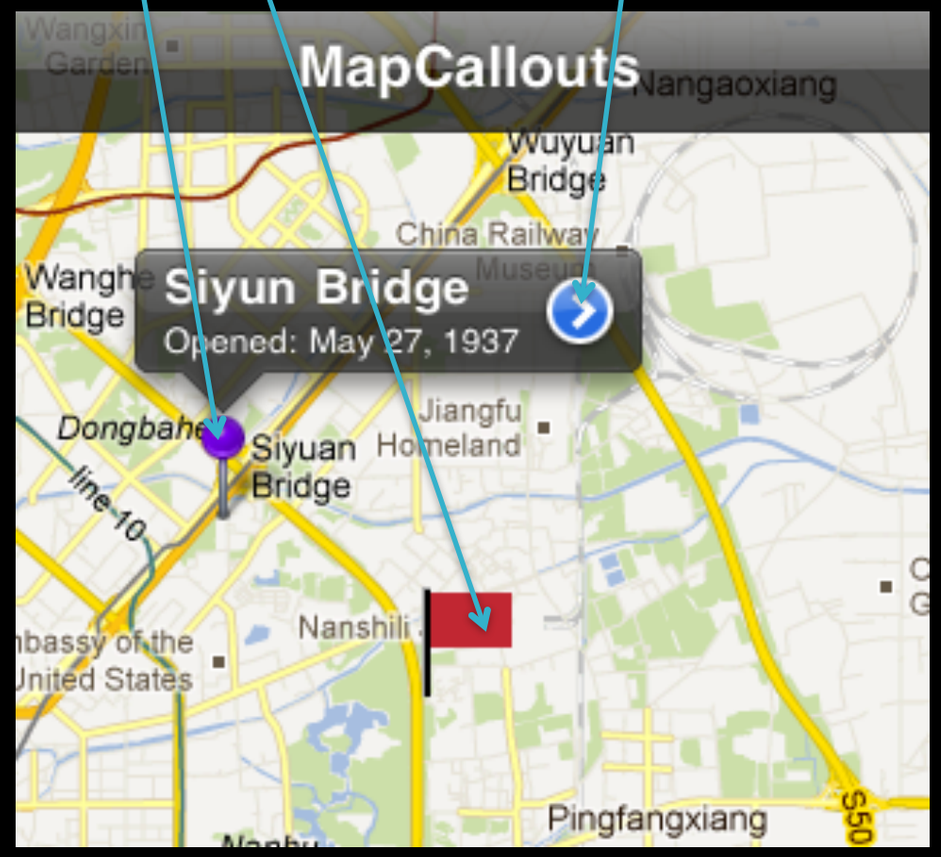
@property (nonatomic, retain) id <MKAnnotation> annotation

@property (nonatomic, retain) UIImage *image

@property (retain, nonatomic) UIView *leftCalloutAccessoryView

@property (retain, nonatomic) UIView *rightCalloutAccessoryView





Display An Annotation

- Add annotations to the map view

```
- (void)addAnnotation:(id < MKAnnotation
>)annotation
- (void)addAnnotations:(NSArray *)annotations
```

- Describe the annotation view for an annotation

```
//the MKMapDelegate method
- (MKAnnotationView *)mapView:(MKMapView *)mapView
viewForAnnotation:(id < MKAnnotation >)annotation
```

Display Overlays on a Map

- Overlays layer content over an arbitrary region of the map
- An overlay is typically defined by multiple coordinates
- Overlays can be contiguous or noncontiguous sets of lines, rectangles, circles, and other shapes
- Overlays can be filled or stroked with color

Overlays


Carrier
4:02 PM
MONTANA
Billings
Bismarck
Yellowstone National Park
SOUTH DAKOTA
WYOMING
NEBRASK
Salt Lake City
UNITED
UTAH
COLORADO
Albuquerque
NEW MEXICO
Phoenix
Tucson
Ciudad Juárez
Nogales
Hermosillo
TEXA


Carrier
10:04 PM
Breadcrumb
塞拉公园
Serra Park
I-280
纪念公园
Memorial Park
Jollyman Park
Legal
AutoNavi

