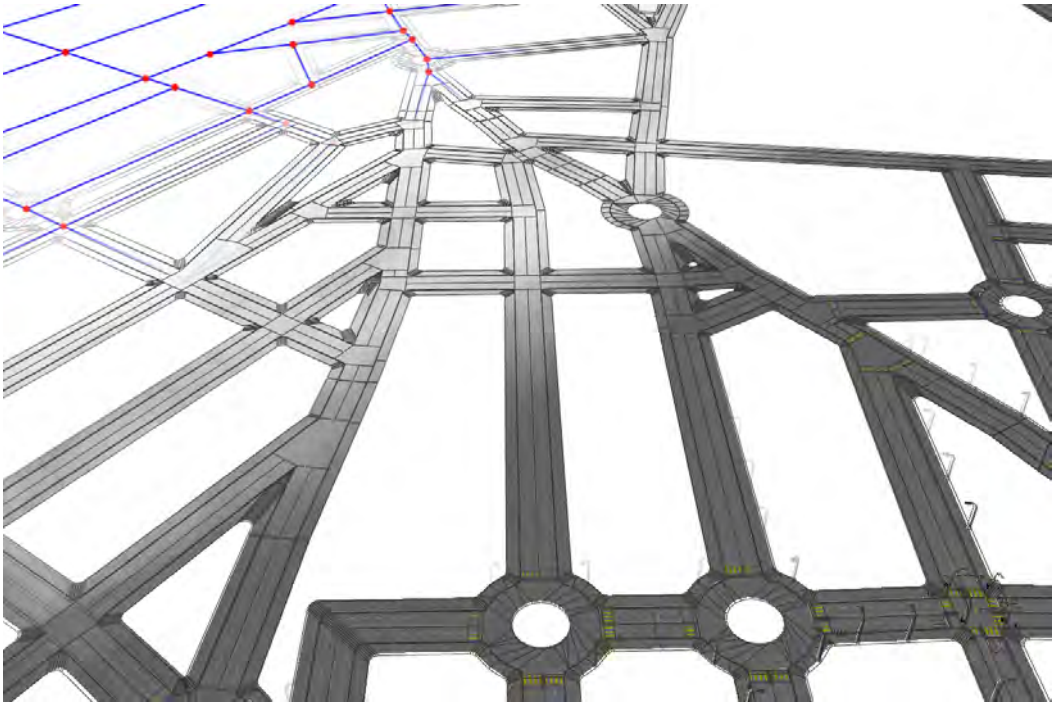
ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Computer Vision Laboratory
Prof. Luc Van Gool
2006

Semester Thesis

Procedural Construction of Streets



Author:
Marco Zimmermann

Supervisor:
Pascal Müller

All the people who contributed to the success of this thesis shall be thanked hereby.

Zürich, October 5, 2006

Marco Zimmermann

Abstract

The main goal of this project is to implement a tool for the construction of a full polygonal 3D street representation out of centerlines (raw vector data). Besides the creation of textured streets, the tool should be able to place objects like street lamps or traffic lights and solve complex intersection constructions. The solution should work for different types of streets, e.g. for ancient Roman lanes as well as for modern highways.

Contents

Chapter 1

Introduction

In this chapter a short introduction into the topic should be given. This includes a section about what should be achieved in detail, a part dealing with possible applications and one mentioning already conducted work.

1.1 Goals

The goal of this project is to be able to create a full 3-dimensional representation of a street-network on the basis of simple vector data. The 3D output should be generated automatically but still remain controllable using adjustable parameters. The module should produce a convincing output for every possible input. In the ideal case, no distinction between an automatically generated 3D model and one created 'by hand' should be possible.

1.2 Motivation

Virtual urban environments are needed for different applications, i.e. nowadays more and more the need arises to have such environments available as 3D models. For instance in movies often virtual cities or at least parts of cities are needed for certain scenes. Cities in games are of course also just virtual cities and the gamers always call for larger environments to explore and use. Other applications can be found in the field of simulation or VR tools, e.g. the Cyberwalk project [6].
For all these applications, especially those that require massive models, the modeling by hand is a very time consuming and extensive work. This is of course also very expensive.
In a virtual city the streets play quite an important role because normally the people will virtually walk or drive around in those cities and they will do this on streets. Therefore a good implementation for the generation of streets is needed. At the moment nothing exists that generates such procedurally 3D street-models.

1.3 Related Work

There is some work that is strongly related to this project which shall be described here. This includes mainly two projects.

1.3.1 The CityEngine

The CityEngine is a tool to produce procedurally generated cities [3, 4]. Based on image maps and many controllable parameters as input, an L-system first produces the road map. The land is then divided into lots and an appropriate geometry for buildings is generated. The buildings appearances follow several superimposed rules. Up to now, he main effort in the CityEngine project has been put into the L-system producing the main geometry [2] and the creation of buildings which should be very flexible to produce different looking buildings but still follow some basic rules (e.g historical)[1].
Not much effort (besides of the work mentioned below) has been put into the creation of the 3D street-network, so the CityEngine indeed features very nice looking buildings but between the buildings some work has to be done.

1.3.2 Procedural Streets Construction

This was a semester thesis with the same goal as this work [5]. The theoretical part was carried out quite detailed and proved to be very helpful for this project, but the implementation of the problem was not complete. Only basic intersection generation and street generation was included. Therefore this project should build on the work already conducted and emphasize the implementation of the theory.

Chapter 2

System Overview

In this chapter an introduction into the system should be given. First a schematic view of the different steps when producing the model will be discussed. Then some common possibilities of input formats for GIS systems as well as the input format of the streets module of the CityEngine will be described. Finally an introduction into the requirements of the output will be given.

2.1 System Pipeline

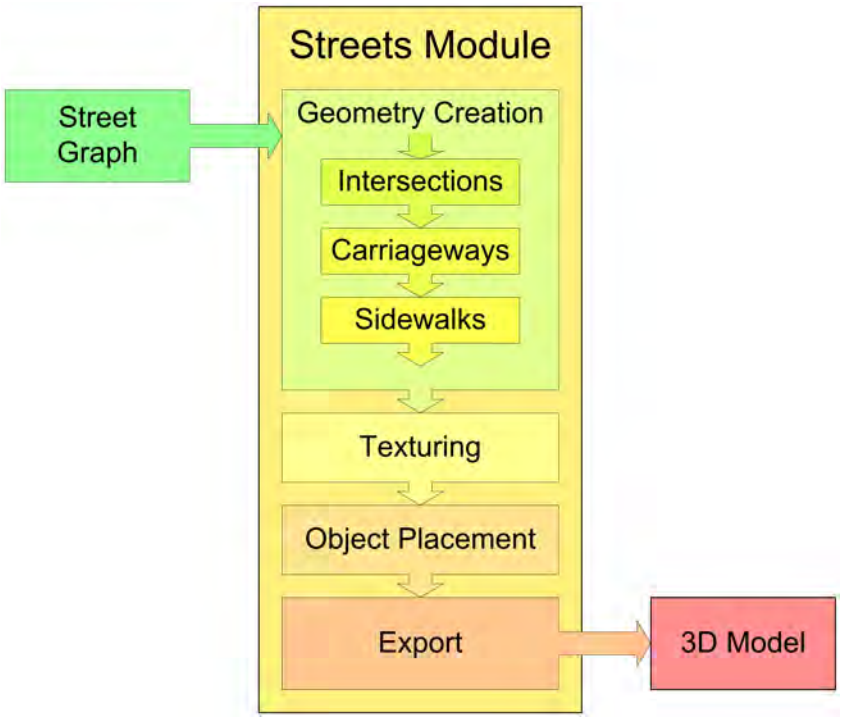


Figure 2.1: The system pipeline

From the input data the module first generates the geometry of the different parts. It starts with the intersections followed by the carriageways and the sidewalks. Then the texture points are calculated and adequate texture pictures are chosen. Next, other drawings of the streets, not included in the basic textures are added. Afterwards, the different types of objects that belong to a street network are placed. Finally the different parts have to be combined to produce a complete 3D model. This model has to be generated on the basis of the calculated points.

2.2 Input

2.2.1 GIS formats

Different formats exist to store street centerlines and geographic features, here are some examples:

- **Shape** [7] is a format originally developed by ESRI (Environmental Systems Research Institute). It is a quite simple format that can store points, lines and polylines together with attributes and very widely spread. There are also limitations, for example shapefiles do not have the ability to store topological information.

- **KML (Keyhole Markup Language)** [9] is an XML grammar and file format for modeling and storing of geographic features which includes many display possibilities for a rich representation of GIS data. This language is used by Google Earth.
- **GML (Geographic Markup Language)** [8] also a very rich XML format developed by the OGC (Open GIS Consortium), a consensus standards organization that is leading the development of standards for geospatial and location based services.
- **GPX (GPS Exchange Format)** [10] is a light-weight XML data format for the interchange of GPS data and developed by Topografix.
- **MIF (MapInfo Data Interchange Format)** [12] is a versatile format that allows the use of normal data together with many graphic elements. It is an ASCII format and used by the MapInfo corporation.

The input data for the CityEngine is a proprietary ASCII format (comma separated list) that describes vertices and edges connecting them, together with additional parameters. Graphically these edges and vertices can be represented as a graph consisting of points (vertices) and lines (edges) connecting the points (this will be called the streetgraph).

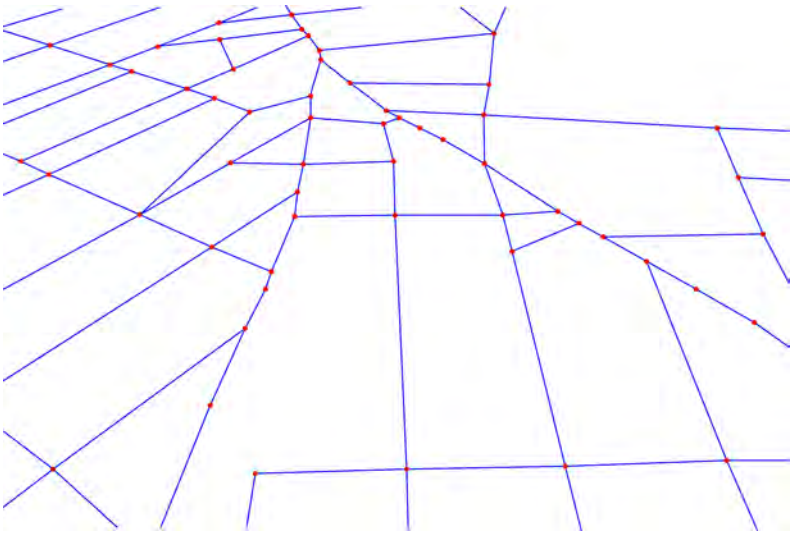


Figure 2.2: The input graph consisting of vertices and edges

2.2.2 Vertices

The vertices are the endpoints of edges. Out of each vertice an intersection is created whose type can be deduced from the number of incoming edges and their

parameters.

- If the number of incoming edges is zero, the vertice is just a point with no connections. Such a point can be ignored by the module.
- If the number is one, the vertice is an endpoint of a street. In this case there has to be crated a dead end.
- If the number is two, the vertice is a point in the central line of a street, a connection between two streettiles. In this case the adjacent streettiles have to be connected properly. Maybe there is only a change in direction, but others are also possible e.g. a change in the width of the street and/or the number of tracks.
- If the number is three or larger, the vertice is an intersection that deserves that name. In this case the type of the intersection has to be determined and the chosen intersection has to be generated.

For more details about the the determination of the type of the intersection, please refer to chapter 3.

2.2.3 Edges

Edges are the lines of the input graph connecting the vertices. An edge symbolizes a part of the street. In comparison to the vertices, edges do also have parameters attached to them which are needed for the construction of the streettile which is created from the edge, as well as the construction of the intersection that is created from each of the two endpoints (vertices) of the edge.

2.2.4 Parameters

There are several parameters that can be changed to get the desired output. In general there are two different sorts of parameters existent in the streets module:

1. As already mentioned above, there are parameters attached to each edge (see figure 2.3). Defaults of these parameters are generated at the creation of the edges by the L-system. They are mainly of geometrical nature (total number of tracks, number of left and right tracks, left and right sidewalk width, left, middle and right avenue width) together with information about the type of the street and some creation information (creation step).

Figure 2.3: The parameters attached to an edge

2. Further there are several other parameters to change the outcome of the module. They are assigned during the generation of the 3D model and include parameters about the basic creation of the geometry (e.g. the probability for a roundabout) and assignment of textures and object models. With the variation of these parameters the look of the output can be changed to make it possible to produce geographically and historically differently looking models.

