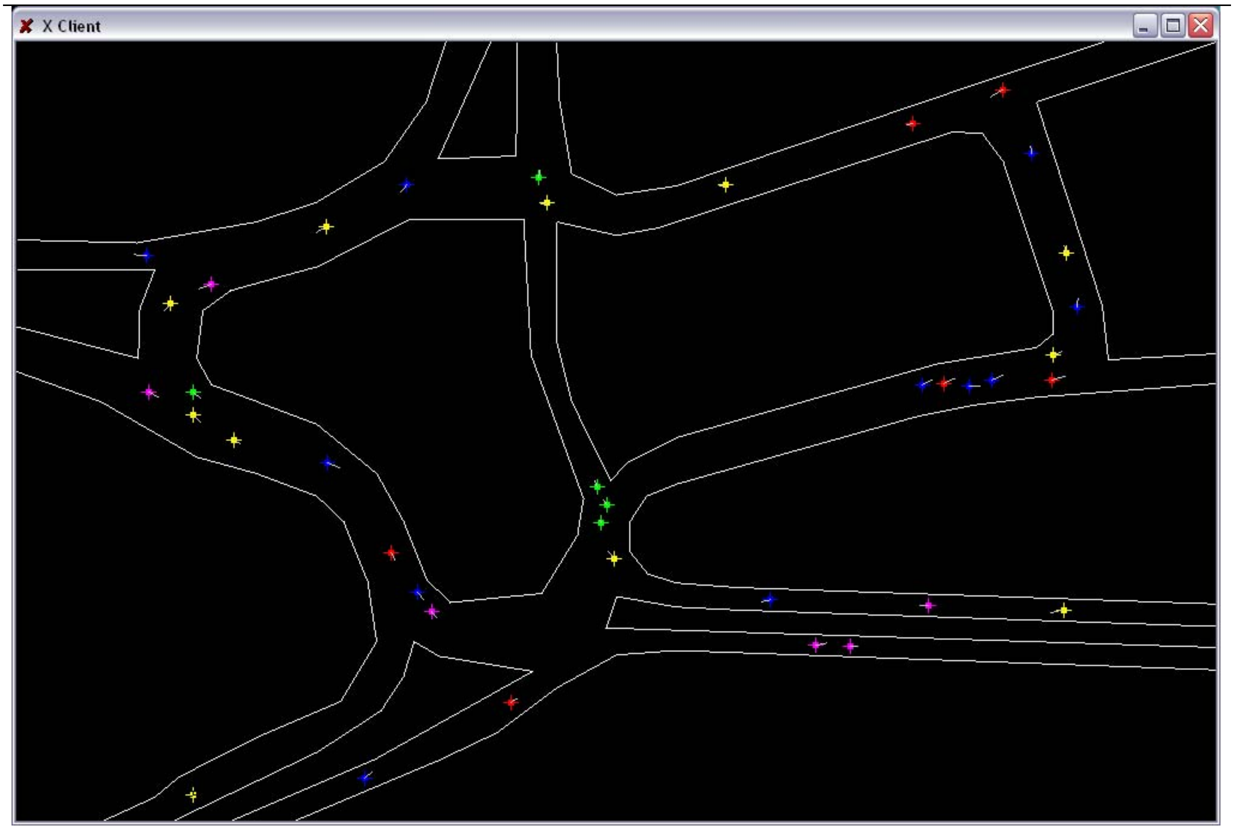


Shape Morphing of Intersections Using Curb Side Oriented Driver Simulation

Michael Balmer, IVT, ETH Zurich
Arnd Vogel, VSP, TU-Berlin
Kai Nagel, VSP, TU-Berlin

Conference paper STRC 2005

STRC | 5th Swiss Transport Research Conference
Monte Verità / Ascona, March 9-11, 2005

Shape Morphing of Intersections Using Curb Side Oriented Driver Simulation

Michael Balmer
IVT
Zurich

Phone: +41-1-633 27 80
Fax: +41-1-633 10 57
email:
balmer@ivt.baug.ethz.ch

Arnd Vogel
VSP
Berlin

Phone: +49-30-314-29522
Fax: +49-30-314-26269
email:
vogel@vsp.tu-berlin.de

Kai Nagel
VSP
Berlin

Phone : +49-30-314-23308
Fax: +49-30-314-26269
email:
nagel@vsp.tu-berlin.de

March 2005

Abstract

In a traffic network, capacities of parts of the network restrict the amount of transport units which can be handled by this network. For example a two lane highway can handle a larger amount of vehicles than an access road through the forest. The capacity of a given traffic network element is not fixed but influenced by parameters such as number of lanes, maximum speed, weather, view horizon and so on. These parameters also define the maximum capacity of intersections and roundabouts. Special shapes of intersections, particularly in urban regions, may further increase or decrease their capacity. This paper investigates how the capacity of such special intersections can be estimated with only the curbsides of the intersection as an input. It is also of interest if changes to the shape decrease the amount of space "wasted" for the traffic intersection while the capacity remains unchanged.

In this case study one special intersection is examined: "Central" in downtown Zurich, Switzerland. The particularity of this intersection is that it partially behaves like a roundabout but also contains two uncontrolled intersections. Due to its central position in the city, the intersection is very busy with both individual cars and public transport vehicles.

In the first part of this paper, a simulation model which is able to produce realistically behaving vehicles only by using information about the curb side locations of the intersection is described The simulation shows how vehicles produce and dissolve congestion and tailbacks, demonstrating that the topology of the road configuration is a major contributor to congestion.

In the second part of the paper, the simulation changes the topology of the scenario based on the observed behavior of the vehicles. Using a feedback loop allows one to optimize the capacity of the intersection while its spatial extents are minimized.

Keywords

simulation – shape of intersections – feedback algorithms – Swiss Transport Research Conference – STRC 2005 – Monte Verità

1. Introduction

Typically, a roundabout is defined by its shape, incoming and outgoing streets, their number of lanes, the number of lanes inside the roundabout, and so on. All these parameters influence the traffic behaviour of car drivers within the roundabout. In order to construct an agent based car driver simulation of a roundabout, each modelled agent must respect these constraints.

Typically the area of a roundabout is limited by existing buildings, necessary pedestrian ways, governmental rules and other additional constraints. To find a shape for a roundabout using as less as possible space while providing the required traffic capacity is therefore an optimization problem (Campbell *et al.*, 1999, Dijkstra and Timmermans, 2002).

In this paper we demonstrate an agent based approach (see Ferber, 1999) solving this optimization problem. For this, we need to calculate the capacity of a given roundabout by only using the shape (the curbs) of this roundabout as input data. Other characteristics like lanes, traffic signs, etc. are not considered. Section 2 describes how a roundabout is defined. It also gives a brief description about the roundabout chosen for this study.

Section 3 describes how an agent based car driver simulation is being developed and applied in order to calculate traffic indicators like congestion or capacity. The simulation model developed in this section produces realistic agent behaviour although only the curbs of the intersection of interest are given as input data.

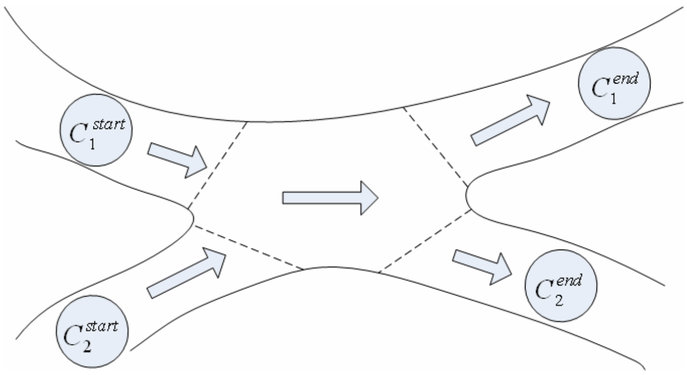
With the knowledge of the calculated indicators of Section 3, a method is shown in Section 4 how those can be used to change the shape of the roundabout such that the calculated indicators will be optimized.

Section 5 describes some first results of the case study of the “Central” roundabout of Zurich. The paper finishes with a summary and a brief outlook on further works.

