CGI2015 manuscript No.
(will be inserted by the editor)

DynaMoVis: Visualization of dynamic models for urban modeling

J. Dearden · M. W. Jones · A. Wilson

Abstract In association with Urban modelers, we have created DynaMoVis, a system for the visualization of dynamic models. The prediction of the evolution of urban and ecological systems is difficult because they are complex nonlinear systems that exhibit self-organization, emergence and path dependence. Without a good understanding of the dynamics, interventions might have unintended side-effects. This study aims to make progress in the understanding of dynamic models in the application areas of urban modelling. Analyzing these simulations is challenging due to the large amount of data generated and the high-dimensional nature of the system. We present a visualization system for exploring the behavior of a simulation from many different points of view. The system contains a number of different modes which allow exploration of: the simulation parameter space, the introduction and effect of noise on the simulation and the basins of attraction in the phase space of the simulation. Through use of this system it has been possible to develop a deeper understanding of the inter-dependencies in the models, their parameter spaces and corresponding phase spaces.

Keywords Dynamic Models · Urban Modeling · Visualization

1 Introduction

We present the results of an ongoing two year collaboration between ourselves and Urban modelers to speed-up and visualize the dynamic models used in urban models. Latterly we also have a collaboration with Biologists who use similar dynamic models for eco-system simulation. As a result of this later collaboration we have been able to identify common requirements, and therefore we have classified the more general problem of simulating and visualizing dynamic models of this type. We followed an iterative design and implementation process. After understanding the data and dynamic models of the urban modelers, we implemented visualizations that triggered further research questions and stages of our iterative process. Informed by the visualizations, we made identified that we could reduce the complexity of the computation by utilizing the reported hyperplane or hypervolume, which opened a further avenue of research for the experts. Through working with the Biologists we have identified the general research questions asked of dynamic models.

The contributions of this paper are:

- We present the various modes of operating dynamic models and map them onto specific example research questions, and signpost visualizations.
- Simulations started from a hyperplane are likely to detect all basins of attraction, and vastly reduces computational time.
- We can introduce noise into the simulation to represent external factors, and do so in a reasonable time due to the previous point.
- We introduce a multiple vector field plot visualization to depict the speed and direction of the simulation towards the attractors.
- We introduce a spline based parallel coordinates plot (PCP) with blending to show future probabilities from a starting state. The splines help the user to spot relationships amongst the parameters. The blending helps to understand the probability of solu-

J. Dearden · M. W. Jones (M.W.Jones@Swansea.ac.uk)
Visual Computing Group, Swansea

A. Wilson
CASA, UCL

tions. The PCP depicts discontinuities in the phase space.
- A fully interactive network visualization incorporating: nodes to represent attractors in phase space, plots at those nodes representing the attractor state, the degree of connectedness between nodes and the possible outflow or inflow between basins due to noise. We also have an adjacency matrix view, various interactive elements and linked views.

This tool is assisting model developers to expand their own research areas and aid their development of suitable models for predicting spatio-temporal dynamic interactions.

2 Related Work

Parameter Space Exploration Paraglide [2] is a system for assisting in setting model input parameters, running models, and comparing outputs. Users interactively define sampling spaces, clustering, testing and through this can run models using their directed input, then compare the parameters and output. Berger et al. [1] samples parameter space in a *star pattern* – along 1D axes around the target, or in a hypersphere around a target. Output is visualized using scatterplots, uncertainty visualization and parallel coordinates. The system assists engineers exploring parameter spaces for a turbo engine. Torsney-Weir et al. [21] generate samples of image segmentation parameters, execute the segmentation, measure against the ground truth, and present plots back to the user for interaction with in-filled simulated runs. The user can determine good control parameters, fill in more exact samples where necessary, and iterate towards a good parameter tuning for segmentation. The above methods present a matrix of plots based on HyperSlice [28]. Vismon [4] also explores parameter spaces and the sensitivity of the model to input parameters providing comparative and predictive tools to the user. Unger at al. [23] develop a visual system to aid validation of earth system simulations. Matkovic et al. [15] display the inputs, model structure and outputs for multiple simulation runs in order to gain insight into how a simulated system works - in this case a diesel engine. Bischi et al. [3] visualize the basins of attraction of a three firm oligopoly game in order to better understand bifurcations in the system. The game is a dynamical system which is similar to many symbolic urban models. We give some examples of three center retail models, but focus on higher dimensions. Some general methods for visualizing the behavior of high dimensional dynamical systems were presented by Gröller et al. [9]. These cover low-dimensional phase space representations. Basins of attraction are a concept linked to the idea of dynamical systems, of which many dynamic symbolic models are examples. Pitzer et al. [18] visualized the basins of attraction of heuristic algorithms and Orrell and Smith [16] visualized the Lorenz attractor.

Common to the first three approaches parameter (phase) space is sampled, running the accurate model, then in-filled with a predictive model, and providing tools to help the user manually target areas for running the full model. In contrast we use our hyperplane sampling to accelerate the model running. We provide our experts with different visualizations and interactivity, suited to their investigation of attractors and their associated basins. We also introduce variance runs to test model stability and present pertinent views. We also extend the multi-dimensional plots used above to include vectorial data, giving information about the speed and direction of the model. Regarding the classification of workflow, we found different requirements which we place in our taxonomy that were related to how simulations are carried out. We also provide network views of the simulations with variance that demonstrate automatic clustering and provide a useful means to select data for inspection through linked views.

