Assignment 1 is Out – Policies

You must typeset, use your favorite text/doc editor:

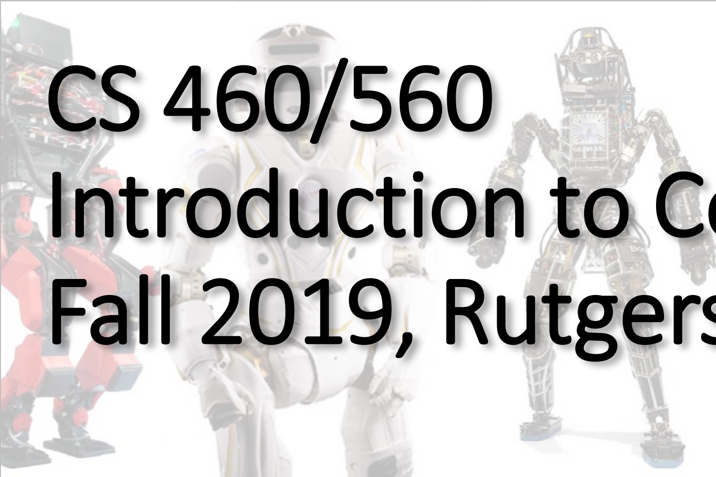
- ⇨ Word (very easy to type math expressions)
- ⇨ Latex (very easy to type math expressions)
- ⇨ Google doc
- ⇨ Plain text + math formula
- ⇨ The point is not to look nice, but legible
- ⇨ You may draw figures by hand and digitize them

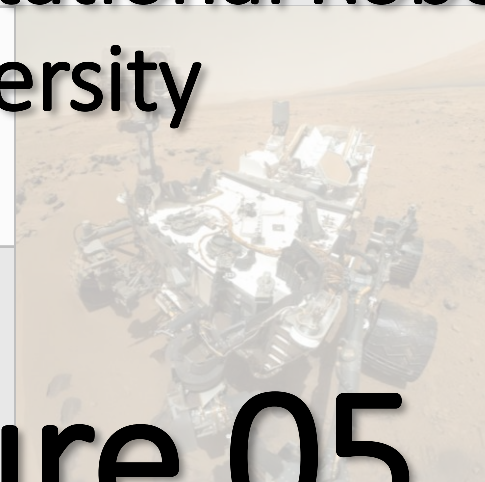
Submit a single PDF for the written part

- ⇨ There could be some coding tasks in assignments
- ⇨ Unless otherwise stated, you don’t need to submit your code for HW assignments
- ⇨ Follow instructions given in each assignment

Collaboration is encouraged

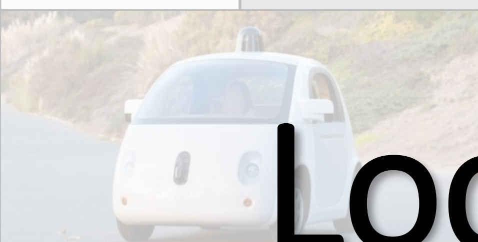
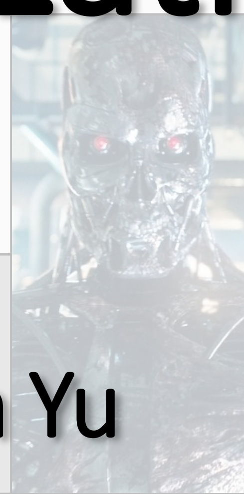
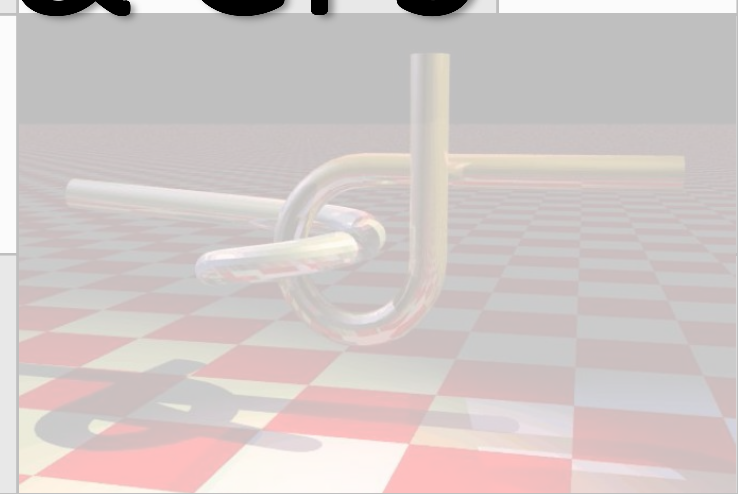
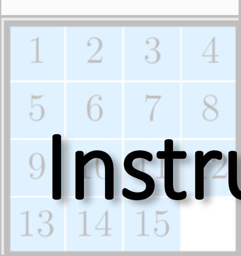
- ⇨ You should disclose with whom you have discussed your problems
- ⇨ This helps especially if multiple people/groups make identical mistakes





CS 460/560
Introduction to Computational Robotics
Fall 2019, Rutgers University

Lecture 05
Localization & GPS

Instructor: Jingjin Yu

Outline

Basic localization methods

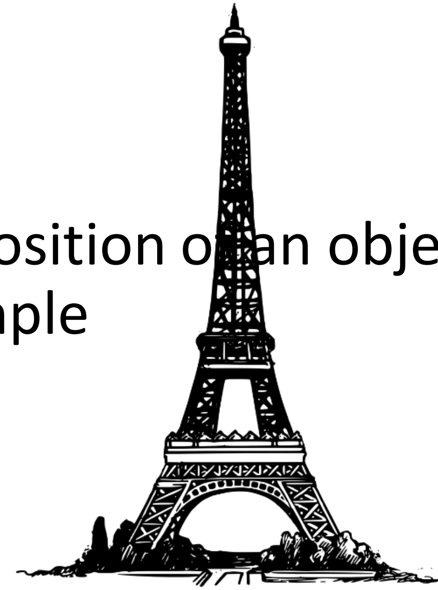
- ⇨Triangulation
- ⇨Trilateration

Global Positioning System (GPS)

- ⇨Overview
- ⇨Enabling technologies
- ⇨Graph search

Localization

Localization: determining the position of an object with respect to some reference frame, for example



Some basic methods

- ⇨Triangulation: localization using measured **angles**
- ⇨Trilateration: localization using only measured **distances**

Trilateration is often mistaken as triangulation!

