

Tech-note: (NetLand) A physical network interface for visualization of abstract data relationship in VR

Jörg Frohnmayer¹, Roland Blach², Günter Wenzel²

¹Merz Akademie, Hochschule für Gestaltung, Teckstr. 58, 70190 Stuttgart, Germany

²CC Virtual Environments, Fraunhofer IAO, Nobelstr. 12, 70569 Stuttgart, Germany

ABSTRACT

We present the application “NetLand”, a novel interface to build and explore networks in immersive virtual environments (VEs). The NetLand prototype uses a physical model for the representation of hierarchical data structures and their interdependencies as well as their relational attributes. The application uses the ODE [1] physics engine integrated in the virtual reality (VR) environment Lightning [4] to simulate physical behavior, giving the user additional information and feedback about the displayed data. The target hardware platforms are VR desktop systems.

CR Categories and Subject Descriptors: I.3.6 [Computer Graphics]: Methodology and Techniques - Interaction Techniques; I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism - Virtual reality; I.3.8 [Computer Graphics]: Applications.

Additional Keywords: virtual worlds, virtual environments, representation, physics engine.

1 INTRODUCTION

We present our current work on the development of the user interface NetLand. We have an increasing need to visualize networks which represent abstract data relationships such as stock markets. This need leads to the development of a new approach for the mapping of abstract data categories. NetLand uses a physical model to describe the dynamic behavior of data in a network. Therefore we take data attributes to map them onto specific physical characteristics. This allows for a dynamic spatial configuration of the network and also permits a differentiated multi-dimensional search on attributes.

We implemented this information visualization prototype within an immersive VR environment, because networks can be represented as 3D-graphs without overlapping. Because of the 3D-graph representation we can additionally benefit from spatial interaction.

Two major research tasks can be identified. Firstly we have to define and develop an adequate mapping of data on physical characteristics. Secondly we implement the generation of net structures in a real-time VE. The NetLand interface focuses on the consistent combination of these two concepts.

The NetLand interface has four key elements.

- Visualization of nodes
- Building network structures
- Movement of nodes
- Exploration of the data

To map data attributes we have focused on different physical characteristics.

- Mass
- Spring forces
- Force fields

Based on Blach et al. [3] we derived a mapping of these characteristics which leads to the following initial assumptions:

- Mass can be used to represent the importance or flexibility of different objects
- Spring forces include a naturally understood way to visualize dependencies of objects by their spatial distance
- Force fields provide a spatial search function that can be applied to several existing categories
- Viscosity can be used to stabilize all objects in a zero gravity space

The value of this interactive visualization is that we are able to draw conclusions from the objects behavior, e.g. if we grab a node we can see the mass of the other connected nodes because of the swiftness of their reaction. Such physical simulations provide useful metaphors for the display of structural relationship in the dataset. Therefore even the time that one node takes to follow contains information for the user.

With these ideas we aim for a specific setup testing environment. The VE is a VR desktop environment with a keyboard and 3D input devices one for the head and one for the hand that is tracked by an optical tracking system. The integration of the prototype is made with the Lightning 2.5 [4] environment where an ODE [1] physics engine is integrated with C++.

The use case is a market analysis. We have used entities like companies and institutions that constitute the European VR market. A collection of abstract institution-company relationships are represented through a network structure (see figure 1). These relationships are represented by links and nodes that have a physical behavior. The nodes have physical attributes, e.g. mass, and stand for entities. The interrelation between different nodes and their links visualize dependencies between different companies in the market. The links represent special categorized relationships like cash flow, cooperation, transfer of people etc. and have the physical behavior of spring forces with a damping factor and a neutral distance.

The user is able to set up his own network as well as load an existing network. She is able to manipulate the nodes with three dimensional interactions and see the result in a physical simulation as a feedback of her input. With our NetLand prototype we especially point out the mapping of physical attributes of ties between datasets. Due to the scenario's abstract nature the physical behavior is not predefined by imitating the real world's behavior but is rather used to visualize the relationship of abstract datasets. The relationship relies on a semantic mapping of abstract data attributes onto physical characteristics.