Iconic Urban Models Since we concentrate on urban models as our application area in this paper, we review that literature here also. Urban models can be categorized as iconic or symbolic. Iconic models [19] are concerned with reproducing the look of a city while symbolic models reproduce how cities function. A survey of iconic modeling methods is presented by Vanegas et al. [26]. UrbanSim (http://www.urbansim.org) [27] software simulates the interactions between housing demand, real estate values, employment, household and others to produce per-cell values. These simulations are used in iconic urban models [24]. Vanegas et al. [25] tightly couple the simulation with modeling by allowing the designer to brush values for variables such as jobs or population, then iterate the dynamical system to equilibrium and generate the geometry. Symbolic rather than iconic models are the focus of our research.

Geographic visualization This has similar requirements to that of urban model visualization because both deal with similar datasets [17, 22]. Dykes and Brunsdon [7] demonstrate density estimation at different bandwidths to produce scale varying visualizations of geographical data. Chang et al. [6] display 3D urban models along with a parallel coordinates view and matrix view of demographic data. Guo [10] visualizes spatial flow (migration of the US population) by analyzing the flow. Flowstrates [5] provides an interactive view of human migration data. Our work differs in that we explicitly support the exploration of the phase space through simulation and visualization for the purposes of understanding the dynamic models, in order to aid the development of the science and systems underpinning urban modeling.

3 Phase Space Exploration

Terminology The n dimensional phase (or state) space contains all possible states of a dynamic system. Each nD point in phase space represents an nD state and each state is a unique point in the nD phase space. Each full parameter in the model is a dimension in the phase space. Dynamic models evolve over time from their initial condition to their converged state which is known as an attractor. The basin of attraction are all those initial states that converge to the attractor in question. During model evolution, the model passes through states (nD points) in phase space, and thus a simulation run traces out a path through the phase space. The model developer is aiming to make their model realistic. They seek to attain this by developing their model according to their understanding the phase space, simulation paths, the relationship of the attractors to initial conditions (basins of attraction) and the influence of noise on the system. Noise is of special interest as it gives some indication of the robustness of the system. If small variations in the model solve to different attractors the system is considered to be at a bifurcation point and chaotic. If a model is chaotic, then small errors in observations used to determine the parameters or small external influencing factors can have a high impact on solutions, and thus the system may not give robust predictions. Conversely, if the model is in a non-chaotic region, then errors in initial conditions or external influencing factors are tolerable and the system predictions can be treated as reliable.

Taxonomy Table 1 presents our classification of the modes of operation of a dynamic model. For each mode of operation, where applicable, we give an end-user use case and a model developer use case. We are concerned with model developers. Each successive row in our taxonomy represents an increase in model use complexity with a corresponding increase in computational time. Suggested typical visual encodings are given in the right-most column, along with references to our contributions in the remaining sections of this paper.

Urban Model Dynamic models typically cover time-dependent behavior and are represented using differential equations. In this section we briefly introduce the urban model used throughout this paper. A full description of the urban model is provided in the supplementary material as its development is the focus of the urban modelers and not this visualization research.

Model parameter	Explanation
e_i	Spend per person (within zone i)
P_i	Population count (within zone i)
W_j	Retail floor space of center j
c_{ij}	Cost of travel between i and j
m_j	Parking / transport costs to j
α	Influence of retail size on the model
β	Influence of travel cost on the model
$e^{-\beta m_j c_{ik}}$	Exponential decay (people prefer to travel to closer retail centers)
KW_j	The cost of running a center
ϵ	System response speed to inputs

Table 2 The terms used in Eq. 1.

$$\frac{dW_j}{dt} = \epsilon\left(\sum_i \left[\frac{e_i P_i W_j^{\alpha} e^{-\beta m_j c_{ij}}}{\sum_k W_k^{\alpha} e^{-\beta m_j c_{ik}}}\right] - KW_j\right) \tag{1}$$

Equation 1 is a system of nonlinear simultaneous differential equations, reflecting the nature of relationships between size and revenue in retail systems. These equations cannot be solved analytically: computer simulation and visualization are essential. Table 2 gives the notation for this model. Parameters can be constant if no subscript is used (e.g., α), depend on individuals if one subscript is used (e.g., P_i (zone population)), or account for interactions (e.g., c_{ij} cost of travel between zones and retail centers). Biological trophic models follow a similar form with population counts, population dispersal, interactions (e.g., $N_{ij} = k$ indicates the rate, k, that population i predates population j).

Currently, an average of approximately 1,400 model runs can be carried out per second (with an average of approximately 1000 time-steps per simulation).

3.1 Multiple model runs, regular sampling

In this mode, the model developer wishes to investigate how varying parameters influence the model output. It would be useful to explore the entire parameter space, but such brute force exploration typically leads to lengthy computation times. The model developer may choose to vary one parameter systematically to explore all the model outputs. We plot the model output on the y-axis, with the single varying parameter on the x-axis. Such an approach was taken by Orrell [16],

