DSCI 554 LECTURE

DEPTH PERCEPTION AND 3D DESIGN, MAPS

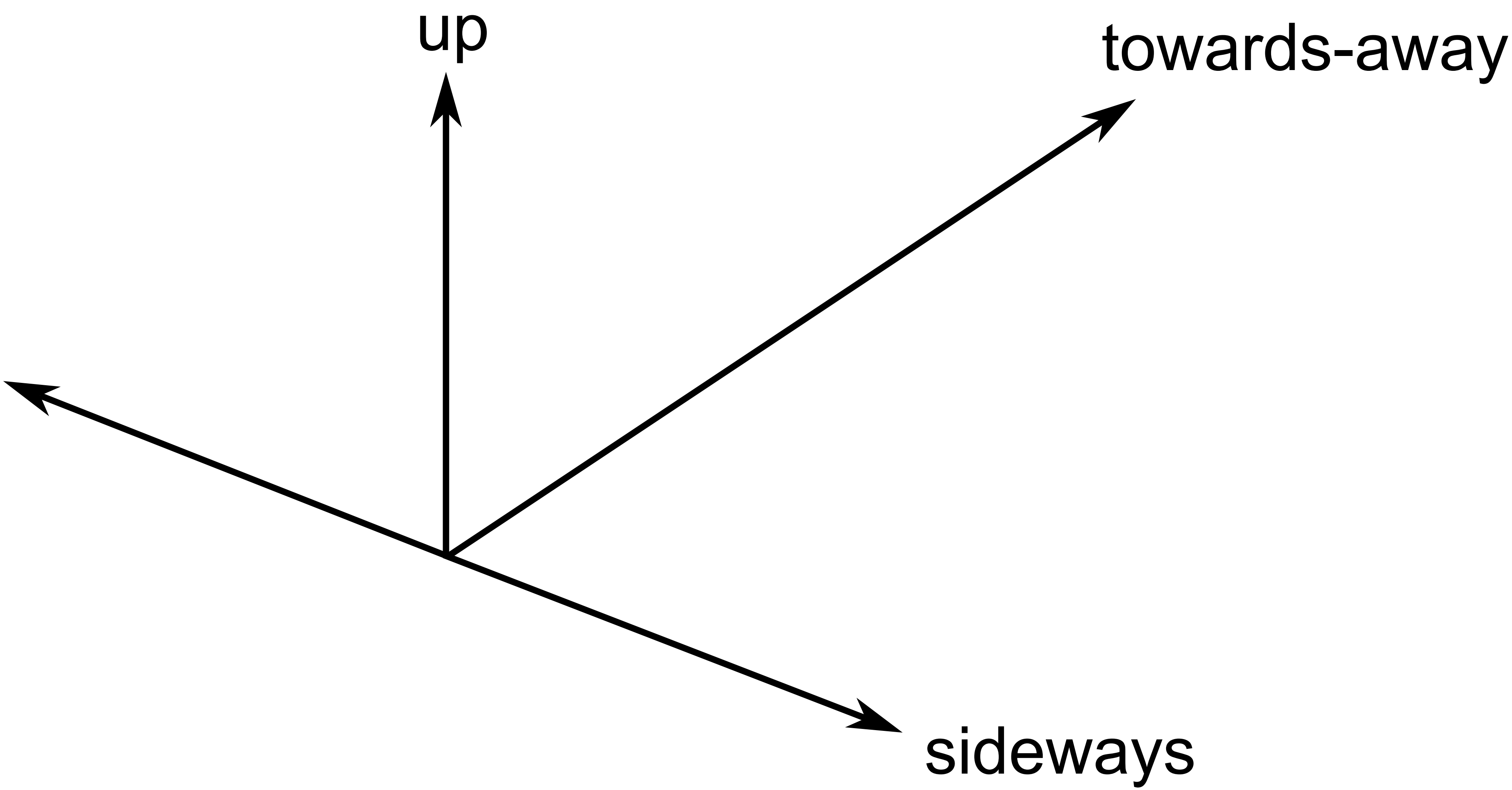
Dr. Luciano Nocera

USC Viterbi
School of Engineering
Integrated Media Systems Center

OUTLINE

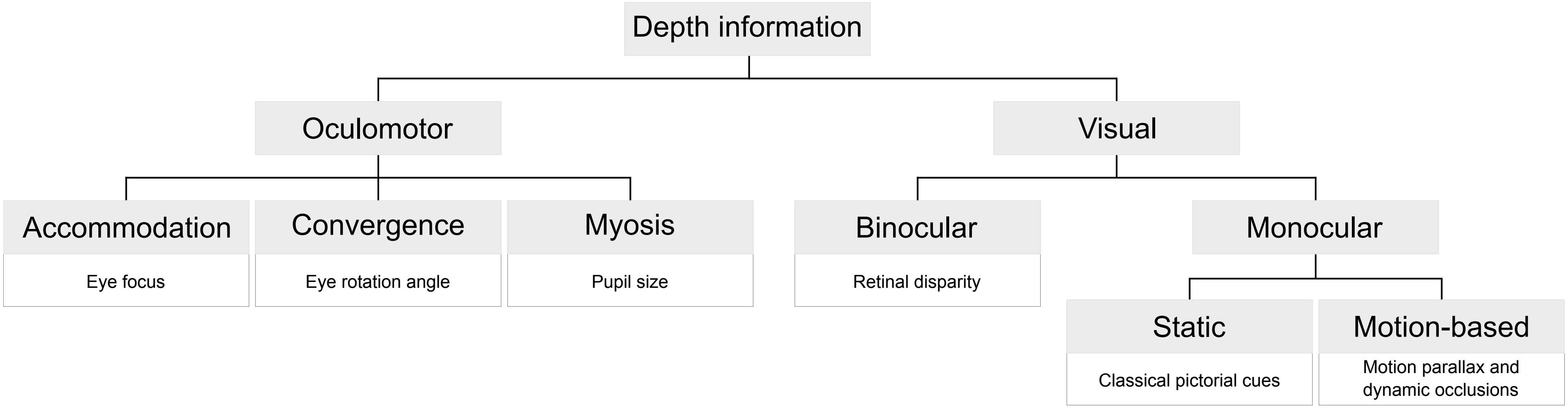
- Depth perception and design considerations
- Introduction to maps
- How maps are built?
- Maps projections and their uses
- Working with maps

PERCEPTUAL EGOCENTRIC SPACE



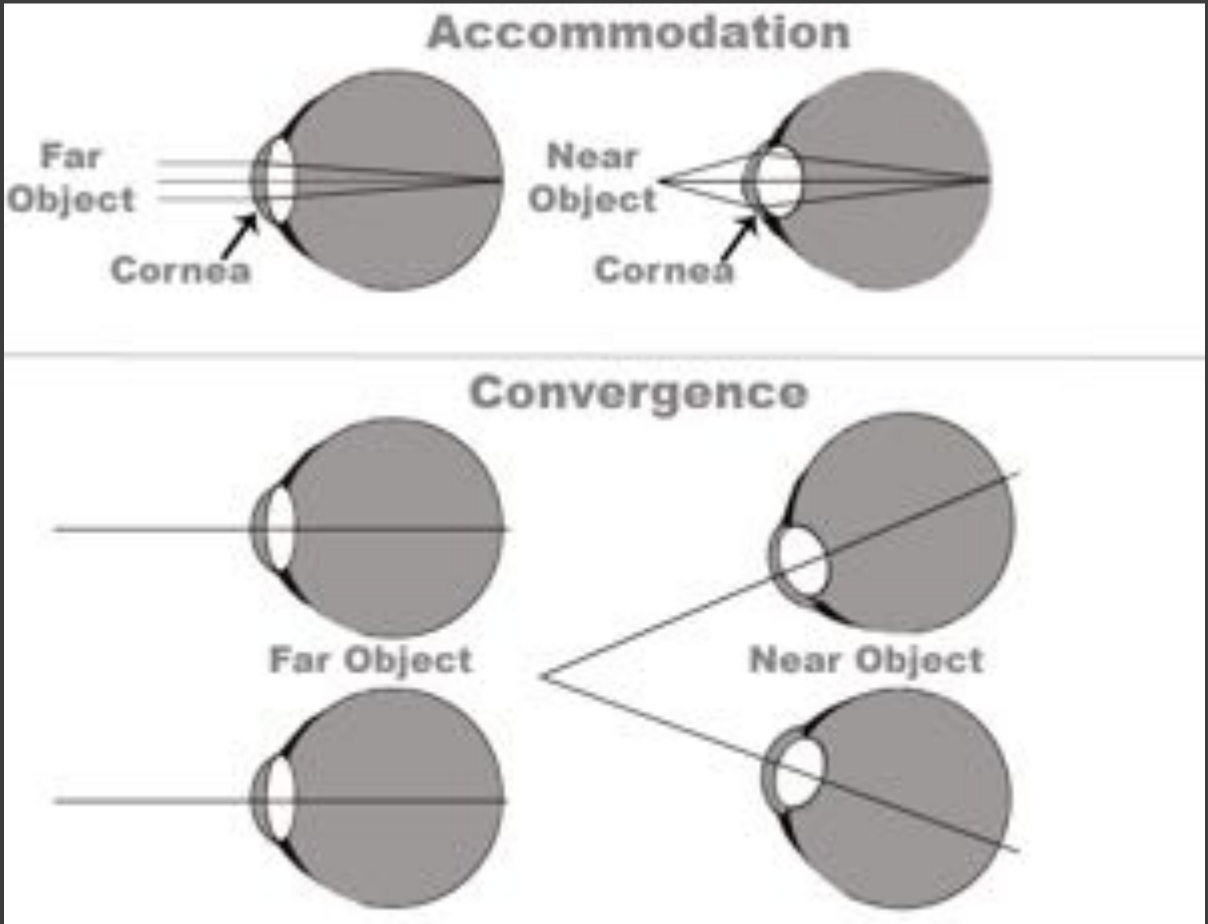
Object occlusions decrease depth information

DEPTH INFORMATION AND CUES



M. J. Tovee, An introduction to the visual system, Cambridge Univ. Press, 1996

OCULOMOTOR

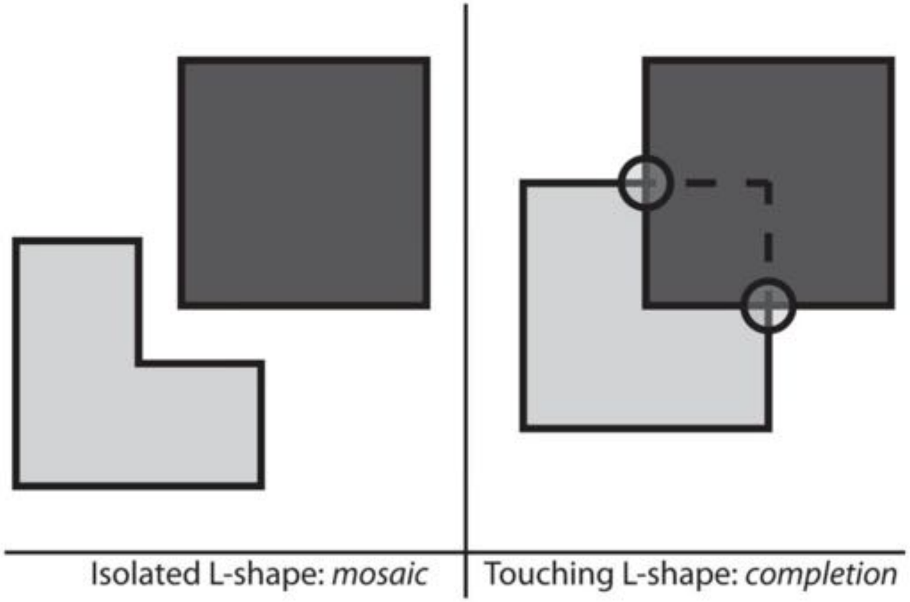


- Accommodation: coordinated changes in vergence, lens shape and pupil size
- Convergence: movement of both eyes to center image is in the retina
- Myosis: constriction (squeezing) of the pupil

CLASSIC PICTORIAL CUES

- Occlusions
- Linear perspective convergence
- Relative size, familiar size
- Texture gradient
- Shadows
- Shading
- Defocus blur
- Atmospheric perspective

An object blocks another

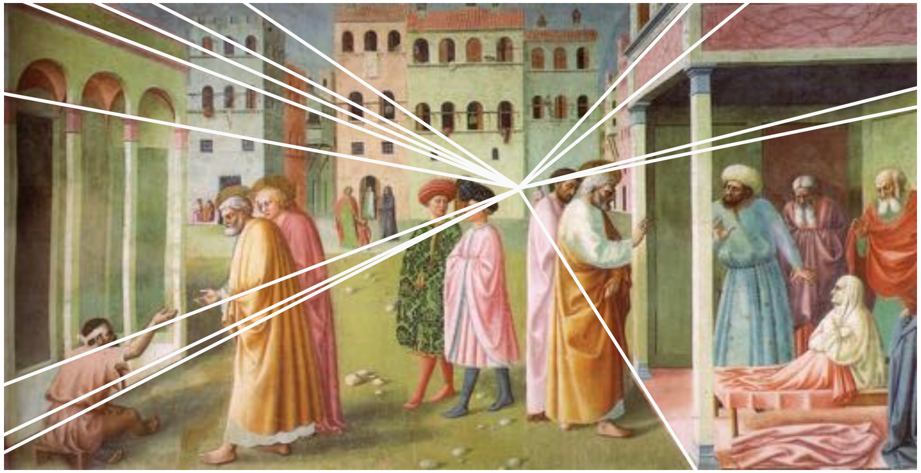


Perdreau F, Cavanagh P. Do artists see their retinas?. Frontiers in human neuroscience. 2011

CLASSIC PICTORIAL CUES

- Occlusions
- Linear perspective convergence
- Relative size, familiar size
- Texture gradient
- Shadows
- Shading
- Defocus blur
- Atmospheric perspective

Parallel lines appear to converge at some point in the distance



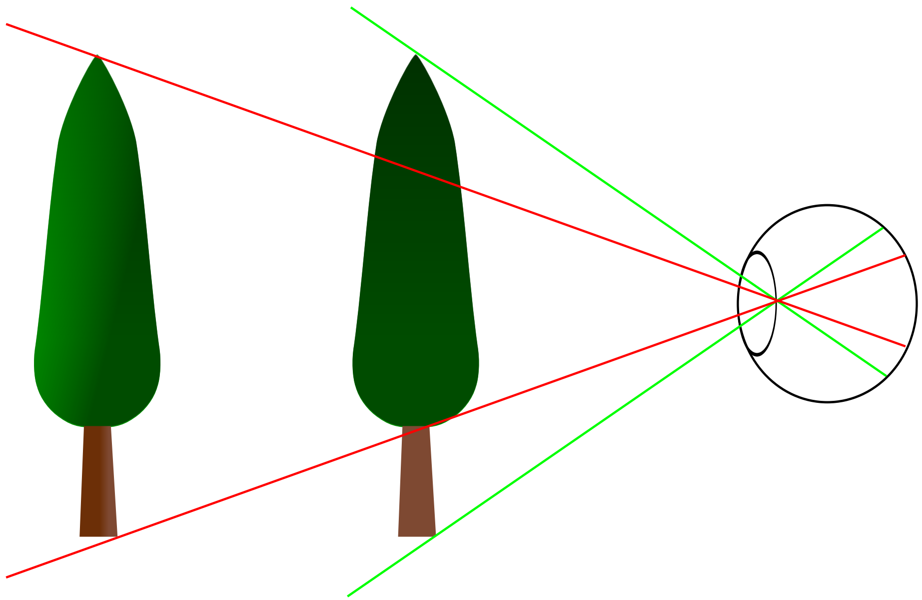
St. Peter Healing a Cripple and the Raising of Tabitha, Masolino, 1425

CLASSIC PICTORIAL CUES

- Occlusions
- Linear perspective convergence
- Relative size, familiar size
- Texture gradient
- Shadows
- Shading
- Defocus blur
- Atmospheric perspective

Retinal projection is:

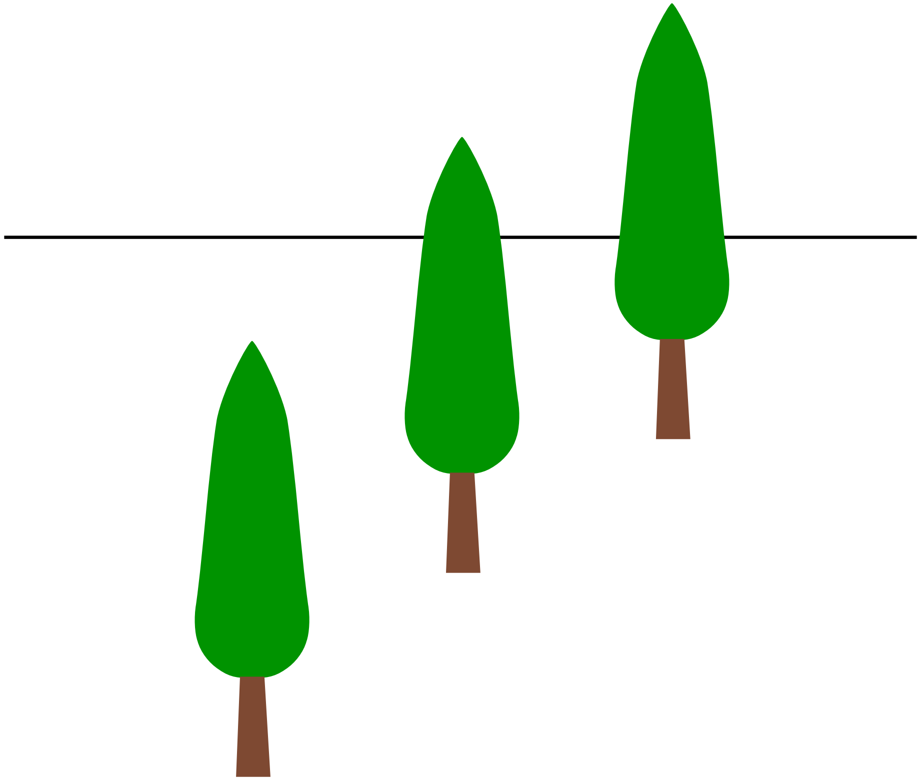
- Proportional to object size
- Inversely proportional to the object distance



CLASSIC PICTORIAL CUES

- Occlusions
- Linear perspective convergence
- Relative size, familiar size
- Texture gradient
- Shadows
- Shading
- Defocus blur
- Atmospheric perspective

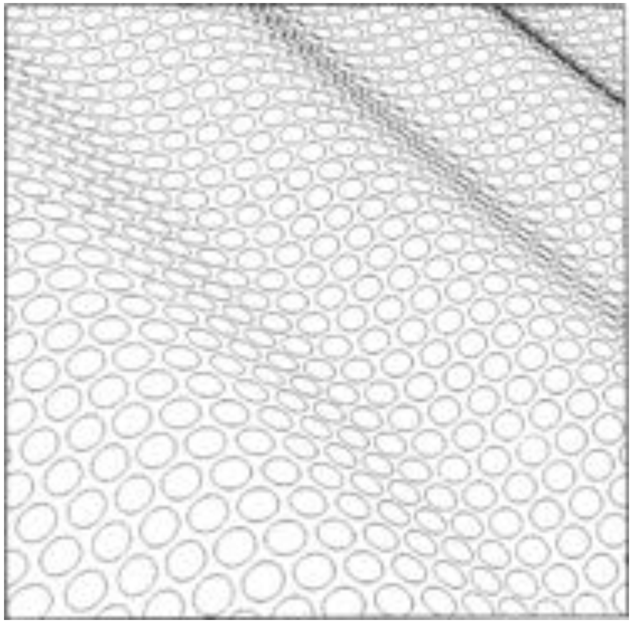
Object closer to the horizon are farther and up



CLASSIC PICTORIAL CUES

- Occlusions
- Linear perspective convergence
- Relative size, familiar size
- Texture gradient
- Shadows
- Shading
- Defocus blur
- Atmospheric perspective

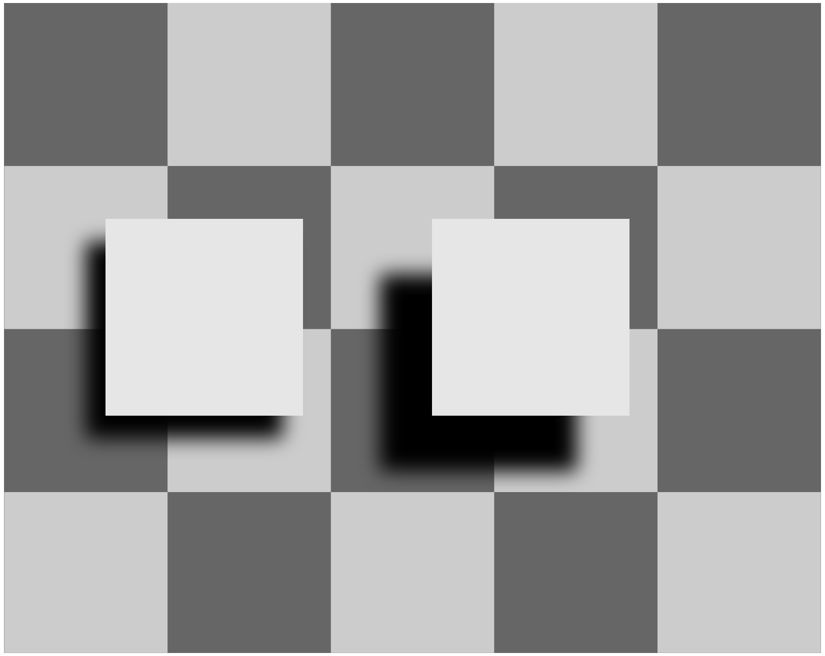
Density, perspective (foreshortening) and distortion of texture elements between closer and farther away objects



CLASSIC PICTORIAL CUES

- Occlusions
- Linear perspective convergence
- Relative size, familiar size
- Texture gradient
- Shadows
- Shading
- Defocus blur
- Atmospheric perspective

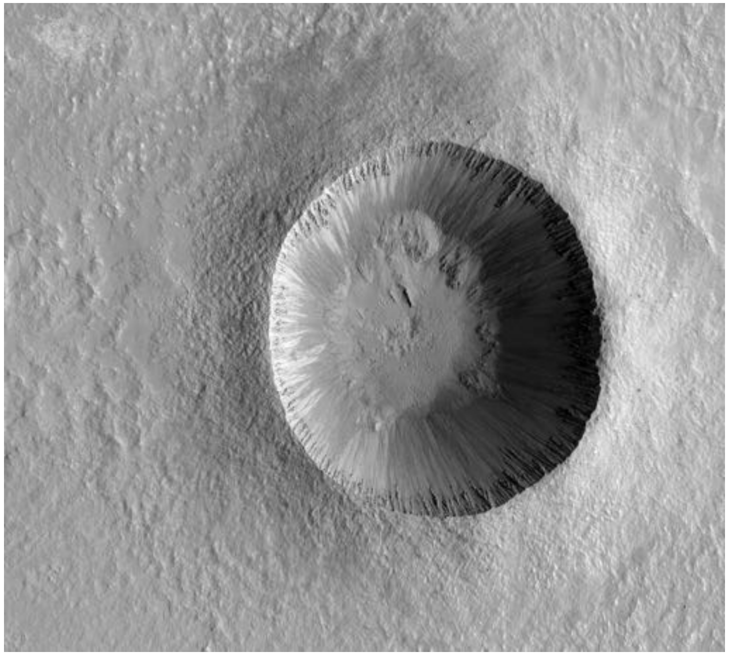
Dark area where light from a light source is blocked by an opaque object



CLASSIC PICTORIAL CUES

- Occlusions
- Linear perspective convergence
- Relative size, familiar size
- Texture gradient
- Shadows
- Shading
- Defocus blur
- Atmospheric perspective

Depicting depth in 3D models or illustrations by varying levels of darkness



Zumba Crater: Fresh 3-Km Crater, Mars. 04 May 2007, NASA/JPL University of Arizona

CLASSIC PICTORIAL CUES

- Occlusions
- Linear perspective convergence
- Relative size, familiar size
- Texture gradient
- Shadows
- Shading
- Defocus blur
- Atmospheric perspective

Aberration in which an image is out of focus

CLASSIC PICTORIAL CUES

- Occlusions
- Linear perspective convergence
- Relative size, familiar size
- Texture gradient
- Shadows
- Shading
- Defocus blur
- Atmospheric perspective

Farther objects:

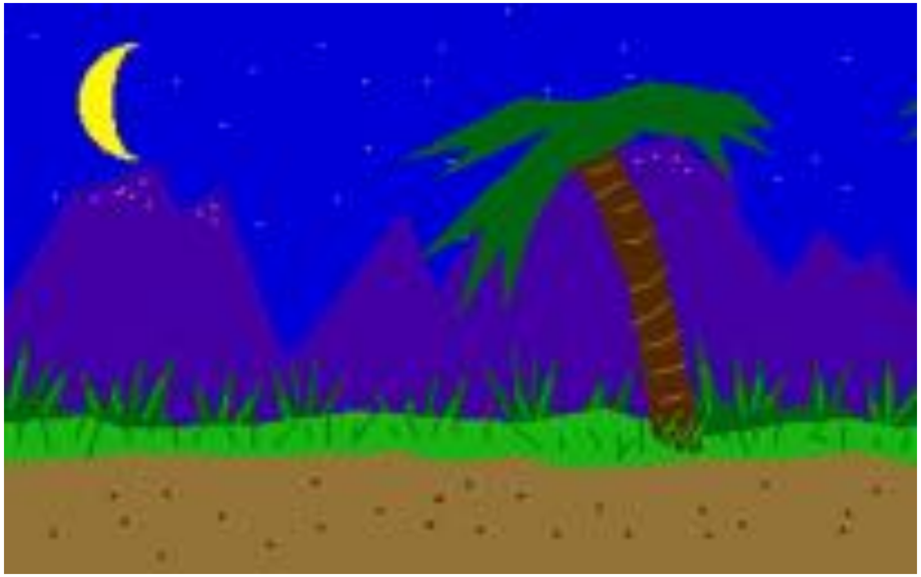
- Less distinct
- Colors less saturated
- Mountains in the back more blue

Paul Cézanne, Mont St. Victoire c. 1887

MOTION-BASED CUES

- Parallax
- Occlusion in motion
- Structure from motion

Closer objects move faster than farther away objects



"Parallax scrolling" uses layers of animated sprites to create the illusion of depth

MOTION-BASED CUES

- Parallax
- Occlusion in motion
- Structure from motion

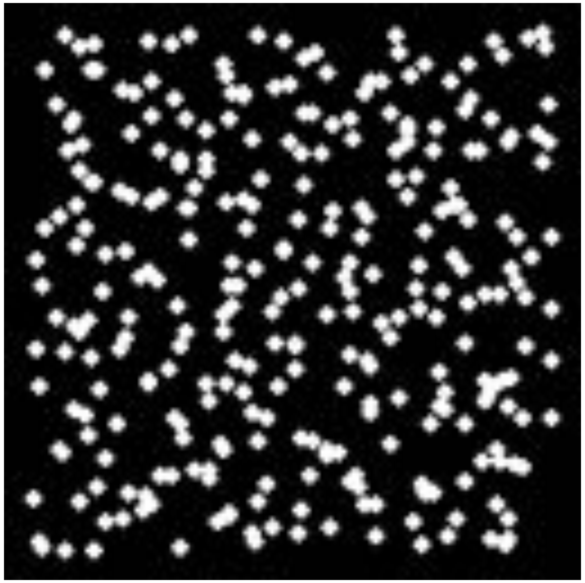
With deletion and accretion a moving object appears farther

Deletion: object becomes occluded, Accretion: object reappears

MOTION-BASED CUES

- Parallax
- Occlusion in motion
- Structure from motion

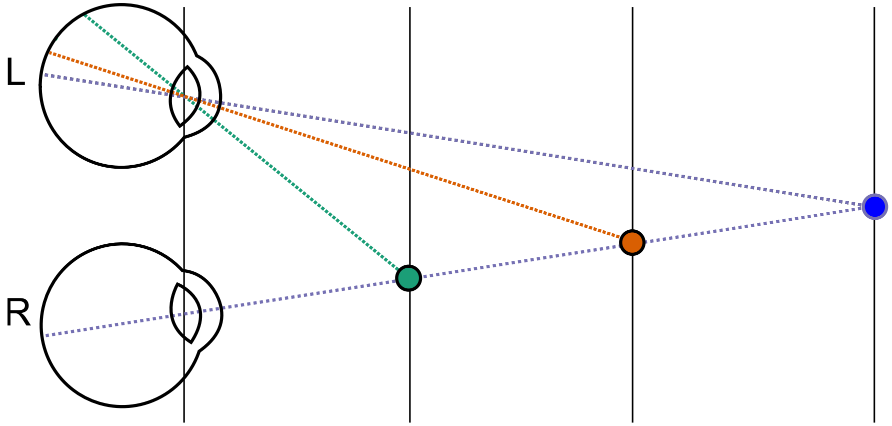
Depth cues from different points of view or moving object



This is an animation of a 3-d shape rocking back and forth, thus cued by relative motion. The dots that carry the motion flicker occasionally so as to eliminate the possible cue of changing local dot density. Despite the elimination of that cue and the flicker, it is relatively easy to perceive (and to judge) the 3-d shape. [M. Landy]

BINOCULAR CUES: STEREOPSIS

Depth cues from both eyes



Biological evidence: *"binocular cells"* [Hubel & Wiesel 1952, 1969], *"disparity detectors"* [Pettigrew 1967]

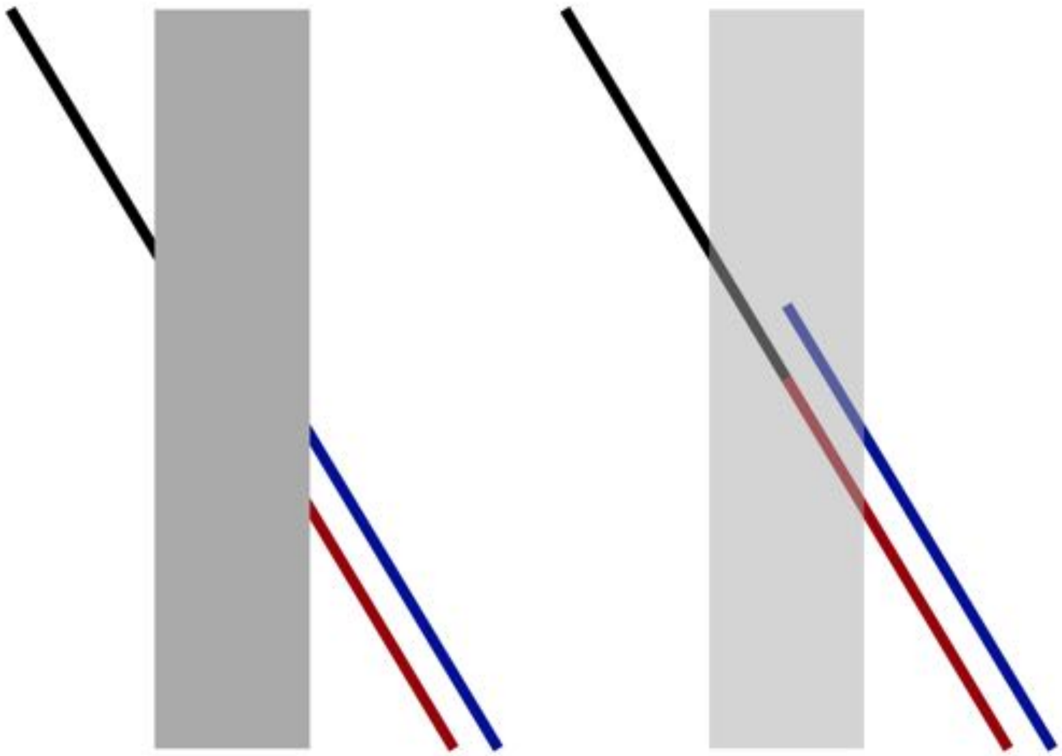
SOME (MORE) *ILLUSIONS*

SIZE CONSTANCY (PONZO) ILLUSION

Psychologist Mario Ponzo suggested that the human mind judges an object's size based on its background

