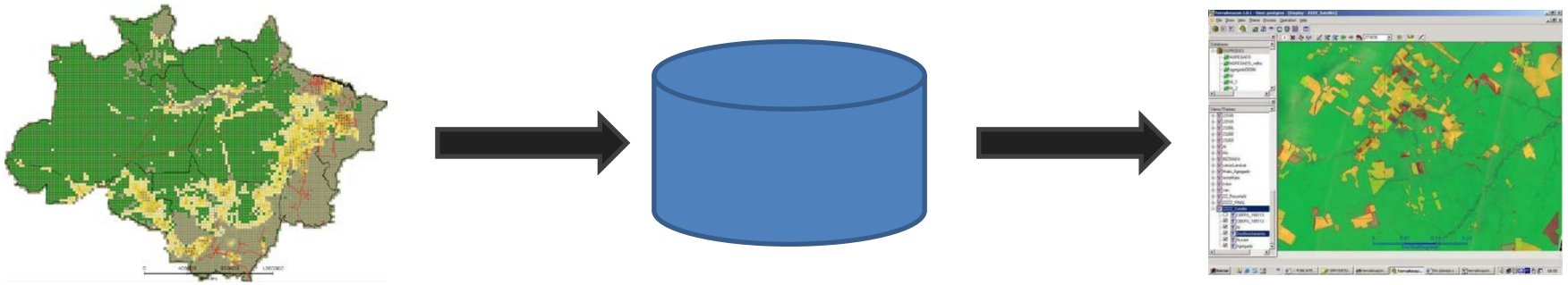
Spatial Databases: Lecture 7+8

Institute for Geoinformatics

Winter Semester 2014



Malumbo Chipofya: room 109

SPATIAL INTELLIGENCE LABORATORY

7 Exercises and Discussion

- Spatial relationships
- Spatial joins
- Projections and the Geography type
- Geometries from Geometries
- A note on validity
- Equality

