

# Projektspezifikation - SWP19 - Gruppe 5:

## Dynamische Netzwerke für TEAMWISE

### TEAMWISE

**TEAMWISE** (Tool for **E**xploration and **A**nalysis of **M**ovement data **w**ithin **I**mmersive **S**ynchronised **E**nvironments) ist ein javascript-basiertes Framework, das die interaktive Analyse von Bewegungsdaten im geographischen Kontext erlaubt. **TEAMWISE** nutzt die Cesium-Plattform (<https://cesiumjs.org/>) als Grundlage, und bietet zusätzlich unter anderem synchronisierte Perspektiven über mehrere Geräte und Browserfenster. Die Daten können dazu entweder aus dem lokalen Dateisystem oder aus der Movebank ([www.movebank.org](http://www.movebank.org)) geladen werden.

### Dynamische Netzwerke

In diesem Projekt soll eine Erweiterung für **TEAMWISE** implementiert werden, die aus geladenen Daten dynamische Netzwerke generiert, sowie eine Visualisierung und automatisierte Analyse dieser Daten ermöglicht. Dynamische Netzwerke sind Netzwerke bzw. Graphen, deren Knotenmenge konstant ist, während die Kantenmenge sich verändern kann, in der Regel in Abhängigkeit von der Zeit. Im Gegensatz zu allgemeinen Netzwerken, kommt in unserem speziellen Fall die Tatsache hinzu, dass die einzelnen Knoten jeweils bereits eine Position im Raum besitzen und insofern ein geringeres Augenmerk auf Fragestellungen wie Layout-Algorithmen liegt.

### Die Erweiterung

Das Grundgerüst der zu entwickelnden Erweiterung soll aus den folgenden vier Schritten bestehen:

1. Netzwerkgenerierung
2. Netzwerkvisualisierung
3. Netzwerkanalyse
4. Visualisierung der Ergebnisse der Analyse

Diese werden im weiteren Verlauf dieses Dokumentes genauer erläutert und jeweils eine Mindestanforderung genannt.

#### 1. Netzwerkgenerierung

- Zunächst die Implementierung einer geeigneten, flexiblen Datenstruktur für den systeminternen Umgang mit Netzwerken. Dies kann entweder durch eigene Implementierung erfolgen, oder aber unter Verwendung einer existierenden JavaScript-Bibliothek. Dabei bieten sich mehrere Ansätze an:
  - o **jsnetworkx**, welches eine JavaScript-Portierung der Python-Bibliothek **networkx** darstellt (<http://jsnetworkx.org/index.html>).
  - o **graph-data-structure**, welches eine minimalistische Implementierung einer Graphdatenstruktur bietet, die einige Standardalgorithmen mitliefert (<https://www.npmjs.com/package/graph-data-structure>).
-

Nun gibt es verschiedene Möglichkeiten, aus den gegebenen Bewegungsdaten tatsächlich Netzwerke zu generieren. Im Rahmen dieses Projekts sollen zunächst drei Methoden zur Generierung implementiert werden, welche wiederum aus zwei Attributen berechnet werden: Distanz und Orientierung. Die Grundidee dabei ist es, eine Kante zwischen zwei Knoten einzufügen, wenn ein bestimmtes Attribut geringer als eine vorgegebene Grenze ist.

