

Using a GIS System for the Generation of Driving Simulator Scenes

Jan-Henrik Haunert, Claus Brenner, Hauke Neidhart
Institute of Cartography and Geoinformatics, University of Hannover, Germany.

Abstract

Driving simulators often require a linear description of scenes in terms of events which happen at specified locations. This concept is very sensible for the actual simulation, since it not only saves space, but also makes simulations repeatable. Since maneuverability is essentially constrained to one dimension, test persons will encounter the very same scenes and challenges during the simulation. However, defining scenes with a reasonable complexity or even scenes resembling real world situations is very elaborate. Furthermore, scene reuse is not supported very well, since cutting and pasting of linear scene descriptions requires a considerable ability to imagine the actual scenes being manipulated.

In this paper, we show how Geographic Information Systems (GIS) can be used to define, visualize, and manage scene descriptions for a simulator system. To that end, we have chosen ArcGIS from ESRI, Inc. for the GIS part and STISIM drive from Systems Technology, Inc. for the simulator part. One key characteristic of the approach is that rather than putting the entire effort into building individual simulator scenes, one builds simulator worlds, from which later different simulator scenes can be derived with little effort.

Key Words: GIS, driving simulator, scene generation, clothoids.

1 Introduction

1.1 Geographic Information Systems

In general, a Geographic Information System (GIS) is a computer-based system which enables the capture, administration, analysis and visualization of geographic information. A fundamental component of a GIS is a database management system, which provides the access to a spatial database. The GIS allows to query spatial features, their attributes and their relationships which are modeled in this database. However, the major benefit of a GIS is its multiple functionality: It is more than a database management system (which lacks the capability of cartographic visualization) and more than an automated mapping application (which does not provide analysis functionality). A GIS provides geometric, topological and thematic information about objects. The geometry of objects is not only used for the graphical visualization of the features, but also result in spatial relationships of objects, which enable spatial joins that can be used in analysis operations. Spatial indexing allows access to features within a specified location. An overview on Geographic Information Systems is given by Longley et al. (2001). Thill (2000) focuses on its relevance for transportation applications.

The application of GIS to the generation of driving simulations is a promising approach. Driving simulations contain many objects of different types which interact with each other. Attributes and relationships of objects can be maintained in databases which can be easily accessed by a GIS. The cartographic visualization of this information which naturally has a spatial aspect becomes feasible. Traffic simulation systems have been implemented directly within special geographic information systems which are able to represent the dynamic behavior of objects. Claramunt et al. (2000) created such a system for the analysis of urban traffic patterns.

1.2 GIS and Driving Simulation

There are a number of simulators available, e.g. for vessels, aircrafts, trains, and cars. In the approach we describe here, the simulation software STISIM drive was used. The task of the GIS is to ease the definition, administration and editing of simulator scenes. Also, a cartographic visualization of the involved objects is possible which helps to understand the spatial characteristics of the simulation. For these tasks a static depiction of the world within a GIS is sufficient. After scenes are defined within the GIS they are transferred to the simulator using its scene description language. The simulator then handles all dynamic aspects. Hence, we used a conventional GIS software which provides possibilities to implement customized editing tools. We have chosen the ArcGIS/ArcMap software, which offers ample facilities for the development of user defined tools. These have been implemented as VBA scripts within a software internal programming interface. ArcGIS offers a wide range of existing software components which form the ArcObjects framework (ESRI 2001a, 2001b). A data-model has been designed and implemented as a personal geo-database.

1.3 Data Acquisition

With the perspective from the field of Geographic Information Sciences, modern trends in this research area were kept in mind when designing the GIS to enable the creation of simulations. In particular this

addresses possibilities for the integration of realistic data to a simulation. Especially roads and buildings are treated here. For this task especially two topical research fields are relevant: The acquisition of data that can be deployed for simulations and the integration of datasets from different sources, which originally do not fit together or do not meet the needs of the application (in this case, the simulation).

Obviously, a realistic simulation can only be as rich in detail as the previously collected datasets. Therefore, the acquisition of information is a crucial task for the creation of realistic simulations. In the last years, progress has been made in this research field, which is especially due to the emergence of laser scanning technology and digital photography/digital photogrammetry. Such acquisition systems can be used to create realistic three-dimensional digital models of buildings (Brenner, 2005). Often additional information is integrated to obtain data about buildings (Neidhart & Sester, 2004). Laser scanning can also be used to derive more detailed models of roads (Hatger, 2002).

