

Traffic Modelling

Reusing this material



This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.

<http://creativecommons.org/licenses/by-nc-sa/4.0/>

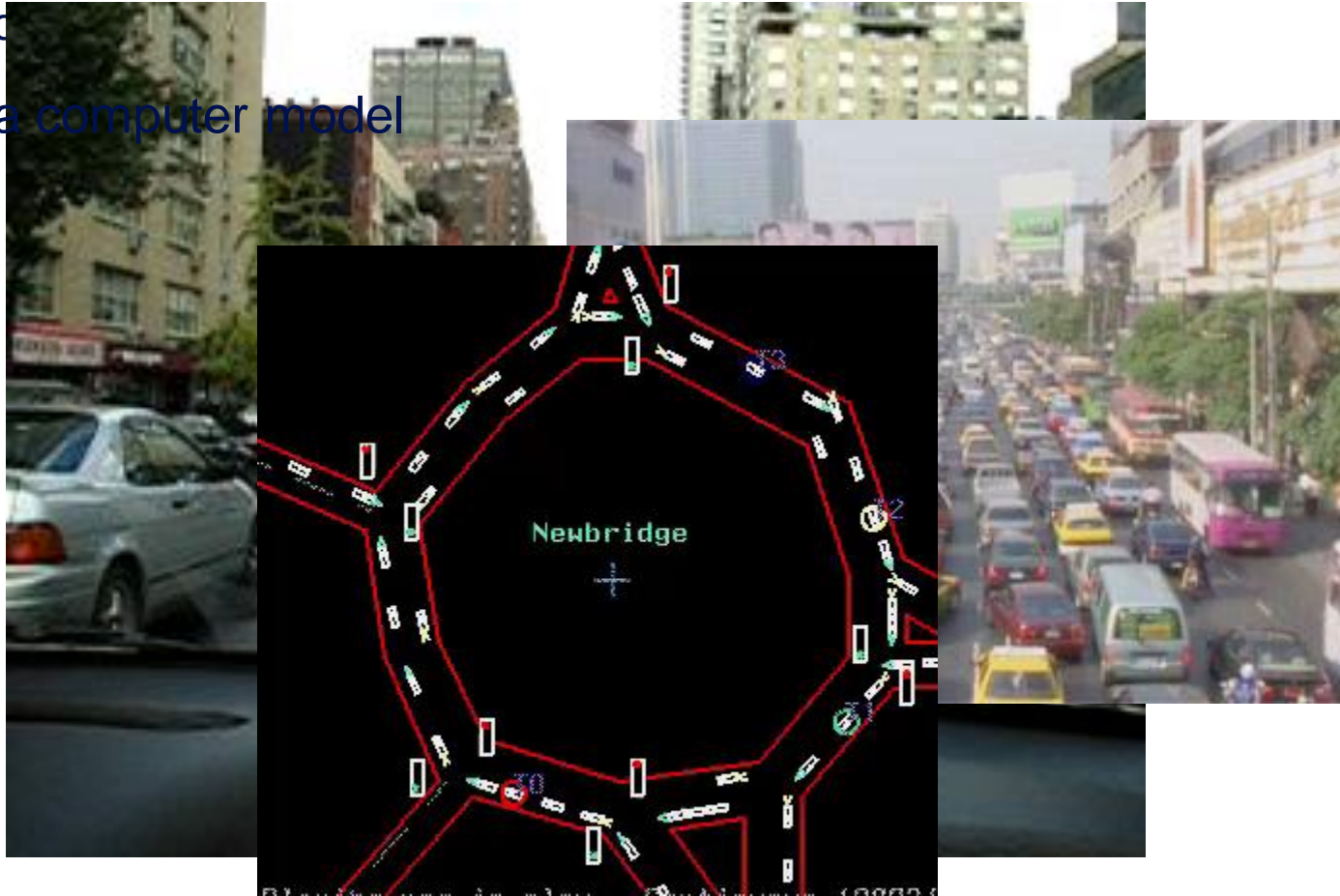
This means you are free to copy and redistribute the material and adapt and build on the material under the following terms: You must give appropriate credit, provide a link to the license and indicate if changes were made. If you adapt or build on the material you must distribute your work under the same license as the original.

Acknowledge EPCC as follows: “© EPCC, The University of Edinburgh, www.epcc.ed.ac.uk”

Note that this presentation contains images owned by others. Please seek their permission before reusing these images.