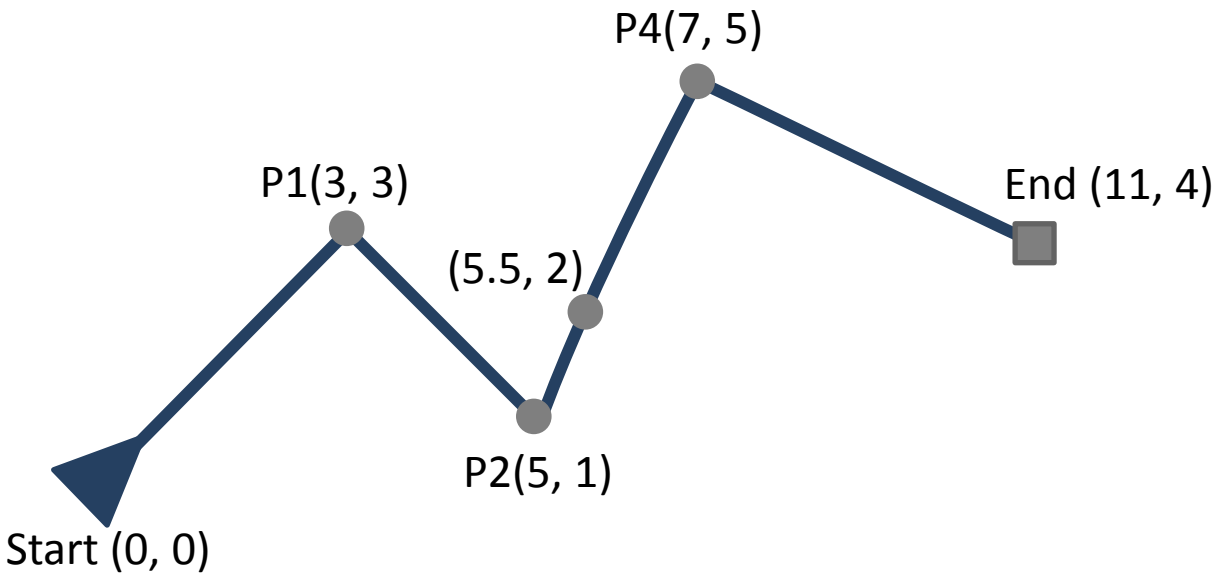
SPATIAL INTELLIGENCE LABORATORY

8 Linear Referencing & pgRouting

SPATIAL INTELLIGENCE LABORATORY

Linear Referencing

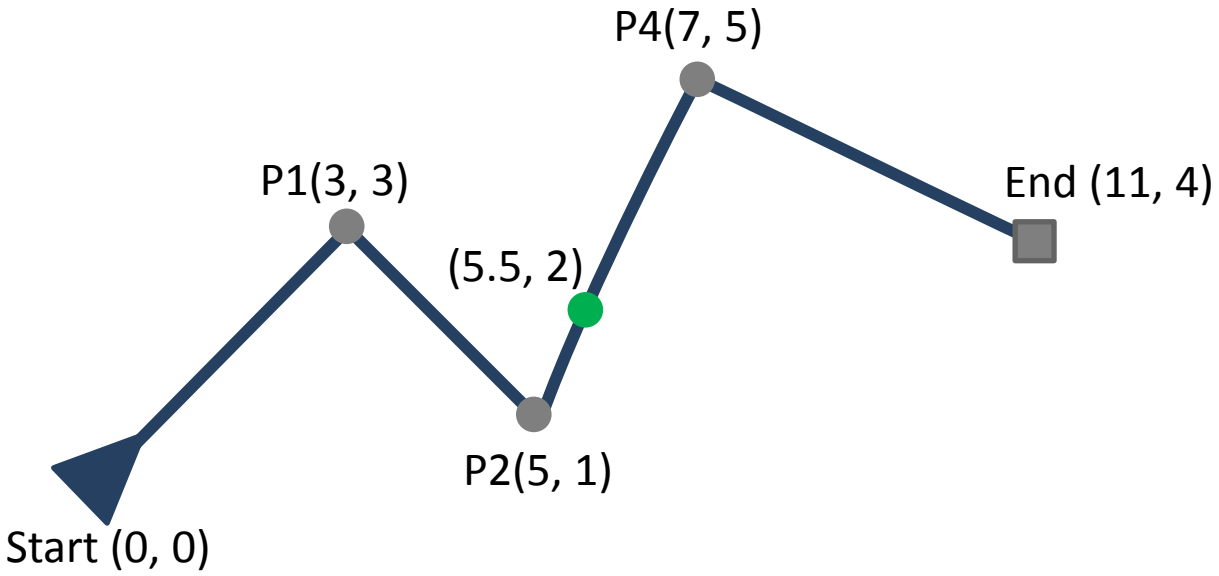
- Locate a point along a line



SPATIAL INTELLIGENCE LABORATORY

Linear Referencing

```
SELECT ST_LineLocatePoint(
        'LINESTRING(0 0, 3 3, 5 1, 7 5, 11 4)',
        'POINT(5.5 2)'
);
```