Due to budget reasons, the collection of new field data for a certain application is normally not feasible. Especially since public providers of geographic data are short of money, the demand for a more efficient use of existing datasets has increased. In practice, it is usual that different map contents are collected by different authorities. For example, building groundplans are collected by cadastral offices, navigable road networks by commercial providers, and height models by national mapping agencies. Even objects of the same class do not necessarily have the same origin. This results in a number of problems, since specifications and geometrical representations are not homogeneous. To reach a higher interchangeability the integration of different spatial databases has become a topic of major interest (Devoegele et al., 1998). Approaches have been developed to describe correspondences of datasets (Walter & Fritsch, 1999) and methods have been realized which allow new application areas of given datasets. Haunert & Sester (2004) show how navigable road datasets can be derived from cadastral data. The same dataset was applied by Elias & Brenner (2004) to extract landmarks which are relevant for future navigation systems.

2 Basic Concepts and Terminology

2.1 Spatial Reference System vs. Linear Reference System

Referencing objects spatially in terms of two- or three-dimensional coordinates which form geometries is a core concept of GIS. Normally, point-objects are defined by specifying pairs of x- and y-coordinates within a spatial reference system. Additionally, the topological relationship to other objects can be stored explicitly. However, within transportation applications it is often useful to define the location of an object by reference to a road object. Often the “kilometer-post representation” is used, which defines the position of a point by its offset along the road, beginning from its start point or a distance marker on the road (Jensen et al. 2003). Scarponcini (2002) describes a generalized linear referencing model, which subsumes different models for linear referencing systems. These models are applied when the absolute position of a feature on the map is not of interest. For some applications, a linear reference system is more useful, because features that are located along a road can be ordered easily by the stored distances to gain a sequence of events. A driving simulator is an example for such an application, as objects are displayed in the order which is defined by their distance along a route.

2.2 Simulator World vs. Simulator Scene

The STISIM driving simulator relies on a linear reference system. A project file consists of events which are defined to be processed after a certain distance along the predefined route is covered. However, it is beneficial to represent features on the map independent of a certain route. This makes it possible that the same object, such as a building, appears in simulations of different routes that are defined on the same road dataset. Therefore, the general approach is to represent objects within the spatial reference system of the map frame. This map represents a *simulator world* from which different *simulator scenes* can be derived. Each time a new simulator scene is created, the relevant objects along a defined route of the road network are collected and transformed to the appropriate representation of the simulator.

The definition of a simulator world within a spatial reference system greatly eases the integration of existing data into a simulation. Data on road networks, buildings and other relevant objects are available from cadastral services, national mapping agencies or private companies. With this data simulations can be created which depict real world scenarios. Figure 1 shows the simulator world within the map interface of the GIS software. The next two sections describe how simulator worlds are built and how simulator scenes are derived. The contents and functionalities depicted in Figure 1 will be described in detail.

