Multidimensional Visualization of Transition Systems

A. Johannes Pretorius, Jarke J. van Wijk
Technische Universiteit Eindhoven
a.j.pretorius@tue.nl, vanwijk@win.tue.nl

Abstract

Transition systems are graphs and it is natural to consider their visual representation as a graph visualization problem. They also pose an interesting multidimensional visualization challenge since every state may be considered as a point in n-space. We discuss a number of approaches toward projecting transition systems to the 2D plane by considering the dimensionality of the states. To asses these techniques, we consider their ability to assist us in answering questions that are difficult to address with conventional methods.

1. Introduction

A state transition system consists of a set of states and a set of transitions and describes a system whose states evolve over time. Transitions are source-action-target triples where the source and target are states and the execution of an action triggers a change of state. Specific actions can only be executed from certain states and under certain conditions.

Transition systems may be generated from process algebraic descriptions of system behavior written in languages such as μCRL [4]. In μCRL the behavior of parallel processes with data can be characterized. Such a description is transformed to a transition system where transitions are labeled and states consist of a valuation of data variables. These variables correspond to the union of all data parameters in the specification, extended with parameters introduced by the translation program. Consequently, the dimension n of every state is equal and there is a strong relationship between the values that the variables assume and the original specification.

Due to the complexity of such transition systems, insight into their structure is often limited. There are two main approaches for their analysis. The first is to derive an abstracted, much smaller variant of the system (typically less than 100 states) that is visualized as a graph by drawing all states and transitions explicitly. Alternatively, specific questions, such as whether the system is free of deadlock, may be formulated and checked (using modal logic, for instance). If it is not possible to define such questions precisely, the transition system cannot be adequately analyzed.

Interactive visualization adds a third technique. Results by Van Ham et al. [10] show that when a transition system is visualized and the users are enabled to interrogate this representation, their knowledge of the system is substantially enhanced. Their method performs pre-processing on the structure of a transition system to generate a tree that serves as a backbone onto which all states and transitions are projected. Ranking and clustering results in layers and branches from which different phases in system behavior are clearly discernable.

2. Challenge

Whereas Van Ham et al. [10] consider the visualization of transition systems as a pure graph visualization problem, it is also a multidimensional visualization challenge. For the transition systems that we consider every state consists of a vector of instantiated state variables. Therefore, we consider a state $s = [s_1, ..., s_n]$ as a vector in n-space, where s_i is a valuation of the i^{th} state variable. The cardinality C_i of the i^{th} variable is usually in the range of 2–20. Also, the dimensionality n is typically in the order of 10–40.

The high dimensionality of such transition systems reflects their complexity. They are finite state machines and the sequence of values that state variables assume represent their behavior. Still, using conventional methods it is often hard to gain insight into the behavior reflected by the variables. A number of questions are difficult to answer:

1. Are there particular subsets of state variables that exhibit interesting behavior? Also, which variables are uninteresting and may be disregarded?
2. For a specific subset of state variables, what does the behavior of the system look like?
3. Are there interesting correlations between subsets of state variables? For instance, are certain variables dependent on others? In particular, it is often difficult to detect correlations between more than two variables.
4. Are there close correlations between actions and state

variables? For instance, are the values that particular variables assume dependent on specific actions?

5. For a subset of state variables, are there states that assume a particular combination of values? Also, do transitions exist from one such state to another?

The ability to answer these questions allows the user to verify functional requirements that have been stipulated for a system. They are also useful for verifying that algebraic specifications are correct. Finally, these questions allow the user to gain insight into systems about which little is known. Although it is often possible to come up with answers using algebraic proof techniques, this requires significant effort and is error prone. Motivated by these challenges and inspired by Van Ham et al.'s [10] results, in the remainder of this paper we consider a number of approaches for visualizing transition systems from a multidimensional perspective.

3. Analyzing dimensions

A common approach for multidimensional data visualization is to use low-dimensional projections into 2D or 3D. Due to a number of issues that arise when rendering 3D scenes on a 2D monitor, 2D visualizations are almost always more appropriate and are widely used [9, 13]. Several researchers have proposed that the user be directly involved in choosing an adequate 2D projection by considering different dimensions individually and with respect to others. Examples include the *rank-by-feature* framework [8] and *hierarchical dimension reduction* [14]. It is argued that the user has a better understanding of results than with automated techniques such as *grand tours* [1].