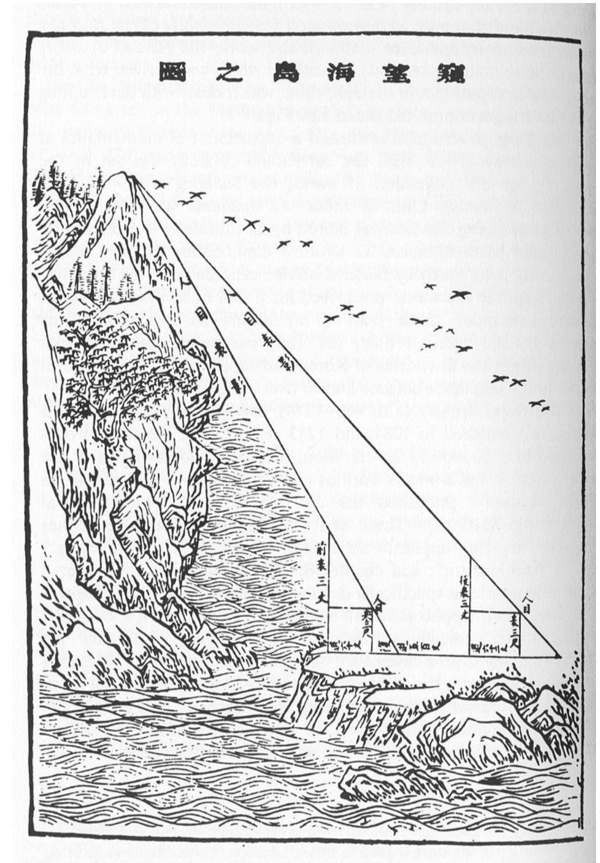
- ⇨E.g., cell tower based localization uses mainly trilateration (more on this later)