2.3 Output

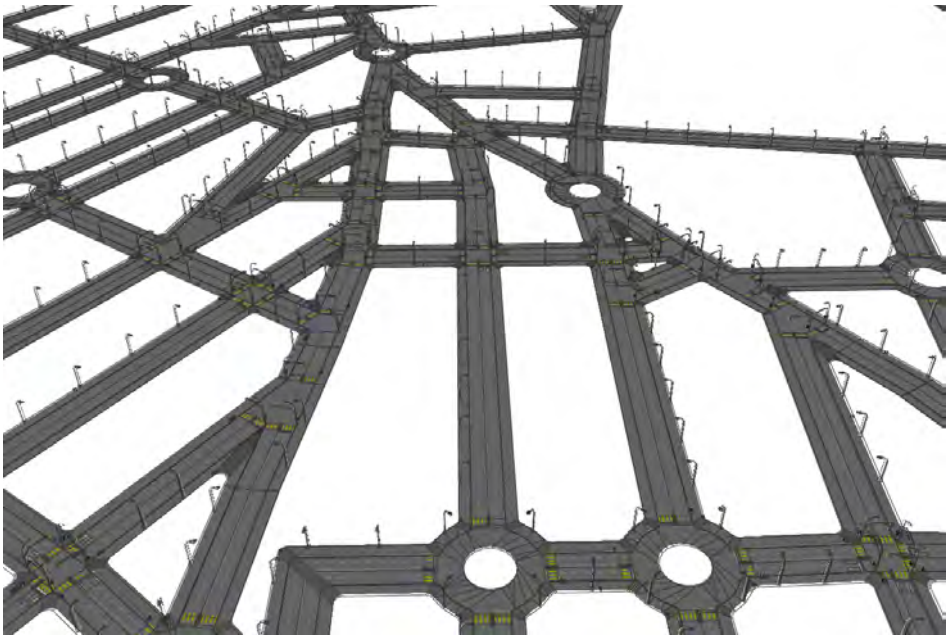


Figure 2.4: Extract of an output

The output is a textured polygonal geometry which forms, together with the outputs of the other modules, a complete 3D scene of a city.
The first subject of the output to address is the geometry. It should be properly generated to form a basis of the output which should be a good looking 3D model with correctly calculated meshpoints. These meshpoints need to be connected in the right order to form desired faces. The faces should not be of a concave shape because this could lead to problems when textures are added.
The second part is the texturing. The model should contain appropriately calculated texture coordinates so that, if good texture images are provided, a nice textured model of the street network can be rendered. The textures should not be stretched or contorted in any way if that is not necessary.
The third part are the other things that belong to the street network. These are paintings on the streets and objects placed along them. They should be placed at

the position where they belong to and their frequency of occurrence should be chosen adequately. Further their dimensions should be not too large neither too small and finally they should also be correctly aligned.
The whole output should be generated automatically. Also for a user with no modeling skills it should be possible to generate such an output with a few mouse clicks and achieve good results. By good result an output with the features stated above and producible out of virtually any possible input is meant.
Still the user should be able to change the look of the output by adjusting parameters. By changing these parameters it should be possible to generate an adequate output for as many of the different looks a street network can have as possible. This means it should be possible to generate an output of medieval character as well as an output of a modern metropolis, be it a European, an American or an Asian one.

Chapter 3

Hierarchical Description of Streets

This chapter emphasizes the theoretical part of the work. By describing the hierarchical structure of the street representation, the final requirements are broken down into sub-problems and theoretically described. This includes an explanation of the notations used as well as the organisation of the sub-problems and general solutions for them.

3.1 Streets

3.1.1 About Streets

The most important part of a street network are of course the streets themself. The main purpose of a street is the possibility for fast individual traveling. Streets have existed as long as humans needed to travel between permanent settlements, although for these first versions of streets one can probably not speak of a street but rather of a trail. Since these beginnings of street networks, their appearance has changed very much. In todays urban areas one can mainly find streets made of concrete. The name street is normally used in an urban context whereas road is usually used when speaking about a larger scale, meaning connection of urban areas (cities).
The input data of the CityEngine distinguishes between different types of which the most important are roads and highways. To have a superordinate term, the different types will all be denoted as streets. This can be justified by the fact that the module is designed for an urban area, so street would in fact be the right notation. The two main types of streets in the CityEngine will now be described briefly.

Roads

The minor streets or side streets are considered roads. They are streets in residential areas, generally rather quiet with a low traffic volume.

Highways

A highway is a major road that connects two points of interest, usually two cities. In the CityEngine they are the main streets of the network, the backbone of the streets geometry generated by the L-system.
The design of normal highways has very different forms. It goes from simple two-track streets to designs with multiple lanes for each side, a median separator and access control.
In the streets module the look of the highways will not differ from the look of the roads, however the type of the street can help when it comes to the decision of which intersection type should be created and which objects and drawings should be placed.

3.1.2 The Streettile

Streets will be formed out of one or multiple streettiles (see figure 3.1). As discussed in chapter 2, an edge has two vertices and several parameters attached to it. With this data a streettile is constructed out of every edge. Such a streettile consists of different components which shall now be described. The description will begin in the middle of the streettile and then move outwards.

Figure 3.1: A street consisting of multiple streettiles

3.1.2.1 Median Separator

The median separator is an optional part of a streettile in the middle of the street and its width is one parameter of an edge. Different possibilities for the look of the median separator exist. For example it can be out of the same material as the street and contain diagonal lines or it can be a vegetated area, maybe even with trees.

3.1.2.2 Carriageways

The carriageways are the main parts of a street. A streettile features two carriageways, one for each direction of traffic. A carriageway can contain one or multiple lanes. The number of lanes for each carriageway are two of the parameters provided for an edge. Together with the parameter describing the width of a lane, the widths of the two carriageways can be calculated. The reasons for introducing two carriageways instead of just one polygon for the street are on one hand an easier implementation of the median separator and on the other hand the flexibility for texturing that results. If a streettile would consist of just one polygon, it would have to be covered width one texture image. Considering the possible number of combinations out of the number of lanes and the widths of the median separator,

one can imagine that many texture images would have to be provided. An other possibility would have been to create a polygon for each lane. The result would have been even more flexible considering the texturing, but it would also lead to a much more complex 3D model with much higher polygon count. Keeping in mind that very large scale models are targeted the approach that considers the two carriageways but not not each lane separately was chosen.
The area covered by a streettile can be constructed out of the quadrangle that includes the median separator and the two carriageways. The four corner points of this quadrangle will be considered the main points of the streettile. The final geometry will consist of corner points adapted to possible changes of the streettile parameters.

3.1.2.3 Side Avenues

This is the area between the carriageways and the sidewalks. It is normally a vegetated area to separate the street from the sidewalk and can even feature trees to form an ally. The width of the side avenues on each side of the street are two more parameters of an edge.

3.1.2.4 Sidewalks

The outmost parts of a streettile are the two sidewalks, one for each side. Sidewalks are usually elevated in respect to the street. When a raised sidewalk is placed next to an not raised street, the border is a curb normally made from long stones. The width of each sidewalk are two additional parameters of an edge, its height is a parameter of the streets module. The decision has been made that internally only streettiles have sidewalks and intersections don't. The reason for this is that else the construction of the sidewalks of the intersections and also of the adjacent streettile could get quite complicated for several situations, compared to the construction with only streettiles having sidewalks (see figure 3.2). Because no real advantages could be found in the approach of adding sidewalks also to intersections, they are now only added to the streettiles and extended to also cover the part that otherwise would have belonged to the intersections.

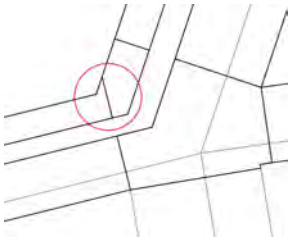


Figure 3.2: Schematic showing why intersections don't have sidewalks

Depending on all these parameters as well as the type of the adjacing intersection or the next streettile, the geometry has to be created. For more details about this construction, please refer to chapter 4.

3.2 Intersections

Intersections are junctions where different streets meet or cross. A junction utilizing grade separation is called an interchange and usually consists of several ramps. Intersections have to ensure a proper traffic flow and exist in many variants. In the CityEngine an intersection is created out of every vertex, but the vertex alone does not define its type. As already stated in chapter 2 the type is chosen depending on the number of incoming edges and their width and type. If the number of incoming edges is zero, one or two, the result is not really an intersection but rather a floating point (zero incoming edges), an end point (one incoming edge) or a connection (two incoming edges). If the number of incoming edges is larger than two however, a junction deserving the name intersection will be the result. Besides their type, intersections will also be characterized by the way they control the traffic. For information about this topic see section 3.2.2.

3.2.1 Intersection Types

For the creation of intersections in the street module three different types have been considered.

3.2.1.1 Crossing

Figure 3.3: A crossing

The crossing is the most basic type of intersection and a general rule when defining the type is: if it isn't something else, it's a crossing. The incoming streets overlap and form the central area of the crossing whose dimensions are defined by the

widths of the incoming streets and their respective angles. The central area usually features no paintings, here the vehicles will cross. To ensure a smooth traffic flow, different actions can be taken. Depending on the traffic volume to expect all the different types of traffic control can occur (see section 3.2.2). In the streets module this decision is made based on the types and the widths of the incoming edges.

3.2.1.2 Roundabout

Figure 3.4: A roundabout

The crossing of the street centerlines here forms the midpoint of a circular [1] street. The traffic all runs in one direction and this way no real crossing of the traffic is necessary. This approach features a very efficient traffic regulation but requires more space then a normal crossing. This type of intersection was not so popular in the early days of street design except for roundabouts around special monuments for example. This is probably the case because there was just no need for traffic regulation. However, this has changed over the last couple of years, and so roundabouts have become much more popular. To change the type from a normal crossing to a roundabout is quite difficult because of the addressed difference in area needed. The frequency of occurrence of roundabouts in the output of the streets module can be regulated by its own parameter. Roundabouts usually work with yield control (see section 3.2.2.2).

[1] there exist also non-circular and off-centered designs but they will not be considered

3.2.1.3 Entry

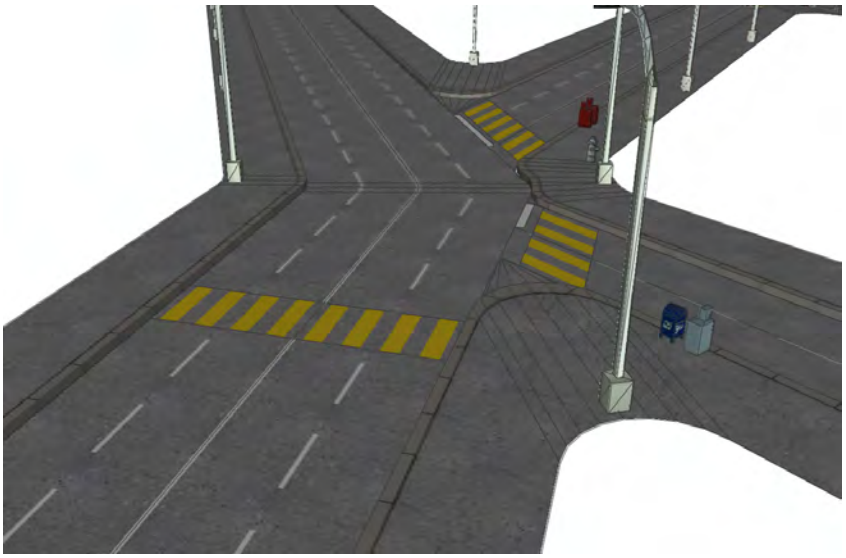


Figure 3.5: An entry

An other very common type of intersections is what will be denoted an entry. At this type of intersection, two of the incoming edges are considered superior to the others. There is no longer a large central area where the vehicles cross, but the two superior edges (denoted main edges) will run through the intersection, meaning that for example their middle marking lines will not be disrupted. So the entry will in first place operate just as a connection (intersection with two incoming edges). The two main edges will be chosen depending on the widths and the types of the incoming edges. The other edges will be considered minor edges and will flow into the street formed by the main edges. Entries will usually work with yield control (see section 3.2.2.2) or stop control (see section 3.2.2.3).

3.2.2 Intersection Traffic Control

For congestion control and smooth operation of the traffic different control instruments exist. They shall here be explained briefly.

3.2.2.1 Uncontrolled Intersections

If no large traffic volume is to be expected (for example in residential areas), no special traffic control is needed, although sometimes a warning sign is installed. The intersection rules may vary by country but normally traffic from the right has priority. If traffic is coming from opposite directions, normally the one going straight has priority over the one turning.

3.2.2.2 Yield Controlled Intersections

This kind of traffic control features yield signs and yield lines to indicate which directions have priority. In comparison to the stop controlled intersection, only slowing down but no complete stop is necessary when reaching the intersection.

3.2.2.3 Stop Controlled Intersections

This type of traffic control works similar to the yield control but the arriving road user must stop his vehicle before he may pass the intersection. This type is normally used at dangerous intersections that don't have enough traffic volume to justify the installation of a traffic sign. The stop signs can also be used for traffic calming, for example in areas where children play.

3.2.2.4 Signal Controlled Intersections

If this type of traffic control is applied, traffic lights signal which traffic has priority and which has to stop. The traffic lights phase can change depending on timers or sensors registering the presence of traffic. Traffic lights are usually installed if a intersection is busy enough so a driver has problems to keep track of what is going on around him. Especially if streets with multiple lanes are part of the incoming streets at an intersection, traffic lights are usually placed.

3.3 Drawings

Additionally to the texturing of streets and intersections, different types of drawings are needed. They represent paintings on the streets and are added separately to the textures. If they weren't added separately, a texture for each possibility of the texturing and painting of a streettile would have to be provided, which would lead to an immense quantity of necessary street textures. Also, the drawings should remain the same size no matter whether the street is wide or narrow. This wouldn't be the case because the street textures are stretched to fit the actual streettile.
Different types of drawings exist (figures 3.6 and 3.7 show examples).

