

# Knowledge-based Spatial Reasoning for IoT-Enabled Smart City Applications

Ralf Toenjes, Daniel Kuemper, Marten Fischer

Faculty of Engineering and Computer Science, University of Applied Sciences Osnabrück  
Osnabrück, Germany

Email: {r.toenjes, d.kuemper, m.fischer} @hs-osnabrueck.de

**Abstract**—For efficient analysis of large smart city data sets in (real) time a knowledge-based approach is suggested, constraining iteratively the search space while increasing accuracy by more complex computation. Results are shown for searching the distance to the nearest object, e.g., parking space.

## I. INTRODUCTION

The Internet of Things (IoT) has a high potential to enhance processes in urban environments and to ease the citizens' live, thus enabling smart cities. However, the uptake of smart city applications is hindered by various issues, such as the difficulty of integrating heterogeneous data sources and the challenge of extracting up-to-date information in real-time from large-scale dynamic data. The European project CityPulse<sup>1</sup> addresses these issues by developing a framework for real-time IoT stream processing and large-scale data analytics. The framework addresses the issues above by four major concepts: (1) Virtualisation hides the heterogeneity of the numerous data and information sources. (2) The large amount of raw data without intrinsic explanation remains meaningless. To enable automated machine interpretation the data is semantically annotated. (3) Large-scale data analytics tools allow for resource efficient event detection in multiple data streams. (4) This data pipeline is controlled by knowledge based reasoning. This split separates the big data processing with need for efficiency from the intelligent reasoning employing ontologies for knowledge representation [1][2]. Reasoning in cities depends heavily on the spatial context. While temperature will be similar in the neighbourhood, noise propagation will depend on shielding buildings and traffic flows on road networks, ongoing construction work, traffic density etc.. Hence, spatial reasoning requires appropriate distance measures. However, the integration of large amounts of data sources and the computational complexity demand for efficient methods to provide the requested information in (real) time. Therefore this paper will focus on spatial reasoning. The next sections will a) briefly review the state of the art, b) discuss different distance measures, c) outline an iterative approach to bring in context knowledge constraining the search space and d) employ spatial interpolation to increase reliability.

## II. STATE OF THE ART

Interpolation of spatial data typically utilises variations of the Euclidean distance to find suitable sources of neighbouring sensors and to weight their impact[3]. Alternative weighting methods are utilising further sensor data (like wind data) [4][5]. This paper proposes the use of available infrastructure knowledge for sensor data analysis.

## III. MODELLING DISTANCE

The Euclidean distance between two locations usually defines a brief coherence if e.g., certain events affect nearby entities or persons. However, applied in a complex city environment this metric does not represent the relevance of nearby events. Road-networks and encapsulated public transportation systems have to be considered since the beeline often does not reflect possible ways or distinct connections that can be used to reach or affect another location. Figure 1 shows the different distance models that can be used in the space of a city.

- The direct radial propagation of an effect issued on a specific location that results in a Euclidean distance as a distance model.
- The consideration of a directed street graph enables a more distinct model for measuring a distance, interpolating traffic related data and calculating the propagation of traffic-related events.
- Public Transportation systems like trains allow only distinct boarding and exit stops, which further decrease the flexibility of the public space and reduce the details of the utilised graph.

Let a distance function  $d$  (e.g., Euclidean on a metric projection where 1 unit is 1 meter) be defined on space  $X$ .  $K$  is a set of indices and the tuple  $(P_k)_{k \in K}$  of nonempty subsets (the locations/events) in  $X$ . The Voronoi cell,  $R_k$ , associated with the site  $P_k$  is the set of all points in  $X$  whose distance to  $P_k$  is not greater than their distance to the other sites  $P_j$ , where  $j \neq k$ . The gap-less Voronoi diagram is defined as the tuple of cells  $(R_k)_{k \in K}$ . Figure 2 shows the Voronoi cells for traffic flow sensors, which are deployed in the road network of the city. The Voronoi cells illustrate that the nearest traffic sensor is unlikely to represent the condition of the illustrated street segments.

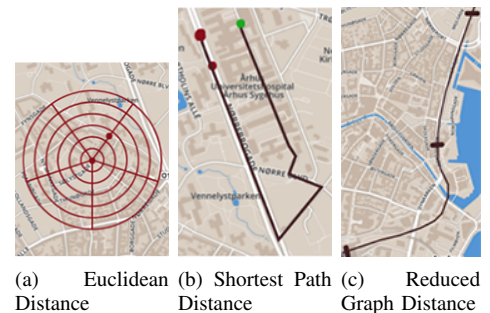


Fig. 1. Comparison of Spaces to Determine Distances

<sup>1</sup>European Union 7th FP under grant agreement n° 609035

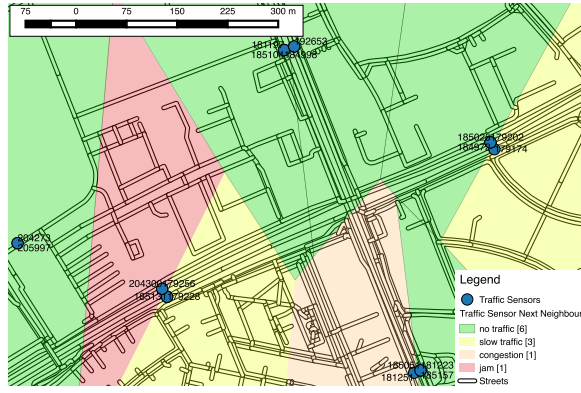


Fig. 2. Voronoi Diagram - Depicting the Nearest Traffic Sensor (labelled with a number) and Traffic Condition Value for Every Street Segment Inside a Voronoi Cell.