- o **Distanznetzwerk:** Eine Kante zwischen zwei Knoten  $u$  und  $v$  existiert genau dann, wenn die räumliche Distanz zwischen  $u$  und  $v$  kleiner als eine festgelegte Schranke  $s$  ist, also  $\{u, v\} \in E$  genau dann, wenn  $\text{dist}(u, v) \leq s$  gilt.
- o **Orientierungsnetzwerk:** Eine Kante zwischen zwei Knoten  $u$  und  $v$  existiert genau dann, wenn der Unterschied in den Orientierungen der Individuen  $u$  und  $v$  kleiner als eine festgelegte Schranke  $s$  ist, also  $\{u, v\} \in E$  genau dann, wenn  $|\text{orientation}(u) - \text{orientation}(v)| \leq s$  gilt.
- o **Distanz-Orientierungs-Netzwerk:** Eine Kante zwischen zwei Knoten  $u$  und  $v$  existiert genau dann, wenn die räumliche Distanz zwischen  $u$  und  $v$  kleiner als eine festgelegte Schranke  $s^1$  ist **und** der Unterschied in den Orientierungen der Individuen  $u$  und  $v$  kleiner als eine festgelegte Schranke  $s^2$  ist, also  $\{u, v\} \in E$  genau dann, wenn  $\text{dist}(u, v) \leq s$  **und**  $|\text{orientation}(u) - \text{orientation}(v)| \leq s$  gilt.

Die Schranke  $s$  soll dabei vom Benutzer festgelegt werden können.

• **Optional:** Generierung des Netzwerkes aus 3-dim. Gabriel-Graph.

## 2. Netzwerkvisualisierung

- Dies umfasst die Darstellung des Netzwerkes innerhalb von **TEAMWISE**. Die dynamischen Kanten sollen also während der (laufenden) Animation zwischen den Individuen eingezeichnet werden.
- **Optional:** Je nach zeitlichem Projektverlauf soll es später auch noch die Möglichkeit geben, das erzeugte Netzwerk zwei-dimensional, abstrahiert darzustellen und ggf. Layout-Verfahren darauf anzuwenden. Dies wird aber, falls relevant, im weiteren Projektverlauf gemeinsam mit den Betreuern besprochen werden.

## 3. Netzwerkanalyse

Die Analyse der erzeugten Netzwerke bietet nun das Herzstück der Erweiterung. Sobald ein Netzwerk generiert wurde, haben wir die ganze Toolbox an Netzwerkanalyse-Verfahren zur Verfügung, von denen es nun gilt, eine geeignete Teilmenge im System zu implementieren.

- Analyse auf Individuen-Level
  - o Zentralitätsmaße
    - ♠ Degree-Centrality
    - ♠ Eccentricity
    - ♠ Closeness
    - ♠ Betweenness
- Analyse auf Gruppen-Level
  - o Bestimmung der zweifachen Zusammenhangskomponenten
- Analyse auf globalem Netzwerk-Level
  - o Anzahl der Kanten / Kantendichte (Verhältnis der existierenden Kanten zu den überhaupt möglichen)
  - o Durchschnittsgrad
  - o Anzahl der zweifachen Zusammenhangskomponenten
  - o **Optional:** Clustering Coefficient

## 4. Visualisierung der Ergebnisse der Analyse

Schließlich gilt es, die Ergebnisse der Analyse in geeigneter Weise darzustellen:

- Für die individuellen Werte eine dynamische Matrix, welche beispielsweise durch Farbgradienten die jeweilige Höhe der Zentralitätsmaße angibt. Dabei sind die Individuen den Zeilen zugeordnet, während die Spalten eine kleine Umgebung der derzeitigen Timeframes umspannt.
- Für die globalen Statistiken bieten sich natürlich die klassischen, einschlägigen Visualisierungen an: Balkendiagramm, Kreisdiagramm, Plotting, ...
- Hier kann auf die bereits in **TEAMWISE** enthaltene **d3.js** Bibliothek zurückgegriffen werden ( <https://d3js.org/>).

### Weitere Bedingungen

- Dokumentation / Kommentierung
- Tests
- Modularer Aufbau: Die Software soll sauber, modular aufgebaut sein, sodass es in Zukunft leicht möglich ist,
  - o weitere Netzwerk-Generierungs-Ansätze hinzuzufügen,
  - o weitere Analysemethoden hinzuzufügen,
  - o die berechneten Attribute (Zentralitätsmaße, Netzwerkstatistiken,...) anderen Programmteilen zugänglich zu machen, z.B. dem Dashboard (Gruppe 4).