An Infrastructure-based Interpolation and Propagation Approach for IoT Data Analytics

Daniel Kuemper, and Ralf Toenjes

Lab for RF-Technology and Mobile Communications
University of Applied Sciences Osnabrueck, Germany
Email: d.kuemper@hs-osnabrueck.de

Elke Pulvermueller

Institute of Computer Science
University of Osnabrueck, Germany

***Abstract*—Interpolation of data in smart city architectures is an eminent task for the provision of reliable services. Furthermore, it is a key functionality for information validation between spatiotemporally related sensors. Nevertheless, many existing projects use a simplified geospatial model that does not take the infrastructure, which affects events and effects in the real world, into account. There are various available algorithms for interpolation and the calculation of routes on infrastructure based graphs and distances on geospatial data. This work proposes a combined approach by interconnecting detailed geospatial data whilst regarding the underlying infrastructure model.**

***Index Terms*—Inverse Distance Weighting; Kriging; Shortest Path; OSM; Smart Cities; IoT**

1. Introduction

The availability of appropriate, accurate, and trustworthy data sources in smart city environments is rapidly growing. However, especially in smaller cities, the amount of available sensor and actuator data sources has not yet reached a critical boundary where available smart city applications could be easily adapted to any new urban area. Furthermore, Smart city infrastructures are using a variety of different information sources facing a divergent reliability in heterogeneous IoT infrastructures. Due to a frequent unavailability of precise sensor data and a missing ground truth, there is a high need for interpolation of available information sources. Common approaches store spatial information as a simple set of coordinates or geometries whereby the entity relationships (e.g., a sensor belonging to a building) and infrastructural limitations (e.g., blocking of light and sound at a large object or the propagation of traffic jams along the streets) are not considered. Furthermore applied interpolation methods do not reflect these infrastructural limitations. This work proposes the integration of a infrastructure based distance algorithm into the common Inverse Distance Weighting (IDW) interpolation method. By integrating freely available infrastructure data and having an open source implementation it allows easy integration into own projects. A set of freely available tools and data sets is used to evaluate the approach and show the advancements of advanced interpolation methods.

The remainder is structured as follows: Section 2 presents the state of the art whereas section 3 describes the integration of advanced distance metrics into the IDW. Section 4 shows a visualised example of the algorithm adaptation whereas a performance evaluation is discussed in section 5. The paper concludes in section 6.

2. State of the Art

The selection of suitable spatial interpolation methods has a huge impact on the quality of site-specific maps that are widely used in farming, environmental and weather applications where only a few infrastrucutral barriers are present [1] [2]. Furthermore, temporal information is considered [3] to enable predictions of spatiotemporal distributions and propagation. Although for routing applications the utilisation of infrastructure-based street graphs is a state of the art approach [4] an integrated knowledge of infrastructure knowledge and time-related propagation has not been reflected for interpolation methods in recent publications [5].

Extended geometries often provide useful expert knowledge beyond point/coordinate information. Interpolation techniques on the other hand either use only point input data or reduce line information to point information. The authors of [6] describe the L-IDW method, which takes the line information and interpolates based on the distances raster point to neighbouring lines. Especially in flat areas, the benefits of L-IDW are obvious for many process modelling approaches. In contrast to our proposed approach, advanced geometry descriptions are taken into account but do not alter the distance model itself.

Having a validated ground truth in heterogeneous smart city networks at any needed location requires extensive processing [7]. The authors of [8] and [9] developed and evaluated a concept for the assessment of node trustworthiness in a network (vehicular ad-hoc networks VANETs to be precise) based on data plausibility checks. For this, they employed a Bayesian filter consisting of a predict/update cycle.

They propose that every node performs a plausibility check to identify malicious nodes sending faulty data. Similarly to this work they use similar data sources in order to find "witnesses" for a given sensor reading. The authors in [10] propose three different approaches to deal with a missing validated data in social media: spatiotemporal, causality, and outcome evaluation. Their concept to use spatiotemporal evaluation to predict future behaviour of humans is similar to our approach disregarding that we evaluate past events.