2. Modelling the Roundabout

We define a roundabout by directed street segments. A street segment consists of one or more driving lanes with the same driving direction and does not have any kind of junctions. An incoming street segment of the roundabout additionally holds a defined area inside the street segment shape where car driving agents are allowed to enter the roundabout. Each chosen roundabout holds n_{in} incoming and n_{out} outgoing street segments. For an outgoing street segment, a similar area is defined where agents are allowed to leave the system. For simplicity we define such an area by a circle $C(\vec{c}, r)$ with centre $\vec{c}$ and radius r. Entering circles are denoted by C_i^{start} and leaving circles by C_j^{end}, while $i = 1 \ldots n_{in}$ and $j = 1 \ldots n_{out}$. Figure 1 shows an example traffic system.

Figure 1 An example traffic system with five street segments



The model of the roundabout provides the following additional information:

- Description of the curbs. The curbs are represented as geometric primitives like geometric nodes and links.
- Description of the driving routes through the roundabout. The driving routes are necessary for simulating the driver agents. Each car driving agent holds information about the streets where he is going to enter and to leave the intersection. Each possible combination of entering and leaving street segments has to be associated with a route through the intersection.

Since we do not want to include any information but the curbs into the model, also the driving routes have to be defined by a set of curb segments.

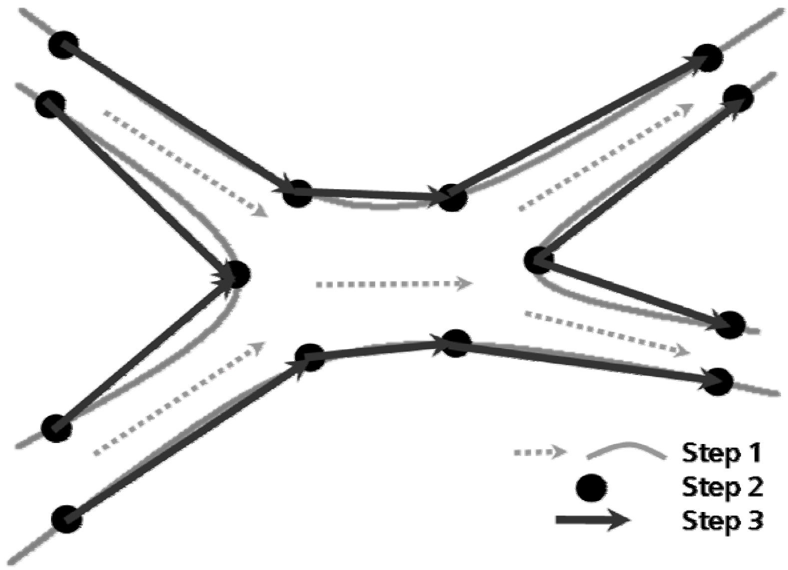
2.1 Modelling the Curbs

Modelling of the curbs is a straight forward process. Instead of using the real shape of the roundabout, we simplify the curbs as line segments. This generic approach has the advantage of allowing the description of any kind of roundabout including any intersections and also larger networks without the need to handle different kinds of geometric primitive types. Furthermore, the degree of detail can be easily increased by creating more and therefore shorter line segments for the curbs. The following shows the curb modelling process:

1. Given a shape and the driving direction of a roundabout and an arbitrary origin of a Euclidian coordinate system
2. Define a set N of nodes $\vec{n} = \begin{pmatrix} n_x \\ n_y \end{pmatrix}$ along all curbs of the given shape
3. Define a set L^{vis} of directed links $\vec{l}^{\,vis} = \vec{l}^{\,vis}\left(\vec{n}^{\,start}, \vec{n}^{\,end}\right) = \begin{pmatrix} n_x^{end} - n_x^{start} \\ n_y^{end} - n_y^{start} \end{pmatrix}$ along the curbs, each connected by a start and an end node from the set N. The direction of the link must follow the driving direction in this street segment.

Note that a link vector has a defined location, direction and length in the coordinate system given by its start and end node. Figure 2 shows the three steps of modelling the curbs in a graphical way.

Figure 2 Modelling the curbs



2.2 Modelling the Driving Routes ("Tunnels")

In order to let the agents travel through the roundabout, a route through it needs to be assigned to each agent. Since we only want to use the curbs as given information about the roundabout, those driving routes (now also called "tunnels") must be described as a set of curb segments (see Helbing *et al.*, 1997 for other ways of describing routes).

A tunnel $T_{ij} = T_{ij}\left(L_{ij}, C_i^{start}, C_j^{end}\right)$ describes the area in which an agent is allowed to drive. The area is described by a set of links L_{ij}. Since a tunnel has exactly one entrance area C_i^{start} and one leaving area C_j^{end}, there are no "wholes" in the tunnel, meaning that a tunnel is defined by exactly two curb sides, a left one and a right one. To provide this, we need to add additional "invisible" links to differentiate street segments which are part of a tunnel from those which are not. The following describes the modelling process for all possible tunnels of the given roundabout:

1. Given a set of links L^{vis} as described in Section 2.1

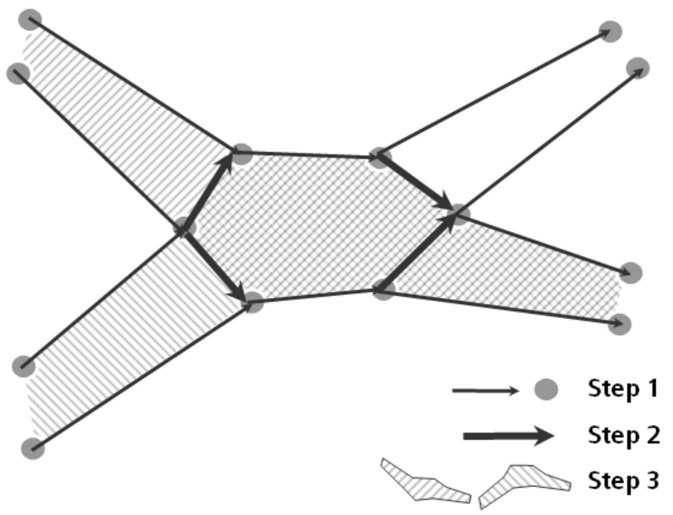
2. Add a set of "invisible" links L^{invis} (an invisible link is denoted as $\vec{l}^{\,invis} = \vec{l}^{\,invis}\left(\vec{n}^{\,start}, \vec{n}^{\,end}\right)$) for each intersection area, such that each street segment is separated from the intersection area. The total set of links is therefore $L = L^{invis} \cup L^{vis}$.

3. For each pair of incoming and outgoing street segments of the roundabout, define one tunnel described by a set of visible and invisible links $L_{ij} \subseteq L$ such that each tunnel holds exactly two curb sides, a left and a right one, starting from the incoming street segment and ending at the outgoing one.

A link element of the set of visible or invisible links of tunnel T_{ij} is denoted as $\vec{l}_{ij} \in L_{ij}$. With this model, exactly $n_{tunnel} = \sum n_{in} \cdot \sum n_{out}$ tunnels are modelled. One can say that there are more that just those, since in a roundabout, it is also allowed to drive in circle several times before leaving it at the desired outgoing street segment. This fact is left out in the model since it is not the typical behaviour of a car driver and does not influence the outcome of the analysis.

Figure 3 shows an example of modelling the tunnels. Note that only two of the four possible tunnels are drawn.

Figure 3 Modelling the tunnels (two of the four tunnels are drawn)



2.3 "Central" Roundabout of Downtown Zurich

This paper uses the "Central" roundabout of downtown Zurich as a case study. There are several reasons for choosing this roundabout:

- It is one of the bottlenecks of the Zurich street network.
- Even if its shape correlates to a roundabout; it still holds two intersections.
- The number of lanes varies inside the roundabout. Therefore, the amount of space used by a street segment also varies within the roundabout.
- A major reconstruction was done during summer 2004.

This study was done before the reconstruction of the Central roundabout. So, part of the analysis of Section 5 can be done by comparing the results with the situation after the reconstruction.

