Beyond the Tunnels: Advanced 3D graphical modulation

Florent Berthaut

Beyond the Tunnels: Advanced 3D graphical modulation

Florent Berthaut*
Université de Bordeaux - LaBRI

ABSTRACT

Tunnels are 3D graphical sliders which modify the appearance of, or ”modulate”, one or several 3D objects simultaneously using these objects as cursors themselves. They rely on a strong visual feedback to provide multiple scales and parameters combinations which makes them appropriate for the use in immersive virtual musical instruments. In this paper we present applications of the *Tunnels* for collaborative modulation tasks, for avatars and weapons modifications in 3D video games and for the control of interactive scenarios. We also propose several improvements to the *Tunnels* such as a method for storing them and a two-dimensional version called *Grids*.

1 INTRODUCTION

Graphical sliders are commonly used in graphical user interfaces to control continuous parameters. Many research have been done to enhance these widgets in 2D interfaces. For example, Fast Sliders [16] provide quick access to sliders using marking menus and also some features that also exist in our system, such as the undo. In-Context sliders [17] are directly attached to objects, allowing users to easily manipulate elements in large sets. The same idea is developed with the Pie-Slider [14], which also allows for the manipulation of several parameters by placing the sliders around the object. Variable resolution of sliders is also explored in [1] and [2], providing user with the choice of granularity for the control of parameters.

In 3D virtual environments, different approaches exist. Some research explore the manipulation of 2D widgets inside the environment [7]. Other rely on 3D representations but with standard slider behaviour [15], [11]. 3D sliders are also embedded into other widgets such as the tBox [10] which allows users to manipulate position, orientation and scale with multi-touch gestures.

In a previous paper, we developed the idea of *Tunnels*, 3D graphical sliders which combine advantages of these solutions while providing new ones, such as the simultaneous modulation of several objects, multiple scales and an improved visual feedback. The purpose of these widgets is to allow users to efficiently modify the appearance, e.g. color, scale, transparency, density, of 3D graphical objects. They were first developed for musical interaction, by associating 3D graphical objects with sound synthesis processes and mapping graphical parameters to musical parameters. In this context, this type of modification is called a ”modulation gesture”, i.e. a gesture which modifies the sound parameters, as opposed to excitation gestures which produce the sound. We also use the term ”modulation” to describe modifications of the appearance of 3D objects in other contexts.

In this paper, we first recall the general approach behind the *Tunnels*. Then, we propose new uses of the *Tunnels* for collaborative modulations in building/artistic applications, modulation by 3D navigation for games and modulation paths for interactive scenarios. We also describe some useful improvements such as *Grids* for two-dimensional modulations, and storing of *Tunnels* with a ”brochette” metaphor.

*e-mail:berthaut@labri.fr

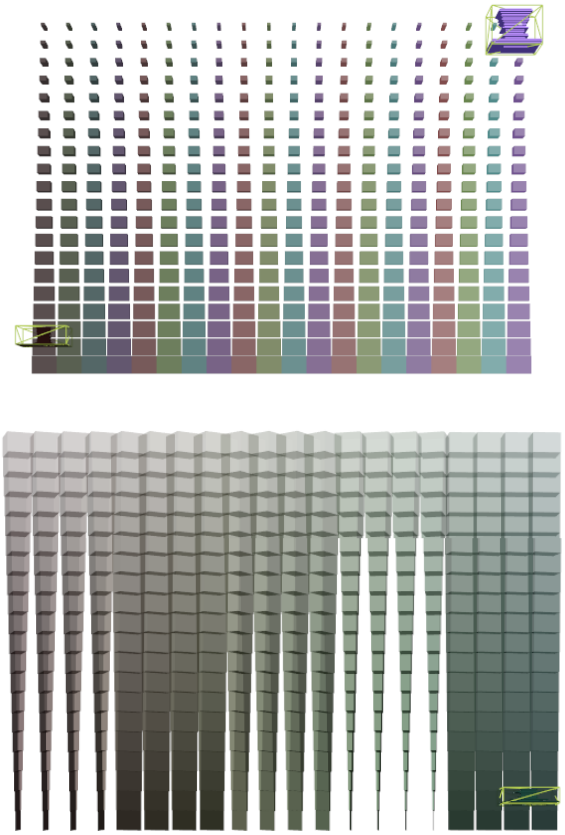

Figure 1: Two different *Grids*. On the top, continuous color scale on the X axis, continuous size (depth) and reverse continuous density (X/Y size) on the Y axis. On the bottom, rotation speed and color discrete scales on the X axis, and transparency and reverse size continuous scales on the Y axis.