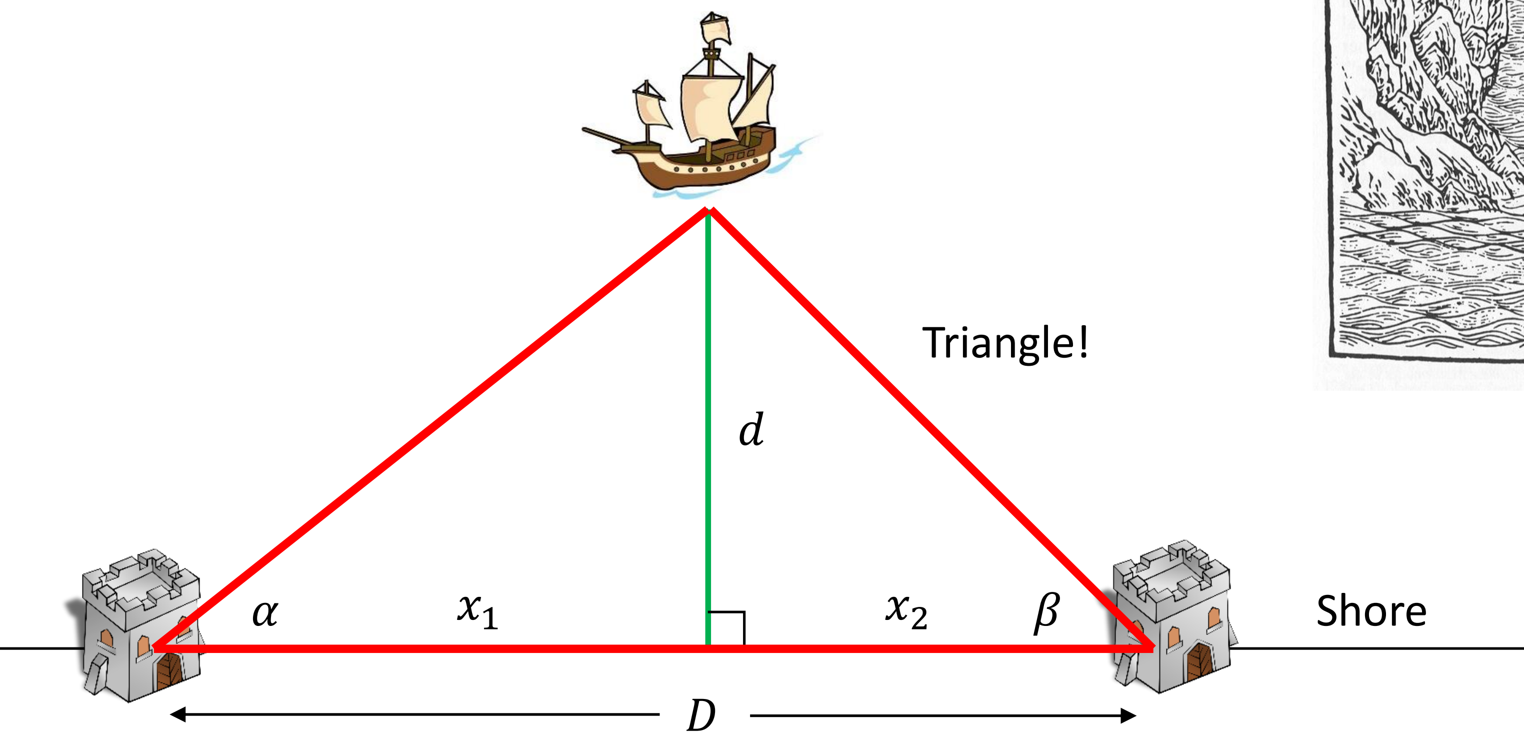
Localization with Triangulation

Triangulation is an ancient technique

- ⇨Known for about 2000 years

The principle is reasonably straightforward





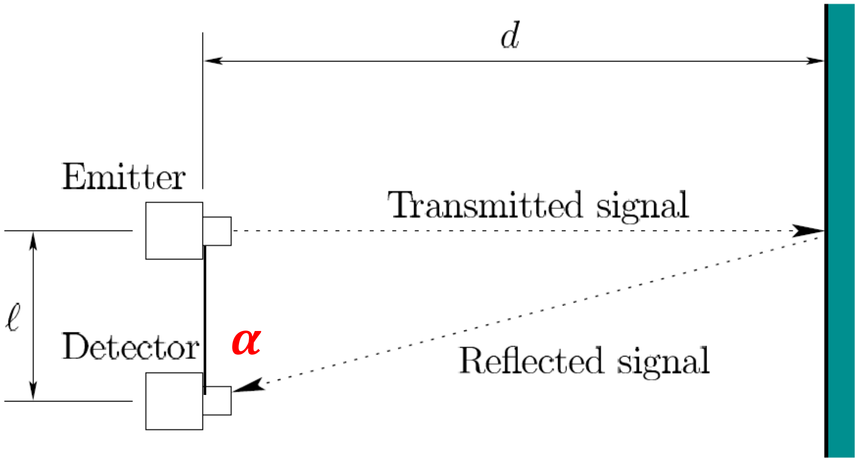
Sextant

$$\tan\alpha = \frac{d}{x_1}, \tan\beta = \frac{d}{x_2}, x_1 + x_2 = D \quad \Rightarrow d = \frac{D}{\dfrac{1}{\tan\alpha} + \dfrac{1}{\tan\beta}}$$

Images: Wikipedia

Localization with Triangulation, Continued

Modern laser-based measurement



$$\Rightarrow d = \ell \tan \alpha$$

There are more complex sensors building on the principle

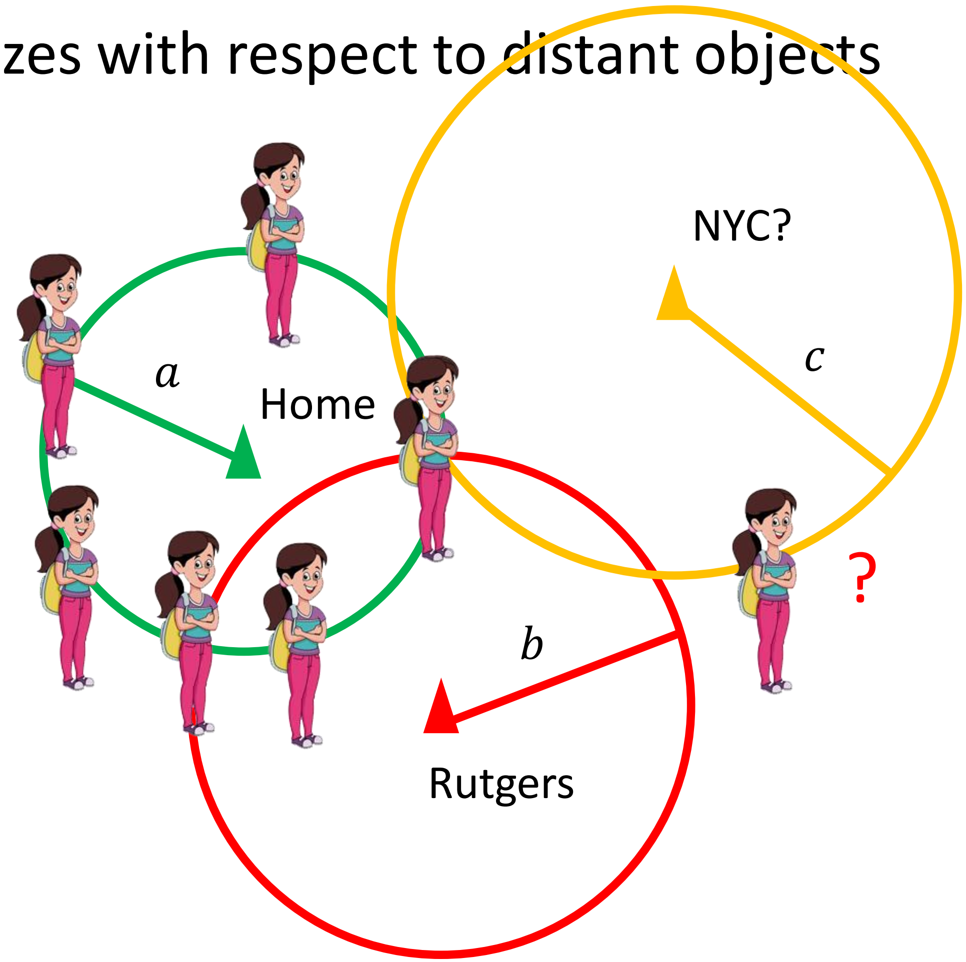
Image from Mobile Robotics by LaValle

Localization with Trilateration

Often, triangulation locates the position of a distant object

Trilateration instead localizes with respect to distant objects

- ⇨2D example
- ⇨If we know the distances
- ⇨Where on the map?