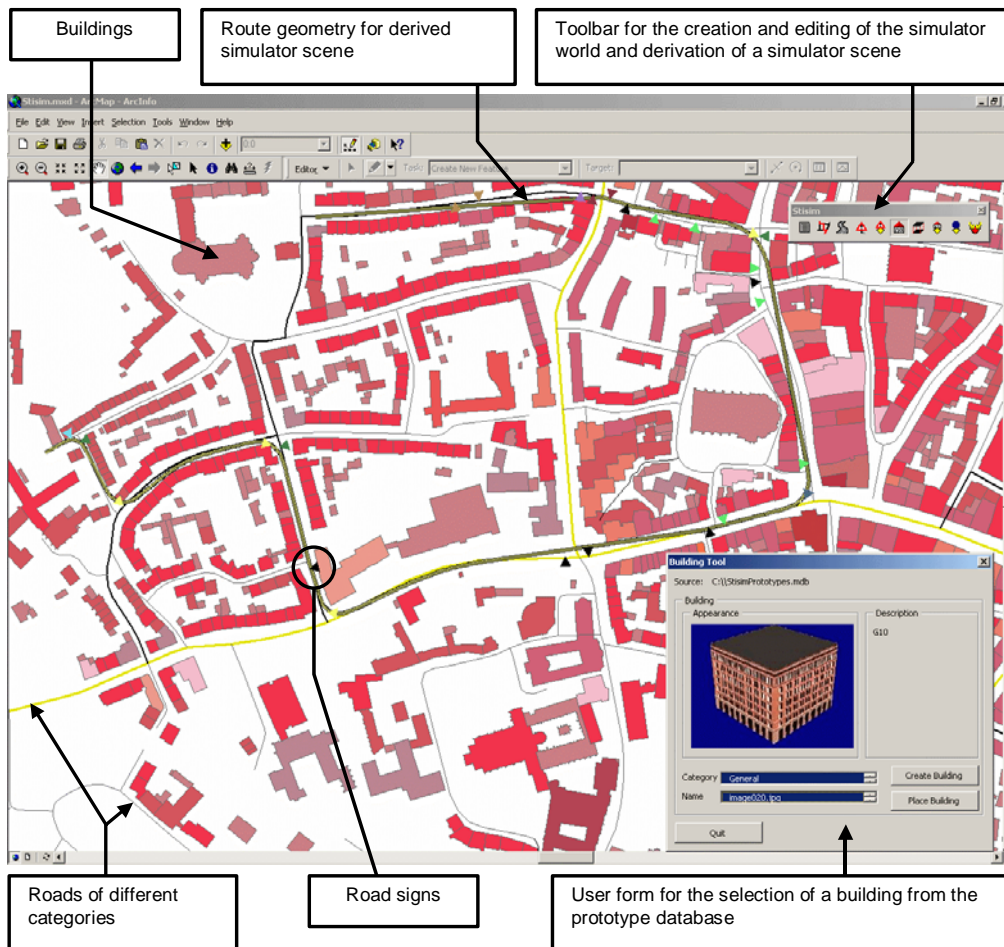


Figure 1: Simulation world within ArcGIS map interface ArcMap

3 Creation of Simulator Worlds

3.1 Integration of Existing Road Network Datasets

The basis of a simulator world is a road network which allows the navigation through the map. The definition of a route for a simulator scene is constrained to this network. Area-wide datasets of road networks have been collected for car navigation systems by companies such as Tele Atlas or Navteq. Alternatively, the functionalities of the GIS software can be used to create own road networks or to edit existing ones. Topologic errors which may exist in a given road dataset can be resolved with these editing tools as well.

Often, existing road datasets also contain information about the class of road objects. Normally, this is represented by an attribute of type integer. The value of this attribute expresses that the object is part of a motorway, main road, side road, etc.. If road classes are available, they can be used to automatically define parameters relevant to the simulator, such as the width and the number of lanes of a road. If these default values are not appropriate for certain road segments, parameters can also be set individually and then supersede the default settings. Within the GIS, this is done via an additional table, which only contains the definitions or roads which differ from the default settings. In case the attributes of a road change within one object, the tools of the GIS software can be used to split the object at a defined point.

3.2 Prototypes and Instances

The STISIM simulator allows the integration of predefined objects as well as user defined objects into the simulation. For example, standard buildings of different categories can be placed in the simulation scene. User defined objects can also be integrated if their geometry is defined in 3D Studio File Format (3ds). To handle these objects in the GIS, a database has been created, which is a catalogue of prototypes. A prototype of a building describes all of its features which are relevant for the simulation but it does not specify the actual position on the map. This information about an individual instance of the prototype needs to be represented separately, so that the same prototype can be instantiated many times in the

simulator world. Such an instance only comprises an identifier, its individual geometry on the map and a link to the corresponding prototype. To add new standard objects to the database user forms have been designed for different classes of objects. Images of objects can be added to the catalogue database. Later, when browsing through the catalogue they serve as icons to identify the selected object (Figure 1). An instance can be created by specifying the location and orientation of the building in the map.

Road signs can be selected similarly from a catalogue, but the positioning on the map works differently. For road signs, a snap to the closest point on the road is activated. The side of the road, on which the sign is located, is specified with the arrow keys, before the location is confirmed. The currently snapped point on the road network as well as the selected side of the road are visualized by an icon in the map frame.

3.3 Expression of Relationships to Roads by Links

The sign event of STISIM drive requires parameters which specify the distance along the route and the road side on which the sign appears. A lateral distance is not required, since the sign will be placed automatically according to the road width. The most natural representation for such an event in the GIS is a point feature which is located on the line which represents the road. Additionally, the information about the side of the road on which the sign is located needs to be stored as attribute value. Here it is important to realize, that this attribute needs to refer to the direction of the road segment from the original road dataset and must not refer to the direction of a route which has been defined for a certain simulator scene. Consequently, a sign which has the attribute value “right” might appear on the left side in the simulation, since both directions might be converse. Without this consideration, it is not possible to create sign objects within the GIS which can be reused for several simulation scenes. To prevent misunderstandings of this concept and to prevent ambiguities that result on junctions, explicit links to road objects are specified for these objects.

3.4 Integration of Buildings from Existing Datasets into the Simulator World

Apart from road objects, also buildings from existing sources can be integrated into the simulator world. Custom buildings can be integrated to the catalogue of prototypes, if a ground plan is available for the GIS and a 3D model exists for the simulator.