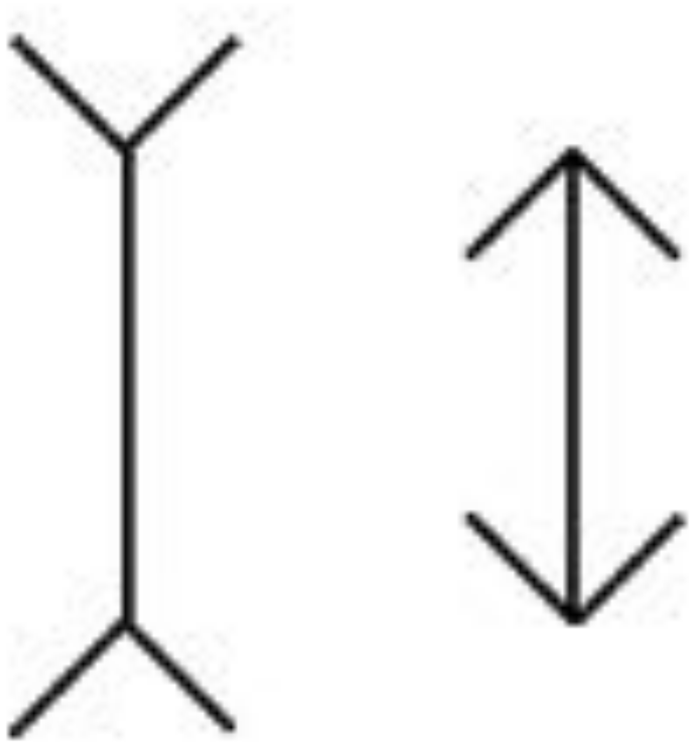
Ponzo Illusion 1911

POGGENDORFF ILLUSION

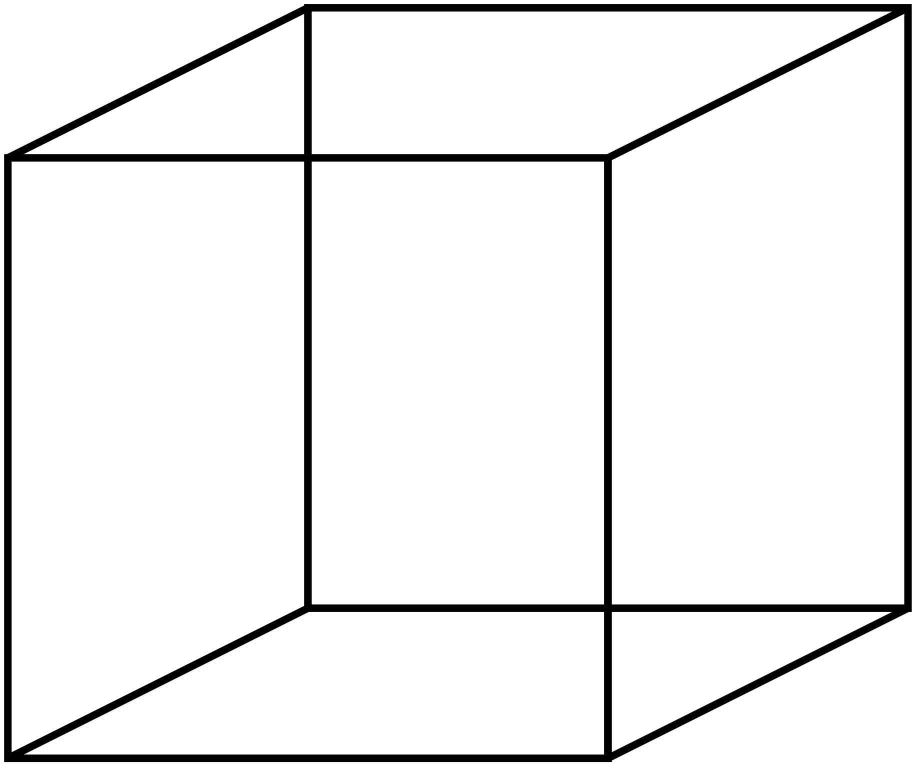


MULLER-LYER ILLUSION

Created by German psychiatrist Franz Muller-Lyer in 1889
One explanation is that opened arrows are perceived as farther away.

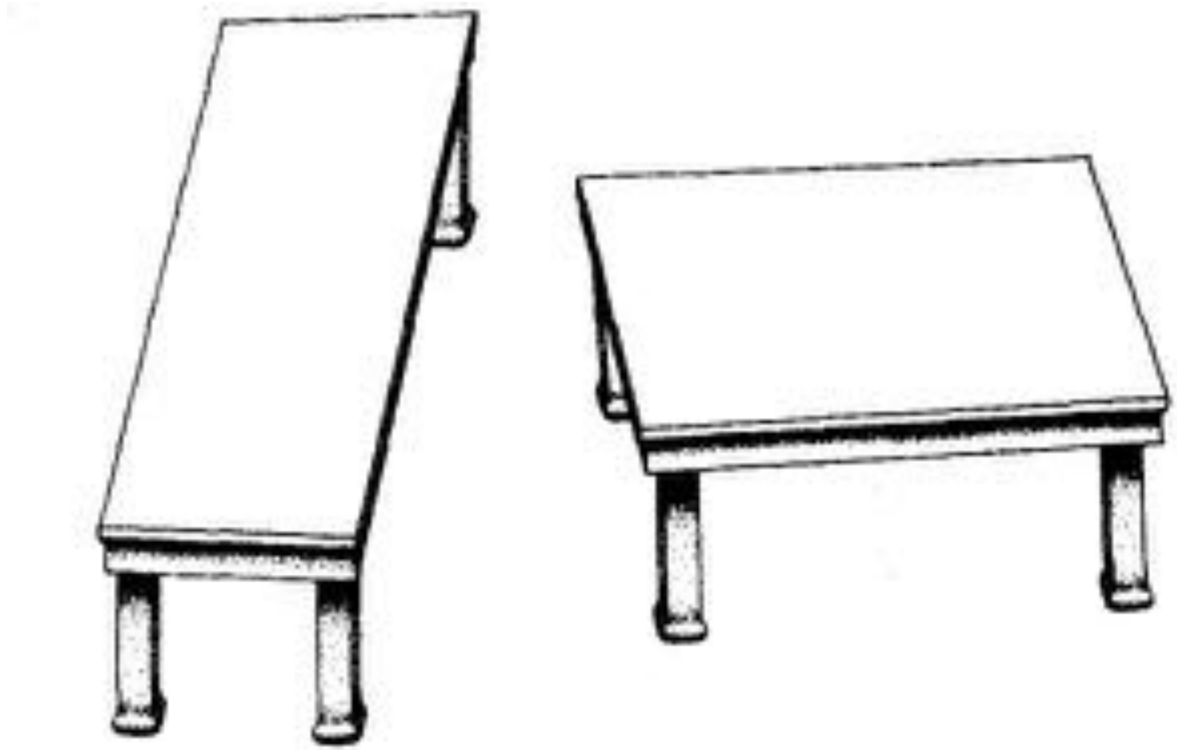


NECKER CUBE ILLUSION

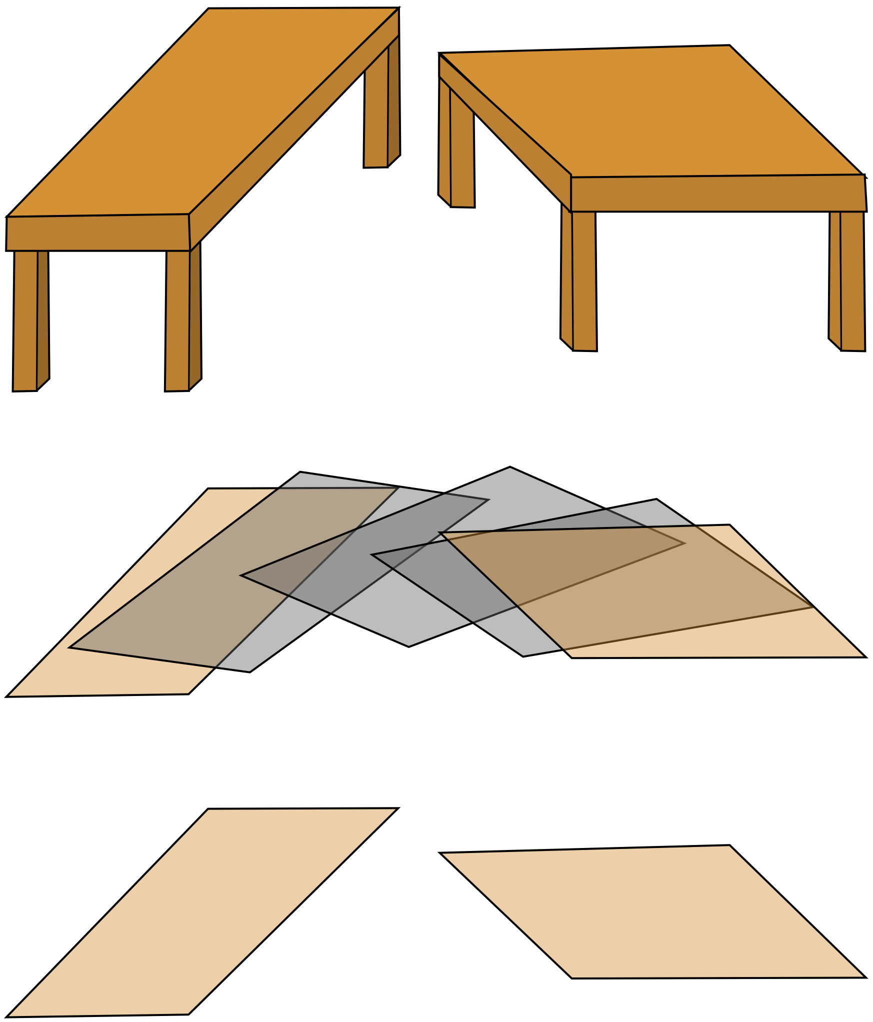


SHEPARD TURNING TABLES ILLUSION [SHEPARD 90]

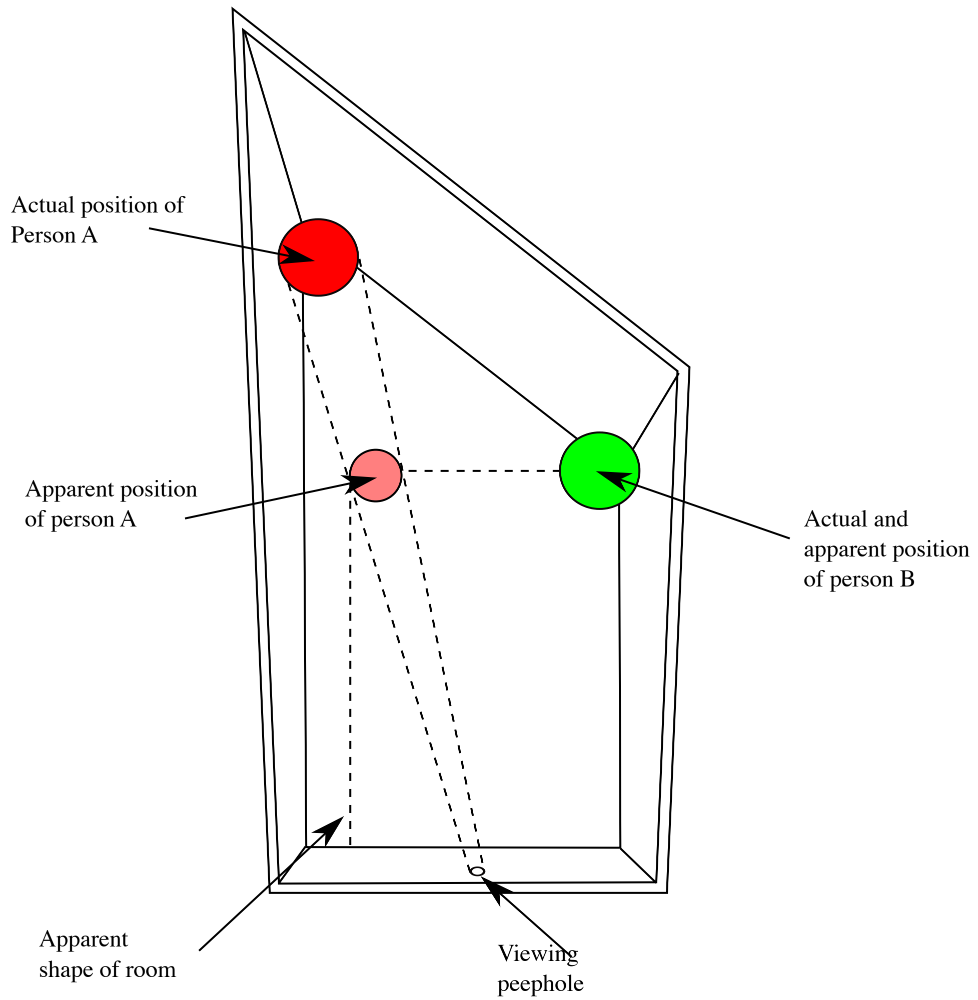
The visual system interprets 2D shape information in 3D



This illusion is based on a drawing of two parallelograms, identical aside from a rotation of 90 degrees. When the parallelograms are presented as tabletops, however, we see them as objects in three-dimensional space. Note that real tables have different geometry.



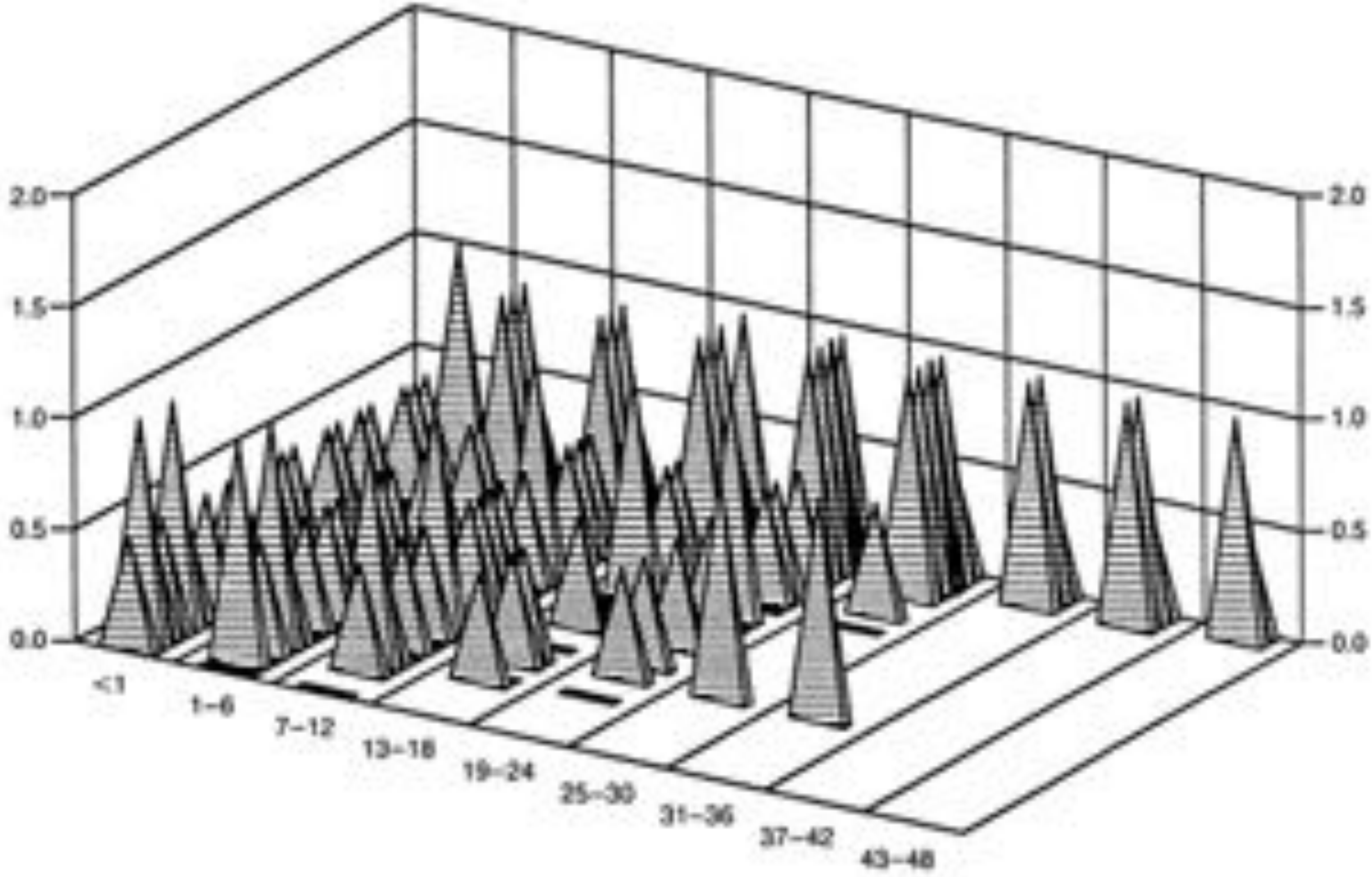
Ames room: what cues are at play?