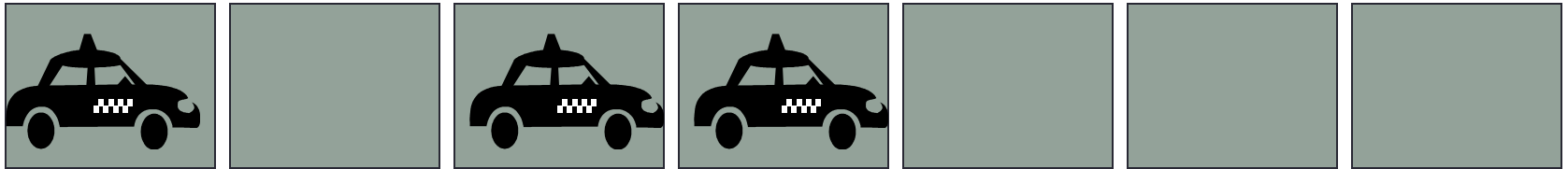
Traffic flow

- we want to predict traffic flow
 - to do this
- build a computer model



Simple traffic model

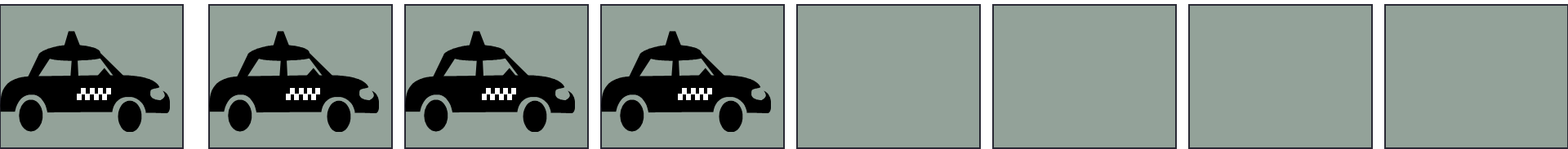
- divide road into a series of cells
 - either occupied or unoccupied
- perform a number of steps
 - each step, cars move forward if space ahead is empty



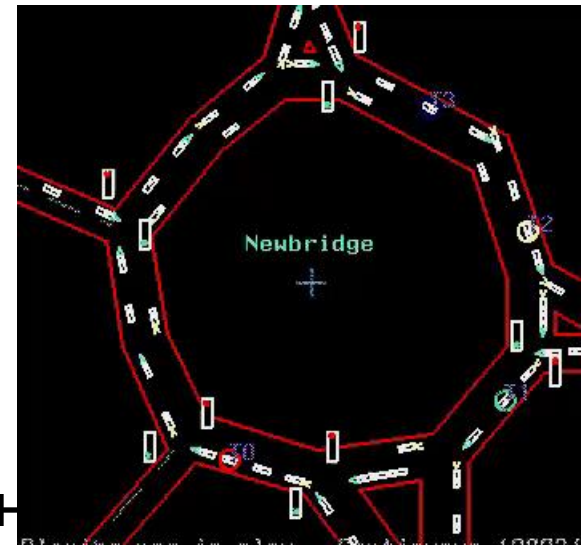
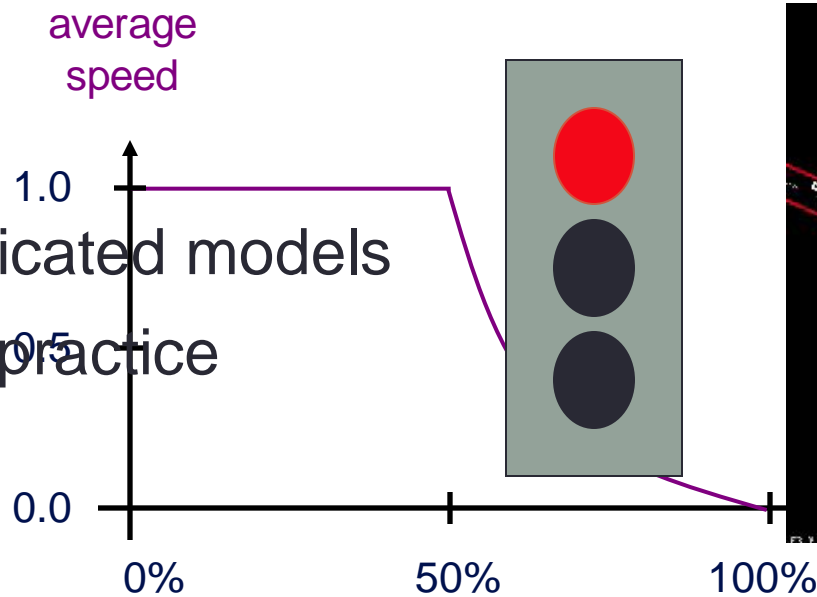
**could do this by moving
pawns on a chess board**