Mode of operation	End user	Model developer	Visual encoding
Single model run (e.g. for 10D model, 1 model run).	What happens to species A when species B is removed from the food chain? What happens to shopping center A when free parking is introduced at centre B?	Not utilized in this mode, see next mode.	Bar chart, histogram, probability plots [2], not the focus of this paper.
Multiple model runs (regular sampling of dimensions) [1], e.g. for 10D model, regularly sampling two of the dimensions with 21 samples equates to 441 runs and regularly sampling all 10D with 21 samples equates to ≈16 trillion model runs (i.e., practically impossible, but see hyperplane §3.1. 21 was chosen as it covers space sufficiently but still allowed a reasonable computation.	What happens to species A when differing numbers of species B are culled? What happens to shopping centre A when the rent per m^2 is varied?	How do parameter changes affect the model outputs? Model testing leading to model understanding.	Small multiples, e.g. Spectral bifurcation [16], parallel coordinates [13,14] and scatterplots [1], Fig. 1 (not the main focus of this paper).
Multiple model runs (introducing variance), e.g. for 10D model, can choose number of noise runs (e.g. 1000 model runs).	How stable are species populations (or retail centre size) based on current observed numbers and assumed parameters? (Using variance integrates any observational errors or external factors into the model).	Model testing leading to understanding. Understanding attractors and exploration of bifurcation and catastrophe theories.	Our new visualization §3.2, Figs. 3 and 4.
Multiple model runs (regular sampling), running each with n noise iterations (e.g. $n = 1000$), e.g. 10D with 21 samples per dimension and 1000 noise, ≈16,000 trillion model runs.		Complete exploration of basins of attraction and bifurcations. Comprehensive understanding of model.	Our new visualizations, §3.3, Fig. 5, made tractable using hyper plane idea §3.1.
Adaptive sampling rather than regular sampling, constraining variance (and adaptive sampling), e.g. using importance sampling / probability density functions or tracking features and sampling either side of the boundary.		As above, but reducing simulation time, and increasing resolution in critical areas (bifurcations).	Future research work. Achieved through manually directing sampling [1,2,21].

Table 1 We have identified the main utilization of a dynamic model. We give just a few examples of the types of questions each mode can answer for end-users and model developers/researchers (our work focuses on helping model developers). We also highlight the visual encoding that could be used for each mode.

who has produced bifurcation diagrams for the Lorenz dynamic model. In the approach 2D plots are created that are useful for determining bifurcations and chaotic behaviour. The visualisations will not scale to higher levels in our taxonomy since they appear to be limited in the numbers of attractors that can be visualised clearly. A second parameter may be varied producing multiple 2D plots which could be visualised using the technique of small multiples. We focused on the more complex problems posed by the higher levels of our taxonomy.

Locating attractors Exploring the entire parameter space to locate all of the attractors requires the simulation to run to equilibrium starting at each point in the phase space. As an example, we executed a small three centre retail model, allowing just floor area to vary as initial condition (3 dimensions). We sampled with a discrete spacing of 21 in each dimension resulting in $21^3 = 9261$ starting conditions for simulation. Each of the 9261 simulations were run to equilibrium, resulting in 3 attractors (we have found up to 7 by exploring parameter space). In this simpler simulation, each attractor represents a solution where that retail center dominates the other two. Visualizations of the

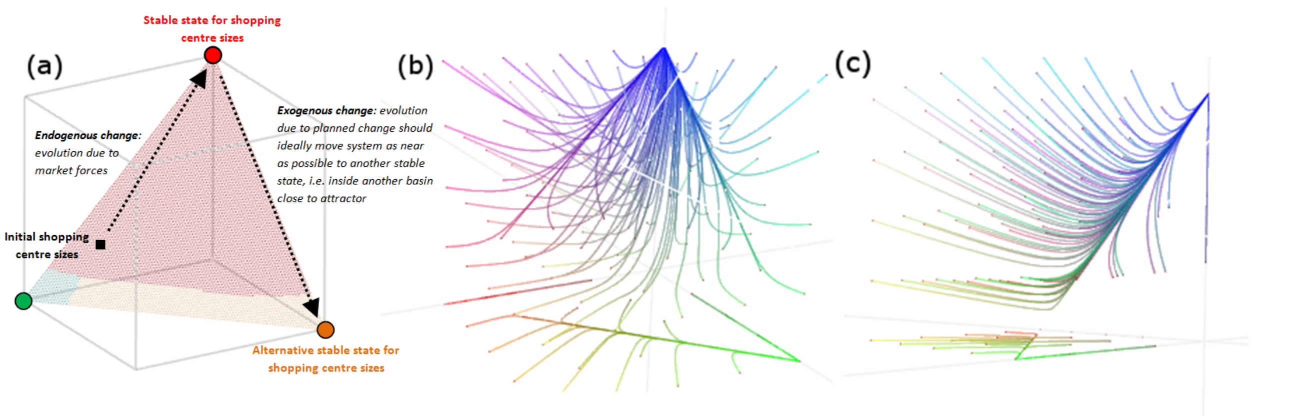

Fig. 1 (a) shows the two-dimensional equal-total hyperplane in three-dimensional phase space with labels providing one possible real world interpretation of the basins of attraction in the phase space of a dynamic retail model. (b) and (c) show a number of trajectories from uniformly spaced start points in three-dimensional phase space moving quickly to the two-dimensional equal-total hyperplane

paths through phase space (Figs. 1(b,c)) show the solution quickly converges to a point on a hyperplane then travels along the hyperplane to the attractor. This prompted the realization that the model has a certain amount of spending power supported by the population that ultimately dictates the amount of retail the system can support. Initial conditions with too much or too little retail quickly move to the hyperplane that represents the equi-total spending power (e.g. retail grows to accommodate spending power, then retail centers compete before one dominates these cases). Further experiments with the urban modeler confirmed this.