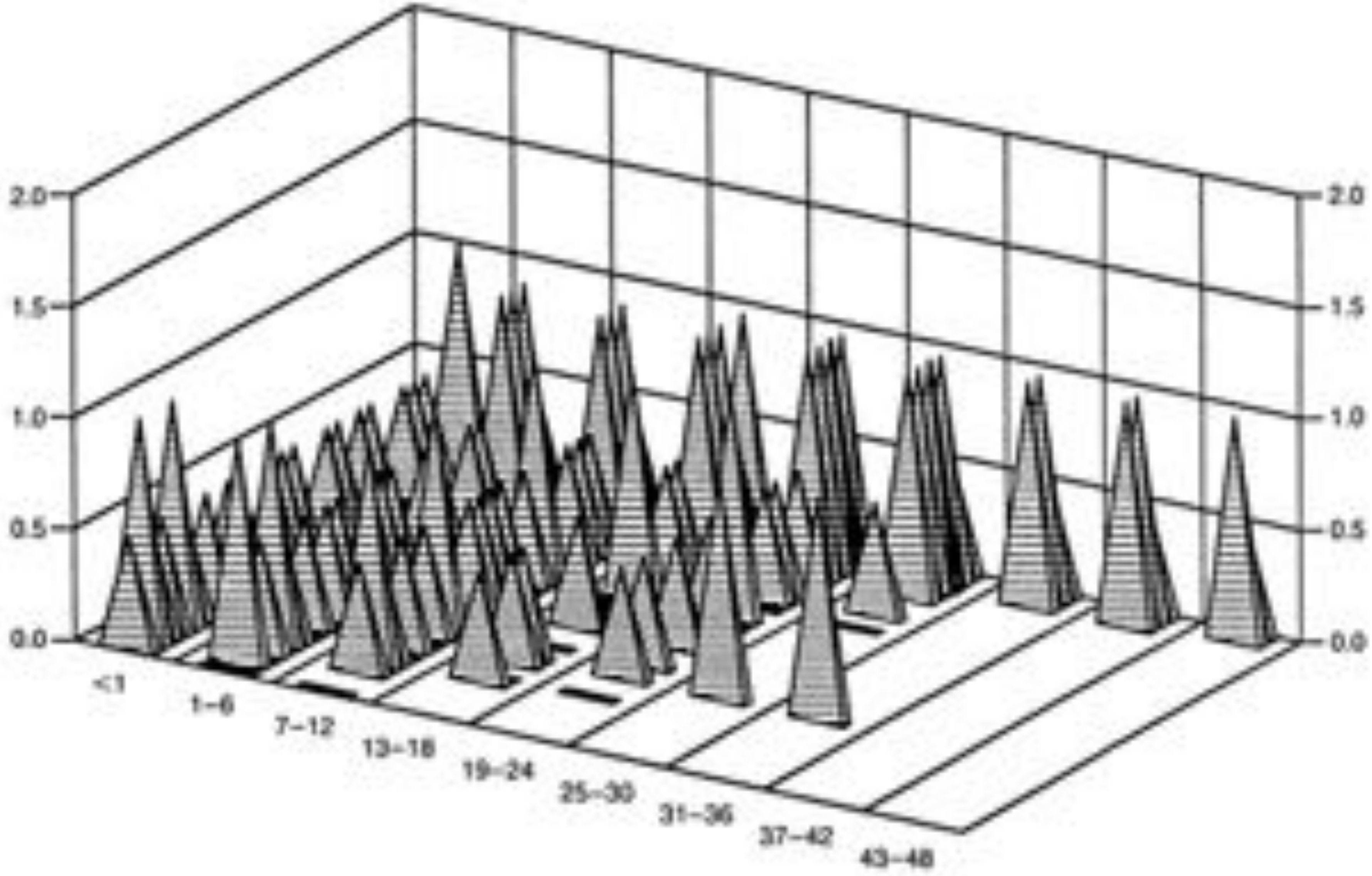
The illusion relies on the following depth cues: relative size, familiar size, perspective convergence

DESIGN CONSIDERATIONS

What is wrong with this chart?

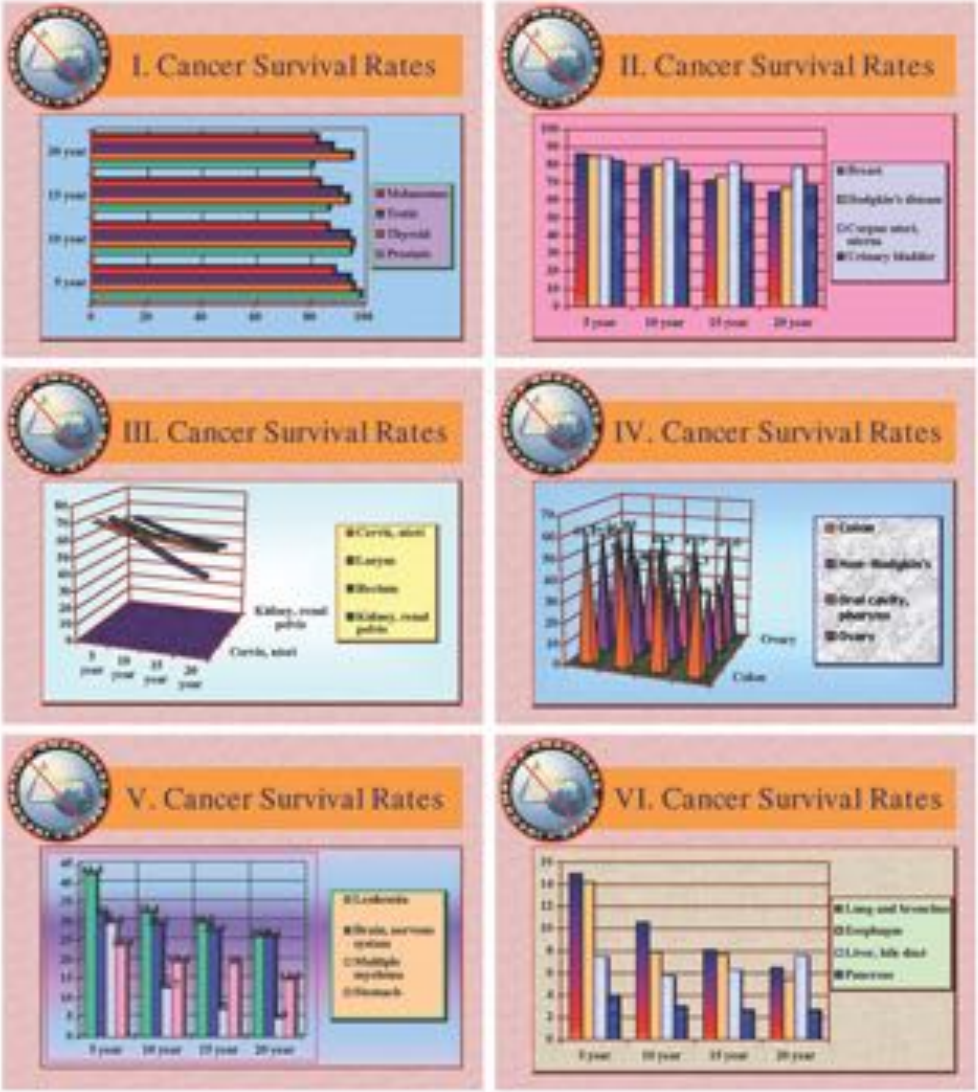


What is wrong with this chart?



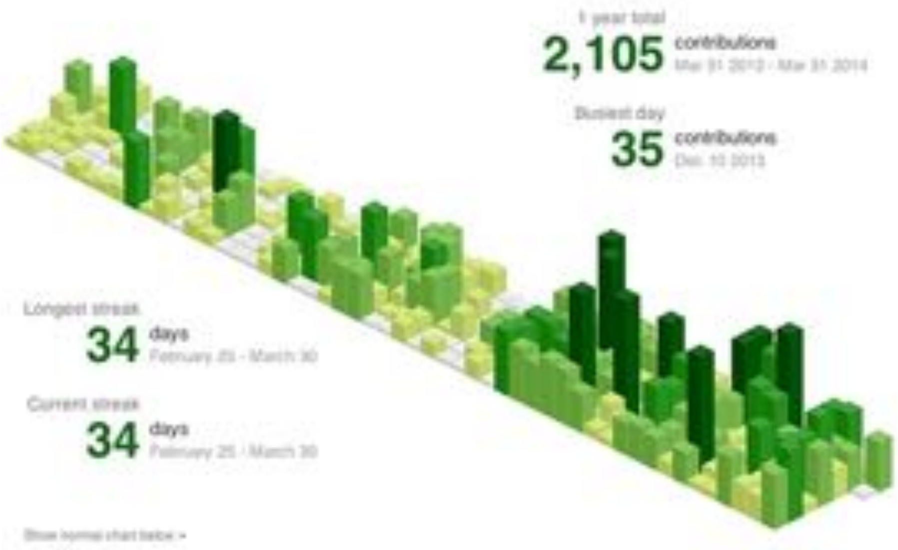
Occlusions, confusing texture, confusing shapes, Necker illusion, projection

What is wrong with these charts?

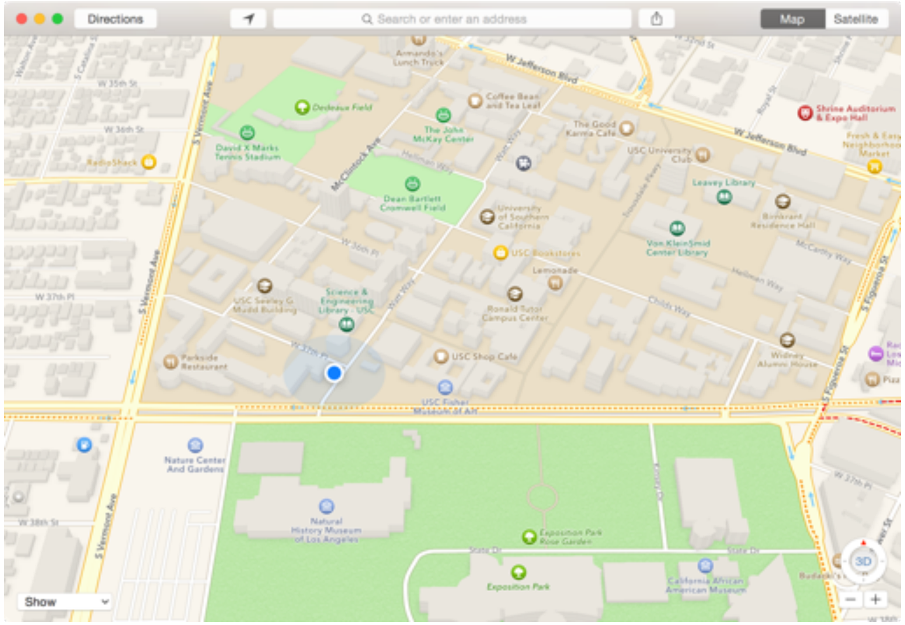


Cancer survival rates: tables, slopegraphs, barcharts, E. Tufte

2.5D DESIGN



Isometric projection use to display bars. An isometric projection is an orthographic projection with coordinate axes appearing equally foreshortened and at 120 degrees from each other



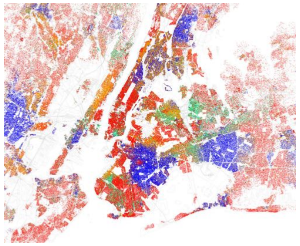
2.5D design with constrained navigation used to alleviate problems with navigating a 3D environment

OUTLINE

- Depth perception and design considerations
- Introduction to maps
- How maps are built?
- Maps projections and their uses
- Working with maps

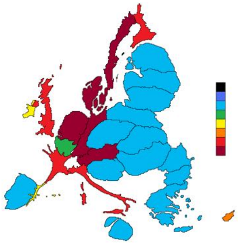
THEMATIC MAPS

Dot Map



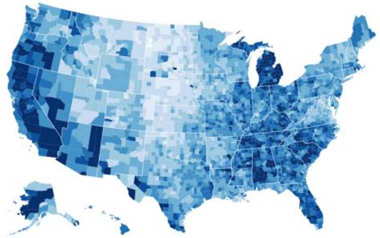
Can be used to locate each occurrence of a phenomenon. One-to-one or one-to-many.

Cartogram



Area used to display value. Distortion used to show continuous variables

Choropleth

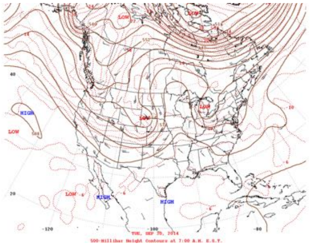


Areas are shaded or patterned in proportion to variable

Proportional Symbol Map

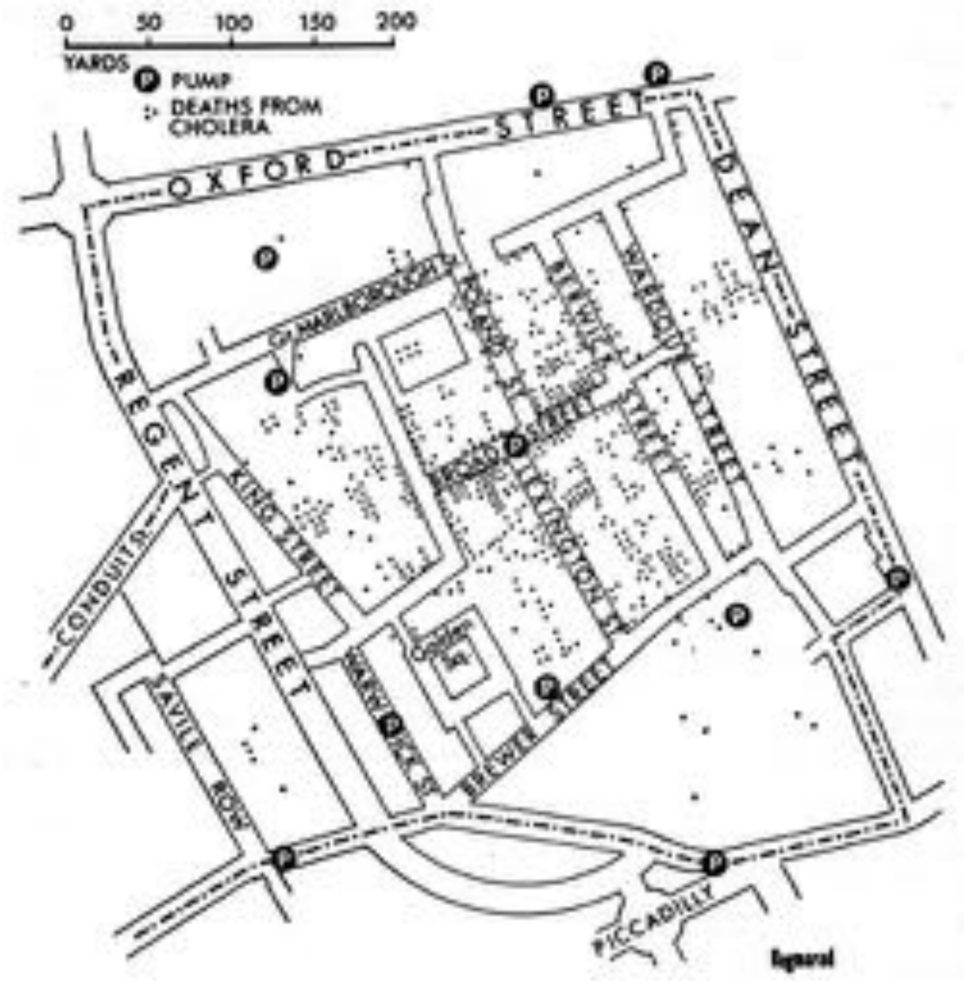
Scaled symbols show data for areas/locations. Also called Graduated Symbol Map.

Isopleth



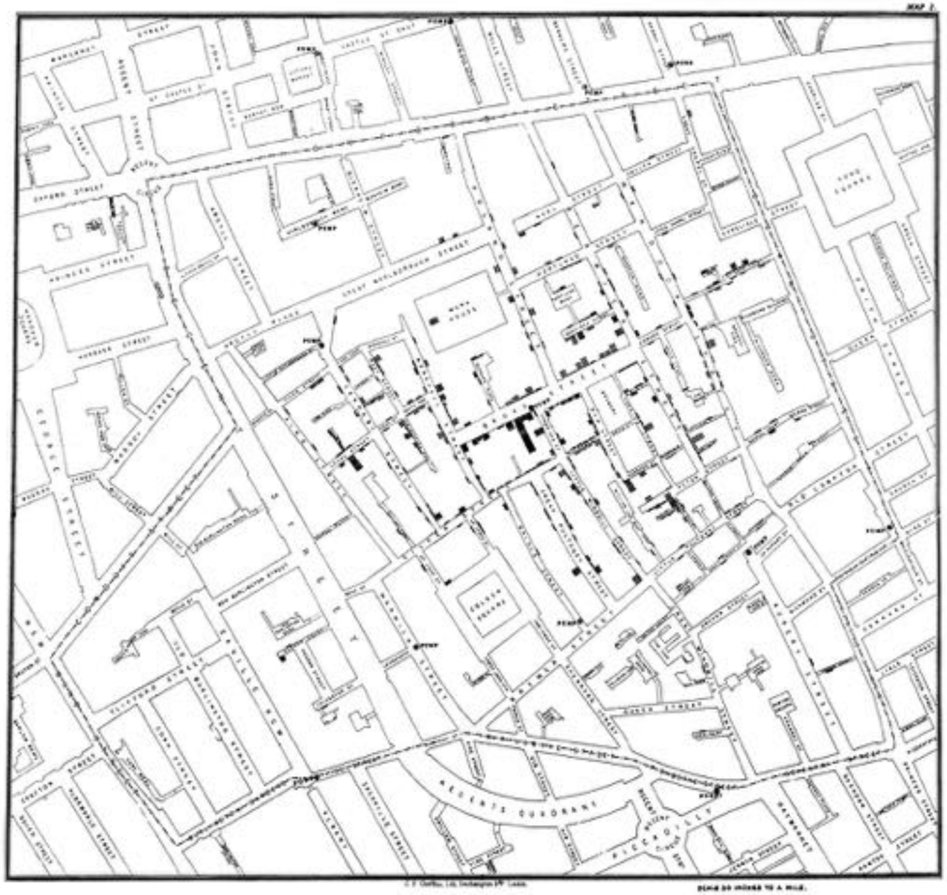
Use contours to show continuous variables. Also called Isarithmic.

