3D Graph Reconstruction from 2D Graphs Projections and Univariate Positional Traces in 3-Space
With Application to 3D Reconstruction of Road's Interchanges

Nir Hershko and Gershon Elber
Computer Science Department, Technion – Israel Institute of Technology, Haifa 32000, Israel

Keywords: 3D Reconstruction, Road Map, Interchanges, Global Positioning System.

Abstract: In this work, we consider a reconstruction problem of a 3D graph, based on its 2D projected model and imprecise 3D positional traces along it, and propose a general method to achieve this reconstruction. Then, we examine a specific application of this problem — 3D reconstruction of road's interchanges from 2D maps and GPS traces. We demonstrate the algorithm and show that the implementation of this method yields a robust and accurate solution compared to real ground truth data.

1 INTRODUCTION

This work presents a method to reconstruct 3D graphs. A 3D graph is a graph in which every node is assigned a position in 3D – as seen in the example in Figure 1(a). The inputs to the proposed reconstruction method are shown in Figure 1(b):

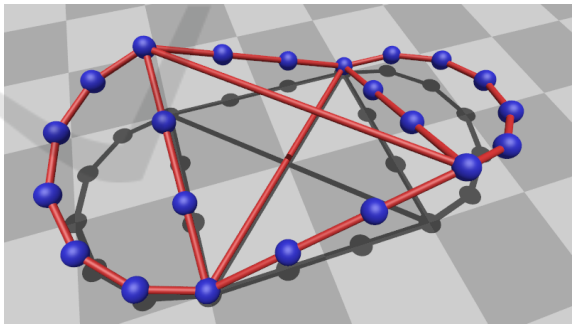
1. A 2D orthographic bijective projection of nodes of the graph – A graph with the same topology, but with nodes' positions in 2D. This 2D graph can be self-intersecting.
2. A set of imprecise 3D positional univariate traces along the graph's edges.

Our goal is to estimate the third coordinate of each node's position — which will henceforth be referred to as an *elevation*. We discuss two variants of the elevation method, both based on registering the traces to the 2D graph:

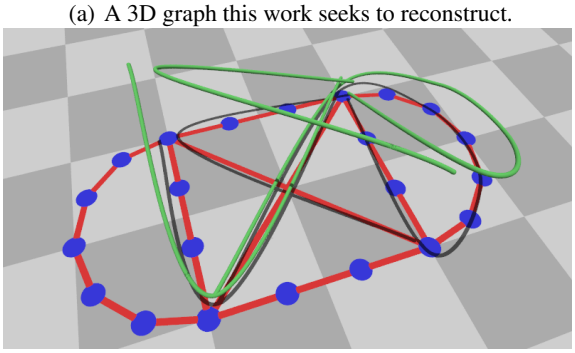
The absolute average variant. (AAV) averages the elevations of the traces at every node.

The relative average variant. (RAV) deduces the elevations using a first-order differences of the traces' elevations.

We argue that the RAV provides results that can be more robust and precise, under certain conditions. We also present a method to reconstruct the 3D graph even when some of it's nodes are not covered with any 3D trace. This is achieved based on some additional assumptions on the structure of the 3D graph.



(a) A 3D graph this work seeks to reconstruct.



(b) Inputs: a 2D projection of the 3D graph and a set of imprecise 3D traces.

Figure 1: An example of output and inputs for our method.

One clear motivation to the proposed method is the need for reconstruction of road's interchanges in 3D. An existing 2D road map serves as a 2D projection of the desired 3D road graph. GPS traces recorded while driving through the interchanges provide the 3D traces of the graph.

Hershko N. and Elber G..
3D Graph Reconstruction from 2D Graphs Projections and Univariate Positional Traces in 3-Space - With Application to 3D Reconstruction of Road's Interchanges.
DOI: 10.5220/0004652500130021
In *Proceedings of the 9th International Conference on Computer Graphics Theory and Applications* (GRAPP-2014), pages 13-21
ISBN: 978-989-758-002-4

The rest of the paper is organized as follows. Section 2 discusses related work. Section 3 explains the reconstruction problem in more details and Section 4 describes our proposed reconstruction method. Section 5 outlines the application of the method to reconstruction of road's interchanges from GPSs. In Section 6, we present results and verification of the implementation against real ground truth data, and finally, Section 7 discusses some future work and concludes.

2 RELATED WORK

There are quite a few examples for 3D graphs reconstruction from imagery data. In the medical field, Coste et al. presented an algorithm to reconstruct a 3D blood vessels network from several 2D projections, acquired from angiographic imaging (Coste et al., 1999). In the field of GeoInformatics, (Chen et al., 2006) and others presented methods to reconstruct 3D road models using LIDAR (LIght Detection And Ranging) data captured airborne.