Prior work of the authors emphasised the importance of an appropriate distance model reflecting infrastructure, e.g., roads, and physics, i.e. traffic or air movements [11]. In [12] the authors show that infrastructure knowledge is eminent for data analytics in the area of IoT and Smart City. The simple utilisation of Euclidean distances should be avoided when datasets are analysed or interpolated. Therefore, this work proposes the utilisation of more sophisticated distance functions that take infrastructure knowledge, like streets, train networks buildings and rooms, into account.

3. Infrastructure Oriented Inverse Distance Weighting

Although Inverse Distance Weighting (IDW) results in applicable results for large area interpolations in areas like precision farming and radiation dispersion models, it is not directly applicable for areas and data sources, which are bound to city infrastructure. Figure 1 shows a full IDW interpolation for a city area. Four Individually reported traffic incidents of varying severity at different are used as an input for the IDW. The Euclidean model does not provide an applicable distance metric for this kind of data since there is no information about the street connections used. Therefore, streets are affected by nearby sensors although the may be no close connection between them.

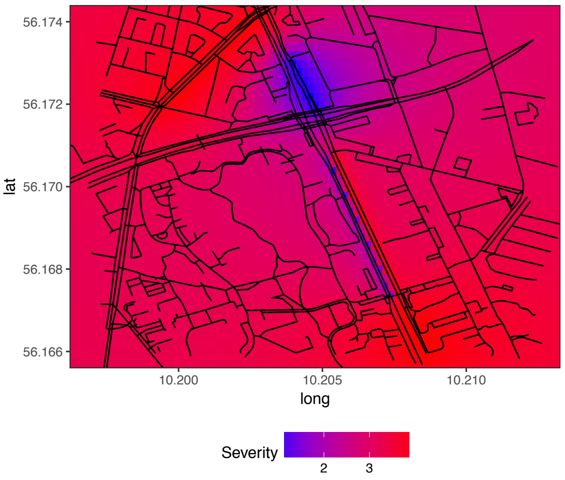


Figure 1: Inverse Distance Weighting Interpolation of Available Traffic Information

Like various common interpolation algorithms in the area of geospatial data analysis, the deterministic IDW method [13] utilises an euclidean model where the distance from x_a to x_b and the distance from x_b to x_a is determined by the length of the line segment connecting them ($\overline{\mathbf{x_a x_b}}$). In Euclidean n-space $\mathbb{R}^n$ it is given by the Pythagorean formula.

$$\begin{aligned}&\mathrm{d_e}(x_a, x_b) = \mathrm{d_e}(x_b, x_a)\\&= \sqrt{(x_{a_1} - x_{b_1})^2 + (x_{a_2} - x_{b_2})^2 + \cdots + (x_{a_n} - x_{b_n})^2}\\&= \sqrt{\sum_{i=1}^{n}(x_{a_i} - x_{b_i})^2}\end{aligned} \tag{1}$$

IDW takes into account a finite set of n samples with their location x_i and their values $z(x_i)$ with $i \in \mathbb{N}$. The value at a given location x_0 is calculated by the estimator function

$$\begin{aligned}z^*(x_0) &= \sum_{i=1}^{n} \lambda_i(x_0) \cdot z(x_i)\\&= \lambda_1(x_0) \cdot z(x_1) + \lambda_2(x_0) \cdot z(x_2) + ... + \lambda_n(x_0) \cdot z(x_n)\end{aligned} \tag{2}$$

with λ as a weight-function.

$$\lambda_i(x_0) = \frac{1}{d(x_0, x_i)} \text{ if } d(x_0, x_i) \neq 0 \tag{3}$$