Figure 3.6: A crosswalk together with a stop line and arrows

Figure 3.7: Part of a crosswalk and a yield line

3.3.1 Crosswalks

Crosswalks are needed to allow pedestrians to cross a street. They are normally painted where it is likely that a car has to stop anyway, meaning before intersections. If there is a stop line, the crosswalk is normally painted closer to the intersection center than that line, so that none of the cars that are queued because they have to stop block the passing of pedestrians. If there are yield lines the crosswalk is usually painted farther away from the intersection center because there the vehicles do not always stop.
Crosswalks do spread over both carriageways. If the street is wide enough, there is also a pedestrian island between the carriageways (see section 3.4.4).

3.3.2 Stop lines

Stop lines are used wherever the vehicles have to stop. This is before stop controlled (see section 3.2.2.3) and signal controlled (see section 3.2.2.4) intersections.
Stop lines spread over one carriageway. Normally they are placed at right angle to the direction of the streettile because the vehicles will be aligned that way and so it's clear where they have to stop. Exceptions exist of course.

3.3.3 Yield lines

Yield lines are used when a vehicle must give way to the vehicles coming from other directions. They are painted before roundabouts and sometimes at an entry of a minor road (see section 3.2.2.2).
Yield lines spread over one carriageway, as the stop lines do. They are normally placed parallel to the ending of the streettile.

3.3.4 Arrows

Arrows are painted to signal the direction in which the vehicles on a lane are allowed to travel at an intersection. If multiple lanes exist, the leftmost lane is usually reserved for left turns, the others are for driving straight on and the rightmost is also for turning right.

3.4 Objects

Besides the drawings, different objects reside next to the streets. What objects exist and where they should be placed will be shown here.

3.4.1 Lamps

Lamps are an important part of the street network. One can assume that in a city, with exception of some minor roads, all the streets are illuminated. Two different kinds of placement of lamps exist.

Figure 3.8: Examples for lamp models (low resolution variants)

- First there is the placement along the streettiles, between the intersections. The lamps are usually placed along the streettiles in steps of equal distance. If there are no avenues the lamps are placed next to the sidewalk, or to the carriageway if no sidewalk exists. Normally, they are placed only on one side of the street. If there is a middle avenue, the lamps are placed there. The side avenues are also priorized over the sides for placement, if they exist.
- Second there is the placement at the intersections. Here the lamps are placed around the intersection pointing toward s its central point. If there are also traffic lights at an intersection, the lamps can be combined with traffic lights.

3.4.2 Traffic Lights

Traffic lights are placed at crossings, if such a high traffic volume is expected that regulation is needed. Traffic lights at crossings normally feature signals for vehicles combined with signals for pedestrians.

3.4.3 Signs

Different signs can occur along the street. They can be divided into different types:

- **Regulatory signs** are used to inform about regulations. Examples are stop sings, yield signs, speed limitation signs, turn prohibition signs, traffic prohibition signs, no parking signs etc.

Figure 3.9: Examples for traffic light models

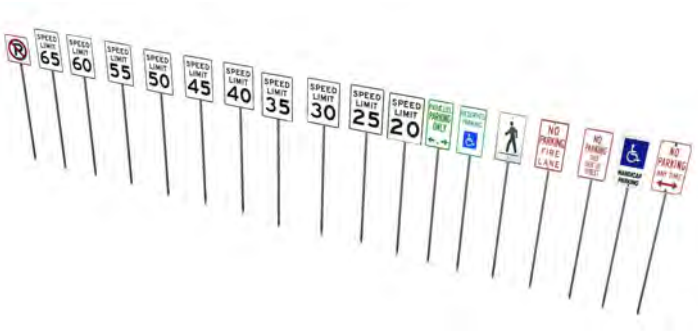


Figure 3.10: Some sign models

- **Warning signs** are used to call attention to unexpected situations. Examples are horizontal alignment or vertical grade signs, roadway condition signs, traffic regulation signs, merging and passing signs, intersection warning signs, etc.
- **Guide signs** are used to direct road users along the streets and help them along their way. Examples are destination and distance signs, route signs, street name signs, parking signs, general information signs, etc.
- **Other signs** are signs used for general information. These specific service signs, tourist oriented signs, recreational and cultural interest area signs, as well as emergency management signs.

Figure 3.11: A pedestrian islan model

3.4.4 Pedestrian Islands

The placement of pedestrian islands in the middle of the street is connected to crosswalks (see section 3.3.1). They allow pedestrians to cross only a part of a large street a time. To cross such a street usually crosswalks exist.

3.4.5 Other Objects

Figure 3.12: Some box models, a mailbox and a newsbox

Figure 3.13: Hydrant models (triangular and quadratic variant)

Different other objects can be encountered along the streets they have to be placed adequately. Here are two examples:

- **Boxes** can be found. These are for example mailboxes or newsboxes which normally reside near intersections.
- **Hydrants** are distributed along the street network so fire fighters can do their work as good as possible.

Chapter 4

Procedural Modeling of Streets

In this chapter the procedural modeling of streets and the implementation of the discussed parts of the streets module will be explained. The first part will be about the geometry construction of the different shapes. These are the different intersection types and the streettiles including sidewalks. Further there are the drawings and the different objects. This explanation will be followed by a section about the texturing of all the parts, meaning how the texture points are calculated and which texture images are applied.

4.1 Geometry Construction

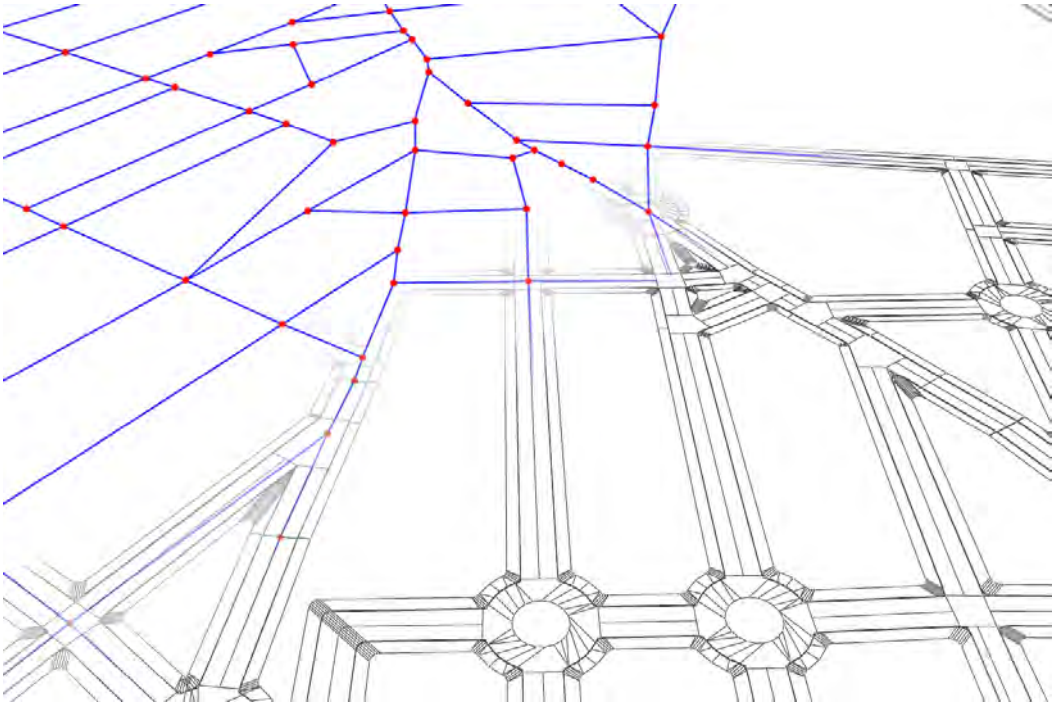


Figure 4.1: Creation of the shapes out of the vector data

The first task of the module is the construction of the 3D shapes. In the construction process, first the edge points of the shape to construct are calculated. Then faces are created on the basis of the points. Doing this one has to be careful to only construct convex faces (because concave ones can sometimes not be handled).
As shown in section 2.1 the intersections are constructed first and this for the following reason. The corner points of the streettiles depend heavily on the intersection they are situated next to, for example the streettile next to a roundabout begins the size of the outer radius of the roundabout away from it. While creating the intersections, their corner points are also stored as the corner points of the respective streettile yet to build.
The carriageways are created before the sidewalks because the sidewalks depend on the carriageways. They are situated the side avenue width away from them.
Because of this sequence in the construction, also the description will be in this order.

4.1.1 Intersections

As already seen in chapter 3 there are three different types of intersections, crossings, roundabouts and entries, which feature different shapes. How the geometry of these different shapes is created will now be explained.

4.1.1.1 Crossing

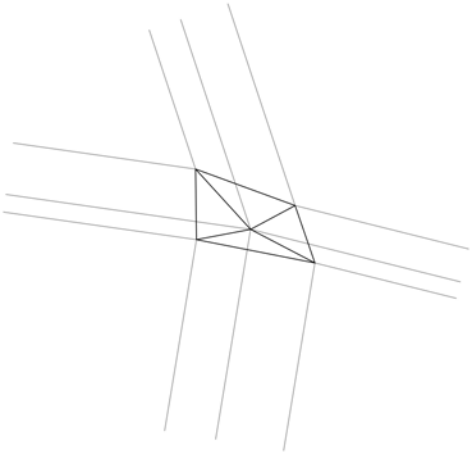


Figure 4.2: Schematic of a crossing without mirroring applied

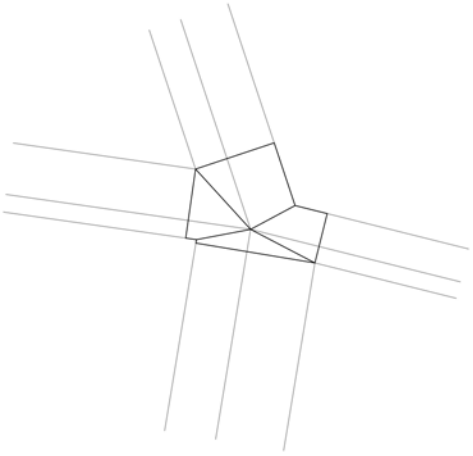


Figure 4.3: Schematic of a crossing with mirroring applied

The first possibility of an intersection type is the crossing. The shape to construct is the central area obtained by the overlay of the incoming streettiles. To find this area, the points where the outer lines of the streettiles cut [1] have to be found (how this is done can be read in A.1). It is thereby important to calculate the points in the right succession in order to allow simple creation of the faces by just connecting one point after the other. The calculated points connected leads to a polygon which represents the desired area (see figure 4.2).
At an intersection there is normally a queuing zone followed by a stop line especially if the traffic flow is controlled by traffic lights. To complete the shape of the intersection, it is therefore desired to have an ending of the adjacing streettile which is at right angle to the direction of the edge. To achieve this the found cutting points have to be mirrored forward or backward, depending on their position (the criteria and calculation can be found in A.2).
Having applied also the mirroring of the points one ends up with the desired shape of a crossing (see figure 4.3).

[1] Probably 'intersect' would be a better word to describe this event, but to avoid a mix-up with 'intersection', the term 'cut' will be used instead

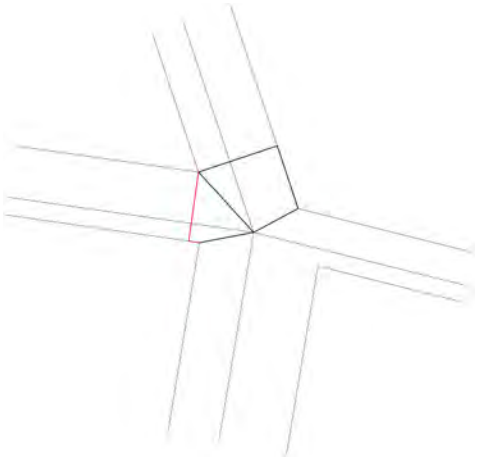


Figure 4.4: Mirroring forward

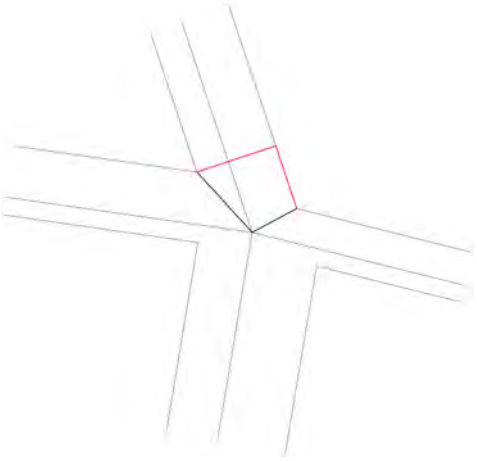


Figure 4.5: Mirroring backward

However there are some border cases where problems can occur. The first problem occurs for an angle of 180° (or at last close to 180°) and widths that are not the same for the two edges in question. There, a cutting point can not be found or is far away from the intersection. For this case, a special treatment is necessary. An other problem is a very acute angle. Also in that case the resulting cutting point is far away resulting in a very large central area. To avoid this problem, the input vector data should be generated carefully.

