Distance Transformations for Accessibility Mapping in the Public Transport Domain: A Performance Assessment

Matthias Müller[1], Tassilo Glander[2]

[1]TU Dresden, Dept. of Geosciences, Geoinformation Systems, 01062 Dresden, Germany
Email: matthias_mueller@tu-dresden.de

[2]Universität Potsdam, Hasso-Plattner-Institut, Computer Graphics Systems, 14482 Potsdam, Germany
Email:Tassilo.Glander@hpi.uni-potsdam.de

1. Introduction

Public transport is a crucial part of modern cities' infrastructures. Transport and road networks, schedule and city structure are interdependent factors that add to the overall performance of a public transport system (European Commission 2001). Being the primary determinant of transit use, accessibility in terms of travel time is subject to many research papers (Beimborn et al. 2003, Biba et al. 2010, Lei & Church 2010, Horner & Murray 2004) as well as basic literature in the transport and urban planning domain (Schnabel & Lohse 1997).

Distance transformations are a fundamental means to provide location-based travel time as an indicator for accessibility maps, but are also a prerequisite for more sophisticated methods in accessibility analysis. Corresponding algorithms and operators are implemented in most geoinformation systems (GIS), many image analysis libraries, as well as on dedicated hardware (De Smith et al. 2008). The choice of algorithm and computation platform clearly depends on the use case's requirements; usually the trade-off is between accuracy and computation speed.

In this work, we will conduct an experiment to assess accuracy and computation speed of two distance transformation algorithms, capturing the performance range of state-of-the-art distance transformations for application in public transport scenarios. The first one is a fast but possibly inaccurate modified Euclidean Distance transformation implemented on graphics hardware. The second is a slower but very accurate algorithm realised with the PCRaster library, incorporating road network and additional surface constraints. The findings will be used to prepare future work in applications for concurrent web-based accessibility mapping, city planning, property value assessment and real-time applications.

2. Foundations

In a one trip model with fixed origin and destination, distance is described by the complex travel time (**1**). It accounts for a door-to-door trip, including the possible use of private vehicles and public transport (Schnabel & Lohse 1997).

$$T_C = T_I + T_W + \sum T_D + \sum T_T + T_F + T_S \qquad (1)$$

T_C Complex travel time

T_I Initial walking time to the first stop

T_W Waiting time at first stop

$\sum T_D$ Sum of driving times in all participating vehicles

$\sum T_T$ Sum of all transit and waiting times for changing vehicles

T_F Final walking time from the last stop

T_S Search time for parking a car or bike, including handling times

In the following, complex travel time is discussed for a simplified case, assuming 1) a walking traveller who 2) has access to public transport. This is the usual scenario for nowadays web-based routing applications operated by public transport companies. The considered traveller knows the schedule, locations of public transport stops and does not spend additional search or handling times, thus allowing for a simplified computation of T_C (**2**). An example for this kind of person is a commuter in a major city.

$$T_C = T_P + T_N \tag{2}$$

T_P Walking time as a pedestrian

T_N Travelling time on the public transport network

$$T_C = k_D \cdot s \cdot \bar{v}^{-1} + T_N \tag{3}$$

As time t is commensurate to distance s at a given velocity v, the pedestrian's walking time can be obtained by computing the distance (**3**). GIS provide two fundamentally different concepts for distance computation in space: surface-based and network-based. In a surface-based computation, pedestrians are moving continuously through space, obeying certain constraints or costs that affect travelling speed or forbid access to some areas. The most common surface-based distance operator is the Euclidean Distance Transformation (EDT, Fabbri et al. 2008, De Smith et al. 2008). It computes the shortest distance between two points along a straight line, regardless of any obstacles. Fabbri et al. (2008) provide an overview and a performance comparison of current EDT algorithms. Due to the simple assumptions, EDT is one of the fastest distance operators. In reality, pedestrians adhere to paved ways or roads, thus an EDT computation will produce too optimistic walking times. To enhance the precision of simple estimators, a detour factor k_D is optionally applied to correct systematic distance underestimates.