Reconstruction of graphs from traces was studied primarily in the context of GeoInformatics — a result of the prevalence of low-cost GPS receivers, making it easy to crowd-source positional traces (Heipke, 2010). While most works focus on reconstruction of 2D road networks (Cao and Krumm, 2009; Chen et al., 2010), it has been demonstrated that GPS traces can also be used to reconstruct a 3D road network (Guo et al., 2007). However, the main problem of (Guo et al., 2007) in reconstruction of road networks is that it results in many cases with a topologically-inaccurate graphs that need to be manually corrected (Fathi and Krumm, 2010).

Another general approach for the reconstruction of 3D graphs from traces considers the input traces as a graph embedded in a metric space – a "metric graph" (Kuchment, 2004). The metric graph is simplified using a method suggested, for example, by (Aanjaneya et al., 2012), and the simplified graph is then re-embedded in $\mathbb{R}^3$. It should be noted that the conversion of traces to a (single connected) metric graph is not natural, and this approach disregards the connectivity of the points in the traces. Also, the method presented by Aanjaneya et al. results with a graph that is guaranteed to be topologically-correct only for a certain limit of positional error.

3 PROBLEM DESCRIPTION

Let $G = (N, E, p)$ be a 3D (possibly directed) graph, comprising of a set of nodes N, a set of (possibly directed) edges E, and position mapping $p : N \to \mathbb{R}^3$. We will regard the edges as the linear segment between the corresponding nodes' positions. The graph's 2D projection is denoted $\tilde{G} = (N, E, \tilde{p})$, with $\tilde{p} : N \to \mathbb{R}^2$. The unknown vertical component of each node, N_k, will be referred to as the node's *elevation* and denoted N_k^e. In the rest of this work, we will use k, l and m as indices of nodes.

A *trajectory* p_i along the 3D graph is a continuous arc-length piecewise-linear parameteric curve $p_i(s) : [0, L_i] \to \mathbb{R}^3$ of length $L_i \in \mathbb{R}$ such that $p_i(s)$ is always on a graph's edge, and its derivative $p_i'(s)$, if exists, adheres to the edge's direction. A *trace* T_i over the 3D graph G is a piecewise-linear sampling $T_i = \{t_{i,j}\}$ such that $t_{i,j} = p_i(s_{i,j}) + d_i(s_{i,j}) \in \mathbb{R}^3$ for some trajectory $p_i(s) : [0, L_i] \to \mathbb{R}^3$ on the graph and some error function $d_i(s) : [0, L_i] \to \mathbb{R}^3$ that is bounded and Lipschitz continuous. In the rest of this work, we will use i as an index of a trace and j as an index of a point in a trace. The word "node" (or "graph node") will be exclusively used for elements in N, whereas the points along some trace will be referred to as "trace points".

Figure 2 presents an example of the expected behaviour of the positional traces, in the 'z' axis. As can be seen from the figure, while the variance of the positions is quite large, the differences in the change

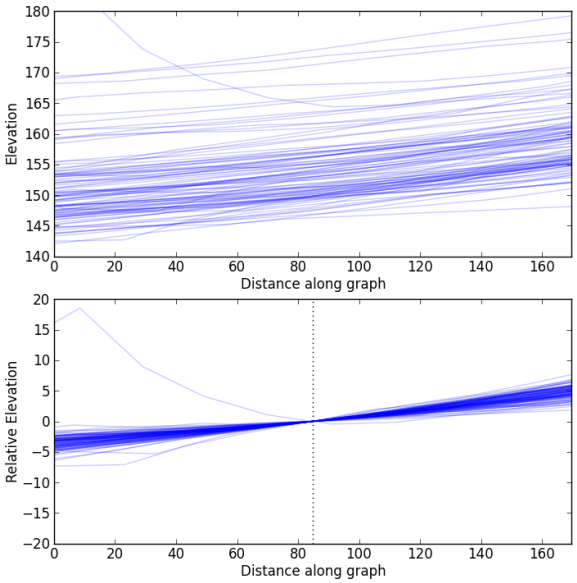


Figure 2: Elevation profiles of a set of imprecise traces, tracing the same route in the graph (top) and the same traces, each coerced to zero at the center, exposing the relative differences in elevations (bottom).

of elevation are quite small. Given the graph's 2D projection $\tilde{G}$ and a set of such traces $T = \{T_i\}$, in the next section we present our approach to reconstruct the graph in 3D — namely, assign each node its elevation.

4 THE METHOD

Assuming the traces' error functions d_i are small enough, we will be able to register each trace to the route that represents it on the 2D graph, and use averages of the traces' elevations as an estimator to the 3D graph position (The AAV approach). However, if the error d_i drifts globally, the entire traces might shift, while the derivatives will remain quite accurate. In this case, one rather use the traces' derivatives as an estimate to the corresponding edges' derivative, and globally resolve them over the graph to estimate the elevation (the RAV approach).