4.1.1.2 Roundabout

An other possibility of an intersection is the roundabout. For the roundabout, first an outer (whole roundabout) and an inner (central island) radius is specified. The radii are chosen depending on the widths of the incoming streets. Having them specified, the cutting points of the incoming streets are generated the same way they are in the crossing case. As a difference, for each of the found points the distance to the central point is then checked.
If it is smaller than the outer radius it lies inside the roundabout (see figure 4.6). In this case, first a tangent to the outer radius with length equal to the width of the carriageway of the first adjacing streettile is generated. Then an arc around the central point with the outer radius as its radius and going from the just generated tangent to where the tangent to the next edge starts is added. To add this arc, first its starting and the ending angle have to be calculated (see A.3 for more information).
If the distance is larger than the outer radius, the point falls outside of the roundabout (see figure 4.7). This means that the dimension of the intersection is to large to fall inside the outer circle of the roundabout. In that case, the construction is is equal to the construction of a crossing.

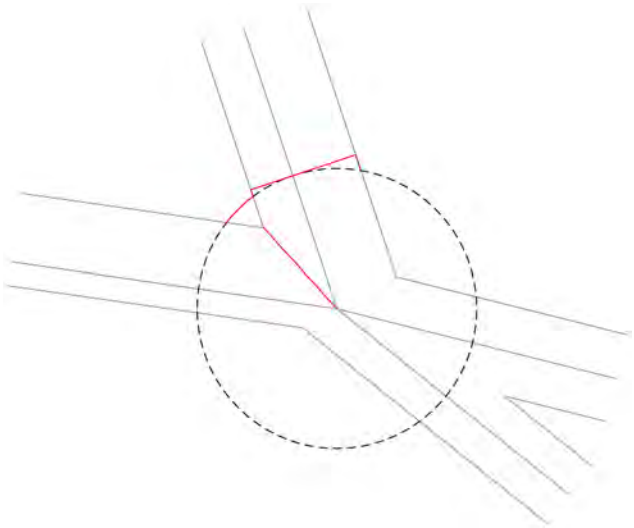


Figure 4.6: Resulting part of the shape for a distance smaller than the outer radius

Finally a circle around the central point with the inner radius is added to form the

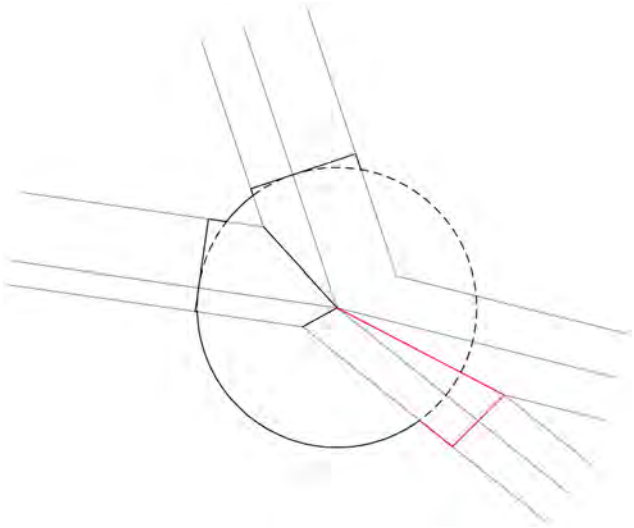


Figure 4.7: Resulting part of the shape for a distance larger than the outer radius

central island of the roundabout and complete the construction of the points of a roundabout (see figure 4.8).

For a roundabout the creation of the faces has to be conducted carefully to avoid concave ones. For this reason always quadrangular or triangular faces are formed by taking one or two points from each, the inside and the outside arc, and making sure that on the inside and the outside arc the process of creating faces has advanced approximately by the same extend (same angle from the starting position).

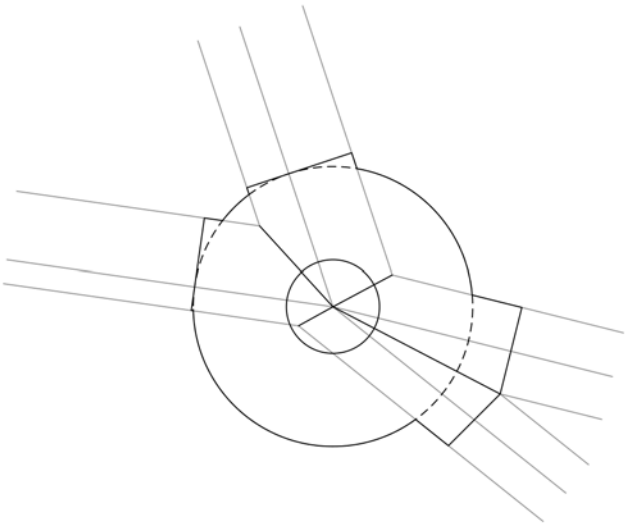


Figure 4.8: Finished construction of the shape of a roundabout

4.1.1.3 Entry

The entry is the most complicated case of an intersection. First, two incoming edges are promoted to be the main edges. These are representing the main road that is not interrupted by any part of the central area. In the case of an entry it is tried to keep the central area as small as possible. The incoming streettiles are not cut at right angle in this case because here there are usually no traffic lights, controlling the traffic flow and no queuing zone, just yield signs or stop signs. The reason for this is, that cars coming from a minor streettile will have no priority and must let the ones on the major street pass.
The first construction step is the creation of the main street (see figure 4.9) that will just run through the intersection.. These are two points constructed as if the intersection type was a connection (two incoming edges).
Then the cutting points of the incoming minor streettiles have to be calculated.

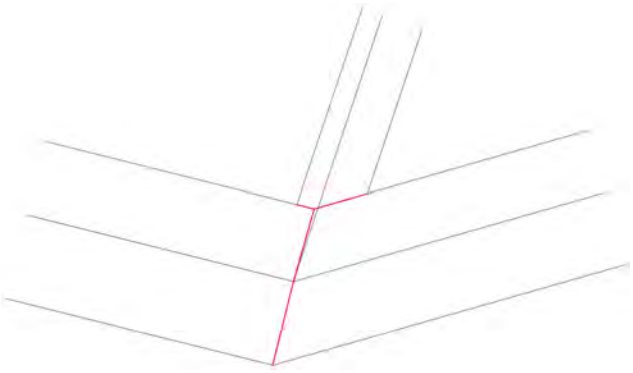


Figure 4.9: The main streets of the entry.
It is no longer clear from the beginning, where the cut will first occur. There are

several possibilities how an incoming minor edge can cut the main edges.

- The border lines of the minor street in question can cut the main street just the way just as if a crossing was to be constructed, meaning the left border line of one of the main edges will cut the border line on the right side of the minor edge and the border line on the left side of the minor street will cut the right border line of the other main street (see top of figure 4.10). If the angle from the first to the second main edge is smaller then 180°, the resulting part of the central area of the entry is the triangle of the two cutting points together with the cutting point of the left border line of the first main street and the right border line of the second main street.

- The cuts can be the same as described above but the angle between the first and the second main edge larger than 180°. In this case two triangles result because the line that symbolizes the cut between the streettile of the minor edge and the entry has to be shifted parallely so the minor streettile and the two main streettiles don't overlap (see bottom of figure 4.10).

- The minor street can also cut the first main street as in the cases above but on the other side also cut this first main street again instead of the other main street. In this case the resulting part of the central area has actually area zero because the minor street can be fit to the first main street perfectly (see right side of figure 4.11).

- The minor can also cut the second main edge first and on the other side the second main street again. In this case also an area of zero results, the minor street can again be fit to the first main street perfectly (see left side of figure 4.11).

- Finally the minor street to be constructed can also cut an other minor street on one or both sides. In this case the contribution to the area of the intersection is a polygon which consists of the cutting point of the minor streets together with the cutting points with the main streets and, depending on which main streets are cut, also a cutting point of the two main streets (see figure 4.12).

One can see that many different possibilities exist of how the cuts can occur in the end and it is not always clear in advance which type it's going to be. So before calculating the real cut, it has to be found out which one it is.
During the construction it has to be ensured that the main streettiles are not touched in any way when it comes to the construction of faces. The minor streettiles can flow into the main road from both sides of the main street. If the side on which the main street running through is cut, is changed from one minor road to the next, additionally the cutting points of the main streets have to be added to the central area to indicate the change.

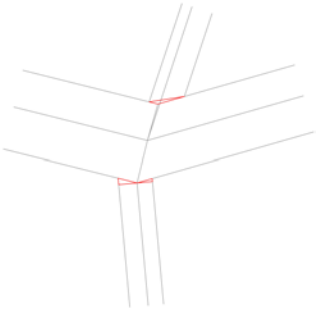


Figure 4.10: Minor streets cutting different major streets on their two sides.

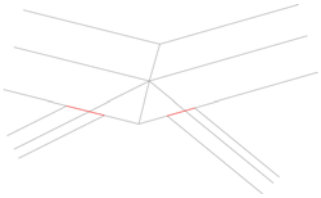


Figure 4.11: Minor street cutting the same main street twice.

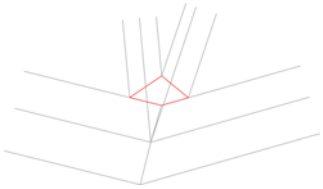


Figure 4.12: Minor street cutting an other minor street.

4.1.1.4 Connections and Endpoints

Connections and endpoints are two special intersection types.

- A connection is the intersection that results from a vertex that has just two incoming edges. In it's construction the basic points for the two incoming streettiles are defined.
- An endpoint is the intersection resulting from a vertex with one incoming edge. Here also the basic points for this edge are generated.

Because these types of intersection have no area it is not necessary nor possible to apply any texture.

4.1.2 Streettiles

Streettiles are the parts of a street between the intersections and are constructed for each edge. To the streettiles also belong the sidewalks and the curbs.

4.1.2.1 Carriageways

The points for the streettiles have to be generated depending on the adjacing intersection. The basic points for the streettile are created when constructing these intersections. They can be considered the corner points of the streettile, if it would just be a quadrangle with the correct length and a width equivalent to the sum of the the widths of the two carriageways and the median separator. For reasons stated in chapter 3 the streettile includes a polygon for each of the two carriageways which have to be constructed based on the basic points. This is done using the widths of the carriageways and the median separator. Summing them up, the

total width of the street is known which is also represented by the the basic points. Of course the distance of the basic points is larger than the total width if the streettile doesn't end at right angle but using the factor of the width of the carriageway divided by the total width, the correct position for the points can be calculated. Constructing the streettiles this way would result in two carriageways having the correct width and separation from each other, each being a quadrangle (two points per neighboring intersection).
There are however several situations where additional points are necessary as well as adaptations of the basic points.

- If the neighboring intersection is a 'real' intersection (has more then two incoming edges), the outer points don't have to be adapted but because the total width was considered when constructing the intersection. A possible middle separator will end there. Points will have to be added and the found points have to be adapted on the inner side of the carriageways to ensure a smooth ending of the middle separator (if there is one).
- If the neighboring intersection is a connection (two incoming edges) and the width of one or both of the carriageways is changed, the inner points don't have to be adapted but points will have to be added and the found points have to be adapted on the outside to ensure a smooth transition of the outer borderlines of the carriageways (see figure 4.13).
- If the neighboring intersection is a connection (two incoming edges) and the width of the median separator changes, additional points will have to be added and the found points have to be adapted on the inner side of the carriageways for smooth transition of the median separator as well as on the outer side, because a change of the median separator also results in a change of the overall width of the street (see figure 4.14).
- The two cases of changes for the neighboring intersection being a connection can also be combined (width of carriageways and median separator changes). Here also additional points and adaptations of the found points on the inner and the outer side are necessary (see figure 4.15).

Figure 4.13: Schematic of the carriageways if their width changes

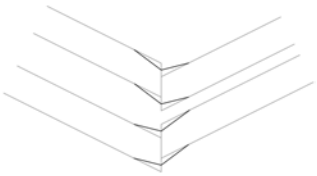


Figure 4.14: Schematic of the carriageways if the width of the median separator changes

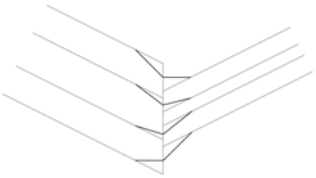


Figure 4.15: Schematic of the carriageways at a connection if both widths change

After the the points have been calculated, the faces need to be created. Here it is again important to avoid the generation of concave faces. If possible quadrangles, otherwise triangles are created.

4.1.2.2 Sidewalks

Further away from the centerline, more and more parameters have to be taken into account, which leads to a more complex calculation of the correct points. For the construction of the sidewalks besides the width of the medium separator and the widths of the carriageways, also the widths of the avenues and the sidewalks have to be considered. For the construction of the sidewalks first the points at the ends of the edge are created. How they are created depends on the neighboring intersection.

- If it is a connection, two endpoints have to be added (shown in figure 4.16). The basic endpoints are created to be the avenue width outside the outer carriageway point (inner point of the sidewalk) and avenue width plus sidewalk width outside the outer carriageway point (outer point of the sidewalk). As in the creation of the carriageway points, different cases have to be considered where adaptations of these points are necessary. If the sidewalk width changes, an adaptation of the outer point is necessary, if the avenue width or the avenue width and the sidewalk width is changed, the inner and the outer points have to be adapted.

