TrailBlazer
Find best tracks for your outdoor trainings!

Georgia Tech

Alexandre PALO   Alex MUELLER
Guillaume BROGGI   Tianyi ZHENG

Introduction

Problem

New to an area and still want to practice mountain bike or hiking ? How to find where to train ?
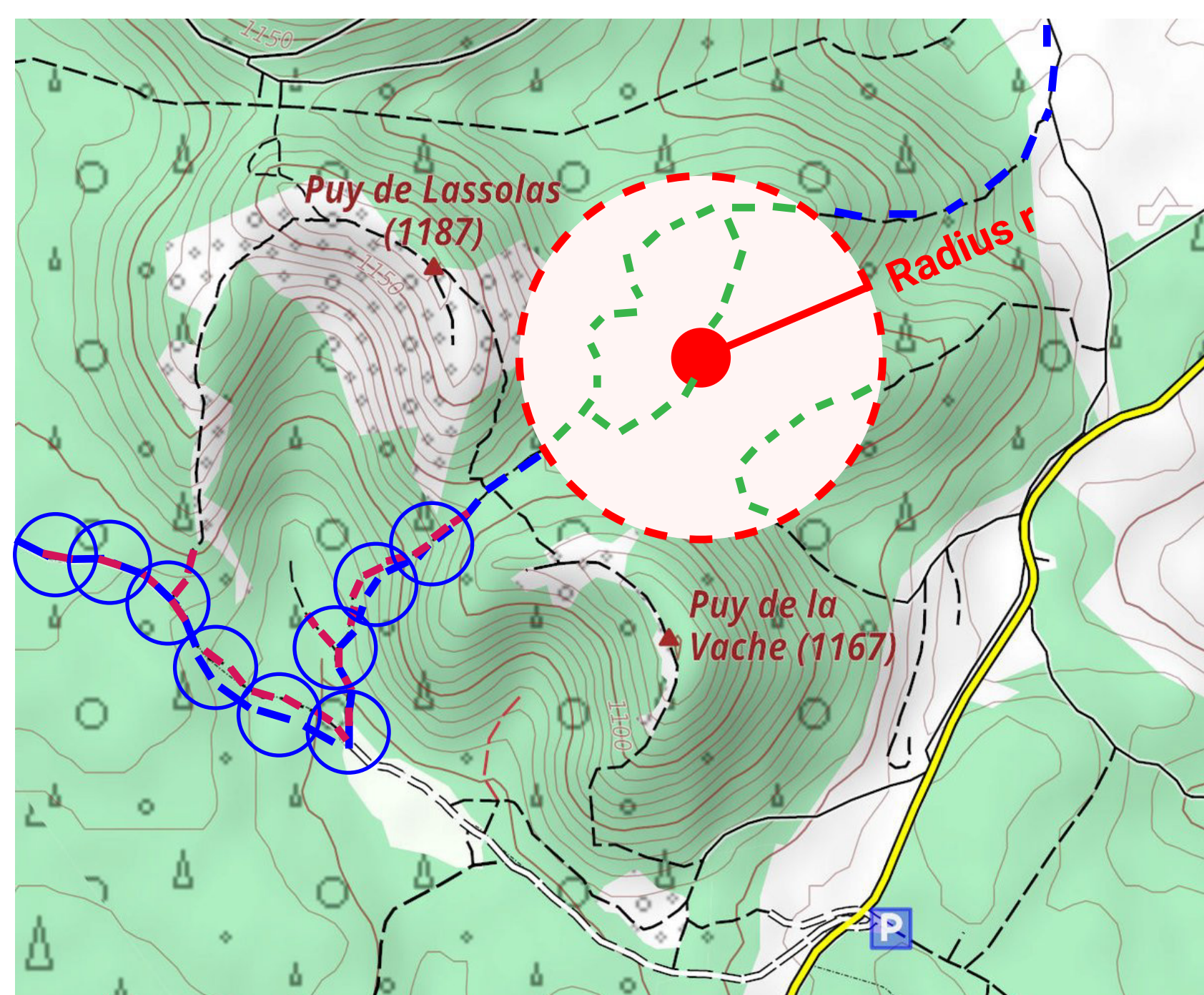
Solutions exist:
- Try to find a track with Google Maps,
- Search for point of interests online and try to reach them with a map.

But ... these solutions aren't viable:
- Google Maps try to find the short way between two points, not the most interesting one,
- If you want to reach a POI, you cannot be sure the way to it is interesting too.

Our approach

Points of interest
- Geocaching

Existing tracks
- Utagawa VTT

Forest map
- OpenStreetMap

User preferences
- Distance, interest in existing tracks & POI, begin location

Map with data
- Forest map + metadata

- Algorithm

- Solution tracks

Preprocessing   **Processing**

What's new?