It can be assumed that $z^*(x_i) = z(x_i)$, since the distance is 0. To avoid the division by 0 the function is normalized as

$$\lambda_i^*(x_0) = \frac{\lambda_i(x_0)}{\sum_{l=1}^{n} \lambda_l(x_0)}$$

To achieve an exponential neglection of samples, which have a high distance, the x_{th} power can be applied.

$$z^*(x_0) = \frac{\sum_{i=1}^{n} \frac{z(x_i)}{d_e^x(x_0, x_i)}}{\sum_{i=1}^{n} \frac{1}{d_e^x(x_0, x_i)}}$$

To avoid the simplified Euclidean distance model $d_e(x_0, x_i)$ will be substituted with a path-related distance $d_p(x_0, x_1, g_i, c_f)$, which is based on a directed infrastructure graph g_i and a configurable cost function c_f. c_f defines the parameters that are used to determine the information-distance between x_0 and x_i. It is individually based on the scenario. For temperature-based interpolations, the overcoming of a building barrier or a vehicle entity as barrier results in a high distance since the measured temperatures may not correlate. In a traffic propagation scenario, the shortest path between two locations is the elementary cost factor. This shortest path distance is calculated out of the sum of the individual Euclidean distances of waypoints on the edges e_{sp} on g_i whilst considering movement restrictions like one-ways. Furthermore switching between different roads or road types lowers the correlation between traffic situations. This basically reflects that only very strong traffic jams propagate from the highway, trough a main road into a residential road. With a street change factor s^c with $s \geq 1$

and $c \equiv numberOfStreetChanges$ this model can be parametrised as follows:

$$d_p(x_0, x_1) = \sum_{i=1}^{n} d(e_{sp}) * s^c$$

Since published work shows a noticeable impact of the propagation speed [12] the distance d_p has to be considered in the interpolation. Therefore, we extend

$$z^*(x_0, T) = \sum_{i=1}^{n} \lambda_i(x_0) \cdot z(x_i, (T - t_d))$$

with a propagation speed v_p and a given datetime T and a time offset $t_d = \frac{d_p(x_0, x_1, g_i, c_f)}{v_p}$.

4. Shortest Path Adaptation of Dynamic Geospatial Data

This section explains the technical approach for getting a efficient searchable infrastructure model for distance computation without the full restriction to an infrastructure graph. It uses infrastructure data from the city of Aarhus as an visualisation example. Figure 2 shows a Map projection of the geospatial street information. Every street segment shows an arrow at the end of the Geometry that is stored as a LINESTRING (set of coordinates that build a path). If the streets that can be used *Bi-Directional*, this Arrow is just relevant for technical purposes. For the red and blue plotted one-way streets and unidirectional lanes of bigger roads the arrow shows the direction the street or lane may be driven.

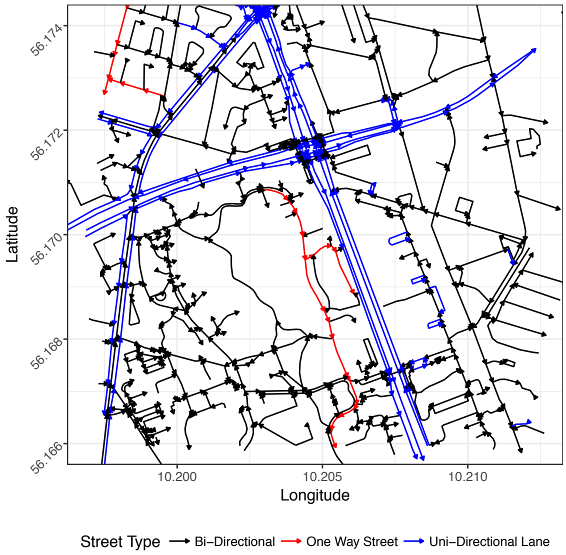


Figure 2: Raw Map Data

Figure 3 shows the directed Graph representation of the connected street network. It contains a quickly searchable data structure of connected Geometries and stores the length of the individual LINESTRING geometries. It hides the information where the streets are located and contains only the information about the connections between individual street segments as well as the information about the cost it takes to use the street segment (e.g., the length of the segment, a multiplicator for the maximum speed or a multiplicator for the trafficability depending on the street type).