For the creation of simple building models, an additional mechanism has been designed. In Germany, detailed datasets with ground plans of buildings are acquired by the cadastral offices. Additionally, the number of floors is sometimes captured for every building, which can be used as a height indication. If a standard value for the height of a floor is defined, simple three-dimensional models can be derived by extruding the ground plan along the vertical axis. To allow the use of such simple models of real buildings, a module has been created which reads a dataset of ground plans, automatically constructs the three-dimensional models of the buildings as described and exports the models in the 3D Studio File Format, as required by the simulator. This module is integrated into the user interface of the GIS. To create buildings for the simulation world, ground plans from the cadastral data are selected within the GIS, prototypes are generated and added to the prototype catalogue. Again, this allows their reuse at several places in the map. However, if desired, the GIS will also immediately instantiate each generated building at the position where it appears in the original cadastral map.

3.5 Definition of Generic Events

The STISIM format allows a huge number of different event types. To create a data model which is able to describe every event type with its parameters would not make sense, because some of the event types are only needed for very special scenarios. Therefore, a general event type was defined for the simulation world, which covers most of the events defined in the STISIM drive description language.

Normally, within STISIM drive, the location of an object in the simulation is defined by two parameters, the distance from the start of the simulation to the point where the event needs to be processed and the distance from this point to the location of the object. The second parameter can be described also as the distance from which the object can be seen. Some events do not require this viewing parameter. In the simulation world a generic event is represented as a point feature which is located on the road network. This point either represents the point in the simulation on which the event needs to be processed, or the point on where the object actually resides. In the second case the viewing parameter needs to be specified as attribute value, so that the distance to the point where the event needs to be processed can be derived from the location of the map. All other parameters can be defined as attribute values of the type string, which simply are passed over to the simulator’s scene description.

The variety of objects which are part of the simulation world can be greatly increased by these generic events. For example, pedestrians, trees, or fog can be added. However, it has to be noted that the absolute location and the extents of the objects are not independent from the route. This is because the values of

the parameter string are not interpreted before they are passed to the simulation. Consider the example of a road sign: The sign can be described as a generic event, but if the parameter string contains the information that the sign is on the right side of the road, the sign will always appear on the right side relative to the direction of the route in the simulator scene. This shows that the strict differentiation between the simulator world and the simulator scene is not realized for generic events. Nevertheless, introducing this event into the GIS is of great help, since the location of the event appears in the map which makes it easier to understand and modify.

4 Creation of Simulator Scenes

4.1 Definition of Routes

Simulations of STISIM drive are constrained to a path. Even though it is possible to deviate from the given path at junctions, the test person will nevertheless find the same situation, no matter what directions were taken. Therefore, a simulation scene can be defined by selecting a linear route through the simulation world. Roads which diverge from this route and objects which are situated along these roads only need to be included into the simulation scene if they are visible from the main route.

A tool has been created which enables the fast definition of a route. Initially, a start point is selected on an edge of the road network. After this, the arrow keys on the keyboard are used to specify the direction in which the route shall proceed. Contiguous segments are automatically added until a junction is encountered. This continues until an end point is defined. The route is added to the map frame of the GIS. All roads which diverge from this route are added to a table, so that the junctions on the route can be modeled automatically for the simulator scene.

4.2 Computation of the Route Geometry

STISIM drive represents any route as a combination of straight lines, circular arcs, and clothoids. The latter are required as a transition between straight lines and circular arcs in order to prevent jumps in curvature. The actual representation is such that the route by default is straight. To change direction, a curve event is inserted into the scene description which consists of a clothoid, followed by a circular arc, again followed by a clothoid. This means that the curvature is zero before the event, increases during the first clothoid, then is constant along the circular arc, and finally decreases during the second clothoid, being zero again after the event.

Clothoids, or Cornu spirals, are curves in two-dimensional space which can generally be described by

$$\begin{bmatrix} x(t) \\ y(t) \end{bmatrix} = a \cdot R \cdot \begin{bmatrix} C(t) - C(t_0) \\ \sigma(S(t) - S(t_0)) \end{bmatrix} + \begin{bmatrix} x_0 \\ y_0 \end{bmatrix}$$

where t is the curve parameter, t_0 is the start parameter, $[x_0, y_0]^T$ is the start position, a is the scale, R is a 2×2 rotation matrix, and $\sigma = \pm 1$ is the sign determining a left or right curve. The curves $C(t)$ and $S(t)$ used in this definition are the two Fresnel integrals

$$C(t) = \int_0^t \cos \frac{1}{2} \pi u^2 du \quad \text{and} \quad S(t) = \int_0^t \sin \frac{1}{2} \pi u^2 du,$$

for which no closed form description exists. Instead, Taylor expansions can be used, or better, rational approximations, as e.g. given in (Heald, 1985).