Figure 4 Top view of the "Central" roundabout of downtown Zurich

Source: http://www.sanday.ch (accessed February 2005)

Figure 4 shows the special shape of the “Central”. Note that the middle road (where left turns are not allowed) is a “short-cut” for leaving the “Central” towards the bridge over the river. Therefore the “Central” is not a “real” roundabout anymore. It holds five incoming and five outgoing street segments. This leads us to 25 different tunnels as described in Section 2.2.

3. Car Driver Simulation

Given a model of a roundabout as described in Section 2 we build an agent based car driver simulation, which respects the following constraints:

1. Every agent is assigned to a specific tunnel and is not allowed to change into another tunnel during the simulation.

2. Given a tunnel, an agent must not drive over the tunnels curb sides.

3. An agent must start at the incoming street segment and end at the outgoing one of the given tunnel.

4. An agent is not allowed to drive “unrealistically” through the given tunnel. He must not drive backwards and he must not drive extremely apart from the driving direction given by the tunnel (i.e. right-angled or driving in opposite direction).

5. An agent must respect the physical rules of acceleration.

6. An agent tries to drive through the system with a “desired driving speed”.

7. An agent can not steer more than a given “maximum steering” constant (otherwise cars could change directions right in place).

8. An agent must respect other agents in the system. He has to decelerate or overtake if a slower agent drives in front of him.

The main idea of this agent based car driver simulation follows the principle of particle simulations with discrete time steps Δt used in various topics in computational science. Assume each tunnel defines a current which flows in direction of the given directed curbs and assume that a car driving agent is one particle of fluid in that current, then the constraints 1, 2, 3, 5 and 8 of the above list are fulfilled (for laminar flow).

By adding more, partially overlapping currents representing the other tunnels, and by fulfilling constraint number 4, 6 and 7, those particles become agents (Ferber, 1999). This idea will be formalized in the following subsections.

3.1 Defining a Car Driving Agent

As mentioned above, each car driving agent a_k is assigned to a tunnel T^{ij} of the given roundabout (denoted as $a_k^{(T_{ij})}$). T^{ij} defines the path of a_k through the system. For each point in time t each agent holds a certain amount of information about his current state:

- At time t, each agent's position is defined as $\vec{a}_k(t) = (x_k(t), y_k(t))$.
- At time t, each agent holds his current driving speed denoted as $s_k(t)$.
- At time t, each agent holds his current driving direction denoted as $\varphi_k(t)$.

Each agent also holds some predefined constant parameters:

- Each agent has a desired driving speed s_k^{des}.
- Each agent has a defined maximum acceleration $acc_k^{\max}$.
- Each agent has a defined shape. For simplicity the shape is defined as a circle with radius r_k.
- Each agent has a maximum steering limit, denoted as $\rho_k^{\max}$.
- Each agent holds a mass m_k.
- Finally, each agent holds a maximum allowed angle with respect to a given flow force vector, called $\theta_k^{\max}$. The flow force will be described in details in Section 3.2.

Note that in the following sections, we do not take account of the mass m_k. Therefore we just define $m_k = 1, \forall a_k$.

With the two scalars $s_k(t)$ and $\varphi_k(t)$ the agent's velocity vector $\vec{v}_k(t)$ is defined by a polar coordinate system, denoted as $\vec{v}_k(t)_{s,\varphi}$. The conversion into the global Euclidian coordinate system is therefore $\vec{v}_k(t)_{x,y} = (s_k(t) \cdot \cos(\varphi_k(t)), s_k(t) \cdot \sin(\varphi_k(t)))$. It is important to notice that we will use the polar representation, since it has one main advantage: An agent who stopped ($s_k(t = t_0) = 0$) still has a direction ($\varphi_k(t = t_0) \in [-\pi, \pi]$) of his car. In Euclidian coordinates we would loose that information.

To follow the idea of a particle simulation each agent in the system reacts to an external force field (similar approaches in Gloor *et al.*, 2003). For each point in time t during the simulation and on each agent's position $\vec{a}_k(t)$ a force $\vec{F}_{tot}\left(\vec{a}_k^{(T_{ij})}(t)\right)$ needs to be calculated which influences the agent. This force consists of three components, a "flow force" $\vec{F}_{flow}\left(\vec{a}_k^{(T_{ij})}(t)\right)$, a "curb repulsion force" $\vec{F}_{curb}\left(\vec{a}_k^{(T_{ij})}\right)$ and a "neighbour agent repulsion force" $\vec{F}_{neigbour}(\vec{a}_k(t))$. Therefore, the force which influences an agent at a given position at a given time is

$$\vec{F}_{tot}\left(\vec{a}_k^{(T_{ij})}(t)\right) = \vec{F}_{flow}\left(\vec{a}_k^{(T_{ij})}(t)\right) + \vec{F}_{curb}\left(\vec{a}_k^{(T_{ij})}\right) + \vec{F}_{neigbour}(\vec{a}_k(t)).$$

As indicated in the formula, the flow force of an agent at a specific position is dependent on time and on the given tunnel. The curb repulsion force does not change in time but differs between different tunnels. And at last, the neighbour agent repulsion force is time dependent but not related to the tunnel which the agent is assigned to. The following sections define the three forces in detail.

3.2 Flow Force

The flow force field defines the flow of a tunnel T_{ij}. It pushes an agent $a_k^{(T_{ij})}$ in the right driving direction through his given tunnel. It is also responsible for letting an agent drive with his desired speed.

The flow force $\vec{F}_{flow}\left(\vec{a}_k^{(T_{ij})}(t)\right)$ of an agent $a_k^{(T_{ij})}$ with tunnel T_{ij} at a position $\vec{a}_k$, desired speed s_k^{des} and velocity vector $\vec{v}_k(t)$ is defined by

$$\vec{F}_{flow}\left(\vec{a}_k^{(T_{ij})}(t)\right) = \alpha \cdot \left(s_k^{des} \cdot \vec{v}_{flow}^{(T_{ij})}(\vec{a}_k) - \vec{v}_k(t)\right)$$

The proportion parameter $\alpha > 0$ describes a similar effect as the viscosity of a fluid. For a small α the flow force does not have much influence to the agent and therefore acceleration of an agent is small. For a big α, agents would accelerate faster. The time independent flow velocity vector $\vec{v}_{flow}^{(T_{ij})}(\vec{a}_k)$ of the tunnel T_{ij} at position $\vec{a}_k$ is

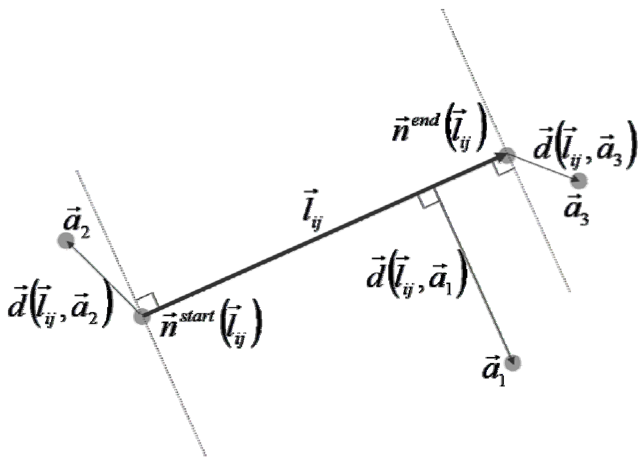
$$\vec{v}_{flow}^{(T_{ij})}(\vec{a}_k) = \frac{\sum_{l_{ij}} \left(\left|\vec{d}\left(\vec{l}_{ij}, \vec{a}_k\right)\right|^{-\beta} \cdot \frac{\vec{l}_{ij}}{\left|\vec{l}_{ij}\right|} \right)}{\left| \sum_{l_{ij}} \left(\left|\vec{d}\left(\vec{l}_{ij}, \vec{a}_k\right)\right|^{-\beta} \cdot \frac{\vec{l}_{ij}}{\left|\vec{l}_{ij}\right|} \right) \right|}.$$