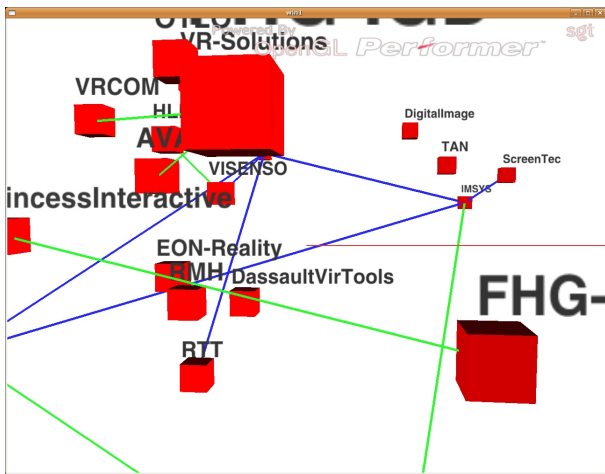


Figure 1. NetLand network

We use the physics for two reasons. First we want to use the active connection of user and physical behavior. This behavior must be forced by the user's will. With this Tech-note we concentrate on building an interface for these deliberately performed manipulations. Second we want to use the physical response of changes within NetLand to be observed by the user. These changes which affect the spatial configuration of NetLand evolve over time and are interesting information carriers.

2 RELATED WORK

Physical models nowadays become quite common in VEs; one fundamental reason for this is the missing realism in virtual objects behavior. In 3D-Games, this more satisfying object behavior improves, increasing the attention for such simulated physical behavior, and thus the degree of realism.

Physical concepts have rarely been used to represent abstract data relationships [3] like e.g. dependencies of companies on the market. Although the physical behavior could support abstract data relations with physical behavior qualities like spring forces, gravity and mass. These qualities could provide a smooth workflow with a strong immediacy effect for the examiner. But these positive effects come with an increase of complexity which means that the VR environment must have a look to its computational resources because of the performance restriction in real-time environments.

The NetLand physical engine is close to the spring forces used by Fruchterman [12], but his focus is in finding a mathematical algorithm to achieve a setting of fixed graph constellations in his simulation. The integration of the ODE [1] was very similar to the implementation described by Fröhlich [7], but his main goal was to achieve a physical multi-hand or multi-user interaction with objects. Koutek [8] has experimented with one handed physical operations, his so called "spring-fork", to achieve a more consistent physical manipulation of virtual objects. However we use the physics to simulate relationship effects among objects in order to achieve a consistent feedback of the analogies in abstract datasets.

The closest approaches to our prototype are presented by Soldati [11] and Osawa [6]. However Soldati uses spring forces to position a photo archive inside a sphere. To search the archive he uses the spring connection for the positioning of the billboards inside the sphere while the springs are connected from every specific search factor that is mapped on to the spheres surface to every single billboard. Thus he doesn't use the springs to map information relationships in between the different photos. Osawa

uses an interesting approach to generate multi foci maps through a thermo dynamical model. His approach of network structures is very similar to ours because of his node-link graphs. However he doesn't use the physics to reveal abstract data relationship. He used one or more heat sources to set one or more clear foci in a graph and generated out of these foci negative foci as well, in a physically consistent way.

3 NETLAND OVERVIEW

The NetLand interaction techniques are especially designed for VR desktop systems like the Picasso [2] system where one can work with common desktop devices like keyboard and mouse and additionally with tracked 6DOF input device (see figure 2). Therefore we use both keyboard and three dimensional input for user manipulation of the scene. The keyboard is used because it still is the most efficient way to enter textual information. We have two main base functions, adding nodes and connecting different nodes with links. Each node has a name and each link has a specific meaning. These semantics enable the examiner later on to reconfigure the whole network by switching different link types active or inactive. The reconfiguration can therefore be used to watch the influence of different factors on the network.