SPATIAL INTELLIGENCE LABORATORY

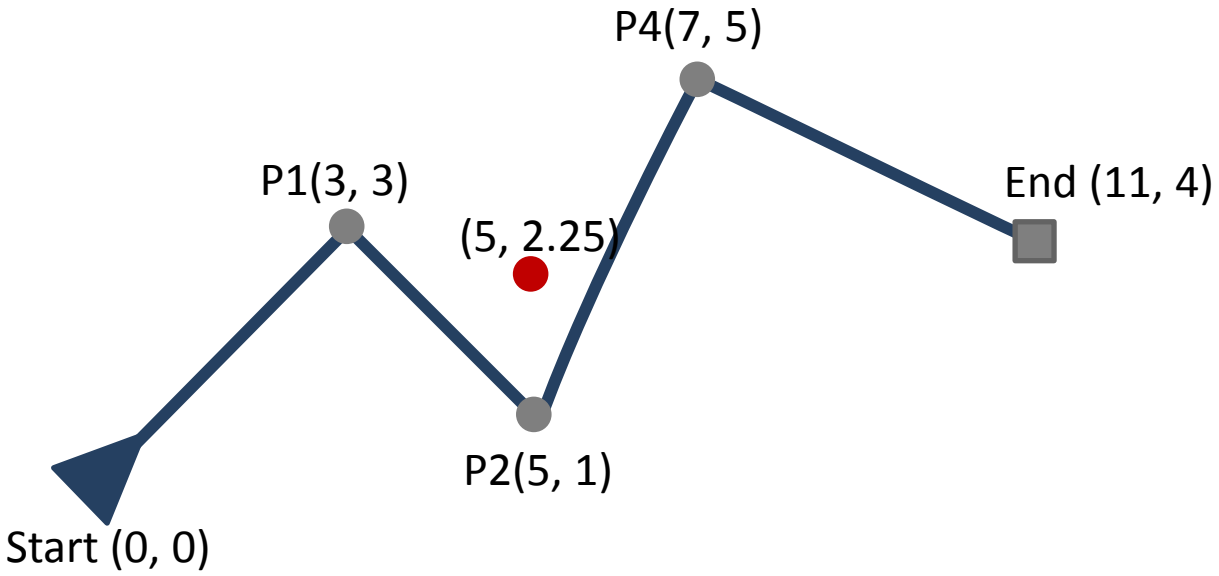
Linear Referencing

```
SELECT ST_LineLocatePoint(
        'LINESTRING(0 0, 3 3, 5 1, 7 5, 11 4)',
        'POINT(5 2.25)'
);
```

SPATIAL INTELLIGENCE LABORATORY

Linear Referencing

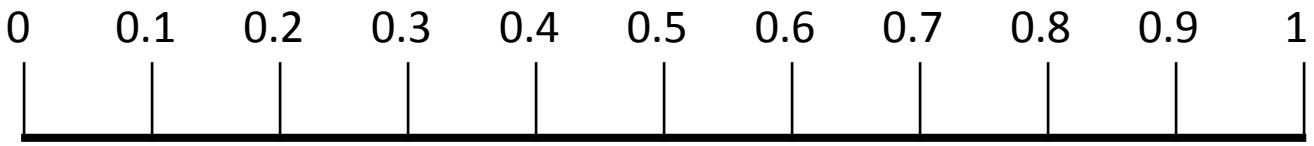
```
SELECT ST_LineLocatePoint(
        'LINESTRING(0 0, 3 3, 5 1, 7 5, 11 4)',
        'POINT(5 2.25)'
);
```



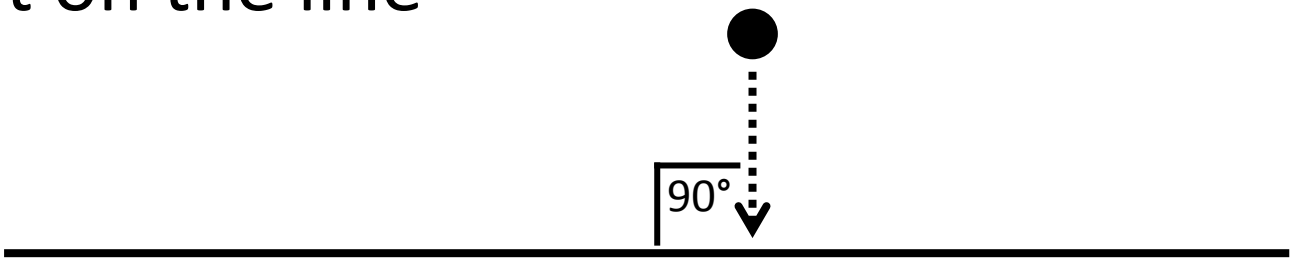
SPATIAL INTELLIGENCE LABORATORY

Linear Referencing

- ST_LineLocatePoint() returns the fraction of a line traversed from start to the located point