Hence, the method we propose to resolve the 3D reconstrcution problem includes the following steps:

1. Registration of the 3D traces to the 2D graph.

2.a. (AAV) Averaging the elevations of the nodes of the 2D graph across all the traces, or ;

2.b. (RAV) Averaging the *elevations' differences* on the edges across the traces, and solving for the absolute elevations using some known boundary conditions.

Figure 3 compares the AAV and RAV in a typical reconstruction case for graphs: two of the three traces are partial (a result of splitting or merging with other paths in the graph), and they don't cover all this path. While the AAV results with a disontinuity, the RAV results with a smooth solution. Figure 4 provides an overview of the different steps in our method, and the different steps are now discussed in detail.

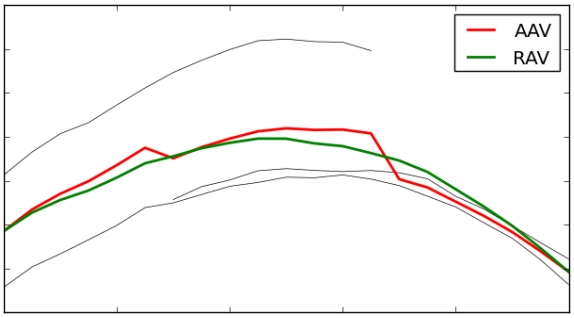


Figure 3: Comparison of the two reconstruction variants on a synthetic example. The original input traces are shown in black.

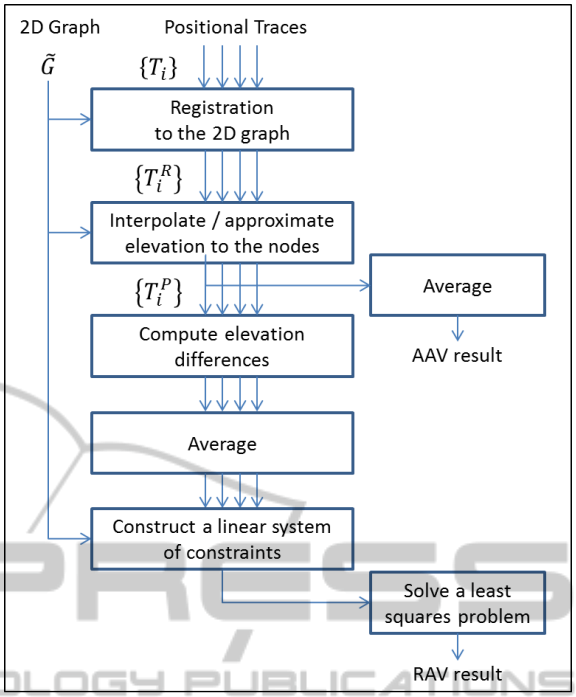


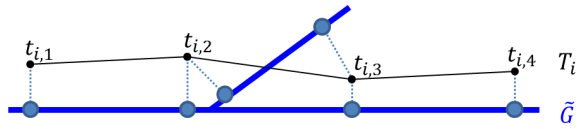
Figure 4: Overview of our method.

4.1 Registration

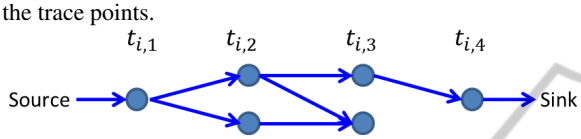
The registration of the traces correlates every trace T_i with the 2D graph $\tilde{G}$, resulting with the registered trace $T_i^R = \{t_{i,j}^R\}$, where every point on the trace, $t_{i,j}^R$, contains the corresponding position on the graph $\tilde{G}$ and the edges in the graph to the position of the next point of the trace, $t_{i,j+1}^R$. The registered trace's *path* is the graph nodes along the trace. It should be noted that the input trace T_i is of the same length as its graph-registered counterpart T_i^R. The registration can be done in various methods that consider the traces' positions and derivatives in relation to the 2D graph, account for the graph's topology, or use other domain knowledge or information that is available in a specific application.

In this work, we use a simple Map-Matching method from the field of Geographic Information Systems, known as *Weight-based Map Matching* (Yin and Wolfson, 2004). For every point $t_{i,j}$ along the trace, we find all possible candidate matched points on the graph, and weigh them based on the distance from the trace. We then build a candidate graph: The nodes are the set of all candidate points, and the edges are the set of shortest paths between pairs of adjacent candidate points, directed as in the trace. An example can be seen in Figure 5. A least-cost path on the graph, calculated using a Dijkstra's algorithm, corresponds to a valid matching of the trace with a minimal total distance from the graph.

For more information about the various Map-Matching methods, (Yuan et al., 2010) provide a recent overview, together with their own approach.



(a) The 2D graph $\tilde{G}$, a trace T_i and the candidate points for the trace points.



(b) Candidate graph.

Figure 5: Construction of a candidate graph for a trace.