Hyperplane method We therefore use this as the basis for our approach. We generate points on the hyperplane as starting points for the simulation. Fig. 1(a) shows a hyperplane through the phase space. Each state on the hyperplane is colored according to its attractor. If an attractor represents a state we are trying to achieve through urban planning, and we start in a different basin from the attractor we want to move to, then we must use some planning policy tool to effect that (exogenous) change. This approach has significant impact on higher dimensional models that were previously impossible to explore. For 10 retail centers and 21 samples in each dimension, $21^{10} \approx 16.6$ trillion simulations would be needed for full exploration of floor space, but in this approach there are ≈ 14 million points on the hyperplane for the same spacing (300 years compared to just under 2 hours at the current speed of 1400 model runs per second).

Hyperplane visualization: Scatter plot approach Model developers wish to understand how models evolve and react to changing parameters, and do this through exploring the basins of attraction. Since they are ultimately targeting their models towards end-users they also wish to understand how a model could be used from that perspective. For our first design approach we chose to render multiple vector fields (like multiple scatter plots) where we plot pairs of dimensions against each other from the higher dimensional hyperplane (Fig. 2). Within each plot we indicate with uniformly sized arrows the direction and speed (using a Color Brewer two hue divergent scale as a continuous scale) of the model evolution towards the attractor. Attractors are indicated by differently sized circles, where size is a categorical label for each attractor. When the current real state of the model is known, this is depicted in the system using a white circle.

Case Study and Interpretation We built a basin network for an 8-zone South Yorkshire (UK) retail system. The largest 8 centers account for approximately 86% of the retail floorspace. The simulation finds 12 basins of attraction (around 0.1% of starting states did not converge to an attractor by the cut-off limit, but these are not considered further at this stage). From a self-interest view, a retail center owner could use the phase space to discover those basins of attraction where their retail center is successful. They could then determine which of the favorable basins is closest to the current state and exert change within their influence to adjust the vector field such that the direction from the current observed state favors their basin choice, and maximize the speed in that direction. For an altruistic planner, they would target stability in the system. They could adjust parameters in their control to bring about the coincidence of the model state with a region of low speed and hence stability.

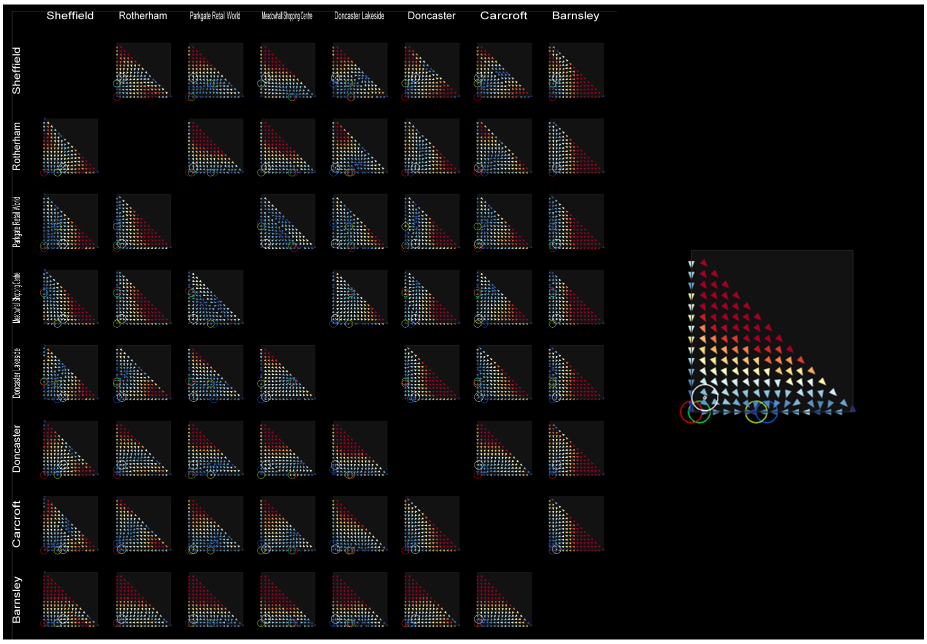

Fig. 2 Multiple vector field plot with zoomed in Parkgate against Rotherham on the right. A Color Brewer two hue divergent scale is used as a continuous scale for model velocity (red fast, blue slow). Model evolution direction and speed is encoded by arrow direction and color. Colored circles are a categorical label for attractors. The white circle is the current observed state (e.g. using real data).

3.2 Single starting state, multiple model runs with variance

Dynamic models are normally deterministic, but the urban modelers wanted to introduce noise into the model leading to non-determinism and the consequent investigation of chaos and bifurcations. The motivation for this is that it is impossible to create a fully accurate model. For example, urban models are sensitive to external influences which can affect the path of the system, e.g. boom and recession, chains of shops opening / closing, planning policies, investment, etc. Biological trophic models are sensitive to external factors such as weather patterns, changes of land use, etc.

We add a stochastic term to the dynamics to give:

$$\frac{dW_j}{dt} = \epsilon \left(\sum_i \left[\frac{e_i P_i W_j^{\alpha} e^{-\beta m_j c_{ij}}}{\sum_k W_k^{\alpha} e^{-\beta m_j c_{ik}}} \right] - K W_j \right) + W_j \phi \quad (2)$$

where ϕ is a stochastic term drawn from a normal distribution with a mean of zero, in this case, introducing a variation to the retail center size, although a multiplier could be integrated into alternative terms.