Figure 2. VR desktop system setup (Picasso at Fraunhofer IAO)

Additionally to these functionalities of creation, the user is able to select one node or a group of nodes to directly manipulate them in their position or delete them. While the user is selecting only a single node he is able to select a link from this specific node with a gesture.

After these functionalities the user is able to add additional information to nodes. This information describes additional properties of the node, which are not necessary used for the physical simulation.

4 NETLAND DETAILED DESCRIPTION

The main functionalities in our application to build networks are:

- creating and defining nodes
- creating links with a specific semantic meaning
- connecting different nodes

Because every node has a label that should be visible from every perspective these node labels are implemented as billboards [13]. The nodes with their names should be flexible in their repositioning in order to rapidly build logical groups. After we had accomplished these main functions of the network interface,

we started to embed, load, save and delete functions to the program which enabled us to reuse good maps and setups.

We started to implement physical behavior to reveal abstract dataset coherences. They do not aim to mimic realistic real world behavior. Therefore we implemented bodies, spring forces and force fields to reuse their physical attributes to emphasize abstract dataset relationships. We were combining abstract data models with the physical properties of the network. In our interface we use a set of springs which represent abstract data relationships and connect nodes with these springs to use the simulation in a zero gravity viscous environment as a visual feedback of the abstract data model.

One main strategy to get a better understandable interface was to provide an accurate visual response [5] for every change in the system that was caused by a user input. This feature was not only added because the end-user will not be a VR expert. It also guarantees that the interface is a dialogue created by user input and interface feedback. Thus the user has a better understanding of the performed computer action. Every time the user generates an input, the program has to generate an adequate visual feedback which directly enables the user to understand the performed computer operation.

4.1 Physical Behavior

For our physical simulations we have integrated the ODE [1] into the Lightning 2.5 environment. The NetLand prototype maps diverse abstract data relationships by visualizing these network models in a physically consistent way. The main difficulty is how to map these abstract factors on physical attributes and especially how to rank them. Our strategy was to use physical characteristics to classify these attributes into a new semantic system. This new semantic system is used to enable differentiated views upon the used dataset.

We have already experimented with the following mappings:

- mass: defines the size of an organization
- mass: defines the autonomy from other organizations
- viscosity: defines the inertia factor in the scene
- force field: provides a spatial search function which can be mapped on different node characteristics
- spring forces: define the relationship of nodes and must get categorized e.g.
 - o cooperates with
 - o cash flow
 - o spin of from
 - o transfer of personnel
 - o consults

The main importance in the mapping of spring forces is that there are two categories of springs appearing. First there is the spring type that stands for relationships where partner A and partner B are equal. These categories have an effect in both directions. Second there is the relationship where partner A is unequal from partner B. These kinds of relationships are categories that have an effect in one direction.

4.2 Interaction Principles

For user interaction a keyboard is used to enable the user to add his information expeditiously to the scene. For spatial interaction the NetLand prototype uses a video based tracking system. This allows the tracking of different reference points like the users head and his input device. The input device is a 6DOF body that is used as a pointing device to fly through the scene or to select and manipulate objects in the scene. Because the prototype is designed for a VR desktop most user manipulations are performed with the 3D pointing device and the keyboard.

4.2.1 Navigation

The examiner has two possibilities to move in the virtual data through the network. The first allows the user to explore the world by a classical point and fly technique [9] to get an impression from every desired perspective. This is a well known technique in VEs to explore the data. The second navigation technique is interest-orientated. If the examiner is interested in one specific node he can point on this node and press a button to start an automatic camera flight to the node's coordinates. This technique is known from Inventor/VRML browsers like Cortona [14]. After the view point is set to the new destination, the user is able to examine the node by rotating around the node and zooming close to or far away from it. The node bound view enables the user to jump rapidly from one node to another. This technique enables the user to switch the focus rapidly in the 3D environment from one cluster of the network to another. The change in focus is seamless because in 3D environments near objects automatically get more attention and therefore a positive focus. The user is also able to follow one specific node in space while the network reconfigures its constellation. This problem of changing focus and context is well known from information visualization [10].