Honouring the existence of natural barriers and uneven surfaces, many GIS provide Cost Distance Transformation (CDT) operators. These operators usually work on surfaces only and compute the shortest path based on local costs that effectively influence travelling speed. Some GIS (Eastman 2006) also offer CDT capabilities for anisotropic surfaces that respect directional costs which occur in an elevated terrain.

Network-based distance computation requires a topology describing navigable space, e.g., a road network. Barriers are modelled implicitly, as pedestrians can travel only along the network's edges. Paths through the network are computed between nodes, and the cost, e.g., time or distance, is based on the weight of the connecting edges. To obtain distances between points outside the network, some additional computation is required. In general, implementations either apply common interpolation algorithms from image analysis or rely on EDT for that task.

Network-based approaches are considered to produce more accurate accessibility estimates for accessibility mapping than surface-based computations, as long as all considered locations are properly connected to the network. Thus, Biba et al. (2010) proposed a parcel-network method that uses large scale parcel centroids to transfer demographic attributes to the network. However, common practice is to connect much larger zones, e.g. statistical units, to the transport network (Ahlfeldt 2008).

Travel time on public transport T_N is determined by the underlying service network and can be queried from dedicated databases. Travel times may change depending on the starting time, thus reflecting different schedules. Hence, a specific travelling time is linked to a certain daytime.

3. Test Case Setup

The commuter scenario for accuracy assessment is settled in Germany's capital Berlin. We use an offline query planner provided by the transport department to get travel times T_N from a central station to all nodes of the public transport network. A pedestrian walking speed of 75 m/min is assumed. For simple isochrone computation algorithms, the transport department recommends a detour factor of 1.4. The road network was extracted from Open Street Map data (OSM 2010). Both algorithms are set up to compute a distance transform in a 10 m resolution raster image (6003x4864 pixels) with 4510 transport stops.

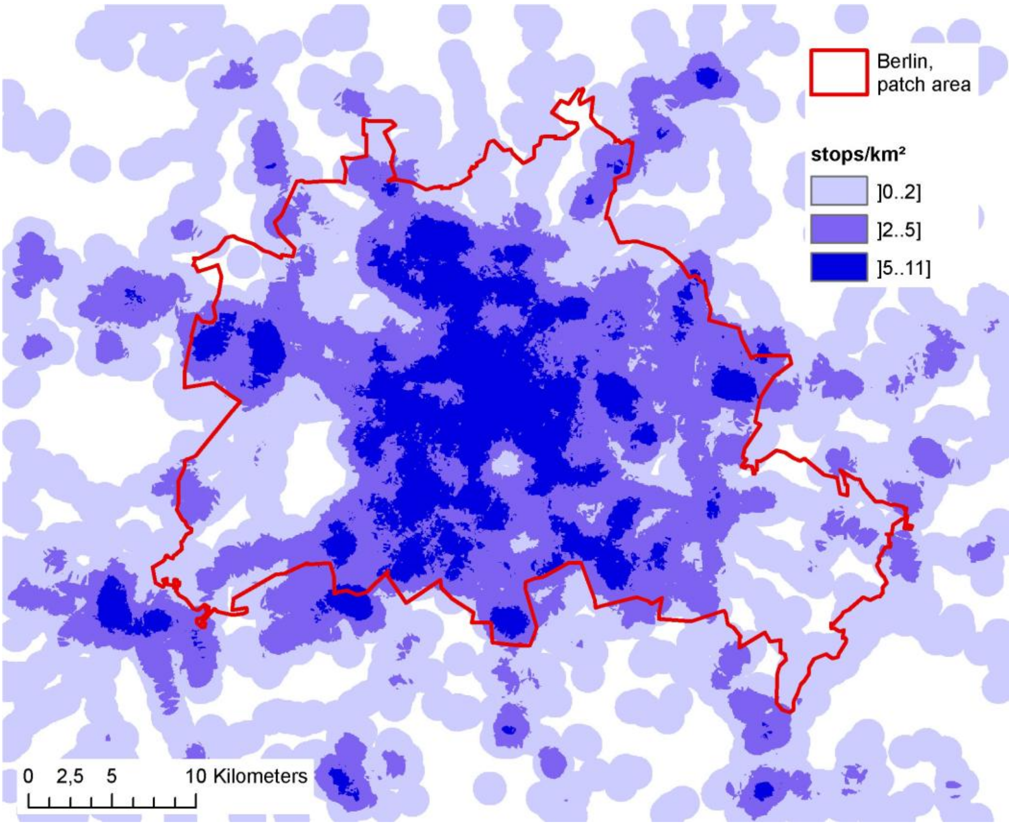

Figure 1: Stop density in the testing area and patch boundaries.