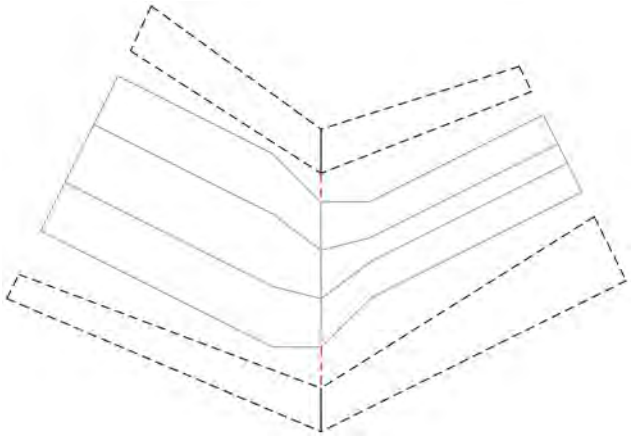


Figure 4.16: Main sidewalk points for at a connection

- If it is a crossing, also two endpoints have to be added and they can be calculated in a way that no adaptations are needed later (see figure 4.17). The calculation is the same as for the points of the crossing (see A.1) except that now the widths are the widths for the calculation of the crossing plus the avenue width (for the inner point) or plus the avenue width and the sidewalk width (for the outer point).

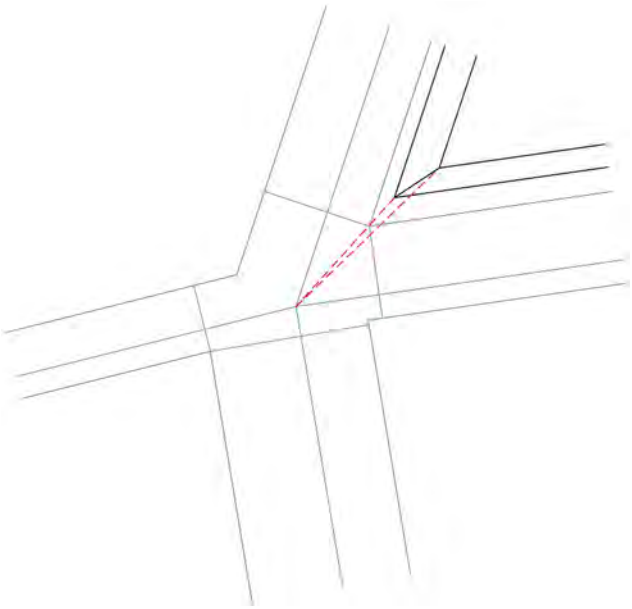


Figure 4.17: Main sidewalk points at a crossing

- If it is a roundabout, often more points are added. If no arc was added to the intersection when creating it (distance to the cutting point larger than the outer radius), the procedure for sidewalk points generation is equal to the one of the crossing. Else, arcs also need to be added to the sidewalks (one on the outer side and one on the inner side, each with adapted radius). If the sidewalk width changes the radius of the outer arc will have to be adapted, if the avenue width or the avenue width and the sidewalk width change, the radii of inner and outer arc will have to be adapted for a smooth transition (see figure 4.18).

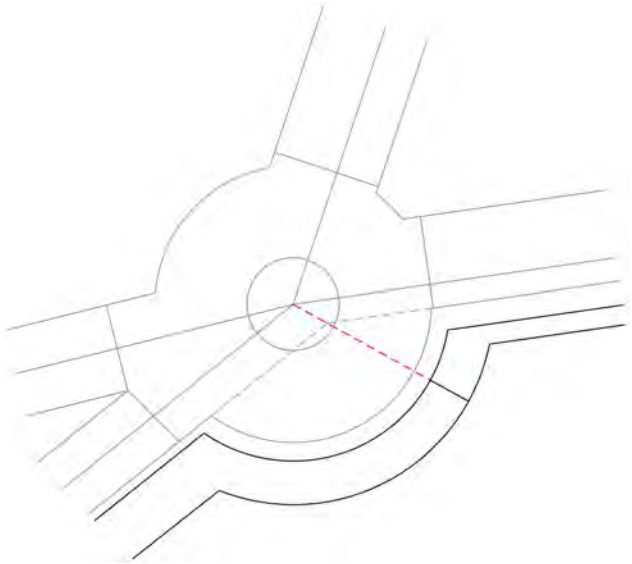


Figure 4.18: Main sidewalk points at a roundabout

- If it is an entry, things can get quite complicated. As already described in section 4.1.1.3 there are different cases of how the minor street can cut the main street or other minor streets. Based on how the street to which the sidewalks that are to be created cut, points will have to be added. The method for the adding of points will be the same as the one used for sidewalks of crossings (again formula described in A.1). The problem here is as already during the creation of the intersection, that it's not clear from the beginning, which cut is going to occur. Three basic possibilities of how points have to be added were found.
  - The simplest possibility is that the cut is the same as it would be for a crossing. For this case the points just have to be added the way they would for a crossing (see right side of figure 4.19).
  - An other possibility is, that when for the crossing case it would cut the first main edge, it first cuts the second. Then the sidewalk has to flow around the corner of the main street, meaning that also the sidewalk points belonging to the corner of the main streets have to be calculated and added to allow this flow around (see left side of figure 4.19).
  - The third possibility is that two minor edges are next to each other but they first cut a main street. In this case, besides the cutting points of the minor edge and the main edge, additional points, which are part of the sidewalk of the main street, have to be added to form the ends of the sidewalks(see middle of figure 4.19).

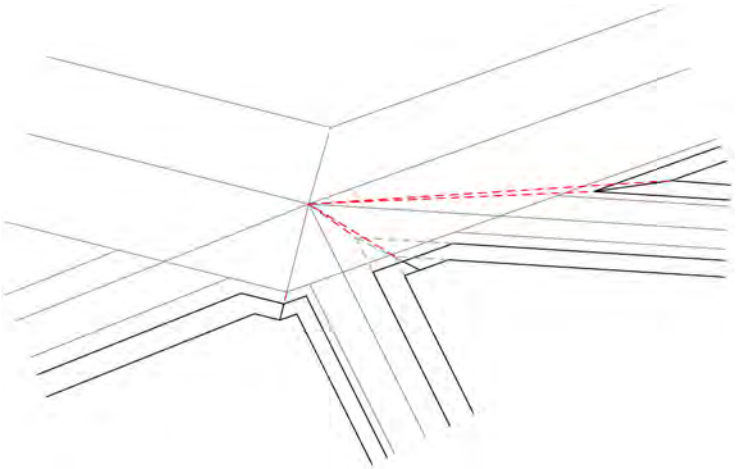


Figure 4.19: Possible sidewalks at an entry

When these basic points are added, the middle points are created. They can occur because of two reasons. First, because the carriageways have additional points added to themselfs (because of width changes). If this is the case correctly shifted

versions of these points have also be added to the sidewalk to ensure a constant avenue width. Second, there may be additional points necessary because the avenue width and/or the sidewalk width changes from one streettile to the next.
After the calculation of the points also here the faces are created. The same proce-

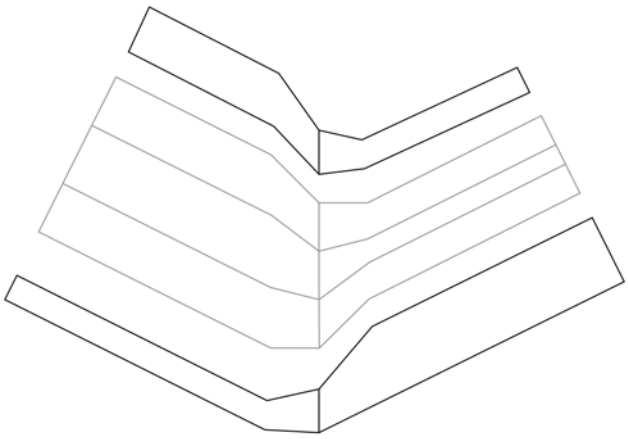


Figure 4.20: Sidewalks at a connection with added middle points

dure as for the creation of the faces of the carriageways is used to create the faces of the sidewalks.

4.1.2.3 Curbs

Because the sidewalk has a certain height, also a curb is necessary. Its construction is quite simple, every point of the sidewalk facing a lane is also part of the curb and this twice. Once with the height of the sidewalk and once with the height of the carriageway. Always four of those points, namely two with sidewalk height and the two respective ones with carriageway height, will then form a rectangle. These rectangles are the desired faces.

4.1.3 Drawings

The shape of the drawings is a simple quadrangle which has to be fit to the according streettile. There are different possibilities of how to fit a drawing. The following parameters exist.

- One parameter determines to which end of the streettile the drawing should be fit.
- An other one specifies the dimensions of the drawing. Different possibilities exist:
  - It can spread over a lane of a carriageway. This type is used for arrows which are just in one lane.

  - It can spread over one whole carriageway. This type is used for yield lines and stop lines, which spread over all lanes of a carriageway.
  - It can spread over both carriageways. This type is used for crosswalks

- There is also a parameter to assign an offset from the beginning of the streettile. Normally lines are close to the beginning whereas crosswalks are farther away.
- Finally, there is a parameter to define how the drawing is fit to the street. The following possibilities exist:
  - Parallel to the ending of the streettile as a rectangle. This is used when the drawing should be parallel to the ending of the streettile but should not be distorted.
  - Parallel to the ending of the streettile as a parallelogram (a distorted rectangle). Here the image is distorted as in opposition to the possibility above.
  - Normal to the direction of the streettile. Used when the image should not be parallel to the ending but normal to the direction of the streettile. The drawing is hereby also painted as a rectangle (not distorted).

Based on these parameters the drawing will be fit onto the street and then be textured.

4.1.4 Objects

The objects are already modeled and just have to be loaded into the module as 3D objects. The module then has to place them at adequate positions as well as scale and align them. For information about what objects are placed along the street network and where they are placed, refer to section 3.4.

4.1.5 Rounding

Rounding is a special feature of the module to smooth the corners of the constructed shapes. Herby arcs are calculated for each cutting point to eliminate spiky corners. The rounding radius can hereby be specified as a parameter. To create the arcs, first the middle point and the starting as well as the ending angle has to be calculated (see A.4 and A.5 for details). Then the points have to be added at the right position. If the points are added to an intersection, special indication is needed for proper face generation afterwards. Therefore first the original cut point is added, followed by the arc points and again the original point. For streettiles and sidewalks, the arc points are just added additionally to the other basic points.
During the face creation for intersection, always triangles are created when creating faces including points of the rounding arc. These triangles always consist of two points of the arc and the original point. This way it is again ensured to always get convex faces.

4.2 Textures

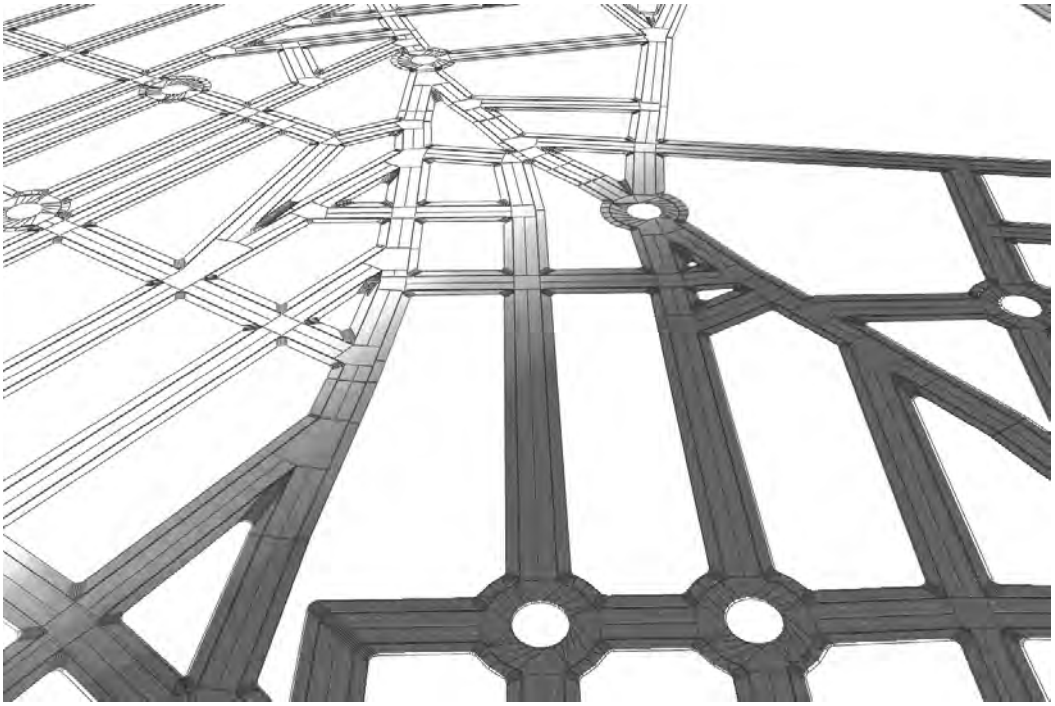


Figure 4.21: Texturing of the created shapes

The desired output is a 3D textured polygonal geometry, which already implies that also texturing is needed. Texturing means that a 2D image is projected onto the 3D shape to determine the look of the surface. First, suitable texture images have to be selected. Then the correct texture coordinates for each point of the shape have to be calculated. These are 2D coordinates (u, v) which represent the position in the texture image. According to these coordinates one point in the 2D texture image is mapped to a 3D point of the shape. If the texture coordinates are calculated incorrectly the texture will be stretched and contorted. This is undesirable, especially if there are any well defined structures on the texture image, painted lines or drawings of the curbstones for example.