2 THE TUNNELS

Tunnels are enhanced 3D graphical sliders. Unlike traditional sliders, they don’t have cursors. Instead, the modulated 3D objects act as cursors themselves. *Tunnels* are displayed as semi-transparent boxes or cylinders composed of series of boxes or cylinders whose appearance reflects the scale of values of the modified graphical parameter, as depicted on figure 2.

They were inspired by information visualization techniques developed by Healey [12], in which sets of objects represent multi-dimensional data using combinations of graphical parameters, e.g. color of objects represent one dimension, size another one , orientation yet another one and so on. We transposed this idea for the manipulation of graphical parameters.

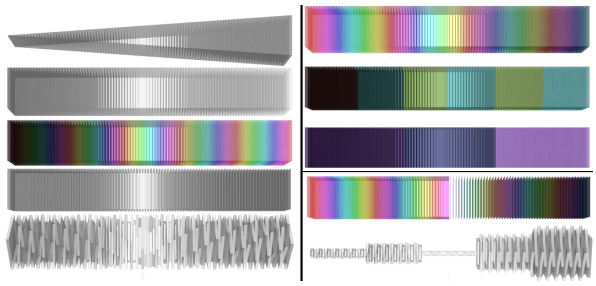

Figure 2: Left: from top to bottom : Size, Transparency, Color, Density and Rotation Speed *Tunnels*. Top Right: Three Color *Tunnels* with different scales (one continuous and two discrete). Bottom Right: Combination of parameters: Color/Density and Size/Rotation/Speed

Unlike other 3D sliders, they provide a visual feedback to facilitate the control with various scales and to avoid unintentional jumps in values. However, parameters representations must be adapted. For example, while the color parameter is obviously represented by setting the color of each component (in our case a mixed hue/luminance scale) , the size parameter uses only the height and depth of components, since width is already used to represent the density parameter, i.e. space between elements of composite objects.

When a 3D object is moved inside a *Tunnel* controlling one parameter, the value of this parameter for the object is modified accordingly, as seen on figure 2. For example, when moving an object inside a Size Tunnel, the object's size changes from one end to the other. The value of the parameter is computed by projecting the center of the 3D objects on the axis of the *Tunnel* to get the position ratio and then taking the scale into account. The modified parameter keeps the set value if the object crosses the *Tunnel* before exiting it, otherwise the previous value is reset. This allows users to undo their modification. *Tunnels* also allow for variable resolution since they can be stretched to give access to more values, as shown on figure 2.

Following the idea of the Local Tools [4], Tunnels are left in the environment and can be picked up, instead of being attached to a specific object or panel. They are also similar to the See-Through Tools described in [6], but the activation is done directly when a collision occurs, instead of having to use a bimanual gesture.

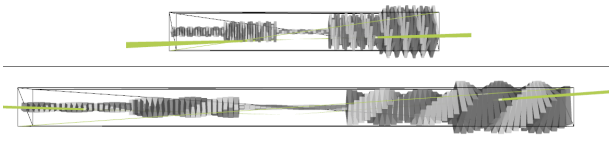

Figure 3: *Tunnels* can be resized by grabbing them with both hands.

Tunnels may also have different scales for each parameter as depicted on figure 2. Therefore, users may predefine "palettes" for the parameters with either continuous linear scales or discrete ones so that specific values can be easily reached. Scales can be changed on-the-fly. Several parameters can be combined on a single *Tunnel* to modify for example the size and transparency simultaneously as shown on figure 2. In a musical application, this results in one-to-many mappings [13] between gestures, i.e. displacement of 3D objects, and sound parameters, while preserving independent visualization of manipulated parameters.