We measure performance in two dimensions: Execution time and result accuracy. Execution time is crucial, if the analysis is to be provided interactively or in real time, e. g. for cooperative activity planning (Neutens 2007). Accuracy is a major concern of city and transport planners, but inaccurate results may also render non-expert applications useless. The accuracy of an adapted EDT algorithm is assessed in relation to the local density of public transport stops (Figure 1). A reference result in terms of accuracy is obtained with an algorithm that relies on the road network as well as CDT to provide location-based travel times.

The combined network and CDT-based algorithm (NCDT) determines travelling times in a two-step process: First, the travel times on the road network are computed with the transport stops as starting locations using an initial penalty on the starting time T_N derived from the schedule. In a second step, any location outside the network is assigned a travel time by computing a CDT that respects natural barriers. The complex travel time T_C is the sum of both. Eventually, like in the parcel-network method, it is assumed that a pedestrian will take the shortest path to get from the network to any outside location. By relying on network-based computation as far as possible and taking advantage of surface-based distance computation to fill the “meshes”, this algorithm is assumed to produce robust yet very accurate results even in cases with sparse network data. It was implemented using the PCRaster library (PCRaster 2010).

The adapted EDT algorithm was enhanced to respect an initial start time (temporal offset – tEDT). It assumes a uniform surface without barriers so pedestrians are able to walk directly from one point, e.g. a bus stop, to arbitrary places. The tEDT algorithm is implemented using the approach of Hoff III et al. (1999) to calculate distance efficiently on graphics hardware: For each transport stop (seed point) a 3D cone is created representing the linear movement of a pedestrian (Figure 2). The slope of the cones equals the pedestrian’s speed; the cones’ tip is positioned according to T_N on the time axis. By rendering this scene from top, the graphics hardware performs depth sorting using a dedicated depth buffer to yield correct render order. The resulting image taken by a virtual camera is the distance map. To circumvent resolution limits of the graphics hardware, tiled rendering is applied for high resolution images.

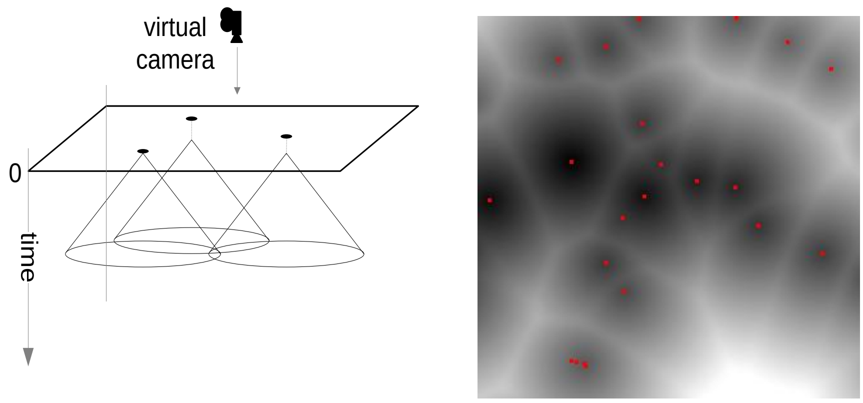

Figure 2. GPU-based tEDT computation: conversion of 3D cones into a distance map.

4. Performance Assessment

The accuracy assessment shows the expected underestimation of travel times, even if a detour factor is applied in the tEDT computation (Figure 3). The error clearly decreases for increasing stop densities. In both cases the error distribution shows a strong shift to underestimation, which might be a major problem for some applications. In combination with a detour factor, tEDT provides good accuracy for general applications in regions having a stop density ≥ 5 stops/km². For some applications the error in regions having a stop density of ≥ 2.5 stops/km² might be acceptable.