Visual encoding The end states of all the simulations are plotted together on a parallel coordinates system that is spline-based (Fig. 3). Each dimension represents the size of one retail zone in the end state of a simulation run. Using splines, rather than piecewise linear segments, also makes it easier to follow a single simulation run or cluster of similar simulation runs across all dimensions of phase space in the plot. The view is rendered in two stages. In the first stage, each spline is plotted with a small red color component value and blended (sourceBlend=One, destBlend=One, blendOperation=Add) so that when splines overlap their red color components are added together. The accumulated red color component value of each pixel is then equivalent to a count of the number of lines crossing that point. In the second stage, a pixel shader is applied that maps the red color component of each pixel onto a user selected color map. In these images we have used a single hue perceptually linear gradient although other maps can be chosen. This procedure makes it easier to compare the line density at different points across the view. Additionally white vertical lines indicate the range of minimum to maximum values for each dimension.

Interpretation The thickness of the combined overplotted splines for each dimension of the parallel coordinates plot illustrates the stability of that retail zone – a thin line (e.g. Fig. 3 part A) meaning there is little variation given noise in the system, while a thicker line (e.g. Fig. 3 part C) means there is more variation. If, for a single dimension, the splines pass through multiple points with a gap between (e.g. Fig. 3 part B) then this suggests that there are multiple possible end

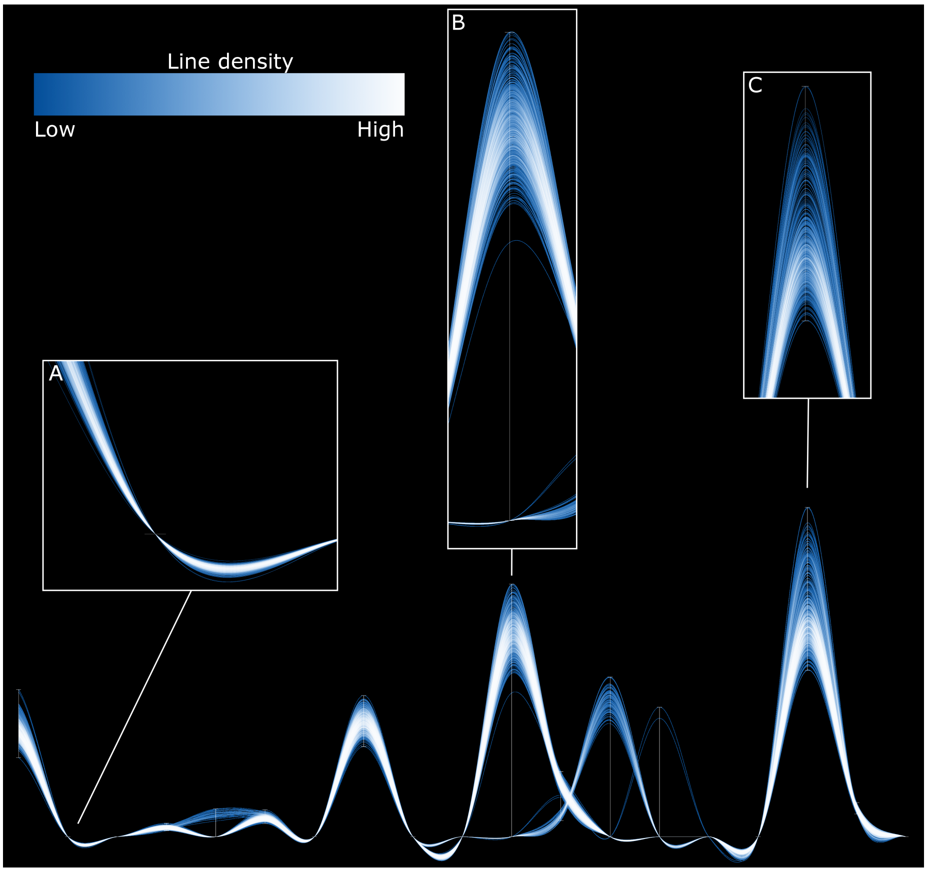

Fig. 3 Possible futures parallel coordinates spline density display of the effect of noise on multiple simulation runs: (A) little variation in a small retail zone. (B) multiple stable solutions for a retail zone. (C) large variation in a large retail zone.

states for this retail center given these input parameters and that the end states represent different attractors in phase space, the noise added to the model having caused a basin boundary to have been crossed. The color intensity indicates the likelihood of a particular state being reached. The advantage of this approach is that multiple attractors or equilibrium solutions can potentially be identified for each point in the parameter space. Alternatives are clustering [29], but such an approach would not give a cue for probability, continuous PCP [12] could be used, but without splines would not so clearly indicate the relationship we can see (in the example below). Curves have before been introduced to parallel coordinates and evaluated [8, 11, 20].

Case study 1000 simulation runs were run for retail centres of South Yorkshire with 4 percent noise (taking about 1 second). We used data from 2004 about retail zone size for South Yorkshire for validation in Fig. 4a. The possible futures generated are shown in Fig. 4b. All of the end states tend to be in one of two solutions or basins of attraction. The first is shown by the low density blue line over a large range through Rotherham and the second by the higher density whiter line going low through Rotherham. This suggests strong competition between Rotherham and Parkgate Retail World with Rotherham the more likely of the two to decline with Parkgate benefiting almost inversely proportional to each other. In reality these two centers are located very close to each other and Rotherham experienced a 30 percent decline by 2009. In other words, using this method in 2004 could have predicted a reasonable chance of decline and explored interventions if we wanted to avoid it. The use of splines allows us to follow the result clearly from axis to axis, clearing establishing relationships like the one just explored. We also see multiple outcomes and the color intensity indicates the strength of probability. This information is gained through one static image.

