Towards a Highspeed, Microscopic Traffic Simulation

Part 1

- Motivation and Background
- Simulation Scales
- Context of the Traffic Micro Simulation
- Fidelity of Traffic Models (Two Extremes)

Part 2

- Modifications of the 'Classical' CA
- Parallelization (Introduction)

Motivation

Need for mobility (business + leisure)

1993: 748.3 billion [pkm]

But: collapsing traffic flow $\longrightarrow$ restricted mobility and negative impact on environment

Traffic Simulation: conventional models contrate either on small regions with high fidelity or large regions with low fidelity
$\longrightarrow$ unsufficient to model today's traffic

Goal

Simulation of large scale regions at high fidelity in better–than–realtime

Required

- Efficient traffic models (CA / Hybrid)
- Parallel supercomputers

Simulation Scales

Fidelity

(quality of update rules)

CA $\Leftrightarrow$ Intelligent Objects

Time Scale

(smallest possible time step)

Tenth of a second $\Leftrightarrow$ Years

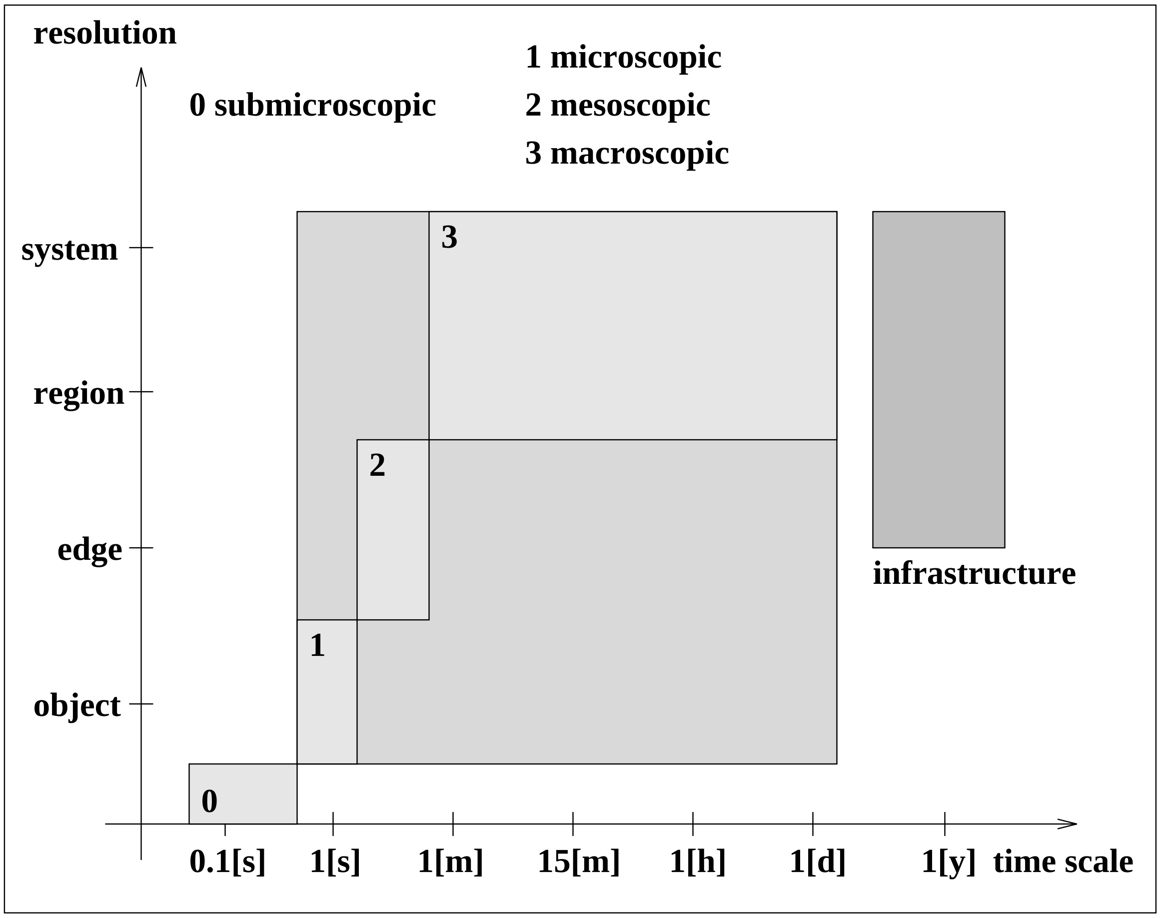
Resolution

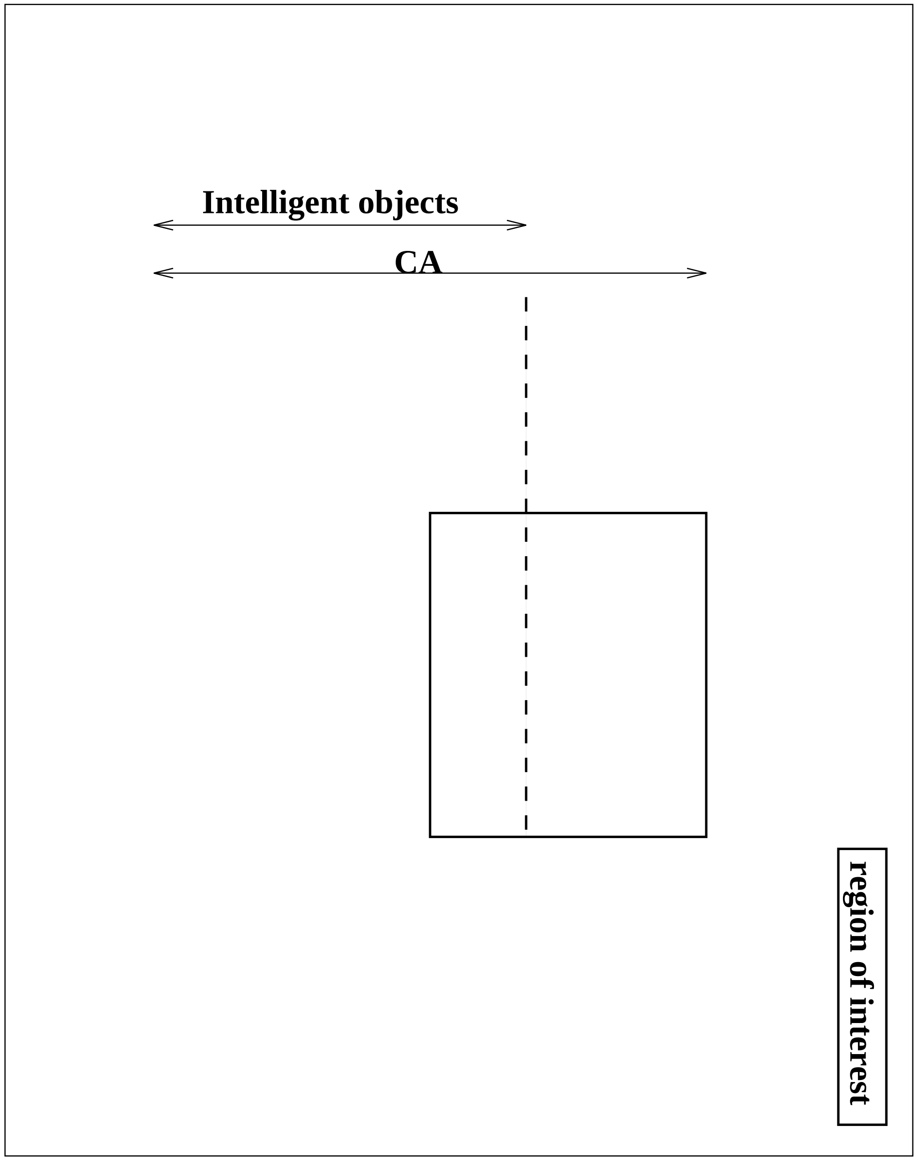
(smallest possible unit)