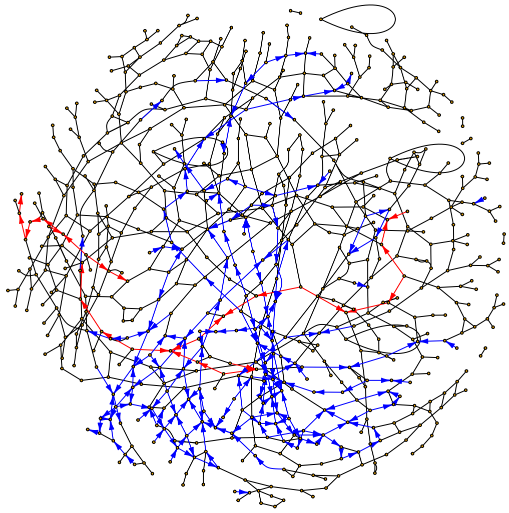
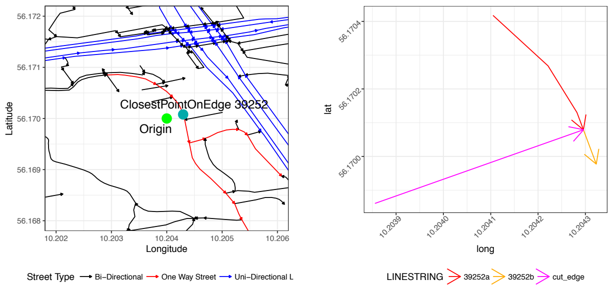


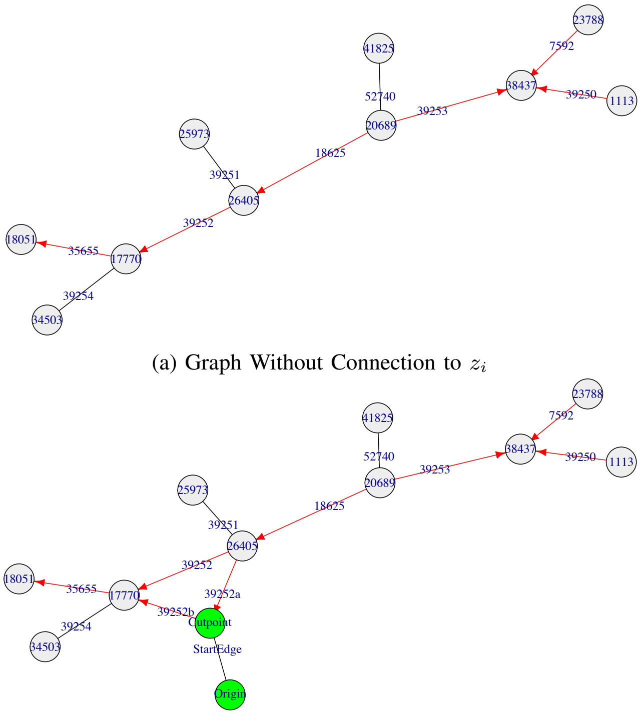
Figure 3: Directed Graph, Fruchterman-Reingold Visualisation

The advantage of a fast path search is owed to a vast simplification of the location model. The graph only contains fixed nodes with a distinct location. The graph has to be adapted if a new sensor is added at an individual location or an interpolation for a coordinate that is not covered with a node is requested. Figure 4a shows a location x_0, which is not located on the street grid (which is a usual frequent case, since a random point practically never is located exactly at one of the edges). To get a connection we identify the nearest location at the nearest LINESTRING (see Figure 4b).