JOHN SNOW'S LONDON CHOLERA MAP



Update of John Snow's London cholera epidemic [Regmarad 1960]

ORIGINAL JOHN SNOW MAP



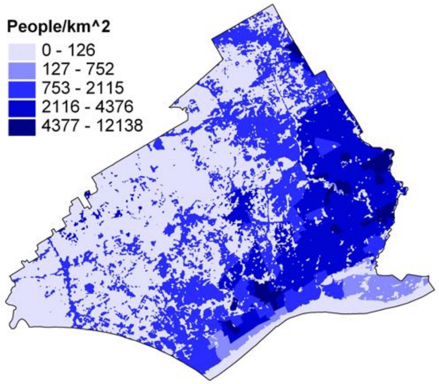
Original map by John Snow showing the clusters of cholera cases in the London epidemic of 1854 [Snow 1854]

CARTOGRAM

Our World in data population growth cartogram. See also (topojson) Hello World Population Cartogram for a D3 rendering.

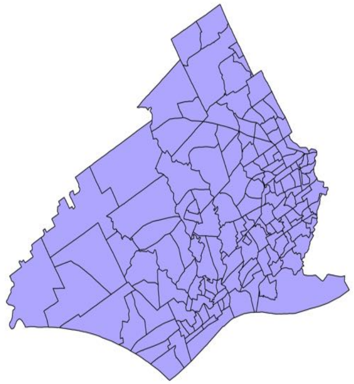
DASYMETRIC MAP

Refined choropleth map where ancillary information is used to model a phenomena [Semenov-Tyan-Shansky 1911]



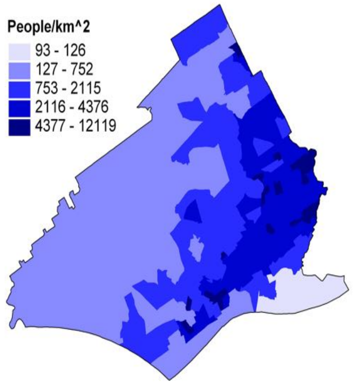
Dasymetric map of population

=



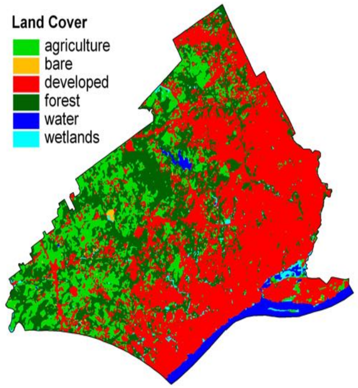
Map of tract data

+



Choropleth of population density

+



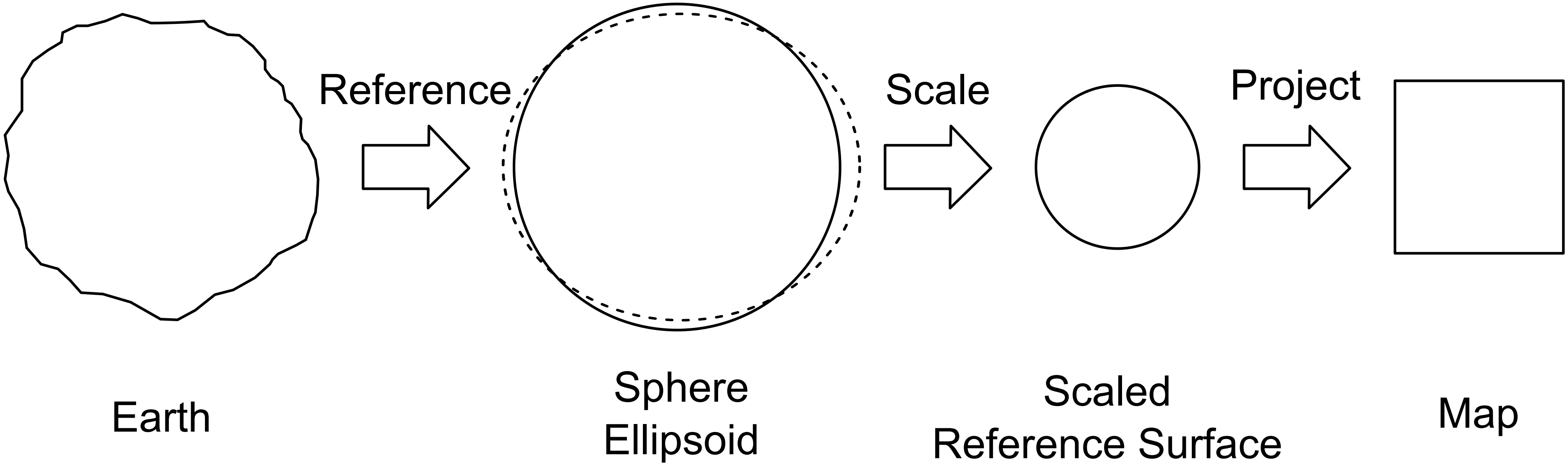
Land cover (ancillary data)

Mennis, J. Generating surface models of population using dasymetric mapping. The Professional Geographer, 2003.

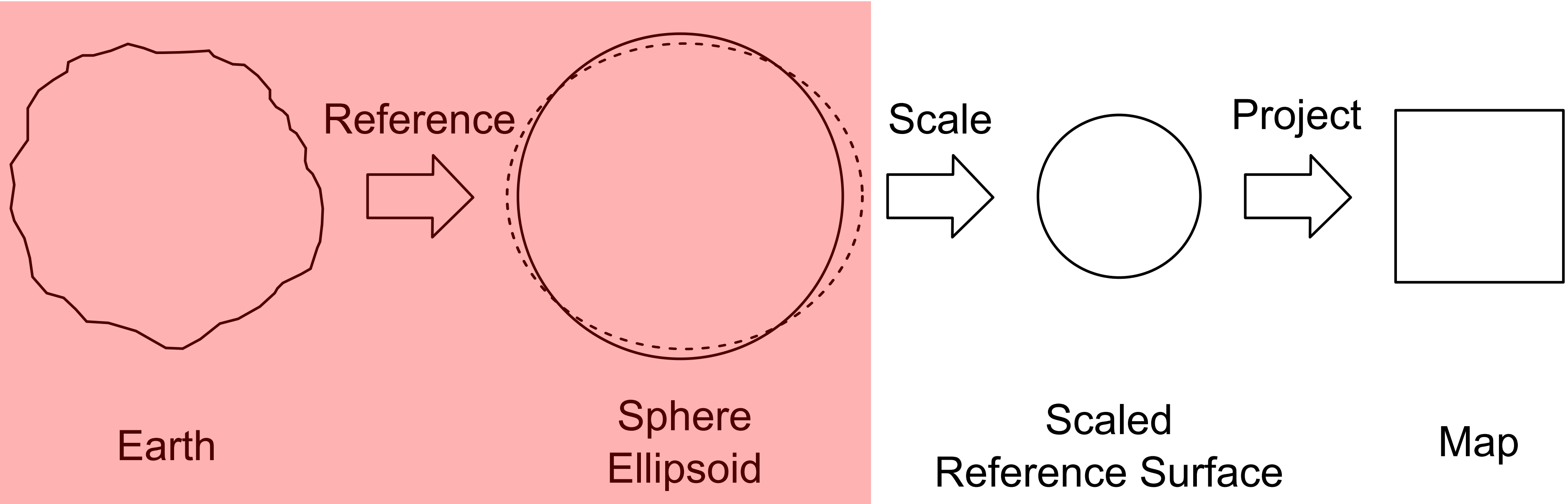
OUTLINE

- Depth perception and design considerations
- Introduction to maps
- How maps are built?
- Maps projections and their uses
- Working with maps

HOW MAPS ARE BUILT?

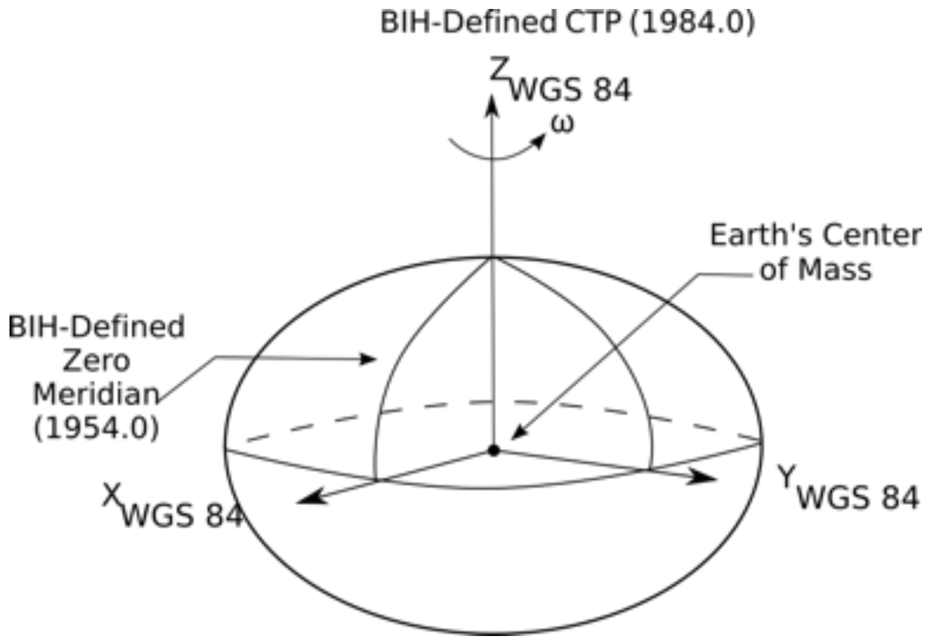


HOW MAPS ARE BUILT?

