A Modular Network Simulator for City Traffic based on CA

Heike Emmerich

Shizuoka University, Johoku 3-5-1, Hamamatsu 432, Japan

Abstract. In this contribution a modular simulation concept with the overall goal to provide a tool for the interactive generation, simulation, visualization and analysis of arbitrary urban networks is introduced. First approaches use AVS – Advanced Visualization Sytems[1] – as computational surface. AVS supports the modular concept by allowing for the computation of self–deviced modules and their combination with system intern ones. Computation of all module types is based on a CA approach. Results and evaluations of a first test of this concept are discussed as well as its relation to similar approaches.

1 Introduction

Like all applied sciences traffic research, too, relies on a permanent exchange between observation of reality and its abstraction by means of models. Various experimental investigations of traffic flow in the past[4][5][6][7] have revealed traffic to be a rather complicated phemomenon especially in the region of its maximum throughput. Models have helped to find descriptions in terms of phase transitions, hysteresis effects, bifurcations, catastrophic collapses etc. by focusing on analogies to well understood non-linear phenomena from other fields of sciences. Models, too, are the basis of computer simulation of traffic flow. Differential– and difference–equation approaches have recently been added by CA models[8]. As fully discrete and easy to parallise models they allow for high computational speeds[9][10]. The concept of the modular network generator introduced in this contribution makes use of this for the creation of a traffic investigation tool designed for traffic engineers dealing with planning, construction and with operations on traffic flow. The section "'Technical Realisation of the Concept"' gives an idea of its setup. The following section compares simulation results with counted rates for an example street configuration in Dortmund–Oespel. To conclude differences to resembling concepts are encountered.

2 Technical Realisation of the Concept

2.1 Traffic Modules

For a concept as general as possible the number of traffic modules is kept small:

- **Sourcemoduls** inject cars in the simulation network.

- **Destinationmoduls** absorb them.
- **Crossingmoduls** are responsible for simulating turning and waiting processes. They model traffic lights or right of way depending on the given situation.
- **Transfermoduls** transfer left–turning cars through the inner crossing area.
- **Streetmoduls** move vehicles on ordinary street segments.

All moduls are computed on CA basis with an underlying cell–grid. For the urban arteries of the simulated street networks a calibration of time–step is considered as follows:

$$\frac{7.5\frac{m}{cell} \cdot 4.5\frac{cells}{m}}{45\frac{km}{h}} = 2.7sec \tag{1}$$

Among other things this value is important for modeling the programs of traffic lights correctly. The movement of vehicles in the inner crossing area can be dissolved in detail due to an underlying cell–geometry.

2.2 The AVS–Surface

The AVS–Surface channels and displays all processes involved at the various hierarchical levels. An impression of the possibilities under the AVS surface is given by the following AVS–network–examplesession: A network consisting of four self-computed traffic modules is interactively generated and simulated. Upon simulation on the **same** surface a display and analysis of traffic evolution using AVS intern modules, such as the 'Display–Image–Module' and the 'Graph–Viewer' in this example, is possible. They give the space–time evolution of cars in the area of a hindrance in form of start– and stop–waves and an analysis of a single vehicle's velocity curve throughout the hindrance, respectively.

2.3 Simulationstructure

The simulation is started under AVS by generating the module network according to a real street network assigment. Upon adjustment of their parameters traffic moduls automatically write those parameters into a file. Actual simulation calculations are parallised under PVM. Thus a 'Call–Simulation–Modul' calls the 'PVM–Father–Process' using the parameter file. The 'PVM–Father–Process' distributes the computations amongst different computational units. Each computational unit writes its results into a file, which can be used by AVS for the purpose of visualization and analysis. Parallalisation together with computation on the basis of very few rather abstract CA–evolutionary rules allows for computational speeds faster than real time. Thus this approach can not only be used for planning and constructing traffic networks but also for online–operations on traffic flow. For a detailed development of concepts treating the latter it is refered to [3].

3 Test

The modular simulation concept was tested for the following region in Dortmund Oespel: For this street configuration turning rates became applicable by