After the route has been defined in the GIS as described above, it consists of a polyline – a sequence of connected straight line segments. Deriving from this a sequence of line segments, clothoids, and circular arcs is a nontrivial task. Meek and Walton (1989) describe the different cases which can occur when clothoids are used to join a given sequence of circles and lines. Meek and Thomas (1991) present a method to construct a curve consisting of lines, clothoids, and circular arcs which passes through a given set of points. However, in both cases, the algorithms are not guaranteed to work for any given set of initial data.

A very attractive method has been described by Schneider and Kobbelt (2000), who use discrete clothoid splines to find the curve. Initially, points are distributed along the original straight line segments. Then, for each point, the discrete curvature is computed based on the two neighbouring points. Points are moved individually one after the other to meet the conditions that the curve is arc length parameterised and has a piecewise linear discrete curvature. The overall curve shape is obtained when the iterations become

stable. We have found this method to work even at interactive speeds. There are, however, two drawbacks. First, from the discrete curve, the parameters of continuous curves have to be derived again. And second, this approach can unfortunately not be used for STISIM drive scenes because the simulator requires the curvature to be zero before and after each curve event. However, there are scenes which do not conform to these restrictions and consequently, the discrete clothoid spline will be a discrete curve which cannot be converted back to continuous curves which fulfil this restriction. In fact, consider a winding spiral which cannot be described by a single clothoid. Then, several clothoids are needed which are joined at points of nonzero curvature.

Since it was the goal to provide a fully automatic procedure to obtain the curve description from the given polyline, we used the following simplified approach. As Meek and Walton (1989) show, joining two line segments by a pair of clothoids is always possible. Similar to this approach, we join two line segments by three curve parts, a clothoid, circular arc, and a clothoid by distributing the angle between the segments to the three parts in equal amounts (Figure 2). The angle $\alpha + \beta + \gamma$ determines the shape of the curve except for the scale, which is defined in a second step.

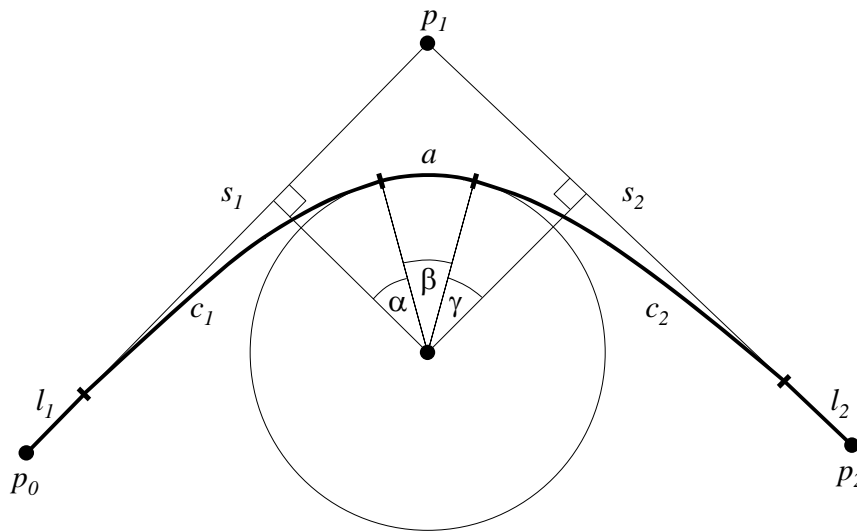


Figure 2: Join of two segments s_1, s_2 connecting points p_0, p_1, p_2 by a curve consisting of line segments l_1, l_2 , clothoids c_1, c_2 , and a circular arc a .

Even though this method has the advantage that it works for any given polyline, its disadvantage is that it is tied directly to the original set of line segments. If there are many short segments present, this will result in a huge number of short clothoid parts which do not represent the overall curve very well. Therefore, we initially reduce the number of segments by applying the algorithm of Douglas and Peucker (1973). This algorithm uses a threshold which determines how well the reduced curve approximates the original polyline. Figure 3 shows an original curve and several resulting curves for different threshold values. One can notice a substantial reduction in the number of curve segments which occurs already at thresholds of two to five meters. The scale of the clothoid – circular arc – clothoid combination at each pair of line segments is finally determined by the length of the segments, subject to the following constraints:

- two successive curves must not overlap
- the radius of the circular arc should not be smaller than a given value
- the distance between the curve and the original points should be small.