4.2.1 Intersections

For the texturing of intersections it is assumed that the central area does not contain any markings nor paintings. If some paintings would be needed, they could be added using drawings. With this restriction it is a quite easy task to calculate the texture coordinates for intersections because the texture image has just to be projected onto the central area (maybe repeated if the dimensions of the image are

small) and don't have to be stretched or distorted in any way to fit the shape. This is applicable for the central areas of all intersection types.

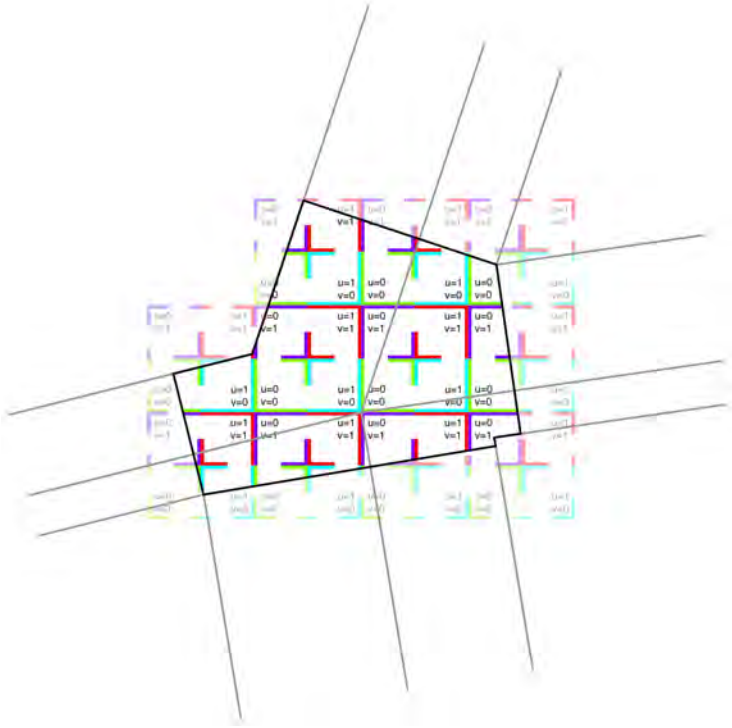


Figure 4.22: Texturing of an intersection (crossing as example)

4.2.2 Streettiles

For the streettiles the calculation of the texture coordinates is more complicated because here specially painted texture images are used. The texture images of the carriageways include the paintings of the lines on the streets, the sidewalk texture images include a drawing of the curbstones.

4.2.2.1 Carriageways and Sidewalks

At the beginning of this thesis it had to be decided whether to use a uniform background texture for the carriageways and sidewalks as it is implemented at the intersections and use shaders to draw the needed street markings or curbstones, or whether to just use texturing, which is how it is solved now. An advantage of the method using shaders would be that the background texture does never have to be stretched and contorted, it could just be projected onto the shape like in the intersection case. This stretching and contortion occurring in the case without shaders

can indeed be a problem. For example when a texture image designed for a relatively narrow street is applied to a much wider street, also the lines on it become wider which is not desired. An other advantage of the method using shaders is, that only one background image texture would be needed. Despite these advantages the method using different textures was chosen, because the use of shaders is quite complicated and because its implementation differs quite a lot when using different rendering software, while texturing works quite in a standardized manner.
The procedure for the calculation of the texture points of the carriageways and the sidewalks is based on the same principle. First the overall length of the inner border line is calculated by adding up all the distances between the shape points on this side. This length is then divided by the length one texture image represents and all the digits behind the comma are cut (so the result is an integral number). This number plus one (to always have at least one texture image painted) is the number of texture images that are placed in a row to form the texturing. In the width there is always just one texture image.
The u-coordinate of the inner border line are now calculated by dividing the added up distance to a certain point by the total length and multiply the number of textures with this factor. The v-coordinate is always zero. To find the u-coordinate of the outer border line is more difficult because the streettile does't have to end right angled. For this reason, also the elongation or shortening of the outer borderline has to be taken into account (see A.6 for formula). The v-coordinate is then always one.
With this texturing method there will often be some stretching and contortion of

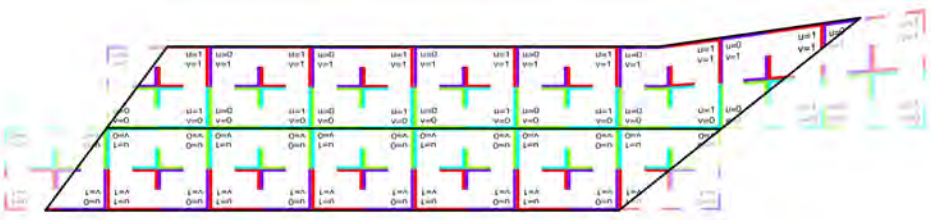


Figure 4.23: Texturing of carriageways and sidewalks

the texture, especially for complex structures as the arcs around roundabouts or the flowing around corners at entries, but it is tried to to eliminate them as much as possible to get a good looking result.

4.2.2.2 Curbs

Because the curbs feature nice rectangular faces, the calculation of the texture points is also rather simple. The u-component can be taken over from those of the inner side of the sidewalk. This also ensures that subdivision stays the same as for the sidewalks. This means that if there is a transition from one curbstone to the

next in the sidewalk model, there is also one in the curb model provided that the textures are painted properly. The v-components are just zero for the points at the height of the carriageways and one for the points at the height of the sidewalks.

4.2.3 Drawings

Here the texture image is the painting of the desired drawing. This image is then fit onto the quadrangular shape of the drawing. The texture coordinates of the four corner points of this shape are also the corner points of the texture image.

4.2.4 Objects

The objects already come with the correctly calculated texture points. All that is left to do is to specify the right texture image.

Chapter 5

Results and Conclusion

In this chapter, first an overview over the achieved results will be given. This includes some pictures of two implementation examples and a discussion. Finally it will be stated what next steps could be undertaken to further improve the developed streets module.

5.1 Results

In the course of this work different goals have been achieved.

1. The first major goal was the generation of the 3D geometry. Here the module now features a solution to create such a geometry for virtually every possible input. Different types of intersections have been implemented to guarantee a good diversity and to provide an adequate solution for every input. The intersection types are crossings, roundabouts and entries. Between those intersections streettiles including sidewalks, middle avenue and side avenues are generated, changes in width or direction are smoothed and corners are rounded to provide a smooth look.

2. The second goal was to texture the produced geometry. For this the module provides a solution that needs only few different texture images, because a polygon is constructed for each carriageway. The intersections as well as the streettiles and the curbs feature texture coordinates which are calculated to allow the use of as few different texture images as possible and still only stretch them where no other solution is possible. Additionally, methods for the simple adequate distribution of different drawings are provided. They also use textures making the more difficult to handle shaders unnecessary.

3. A third goal was to position objects along the street network. To allow this the module provides methods for the loading of external models and for the simple positioning of them.

The implemented module was tested for different input data to check if it produces the desired output. On the following pages two applications of the developed module will be presented.

5.1.1 Smallville

The first application presented is a visualization of a rather small town. The streets are not very wide, they usually consist of one lane per carriageway and the houses have a slight European touch with gabled roofs. The city will be called 'Smallville'. Figures 5.1 - 5.3 show some screenshots of a larger area of the Smalville model. Basic models of buildings have been added to give an impression how the surrounding city would look like.
Figures 5.4 - 5.10 show a smaller area with more details visible. In some pictures, drawings and certain objects are not added.