Single object $\Leftrightarrow$ Density in regions

Simulation Scales (cont'd)

Overview




Intelligent objects
CA
region of interest

Simulation Input

- Static Count Data from permanent traffic sensors along major street segments (highly aggregated)

- Demoscopic Data from traffic surveys (e.g. acceptance of measures controlling traffic flow) or census (e.g. composition of households: number of cars, number of persons)

- Network consisting of nodes and segments plus additional node and segment attributes

- Route Plans list of nodes or segments defining a path in network from source to destination

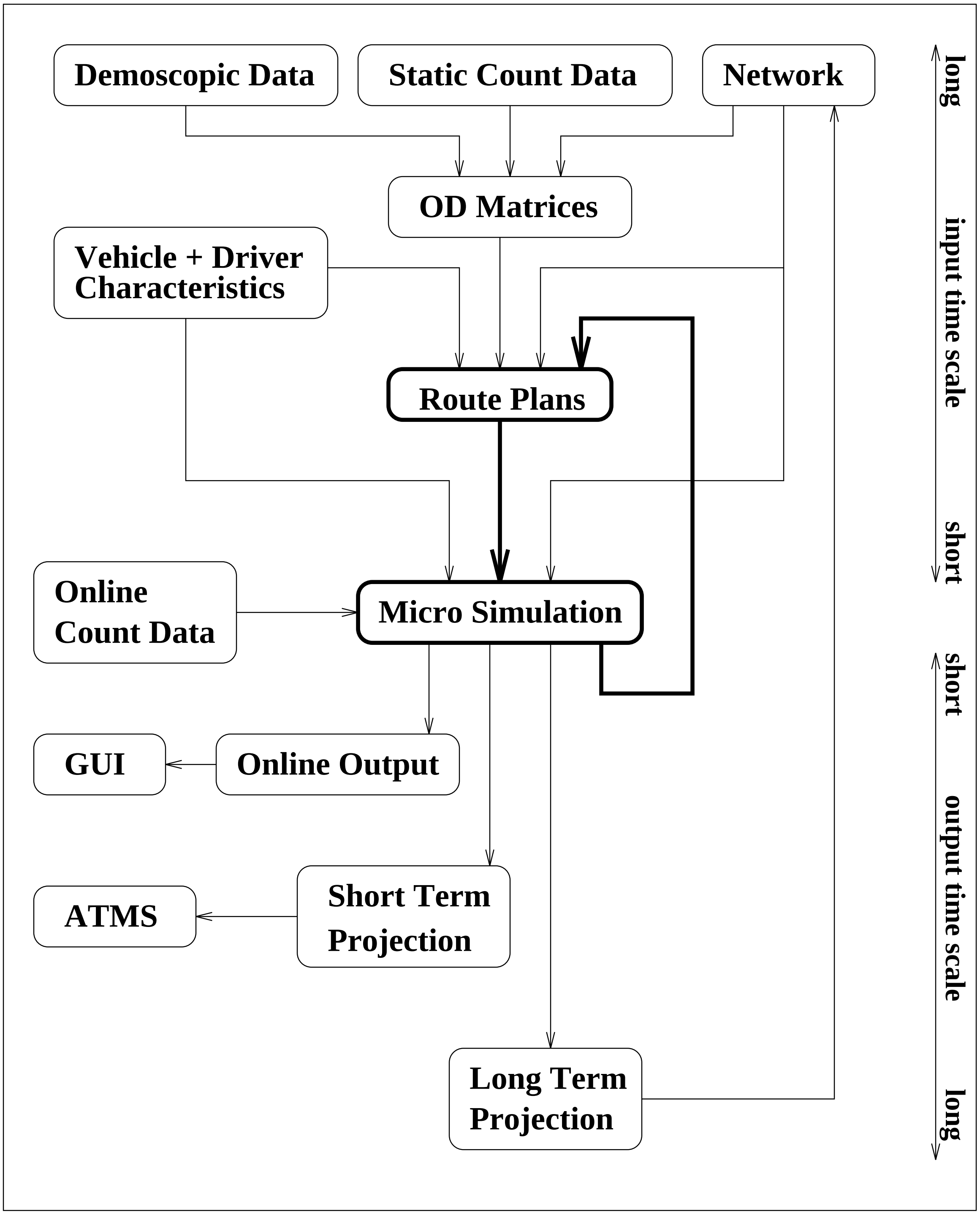
Simulation Input (cont'd)