The first constraint poses an upper bound on the scale factor. On the other hand, the second constraint results in a lower bound. If this lower bound conflicts with the first constraint, the scale is chosen as large as possible. Figure 4 shows the results for different settings for the minimum radius. The maximum distance between original points and curve results in another upper bound for the scale. Figure 5 shows the effect of different settings for the maximum distance.

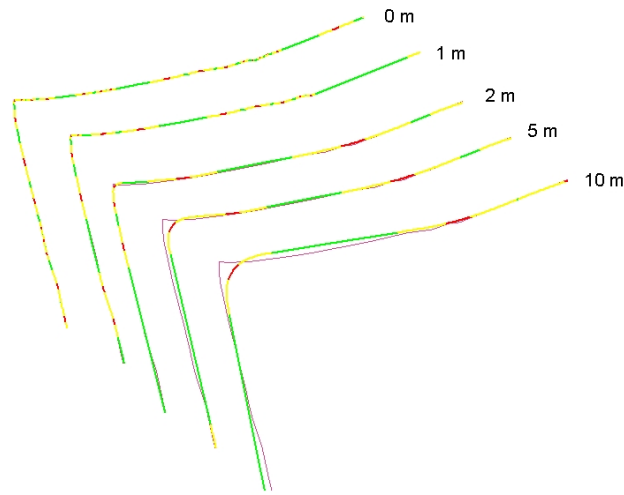


Figure 3: Results of the line simplification with different thresholds. Green: straight line segments, yellow: clothoids, red: circular arcs.

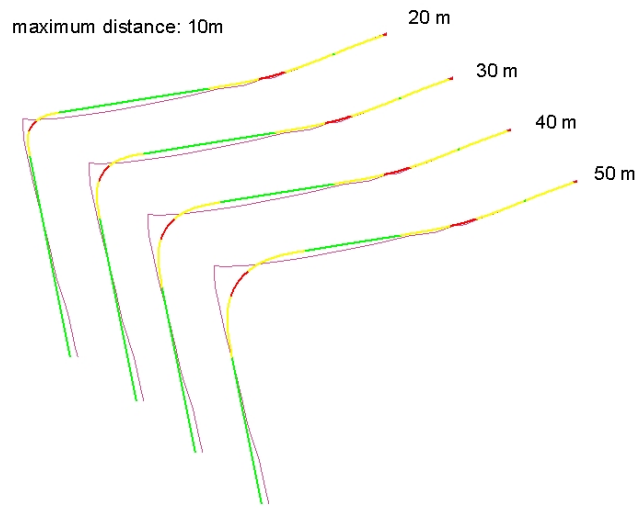


Figure 4: Results of the curve generation for a fixed distance threshold and different minimum curve radii.

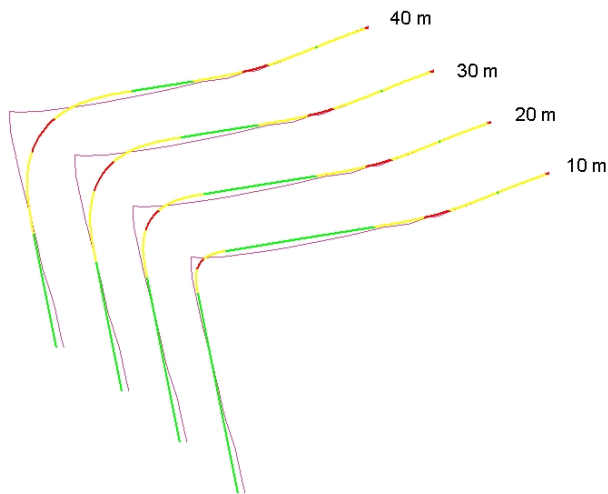


Figure 5: Results of the curve generation for different maximum distances.

After the entire curve has been computed, the longitudinal positions of all original points are determined. This is necessary for the subsequent calculation of the positions of signs and houses. Also, intersections

are handled. STISIM drive has a dedicated intersection event which however can only represent right angle intersections. These are detected in the scene by an analysis of the angle of all edges adjacent to a junction. Right angled junctions are converted to an intersection event, whereas other angles will be represented by polygons or static objects in the driving simulator scene definition.

4.3 Generation of Other Events in the Scene Description

The route geometry is directly exported in the simulator's scene description file format. Other objects along the route are handled as follows. Road signs which are explicitly linked to segments of the route and generic events which are located on the route are selected automatically and added to the scene description. Relevant buildings are found by applying a buffer to the route and its side roads. Buildings which intersect this buffer are selected and exported to the scene description as well. Attributes such as the left/right information of road signs are adjusted at this point to fit the selected route.