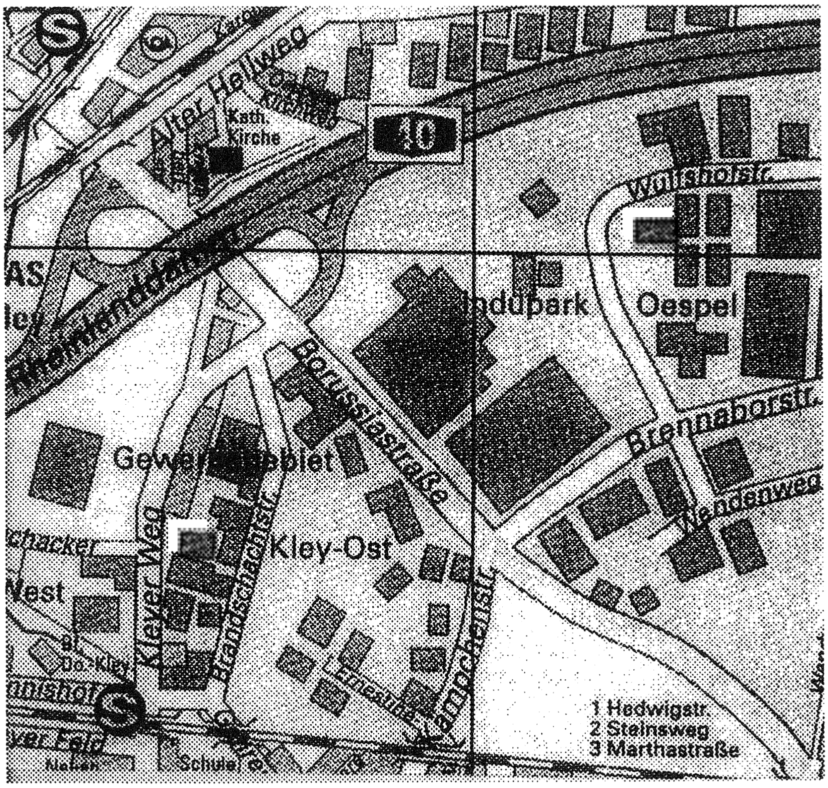

Fig. 1. Example street configuration in Dortmund–Oespel

the Dortmund City Traffic Department. These turning rates could very well be recovered by simulation. However, it has to be said, that the test was set up in exactly the way it was, because the turning rates were the only data material applicable. For a real evaluation of the concept this test is rather unsatisfying since data material was only coarse grained (given for 30 minutes intervals). Validation had to be done under the *asumption* that throughout the given interval traffic was stationary. However, turning rates of stationary traffic can very well be grasped by easy bilance–models. Thus this is not a real field of application for the simulator. It would be far more interesting to proof whether the turning points of traffic's dynamics can be recovered as well. Also details of the simulated traffic szenario like the jam patterns in front of a crossing should be related to real measurements. Thus further test have to be done with turning rates taken for much smaller time intervals and data, involving for example lengthes and evolutionary speeds of jams in front of traffic lights. Information like that could be given by positionable video cameras as used for example in Duisburg City.

4 Relation to similar approaches

There are two special features to this concept when compared with similar approaches:

1. **One** surface for interactive generation of traffic networks, their simulation, visualization and analysis
2. Detailed modulation of vehicle–movement in the crossing area.

On the basis of point 2. three cases of traffic evolution in front of a signal could be distinguished:

- **Case 1**: Crossing operates below capacity. Jams due to the signal's red–phase dissolve completely during the next green–phase.
- **Case 2**: Crossing still below capacity. Jams of various red–phases grow into each other.
- **Case 3**: Crossing above capacity. Congestion of turning lines with resulting impact on straight flow.

Only in **Case 3** simulation actually becomes necessary. Cases 1 and 2 allow for estimating calculations, as well, as shown in [2].

References

1. Advanced Visual System Inc.: AVS User's Guide. MA 02154 (1992)
2. H. Emmerich: Simulation im Strassenverkehr. Dissertation, Universitä Dortmund, Fakultät Bauwesen (1997)
3. J. Esser: Simulation von Stadtverkehr auf der Basis zellularer Automaten. Dissertation, Gerhard–Mecator–Universität–Gesamthochschule Duisburg, FB Physik–Technologie (1997)
4. Hall, F.L., Gunter, M.A.: Further Relationship of Flow–Concentration Relationship. Trans.Res.Rec. **1091** 1–9
5. Leutzbach, W.: Introduction to the Theory of Traffic Flow. Springer–Verlag, Berlin (1988)
6. May, A.D.: Traffic Flow Fundamentals. Prentice Hall, Englewood Cliffs, NJ (1990)
7. Treiterer, J.: Investigation of traffic dynamics by aerial photogrammetry techniques. Report No. PB **246 094**, Ohio State University, Columbus, Ohio
8. Nagel, K. , Schreckenberg, M.: A cellular automaton model for freeway traffic. J. Phys. I. France, **2:2221**, (1992)
9. Nagel, K. , Schleicher, A.: Parallel Comput. **20**, 125, (1995)
10. Rickert, M.: PAMINA – Parallel Microscopic Network Algorithm. Technical Report, Universität Köln, (1994)