3.3 Multiple starting states, multiple model runs with variance

For this mode the simulation is run using noise runs as in the previous section, but for every initial state. For the urban model, each attractor comprises the simulated floor size parameters for each retail center and the whole system has the current set of parameters in use. For each initial state, we know the attractor it con-

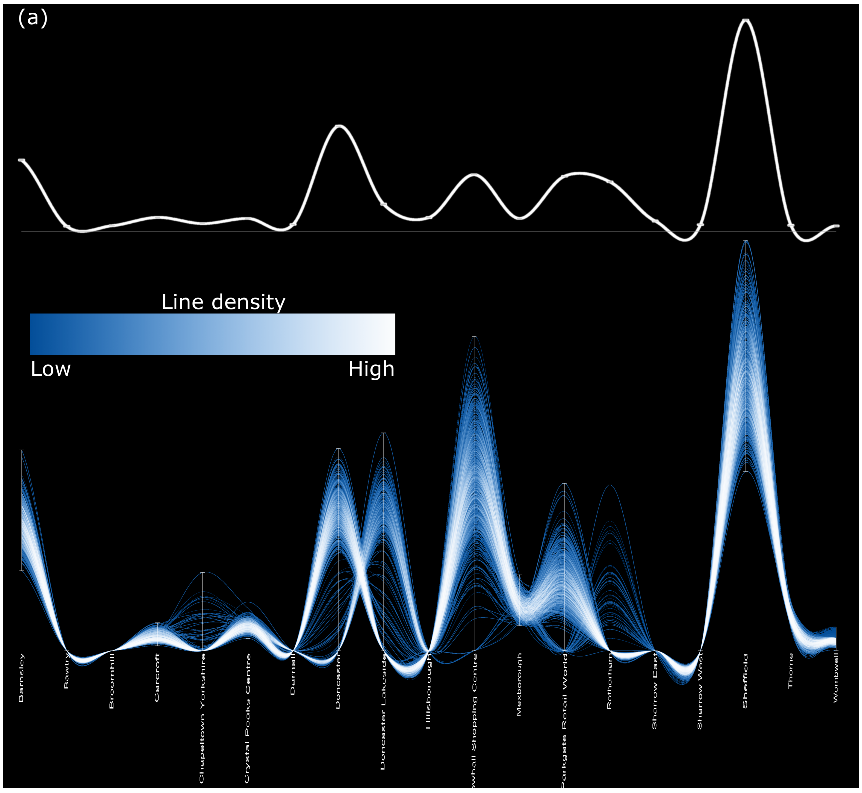

Fig. 4 (a) South Yorkshire retail zone size data for 2004 and (b) 1000 simulation runs run with 4 percent noise.

verges to, and also the attractors it converges to in the noisy runs. Together, this data makes up a *basin network*. We target expert model developers who want to understand basins of attraction, the start states that led to them, the effect of noise on their models, and probability of alternative futures in order to aid their own research to develop reliable models.

Visual encoding We tried graph layout tool Gephi (http://gephi.github.io/), which provided inspiration for our experts and our design. This background is provided as additional material. We implemented our own force-directed graph layout in order to integrate with our other views of the data to give multiply linked views for phase space exploration and to overcome some limitations. Our data consists of the attractors, initial states, and for each initial state we have the noise runs and percentage of runs that moved towards an alternative attractor. We also have the distance in nD phase space between the initial state and its attractor on the deterministic run. We can map any of this data to the visual channels of color (of node), size (of node) and length (of edge between attractor and state). We use force directed layout with an additional relaxation steps on the image plane. We give various views in the additional material, but for Fig. 5 we map distance to node size and attractor to color. Light edges connect initial states to their deterministic attractor. An opaque edge connects attractors that are neighbors in phase space (when noise is added some states move to the neighbor). The percentage of nodes flipping state is mapped to the two color divergent blue to red color map (blue=highly likely to fall to a different attractor, red=low) in the adjacency matrix. There are examples in the additional material where this is mapped to node color. The different views are linked with a single data point highlighted across views.

Interpretation This method ensures attractors become visible in the visualization since they become the center of large basins. The basins popout since they are the clusters of states that converge to that basin on the deterministic run. States that jump to alternative basins of attraction are notable using color in the examples in the additional material. The numbers of nodes in the cluster visibly give an indication of basin size.

There are several forms of interaction: Direct selection using mouse interaction (attractors, initial states, rectangular regions and multiple selections); Selections through an adjacency matrix view of the basins (clicking on the square selects the corresponding initial state nodes that bifurcate to both attractors); Filtering of selections (using sliders to filter in or out based on nD distance, and noise). The parameters of the nodes se-