Traffic behaviour

- model predicts a number of interesting features
- traffic lights

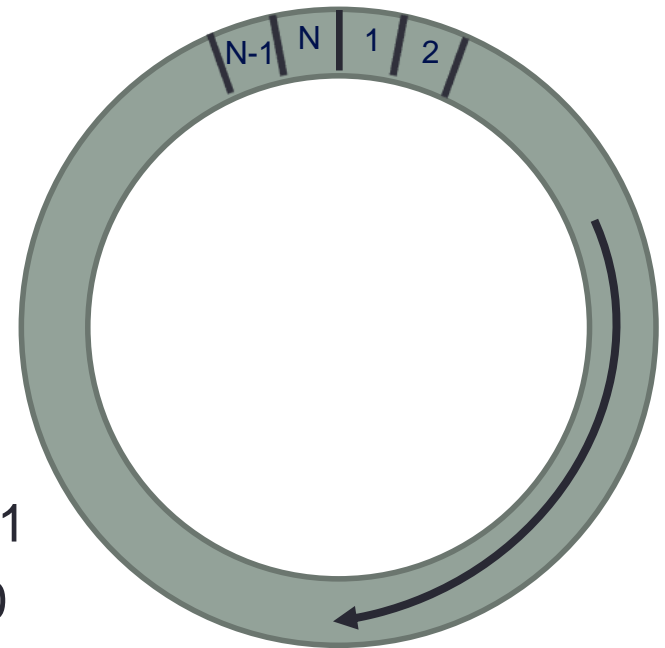


- congestion
- more complicated models are used in practice



Boundary conditions

- What happens if a car falls off the end of the road?
 - when does a car appear at the start?
- Consider a roundabout
 - periodic boundary conditions:
 - up from last cell N is first cell 1
 - down from first cell 1 is last cell N
- Implement with *ghost* or *halo* cells
 - road has $N+2$ cells 0, 1, 2, ..., $N-1$, N , $N+1$
 - copy cell 1 to cell $N+1$ and cell N to cell 0
 - then update cells 1 to N as normal



Pseudo Code: traffic on a roundabout

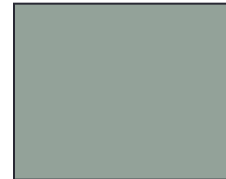
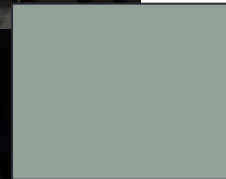
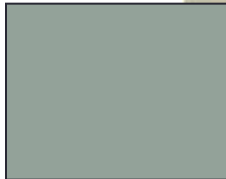
```
declare arrays old(i) and new(i),  $i = 0, 1, \dots, N, N+1$   
initialise old(i) for  $i = 1, 2, \dots, N-1, N$  (e.g. randomly)  
loop over iterations  
    set old(0) = old(N) and set old(N+1) = old(1)  
    loop over  $i = 1, \dots, N$   
        if old(i) = 1 (occupied)  
            if old(i+1) = 1 then new(i) = 1 else new(i) = 0  
        else (unoccupied)  
            if old(i-1) = 1 then new(i) = 1 else new(i) = 0  
        end loop over i  
    set old(i) = new(i) for  $i = 1, 2, \dots, N-1, N$   
end loop over iterations
```

also need to count number of moves

be careful to count EITHER car entering OR car leaving!

How fast can we run the model?

- measure speed in Car Operations Per second
 - how many COPs?
- around 2 COPs
- how fast would you expect?



Estimating speed

- Back of the envelope estimate
 - each cell requires about 7 operations per iteration
 - two if's, a cell update, a cell copy back, count moves, two loop constructs
 - my CPU operates at 2GHZ
 - so about 300 MCOPs?
- Could be faster
 - modern CPUs do multiple operations per cycle (e.g. vectorisation)
- Could be slower
 - real programs often limited by memory bandwidth not CPU speed
- Depends on problem size
 - small problems fit into fast cache memory