Since the geometrical representation of all features in the simulator world refers to the spatial reference system of the map frame, a transformation needs to be applied which generates a description that refers to the linear reference system of the simulator. For point objects which are located on the route only the longitudinal distance needs to be calculated. For buildings, the reference points of the ground plan are projected perpendicularly onto the route, to gain the longitudinal distance from the beginning of the route, the lateral distance from the route, and the orientation angle relative to the route (Figure 6).

However, these values cannot be used directly since the originally selected route polyline has been replaced by a simulator description in terms of line, clothoid, and circular arc segments, which changes the length. Especially, the line simplification algorithm reduces the total length. Because of this, a sign placed in front of a curve might appear after the curve in the simulation, if the measured longitudinal distance is transferred directly to the simulator's scene description.

The required correction is a transformation between two linear reference systems. The first system is defined by the longitudinal distance along the linear route segments in the GIS, the second system by the computed route for the simulator. The transformation is carried out as follows. The module which computes the route as described in the previous section also determines the longitudinal position of so called control points. These are all points where a new element of the route begins, i.e. a new line, new clothoid, or a new circular arc. Using this, the positions along the original route are transformed using a rubber band approach. This transformation ensures that the positions of simulation objects relative to events that express changes of the route geometry like curve events are preserved.

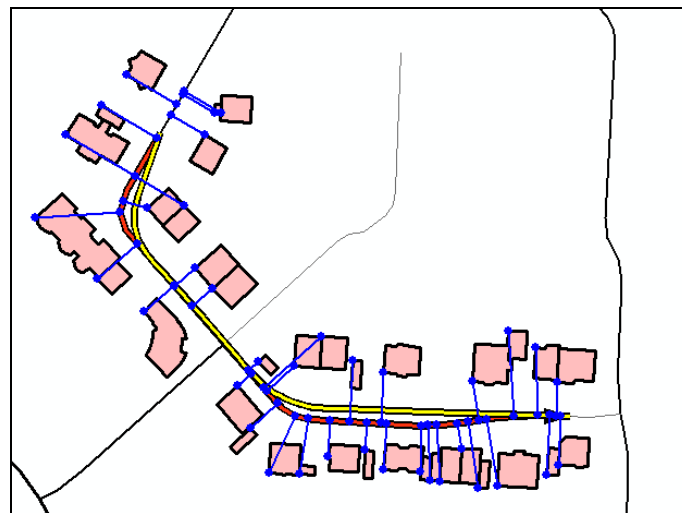


Figure 6: Buildings along the route are selected via a buffer operation. Their reference points are projected onto the route to find their longitudinal and lateral position.

Contrary to the longitudinal distance of an object, its lateral distance from the route as well as its orientation angle relative to the route are not changed. This guarantees that relationships between objects and the route are preserved. For example, a building which is parallel to the road in the map will also have this characteristic in the simulation and will appear at the same lateral distance from the road. **Fehler! Verweisquelle konnte nicht gefunden werden.** shows a simulator world in the GIS, the defined route, selected buildings along the route, and the final simulator scene as rendered by the STISIM simulator.



Figure 7: Example of a driving simulator scene with GIS data on the left and derived simulator scene on the right. The simulation route is shown in yellow and starts close to the lower left corner. Other roads of the road network are shown as thin black lines. Grey buildings in the scene were automatically generated from the ground plans shown in red in the GIS. The white skyscraper building was taken from the building library of STISIM and inserted manually. It appears in the GIS in white on the right hand side of the road.

5 Conclusions

In this paper, we have shown how a GIS can be used to define, visualize and manage scene descriptions for a driving simulator. Instead of modeling a specific route directly, the GIS is used to build a simulator world from which later many different simulator scenes can be obtained. The world is permanently stored into the GIS database. Incremental modeling is possible, so that when shortcomings are detected in a scene, they will be fixed in the GIS world representation, being available for any scene that is derived from that point on. The GIS allows to use existing geo-information such as road databases and building groundplans, and its capability to visualize the simulator world in two dimensions makes it easy to build, understand, and edit the spatial structure. Furthermore, since the GIS uses a real-world spatial reference system, new applications are feasible. For example, real-world coordinates can be output to car navigation systems during the simulator run, making it possible to expose test persons to real car navigation route guidance.

We have realized our approach using ArcGIS/ArcMap for the GIS part and STISIM drive for the simulation part. The basic GIS functionality is augmented by VBA routines as well as additional utility programs which are all controlled directly from a toolbar within the GIS.

In a first step, the simulator world is built starting from a given road network, however it can also be digitized manually using standard GIS functionality. Road signs are then added and appear in the GIS map frame, where they can be manipulated. Buildings from a library can be manually placed in the simulator world where they are represented by their ground plans. Moreover, if ground plans and a height attribute are available from existing geodata, flat-roof buildings can be generated and placed in the simulator world automatically. For a high flexibility, the system also allows to define generic events, so that almost any event defined by the simulator can be integrated into the simulator world.