Note, that the tunnel flow speed is $\left|\vec{v}_{flow}^{(T_{ij})}(\vec{a}_k)\right| = 1$. The "real" tunnel flow speed is different for each agent because their desired speeds vary. The parameter $\beta \geq 0$ defines the influence of links far away. If it's big, such a link gets less important. If $\beta = 0$, then each link of the tunnel is weighted equally for calculating the flow velocity. The distance vector $\vec{d}(\vec{l}_{ij}, \vec{a}_k)$ between a link of tunnel T_{ij} and the position of agent $\vec{a}_k$ is defined as:

$$\vec{d}(\vec{l}_{ij}, \vec{a}_k) = \begin{cases} \vec{a}_k - \vec{n}^{start}(\vec{l}_{ij}) & , if \quad \vec{l}_{ij} \cdot (\vec{a}_k - \vec{n}^{start}(\vec{l}_{ij})) < 0 \\ \vec{a}_k - \vec{n}^{end}(\vec{l}_{ij}) & , if \quad \vec{l}_{ij} \cdot (\vec{a}_k - \vec{n}^{start}(\vec{l}_{ij})) > |\vec{l}_{ij}| \\ \vec{a}_k - \vec{n}^{start}(\vec{l}_{ij}) - \vec{l}_{ij} \cdot (\vec{a}_k - \vec{n}^{start}(\vec{l}_{ij})) \cdot \frac{\vec{l}_{ij}}{|\vec{l}_{ij}|} & , else \end{cases}$$

Figure 5 shows the graphical interpretation of the distance vector. As long as an agent's position is inside the area given by the two dotted lines, the distance vector is right-angled to the link, otherwise the start node (end node, resp.) of the link defines the distance vector to the agent's position.

Figure 5 Distance vector between a link and a position of an agent



Assume that the flow force is the only one which influences an agent; it is possible for him to drive over the curbs of his tunnel. He also will not respect other driving agents, meaning he just drives through them. The following two forces prevent these.

3.3 Curb Repulsion Force

The curb repulsion force field pushes agents away from the curb sides of a tunnel in order to prevent the agents for driving across them (see Stucki, 2003 for a similar approach). The closer an agent gets to a curb, the stronger he will get pushed away from it.

The curb repulsion force $\vec{F}_{curb}\left(\vec{a}_k^{(T_{ij})}\right)$ of an agent a_k with tunnel T_{ij} at a position $\vec{a}_k$, desired speed s_k^{des} and car radius r_k is defined by

$$\vec{F}_{curb}\left(\vec{a}_k^{(T_{ij})}\right) = \sum_{\vec{l}_{ij}} \vec{F}_{curb}\left(\vec{a}_k^{(T_{ij})}, \vec{l}_{ij}\right),$$

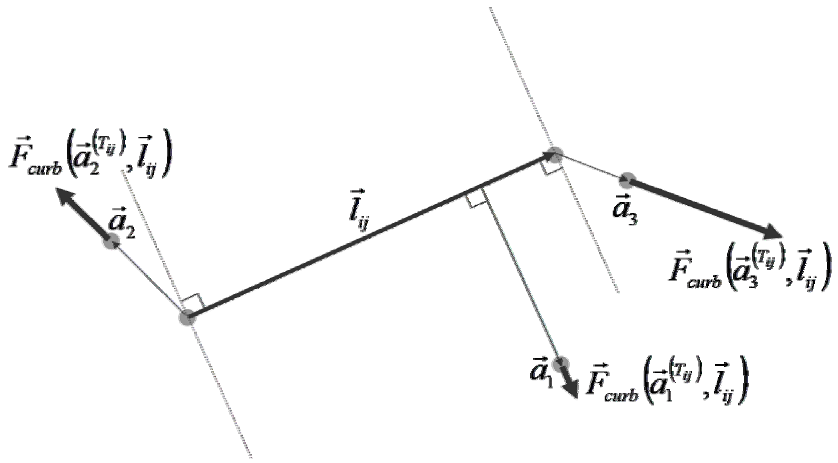
while the repulsion force of one link $\vec{l}_{ij} \in L_{ij}$ of tunnel T_{ij} is calculated as

$$\vec{F}_{curb}\left(\vec{a}_k^{(T_{ij})}, \vec{l}_{ij}\right) = s_k^{des} \cdot \left(\left|\vec{d}\left(\vec{l}_{ij}, \vec{a}_k\right)\right| - r_k\right)^{-\gamma} \cdot \frac{\vec{d}\left(\vec{l}_{ij}, \vec{a}_k\right)}{\left|\vec{d}\left(\vec{l}_{ij}, \vec{a}_k\right)\right|}.$$

The distance vector $\vec{d}\left(\vec{l}_{ij}, \vec{a}_k\right)$ is already defined in Section 3.2. Since an agent's car has a certain extent r_k, this has to be subtracted from the distance. With parameter $\gamma > 0$ the repulsion force of a link near to an agent is larger than the one of a link far away.

Figure 6 shows a graphical interpretation of the curb repulsion force $\vec{F}_{curb}\left(\vec{a}_k^{(T_{ij})}, \vec{l}_{ij}\right)$. As we can see, agents who are nearer to the given link are receiving a stronger repulsion force than an agent far away. Note that an agent who touches the link tangentially receives an infinite repulsion force.

Figure 6 Curb repulsion force on an agent by a given link



Adding this force to the flow force described in Section 3.2 will guarantee that an agent is driving through his tunnel without crossing the tunnel's curb side.

3.4 Neighbour Agent Repulsion Force

By adding the neighbour agent repulsion force field to the total force of an agent in the simulation system, the agents respect their counterparts (Helbing *et al.*, 2000).

The neighbour agent repulsion force $\vec{F}_{neigbour}(\vec{a}_k(t))$ of an agent a_k at position $\vec{a}_k$ at time t with speed s_k^{des} and car radius r_k is defined by

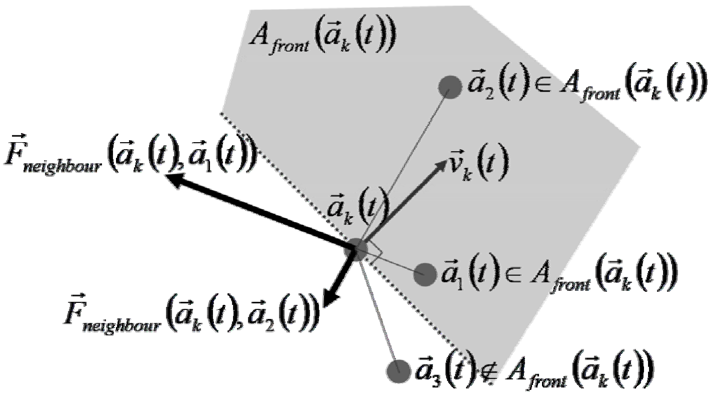
$$\vec{F}_{neigbour}(\vec{a}_k(t)) = \sum_{\substack{\vec{a}_m(t)\in A_{front}(\vec{a}_k(t)) \\ a_m \neq a_k}} \vec{F}_{neigbour}(\vec{a}_k(t), \vec{a}_m(t)),$$

where $A_{front}(\vec{a}_k(t))$ is the area in front of agent a_k. The position of an agent $\vec{a}_m(t)$ is part of area $A_{front}(\vec{a}_k(t))$, only if $\vec{v}_k(t)\cdot(\vec{a}_m(t)-\vec{a}_k(t)) \geq 0$. The repulsion force given by an agent a_m on an agent a_k is

$$\vec{F}_{neigbour}(\vec{a}_k(t), \vec{a}_m(t)) = s_k^{des} \cdot \left(\left|\vec{a}_k(t)-\vec{a}_m(t)\right| - r_k - r_m\right)^{-\delta} \cdot \frac{\vec{a}_k(t)-\vec{a}_m(t)}{\left|\vec{a}_k(t)-\vec{a}_m(t)\right|}.$$

With parameter $\delta \geq 0$ the neighbour agent repulsion force of an agent a_k near to agent a_m is higher then the one of another agent far away. Figure 7 shows a graphical interpretation of force $\vec{F}_{neigbour}(\vec{a}_k(t), \vec{a}_m(t))$. It also points out the area in which other agents have influence to agent a_k.