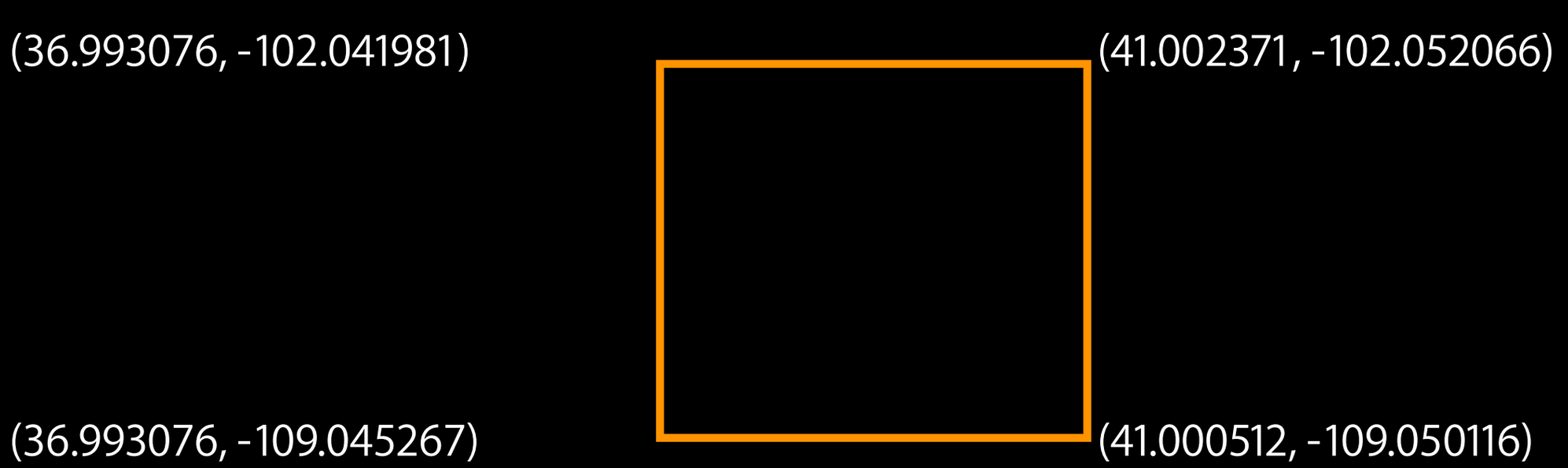
An Overlay

- An overlay object -- an object that conforms to the MKOverlay protocol and manages the data points of the overlay

- An overlay view -- a view (a sub-class of MKOverlayView) used to draw the visual representation of the overlay on the map interface

The MKOverlay Protocol

- An array of the coordinates of the points in the overlay
- Built-in overlays: MKCircle, MKPolygon, MKPolyline



Create a Polygon Overlay Object

```
CLLocationCoordinate2D  points[4];

points[0] = CLLocationCoordinate2DMake(41.000512, -109.050116);
points[1] = CLLocationCoordinate2DMake(41.002371, -102.052066);
points[2] = CLLocationCoordinate2DMake(36.993076, -102.041981);
points[3] = CLLocationCoordinate2DMake(36.99892, -109.045267);
```

```
MKPolygon* poly = [MKPolygon polygonWithCoordinates:points
                                              count:4];
poly.title = @"Colorado";

[map addOverlay:poly];
```

The Overlay View

```
- (MKOverlayView *)mapView:(MKMapView *)mapView
viewForOverlay:(id<MKOverlay>)overlay
{
    if ([overlay isKindOfClass:[MKPolygon class]])
    {
        MKPolygonView* aView = [[MKPolygonView alloc]
                               initWithPolygon:(MKPolygon*)overlay];

        aView.fillColor =
                 [[UIColor cyanColor]  colorWithAlphaComponent:0.2];
        aView.strokeColor =
                 [[UIColor blueColor] colorWithAlphaComponent:0.7];
        aView.lineWidth = 3;

        return aView;
    }

    return nil;
}
```

Local Search — *MKLocalSearch*

- Search for points of interest and display the results on maps
- Search queries: name, address, or type (coffee or pizza) of the locations
- An *MKLocalSearchRequest* bundles the search query
- An *MKLocalSearch* object forms asynchronous search
- The search result — an array of *MKMapItem* objects

Route-based Directions — *MKDirections*

- Access the route-based directions data from Apple's server
- An *MKDirections* object: start and end points of a route
- The Apple's server returns the route-based data
  - walking and driving directions
  - alternate routes

Location-based Services

Location-based Services

- Standard location service: a configurable way to get the current location and location updates

- Significant-change location service (iOS 4.0+): low-power way to
  - Get current location
  - Be notified of the changes to the location

- Region monitoring: monitor boundary crossings at defined area (iOS 4.0+)

Get the Current Location

- The standard location service
  - Configurable & general-purpose
  - Is supported in all iOS versions
  - No update when the app is not running

- The significant-change location service
  - On devices with cellular radios
  - Wakes up an app from suspended or not running
  - Low-power

The Standard Location Service

- A power-intensive operation: query cell towers, Wi-Fi hotspots, and/or GPS satellite

- Take seconds to get readings

- Not suitable for regular position updates

Location Service Availability

- The user disables location services in Settings
- The user denies location services for an app
- The device is in Airplane mode

Check the Availability

```
let status = CLLocationManager.authorizationStatus()
if status == .restricted || status == .denied {
    showLocationServicesDeniedAlert()
} else {
    if status == .notDetermined {
        locationManager.delegate = self
        locManager.requestWhenInUseAuthorization()
    } else {
        startLocationService()
}
```

Start the Standard Location Service

```
if !isUpdatingLocation {
    locManager.delegate = self
    locManager.distanceFilter = kCLDistanceFilterNone
    locManager.desiredAccuracy =
kCLLocationAccuracyThreeKilometers
    locManager.startUpdatingLocation()
}
```

Receive the Location Data

```
func locationManager(_ manager: CLLocationManager,
                     didUpdateLocations locations: [CLLocation]) {
    let newLocation = locations.last!

    if newLocation.timestamp.timeIntervalSinceNow < -5 { return }

    if newLocation.horizontalAccuracy < 0 { return }
```

```
    if let location = curLocation, newLocation.horizontalAccuracy >
location.horizontalAccuracy {return}

    curLocation = newLocation
     print("didUpdateLocations \(curLocation)”)
}
```