In addition to modifying single objects, *Tunnels* allow for the manipulation of several 3D objects at the same time, by manipulating the *Tunnel* directly, as seen on figure 2. On the contrary to what can be done with traditional sliders here all objects can be set with one widget and one gesture either all to the same value or to different ones depending on the trajectory of the *Tunnel*.

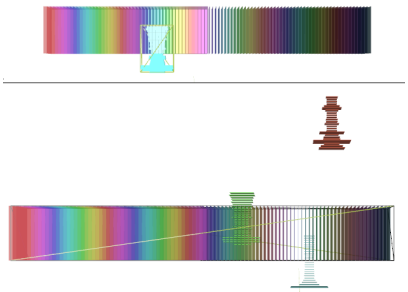

Figure 4: Two different use of *Tunnels*. Top: Grabbing a 3D object and passing it through the *Tunnel*. Bottom: Grabbing a *Tunnel* in order to modulate several objects simultaneously.

If we apply the instrumental interaction model designed by Beaudoin-Lafon [3], *Tunnels* are the virtual part of instruments which allow users to manipulate objects of interests, here 3D objects. The degree of indirection, i.e. spatial distance between the object and the instrument is very low because 3D objects are placed directly inside the *Tunnels*. Temporal offset is null since the modifications are applied immediately. The degree of integration is 3/N , 3 because we use a 3DOF input device and N being the number of parameters combined on the *Tunnel*. Finally, the degree of compatibility is very high because manipulations done by users, i.e. moving the objects and *Tunnels*, are reflected directly in the 3D virtual environment.

3 Beyond the Tunnels

This section describes contexts where *Tunnels* have proven to be useful, and also several needed improvements over the original approach.

3.1 Modulations for 3D video games

One interesting use of the *Tunnels* can be found in the context of video games and in particular first person shooters. These games heavily rely on expert navigation in 3D virtual environments. Indeed, players have to chase other players to shoot them, and navigate to specific targets such as bases or flags. They also pick up items which modify their avatar or weapons characteristics. However, these items often correspond to discrete predefined values.

In this context, *Tunnels* may allow players to efficiently fine-tune game parameters, by simply walking/running/jumping through them. For example instead of having a set of imposed option for lightness/strength of armors or speed/capacity of weapons, players may select these continuously using respectively density or rotation speed *Tunnels*.

Therefore, we implemented avatar-sized *Tunnels* in a musical first person shooters, which can be seen on figure 5. In this game, players can modify musical parameters associated to their avatar, which propagate to other players when shooting them.

3.2 Collaborative modulations

In collaborative virtual environments, several possibly distant users work together on various tasks. In particular, artistic collaboration may involve cooperative modification of 3D objects. These modifications sometimes have to be coordinated. For example, several

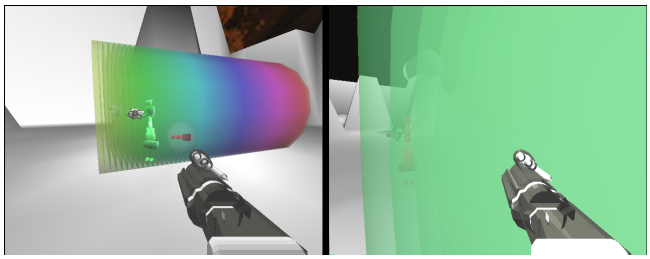

Figure 5: Split-screen first person shooter game. Right player is moving through an avatar-sized *Tunnel* to modify the characteristics of its avatar and weapon.

users may have to set the same color for their own objects. In an immersive virtual musical orchestra, a conductor may need to indicate to other musicians a specific modulation to do. However, it can be difficult to understand other musicians actions when they are represented as avatars, especially when using private tools which are not easily visible by other users, or even public tools with little visual feedback and concurrency issues. These issues are often tackled in research on collaborative virtual environments as described by [5].