In order to reduce the complexity of a multidimensional data set, another common approach is to select a subset of the original dimensions and to visualize the data with respect to these [8, 14]. In our case the user is likely to be interested in different subsets of state variables at different times. For instance, users may want to consider specific variables that allow them to verify whether certain functional requirements are satisfied.

To explore different dimensions, the user must be provided with a good overview that makes it easy to compare dimensions [9, 8, 14]. Our point of departure is to consider the m states of dimension n as n arrays of dimension m. These n arrays correspond to the state variables and are visualized as n *parallel histograms* (see Figure 1). States are distributed along the x-axis (typically according to a breadth-first traversal of the corresponding graph). In order to identify overall patterns, the histograms are normalized, making them scaling invariant.

Interesting state variables are identified by visually discovering patterns, correlations, gaps and outliers in the data set. Histograms are also labeled, enabling the user to identify specific variables. Individual histograms may be selected, deselected, dragged to and dropped in new vertical positions. As will be seen in following sections, the vertical ordering of dimensions has a direct influence on the positions to which states are rendered in eventual 2D projections. Deselected dimensions have no such influence. In this way, the parallel histograms serve as a filtering and zooming device and facilitate *interactive dimension reduction*. The use of parallel histograms combined with an approach to view the resulting low-dimensional data sets also adheres closely to Shneiderman's mantra [9]: *"overview first, zoom and filter, then details on demand"*.

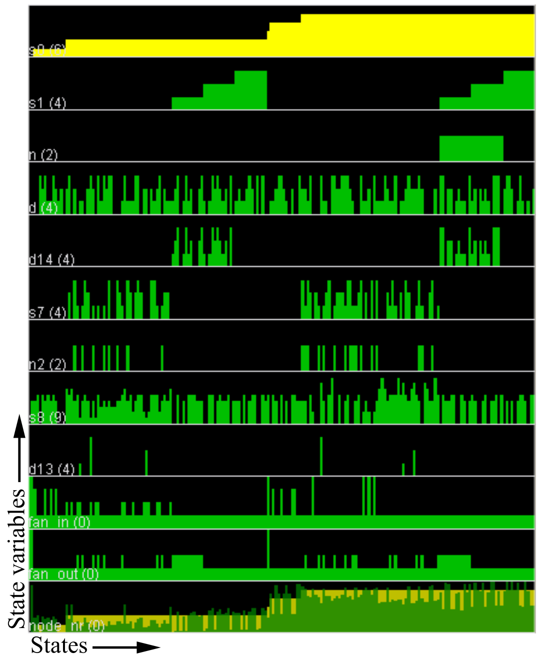

Figure 1. Parallel histograms.

Parallel histograms resemble the approach taken with *table lens* [7]. However, we have extended it with a feature that makes it particularly easy to visually compare different dimensions. When a specific dimension is selected it may be dragged over another, resulting in an overlay of the two (consider the lower part of Figure 1). Whereas most automated techniques perform *similarity analysis*, such a visual comparison also makes it intuitively clear how dimensions differ from each other and enables the user to detect interesting behavior such as alternating patterns.

The states may also be sorted based on a selected state variable; states are reordered along the x-axis according to an ascending or descending sorting of the values that this variable assumes in the different states. All histograms are updated to reflect this ordering. This further enables the user to detect interesting correlations or differences in a fashion analogous to *permutation matrices* [3]. When subsequent dimensions are selected and sorted, existing partitions are maintained; states are reordered only in intervals where the values of previously sorted variables are equal. This allows for the detection of patterns within patterns as illustrated in Figure 1 (consider the upper three histograms).

The transition system corresponding to Figure 1 was generated from a description of the Alternating Bit Protocol (ABP), a communication protocol for data transmission over an unreliable channel. The transition system has 191 states. The upper three histograms correspond to the operational modes of the sender, of the forward transmission channel and the value of the alternating bit for this channel.