- Points not on line are projected to the nearest point on the line



SPATIAL INTELLIGENCE LABORATORY

Linear Referencing

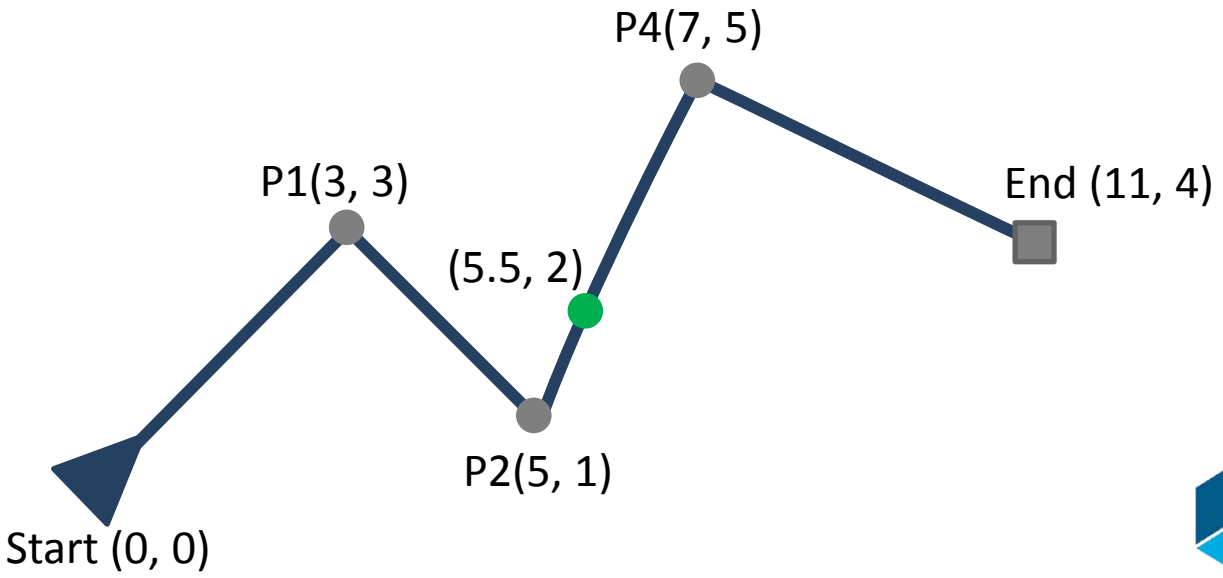
```
SELECT ST_AsText(ST_LineInterpolatePoint(
        'LINESTRING(0 0, 3 3, 5 1, 7 5, 11 4)',
        0.522720546074802
));
```

ST_LineInterpolatePoint() returns the point at the specified location along a line