- <u>Origin–Destination–Matrices</u> containing vehicle counts for each origin–destination pair ($M_{OD} = M_{OD}(t)$).
- <u>Vehicle + Driver Characteristics</u> like acceleration, maximum velocity, exhaust gases, desired velocity
- <u>Online Count Data</u> from permanent traffic sensors along major street segments (slightly aggregated)

Simulation Output

- Short Term Projection for e.g. traffic control measures (intelligent traffic signs, ATMS, IVHS) or individual vehicle routing (time critical!)
- Mid Term Projection for e.g. estimate of impact on environment (serves as input for meteorological models)
- Long Term Projection for e.g. planning of infrastructure
- Visualization

Simulation Context



Simulation Fidelity

Comparison of CA $\Leftrightarrow$ Intelligent Objects

Intelligent Objects: model driver behaviour and vehicle characteristics explicitly.
Cellular Automata: as simple as possible, only implicit modelling

characteristic	CA	IO
# rules	small	large
# parameters	small	large
fidelity	low	high?
consistency check	easy	difficult
computational speed	high	low
boundary definition	easy	difficult
validation	required	ideally not

Question

Will CA model be sufficient to model ALL traffic phenomena? For example: toxic components of exhaust gases increase up to 1000-fold beyond certain thresholds.

Try to find criteria to detect neccessity for high fidelity simulation $\longrightarrow$ Hybrid Simulation

Network

Possible Testbeds

a) Cologne

b) Aachen – Cologne – Duesseldorf

c) State Nordrhein–Westfalen

d) Germany

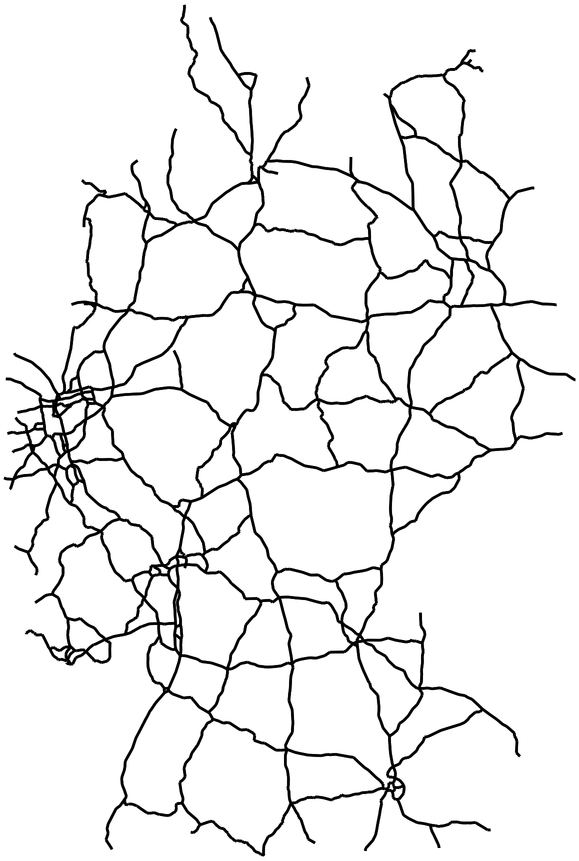
level	Autobahn [km]	nodes	edges
a	~ 200	~ 60	~ 70
b	~ 750	~ 240	~ 250
c	~ 2000	~ 550	~ 580
d	~ 12500	~ 3300	~ 3400

Include federal routes $\longrightarrow$ factor 5!

Network

German Autobahn Network

- 3300 nodes
- 3400 edges
- ~ 50000 kilometers (lane corrected)



Biggest Problem: Data

Data is either not available at all or only fragmented and in poor quality and/or resolution.