Initially we were baffled by the fact that the alternating bit assumes a value of 1 for a relatively few number of states (third histogram from the top). However, after careful visual analysis we were able to deduce that the states where the top variable assumes the values of 2 and 5 correspond to an alternating bit value of true and false at the sender. We were then able to infer that only in such states is the transmission channel able to assume a value other than its default (0). This explains the two step-like shapes in the second histogram for states where the top histogram assumes the value 2 or 5. We were also able to deduce that the value of the alternating bit is only stored by the transmission channel during two of its modes of operation (corresponding to the values of 1 and 2 in the second histogram). Since no distinction is made between a default value and an alternating bit with a negative value (both are represented by the value 0), this explains the fact that the alternating bit is positive for so few states: most of the time it simply assumes its default value.

The above example shows that it is relatively easy to identify correlations that span more than two dimensions with parallel histograms. The ease of interpreting parallel histograms may be attributed to the fact that every histogram is pre-attentively perceived as a single shape or symbol [12]. Due to the fact that position and shape are separable visual cues, it may be argued that is easy to visually compare different histograms.

4. Projecting to 2D

The analysis and consequent deselection of dimensions with the parallel histogram mechanism leads to a lower-dimensional transition system that must be mapped to 2D. Suppose that there are p dimensions to consider (where typically $p < n$). We now view the transition system as a graph, but approach the positioning of states from a multi-dimensional perspective by considering this reduced set.

We consider any state $s = [s_1, ..., s_p]$ as a linear combination

$$s = \sum_{i=1}^{p} s_i \cdot e_i$$

where e_i is the i^{th} basis vector and s_i is the valuation of the i^{th} state variable. To find the projection s' of s on the 2D plane it may be defined as

$$s' = \sum_{i=1}^{p} s_i \cdot e'_i$$

where e'_i is some projection of e_i in 2D. Hence, the challenge is to find suitable e'_i for $i = 1, ..., p$ such that the questions identified in section 2 may be answered. Finding adequate e'_i turns out to be quite challenging and a number of strategies for choosing them are discussed below.

4.1. Uniform distribution

When projecting multidimensional data into 2D the e'_i may be selected such that they are distributed uniformly in the plane. This is similar to the approach taken with *star coordinates* [6]. Uniform distribution allows the user to determine the overall "spread" of the values that state variables assume. In particular, if the distribution occurs in the order that the corresponding histograms are arranged, this offers an interesting way to analyze the influence of different dimensions by reordering and deselection.

Figure 2 illustrates the results of a uniform distribution of nine dimensions of the ABP example. Transitions are rendered as curved line segments. Direction is encoded in the orientation of these arcs which should be interpreted in a clockwise fashion. Kandogan [6] argues that uniform distribution leads to the discovery of clusters of objects that share characteristics . However, since there is no guarantee that distinct p-dimensional points map to distinct positions in 2D, it is quite difficult to interpret results. Moreover, due to the fact that p dimensions are forced into 2D, it is not possible to attach precise semantics to a particular position.

4.2. Manual distribution

One line of attack that addresses the interpretation of the mapping to 2D is to enable users to manually alter the e'_i. In this way a clearer understanding of the influence of particular state variables may be deduced by considering the changes in the mapping as the e'_i are being adjusted. This does not prevent several states from being mapped to the same position, however.

An interesting observation is that users often resort to comparing two dimensions at a time by arranging the e'_i orthogonally in a fashion that resembles a 2D scatterplot. For instance, through manual interaction, figure 2 was reduced to figure 3 (by deselecting all but two state variables and positioning the e'_i orthogonally). In this figure we were able to identify a number of duplicate states that are mapped to the center-most node. By considering the transitions entering and leaving this node, we were able to determine that it

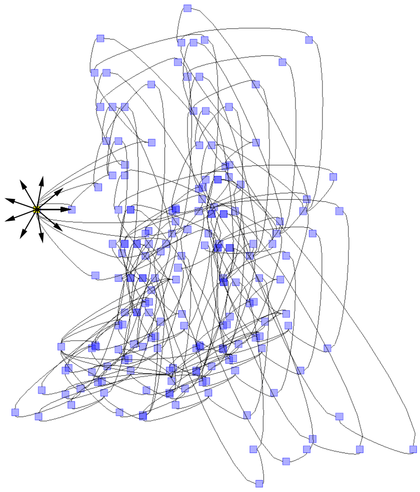

Figure 2. Uniform distribution.