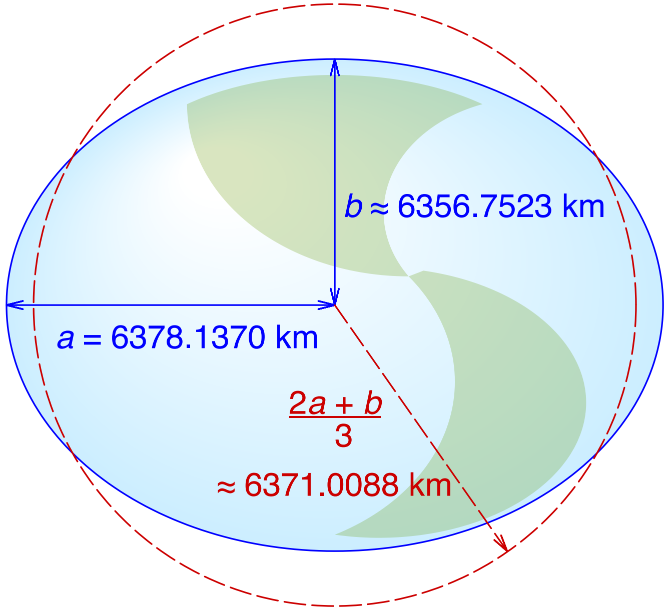


REFERENCE ELLIPSOID

World Geodetic System 1984 (WGS 84), used for GPS



WGS 84 ellipsoid



WGS 84 ellipsoid semi-major and semi-minor axes

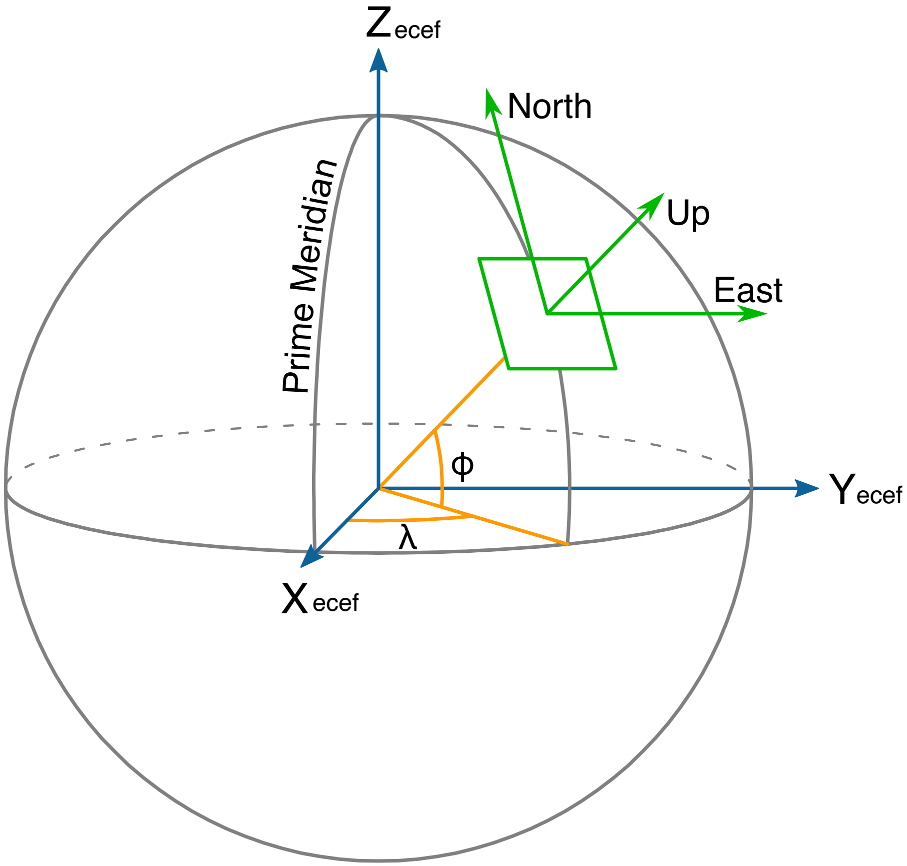
GEOGRAPHIC COORDINATES

Coordinates used to locate points on a reference ellipsoid

$(\phi, \lambda) = (\text{latitude, longitude})$ angles in degrees

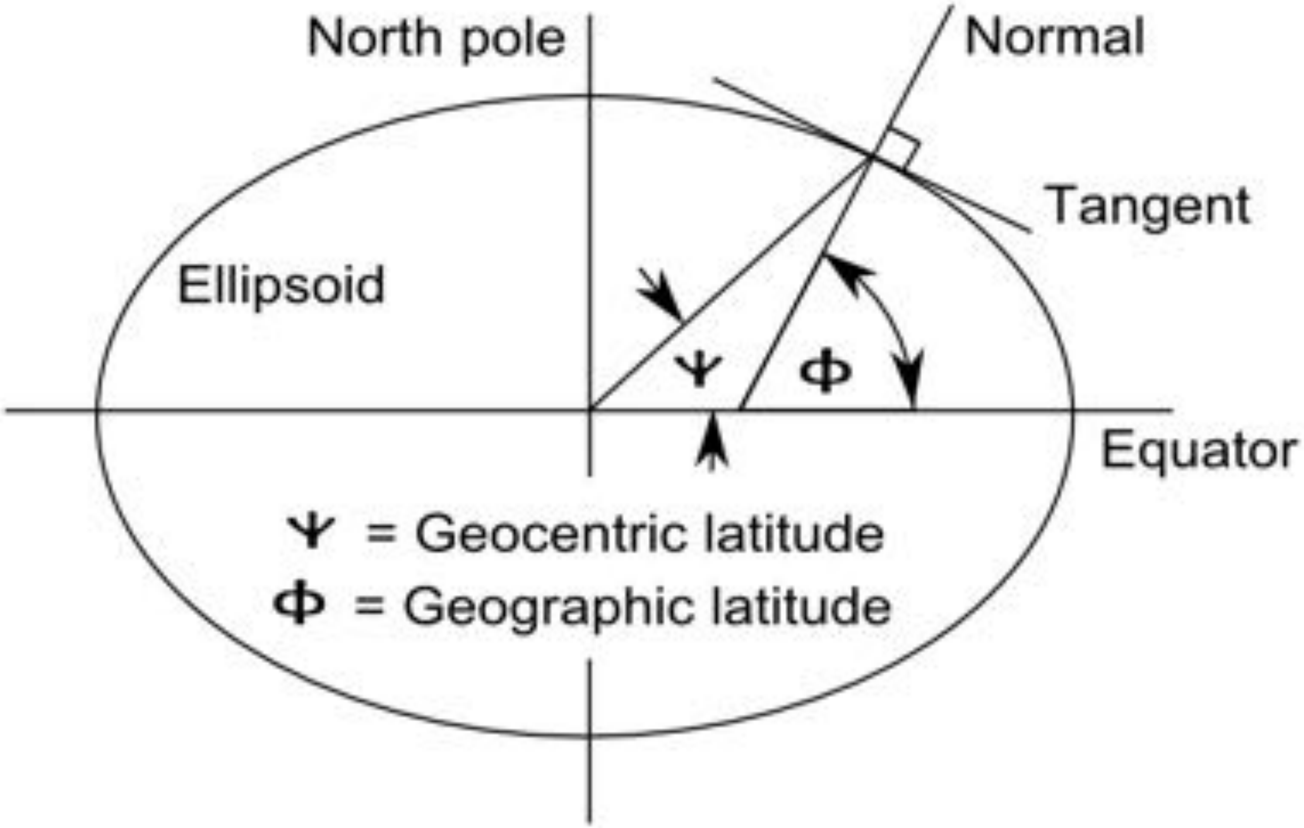
Example: Culver City

34.02232116570379, -118.39584935765946
34°01'21.0"N 118°23'44.1"W



GEOGRAPHIC VS. GEOCENTRIC LATITUDE

If not specified the geographic latitude is used



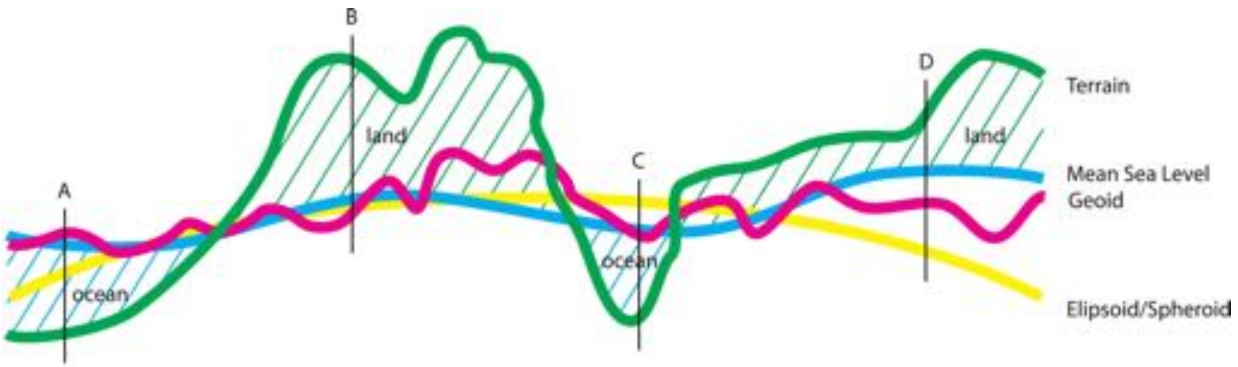
GEODETIC DATUM/SYSTEM

A coordinate system and reference to locate places defined by a horizontal and a vertical datum

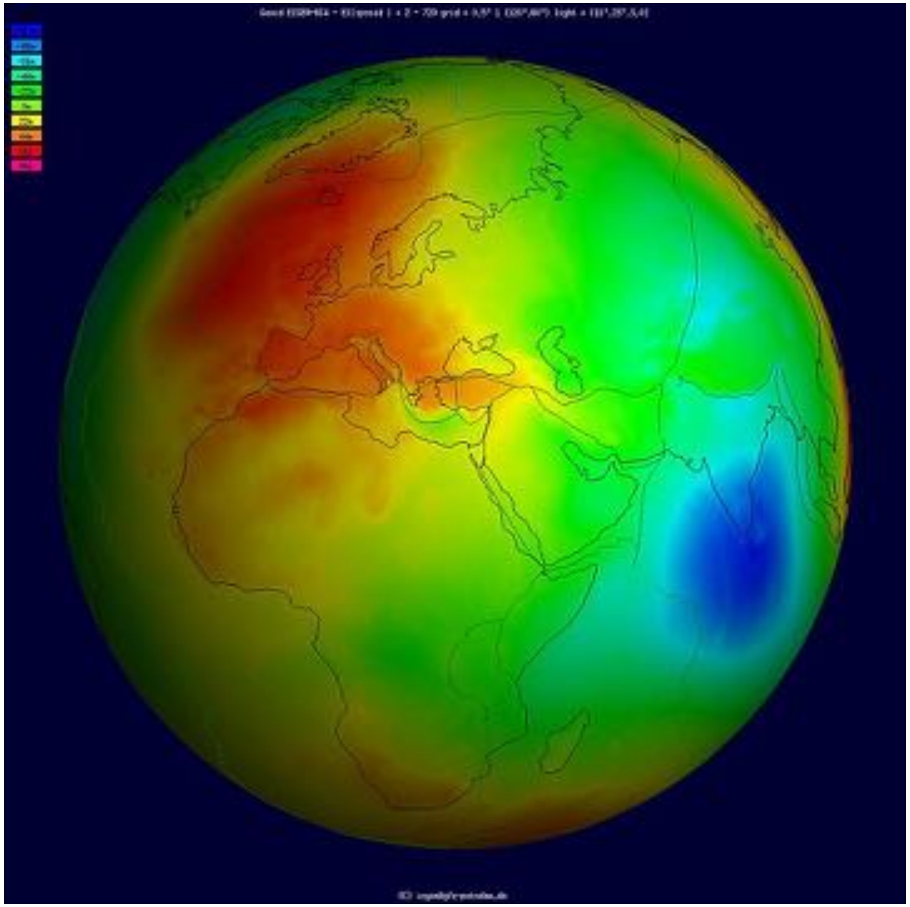
Horizontal datum is defined by a reference ellipsoid

Vertical datum is defined by how we measure elevation:

- *"Geodetic"* if based on ellipsoid of horizontal datum
- *"Tidal"* if based on sea levels
- *"Gravimetric"* if based on a geoid

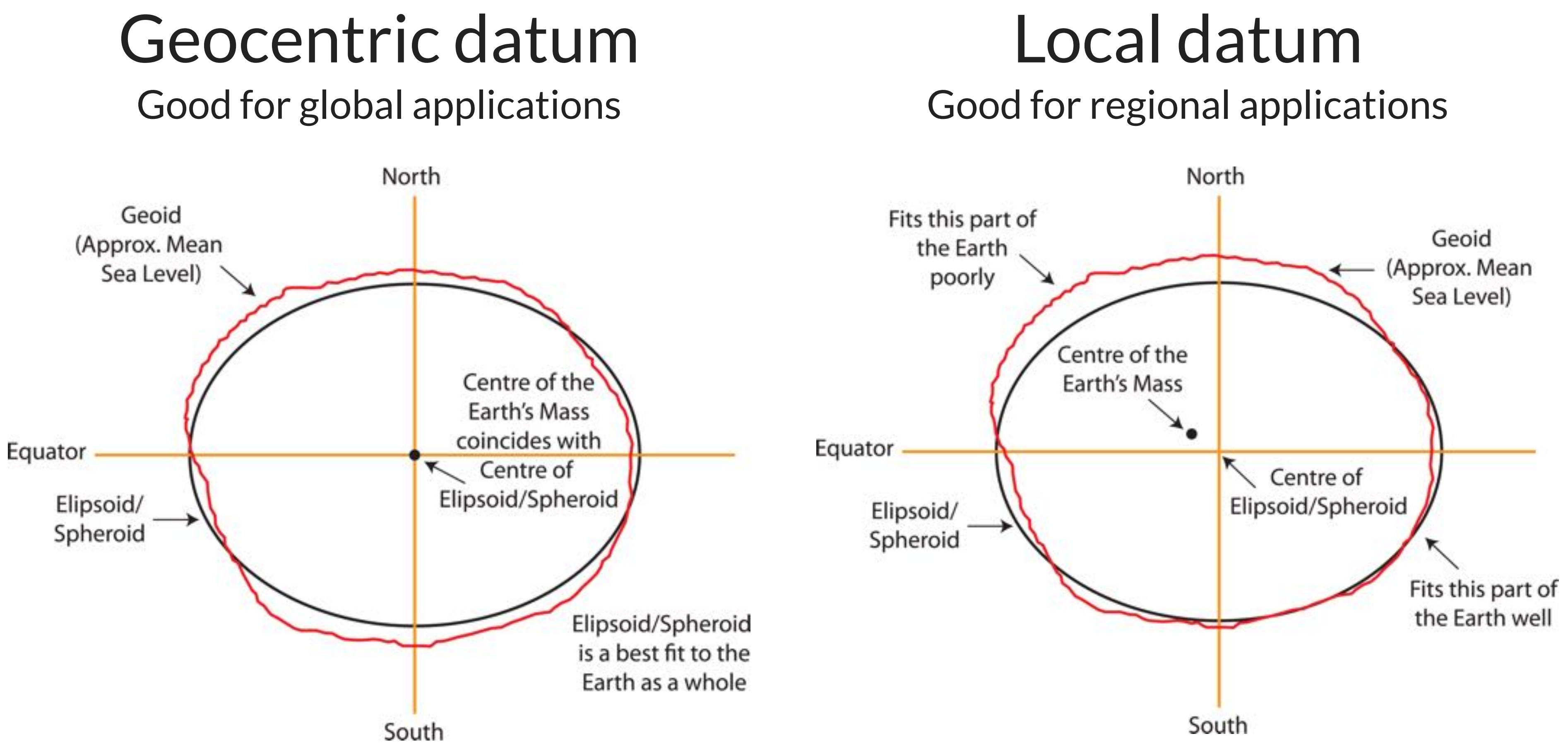


GEOID



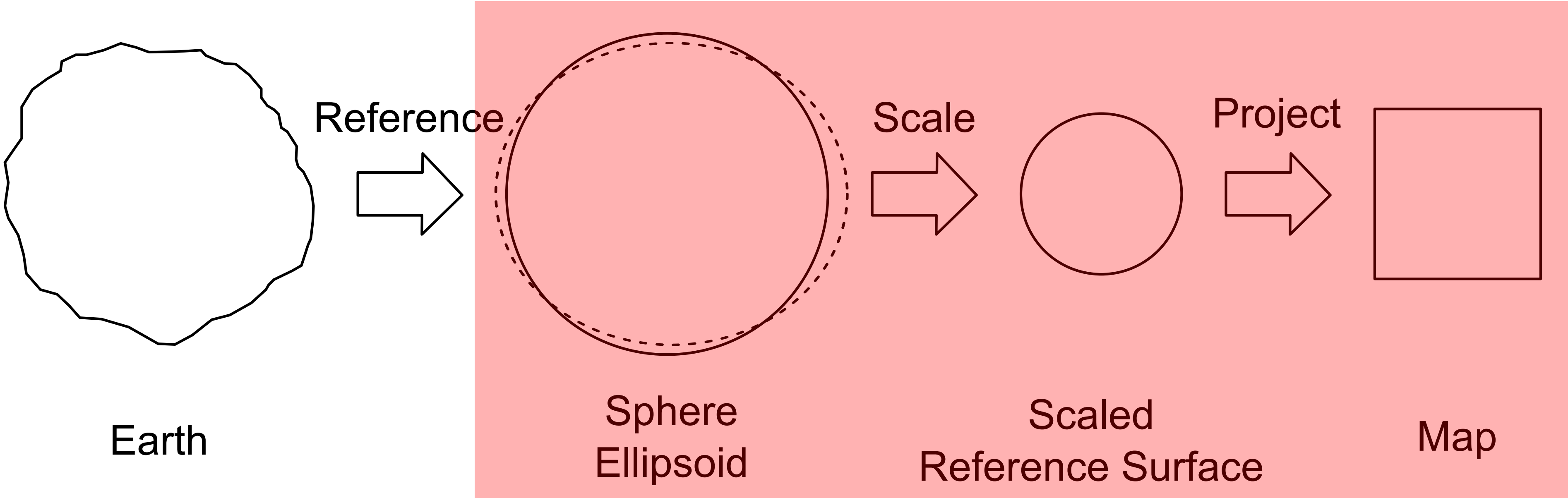
Geoid undulation in false color, to scale.
By International Centre for Global Earth Models (ICGEM)

GEOCENTRIC VS. LOCAL GEODETIC DATUM

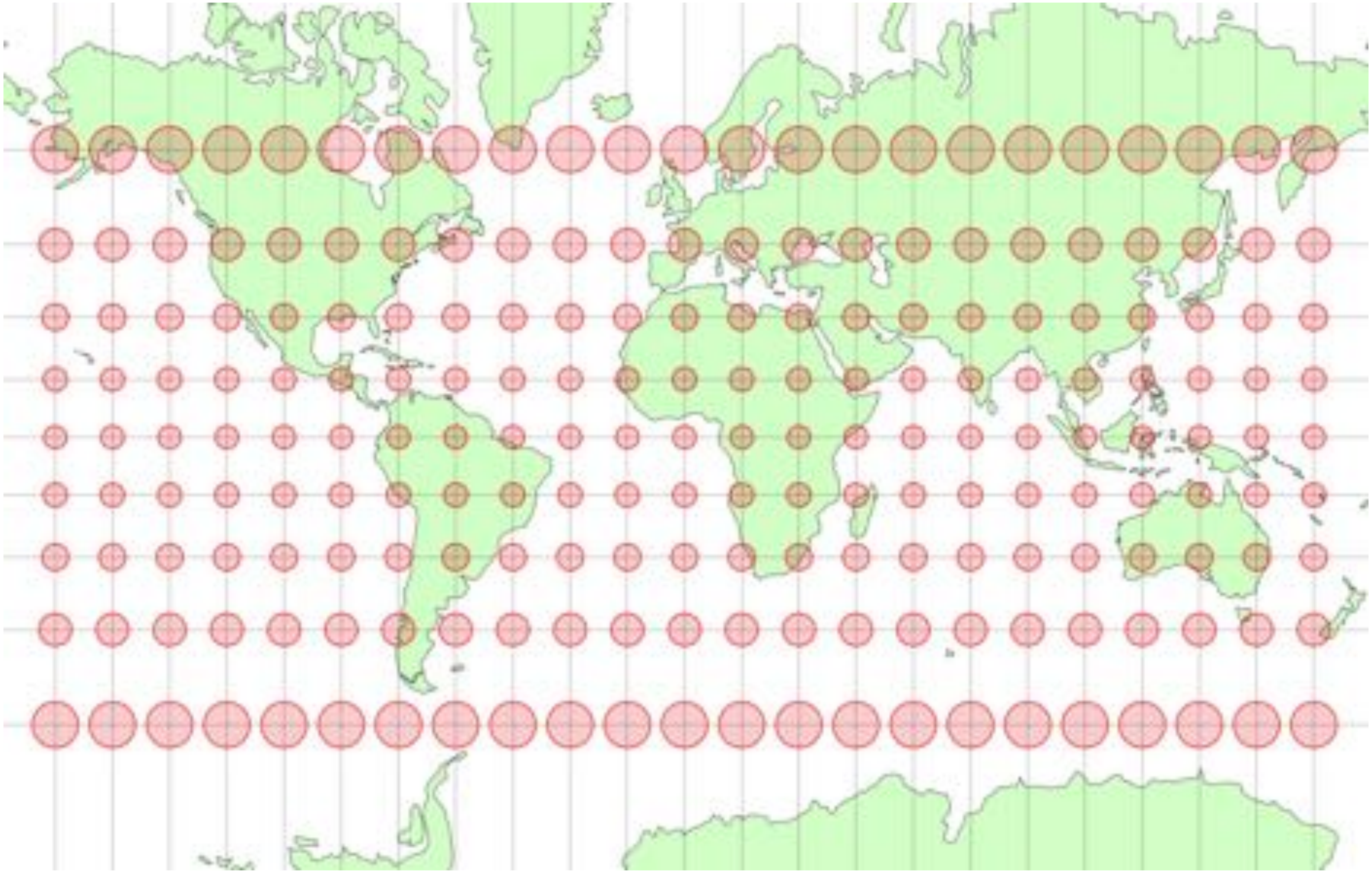


Geocentric and local datum is defined by the type of horizontal datum used

HOW MAPS ARE BUILT?