What we have:

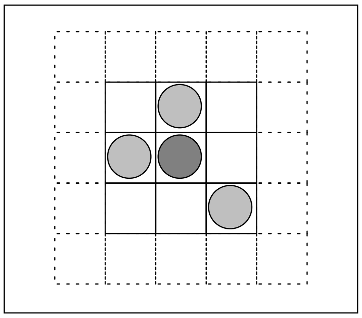
- location of nodes
- edges

What we need:

- number of lanes
- speed limits
- geometry of nodes (intersections)
- slope

Modifications of 'Classical' CA

Classical:



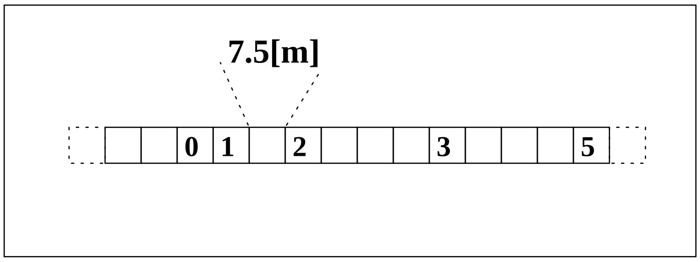
- discrete in time and space
- only nearest neighbour dependencies
- all sites and 'particles' are equivalent
- usually deterministic

Single Lane CA:

- look ahead of $v_{max} = 5$ sites
- stochastic rule to model erratic driver behaviour (traffic jams)

Single Lane CA

Nagel / Schreckenberg



1 Accelerate

$$v := \max(v_{max}, v + 1)$$

2 Avoid crash:

$$v := \min(gap, v)$$

3 Randomize:

$$rand() < p_{dec} \quad \Rightarrow \quad v := \max(v - 1, 0)$$

- Perform parallel update

Modifications of 'Classical' CA (cont'd)

Network Extension:

- sites have individual charateristics (speed limit, passing prohibition)
- multilane with lane changing rules
- vehicles have identities (individual route plan, 'individual' driving behaviour)

◇ refinement of CA parameters (acceleration and free–flow–velocity)

- variable vehicle lengths and possibly variable site lengths (promising: 3.75 [m]).

◇ additional passive functionality (handling of route plans through lane selection)

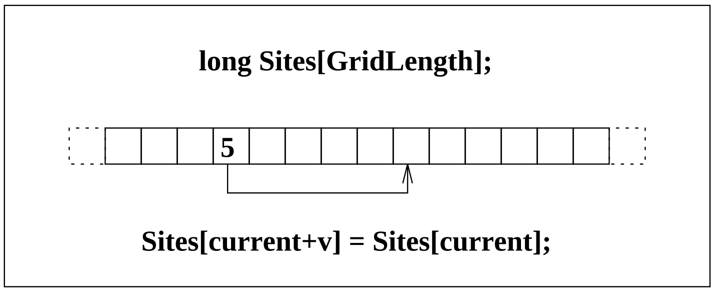
- definition of transition points between network elements

Modifications of 'Classical' CA (cont'd)

'Active' View

CA vehicle assumes new velocity and position depending on the configuration of its neighbours and its local <u>base data</u>:

- internal state (v)
- internal flags (lane changing behaviour)

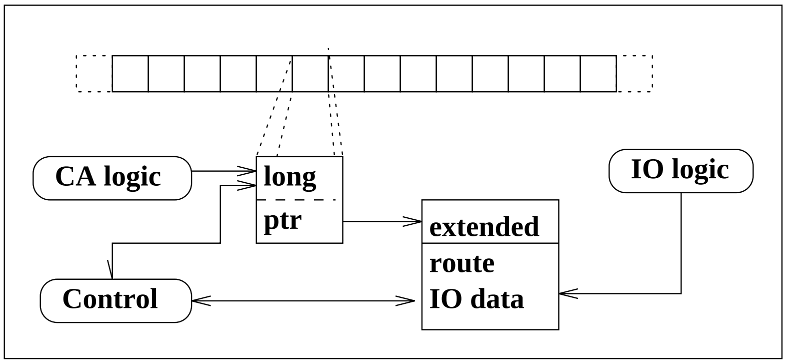


The ideal case is that all base data can be stored in one computer memory unit (C: long int). One update corresponds to moving the memory unit.