Note that both techniques can be used for localization and for measuring position of a distant object

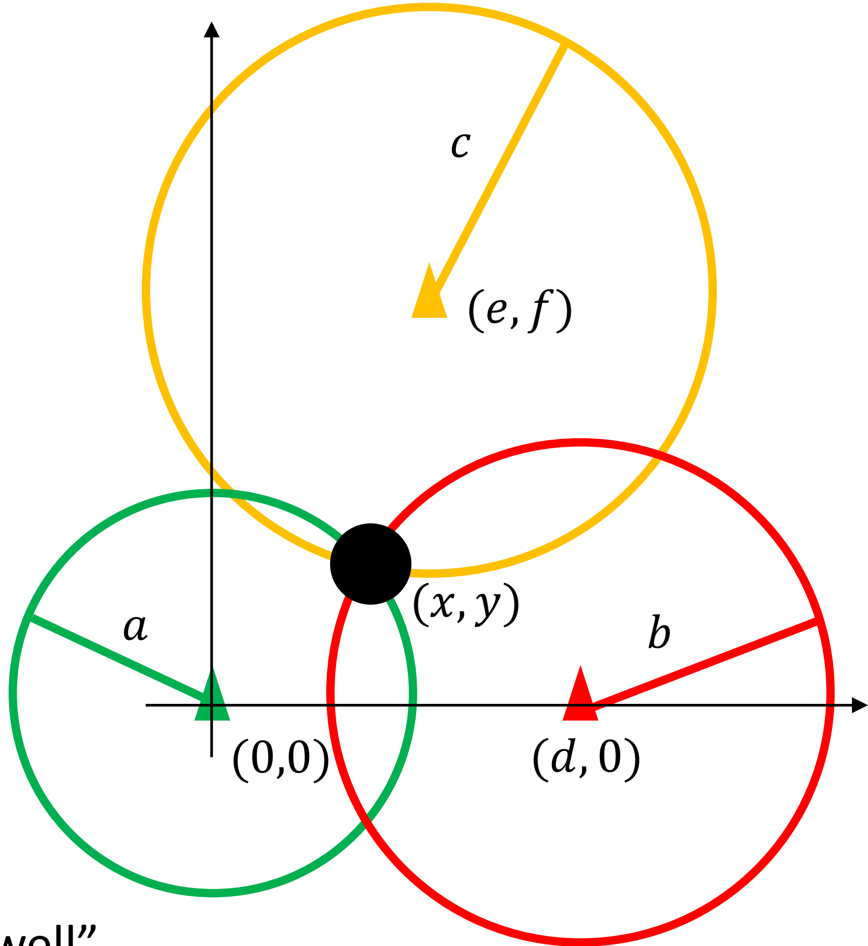
Localization with Trilateration, Continued

Let’s do some math…

$$x^2 + y^2 = a^2$$
$$(x-d)^2 + y^2 = b^2$$
$$(x-e)^2 + (y-f)^2 = c^2$$

Note that from first two equations

$$x = \frac{a^2+d^2-b^2}{2d}$$
$$y = \pm\sqrt{a^2-x^2}$$



Using the third equation, figure out y

Problematic cases?

⇨The three cycles must surround (x, y) “well”

⇨I.e., note like

Applications: GPS, …

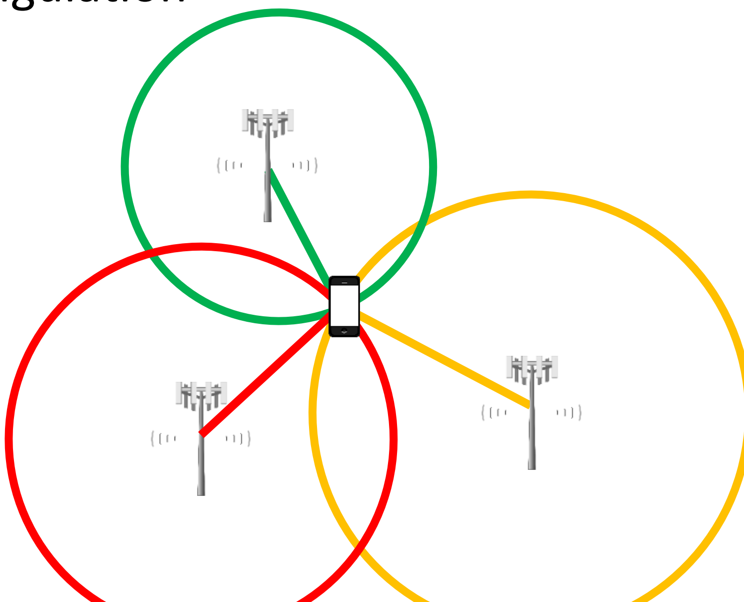
Triangulation vs Trilateration

The difference between triangulation and trilateration is clear

- ⇨ Triangulation uses **angle** measurement
  - ⇨ Often, it refers to methods using both angles and distances
    - ⇨ Triangulateration
- ⇨ Trilateration infers location only through **distance** measurement
  - ⇨ In particular, uses three (or some other # of) distances to localize
  - ⇨ If you do not have enough distances, can still use it!

Somehow, trilateration is often mistaken for triangulation

- ⇨ Exacerbated by the misuse of “cell tower triangulation”
- ⇨ The main principle is trilateration!