4.2.2 Manipulation

With the spatial interaction device and the keyboard the user can perform different actions. At first the user is able to place nodes to the position of his input device and type a name for the placed node. Now after the first node is placed in the scene one is able to grab the node with a ray coming out of the input device. At the moment the ray intersects the node the node's font extends its size and the node gets highlighted by a transparent bounding box to give a visual response to the user that one has hit the node (see figure 3). The font size extends to ensure that also distant text is readable when it is highlighted and the transparent box appears to mark the selected node.

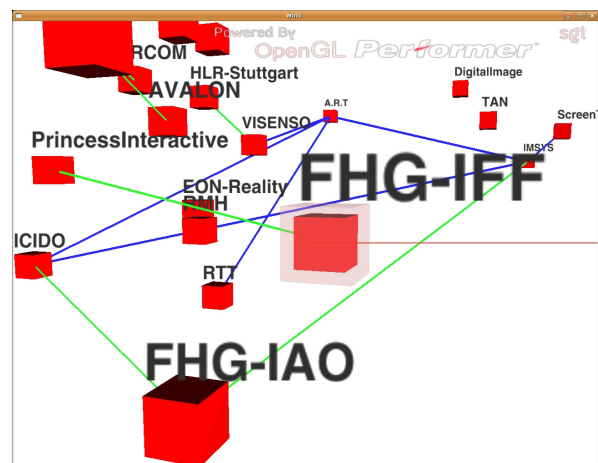


Figure 3. Node highlight identified by bigger font size and bounding box

When the node is hit it can be grabbed by pressing a button. The grab moves the node around the hand while the hand points to the desired destination. Simultaneously the object can be moved forward or backward in ray direction to assure its final destination. This grab mixes two movements, the angular movement of the node and the movement in the ray's pointing direction. An interesting property of this interaction technique is that the first angular movement is an isomorphic manipulation categorized by Bowman [9] and the movement in the ray direction

is a non-isomorphic manipulation. This mixture of the two movements has one main benefit; the user can reposition nodes quickly without any need to re-grab the node to move it into or out of the scene. To ensure that this specific node moves to the desired destination and is not affected by the physical simulation, the forces that affect this node are excluded from the simulation as long as the node is grabbed. But this node still has an effect on the other nodes connected by the spring forces and is also able to push other nodes out of the way.

After creating several nodes, the user is able to link them. To link two nodes she has to point at the first node and press a button. Then a line appears, giving the user a visual feedback that she has selected the first node (see figure 4.). This interaction technique is known from 2D-diagram tools like MS-Visio [15]. Then the user has to connect the second end of the line to another node by pressing the same button. Now one is able to categorize the link type by writing a category with the keyboard. These categories can be used later to visualize different relationships inside the network just by switching different categories on or off. In combination with the physical spring force the changing between different link categories automatically reconfigures the structure of the whole network.

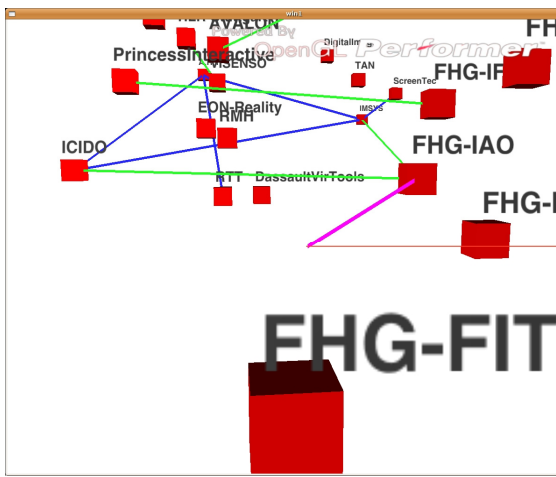


Figure 4. Link feedback

Links can be selected by pointing to one of the connected nodes. Afterwards the examiner has to press a button and perform a gesture in the pointing direction of the desired link to select the right one. This is a simple but efficient method to select links in 3D tracked network structures directly from the nodes.