Modifications of 'Classical' CA (cont'd)

'Passive' View

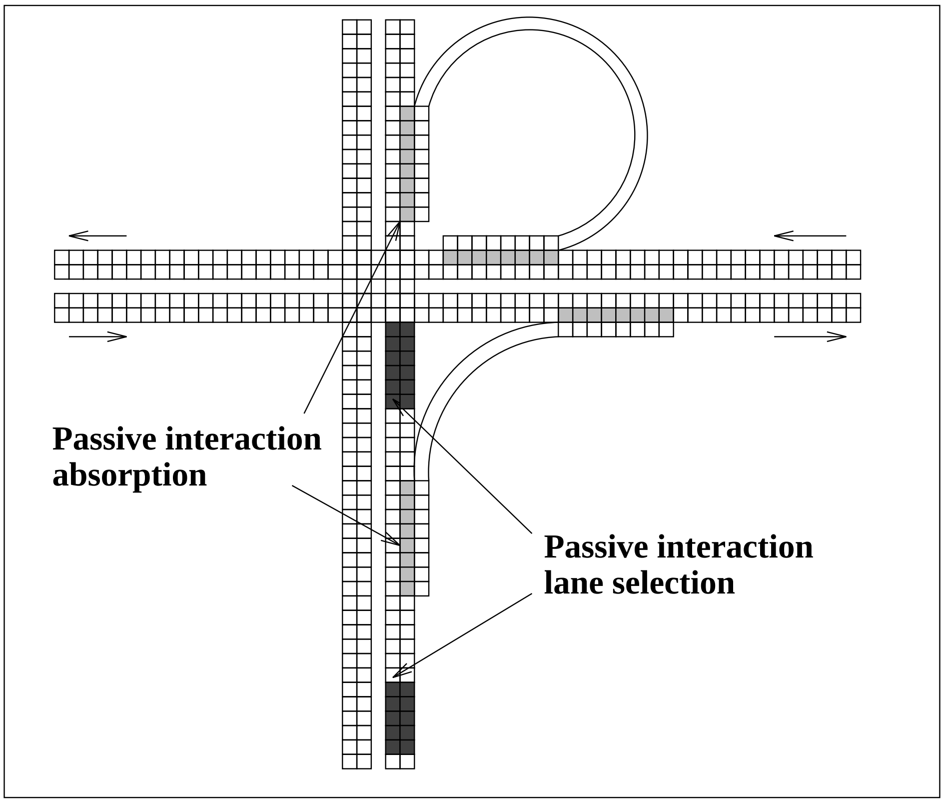
CA vehicle has additional pointer to <u>extended data</u> (route plan, IO-data). External objects use the extended data to modify the base data whenever neccessary, but as infrequently as possible.



Extended data available for both models $\longrightarrow$ easy switching between models in hybrid simulation.

Passive View cont'd

Example: intersection



Refinement of CA Parameters

Single Lane CA Parameters:

- ruleset itself (Holy Grale! Never touch!)
- constant maximum velocity v_{max}
- constant decelaration probability p_{dec}

Network CA Parameters:

- rule set $\longrightarrow$ rule set (flags)
- $p_{dec} \longrightarrow p_{dec}(vehicle, site, velocity)$
- $v_{max} \longrightarrow v_{max}(vehicle, site)$

$p_{dec}(v = 0 \ldots ..v_{max} - 1)$ defines acceleration envelope. v_{max} and $p_{dec}(v_{max})$ define exact free–flow velocity.