Figure 7 Neighbour agent repulsion force on an agent



3.5 Acceleration and Steering

As already mentioned at the beginning of Section 3 an agent at position $\vec{a}_k(t)$ at time t reacts in two ways to a given force $\vec{F}_{tot}\left(\vec{a}_k^{(T_{ij})}(t)\right)$. In each time step Δt, he accelerates (decelerates, resp.) and he changes the driving direction by steering. This section describes the update rules for the agent's speed and direction by the calculated total force. The update is done in the following two steps.

Step I:

Let us first define an angle ϕ as the one between the velocity vector $\vec{v}_k(t)_{s,\varphi}$ and the total force $\vec{F}_{tot}\left(\vec{a}_k^{(T_{ij})}(t)\right)$ with

$$\phi = \angle\left(\vec{v}_k(t)_{s,\varphi}, \vec{F}_{tot}\left(\vec{a}_k^{(T_{ij})}(t)\right)\right), \quad \phi = [-\pi, \pi].$$

According to the physical rules of motion, an agent at position $\vec{a}_k(t)$ with speed $s_k(t)$, direction $\varphi_k(t)$ and force $\vec{F}_{tot}\left(\vec{a}_k^{(T_{ij})}(t)\right)$ reacts like the following

$$\begin{aligned} s_k^I(t) &= s_k(t) + acc_k(t)\cdot \Delta t \\ \varphi_k^I(t) &= \varphi_k(t) + \rho_k(t)\cdot s_k(t)\cdot \Delta t \end{aligned}.$$

The acceleration $acc_k(t)$ of the agent induced by the total force can be calculated with formula

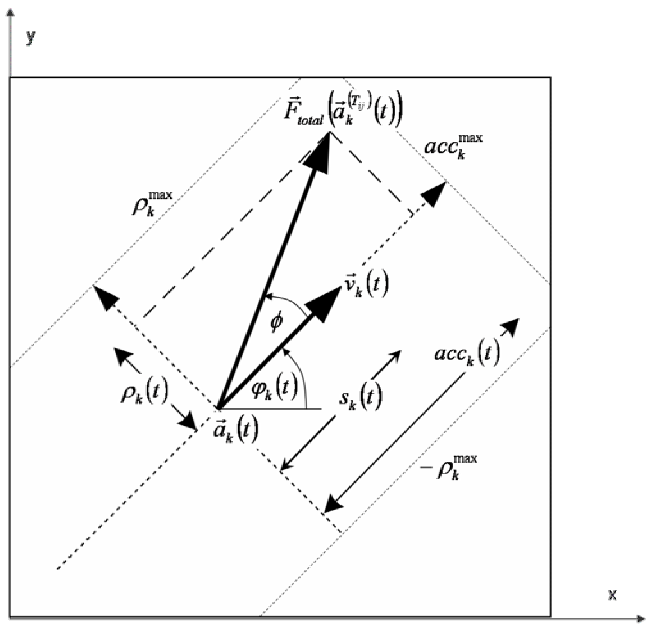
$$acc_k(t) = \begin{cases} acc_k^{\max} & , if \quad \left|\vec{F}_{tot}\left(\vec{a}_k^{(T_{ij})}(t)\right)\right|\cdot\cos(\phi) > acc_k^{\max} \\ \left|\vec{F}_{tot}\left(\vec{a}_k^{(T_{ij})}(t)\right)\right|\cdot\cos(\phi) & , else \end{cases}$$

and the steering $\rho_k(t)$ induced by the total force is

$$\rho_k(t) = \begin{cases} \rho_k^{\max} & , if \quad \left|\vec{F}_{tot}\left(\vec{a}_k^{(T_{ij})}(t)\right)\right|\cdot\sin(\phi) > \rho_k^{\max} \\ -\rho_k^{\max} & , if \quad \left|\vec{F}_{tot}\left(\vec{a}_k^{(T_{ij})}(t)\right)\right|\cdot\sin(\phi) < -\rho_k^{\max} \\ \left|\vec{F}_{tot}\left(\vec{a}_k^{(T_{ij})}(t)\right)\right|\cdot\sin(\phi) & , else \end{cases}$$

Figure 8 shows the calculation of these two scalars. Therefore, $acc_k(t)$ and $\rho_k(t)$ are the values of the abscissa, ordinate resp. of the total force in the local coordinate system given by the velocity vector. To ensure that the calculated values are inside the defined range given by the maximum acceleration and maximum steering, they have to be reduced to those limits in case that they are out of range.

Figure 8 Calculation of acceleration and steering of an agent



Therefore, the velocity vector after step I is $\vec{v}_k^I(t)_{s,\varphi} = \vec{v}_k^I\left(s_k^I(t), \varphi_k^I(t)\right)_{s,\varphi}$.

Step II:

Let us first define an angle θ as the one between the velocity vector $\vec{v}_k^I(t)_{s,\varphi}$ and the flow force $\vec{F}_{flow}\left(\vec{a}_k^{(T_{ij})}(t)\right)_{s,\varphi} = \vec{F}_{flow}\left(r_{flow}\left(\vec{a}_k^{(T_{ij})}(t)\right), \varphi_{flow}\left(\vec{a}_k^{(T_{ij})}(t)\right)\right)_{s,\varphi}$ with

$$\theta = \angle\left(\vec{v}_k^I(t)_{s,\varphi}, \vec{F}_{flow}\left(\vec{a}_k^{(T_{ij})}(t)\right)\right), \quad \theta = [-\pi, \pi].$$

Since an agent is still allowed to drive backwards (negative speed) and to drive extremely apart from the given flow direction of his tunnel, we need to correct the velocity vector $\vec{v}_k^I(t)_{s,\varphi}$ such that it respects these constraints. The correction is defined as

$$s_k(t+\Delta t) = s_k^{II}(t) = \begin{cases} 0 & , if \quad s_k^I(t) < 0 \\ s_k^I(t) & , else \end{cases}$$

$$\varphi_k(t+\Delta t) = \varphi_k^{II}(t) = \begin{cases} \varphi_{flow}\left(\vec{a}_k^{(T_{ij})}(t)\right) + \theta_k^{\max} & , if \quad \theta > \theta_k^{\max} \\ \varphi_{flow}\left(\vec{a}_k^{(T_{ij})}(t)\right) - \theta_k^{\max} & , if \quad \theta < \theta_k^{\max} \\ \varphi_k^I(t) & , else \end{cases}.$$

The speed is therefore reset to zero if it is negative and the direction of the velocity vector is turned towards the flow force vector if the angle between those two vectors is too large. The final updated velocity vector is therefore

$$\vec{v}_k(t+\Delta t) = \vec{v}_k^{II}(t) = \vec{v}_k^{II}\left(s_k^{II}(t), \varphi_k^{II}(t)\right)_{s,\varphi}$$

with speed

$$s_k(t+\Delta t) = s_k^{II}(t)$$

and direction

$$\varphi_k(t+\Delta t) = \varphi_k^{II}(t).$$

We can now calculate the position of the agent at time $t+\Delta t$:

$$\vec{a}_k(t+\Delta t) = \vec{a}_k(t) + \vec{v}_k(t+\Delta t).$$

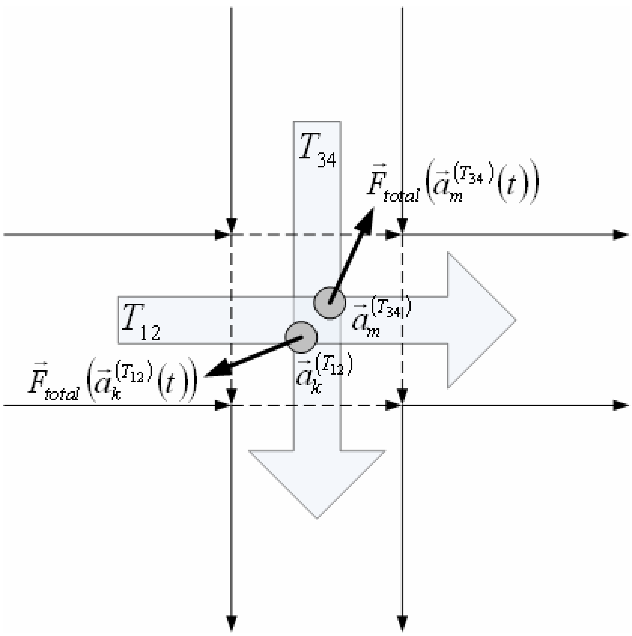
3.6 Congestion

The above described simulation model produces car driving agents who "realistically" drive through a roundabout inside a defined tunnel. They can overtake or follow other agents in the system. Especially slow driving agents can produce tailbacks. But this does not mean that the simulation produces congestion in terms of capacity constraints of a street network. Typically entering lanes and crossroads normally are the cause of occurring congestions. The simulation developed in this paper is able to reproduce that. Figure 9 gives an example of a congested situation in the simulation. In this example we define a crossroad with two tunnels. One agent uses tunnel T_{12}, the other tunnel T_{34}. Because of the short distance between the two agents, the neighbour agent repulsion force has the largest contribution to the total force of each agent. Therefore the total force is directed more or less in the opposite directions of the desired directions of the agents, which means that they have to decelerate and finally stop driving (speed equals zero).