In this context, *Tunnels* provide a way to coordinate users actions thanks to the visual feedback on parameters scales and because users objects are directly used as cursors, which prevents concurrency issues on tools. By placing a *Tunnel* in a vertical position at the centre of a scene, as depicted on figure 6, users may perceive others actions and use the same parameter scale for their modulation.

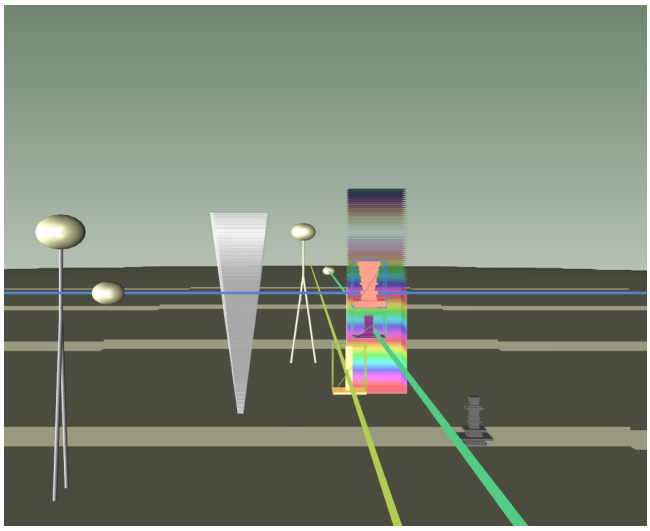

Figure 6: An immersive virtual musical orchestra with three musicians. We can see the avatars of the two other musicians. Each musician is controlling a virtual ray of a different color, and passing 3D objects through a shared Color *Tunnel* which modifies the pitch of associated sound processes.

3.3 Storing Tunnels

A common issue with *Tunnels* and other tools that remain in the environment (as opposed to tools which appear only when used) is that a large number of them may eventually clutter the 3D environment. Another problem is the transport of *Tunnels* when navigating the environment. For example, musicians may want to bring a *Tunnel* modifying a specific sound parameter from one set of 3D musical objects to another.

We propose a solution for easy storage and retrieval of *Tunnels*, which could be applied to other tools, using a "brochette" metaphor. One of the most common metaphor for 3D selection [8] is the Virtual Ray, which allow users to point at 3D objects by manipulating the position and orientation of a ray projected from their hand into the environment. Using a specific button or gesture, *Tunnels* are scaled-down and attached to the closest end of the virtual ray, as depicted on figure 7. They remain attached and visible while using the ray and navigating the environment. Users may then grab a stored *Tunnel* with the ray held by the other hand to release it and restore its initial size.

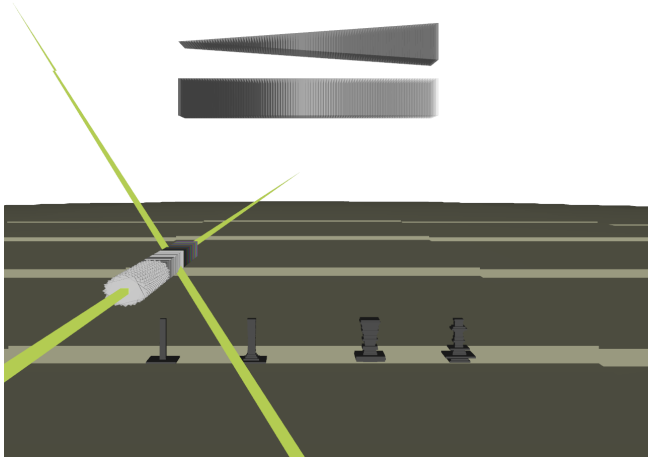

Figure 7: Brochette metaphor used to store *Tunnels*. Here three *Tunnels* have been attached to the left hand Virtual Ray using a special gesture. With the Ray controlled by his right hand, the user may select and release stored *Tunnels*.

3.4 Modulation paths

Current version of the *Tunnels* limit modulations of 3D objects to horizontal displacement. But more complex 3D movements may provide different meanings for modulations. For example, a 3D path with branches may represent a scenario for an artistic performance, with several possible chain of events, e.g. lights controls. In the field of music computing, some sequencers such as Iannix [9] also use multiple curves and complex paths to represent time lines.