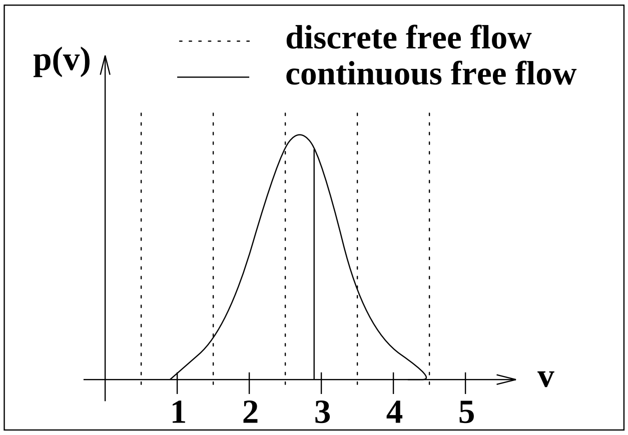
Refinement of CA Parameters (cont'd)

Continuous Free–Flow Velocity

Single lane CA has five discrete free–flow velocities

$$1 - p_{dec}, 2 - p_{dec}, \ldots v_{max} - p_{dec}.$$

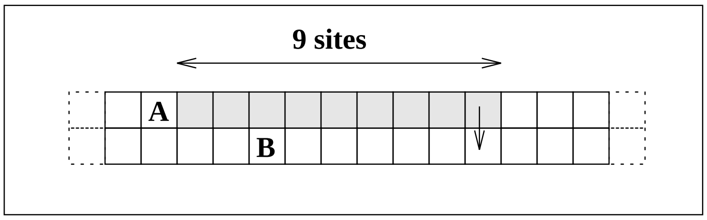
A relalistic vehicle fleet has a distribution of free–flow velocities. For a random, continuous value choose a combination of v_{max} and $p_{dec}(v_{max})$.



During the simulation vehicles are created with $(v_{max}, p_{dec}(v_{max}))$ pairs according to given input distributions.

Example: Passing of Trucks

Discrete free–flow velocity: truck A wants to pass truck B. It has to advance 9 sites (according to lane changing rules) before it can change back to the left lane.



p(A faster than B)	0.25	(a)
p(B faster than A)	0.25	(b)
p(A and B equally fast)	0.5	(c)

Neglect (c) and consider consider coordinate system located in B. A performs random walk around B. Probability to pass is $\sim 2^{-9} \sim 1/500$. Include (c): probability $\sim 1/1000$. The truck needs more than 15 minutes to pass!

Continuous free–flow velocity: difference of 0.1 in free flow velocity results in approximately 90 seconds.

Decision Horizon

= maximum look ahead length or radius in which objects are influenced by each other.

action	horizon	radius
simple CA	v_{max}	35 [m]
multilane CA	$c * v_{max}$	100 [m]
static route plans	segment	500 [m]
IOs	vision range	500 [m]
local rerouting		5 [km]
regional rerouting		5-100 [km]

Parallelization

Distributed Memory and Message Passing

- current standard in supercomputer hardware (dedicated systems: e.g. Intel XP/S-10, Cray T3D, Parsytec GCel-1024, workstation clusters, highspeed clusters IBM SP1, SP2)
- geometric distribution of street network
- exchange of traffic state in boundaries
- dynamic load balancing

Goal: keep boundaries small and infrequent
$\longrightarrow$ keep decision horizon as small as possible

Parallelization (cont'd)

Shared Memory + Multi Threading

- future? standard in hardware (e.g. Silicon Graphics Challenger)
- parallel computation of 'network units' in threads (e.g. segments)
- global availability of data

$\longrightarrow$ no restrictions for horizon

Conclusion

- Life is tough, simulation is worse ☺
- Micro simulation itself is only a small part of the deal.
- Classical CA can be extended in many aspects to cover net traffic simulation.
- Additional simulation features are strongly dependant on implementation techniques.

Further Topics (Coming Talks)

- High Fidelity Models (PW?)
- Visualization (CG?)
- Conclusion of Parallelization and Dynamic Load Balancing (MR)
- Data Retrieval and Validation (?)
- OD Matrices and Routing (?)