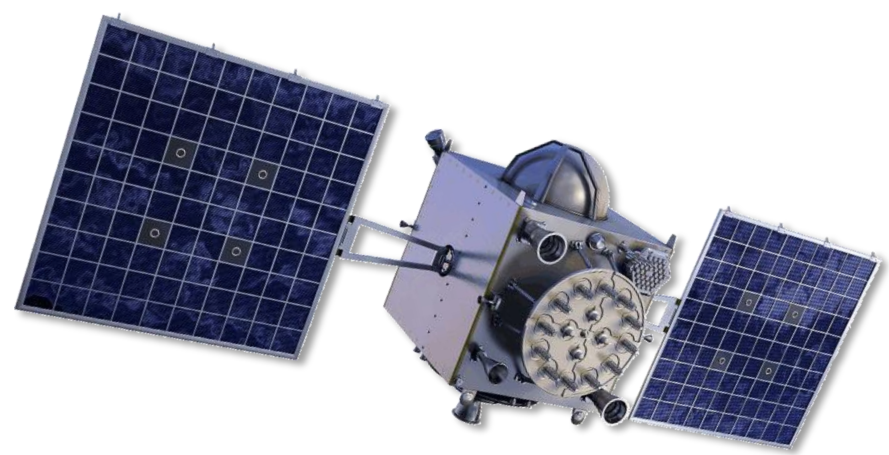


GPS: Localization with **Trilateration**

The Global Positioning System (GPS) is a **space-based navigation system** that provides **location and time information** in all weather conditions, **anywhere on or near the Earth where there is an unobstructed line of sight to four or more GPS satellites**.

GPS is a U.S. owned system. Many other countries have their own

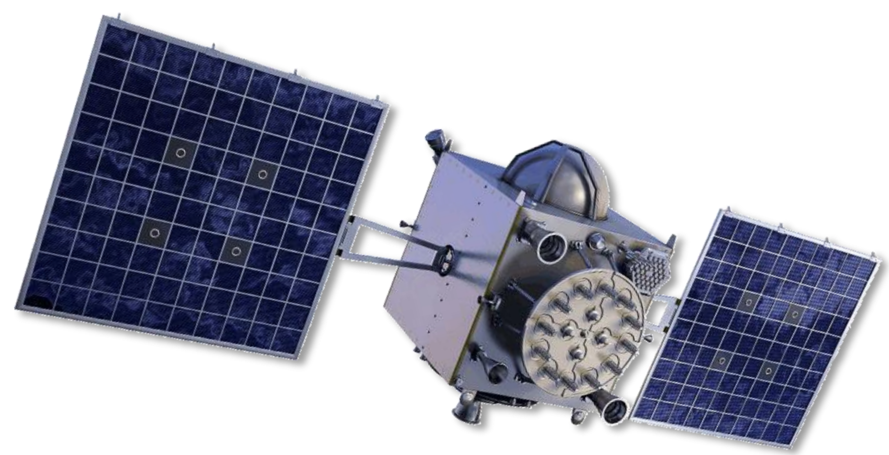
- ⇨GLONASS (Russia, about the same time)
- ⇨Galileo (EU), BeiDou (China)
- ⇨IRNSS (Indian, regional), Quasi-Zenith (Japan, regional)



Text source: Wikipedia
Image source: Garmin, teletrac.com

A Brief History of GPS

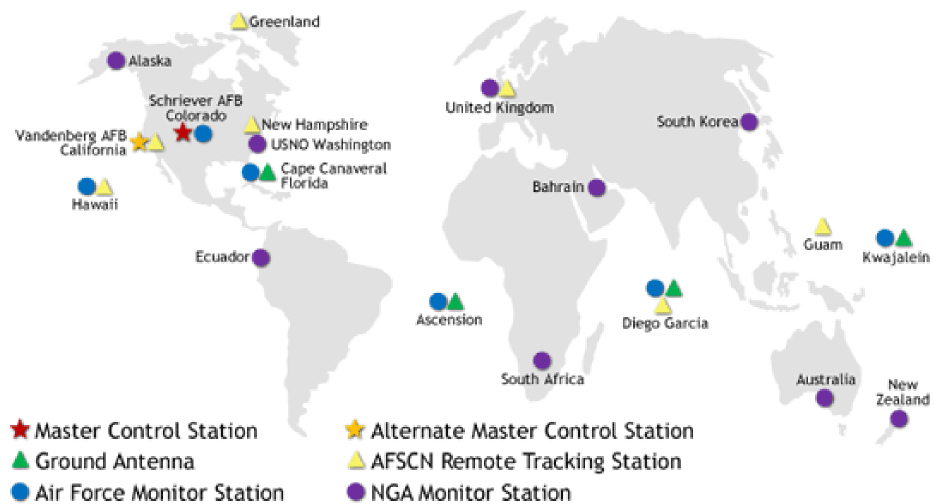
- ⇨1960s – conceptualization in U.S. military
- ⇨1978 – the launch of the first GPS satellite
- ⇨1989 – The introduction of the first hand-held GPS receiver
- ⇨1992 – Used in Operation Desert Storm
- ⇨1996 – Under President Clinton, GPS became free for civilian use
  - ⇨Great! There are many things I would not be able to do without GPS!
  - ⇨Driving to Rutgers, track my runs, outdoor exploration, …



Source: loc.gov
Image source: Garmin, teletrac.com

Segments of GPS

- ⇨GPS has three segments
- ⇨**Space segment (satellites)**:
  - ⇨**24** GPS satellites needed, in six obits, 4 each
  - ⇨About 20,200 kilometers altitude
  - ⇨Ensures at least 4 satellites are visible anywhere on earth
  - ⇨Currently **31** GPS satellites for redundancy
- ⇨**Control segment (stations):**
  - ⇨Controls the satellites
  - ⇨Make sure they work well



- ⇨ **User segment (receivers):**

Image source: Garmin, gps.gov

How does Global Positioning System Work?