4.2 Absolute Average Variant (AAV)

In this reconstruction variant, every node's elevation is calculated as a simple average of the elevations of all traces through it. First, every registered trace T_i^R is used to calculate the elevations of the graph nodes in the trace's path, using some interpolation or approximation scheme, resulting with $T_i^P = \{t_{i,j}^P\}$, such that $N[t_{i,j}^P]$ is the corresponding node in the graph and $e[t_{i,j}^P] \in \mathbb{R}$ is the node's interpolated/approximated elevation according to *this* trace. It should be noted that T_i^P is usually *not* of the same length as the registered trace T_i^R that it is based upon. Our implementation is using a linear interpolation of the elevations on the distance along the trace's path in the graph, as is shown in Figure 6.

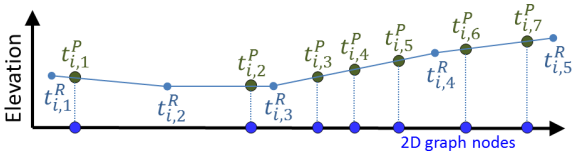


Figure 6: Constructing T_i^P using linear interpolation of the elevation of the registered trace T_i^R on nodes along its path. Note that this is done for every trace T_i, so each node is usually assigned with more than one elevation – from different traces.

After calculating the elevations of the nodes along the traces, $e[t_{i,j}^P]$, we simply average the elevations of each node in the graph. For the sake of brevity, we will use $O(k)$ to denote the set of all the elevated nodes (based on the different traces) corresponding to the node N_k:

$$O(k) = \left\{ t_{i,j}^P \;:\; N[t_{i,j}^P] = N_k \right\} . \tag{1}$$

In the AAV, the final elevation of the node N_k will then be:

$$AAV_k = \frac{1}{\|O(k)\|} \sum_{t_{i,j}^P \in O(k)} e[t_{i,j}^P] . \tag{2}$$

4.3 Relative Average Variant (RAV)

In this reconstruction variant, the interpolated traces $\{T_i^P\}$ are calculated as in the AAV, but are used only for the first-order elevation differences between adjacent nodes. The elevation differences are averaged and then globally resolved in the least-squares sense, over an over-constrained linear system, in which the unknowns are the nodes' elevations $\{N_k^e\}$. Let $O(k,l)$ be the set of all pairs of adjacent elevated nodes on a trace:

$$O(k,l) = \left\{ (t_{i,j}^P,\, t_{i,j+1}^P) \in O(k) \times O(l) \right\} . \tag{3}$$

Each such averaged elevation difference over a graph's edge is expressed in the linear system as the constraint:

$$N_k^e - N_l^e = \frac{1}{\|O(k,l)\|} \sum_{\substack{(t_{i,j}^P,\, t_{i,j+1}^P) \\ \in O(k,l)}} e[t_{i,j}^P] - e[t_{i,j+1}^P] . \tag{4}$$

In additions to the relative elevations, some absolute elevation are also prescribed as boundary conditions.

$$N_k^e = BoundaryElevation_k . \tag{5}$$

A different weight could be assigned to some constraints, representing our confidence in these elevations' accuracy.

The RAV also allows us to assign elevation to nodes that are not covered by any trace. Under the assumption that the graph's elevations tend to change smoothly and gradually, one can assign a zero-elevation-difference contraints to all edges that have no elevation difference information:

$$N_k^e - N_l^e = 0 \quad \text{if} \quad O(k,l) = \emptyset . \tag{6}$$

Let N_k, N_l, and N_m be three consecutive nodes in the graph, such that N_l is connected only to N_k and N_m, and there is no trace that traverses any of the edges N_kN_l and N_lN_m. To obtain a linear elevation change over these edges, we assign a weight of $1/\sqrt{d(k,l)}$ to the constraint in Equation (6), where $d(k,l)$ is the xy length of the graph edge N_kN_l. As no other constraint depends on N_l^e, minimizing the energy of the two weighted constraints

$$E = \left(\frac{N_k^e - N_l^e}{\sqrt{d(k,l)}} \right)^2 + \left(\frac{N_l^e - N_m^e}{\sqrt{d(l,m)}} \right)^2 , \tag{7}$$

with respect to N_l^e (i.e. differentiating E with respect to N_l^e and equating to zero), results with

$$N_l^e = \frac{N_n^e d(k,l) + N_k^e d(l,m)}{d(k,l) + d(l,m)} , \quad (8)$$

which describes the linear dependency of N_l^e on its two neighbors. Repeating the process on pairs of edges along an "uncovered" path explains the linear change of the elevation between the two terminal nodes.

The system of linear constraints formed out of (4), (5) and (6) is solved in the least-squares sense. As this is a sparse system, a sparse least-squares solver such as LSQR (Paige and Saunders, 1982) can be employed.

5 APPLICATION TO ROAD'S INTERCHANGES

This section describes the implementation of the presented method in the specific application of reconstruction of road's interchanges in 3D from a 2D road map and GPS traces.