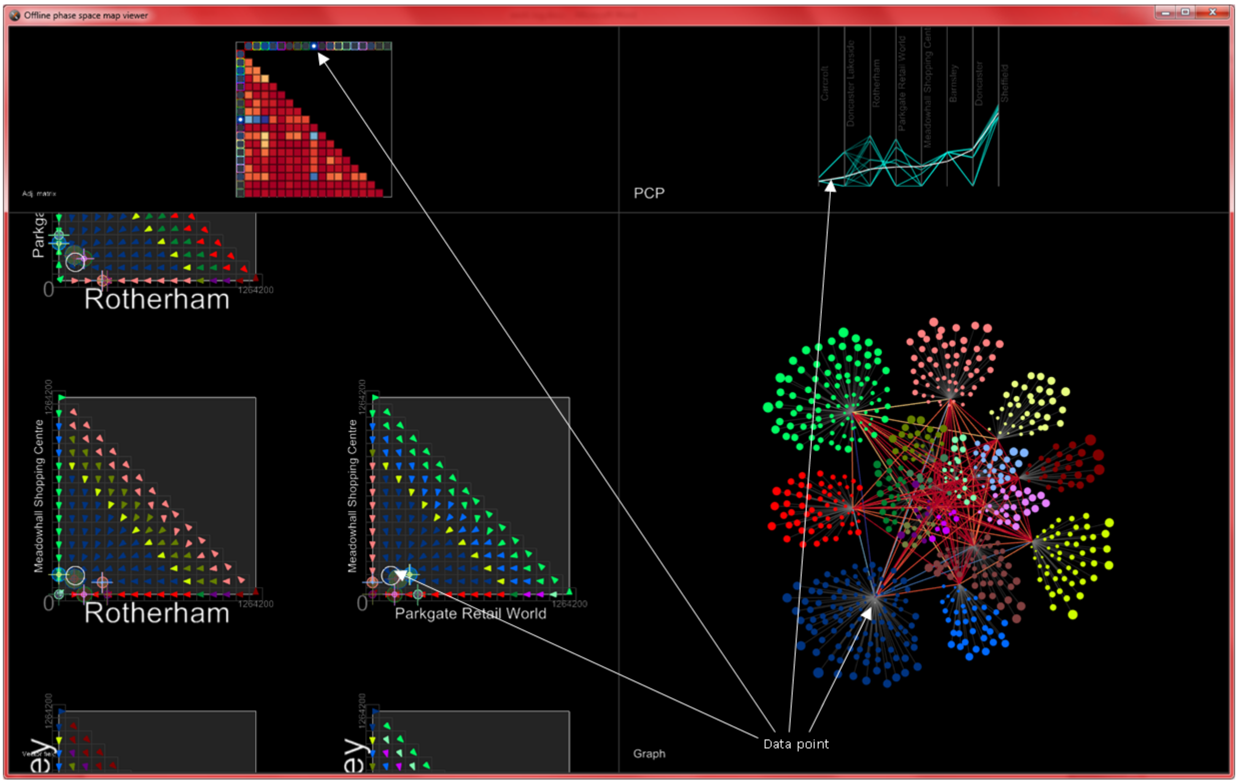

Fig. 5 Overall application. In the graph view we see the attractors / basins. Color indicates start states that converge to that basin. Size is distance from the basin in nD space. A network connecting basins is displayed where an edge means the two attractors have states that jump to the alternative attractor in noisy runs. We can switch the graph view between any of the alternative views given in the additional material where properties are mapped to different elements of the visualization.

lected (i.e., retail floor size) are plotted on the PCP view of the data which can be flipped between splines or linear piecewise segments. The user can zoom and pan each display independently.

4 Feedback and Summary

We have created DynaMoVis as a system for the visualization of dynamic models. We have categorized the main functionality of such models, and explored useful visualizations for each category, designed with our experts. We have presented an urban model case study throughout the paper to demonstrate our spline based PCPs for noise simulations, our network visualization for basin and attractor understanding and multiple vector field plots for understanding model velocity and direction.

We have written our system with two urban modelers and two biologists, who work within larger groups for wider dissemination and use. From our urban modeling domain experts, their feedback on contributions are: The hyperplane observation which became apparent when visualizing the data and was verified by them is an observation that enables a massive reduction in the model runs. Therefore, for the first time, they have been able to compute models over wide parameter spaces with noise. These additional modes of operation and the interactive querying has led for the first time for them to be able to understand the interaction of basins with the view of model velocity and give *a more complete view of their models than before*. From our biologist colleagues, they also used nested loops to iterate over Matlab code for exploring parameters over their models, so likewise the massive improvement in speed has led to great changes in their research scope. We are currently exploring variations on the hyperplane idea where we use rule based approaches to cut down the domain of parameters for model input. The most important contribution from their viewpoint is that the already realized implementation for urban modelers that includes the exponential decay term for distance can allow their models to include population interactions correctly taking into account distance. The basin network view allows them to understand the stability of populations and answer questions such as *What happens to other species if species is removed from the model*. The model velocity allows them to understand how quickly species conditions are changing. One interesting outcome of the cross-discipline work has been the generalization of the approach, and also the application of models in one area across to another.