(a) Closest Point to Origin (b) Division of Closest Edge

Figure 4: Connecting Origin to Closest Point of the Street Network



(a) Graph Without Connection to z_i

(b) Added $Origin$ node for z_i and $Cutpoint$ node to devide edge 39252. Added edges to substitute Edge 39252

Figure 5: Adding the Edges to the Graph

After the geospatial calculation, temporary edges are added to the infrastructure graph g_i (see Figure 5) and allow the shortest-path calculation between individual locations covered the infrastructure area.

Listing 4 shows the pseudo code of the distance calculations of an estimated value z^*_{xi} considering the distance function $d_p(x_0, x_i, g_i, c_f)$.

The distance list, resulting out of Algorithm 1 is used inside of the adapted IDW-Method. Therefore, d_e is directly substituted with d_p.

5. Results

To validate the importance of infrastructure-based distance metrics for interpolation, the following subsections illustrate the results in two steps. Section 5.1 shows the deviation between Euclidean distance and shortest path approaches. Section 5.2 shows correlation between sensor nodes in a network based on different distance metrics. Section 5.3 gives a link to the sourcecode and datasets, which were used for these experiments.

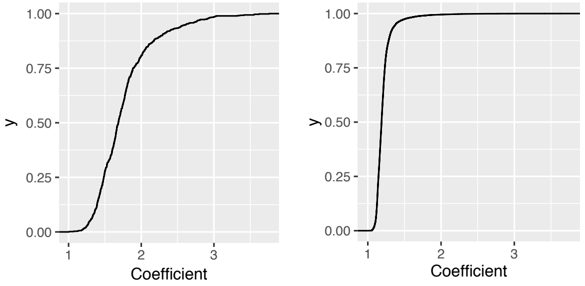
5.1. Deviation of Actual Distances in Road Propagation

To evaluate the error, which comes with the simplified distance calculation, the previously selected scenario origin (see Figure 4a) is evaluated against all graph node location of the directed graph. Therefore, the Euclidean distance d_e and the shortest path distance d_p are calculated between the newly attached origin node and all existing nodes in the graph. To testify the error, a distance coefficient $o_d = \frac{o_p}{o_e}$ is calculated for every pair of distances. Since d_p is the the sum a set of euclidean distances of the path elements and a single line is the shortest connection between two locations we get $o_d \geq 1$.

Figure 6a shows the cummulative distribution function (CDF) of o_d for the 661 nodes in the example area. Figure 6a shows the CDF of o_d for the whole Aarhus area (41757 nodes). Figure 7 shows o_d at x_b of the distance calculations $d(origin, x_b)$. Due to the one-way roads, which are connected to the origin location, quite high coefficients of $o_d > 3$ are calculated. This signifies that by taking the restricted infrastructure into account, the distance of the shortest path is more than 3 times the Euclidean distance. In Figure 7b the coefficient for nodes in the whole city area is plotted. There are some areas with restricted or wide spread roads where $o_d > 3$. The arithmetic mean $\overline{o_d} = 1.210565$ and the standard deviation $\sigma = 0,128$. These results show that the mean deviation between the analysed distance metrics is at around 21% but in certain areas there are exceptional high coefficients, which can easily lead to misinterpretation of data.

Algorithm 1: Bridge from Geospatial Data into Routing in a Directed Graph

```
Data: $origin$, $destination$, $spatialStreetData$,
      $directedGraph$,$costFunction$
Result: Distance / Shortest way
load dataset;
/* Connect $origin$ ($x0$) and $destinations$ (
   $x_i, i = 1 \ldots n$) geometry to $directedGraph$ */
edgelist ← new list() ;
for $p_i$ in origin,destinations do
    node $n_i$ ← CreateNode ($p_i$) ;            // for
    unconnected location
    search geometry $g_c$ with closest point $p_c$ to $p_i$ on
    $spatialStreetData$;
    node $n_c$ ← CreateNode ($p_c$);
    divide $g_c$ at $p_c$ result in $g_{c_a}$ , $g_{c_b}$ ;
    $l_a$ ← Length($g_{c_a}$) ;
    $l_b$ ← Length($g_{c_b}$);
    edge $e_a$ ← CreateEdge (node $n_i$ to node $n_c$ );
    edge $e_a$ ← CreateEdge ($n_c$ to $n_a$ with weight $l_a$);
    edge $e_b$ ← CreateEdge ($n_c$ to $n_b$ with weight $l_b$);
    edge $e_i$ ← CreateEdge ($n_i$ to $n_c$ with weight $l_o$);
    temporarily add edges $e_a$, $e_b$, $e_i$ to $graph$;
    edgeslist.add($e_i$)
end
distancelist ← new list() ;
for $e_i$ in 2:length(edges) do
    sp ← ShortestPathAlg(e1,ei,
    directedGraph,costFunction);
    distancelist.add(Length (sp));
end
Return(distancelist)
```