The principle is **trilateration**: determining absolute or relative location of points by **measurement of distance**

⇨We have seen 2-dimentional trilateration

⇨What about GPS? How many distances?

⇨GPS is three-dimensional

⇨4+ satellites!

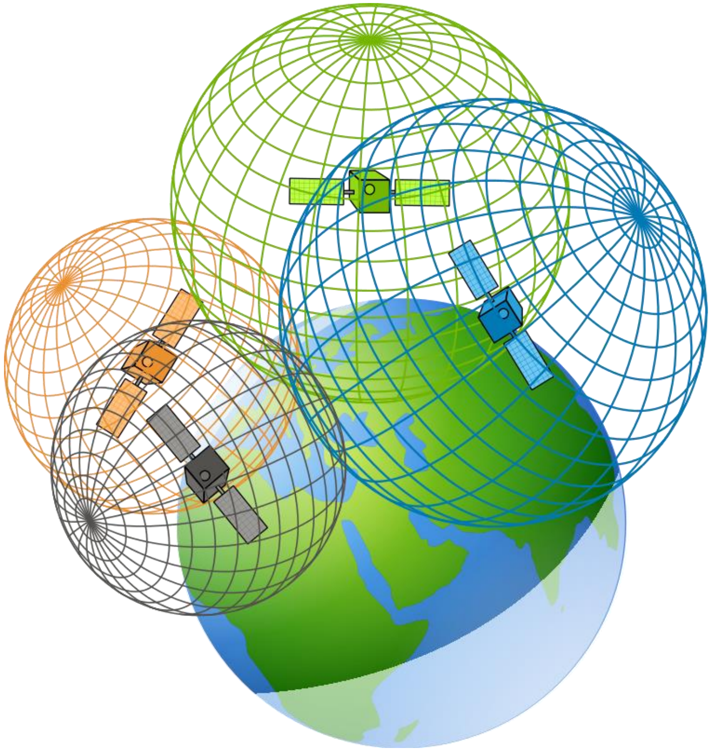
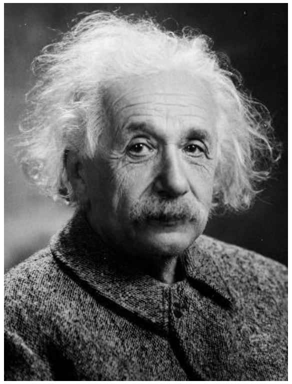