This is a typical “Deadlock” situation, which has to be prevented. Fortunately, it also indicates “difficult” intersection topologies and therefore we can use that information for changing the topology (details in Section 4). Nevertheless, we need to resolve this deadlock situation, which is done in quite a simple way. If an agent’s velocity is zero, he starts counting the number of time steps he doesn’t go on driving. The higher this number is the more probable it becomes that he will just drive on in the next time step. That means that the agents then do not respect the other agents anymore for the next time step. Therefore they will just drive across each other.

Of course this is not realistic anymore, but on the other hand, in uncontrolled intersections (and sometimes also in controlled intersections) similar situations occur. Cars are getting stuck similar to a deadlock situation and then, they try to find a gap and “squeeze” themselves trough it without respecting driving rules anymore. The way the simulation handles this is just a simplification.

Figure 9 Congestion in the simulation



The probability of driving on in a deadlock situation is calculated as followed:

$$p(drive\ on) = \begin{cases} \dfrac{numberOfStepsBeeingStucked}{100} & ,\ if \quad numberOfStepsBeeingStucked < 100 \\ 1 & ,\ else \end{cases}$$

With this simple approach occurring deadlocks can be resolved.

4. Shape Morphing of the Roundabout

The main idea of morphing the shape of a roundabout can be described by the following statement:

Congestion occurs because the street segment (or intersection segment) is too small.

This simple statement gives us the idea how we could morph the shape of a roundabout. Everywhere congestion happens, the roadside corners (nodes) should move away. As we already described in Section 3.6 an agent is in a congested area when he has stopped because of a deadlock situation and then just drives on without respecting the other agents anymore. In other words, a car driver simulation as described in Section 3 can produce "drive-on" events. In the following, we will use the position of the agent which produces such an event (denoted as the agent's position $\vec{a}_k$) in order to modify the shape of the intersection.

Morphing of the shape will be done iteratively. This iteration process is presented in the following section which is based on iterative learning processes like described in Raney and Nagel (2004).

4.1 Iteration Process

The iteration process is done by the following steps:

1. Given an initial shape of the roundabout (modelled as described in Section 2).
2. Run the agent based car driver simulation for a defined time period with simulation step Δt (as described in Section 3) and keep track of the positions of all the "drive-on" events.
3. Change the shape of the roundabout by using the information of the "drive-on" events from the previous simulation.
4. Rerun the simulation with the changed roundabout, and so on.

The advantage of this process is that the morphing algorithm can easily replaced by another. The following section describes one possible algorithm to morph the shape on iteration step 3.

4.2 Morphing Model

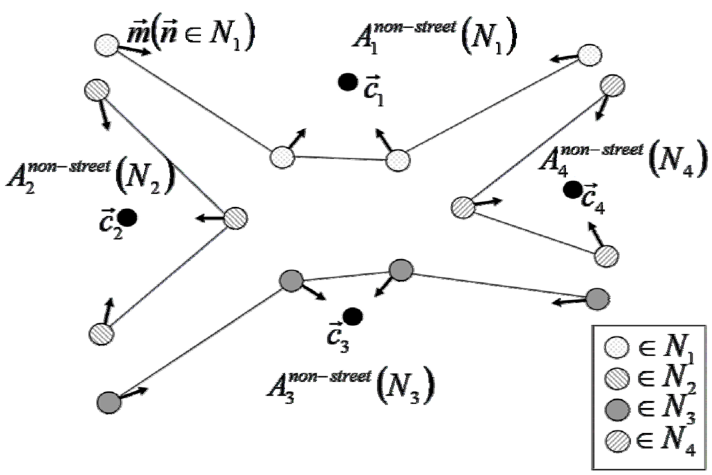
The model should provide the following feature: In congested areas the streets should get wider while in non congested areas the streets should shrink. For that we just need to move the roadside corners (nodes $\vec{n}$), since the roadside links $\vec{l}$ are defined by its start and end node. But to move the nodes we need to know the moving direction. A simple but robust way is to calculate the centroid of each disjoint "non-street" area $A_i^{non-street}(N_i)$ (defined by the set of its border nodes $\vec{n} \in N_i$) and to move the nodes towards these centroids. Figure 10 gives an example of those centres. It is calculated as the arithmetic average of the border nodes of this area:

$$\vec{c}_i = \frac{\sum_{\vec{n} \in N_i} \vec{n}}{\|N_i\|}$$

The normalized moving direction for each node $\vec{n} \in N_i$ of area $A_i^{non-street}(N_i)$ is therefore

$$\vec{m}(\vec{n} \in N_i) = \frac{\vec{c}_i - \vec{n}}{|\vec{c}_i - \vec{n}|}.$$

Figure 10 Example of four independent non-street areas and their centres



Now, we need to calculate an influence parameter $\kappa(\vec{n}, \vec{a}_k)$ of a node $\vec{n} \in N_i$ by a given "drive-on" event $\vec{a}_k$. The calculation is done linear inverse proportional to the distance between the node and the event:

$$\kappa(\vec{n}, \vec{a}_k) = \begin{cases} 0 & , if \quad |\vec{n} - \vec{a}_k| > r_\kappa^{\max} \\ -\frac{1}{r_\kappa^{\max}} \cdot |\vec{n} - \vec{a}_k| + 1 & , else \end{cases} \quad , \quad \kappa(\vec{n}, \vec{a}_k) = [0,1]$$

The parameter $r_\kappa^{\max}$ defines the maximum radius of influence of an event. If a node is located outside of the influence area of an event, $\kappa(\vec{n}, \vec{a}_k)$ is zero. Since there can be more than just one event during the agent based simulation, we sum up the calculated influences for each node:

$$\kappa(\vec{n}) = \sum_{\vec{a}_k} \kappa(\vec{n}, \vec{a}_k)$$

Since we can't control the absolute number of events which occur during a simulation, the range of $\kappa(\vec{n})$ can vary a lot from between iterations. By normalizing it, we calculate an appropriate value which describes the offset by which a node should be moved towards its corresponding centroid $\vec{c}_i$. Additionally we also want to allow that streets can shrink, which means, that the nodes with minimal influence should move away from its centre. Therefore the "moving length" $l(\vec{n})$ of a node $\vec{n} \in N_i$ of area $A_i^{non-street}(N_i)$ with centre $\vec{c}_i$ can be calculated as

$$l(\vec{n}) = \frac{l^{\max} - l^{\min}}{\max_{\vec{a}_k}(\kappa(\vec{n}, \vec{a}_k)) - \min_{\vec{a}_k}(\kappa(\vec{n}, \vec{a}_k))} \cdot \left(\kappa(\vec{n}) - \max_{\vec{a}_k}(\kappa(\vec{n}, \vec{a}_k)) \right) + l^{\max} .$$

The parameter $l^{\max} \geq 0$ defines the maximal length a node is allowed to move towards its centre, while $l^{\min} \leq 0$ defines the maximal length a node is allowed to move away from its centre. If $l^{\max} = 0$ the street segments do not grow, while with $l^{\min} = 0$, they do not shrink.