Figure 5.1: Smallville: Overview 1

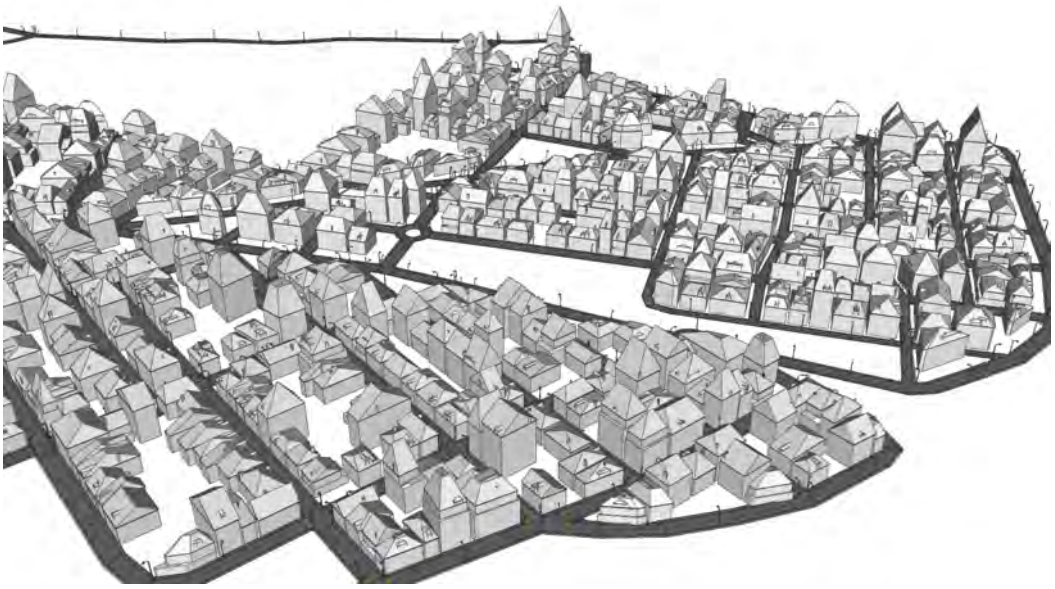


Figure 5.2: Smallville: Overview 2

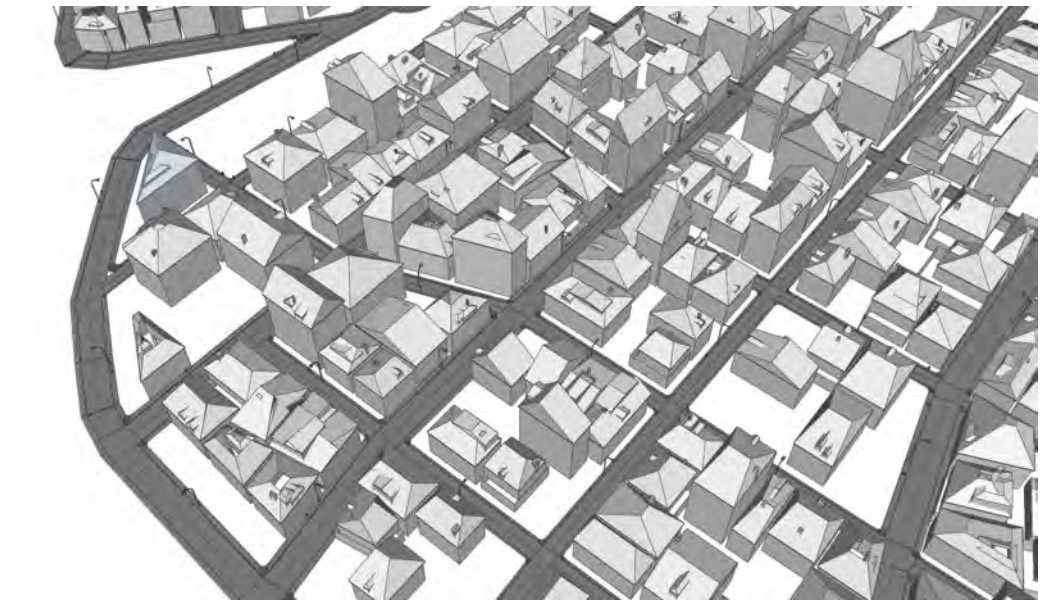


Figure 5.3: Smallville: Overview 3

Figure 5.4: Smallville: Closer look

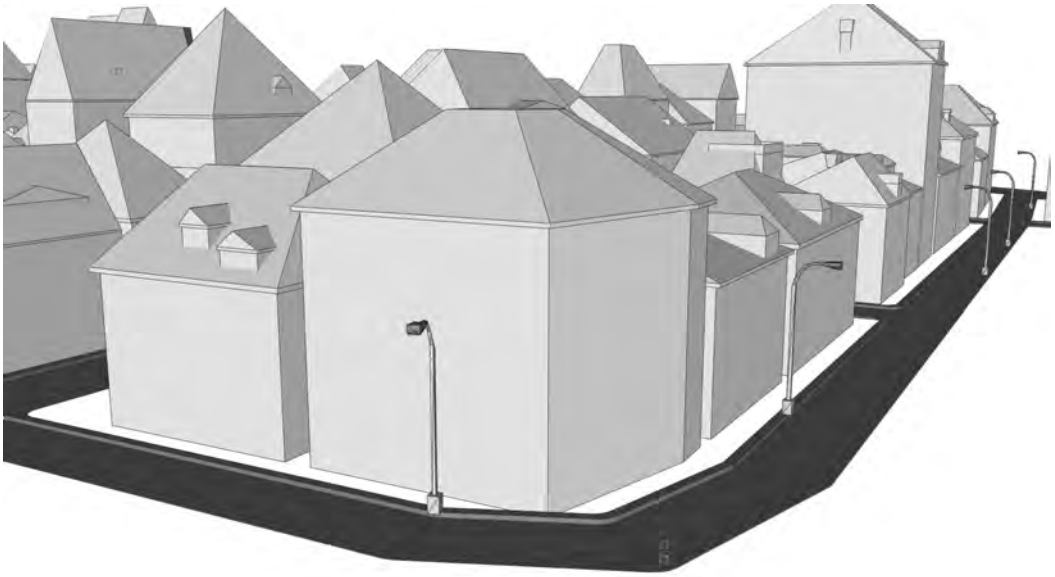


Figure 5.5: Smallville: Extract without drawings and only some of the objects

Figure 5.6: Smallville: Extract with drawings and all objects

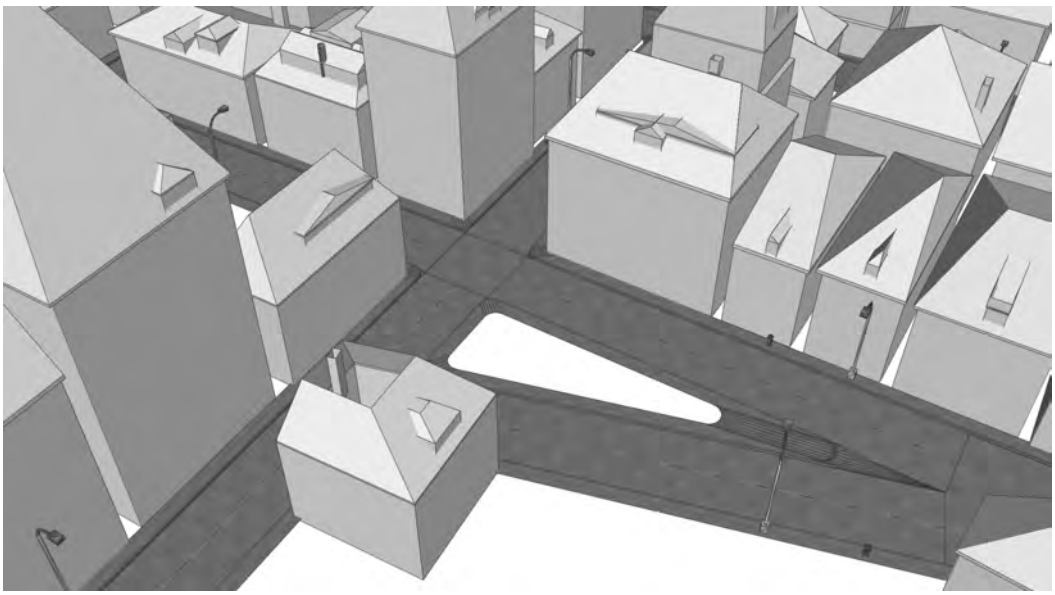


Figure 5.7: Smallville: Second extract with drawings and some objects left out

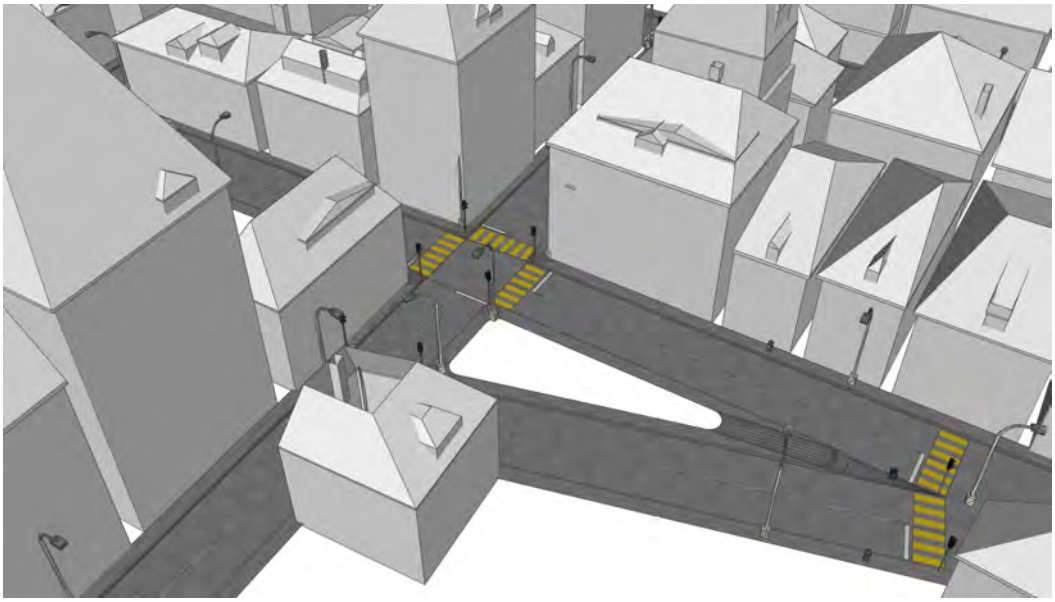


Figure 5.8: Smallville: Second extract with drawings and all the objects added

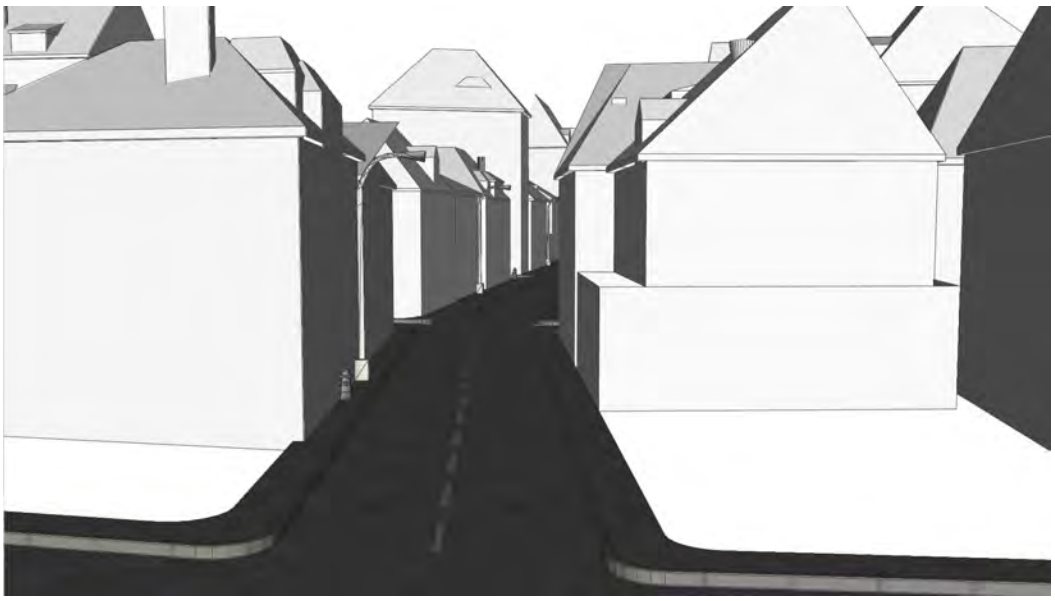


Figure 5.9: Smallville: Another extract

Figure 5.10: Smallville: A last picture

5.1.2 Metropolis

The second city is a large one with skyscrapers and will be called 'Metropolis'. The streets are much wider and consist of multiple lanes per carriageway. The scene of this model features far more vertices and edges than the one before. For this test data the streets model alone, without any buildings, already features over 3.5 million polygons and this even though only very low resolution models of the objects were used to minimize polygon count. For example the used hydrant model consists of 39 polygons whereas the high resolution version features nearly 11000. One can easily imagine that the computing power necessary to create such a large model is demanding. One must also keep in mind that the street network is just the very beginning of the model of a whole city. For a complete model also outputs of other modules as the buildings module or the vegetation module have to be added.
Figure 5.11 again shows an overview of the model.
Figures 5.12 - 5.14 allow a closer look of the output. Again, basic models of buildings have been added to give a better impression of how a model including the output of all the different modules would look like.