We used data from the OpenStreetMap project at http://www.openstreetmap.org/ (Haklay and Weber, 2008) – for both the tagged 2D road map and GPS traces uploaded by the users.

5.1 Input processing

GPS traces have the property of a global drifting error as mentioned in the problem description (with time as the curve parameter). In fact, Figure 2 presents GPS traces – it consists of GPS traces recorded using the same device and in the same place, over the course of several months. However, during the recording, the user can change his/her speed and can even come to a stop. As the position of the GPS trace continues to drift, it results with a significant error over a short distance. This can be seen in Figure 7. Our proposed remedy is to remove and split the traces where the speed is too low – making the error as a function of the distance small – satisfying the problem's requirements. One should note that in practice this difficulty is not manifested in interchanges as cars usually drive at some minimal speed.

We pre-process the 2D road map as well. Long straight roads are usually represented in OpenStreetMap as a single segment (consisting of the two nodes at the extremes). However, in 3D it is often the case that the road is not straight but has some vertical alterations. To allows the reconstruction to express these higher frequency details, we refine all

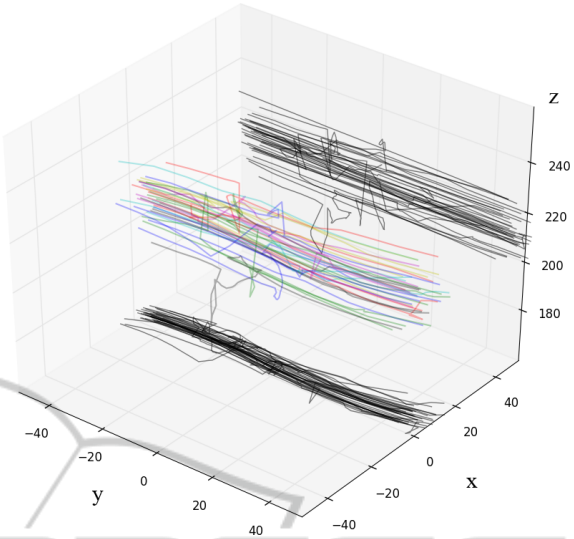


Figure 7: Several GPS traces recorded with the vehicle coming to a stop (before a traffic light), and their projection on the xy and yz planes. Note the large variance in z compared to xy.

roads so the maximum segment is of bounded length – as demonstrated in Figure 8.

Figure 8: Side-view of a long road segment (bottom), and its refinement (top) revealing some elevation curvature.

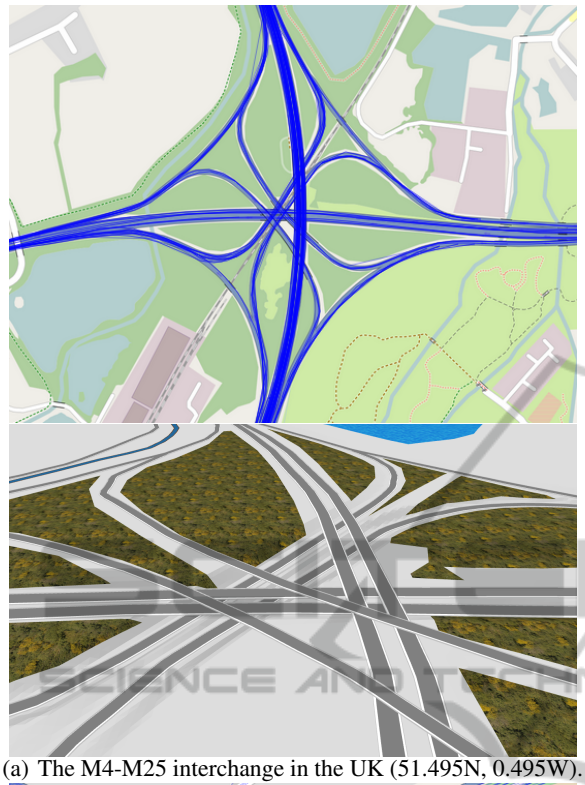
5.2 Boundary Conditions

As boundary conditions in the RAV, we use the circumference of the approximated region – where the calculated road network is stitched to the rest of the world. We seek to process (i.e. render) the algorithm's result together with parts of the road's network that don't have an approximated elevation. Hence, all roads that enter/leave the region under approximation, will be set with a boundary condition that will force the stitching to the surroundings, to be continuous.