SPATIAL INTELLIGENCE LABORATORY

Linear Referencing

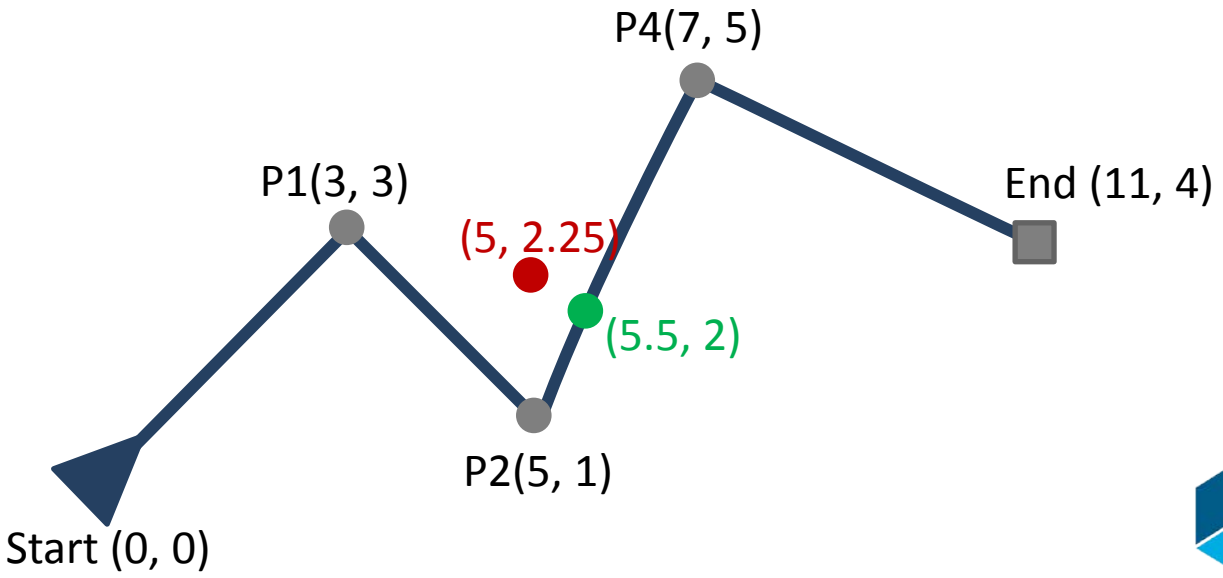
```
SELECT ST_AsText(ST_LineInterpolatePoint(
        'LINESTRING(0 0, 3 3, 5 1, 7 5, 11 4)',
        0.522720546074802
));
```



SPATIAL INTELLIGENCE LABORATORY

Linear Referencing

```
SELECT ST_AsText(ST_LineInterpolatePoint(
        'LINESTRING(0 0, 3 3, 5 1, 7 5, 11 4)',
        ST_LineLocatePoint(
                'LINESTRING(0 0, 3 3, 5 1, 7 5, 11 4)',
                'POINT(5 2.25)'
        )
));
```



SPATIAL INTELLIGENCE LABORATORY

pgRouting

- A postgres/postgis extension for computing routes and routing information
  - To use pgRouting you must first enable it for your database

```
CREATE EXTENSION pgRouting;
```

SPATIAL INTELLIGENCE LABORATORY

pgRouting

- Let's take a section of the nyc_streets database

```
SELECT DISTINCT st.* INTO si_streets FROM
        nyc_streets st,
        nyc_neighborhoods nb
WHERE
        ST_Intersects(st.geom, nb.geom) AND
        nb.boroname = 'Staten Island';
```

SPATIAL INTELLIGENCE LABORATORY

pgRouting

- To do routing we need a routable network:
  1. Ensure network is almost connected
  2. Identify street intersections
  3. Split all streets with intersections at interior points
  4. Create a network topology

SPATIAL INTELLIGENCE LABORATORY

pgRouting

- creating a topology:
  1. Identify start and end points of your streets
  2. Let pgRouting create the topology for you

```
ALTER TABLE si_streets ADD COLUMN "source" integer;
ALTER TABLE si_streets ADD COLUMN “target" integer;

SELECT pgr_createTopology(' si_streets', 0.00001, 'geom', 'gid');
```

SPATIAL INTELLIGENCE LABORATORY

pgRouting

- If not already created you can add indexes for your two new columns (check now)

  CREATE INDEX si_streets_source_idx ON si_streets("source");

  CREATE INDEX si_streets_target_idx ON si_streets("target");

- Plus we must declare a cost column

  ALTER TABLE si_streets ADD COLUMN “cost" double precision;

  ALTER TABLE si_streets ADD COLUMN “reverse_cost" double precision;