### DISTANCE MEASURE ACCURACY

To find locations such as nearest parking space or closest hospital an iterative strategy is suggested reducing candidates and increasing accuracy and computational complexity step-wise. While the computation of the Euclidean distance is fast, the consideration of the road network is more accurate. In a first step the Euclidean distance can reduce candidates. But for exact navigation in a city the roadmap is needed.

To analyse the accuracy limits for a real scenario an infrastructure dataset from the city of Aarhus in Denmark was investigated, containing a routable graph, 3 hospitals, 13 pharmacies, 25 atms, 36 toilets, 45 waste baskets and 288 parking places in the examined area. For random locations in the city the Euclidean distance and the routing distance, calculated with a shortest path algorithm in the cities street network, have been compared. Table I shows an exemplary result of the experiment. The column *From(lon lat)* describes the starting point of the search; *OsmId* is OpenStreetMap identifier of the hospital that was found in the search; *EuclDist(m)* shows the Euclidean distance between the starting point and the hospital in meters; *RouteDistance(m)* shows the shortest path, using the street network of the city in meters. In this example, the first hospital appears to be the nearest one by utilising the simple Euclidean distance. Taking into account the road network, the second hospital is 11% nearer than the first one. Figure 3 shows the overall results of the experiment. It depicts the error ratio in a list of the nearest 2-5 nearest objects. If the Euclidean distance order was correct an entry is marked as correct. If the routing distance alters the order, the route is marked as incorrect. The experiment was repeated with 20000 random locations for the 6 objects. Since there are only 3 hospitals available, the hospital experiment was capped at a list length of 3. For this experiment one-ways and vehicle restrictions have been ignored.

The distance difference depends on terrain and road interconnectivity, based on the length of street segments and the street density. These static measures can be computed in advance and are used to control the stepwise selection and computation of distance measures in (real) time.

Order	Object	From(lon lat)	OsmId	EuclDist	RouteDist
1	hospital	10.19062 56.19125	52592	2665	3422
2	hospital	10.19062 56.19125	10646	2694	3046

TABLE I. EXAMPLE RESULTS FOR MISLEADING DISTANCES

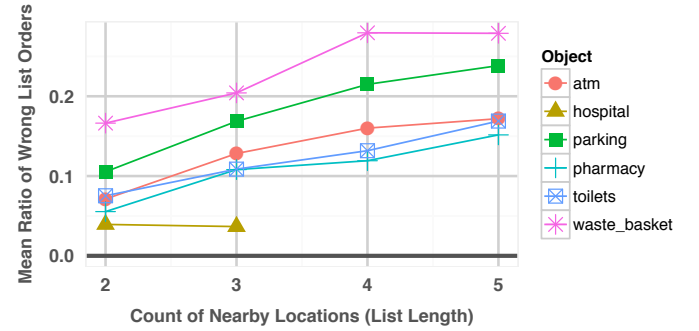


Fig. 3. Evaluating Euclidean Distance Results with Actual Road Distances

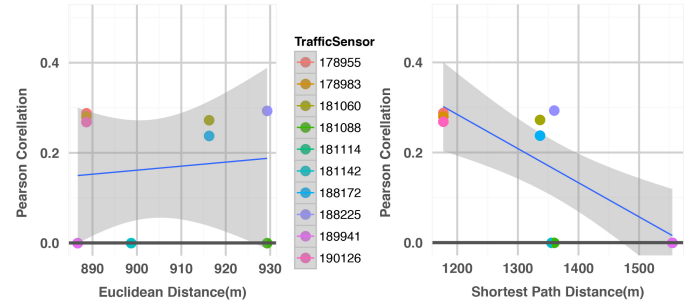


Fig. 4. Evaluation of Pearson Correlation Between Traffic of Parking Garage and Nearest Traffic Sensors on Streets - Comparison of Euclidean Distance(left) and Shortest Path Distance(right)

### IV. KNOWLEDGE BASED SPATIAL INTERPOLATION

Sensor data is often correlated and can be combined to increase reliability. Figure 4 shows the Pearson correlation between parking garage usage and close traffic sensors. The grey variance area of the regression points out the necessity to use shortest path distance for reliable sensor fusion. Selecting sensor pairs by shortest path results in higher correlation (Pearson correlation, Minkowski distance and Bray-Curtis dissimilarity) than using the Euclidean distance.

### REFERENCES

- [1] R. Tönjes, P. Barnaghi *et al.*, "Real time iot stream processing and large-scale data analytics for smart city applications," in *European Conference on Networks and Communications (EUCNC)*, Bologna, 2014.
- [2] A. Mileo, "Web stream reasoning: From data streams to actionable knowledge," in *Reasoning Web. Web Logic Rules*, ser. Lecture Notes in Computer Science, W. Faber and A. Paschke, Eds. Springer International Publishing, 2015, vol. 9203, pp. 75–87.
- [3] D. Zimmerman, C. Pavlik, A. Ruggles, and M. P. Armstrong, "An experimental comparison of ordinary and universal kriging and inverse distance weighting," *Mathematical Geology*, vol. 31, no. 4, pp. 375–390, 1999.
- [4] L. De Mesnard, "Pollution models and inverse distance weighting: Some critical remarks," *Computers & Geosciences*, vol. 52, pp. 459–469, 2013.
- [5] L. Li, J. Gong, and J. Zhou, "Spatial interpolation of fine particulate matter concentrations using the shortest wind-field path distance," 2014.