Our solution takes advantages of existing information on the internet: mountain bike tracks and POI. According to user preferences, the algorithm search best tracks around and propose them through a web interface. These tracks are optimized and the best regarding user desires.

Our data & preprocessing method

Geocaching database 250,000 POI ✚ **Utagawa VTT database 11,000 mountain bike tracks**

POI preprocessing
- 🔴 POI n°X
- ⭕ Limit of neighbourhood r=1km
- Interesting roads Increase POIWeight

Tracks preprocessing
- Track n°Y
- ⭕ Track approximations r=25m
- Interesting roads Increase TrackWeight

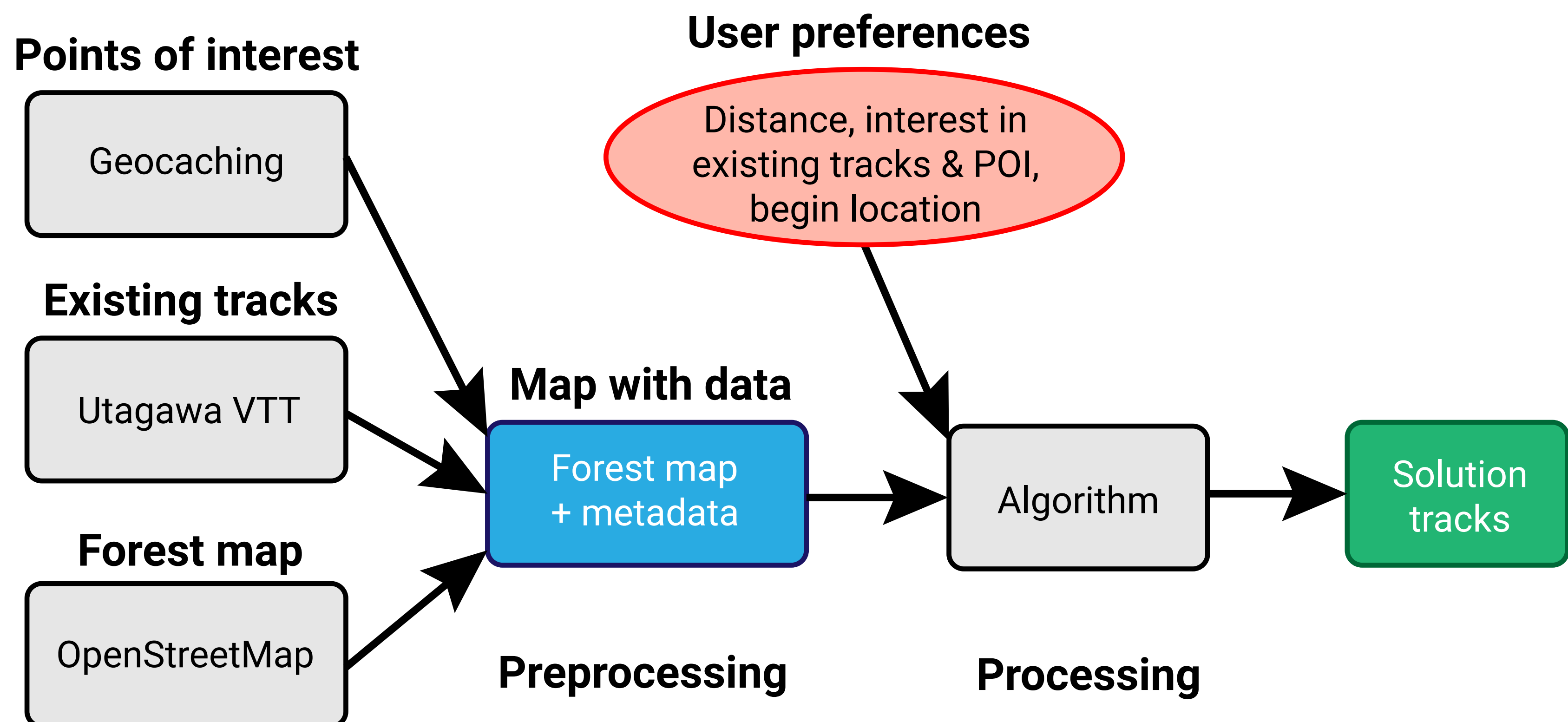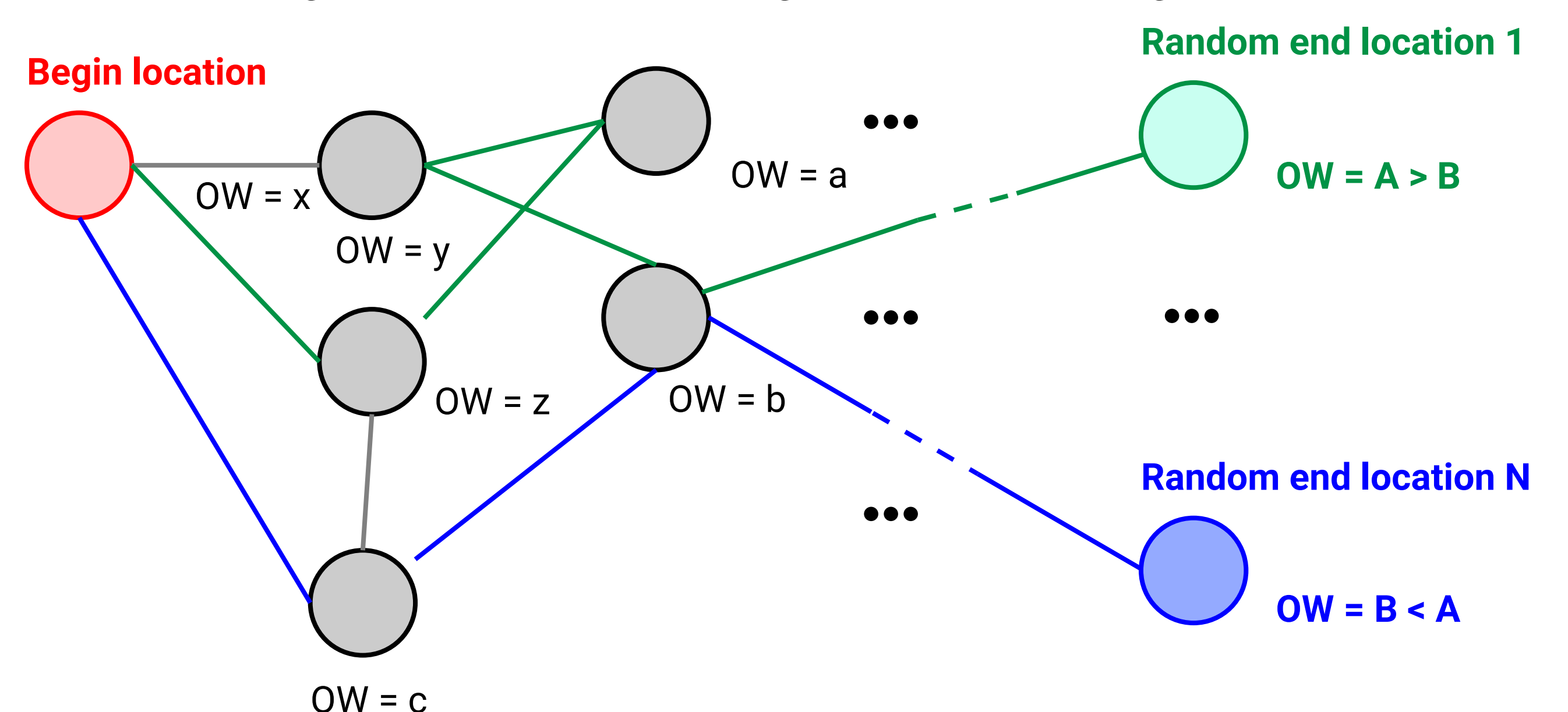➡ **GeoJson map with POIWeight & TrackWeight metadata**