In a second step, a simulator scene is derived from the simulator world. For this, one can select and step through the required road edges in the GIS. The resulting polyline is automatically converted to a route description in terms of linear segments, clothoids, and circular arcs, as required by the simulator. Road parameters taken from a default table based on the road class, but can also be specified for each road segment individually. Signs, generic events, and buildings along the route are collected automatically and inserted into the driving simulator scene description at their correct location. The resulting scene can be directly loaded into the simulator.

6 References

- Brenner, C.; 2005: Building reconstruction from images and laser scanning, *International Journal of Applied Earth Observation and Geoinformation*, 6(2005), 187-198, <http://dx.doi.org/doi:10.1016/j.jag.2004.10.006>
- Claramunt, C.; Jiang, B.; Bargiela, A.; 2000: A new framework for the integration, analysis and visualisation of urban traffic data within geographic information systems, in: Thill, Jean-Claude; 2000: *Geographic Information Systems in Transportation Research*, Elsevier Science, Amsterdam, pp. 167-184.
- Devoegele, T.; Parent, C.; Spaccapietra, S.; 1998: On spatial database integration. *International Journal of Geographic Information Science* 12(4), pp. 335-352.

- Douglas, D. H.; Poiker, T. K.; 1973: Algorithms for the reduction of the number of points required to represent a digitized line or its caricature, the Canadian Cartographer 10:112-22
- Elias, B.; Brenner, C.; 2004: Automatic Generation and Application of Landmarks in Navigation Data Sets, in: Peter Fisher (Ed.): Developments in Spatial Data Handling - 11th International Symposium on Spatial Data Handling, Springer Verlag, pp. 469-480.
- ESRI; 2001a: Exploring ArcObjects, Volume I-Applications and Cartography, Redlands, USA.
- ESRI; 2001b: Exploring ArcObjects, Volume II-Geographic Data Management, Redlands, USA.
- Hatger, C.; 2002: Assigning the 3rd Dimension to Roads by Use of Laser Scanning Data, in: International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. 34, Part 4 "Geospatial Theory, Processing and Applications", Ottawa, Canada.
- Hauert, J.-H.; Sester, M.; 2004: Using the Straight Skeleton for Generalisation in a Multiple Representation Environment. ICA Workshop on Generalisation and Multiple Representation, Leicester, UK.
- Heald, M. A.; 1985: Rational Approximations for the Fresnel Integrals, Mathematics of Computation, 44(170), American Mathematical Society, April, 459-461.
- Jensen, C. S.; Pedersen, T. B.; Speičys, L.; Timko, I.; 2003: Data Modeling for Mobile Services in the Real World, in: Proceedings of the Eighth International Symposium on Spatial and Temporal Databases, Santorini, Greece, pp. 1-9. Lecture Notes in Computer Science, Volume 2750, July 24-27.
- Longley, P. A.; Goodchild, M. F.; Maguire, D. J.; Rhind, D. W.; 2001: Geographic Information Systems and Science. John Wiley & Sons, Chichester, England.
- Meek, D. S.; Thomas, R. S. D.; 1991: A guided clothoid spline, Computer Aided Geometric Design 8(1991), 163-174.
- Meek, D. S.; Walton, D. J.; 1989: The use of Cornu spirals in drawing planar curves of controlled curvature, Journal of Computational and Applied Mathematics 25(1989), 69-78.
- Neidhart, H & Sester, M.; 2004: Identifying Building Types and Building Clusters Using 3D-Laser Scanning and GIS-Data, in: International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. 35, ISPRS, Istanbul.
- Scarponcini, P.; 2002: Generalized Model for Linear Referencing in Transportation. GeoInformatica 6(1), pp. 35-55.
- Schneider, R.; Kobbelt, L.; 2000: Discrete Fairing of Curves and Surfaces Based on Linear Curvature Distribution, Laurent, P.-J., Sablonnière, P., Schumaker, L. L. (eds.), Proc. Curve and Surface Design, Saint-Malo 1999, Vanderbilt University Press, Nashville, TN.
- Thill, J.-C.; 2000: Geographic information systems for transportation in perspective, in: Thill, Jean-Claude; Geographic Information Systems in Transportation Research, Elsevier Science, Amsterdam, pp. 3-12.
- Walter, V; Fritsch, D.; 1999: Matching spatial data sets: a statistical approach. International Journal of Geographical Information Science 13(5), pp. 445-473.