MAP PROJECTION

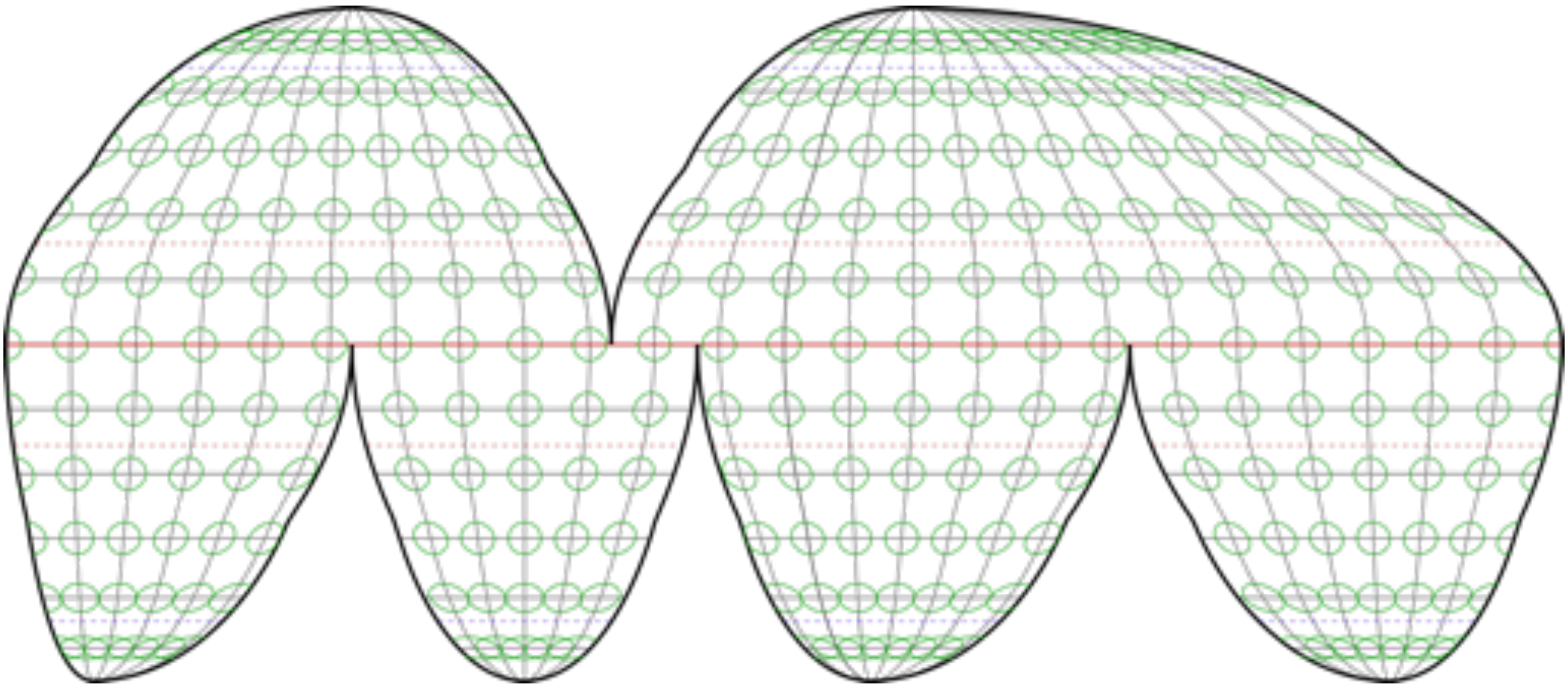


All projections introduce some distortion
[Tissot’s indicatrix by Jason Davies](#)

PROPERTIES PRESERVED IN MAPS

Name	Property preserved
Conformal	Shape of small regions. At any point same scale in all directions, 90^o between parallels and meridians, angles preserved at each point.
Equal-area	Areas proportional to areas on Earth
Equidistant	Scale along one or more lines, or from one or two points to all other points
Azimuthal (true direction)	Directions from a central point: great circles through the central point are straight lines

A MAP CANNOT BE BOTH CONFORMAL AND EQUAL-AREA



Cannot flatten an orange peal without tearing!

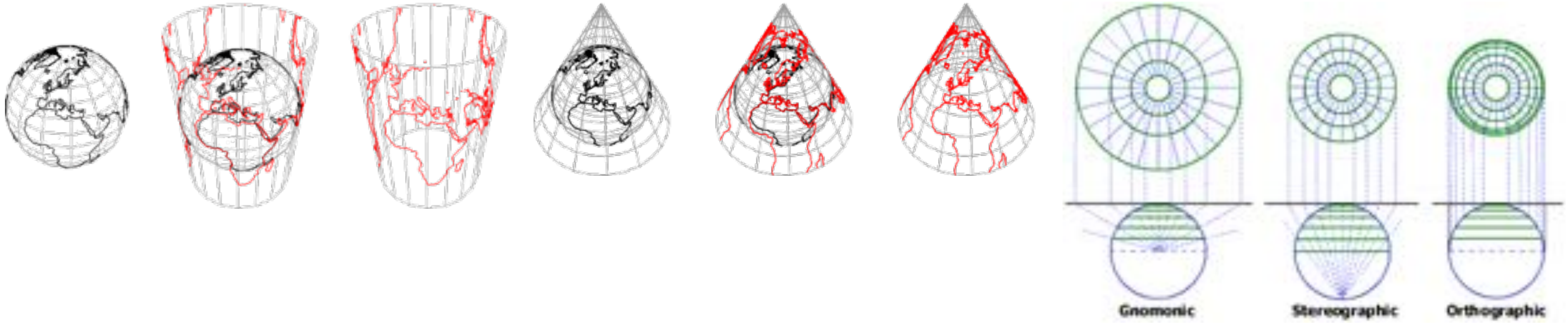
OUTLINE

- Depth perception and design considerations
- Introduction to maps
- How maps are built?
- Maps projections and their uses
- Working with maps

NAMED PROJECTIONS TYPES

Named after the developable surface used*

Cylindrical Conical Azimuthal



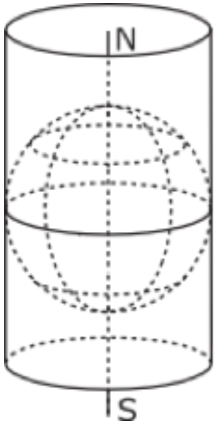
* Developable surface: surface that can be unfolded or unrolled into a plane or sheet without stretching, tearing or shrinking

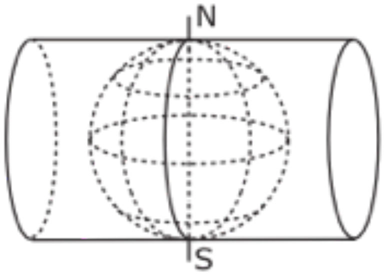
NAMED PROJECTIONS SUB-TYPES

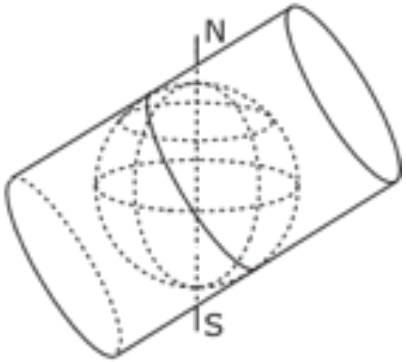
Named based on orientation and intersection

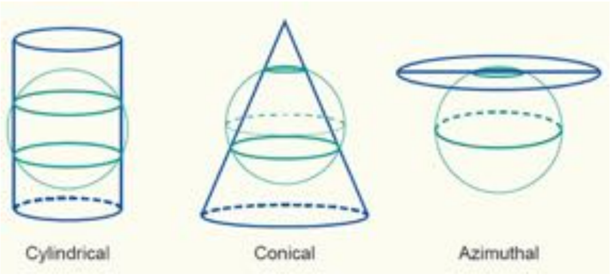
<u>Orientation</u> | <u>Intersection</u>

Normal | Transverse | Oblique | Tangent or Secant









LAMBERT AZIMUTHAL EQUAL AREA

PROJECTION: AZIMUTHAL, EQUAL-AREA

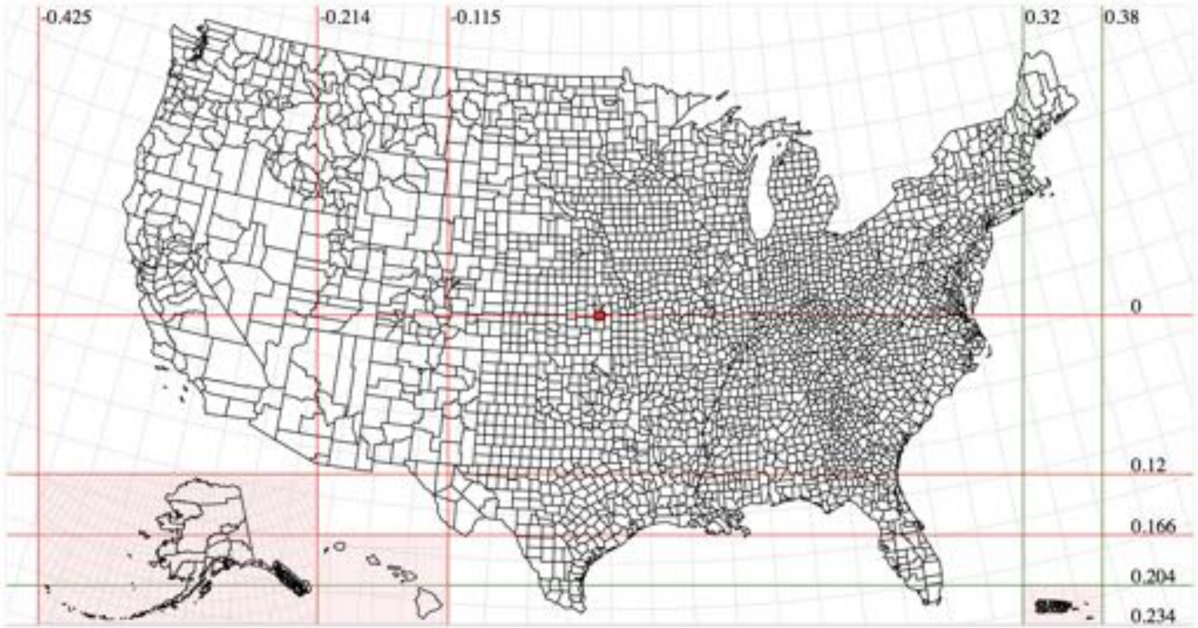
National Atlas of the United States, United States Department of the Interior (2002)

- Named after Johann H. Lambert
- **Useful for choropleth maps**

ALBERS CONIC

PROJECTION: CONIC, EQUAL AREA

- Named after Heinrich C. Albers
- Used by the United States Geological Survey and Census Bureau
- **Useful for choropleth maps**

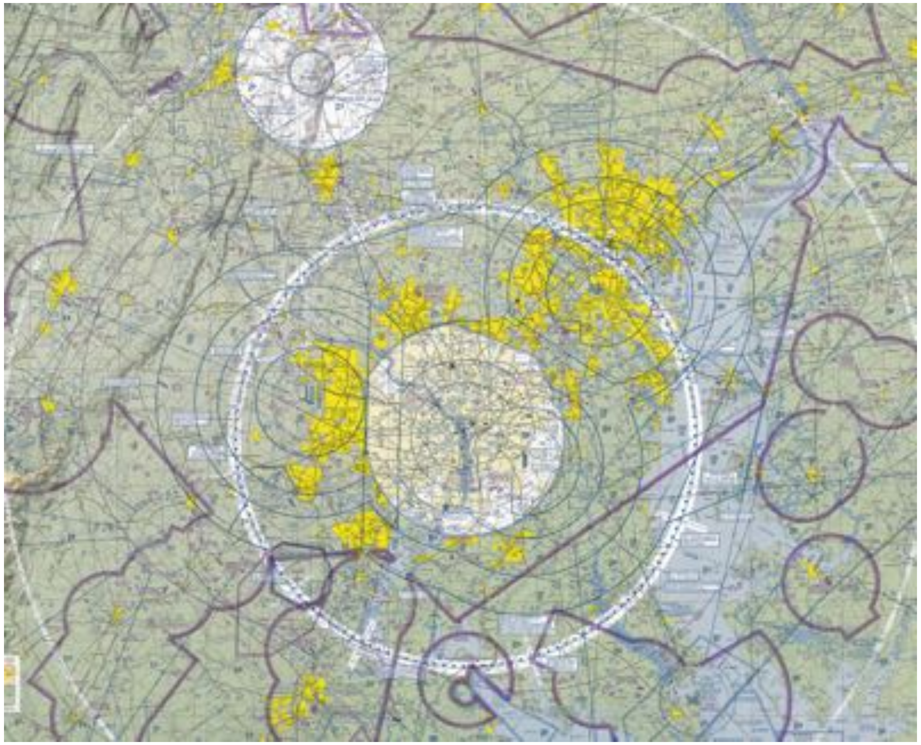


Albers USA U.S.-centric composite projection of 3 Albers equal-area conic projections. Alaska is at $0.35x$ of its true relative area.

LAMBERT CONFORMAL CONIC (LCC)

PROJECTION: CONIC, CONFORMAL

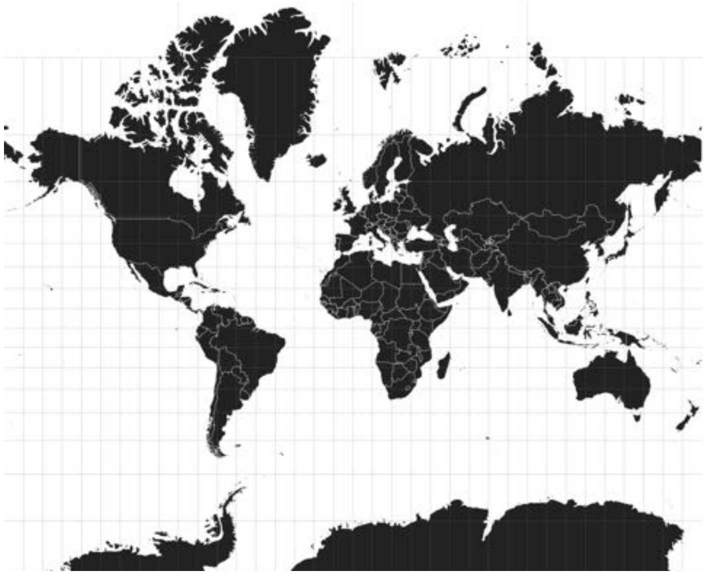
- Named after Johann H. Lambert
- **Useful for aeronautical charts**



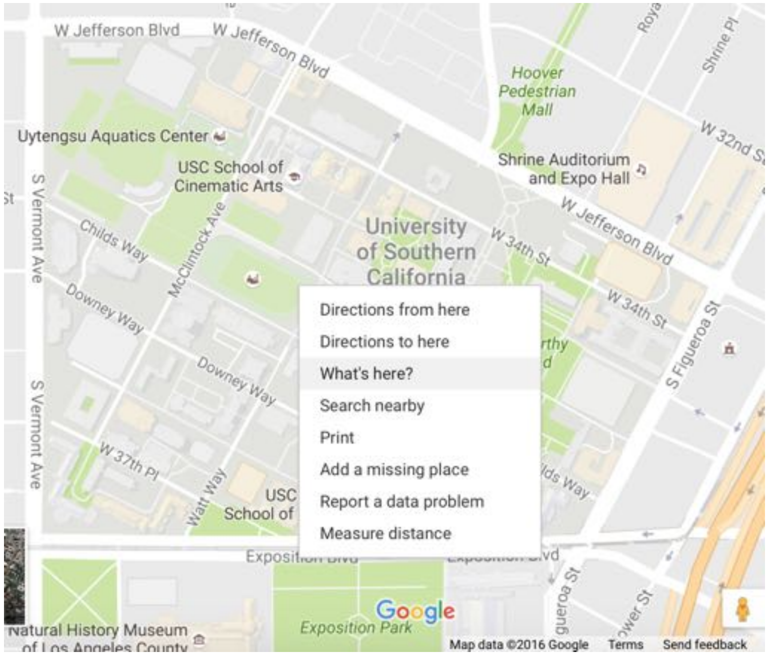
Aeronautical chart on Lambert conformal conic projection.

MERCATOR PROJECTION

PROJECTION: CYLINDRICAL, CONFORMAL



- Named after Gerardus Mercator
- **Standard for Web mapping applications**
- Variations include Web Mercator, Google Web Mercator, Spherical Mercator...