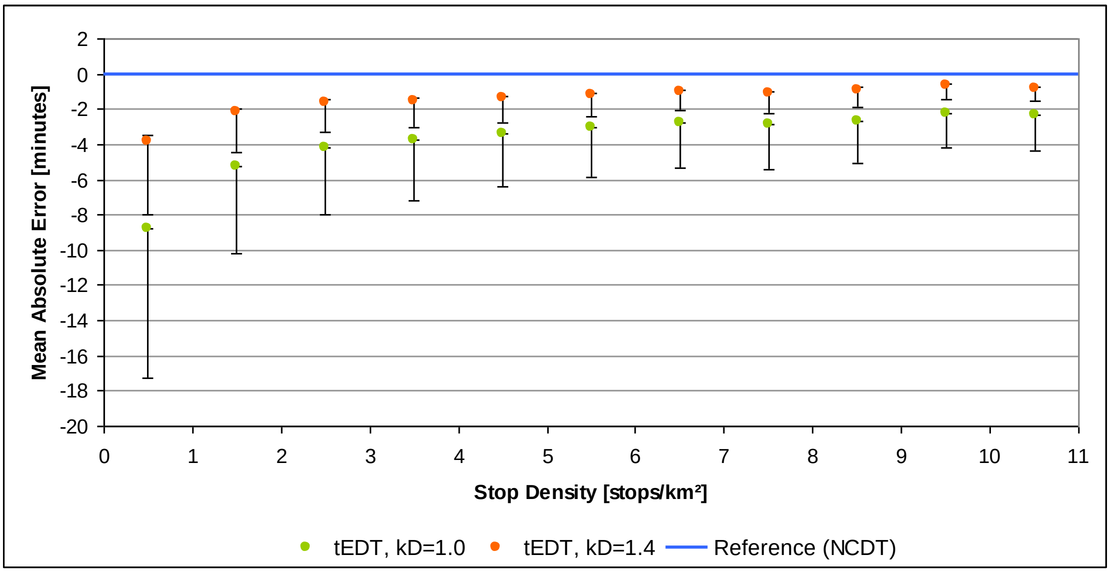

Figure 3. Mean absolute error (MAE) of tEDT computations compared to the NCDT algorithm.

Considering the computation time, the NCDT algorithm in PCRaster requires 260 seconds to compute the test area, compared to 2.4 seconds the GPU-based tEDT needs to complete. Taking a closer look at the tEDT implementation, processing time increases by square of resolution as eventually more raster image tiles have to be rendered. Computation time scales approximately linear with the number of seed points, i.e. transport stations (Table 1). Measurements were done on an Intel 2.4 GHz CPU utilizing only a single core with 4GB RAM. The graphics card used is an nVidia Geforce 9800 having 512MB VRAM.

Table 1. Performance of tEDT on a GPU at different resolutions (a) and station counts (b, at 10 m resolution).

a)

Resolution [m]	Computation Time[s]
20	0.82
10	2.42
5	9.08

b)

Stations [n]	Computation Time[s]
1000	0.50
2000	0.83
3000	1.19
4000	1.80

Based on the performance results, we can conclude that the GPU-based tEDT offers excellent computation performance in terms of timing, even at high resolutions. Depending on the application scenario, the systematic errors of the tEDT may be judged as acceptable, especially in general purpose applications like concurrent web-based distance mapping. With the upcoming adoption of WebGL (Kronos Group 2010), the computation intensive part of the distance mapping could be conducted on the client side thus reducing the server load. The increasing underestimation of distance in regions with sparsely distributed stops is certainly a drawback for some use cases. To preserve the speed advantage the tEDT algorithm might be enhanced with a correction factor based on local stop density to compensate for the systematic error.

The NCDT algorithm based on a GIS library delivers precise results, especially in peripheral regions. Due to its complexity and the possibility to incorporate additional information like barriers and spatial resistance it better suits an expert domain, such as traffic and city planning. There, the trade-off between long computations times and low errors is acceptable for many use cases. The derived large scale accessibility maps may be used to optimise the passenger network, city infrastructure, transport schedules or even for property value assessment.