(a) CDF of Coefficients for the small example area (b) CDF of Coefficients for the city area

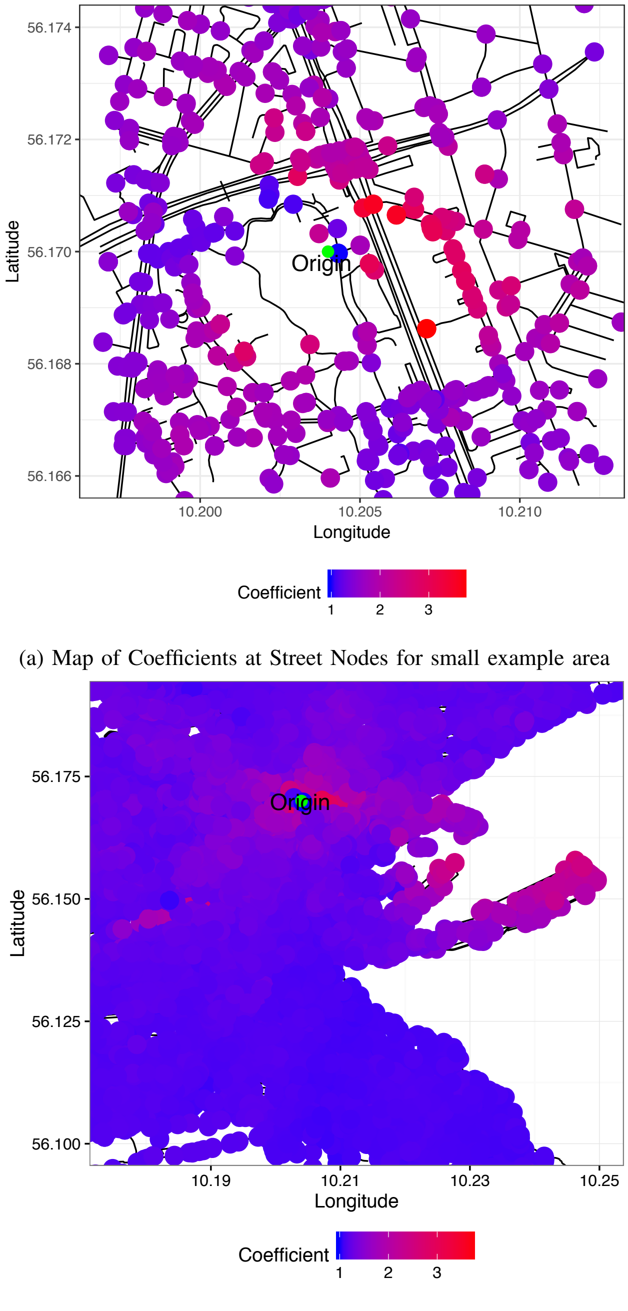
Figure 6: Visualisation of Shortest Path Distance / Euclidean Distance Coefficient o_d (from Origin Point)

5.2. Correlation of Sensors Based on Different Distance Metrics

To verify the significance of infrastructure-adapted distance metrics for interpolation, the correlation between traffic sensors was analysed. For the evaluation of individual distance metrics a traffic dataset covering 12 months of traffic data in Aarhus, Denmark was used. A time series (vehicle count and average speed) of 449 traffic sensors was pairwise compared using the Pearson correlation. The correlation coefficients have been calculated for each combination and every week in the given period.