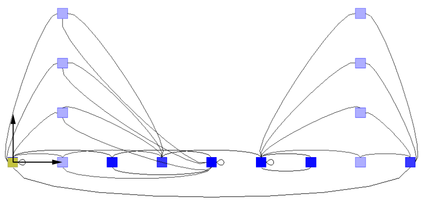

Figure 3. Manual distribution.

completely mimics the behavior of the initial state (on the far left). The reason for this is that the program that extracts the transition system from the original specification duplicates certain states.

4.3. Hypergrids

A hypergrid generalizes the notion of a 2D scatterplot to n dimensions. *Scatterplot matrices* also do this, but it is difficult to detect relationships that span more than two dimensions [2]. In essence, a hypergrid consists of recursively nested 2D grids. At every coordinate of a higher level grid a lower level grid is nested and so forth.

To see how a hypergrid is constructed, first consider the case where the projection vectors e'_i are restricted to being parallel to the x- or y-axis. Now, $e'_i = (x_i, y_i)$ with

$$x_i = \begin{cases} 0 & \text{if } i > p/2 \\ \delta_x & \text{if } i = p/2 \\ C_{i+1} \cdot x_{i+1} & \text{if } i < p/2 \end{cases}$$

and

$$y_i = \begin{cases} 0 & \text{if } i < p/2+1 \\ \delta_y & \text{if } i = p/2+1 \\ C_{i-1} \cdot y_{i-1} & \text{if } i > p/2+1. \end{cases}$$

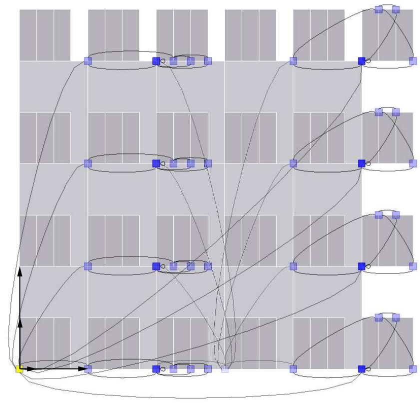

Figure 4. Orthogonal hypergrid.

where δ_x and δ_y are the minimum permissible horizontal and vertical intervals and C_i is the cardinality of dimension i.

With this method the coordinate system generated by e'_1 and e'_p takes into account the areas needed for nesting the coordinate system that e'_2 and e'_{p-1} generate at each of its coordinates, and so forth. This is achieved by considering the cardinalities, C_2 and C_{p-1}, of d_2 and d_{p-1} as well as the area needed for any deeper nested grids (by considering the cardinalities of their generating dimensions). Also, it ensures a minimum distance in the x- and y-axis between any two successive coordinates. Figure 4 illustrates an orthogonal hypergrid generated from four variables of the ABP example.

Hypergrids constructed in the above fashion are similar to the notion of *dimensional stacking* [11]: dimensions all contribute to displacing data items in directions parallel to the x- and y-axis. We have refined the idea further to more explicitly represent the influence of different dimensions by rotating successive grids slightly from the previous. It is argued that the *(dis-)similarity* in terms of orientation and the fact that position and orientation are *separable* visual cues make different dimensions easier to perceive [12]. To achieve this, let $e'_i = (x_i, y_i)$ with

$$x_i = \begin{cases} 0 & \text{if } i = p \\ s(i) \cdot \delta_x & \text{if } i = p-1 \\ s(i) \cdot C_{i+1} \cdot x_{i+1} & \text{if } i < p-1 \end{cases}$$

and

$$y_i = \begin{cases} 0 & \text{if } i = 1 \\ \delta_y & \text{if } i = 2 \\ C_{i-1} \cdot y_{i-1} & \text{if } i > 2 \end{cases}$$

where

$$s(i) = \begin{cases} 1 & \text{if } i \le p/2 \\ -1 & \text{if } i > p/2. \end{cases}$$

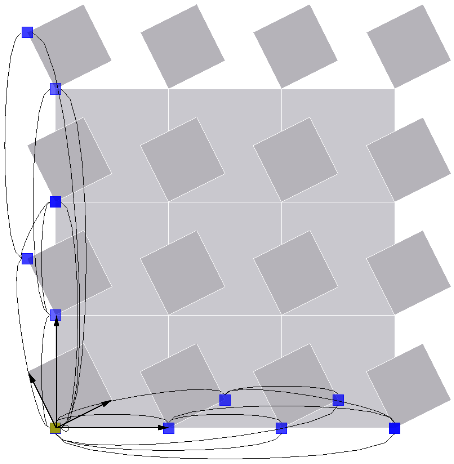

Figure 5. Rotated hypergrid.