References

1. Berger, W., Piringer, H., Filzmoser, P., Gröller, E.: Uncertainty-aware exploration of continuous parameter spaces using multivariate prediction. In: Proceedings of the 13th Eurographics / IEEE - VGTC Conference on Visualization, EuroVis'11, pp. 911–920 (2011)
2. Bergner, S., Sedlmair, M., Möller, T., Abdolyousefi, S., Saad, A.: Paraglide: Interactive parameter space partitioning for computer simulations. Visualization and Computer Graphics, IEEE Transactions on **19**(9), 1499–1512 (2013)
3. Bischi, G.I., Mroz, L., Hauser, H.: Studying basin bifurcations in nonlinear triopoly games by using 3D visualization. Nonlinear Analysis: Theory, Methods & Applications **47**(8), 5325 – 5341 (2001)
4. Booshehrian, M., Möller, T., Peterman, R.M., Munzner, T.: Vismon: Facilitating Analysis of Trade-Offs, Uncertainty, and Sensitivity In Fisheries Management Decision Making. Computer Graphics Forum **31**(3), 1235–1244 (2012)
5. Boyandin, I., Bertini, E., Bak, P., Lalanne, D.: Flowstrates: An approach for visual exploration of temporal origin-destination data. Computer Graphics Forum **30**(3), 971–980 (2011)
6. Chang, R., Wessel, G., Kosara, R., Sauda, E., Ribarsky, W.: Legible Cities: Focus-Dependent Multi-Resolution Visualization of Urban Relationships. IEEE Transactions on Visualization and Computer Graphics **13**, 1169–1175 (2007)
7. Dykes, J., Brunsdon, C.: Geographically weighted visualization: Interactive graphics for scale-varying exploratory analysis. IEEE Trans. Vis. Comput. Graph. **13**(6), 1161–1168 (2007)
8. Graham, M., Kennedy, J.: Using curves to enhance parallel coordinate visualisations. In: Information Visualization, 2003. IV 2003. Proceedings. Seventh International Conference on, pp. 10–16 (2003)
9. Gröller, E., Löffelmann, H., Wegenkittl, R.: Visualization of dynamical systems. Future Generation Computer Systems **15**(1), 75 – 86 (1999)
10. Guo, D.: Flow mapping and multivariate visualization of large spatial interaction data. Visualization and Computer Graphics, IEEE Transactions on **15**(6), 1041 –1048 (2009)
11. Heinrich, J., Luo, Y., Kirkpatrick, A.E., Weiskopf, D.: Evaluation of a bundling technique for parallel coordinates. In: GRAPP & IVAPP 2012, pp. 594–602 (2012)
12. Heinrich, J., Weiskopf, D.: Continuous parallel coordinates. Visualization and Computer Graphics, IEEE Transactions on **15**(6), 1531–1538 (2009)
13. Inselberg, A.: The plane with parallel coordinates. The Visual Computer **1**(2), 69–91 (1985)
14. Liu, S., Cui, W., Wu, Y., Liu, M.: A survey on information visualization: recent advances and challenges. The Visual Computer **30**(12), 1373–1393 (2014)
15. Matkovic, K., Gracanin, D., Jelovic, M., Ammer, A., Lež, A., Hauser, H.: Interactive visual analysis of multiple simulation runs using the simulation model view: Understanding and tuning of an electronic unit injector. Visualization and Computer Graphics, IEEE Transactions on **16**(6), 1449–1457 (2010)
16. Orrell, D., Smith, L.A.: Visualizing bifurcations in high dimensional systems: the spectral bifurcation diagram. International journal of bifurcation and chaos **13**(10), 3015–3027 (2003)
17. Pinnel, L.D., Dockrey, M., Brush, A.B., Borning, A.: Design of Visualizations for Urban Modeling. In: VisSym 00 : Joint Eurographics - IEEE TCVG Symposium on Visualization, pp. 199–208 (2000)
18. Pitzer, E., Affenzeller, M., Beham, A.: A closer look down the basins of attraction. In: 2010 UK Workshop on Computational Intelligence, UKCI 2010, pp. 1–6 (2010)
19. Pueyo, O., Patow, G.: Structuring urban data. The Visual Computer **30**(2), 159–172 (2014)
20. Theisel, H.: Higher order parallel coordinates. In: VMV, pp. 415–420 (2000)
21. Torsney-Weir, T., Saad, A., Moller, T., Hege, H.C., Weber, B., Verbavatz, J.M.: Tuner: Principled parameter finding for image segmentation algorithms using visual response surface exploration. IEEE Transactions on Visualization and Computer Graphics **17**(12), 1892–1901 (2011)
22. Turkay, C., Slingsby, A., Hauser, H., Wood, J., Dykes, J.: Attribute signatures: Dynamic visual summaries for analyzing multivariate geographical data. Visualization and Computer Graphics, IEEE Transactions on **20**(12), 2033–2042 (2014)
23. Unger, A., Schulte, S., Klemann, V., Dransch, D.: A visual analysis concept for the validation of geoscientific simulation models. IEEE Transactions on Visualization and Computer Graphics **18**(12), 2216–2225 (2012)
24. Vanegas, C.A., Aliaga, D.G., Beneš, B., Waddell, P.: Visualization of simulated urban spaces: Inferring parameterized generation of streets, parcels, and aerial imagery. IEEE Transactions on Visualization and Computer Graphics **15**(3), 424–435 (2009)
25. Vanegas, C.A., Aliaga, D.G., Beneš, B., Waddell, P.A.: Interactive design of urban spaces using geometrical and behavioral modeling. ACM Trans. Graph. **28**(5), 111:1–111:10 (2009)
26. Vanegas, C.A., Aliaga, D.G., Wonka, P., Mller, P., Waddell, P., Watson, B.: Modelling the appearance and behaviour of urban spaces. Computer Graphics Forum **29**(1), 25–42 (2010)
27. Waddell, P.: Urbansim: Modeling urban development for land use, transportation and environmental planning. Journal of the American Planning Association **68**, 297–314 (2002)
28. van Wijk, J., van Liere, R.: Hyperslice. In: Visualization, 1993. Visualization '93, Proceedings., IEEE Conference on, pp. 119–125 (1993)
29. Zhou, H., Yuan, X., Qu, H., Cui, W., Chen, B.: Visual clustering in parallel coordinates. Computer Graphics Forum **27**(3), 1047–1054 (2008)