$euclidean\_distance$: Euclidean distance between two sensors in metres.
$shortest\_path\_distance$: Route distance (shortest path on roads) between two sensors in metres.
$route\_duration$: Travel time by car of the route between two sensors.
$route\_steps$: Number of steps (road crossover, turns, etc.) of the route between two sensors.
$directed\_duration$: Additive composition of the route distance and the angle between two sensors measurements (to prefer sensors that measure the same direction).

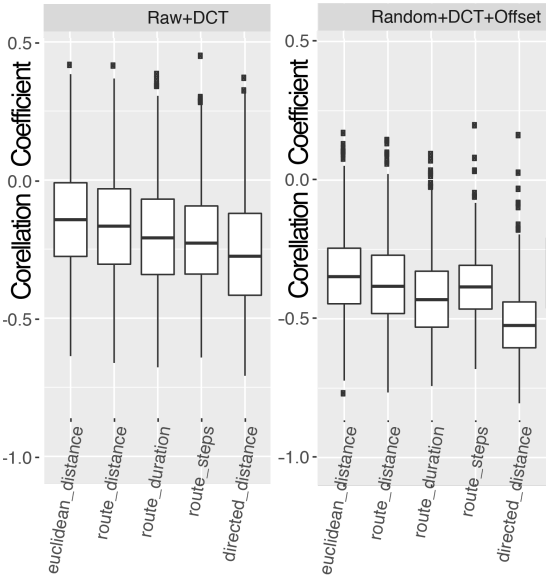
Figure 8 shows the resulting correlation values of the experiment. For every metric, a set of 201152 values was visualised in a box plot (showing the 25th and 75th percentiles, whiskers end at $1.5 * IQR$). The correlation values are negative since the similarity between traffic sensor time series increases with shorter distances. It depicts the different correlations with the consideration of the full data set in Figure 8a. In Figure 8b the time offset shifts between different time series (t_d) were used to enable the modelling of propagation speed between sensors (traffic moving on route). Before calculating the Pearson correlation, the compared time series were shifted according to the estimated time difference needed for the propagation of the traffic (Calculated by a routing algorithm). The results show that using infrastructure knowledge shows much higher correlations than the utilisation of simple Euclidean distances. Therefore, it is a key parameter that should be used in mod-



(a) Map of Coefficients at Street Nodes for small example area

(b) Map of Coefficients at Street Nodes for city area

Figure 7: Visualisation of Shortest Path Distance / Euclidean Distance Coefficient (from Origin Point)



(a) Correlation Results (b) With Time Offset Shift

Figure 8: Correlations for Different Metrics and Pre-Processing

ern machine learning, validation and interpolation scenarios.

5.3. Implementation and Data

The source code that was used to generate the figures of this publication can be found in the following github repository:
https://github.com/dkuemper/Infrastructure-Based_Interpolation
The open infrastructure data set originates from OpenStreetMap and can be downloaded via:
https://mapzen.com/data/metro-extracts/
Traffic data sets for correlation calculation can be downloaded from
http://www.ict-citypulse.eu/page/content/tools-and-data-sets (last time checked: 2016-02-16).

6. Conclusion

The spatial infrastructure exhibits a high impact on the interdependency of sensors. While the temperature will be similar in the neighbourhood, noise propagation will depend on shielding buildings and traffic flows depend on road networks, on-going construction work, traffic density etc. Hence, spatial reasoning requires appropriate distance measures that are based on the adapted propagation model. The use of the Euclidean distance between two locations is suited, for example, for events affecting nearby entities or persons. However, applied in a complex city environment this metric does not reflect the relevance of nearby events. A combined metric that utilises infrastructure knowledge, e.g. road networks, shows much better correlation. Consideration of event propagation and the correction by appropriate offset-times improves the correlation between different streams. In conclusion, the suggested algorithm provides methods to cope interpolation analysis for heterogeneous data sources in smart city applications.