Image source: moorefamrsbg.org

Many Additional Technologies

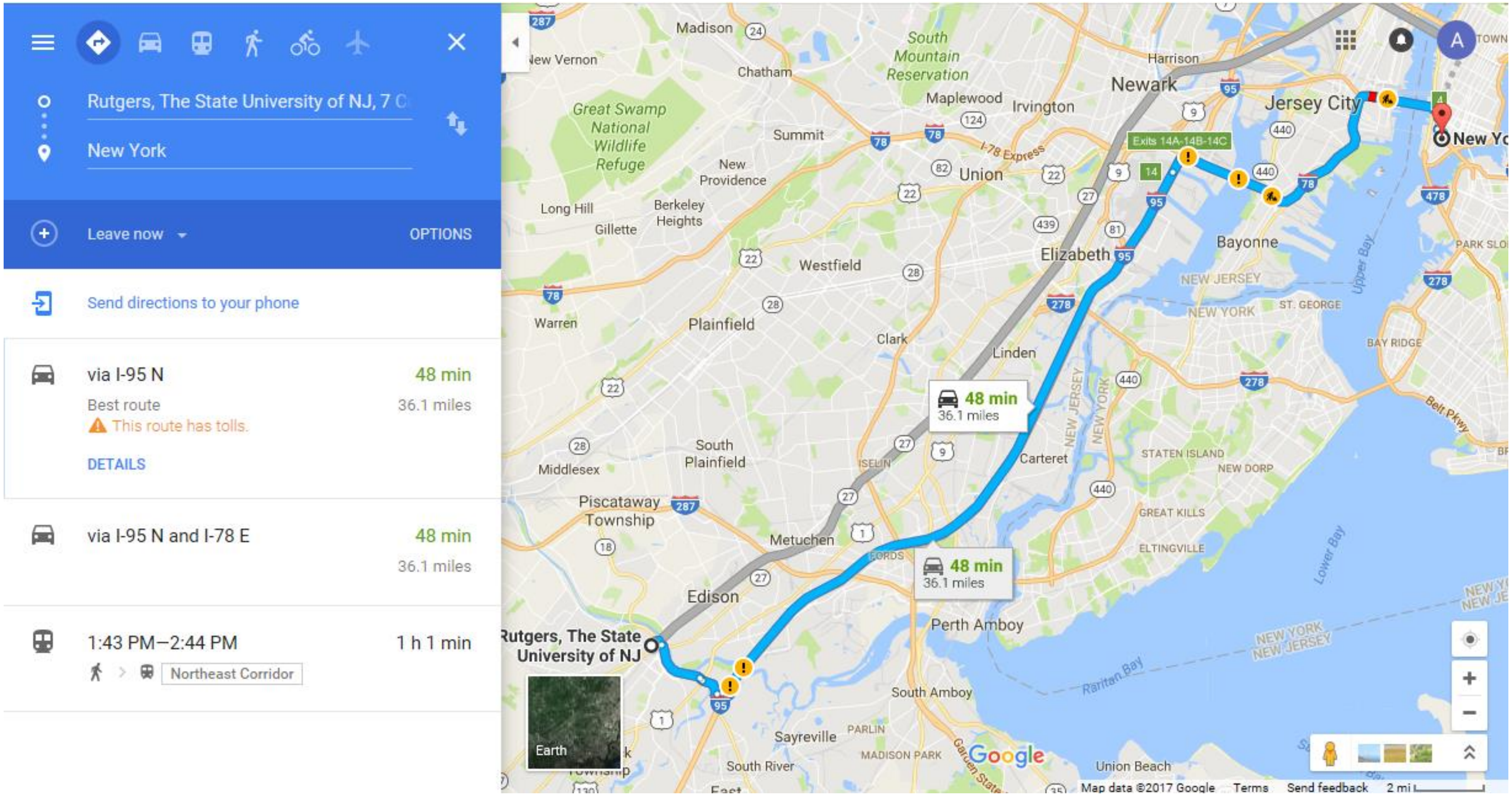



- ⇨GPS is certainly rocket science
- ⇨Satellites are highly complex
  - ⇨Must have a very accurate clock to encode its signal
  - ⇨Also uses Einstein's special and general theories of relativity
  - ⇨Special theory of relativity: clocks on faster moving objects are slow
    - ⇨For GPS satellites moving very fast, ~7 microseconds slower
  - ⇨General theory of relativity: clocks closer to massive objects are slower
    - ⇨Clocks on earth are ~45 microseconds slower
  - ⇨A total of ~38 microseconds difference every day
  - ⇨GPS must have clocks at nanosecond accuracy
- ⇨Even with these, error on the ground can be about 30 meters
  - ⇨Something called Kalman filter is used to reduce the error to about 1-5 meters

Image source: aerospace-technology.com, Wikipedia

Route Planning in GPS Navigation

⇨ Now we know where we are on a map. How do we get to our desired destination?



⇨ For doing this automatically, we need **search algorithms**

What is an Algorithm?

- ⇨What is an algorithm?
- ⇨Algorithm: a self-contained step-by-step set of operations to be performed (by machine, human, and so on)
- ⇨Or simply, a finite set of “code”
- ⇨In a nutshell, an algorithm is a set of **operations** that manipulates some **data**.
- ⇨Algorithms themselves are straightforward
  - ⇨Even computers can run them!
- ⇨Understanding and designing good algorithms can sometimes be challenging
- ⇨In our case: **search algorithm**
  - ⇨**Data structure**: a **graph**
  - ⇨**Operations**: **search** over the graph

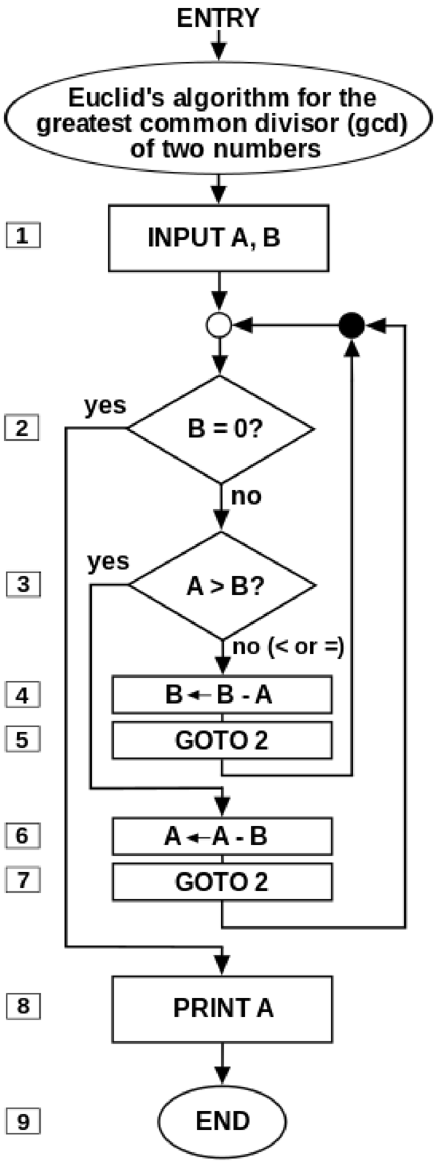
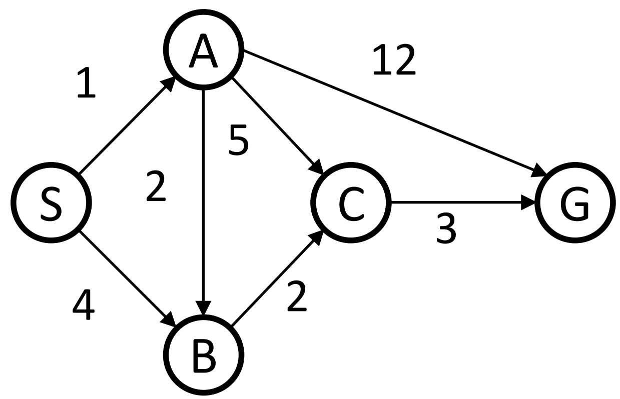


Image source: Wikipedia

Graph: Nodes (Vertices) and Edges

⇨A graph has a set of nodes and edges, e.g.,



⇨Nodes: $V = \{S, A, B, C, G\}$

⇨Nodes are like intersections of roads

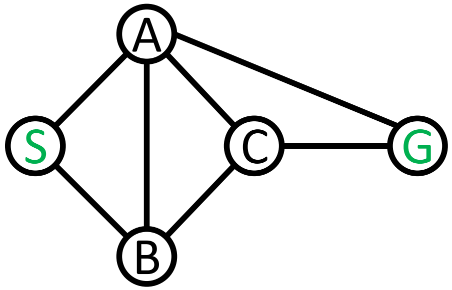
⇨Edges: $E = \{(S, A), (S, B), (A, B), (A, C), (A, G), (B, C), (C, G)\}$