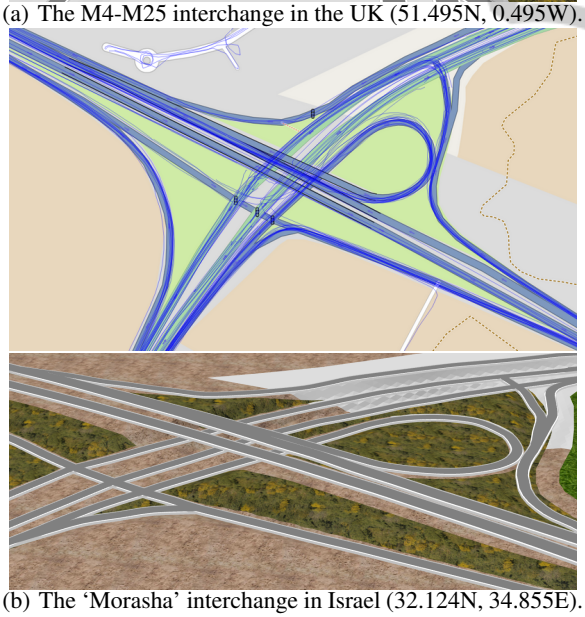
6 RESULTS AND VERIFICATION

Figure 9 presents a pair of results of the algorithm using maps and traces from the OSM project. The road network in these examples was refined based on a maximum segment length of 30 meters.

In order to validate our approach, we use the 'Mesubim' interchange in Israel, at (32.038N, 34.830E), for which ground-truth data is available. Figure 10 shows its map and a set of traces over it



(a) The M4-M25 interchange in the UK (51.495N, 0.495W).



(b) The 'Morasha' interchange in Israel (32.124N, 34.855E).

Figure 9: The result of applying the RAV reconstruction algorithm on some interchanges. Shown are the traces (top) and the 3D reconstruction (bottom) of both interchanges.

and Figure 11 shows the result of applying the RAV algorithm to the interchange. This map consists of 601 nodes and 633 edges. 118 traces were processed into a linear system of 642 constraints, including 274 relative-elevations constraints (Equation (4)), 359 zero-elevation-difference constraints (Equation (6))

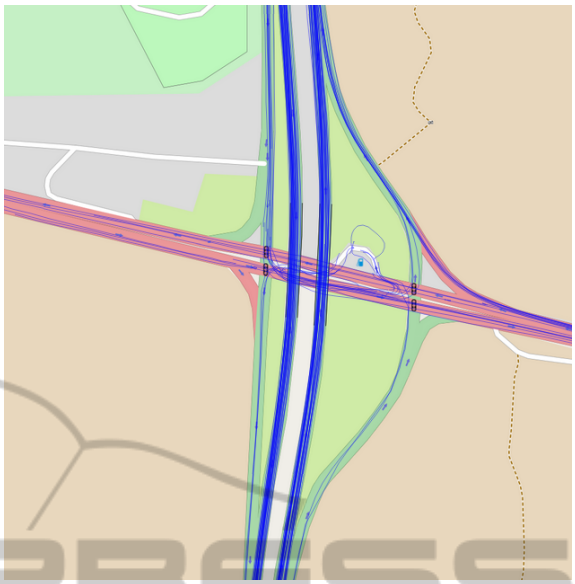


Figure 10: The OSM map and the available GPS traces at the 'Mesubim' interchange (32.038N, 34.830E).

Figure 11: The result of applying the RAV reconstruction algorithm on the interchange in Figure 10.

and 9 boundary conditions Equation ((5)). Matching the traces to the map took 6 seconds and solving the least-squares problem took half a second – both on a single 2.8GHz CPU core. We use 'ground truth' elevation data that is derived from a photogrammetric measurement and has an elevation accuracy of 0.3 to 0.5 meters, and horizontal sampling interval along the roads is in the range of 8 to 15 meters.

Since the OpenStreetMap 2D data does not perfectly match, in xy, to the ground truth, the roads' 2D locations were registered manually (with changes of up to 12 meters – perpendicular to the road direction) for the sake of this verification. The OSM roads at this interchange were then resampled at intervals of one meter, and the elevations resulting from our algorithms were compared to the elvations at those uniformly-sampled 'ground truth' points.

We compared both presented variants. Note that while the RAV provides elevation for roads that are