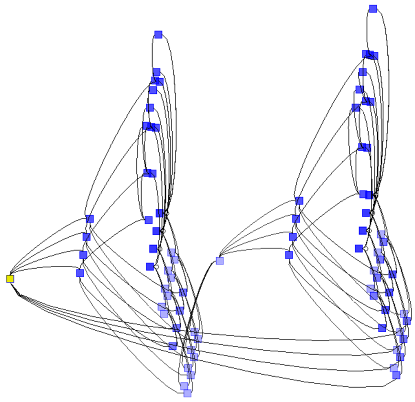

Figure 6. Principal component analysis.

The function $s()$ ensures that alternating pairs of projection vectors are "more orthogonal" in the sense that the e_i' with end points above the line $x = y$ are reflected about the y-axis.

In Figure 5 the larger grid plots two variables reflecting the modes of the two transmission channels in ABP. For the nested grids, the axes correspond to the value of the alternating bit for these channels. From this figure we were able to confirm that due to the sequential nature of ABP the two channels operate completely independently. For instance, it is easy to verify that no state exists in which both channels simultaneously assume a mode other than the default (0).

Figures 4 and 5 also show that the nested nature of hypergrids utilizes the well-known principle of *containment* [12], enabling the user to attach precise meaning to a particular position.

Due to the fact that state variables in the transition systems that we are considering generally have a small cardinality, the hypergrid approach may be applied here. In general, though, hypergrids do not scale that well as the cardinality or number of dimensions to consider increases. A further drawback is that hypergrids are quite expensive with regard to screen real estate; the area needed to ensure no overlap increases dramatically with every dimension that is included. However, when coupled with parallel histograms, where the order is reflected in the nesting of grids, hypergrids do offer an interesting way for analyzing relationships that span more than two dimensions.

4.4. Principal component analysis

Principal component analysis (PCA) is one of the oldest and most prominent techniques for multivariate data analysis [5]. PCA reduces the dimensionality of a data set while retaining as much variation as possible. In light of the current discussion, the aim is to map p dimensions to 2D by identifying the first two principal components. This entails finding the two largest eigenvalues of the covariance matrix of the dimensions and considering the two corresponding eigenvectors. More concretely, $e_i' = (x_i, y_i)$ with $x_i = c_{1,i}$ and $y_i = c_{2,i}$ where $c_1 = [c_{1,1}, ..., c_{1,p}]$ and $c_2 = [c_{2,1}, ..., c_{2,p}]$ are the eigenvectors mentioned earlier.

Figure 6 illustrates the result of performing PCA on eight dimensions from the ABP transition system and calculating the e_i' as outlined above. PCA offers a way to identify those dimensions that have the most dominant influence on the variation of the data set. The cardinalities of state variables influence the amount of variation, however. This may be addressed by normalizing every dimension before performing PCA. It may be argued that a state variable with a large degree of variation is not necessarily interesting. Also, as with uniform and manual distribution, no meaning can be attached to a position when $p > 2$. Nonetheless, from Figure 6 the two-phased behavior of ABP is clearly discernable.

5. Conclusion

We have introduced a number of methods to visualize transition systems by considering them as n-dimensional graphs. Table 1 summarizes how these techniques assist us in answering the questions identified in section 2.

Although simple, parallel histograms prove to be a useful aid for addressing many of our questions. Out of all techniques considered it provides us with the most sufficient overview of state variables. For instance, interesting variables can be detected by comparing the values that they assume. Besides providing insight into system behavior for subsets of state variables, we also find this mechanism to be an intuitive steering device for 2D projections. Often this

Table 1. How the different visualization approaches meet the challenges set out in section 2.

	Histograms	Uniform	Manual	Hypergrids	PCA
Question 1	Yes	No	No*	No	No**
Question 2	Yes	No***	No***	Yes	No
Question 3	Yes	No***	No*	Yes	No
Question 4	No	No	No	No	No
Question 5	Yes	No***	No***	Yes	No

*The movement of e'_i may be useful to some extent. **Unless variation is considered interesting, also for identifying phases of behavior. ***If $p > 2$.

compensates for the fact that, when considered in isolation, some projection methods do not scale that well beyond two dimensions. Moreover, the ability to quickly select, deselect and order dimensions allows us to identify correlations spanning multiple dimensions.