Acknowledgment

This work has been supported by the European Union 7th FP Project CityPulse under grant agreement n. 609035

References

[1] T. Mueller, N. Pusuluri, K. Mathias, P. Cornelius, R. Barnhisel, and S. Shearer, “Map quality for ordinary kriging and inverse distance weighted interpolation,” *Soil Science Society of America Journal*, vol. 68, no. 6, pp. 2042–2047, 2004.

[2] S. M. Pingale, D. Khare, M. K. Jat, and J. Adamowski, “Spatial and temporal trends of mean and extreme rainfall and temperature for the 33 urban centers of the arid and semi-arid state of rajasthan, india,” *Atmospheric Research*, vol. 138, pp. 73–90, 2014.

[3] Y. Ramos, B. St-Onge, J.-P. Blanchet, and A. Smargiassi, “Spatio-temporal models to estimate daily concentrations of fine particulate matter in montreal: Kriging with external drift and inverse distance-weighted approaches,” *Journal of Exposure Science and Environmental Epidemiology*, 2015.

[4] H. Zou, Y. Yue, Q. Li, and A. G. Yeh, “An improved distance metric for the interpolation of link-based traffic data using kriging: a case study of a large-scale urban road network,” *International Journal of Geographical Information Science*, vol. 26, no. 4, pp. 667–689, 2012.

[5] C. S. Pitombo, A. R. Salgueiro, A. S. G. da Costa, and C. A. Isler, “A two-step method for mode choice estimation with socioeconomic and spatial information,” *Spatial Statistics*, vol. 11, pp. 45–64, 2015.

[6] W. Gossel and M. Falkenhagen, *Line-Geometry-Based Inverse Distance Weighted Interpolation (L-IDW): Geoscientific Case Studies*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 333–337.

[7] A. Artikis, M. Weidlich, F. Schnitzler, I. Boutsis, T. Liebig, N. Piatkowski, C. Bockermann, K. Morik, V. Kalogeraki, J. Marecek *et al.*, “Heterogeneous stream processing and crowdsourcing for urban traffic management.” in *EDBT*, vol. 14, 2014, pp. 712–723.

[8] N. Bissmeyer, S. Mauthofer, K. M. Bayarou, and F. Kargl, “Assessment of node trustworthiness in vanets using data plausibility checks with particle filters,” in *Vehicular Networking Conference (VNC), 2012 IEEE*, Nov 2012, pp. 78–85.

[9] N. Bissmeyer, J. Njeukam, J. Petit, and K. M. Bayarou, “Central misbehavior evaluation for vanets based on mobility data plausibility,” in *Proceedings of the Ninth ACM International Workshop on Vehicular Inter-networking, Systems, and Applications*, ser. VANET ’12. New York, NY, USA: ACM, 2012, pp. 73–82.

[10] R. Zafarani and H. Liu, “Evaluation without ground truth in social media research,” *Communications of the ACM*, vol. 58, no. 6, pp. 54–60, 2015.

[11] R. Toenjes, D. Kuemper, and M. Fischer, “Knowledge-based spatial reasoning for iot-enabled smart city applications,” in *2015 IEEE International Conference on Data Science and Data Intensive Systems*. IEEE, 2015, pp. 736–737.

[12] D. Kuemper, M. Fischer, T. Iggena, R. Toenjes, and P. Elke, “Knowledge-based spatial reasoning for iot-enabled smart city applications,” in *2016 IEEE 3rd World Forum on Internet of Things (WF-IoT) - IoT Smart Cities*. IEEE, 2016.

[13] D. Shepard, “A two-dimensional interpolation function for irregularly-spaced data,” in *Proceedings of the 1968 23rd ACM national conference*. ACM, 1968, pp. 517–524.