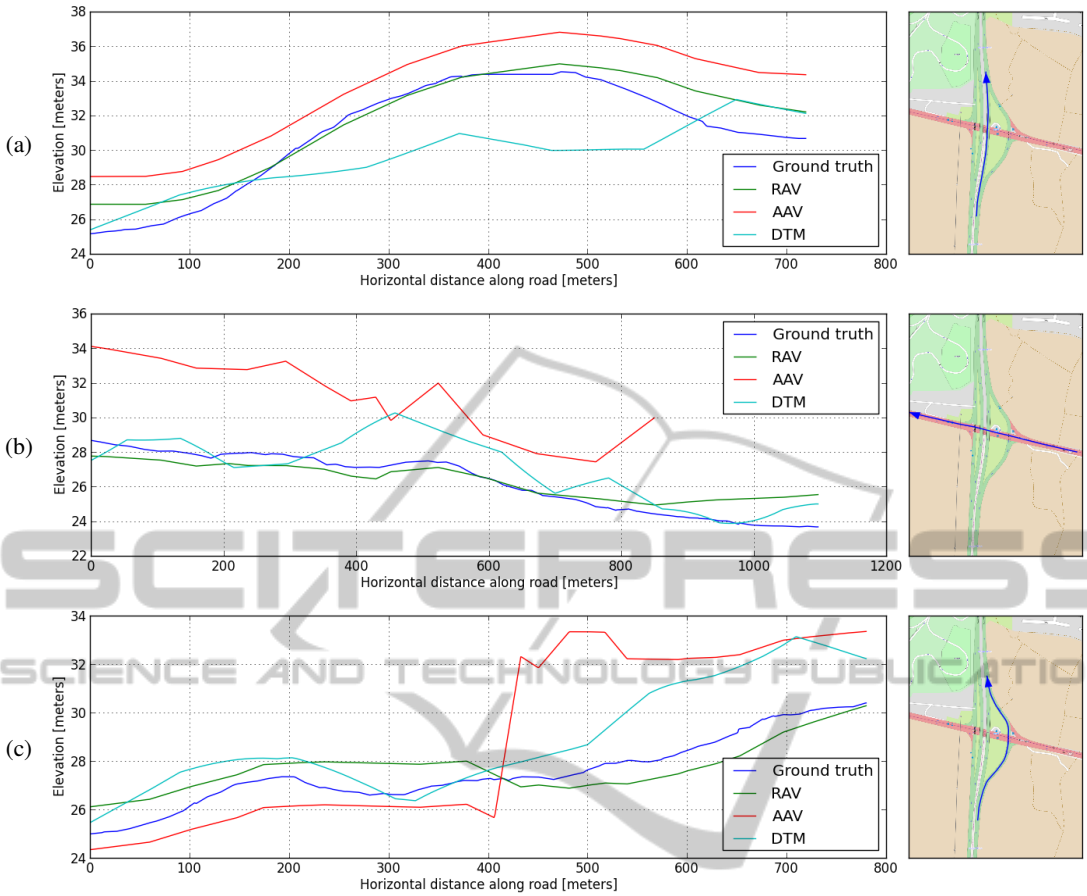


Figure 12: Plots of elevations along several roads of the interchange in Figure 10.

not covered with any trace, the AAV does not. Figure 12 presents several of the road sections at the interchange, with the elevations along them, including the ground truth, the results of both the AAV and the RAV, and elevations derived from a publicly available Digital Terrain Model (DTM), for comparison. From Figure 12, several points can be noted:

- Figure 12(a), which depict elevations on the bridge, have the least agreement between the ground truth and the DTM (a difference of 4 meters). It seems that due to the low resolution of the DTM, it only partially captures the bridge's elevation details. This can also be seen in Figure 12(b), which depicts the road under the bridge.
- Since the road in Figure 12(a) have the most GPS traces, and it is uninterrupted (as there are no traffic lights or other stops along those roads) both variants yield good agreements with the ground truth.
- Since Figures 12(b) and 12(c) have low and intermittent coverage, the AAV behaves poorly. Specifically, in Figure 12(c) at the 410m mark, we leave a road covered by only 2 traces, and go through several nodes that are covered by many different traces. However, the RAV continues to perform well in these cases, taking into account only elevation differences in traces, and not between traces.
- In the right side of Figure 12(b), it is visible that where the AAV has no information, the RAV falls back to a linear interpolation.
- The RAV was calculated for this intersection on a larger area than the provided ground truth. Therefore, the boundary conditions are not set exactly on the ends of Figures 12 (a) and (b). More so, the road shown in Figure 12 (c) is even farther away from the boundaries.

In Figure 13, we see a histogram of the differences between the ground truth and the results of applying both variants (in road segments where the AAV provides a result). These differences' aggregates are specified in Table 1. From these results, it is clear that the RAV yields elevations that are closer to the

ground truth than the AAV — both in terms of mean and variance of the difference.

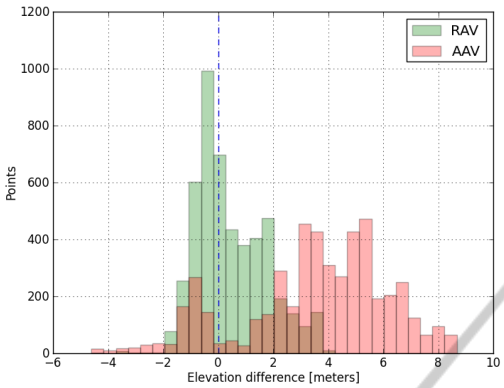


Figure 13: Histograms of the elevation difference between the tested method variants and the ground truth.

Table 1: A comparison of the different methods' results to the ground truth. "coverage" refers to all the points with at least one trace covering it.

Vatiant	Data	Mean	Variance
RAV	all points (6606)	+0.59	1.51
RAV	coverage (4923)	+0.47	1.63
AAV	coverage (4923)	+3.56	6.88

One should note that since the GPS receiver is typically not located at the ground level while recording, it is expected that the AAV will results with some offset in z above the ground truth, but this should not affect the variance of the comparison with the ground truth.