5. Setup and First Results

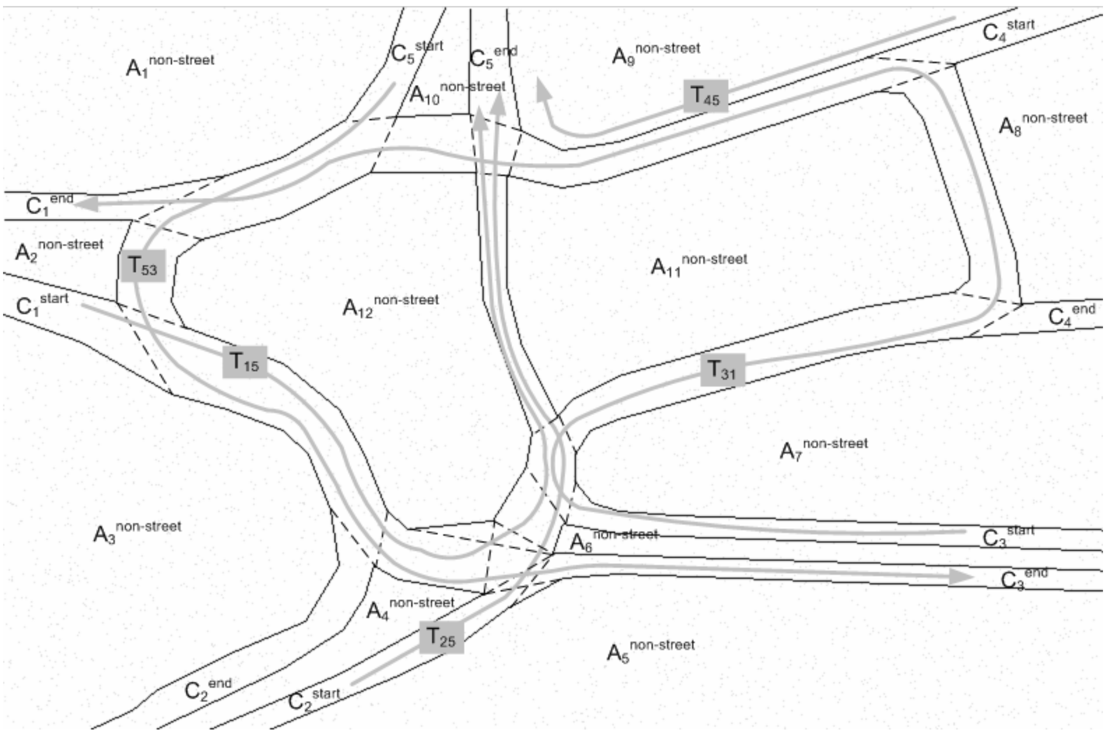
In general it is not that easy to verify the results using the above describe model. On the other hand qualitative comparisons can give us some good indications about the benefits of such a model. The major reconstruction works on the "Central" intersection which ended in autumn 2004 enable us to evaluate the results of the shape morphing against some real world experiences.

5.1 Setup

The Roundabout Model

Figure 11 shows the roundabout model of the "Central". It consists of 108 nodes, 98 visible links and 26 invisible links. There are five incoming and five outgoing street segments and therefore five start and five end circles. This ends up by 25 different tunnels. 12 disjoint non-street areas are set.

Figure 11 Setup of the central roundabout (5 of 25 Tunnels are highlighted)



The Agents

The following parameters are equal for each agent in the system:

- Radius $r_k = r = 1.3\,m$
- Maximum steering limit $\rho_k^{\max} = \rho^{\max} = \pi/4$
- Maximum acceleration $acc_k^{\max} = acc^{\max} = +\infty \frac{m}{s^2}$ (no upper boundary)
- Maximum allowed angle to flow force vector $\theta_k^{\max} = \theta^{\max} = \pi/12$

The desired speed is set different for each agent:

- Desired speed s_k^{des} is uniformly distributed in the range $\left[\frac{20}{3.6}\frac{m}{s}, \frac{50}{3.6}\frac{m}{s}\right]$

Injecting the agents into the system is done by the following rule: If there is no agent inside the entering area C_i^{start}, add a new one at the centre of the entering area with a random tunnel T_{ij}. The radius of C_i^{start} is set to 30 meters.

The Forces

- Proportion parameter of the flow force $\alpha = 5$
- Distance influence parameter of flow force $\beta = 3$
- Distance influence parameter of curb repulsion force $\gamma = 3$
- Distance influence parameter of neighbour agent repulsion force $\delta = 3$

The Time Step and Simulation Time

The time step is set to $\Delta t = 0.05\sec$. Note that it has to be chosen carefully, because an inadequate combination of time step duration and maximum allowed angle to the flow direction can cause inconsistency. If the time step is set too big, it could happen that an agent “jumps” over a curb just in one time step.

The simulation time is set to 240 seconds. Therefore we simulate 4800 time steps starting with an empty roundabout (no agents in the system at time equals zero). Since the parameters which are used by the morphing model are normalized, it is not that important how many steps are simulated. We only need to make sure that there is enough time for a substantial number of agents to leave the roundabout before the simulation ends. Otherwise, we only would simulate a scenario where the roundabout is getting “filled up”.

The Morphing Model

- Maximum radius of influence of an event $r_{\kappa}^{\max} = 10\ m$
- Maximal growing length $l^{\max} = 1.0\ m$
- Maximal shrinking length $l^{\min} = -0.2\ m$

5.2 Results

Car Driver Simulation

As the outcome of the case study shows, the car driver simulation produces the expected results. Agents find their ways trough the system inside their tunnels. They do not drive over a curb side and they respect other agents in the system. They also overtake or follow slower agents depending on the width of the street segment. Congested situations occur and dissolve dependent on the amount of agents in the system. The “drive-on” rule resolves deadlock situations.

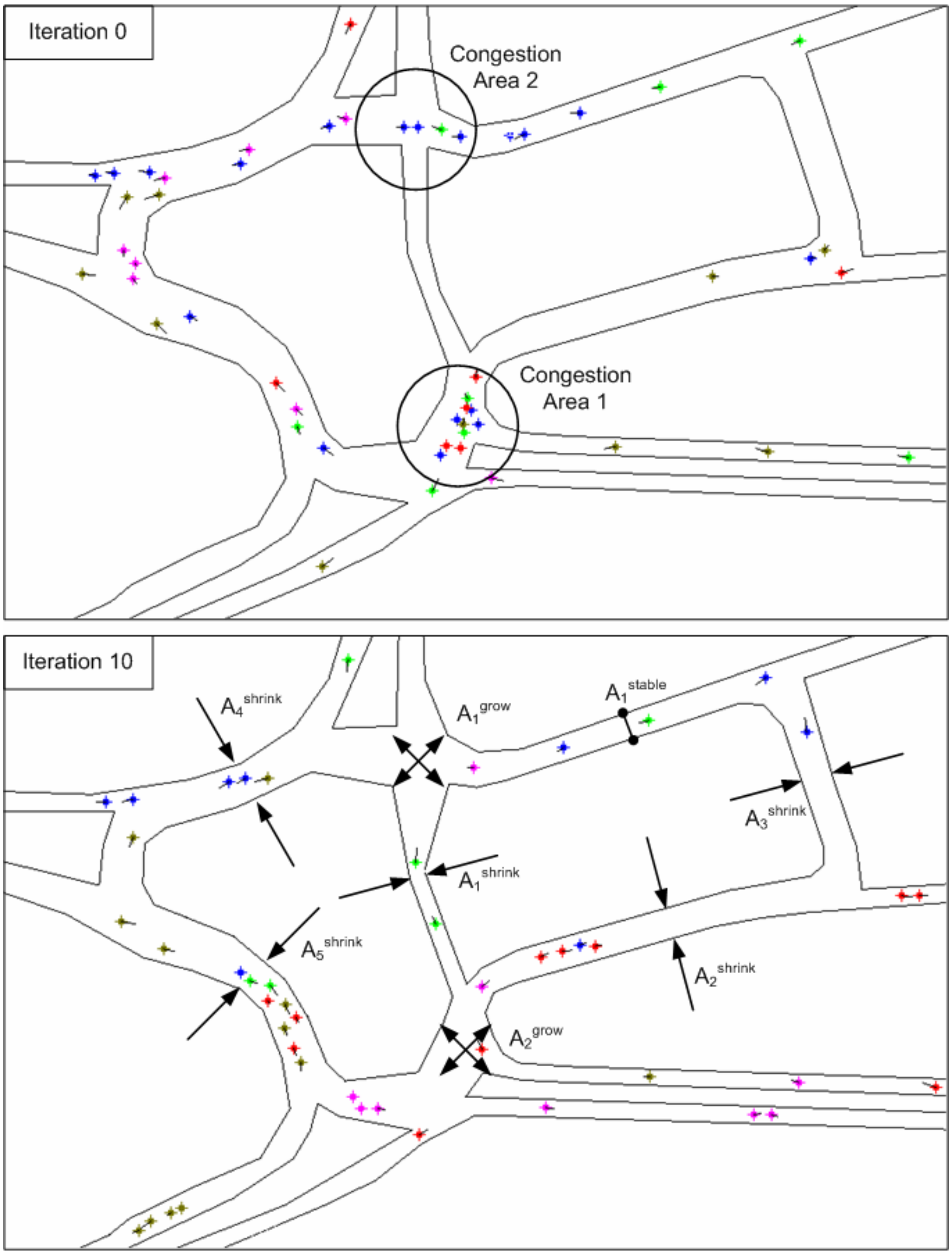
An important fact is that “drive-on” events occur only in congested areas, which is important for the morphing model.