Another technique that has an effect on nodes is the use of the force field. The examiner is able to activate a force field and add a category with the keyboard to it. Now the force field affects every node of this category within a specific range. This technique enables the user to ensure that an individual set of objects stays in the range of interaction and to maintain it in his focus.

It occurs that these inputs generate indirect actions that are handled by the physical simulation as a response to the manipulations. We seek to get additional information from the dynamic data behavior. We think that this information should provide knowledge beyond the inner data relationship even in other datasets.

5 CONCLUSION AND FUTURE WORK

We have implemented a working prototype for testing different mappings of abstract categories on physical attributes. In physical simulations stability in case of multi body simulations is a known

problem which has to be solved with adequate parameterization of the dynamic system. In our case of abstract data we are more free to choose the associated physical properties than in realistic physical simulations because the dynamics only have to create a meaningful and stable behavior. We achieved a frame rate of 21 fps with 1000 boxes, lines, physical bodies and spring forces computed by an Intel Pentium 4 with 1GB RAM, a NVIDIA 6600 GT 128MB graphics card and a window size of 640x480. With this prototype one is able to create networks and implement different data types. In first stage of the prototype the user is able to interact with all data in the scene and the system provides necessary direct manipulations to work in a dynamic environment.

For the future the main focus lies on building a framework to embed additional data into the network. This framework should be able to display all datasets and simultaneously allow the user to modify all of the parameters. This real-time access will control the actual data behavior inside the scene. We also want to test the prototype with other datasets and carry out user studies.

REFERENCES

- [1] ODE Library, <http://www.ode.org>
- [2] Hoffmann, H., Stefani O., Harshada P., 2006: Extending the Workplace by a Portable Virtual Reality System; Special Issue of the International Journal on Human Computer Studies, Volume 64, Issue 3, Elsevier.
- [3] Blach R., Wenzel G., Dangelmaier M., Frohnmayer J., «Towards a Physical Based Interaction-Model for Information Visualization», In Proceedings of the HCI International, 2007, p.750 ff.
- [4] Blach, R., J. M. Landauer, A. Simon, A. Rösch, «A Flexible prototyping Tool for 3D Real-Time User-Interaction.», In Proceedings of the Eurographics Workshop, 1998, pp. 195-203.
- [5] Myron W. Krueger: «Responsive Environments», in AFIPS 46 National Computer Conference Proceedings, 1977, pp. 423-433.
- [6] Osawa, N.: A multiple-focus browser for network structures using heat models, The 59th National Convention of Information Processing Society of Japan, 5K-09 (1999).
- [7] Fröhlich, B., Tramberend, H., Beers, A., Baraff, D.: Physical-Based Manipulation on the Responsive Workbench. In: Proceedings IEEE Virtual Reality 2000, pp. 5–11 (2000).
- [8] Koutek, M., van Hees, J., Post, F.H., Bakker, A.F.: Virtual Spring Manipulators for Particle Steering in Molecular Dynamics on the Responsive Workbench. In: Proc. Eurographics Virtual Environments 2002, Barcelona, pp. 53–62 (2002).
- [9] Bowman, D.A, Kruijff, E., Laviola, J.J., Poupyrev, I.: 3D User Interfaces Theory and Practice. Addison-Wesley, Boston (2005).
- [10] Spence, R.: Information Visualization. Addison-Wesley, Harlow (2001).
- [11] Soldati, M., Doulis, M., Csillaghy, A.: Sphere Viz. In: 11th International Conference Information Visualization, Zürich 2007, pp. 680-683.
- [12] Fruchterman, T., Reingold, E.: Graph Drawing by Force-directed Placement. In: Software-Practice and Experience, Vol. 21. November 1991.
- [13] Akinie-Möller, T., Haines, E.: Real-Time Rendering, A K Peters Ltd., Massachusetts, 2002.
- [14] Cortona, <http://www.parallelgraphics.com/products/cortona/>
- [15] MS Visio, <http://office.microsoft.com/de-de/visio/>