7 CONCLUSIONS AND FUTURE WORK

In this paper, we presented an approach for the reconstruction of 3D graphs from 3D positional traces and 2D graphs, and shown that this method can provide good results in the application of road interchanges.

We believe that the RAV method might also be applied to other applications such as medical reconstruction, cave mapping, or flight paths. In order to adapt this method to other applications, we suggest a variation to the problem as follows. Since we use in this method the graph's 2D positions only for the purpose of registering the traces, we could think of a variation of the problem in which the nodes' 2D positions are not known, and instead other information is available to allow us to register the traces to the graph, or the 2D positions are not accurate but can still be used for registration. In that case, instead of calculating the nodes' elevations, every coordinate of the nodes' positions is calculated separately and independently in the same manner.

For the application of 3D road network reconstruction from traces, we suggest a semi-automatic method to provide more topologically-accurate results than existing methods that reconstruct a road network from traces alone. With the evident success of OpenStreetMap and other commercial 2D maps, we believe that using our method on existing 2D datasets to ameliorate them to 3D can provide with results that are more robust, due to the high quality of these existing datasets. In general, the system would consist of the following steps:

1. Reconstruction of a 2D graph from the univariate traces using an existing 2D reconstruction method, such as (Guo et al., 2007; Cao and Krumm, 2009; Chen et al., 2010).
2. Augmentation of the 2D graph into 3D using the method presented in this work.

This proposed approach also allows one to make use of traces that have only 2D information in the first step, and to achieve reasonably good results with a small amount of 3D traces.

Finally, a more comprehensive data analysis on the GPS output signal might provide with an appropriate filtration and noise-removal method, which might improve the reconstruction quality for the interchanges application. Still, the basic property of a global trace error will still manifest in traces recorded by consumer devices, and thus the method presented in this paper or an equivalent method is still required.

ACKNOWLEDGEMENTS

This research was supported in part by the E. and J. Bishop Research Fund, Technion.

We would also like to thank *Armi Grinstein – Geodetic Engineering Ltd* (http://www.armig.co.il/) for kindly providing us with the ground truth data for the 'Mesubim' interchange.

REFERENCES

Aanjaneya, M., Chazal, F., Chen, D., Glisse, M., Guibas, L., and Morozov, D. (2012). Metric graph reconstruction from noisy data. *International Journal of Computational Geometry & Applications*, 22(04):305–325.

Cao, L. and Krumm, J. (2009). From gps traces to a routable road map. In *Proceedings of the 17th ACM SIGSPA-*

TIAL International Conference on Advances in Geographic Information Systems, pages 3–12. ACM.

Chen, D., Guibas, L. J., Hershberger, J., and Sun, J. (2010). Road network reconstruction for organizing paths. In *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1309–1320. Society for Industrial and Applied Mathematics.

Chen, L.-C., Lo, C.-Y., Shao, Y.-C., and Teo, T.-A. (2006). Automatic reconstruction of 3d road models by using 2d road maps and airborne lidar data. In *Proceedings of 27th Asian Conference on Remote Sensing (ACRS2006)*, pages 9–13.

Coste, E., Vasseur, C., and Rousseau, J. (1999). 3d reconstruction of the cerebral arterial network from stereotactic dsa. *Medical physics*, 26:1783.

Fathi, A. and Krumm, J. (2010). Detecting road intersections from gps traces. In *Geographic Information Science*, pages 56–69. Springer.

Guo, T., Iwamura, K., and Koga, M. (2007). Towards high accuracy road maps generation from massive gps traces data. In *Geoscience and Remote Sensing Symposium, 2007. IGARSS 2007. IEEE International*, pages 667–670. IEEE.

Haklay, M. and Weber, P. (2008). Openstreetmap: User-generated street maps. *Pervasive Computing, IEEE*, 7(4):12 –18.

Heipke, C. (2010). Crowdsourcing geospatial data. *ISPRS Journal of Photogrammetry and Remote Sensing*, 65(6):550–557.

Kuchment, P. (2004). Quantum graphs: I. some basic structures. *Waves in Random media*, 14(1):S107–S128.

Paige, C. C. and Saunders, M. A. (1982). Lsqr: An algorithm for sparse linear equations and sparse least squares. *ACM Trans. Math. Softw.*, 8(1):43–71.

Yin, H. and Wolfson, O. (2004). A weight-based map matching method in moving objects databases. In *Scientific and Statistical Database Management, 2004. Proceedings. 16th International Conference on*, pages 437–438. IEEE.

Yuan, J., Zheng, Y., Zhang, C., Xie, X., and Sun, G.-Z. (2010). An interactive-voting based map matching algorithm. In *Mobile Data Management (MDM), 2010 Eleventh International Conference on*, pages 43–52. IEEE.