Morphing Model

Also the morphing model shows the expected behaviour. Congested street areas are getting larger while free flow areas are shrinking. But the shape of the roundabout is getting more and more unrealistic. This happens because of the extremely simply morphing rules. From an engineering point of view, one could say that this result is not usable. Nevertheless the created shape of the roundabout gives us very good indication about areas where the system has too much capacity and vice versa.

Figure 12 shows us the result of the morphing process. The car driver simulation produces two main congestion areas (shown in iteration 0 of Figure 12). Those areas are expanding (A_1^{grow} and A_2^{grow}, shown in iteration 10 of Figure 12). On the other hand there are several street segments that shrink. Interestingly, almost the whole left part of the roundabout is shrinking (A_4^{shrink} and A_5^{shrink}), even though there are junctions. The streets on that area were built with two or three lanes (see also the schematic drawing in Figure 4). This could lead us to a conclusion that at least one lane can be closed.

Figure 12 Growing and shrinking areas during the iteration process.



Another interesting shrinking area is A_1^{shrink}. That street segment almost shrinks to the size of an agent (2.6 meter width). It also looks as if this street could be the cause of the two congested areas. Since agents who want to leave the roundabout at the outgoing street segment on the top (C_5^{end} of Figure 11) could also drive along the right loop. So, it is possible to completely close this street segment.

A_1^{stable} shows that there are also some street segments which–at least for this model–have more or less the proper capacity (equal to a two lane street segment). Other stable areas can be found at the incoming and outgoing street segments.

5.3 Qualitative Comparison to Reality

Until spring 2004 the shape of the “Central” roundabout was looking like the Figure 4. During the peak hours traffic policemen were used to control the traffic at the two major congestion areas shown in the upper picture of Figure 12. During the reconstruction process the street segment at area A_1^{shrink} (see Figure 12) was closed for a long time and the drivers were redirected along the right loop. Figure 13 shows the shape of the “Central” after the reconstruction was finished. The shrinking and growing areas of Figure 12 are labelled for better orientation. As we can see in the two areas A_4^{shrink} and A_5^{shrink}, there is a reduction from three to two driving lanes and the centre street segment is reopened again.

It is quite fascinating that the simulation shows the same changes. Even more interesting is that the simulation indicates that the middle street segment is causing too much problems and therefore should be closed in this special case. It would be of interest to measure the behaviour of the reconstructed “Central” if we would close this street again for a longer time period.

Figure 13 Some views on the "Central" roundabout after reconstruction

6. Future Work

The Roundabout Model

The abstraction of the curb sides into nodes and directed links is quite a simple but general solution. It can be reused for any kind of scenarios. On the other hand it could loose some flexibility for the morphing process. Figure 12 shows us an example: The shrinking street segment A_2^{shrink} does not shrink as much as we might expect from the fact that congestion almost never occurs there. Because the links of this street segment are very long, there are only a few nodes which can be moved. This small but important problem can be solved by splitting up long links into several short ones.

The Car Driver Simulation

At the moment each agent calculates his present total force completely "from scratch" for each link of his tunnel and for all other agents in the system at every time step. This wastes a lot of the available computational performance. As the above defined formula already described, there are several issues where it would make sense to pre-compute forces at the beginning of each car driver simulation run (discretization of space; see also Nishinari *et al.*, 2001 and Schadschneider, 2001). With more appropriate data-structure (i.e. Quad-Trees) neighbour agents could also be found much faster.

It is also of interest to add other traffic participants, like trams an pedestrians. Especially for the "Central" the pedestrians influence the capacity of the roundabout a lot, because the direct way from the Zurich main station to the University goes through the "Central". With this, during the morning and the evening peak the place is "flooded" with pedestrians.

The Morphing Model

The above described morphing model is a quite simple. The nodes change their position only along a given line. The model also does not respect geographical constraints like already existing buildings, pedestrian roads, etc. Last but not least the resulting shape of the scenario does not look like streets anymore.

7. Summary

This paper shows two approaches: First, a “realistic” agent based car driver simulation using only the curb side information of the scenario as an input and second, a morphing model for changing the shape of the given roundabout. With a simple iteration process it is shown that good indications can be found for optimizing the shape of the scenario. The iteration process allows us replace the given morphing model by a more enhanced one.

Apart of the above described, using iteration processes for optimization problems has at least one other great advantage: It allows us to separate the problem into pieces such that they are easier to understand, monitor and analyse.

8. References

Campbell, M. I., J. Cagan and K. Kotovsky (1999). A-design: An agent-based approach to conceptual design in a dynamic environment. In *Research in Engineering Design*, **11**(3), pages 172–192. Springer-Verlag London Limited.

Dijkstra, J. and H. Timmermans (2002). Towards a multi-agent model for visualizing simulated user behaviour to support the assessment of design performance. In *Automation in Construction*, **11**, pages 135–145. Elsevier Science B.V.

Ferber, J. (1999) Multi-agent systems. An introduction to distributed artificial intelligence. Addison Wesley.

Gloor, C., D. Cavens, E. Lange, K. Nagel and W. Schmid (2003). A pedestrian simulation for very large scale applications. In A. Koch and P. Mandl, editors, *Multi-Agenten-Systeme in der Geographie* **23**, in Klagenfurter Geographische Schriften, pages 167-188. Institut fuer Geographie und Regionalforschung der Universität Klagenfurt.

Gloor, C., L. Mauron and K. Nagel (2003). A pedestrian simulation for hiking in the Alps. In *Proceedings of Swiss Transport Research Conference (STRC)*, Monte Verita, Switzerland. See http://www.strc.ch (accessed Feb. 2005).

Helbing, D., I. Farkas and T. Vicsek (2000). Simulating dynamic features of escape panic. *Nature*, **407**:487-490.

Helbing, D., F. Schweitzer, J. Keltsch and P. Molnar (1997). Active walker model for the formation of human and animal trail systems. *Physical Review E*, **56**(3):2527-2539.

Nishinari, K., A. Kirchner, A. Nazami and A. Schadschneider (2001). Extended floor field CA model for evacuation dynamics. In *Special Issue on Cellular Automata of IEICE Transactions on Information and Systems*, volume E84-D.

Raney, B. and K. Nagel (2004). An improved framework for large-scale multi-agent simulations of travel behavior. In *Proceedings of Swiss Transport Research Conference (STRC)*, Monte Verita, Switzerland. See http://www.strc.ch (accessed Feb. 2005).

Schadschneider, A. (2001). Cellular automaton approach to pedestrian dynamics – theory. In M. Schreckenberg and S. D. Shama, editors, *Pedestrian and Evacuation Dynamics*, Proceedings of the 1st international conference, pages 75-85. Springer.

Stucki, P. (2003). Obstacles in Pedestrian Simulations. Diploma thesis, Swiss Federal Institute of Technology, ETH Zurich, Switzerland.