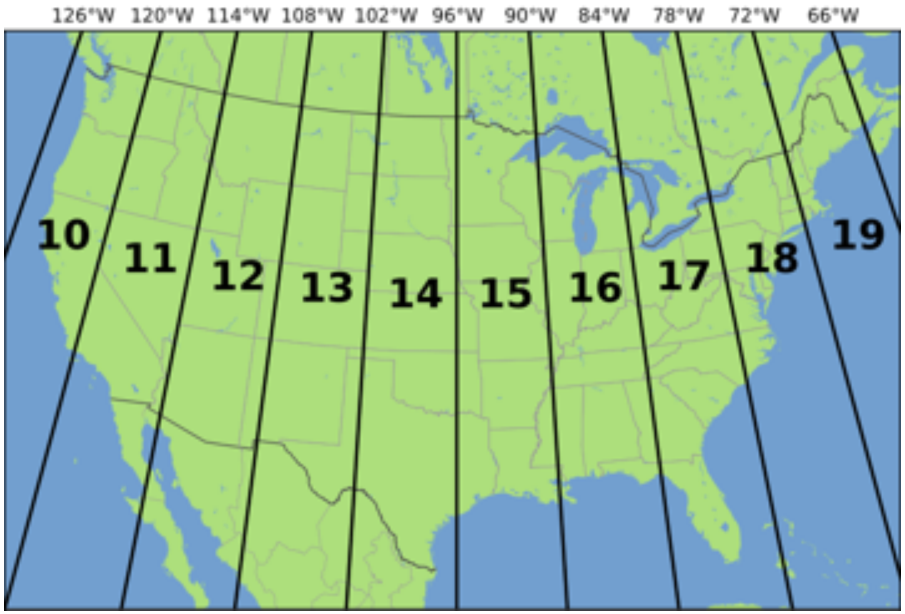


Google Maps uses Google Web Mercator

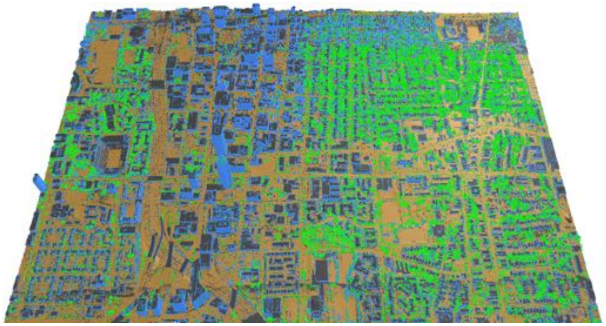
UNIVERSAL TRANSVERSE MERCATOR (UTM)

Cartesian coordinates in meters as Easting and Northing:

`48 N 377299m 1483035m`



- Named after Gerardus Mercator
- Over $61x6^o$ zones



Lidar points in UTM coordinates from Gao, Z., Nocera, L., Wang, M. & Neumann, U. Visualizing aerial LiDAR cities with hierarchical hybrid point-polygon structures, Graphics Interface Conference, 2014.

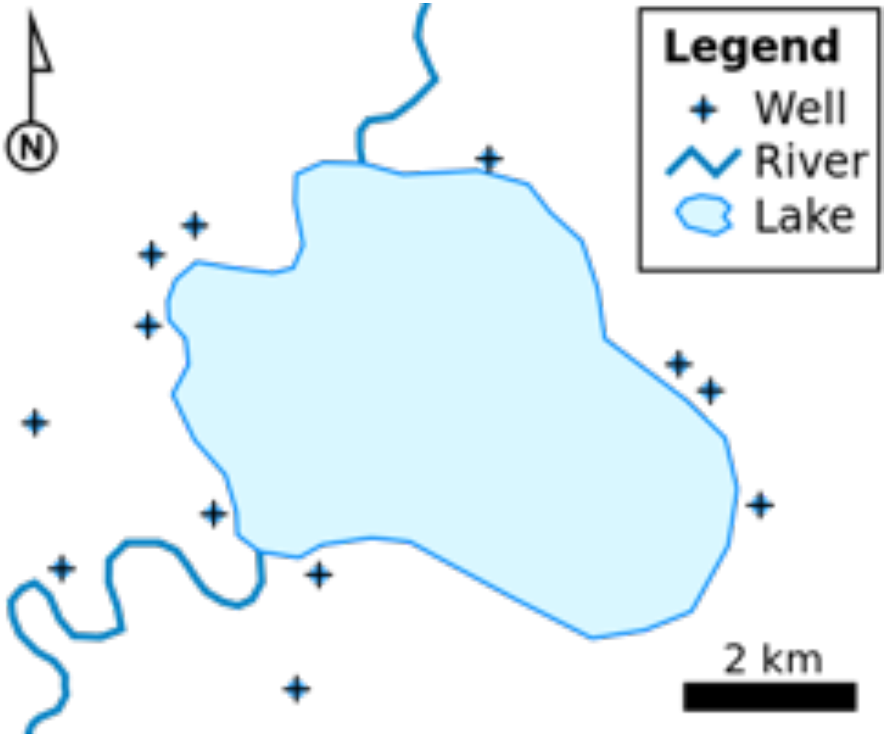
OUTLINE

- Depth perception and design considerations
- Introduction to maps
- How maps are built?
- Maps projections and their uses
- Working with Maps
  - Data formats and software
  - D3 Maps
  - Slippy Maps

MAPS FORMATS

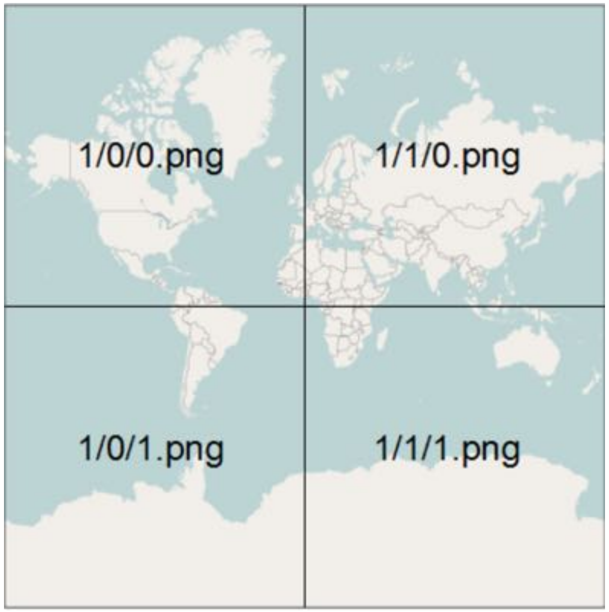
RASTER

VECTOR



TILE MAP SERVICE (TMS)

Web service providing tiles to modern web maps




```
OSM: http://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png
OSM B&W: http://{s}.www.toolserver.org/tiles/bw-mapnik/{z}/{x}/{y}.png
OpenCycleMap: http://{s}.tile.opencyclemap.org/cycle/{z}/{x}/{y}.png
MapQuest: http://otile{s}.mqcdn.com/tiles/1.0.0/sat/{z}/{x}/{y}.png
MapQuest-OSM: http://otile{s}.mqcdn.com/tiles/1.0.0/map/{z}/{x}/{y}.png
Stamen: http://{s}.tile.stamen.com/watercolor/{z}/{x}/{y}.jpg
OSM Mapnik bw: http://{s}.www.toolserver.org/tiles/bw-mapnik/{z}/{x}/{y}.png
```

MAP DATA FORMATS AND SOFTWARE

Format	Software
Raster image, e.g., png, jpeg	IMG, web mapping platforms
Vector image, i.e., svg	SVG, d3, web mapping platforms
Raster or vector tiles	web mapping platforms
JSON format: GeoJSON, TopoJSON	D3, Vega, web mapping platforms
Shapefiles (ESRI proprietary format)	GIS software

GeoJSON

```
{"type":"FeatureCollection",
  "features":[
    {"type":"Feature",
      "id":"01",
      "properties":{"name":"Alabama"},
      "geometry":{"type":"Polygon","coordinates":[[[-87.359296,35.00118],[-85.606675,34.9
    {"type":"Feature",
      "id":"02",
      "properties":{"name":"Alaska"},
      "geometry":{"type":"MultiPolygon","coordinates":[[[[-131.602021,55.117982],[-131.56
    {"type":"Feature",
      "id":"72",
      "properties":{"name":"Puerto Rico"},
      "geometry":{"type":"Polygon","coordinates":[[[-66.448338,17.984326],[-66.771478,18.
  ]
}
```

Example GeoJSON of Alabama, Alaska and Puerto Rico boundaries

GeoJSON

Geometry: Point, LineString, Polygon, MultiPoint, MultiLineString, MultiPolygon, GeometryCollection

```
"geometry": {
  "type": "Point",
  "coordinates": [-118.2851, 34.0224]
}
```

Feature: one or more geometry object with properties that can be used to encode the data

```
{
  "type": "Feature",
  "geometry": {
    "type": "Point",
    "coordinates": [-118.2851, 34.0224]
  },
  "properties": { "name": "USC" }
}
```

FeatureCollection: one or more Features

```
{
  "type": "FeatureCollection",
  "features": [ ... ]
}
```

See geojson.org for full specifications

TopoJSON

TopoJSON is a GeoJSON extension encoding topology where the Geometry is indexed with **arcs**

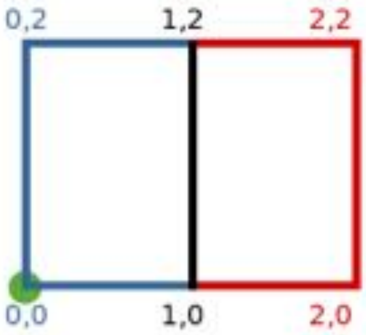
TopoJSON files are up to 80% smaller than GeoJSON files

DUPLICATE BOUNDARIES

With boundaries encoded using Geometry polygon, adjacent boundaries are duplicated and result in large GeoJSON files

ENCODING TOPOLOGY WITH ARCS


```
{
  "type": "Topology",
  "transform": {
    "scale": [0.036003600360036005, 0.017361589674592462],
    "translate": [-180, -89.99892578124998]
  },
  "objects": {
    "aruba": {
      "type": "Polygon",
      "arcs": [[0]],
      "id": 533
    }
  },
  "arcs": [
    [[3058, 5901], [0, -2], [-2, 1], [-1, 3], [-2, 3], [0, 3], [1, 1], [1, -3], [2, -5], [1, -1]]
  ]
}
```

See how To Infer Topology, Mike Bostok and topojson/topojson Wiki ‣ Introduction

OUTLINE

- Depth perception and design considerations
- Introduction to maps
- How maps are built?
- Maps projections and their uses
- Working with Maps
  - Data formats and software
  - D3 Maps
  - Slippy Maps

GEOJSON AND TOPOJSON IN D3

- Arbitrary extensions, e.g., .json, .geojson, .topojson
- Load with d3.json
- Geometry can be specified in **geographical** or **projected** coordinates
- How to load the data and the map data:
  - As separate files:
    - Using Promise.all()
    - Nesting promises, e.g., d3.json and d3.csv
    - Use d3-queue, e.g., block 1696080
  - Data embedded as geometry properties in GeoJSON

D3 GEOJSON MAP

```
<svg width="960" height="600"></svg>

<script>
var svg = d3.select("svg"), width = +svg.attr("width"), height = +svg.attr("height");

d3.json("us.json")  //1. load GeoJSON
  .then(json => {
    var projection = d3.geoAlbersUsa()  //2. set-up the projection
      .fitSize([width, height], json);

    var path = d3.geoPath()  //3. set-up a geo path generator
      .projection(projection);

    svg.selectAll("path")  //4. perform a data join with features
      .data(json.features)
      .enter()
      .append("path")
      .attr("fill", "white")
      .attr("stroke", "black")
      .attr("d", path);  //for each feature use geo path generator to generate the path
  });
</script>
```

See d3.geoPath, d3.geoAlbersUsa and d3-geo

D3 TOPOJSON MAP

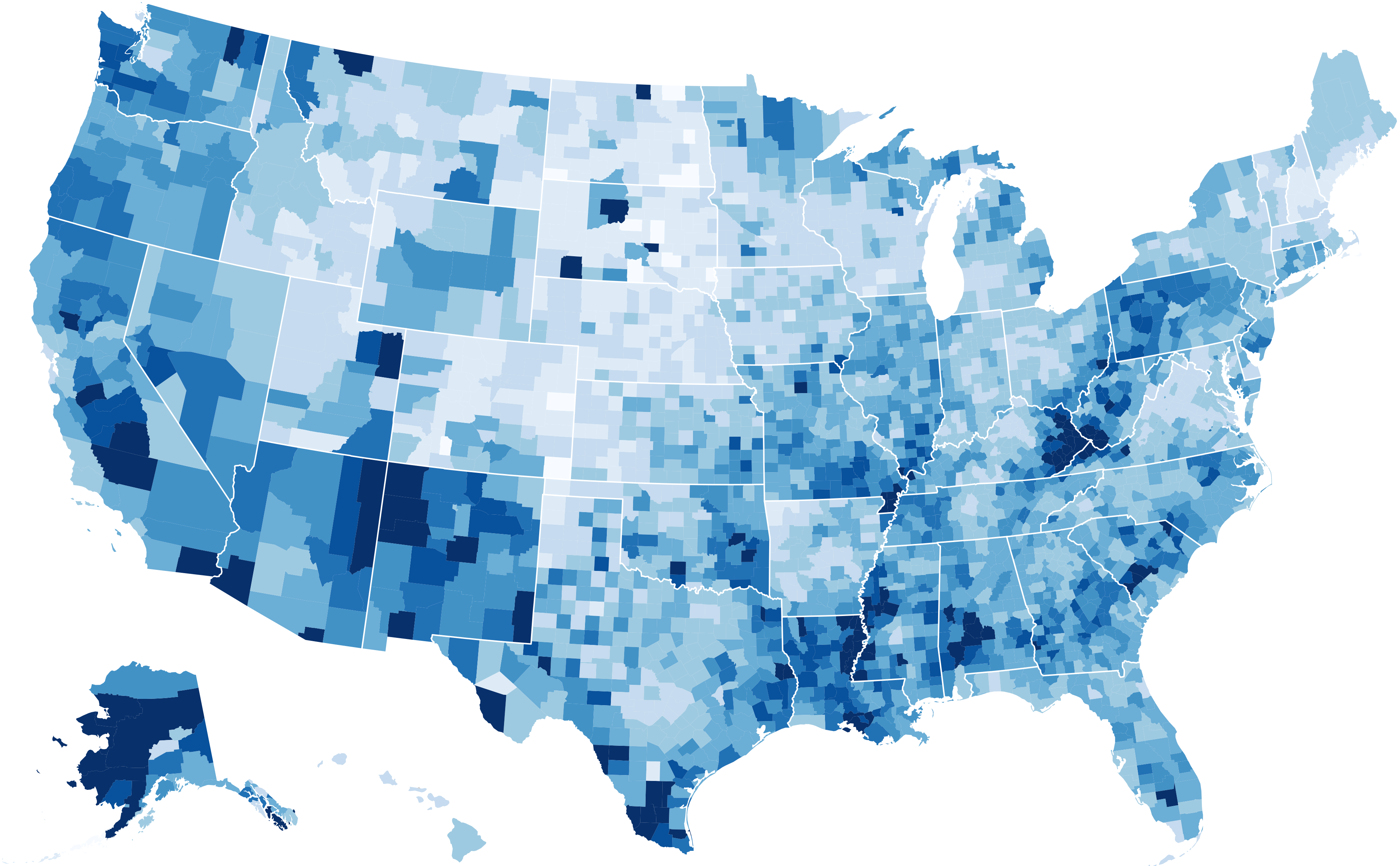
```
<svg width="960" height="600"></svg>

<script>
  d3.json("us-10m.v1.json") //1. load TopoJSON
    .then(us => {
      //2. set-up the projection (not needed as TopoJSON is already projected)
      var path = d3.geoPath();  //3. set-up a geo path generator

      var svg = d3.select("svg");
      svg.append("g")  //4. perform a data join with features
        .attr("class", "states")
        .selectAll("path")
        .data(topojson.feature(us, us.objects.states).features)  //convert to GeoJSON features
        .enter()
        .append("path")
        .attr("d", path);

      svg.append("path")
        .attr("class", "state-borders")
        .attr("d", path( //generate path for GeoJSON features of interior boundaries
          topojson.mesh(us, us.objects.states, function (a, b) { return a !== b; })
        ));
    });
</script>
```

See Choropleth in Observable and topojson.feature, topojson.mesh



SVG generated for this Choropleth in Observable

OUTLINE

- Depth perception and design considerations
- Introduction to maps
- How maps are built?
- Maps projections and their uses
- Working with Maps
  - Data formats and software
  - D3 Maps
  - Slippy Maps

SLIPPY MAPS

Modern web maps with pan and zoom

Mapbox GL JS used to render a map of UPC

```html
<!DOCTYPE html>
<html>

<head>
  <script src="https://api.mapbox.com/mapbox-gl-js/v2.10.0/mapbox-gl.js"></script>
  <link href="https://api.mapbox.com/mapbox-gl-js/v2.10.0/mapbox-gl.css"/>
</head>

<body>
  <div id="map" style="width: 960px; height: 600px;"></div>

  <script>
    //see https://github.com/mapbox/mapbox-gl-js for styles
    mapboxgl.accessToken = "MAPBOX_ACCESS_KEY";
    var map = new mapboxgl.Map({
      container: "map",  //container id
      zoom: 14,  //starting zoom level
      center: [-118.276528, 34.0192210],  //lon, lat
      style: "mapbox://styles/mapbox/dark-v10"
    });
  </script>
</body>
</html>
```

Mapbox GL JS code of previous example