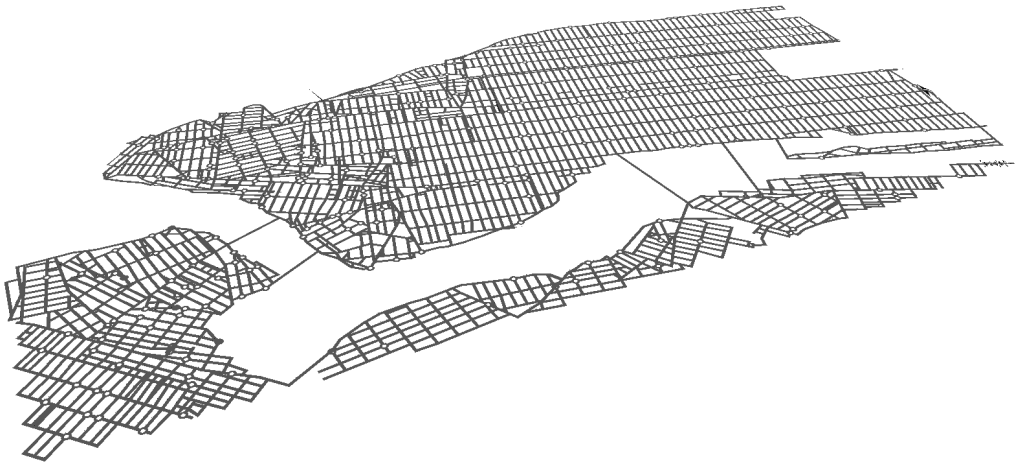


Figure 5.11: Metropolis: The overview

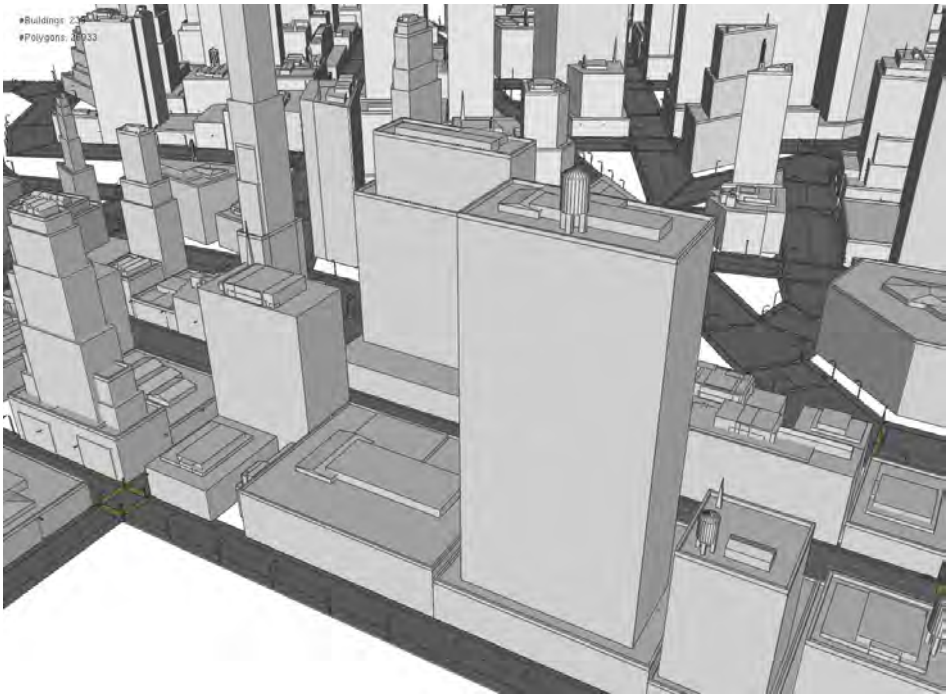


Figure 5.12: Metropolis: Closer look 1

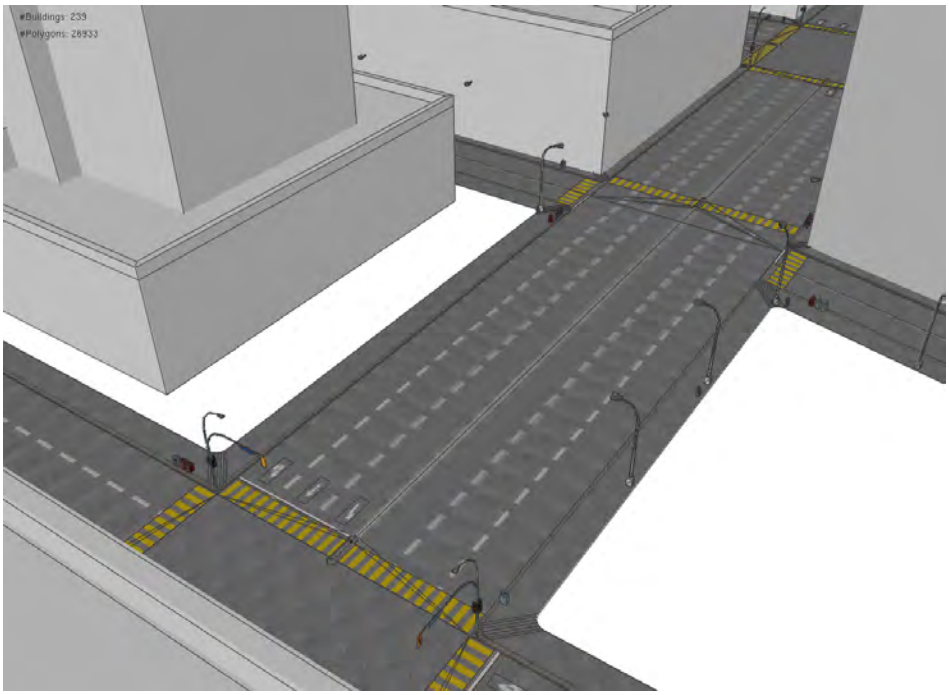


Figure 5.13: Metropolis: Closer look 2

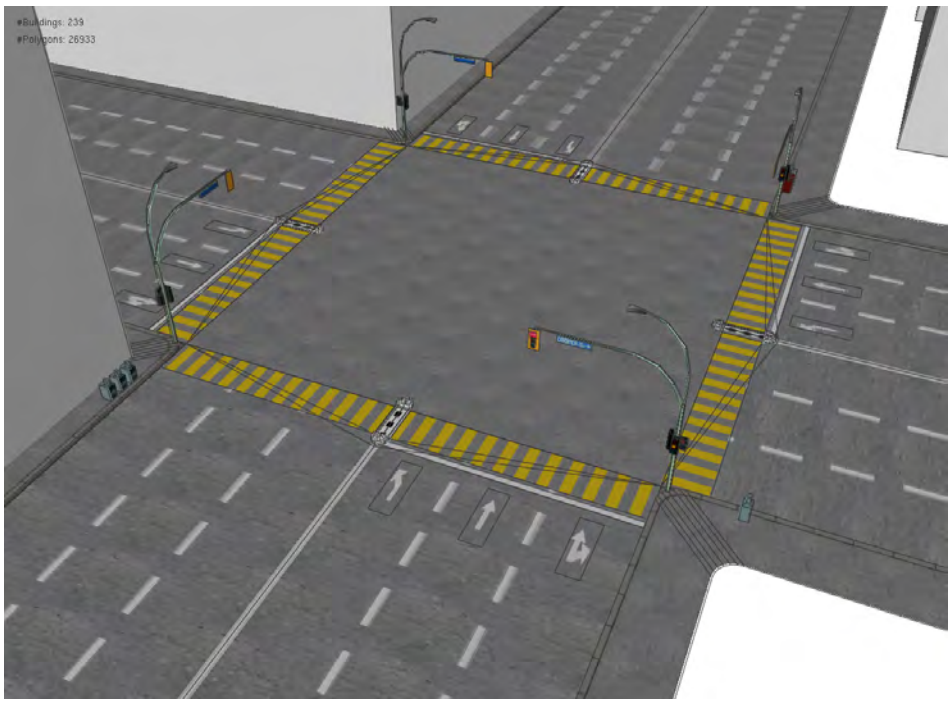


Figure 5.14: Metropolis: Closer look 3

5.2 Outlook

This project has shown, that the automatic construction of a street network just out of vector data is possible. However, different tasks could be conducted to further perfect the module:

- **Merging** of overlapping intersections could be implemented. This algorithm is needed to produce a nice output even if the L-system generates vector data that result in a model that is normally not possible, for example vertices closer than the street width of the adjacent edges or edges with a distance smaller then their street width[1].
- **Freeways** and with them their special intersection types (interchanges) could be implemented. For this also a good definition of which streets should be promoted to freeways has to be found.
- The use of **Height** could be improved. Until now the streets don't feature a good height treatment. Solutions for not steadily changing height between intersections should be found and implemented. Possible ones would be to adapt the underlying height model or to add an embankment to the street-tiles.

[1] In fact such an implementation was begun but not finished due to timing problems

- **Level of detail (LOD)** could be introduced. To allow the display of massive scenes as well as detailed closeups, different levels of detail should be implemented.
- An **Integration into the RenderMan pipeline** could be implemented. This includes the application of shaders and lights (traffic lights and lamps) as well as the object handling with instances.
- A **generic approach with a street grammar** could be implemented. Meant by this is a high level encoding of the different street layouts. This encoding would then control the whole process of the generation and apply appropriate parameters to this adapt the the look of the resulting model. Also included could be a traffic simulation on the basis of which a better model could be created. For example such a simulation could be used to apply the correct number of lanes, for the placement of adequate signs and other objects as traffic lights etc.

Appendix A

Notes And Formulas

A.1 Finding the point where the outer lines of shapes cut

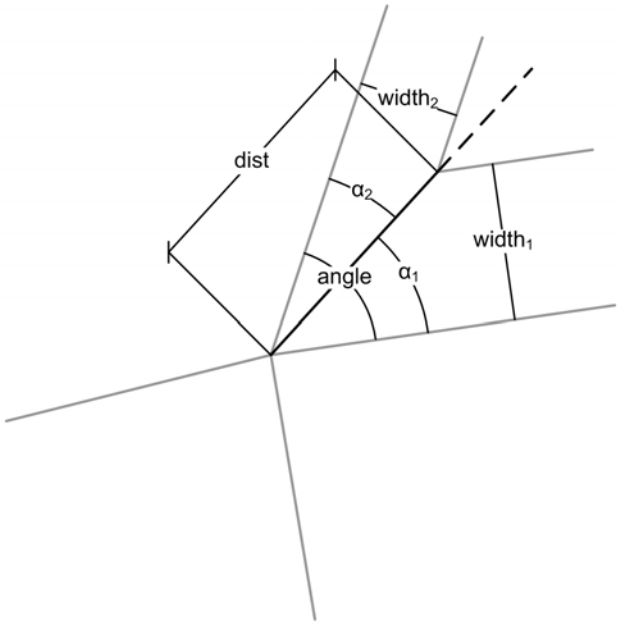


Figure A.1: Finding the point where outer lines cut

This formula is used wherever a cut of shapes has to be found. The known parameters are the two widths (width $_1$, width $_2$) and the angle (angle) between the edges. The unknowns are the angle to the line that connects the midpoint with the searched point (α_1) and the distance (dist). The solution is found using the following two equations.

$$dist * \sin(\alpha_1) = width_1 \tag{A.1}$$

$$dist * \sin(angle - \alpha_1) = width_2 \tag{A.2}$$

Solving these equations yields:

$$\alpha_1 = \arctan(\frac{(width_1/width_2) * \sin(angle)}{1 + (width_1/width_2) * \cos(angle)}) \tag{A.3}$$

$$dist = \frac{width_1}{\sin(\alpha_1)} \tag{A.4}$$

A.2 Mirror points forward or backward — criteria and calculation

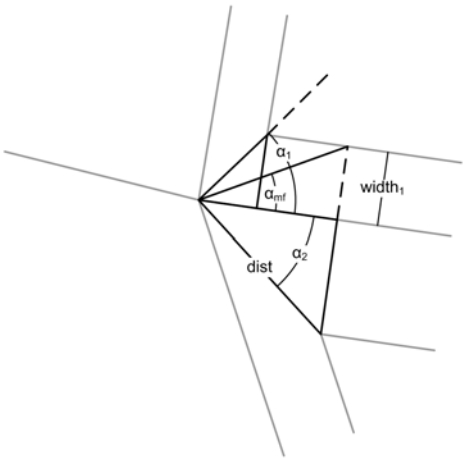


Figure A.2: Criteria for mirroring a point forward

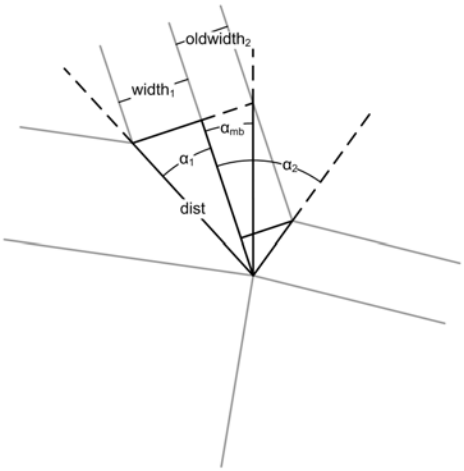


Figure A.3: Criteria for mirroring a point backward

If a point is mirrored back or forward depends on the angles to the cutpoint(α_1 and α_2) and the two widths (width$_1$, width$_2$).
The angle (α_{mf}) and distance (dist$_mf$) of a mirrored forward point are:

$$\alpha_{mf} = \arctan(\frac{width_1}{\cos(\alpha_2) * dist}) \tag{A.5}$$

$$dist_{mf} = \frac{width_1}{\sin(\alpha_{mf})} \tag{A.6}$$

A point is mirrored forward if:

$$\alpha_1 > \alpha_{mf} \tag{A.7}$$

The angle (α_{mb}) and distance (dist$_m b$) of a mirrored backward point are:

$$\alpha_{mb} = \arctan(\frac{oldwidth_2}{\cos(\alpha_1) * dist}) \tag{A.8}$$

$$dist_{mb} = \frac{oldwidth_2}{\sin(\alpha_{mb})} \tag{A.9}$$

A point is mirrored back if:

$$\alpha_2 > \alpha_{mb} \tag{A.10}$$

A.3 Finding the staring and ending angle of an arc to add for a roundabout

Also these angles depend on the widths of the involved carriageways (width$_1$, width$_2$) as well as the outer radius of the roundabout (radius). The arc then goes from:

$$angle_{actual} + \arcsin(\frac{width_1}{radius}) \tag{A.11}$$

to:

$$angle_{next} - \arcsin(\frac{width_2}{radius}) \tag{A.12}$$

A.4 Finding the middle points for rounding arcs

The scenario for finding the middle point depends on whether the cut of two straight lines (figure A.4) or the cut of a straight line and a circle (figure A.5)has to be rounded.
The simpler case is the one with the two straight lines. There the midpoint is in direction

$$\alpha = angle_{actual} + \frac{angle}{2} \tag{A.13}$$

and distance

$$dist = \frac{radius_{rounding}}{arcsin(angle/2)} \tag{A.14}$$

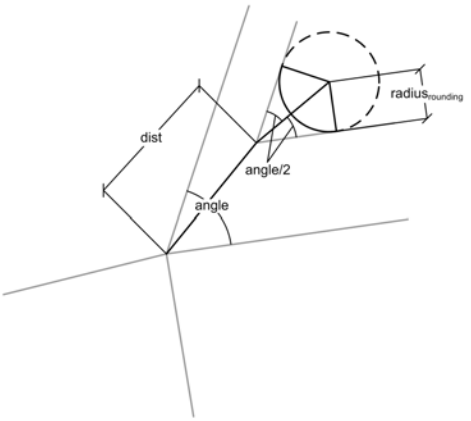


Figure A.4: Finding the middle point if the cut of two straight lines has to be rounded

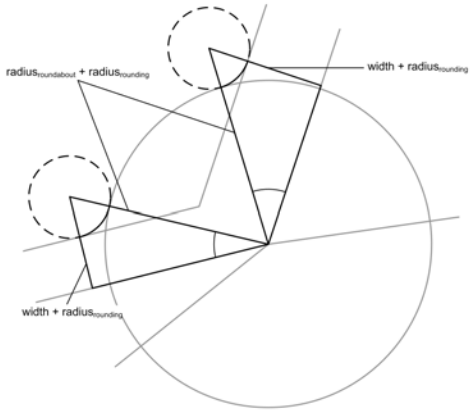


Figure A.5: Finding the middle point if the cut of a straight line and a circle has to be rounded

away from the cutting point.
A bit more complicated is the situation of a cut between a circle and a straight line. There the midpoint is in direction

$$\alpha = angle_{actual} \pm \arcsin(\frac{(width + radius_{rounding})}{(radius_{circle} + radius_{rounding})}) \tag{A.15}$$

and distance

$$dist = radius_{circle} + radius_{rounding} \tag{A.16}$$

away from the midpoint of the circle.

A.5 Finding the staring and ending angle of an arc to add for rounding

For a straight line this angle is at a right angle to the straight line in question.
For a circle the angle is the one from the calculated midpoint to the midpoint of the circle.

A.6 Determine the shortening or elongation of the outer borders of carriageways or sidewalks

For this, first the values are calculated using Pythagoras:

$$\delta = sqrt(dist^2 - width^2) \tag{A.17}$$

Then the sign is determined using the direction of the connection from the inner to the outer point on the carriageway or streettile and the direction of the edge (from one vertice to the other).

Bibliography

[1] P. Mueller P. Wonka S.Haegler A.Ulmer and L. Van Gool. *Prozedurales Modelieren einer Stadt*. In *Proceedings of ACM SIGGRAPH 2006 / ACM Transactions on Graphics (TOG)*, volume 25, pages 614–623. ACM Press, 2006.

[2] Pascal Müller. *Prozedurales Modelieren einer Stadt*. Semester Thesis, 1999. ETH Zurich.

[3] Pascal Müller. *Design und Implementation einer Preprocessing Pipeline zur Visualisierung prozedural erzeugter Stadtmodelle*. Master's thesis, ETH Zurich, 2001.

[4] P. Mueller and Y. Parish. *Procedural Modeling of Cities*. In *Proceedings of ACM SIGGRAPH 2001*, pages 301–308. ACM Press, 2001.

[5] Jared Schirm. *Procedural Streets Construction*. Semester Thesis, 2005. ETH Zurich.

[6] *Cyberwalk Project*. http://www.cyberwalk-project.org/.

[7] *ESRI Shapefile Technical Description*, 1998.

[8] *Geography Markup Language (GML)*, 2004.

[9] *Google Earth KML 2.0*. http://earth.google.com/kml/.

[10] *GPS Exchange Format*. http://www.topografix.com/gpx.asp.

[11] *Manual on Uniform Traffic Control Devices*. http://mutcd.fhwa.dot.gov.

[12] *MapInfo Data Interchange Format*. http://www.gismngt.de/format/midmif1.htm.

[13] *Virtual Terrain Project*. http://www.vterrain.org/.

[14] *Wikipedia*. http://www.wikipedia.org.