SPATIAL INTELLIGENCE LABORATORY

pgRouting

- And fill them both with up data

```
UPDATE si_streets SET
        cost = ST_Length(geom),
        reverse_cost = ST_Length(geom);
```

- The length is going to be used as our cost here

SPATIAL INTELLIGENCE LABORATORY

pgRouting

- You are now ready to route!
- A story
  - An angry person shot someone dead in Staten Island and fled by car. The event is recorded with id 1408 in the homicides table. In a haste the same person apparently run over another person and killed her as well (recorded with id 2388). The question is: assuming the person took the shortest escape route between the first and second events, which route was it?

SPATIAL INTELLIGENCE LABORATORY

pgRouting

- First let's find the points IN the street network to/from which we have to route
- Exercise!

SPATIAL INTELLIGENCE LABORATORY

pgRouting

- Dijkstra

```
SELECT seq, id1 AS node, id2 AS edge, cost FROM
pgr_dijkstra('
                SELECT gid AS id,
                        source::integer,
                        target::integer,
                        cost::double precision
                       FROM si_streets',
                P1, P2, false, false
);
```

SPATIAL INTELLIGENCE LABORATORY

pgRouting

- Dijkstra: return geometry not sequence

```
SELECT seq, id1 AS node, id2 AS edge, rt.cost,
si.geom FROM pgr_dijkstra('
                SELECT gid AS id,
                        source::integer,
                        target::integer,
                        cost::double precision
                       FROM si_streets',
        P1, P2, false, false
) rt LEFT JOIN si_streets si ON (rt.id2 = si.gid);
```

SPATIAL INTELLIGENCE LABORATORY

pgRouting

- An update to the story
  - As we dig for more evidence we encounter a witness who says he saw the getaway car between the times of the two events at Huguenot subway station (id 484). Does this fact change the most probable route?

SPATIAL INTELLIGENCE LABORATORY

pgRouting

- Exercise!

SPATIAL INTELLIGENCE LABORATORY

Exercise

- Follow the tutorial at

http://workshops.boundlessgeo.com/postgis-intro/linear_referencing.html

- Outcome: linear referencing of subway stations

SPATIAL INTELLIGENCE LABORATORY

Exercise

- Split all street linestrings containing references to subway stations into parts.
  - From the start to the first subway station
  - From each subway station to the next subway station or end of linestring whichever comes first
- Create a new streets table where any split linestrings are replaced by their pieces.
- Make the new streets table ready for pgRouting

SPATIAL INTELLIGENCE LABORATORY

Exercise

- Compute the station to station distance for all stations on a given route assuming that consecutive stations are spatially nearest to each other (break ties arbitrarily)
- Create a new table of subway routes with a line string for each route
- Make the new subway routes table ready for pgRouting

SPATIAL INTELLIGENCE LABORATORY

Exercise

- For the next tasks assume that the cost of travel by road the distance and consider three cost scenarios for travel by subway
  - Straight line distance between consecutive stations
  - 0.5 times the straight line distance
  - 2 times the straight line distance

Exercise

- Use Shortest Path Dijkstra to find the shortest routes for the following journeys using a combination of travel modes by car and subway
  - P1 to P2
  - P1 to P3
  - P2 to P4
  - P2 to P3
- Repeat the task above but this time avoid subway stations within 100 meters of a homicide

SPATIAL INTELLIGENCE LABORATORY

Exercise

- Repeat the two tasks but this time use the A* variant of shortest path
- Comment on the differences between the routes under the different cost assumptions and restrictions

References

- http://postgis.net/docs/
- http://workshops.boundlessgeo.com/postgis-intro/index.html
- http://pgrouting.org/documentation.html
- http://workshop.pgrouting.org/
  - See especially chapters 5, 7, and 8

SPATIAL INTELLIGENCE LABORATORY

That’s all for today

Thank you!

Questions?

SPATIAL INTELLIGENCE LABORATORY