However, when frequently updated real-time data from sensors (e.g. telemetry data for traffic management or passenger routing) has to be processed promptly, long computation times may disqualify a precise calculation and favour a simpler EDT algorithm. The availability of these fast distance operators currently is the only way to calculate vast numbers of distance maps on ordinary and thus affordable hardware. An application example is the generation of global proximity indicators to derive global (N to M) centrality measures, where the number of individual distance maps is equivalent to the number of possible origins or cells in raster-based data models. If applied in a professional context, simplified high-speed distance transformations should also deliver an additional location-based error estimator to support error propagation in geoprocessing.

In future work, it will be interesting to research how the road network and the introduction of simple constraints could be exploited in GPU-based implementations to achieve high accuracy at short computation time. Complementary, applications relying on simple distance transformations have to be enhanced to supply information about the expected error in the computation result. Once the error is communicated to the user (or the subsequent processing steps) it can be dealt with. Finally, the provided performance assessment should also help to design some of the suggested applications and thus create new findings in neighbouring research disciplines.

Acknowledgements

The authors would like to thank Verkehrsverbund Berlin-Brandenburg for providing the data and the anonymous reviewers who helped improving this submission.

References

Ahlfeldt, G. M. (2008): If Alonso Was Right: Accessibility as Determinant for Attractiveness of Urban Location. *Hamburg Contemporary Economic Discussions*, 12. University of Hamburg.

Beimborn, E. A.; Greenwald, M. J.; Jin, X. (2003): Accessibility, Connectivity, and Captivity: Impacts on Transit Choice. *Transportation Research Record*, Vol. 1835: 1-9.

Biba, S.; Curtin, K. M.; Manca, G. (2010): A new method for determining the population with walking access to transit. *International Journal of Geographical Information Science*, 24(3): 347-364.

De Smith, M. J.; Goodchild, M. F.; Longley, P. A. (2009): Geospatial Analysis. A Comprehensive Guide to Principles, Techniques and Software Tools. 3rd edition. Troubador Publishing, Market Harborough.

Eastman, J. R. (2006): IDRISI Andes. Guide to GIS and Image Processing. Clark University, Worcester, MA.

European Commission (2001): European transport policy for 2010: time to decide (White Paper). COM(2001) 370. Brussels.

Fabbri, R.; Bruno, O. M.; Torelli, J. C.; da F. Costa, L. (2008): 2D Euclidean distance transform algorithms: A comparative survey. *ACM Computing Surveys*, 40(1): Article 2.

Hoff III, K. E.; Keyser, J.; Lin, M.; Manocha, D.; Culver, T. (1999): Fast Computation of Generalized Voronoi Diagrams Using Graphics Hardware. *Proceedings of the International Conference on Computer Graphics and Interactive Techniques*, New York, NY, USA. ACM Press/Addison-Wesley Publishing Co: 277–286.

Horner, M. W.; Murray, A. T. (2004): Spatial Representation and scale impacts in transit service assessment. *Environment and Planning B: Planning and Design*, 31(5): 785-797.

Khronos Group (2010): WebGL Specification. Working Draft. WWW document, https://cvs.khronos.org/svn/repos/registry/trunk/public/webgl/doc/spec/WebGL-spec.html (last access: 29.6.2010).

Lei, T. L.; Church, R. L. (2010): Mapping transit-based access: integrating GIS, routes and schedules. *International Journal of Geographical Information Science*, 24(2): 283-304.

Neutens, T.; Witlox, F.; Van De Weghe, N.; De Maeyer, P. H. (2007): Space-time opportunities for multiple agents: a constraint-based approach. *International Journal of Geographical Information Science*, 21(10): 1061-1076.

OSM (2010): The Open Streetmap Project. URL: http://wiki.openstreetmap.org/ (last access: 29.6.2010).

PCRaster (2010): PCRaster Environmental Software. URL: http://pcraster.geo.uu.nl/ (last access: 29.6.2010).

Schnabel, W.; Lohse, D. (1997):Grundlagen der Straßenverkehrstechnik und der Verkehrsplanung, Teil 1: Verkehrsplanung. Verlag für Bauwesen, Berlin.