⇨They can have some distances associated with them

⇨Edges are like roads connecting the nodes

⇨We can also allow undirected edges

A Generic Graph Search Algorithm

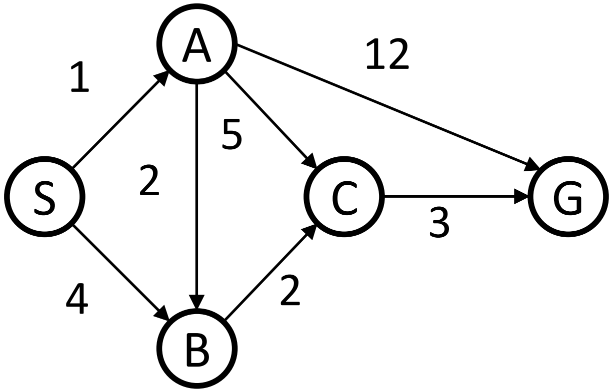



```
AddToQueue(S);     // Add S to a queue
while(QueueNotEmpty())
    x = GetFrontOfQueue();          // Retrieve the front of the queue
    if(x.processed == true) continue;   // Do not work on a node twice
    x.processed = true;             // Mark x as "processed"
    if(x == G) return solution;     // Return if x is goal
    for each neighbor n of x        // Add all neighbors of to the queue
        if(n.processed == false) AddToQueue(n)
return failure;
```

- ⇨In a nutshell, we start with S and maintain a queue of nodes
- ⇨Then repeatedly doing the same operation of processing nodes
- ⇨The most important operation is **AddToQueue()**
    - ⇨Decides which node is in the front of the queue

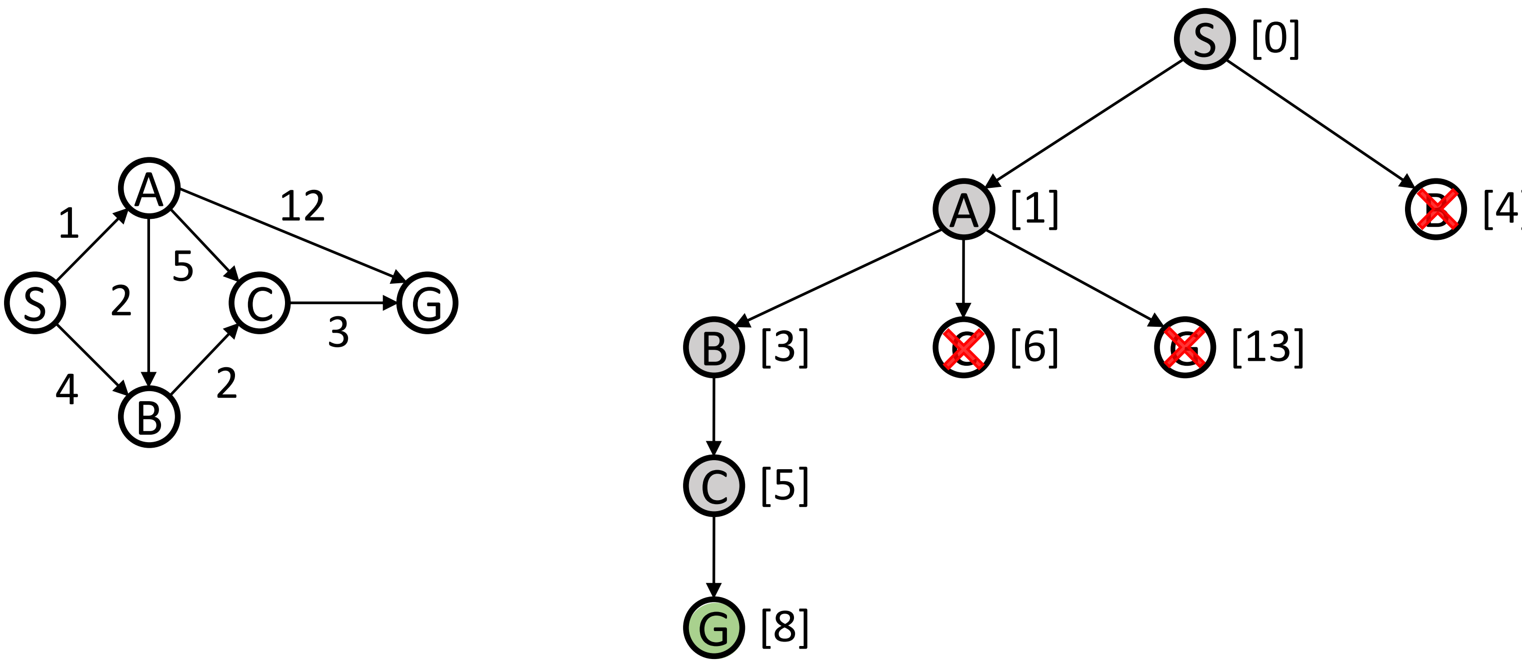
Priority Queue

- We will introduce a very simple **priority queue**
  - Basically, we compute the distance of a node from S
    - We do this one step at a time
    - Remember, the edges are like roads and have distances
    - The distances may not be the best initially
  - Node with smaller distance values is put in the front of the queue
- This yields uniform-cost search
- Let's work with an example



Uniform-Cost Search

Maintain queue order based on current cost



⇨ Produces an **optimal** path!

⇨ This is basically Dijkstra's algorithm

⇨ We will revisit search algorithms in more detail later