Processing algorithm

Overall weight = OW = POI weight + Track weight

Begin location

OW = x
OW = y
OW = a
OW = z
OW = b
OW = c

Random end location 1
OW = A > B

Random end location N
OW = B < A

Let's choose the best generated track, ie. the larger OW value, in the N random end locations based track. Here for instance the best one is the number 1 in green.

Experiments & results

Back end experimentation

	Ave. poi weight per point	Ave. track weight per point
Random point	< 0.001	0.055
Google Maps (begin location & random end)	< 0.001	0.062
Utagawa (not correlated with begin location)	0.009	**1.24**
Algorithm (begin location & random end)	**0.012**	0.87

The algorithm gives tracks with higher weight than a simple Google Maps one. Morever it's more useful that just using random directions too. Utagawa tracks are sometimes better, but they cannot be used because not related to right begin location and track distance.

Front end experimentation

UX iteration	1	2	3
Broggi	5.0	7.5	8.0
Palo	7.0	8.0	10.0
Mueller	4.5	6.5	9.0
Zheng	5.0	7.0	8.5
Ave.	**5.34**	**7.25**	**8.88**

Personal evaluation (base 10)

For each iteration, every member give a mark for UX regarding simplicity, efficiency and quality. A new release is built until the average mark reach the threshold fixed to 8/10.