It is often important to determine whether states containing specific valuations of a subset of state variables exist. This can be verified with parallel histograms, but we also find hypergrids useful. Due to the nested nature of the axes, we are able to locate positions where such states are or would have been projected to. This is because hypergrids represent a high-dimensional space such that coordinates in it are still geometrically interpretable in 2D. It is also this characteristic of hypergrids that allows us to analyze multiple dimensions in a single 2D projection and to accurately associate meaning with particular positions.

As for the other projection techniques, on occasion we find manual interaction with the projection vectors useful for gaining insight into system behavior in terms of selected variables. Although 2D projections based on PCA do not provide us with precise answers to our questions, they do often give us a better idea of different phases in system behavior. We also note that the use of curved arcs seems to work well for encoding the direction of transitions.

Unfortunately none of the methods considered allow us to identify interesting correlations between state variables and specific actions (or transition labels). It is certainly true that significant behavioral characteristics of a transition system are captured in its transitions and actions. Therefore, we feel that this aspect requires closer attention and identify it as a candidate for future work.

6. Acknowledgements

This work was supported by the Dutch Foundation for Scientific Research *(NWO)* under grant 612.065.410. We also wish to thank Jan Friso Groote, Aad Mathijssen and Muck van Weerdenburg for their input and suggestions.

References

[1] D. Asimov, "The grand tour: a tool for viewing multidimensional data," *SAIM J. Scientific and Statistical Computing,* vol.6, no. 1, 1985, pp. 128-143.

[2] R.A. Becker, W.S. Cleveland and A.R. Wilks, "Dynamic Graphics for Data Analysis," *Statistical Science,* vol. 2, no. 4, 1987, pp. 355-395.

[3] J. Bertin, *Graphics and Graphic Information-Processing,* De Gruyter, 1981.

[4] S.C.C. Blom, W.J. Fokkink, J.F. Groote, I. van Langevelde, B. Lisser and J.C. van de Pol, "μCRL: a toolset for analysing algebraic specifications," LNCS 2102, 2001, pp. 250-254.

[5] I.T. Jolliffe, *Principal Component Analysis,* Springer-Verlag, 1986.

[6] E. Kandogan. "Visualizing multi-dimensional clusters, trends, and outliers using star coordinates," *Proc. 7th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining,* San Francisco, CA, 2001, pp. 107-116.

[7] R. Rao and S.K. Card, "The table lens: merging graphical and symbolic representations in an interactive focus+context visualization for tabular information," *Proc. 1994 ACM Conf. Human Factors in Computing Systems (CHI '94),* New York, NY, 1994, pp. 318-322, 481-482.

[8] J. Seo and B. Shneiderman, "A rank-by-feature framework for unsupervised multidimensional data exploration using low dimensional projections," *Proc. 2004 IEEE Symp. Information Visualization (InfoVis 2004),* Austin, TX, 2004, pp. 65-72.

[9] B. Shneiderman, "The eyes have it: a task by data type taxonomy for information visualizations," *Proc. 1996 IEEE Conf. Visual Languages,* Boulder, CO, 1996, pp. 336-343.

[10] F. van Ham, H. van de Wetering and J.J. van Wijk, "Interactive visualization of state transition systems," *IEEE Trans. Visualization and Computer Graphics,* vol. 8, no. 3, 2002, pp. 319-329.

[11] M.O. Ward, "XmdvTool: integrating multiple methods for visualizing multivariate data," *Proc. 1994 IEEE Conf. Visualization (Visualization '94),* Washington, DC, 1994, pp. 326-333.

[12] C. Ware, *Information Visualization: Perception for Design,* Morgan Kaufmann, 2000.

[13] C. Ware, "Designing with a $2\frac{1}{2}$D attitude," *Information Design J.,* vol. 10, no. 3, 2001, pp. 255-262.

[14] J. Yang, M.O. Ward, E.A. Rundensteiner and S. Huang, "Visual hierarchical dimension reduction for exploration of high dimensional datasets" *Proc. 2003 Symp. Data Visualization,* Grenoble, 2003, pp. 19-28.