Tunnels may be used to provide such a functionality with additional visual feedback, and a direct control of the cursor, which can be placed at any point of the path. Therefore, we define 3D modulation paths as sequences of oriented *Tunnels*, as depicted on figure 8. These paths can be used by grabbing and moving 3D objects acting as cursors through them. They may also be expanded to allow users to travel through them, as it is done with avatar-sized *Tunnels* described in section 3.1. Several parameters can be controlled and visualized within the paths, as *Tunnels* allow for parameters combinations. So in the case of the control of an artistic performance, one parameter may be associated with lights intensity while another would be associated to the movements of motorized objects or sound effects.

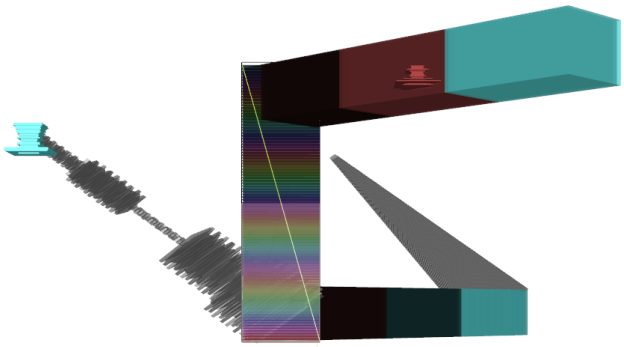

Figure 8: Modulation path made up of five *Tunnels*. Two 3D cursors are being manipulated at the same time.

3.5 Grids

One needed feature, especially for musical application, is the ability to modulate two parameters simultaneously but independently. For example, a common combination consists of the cut-off frequency and the resonance of an audio low-pass filter. 2D sliders are also often used in 2D graphical interfaces as color pickers.

In order to provide such possibility in 3D virtual environments, while preserving the visual feedback and parameters combinations of the *Tunnels* we them to two-dimensional *Grids*, which are 2D planes composed of small boxes, as shown on figure 1. Our current implementation is made of 20x20 boxes, but denser *Grids* would provide a more continuous visualization of parameters scales. However, parameters scales representations used in the *Tunnels* must be adapted. For example, the size parameter is represented using the depth of boxes while the density parameter is represented by modulating their scale on the vertical and horizontal axes. The value set to the 3D object is computed using the projections on the two axes of the *Grid*. Because we can manipulate 3D objects in three dimensions, activation, i.e. when an object begins to be modulated, can be done at any starting value using the axis perpendicular to the grid. This allows users to place 3D objects in the *Grid* in a way that the modulation starts at their previous parameter value. The undo feature of *Tunnels* may also be used. Indeed, the new value for a parameter is kept only if the modulated object crosses the *Grid*, i.e. exits behind it first, otherwise it takes back its previous value.

4 Conclusion

In this paper, we presented new uses and modifications of the *Tunnels*. These 3D graphical tools allow users to modify the appearance of one or several 3D objects simultaneously with a strong visual feedback and multiple scales. *Tunnels* have proven to be useful in various fields, such as continuous modulations for players characteristics in video games, modulation paths for the control of parameters according to scenarios or collaborative tools for objects modification tasks. We also added a storing method for *Tunnels* using a brochette metaphor. Finally, we extended our approach by defining 3D tools called *Grids* which allow for the simultaneous independent manipulation of two set of parameters, while preserving the advantages brought by *Tunnels*.

We already collected remarks from users about the *Tunnels* during demos in various events. However, a more formal evaluation should be conducted on specific tasks in order to compare the *Tunnels* to standard 3D sliders. In particular, we want to study the performance of the *Tunnels* compared to other conditions, e.g. one slider per object or one slider for selected objects, in a modulation task where subjects are asked to modulate multiple objects to match a model.

Another point that we want to explore is the extension of the set of controllable parameters. In fact, for now only five of them have been implemented. Each parameter has to be adapted so that it can be represented without disturbing the perception of other parameters. We might need to change the elements that compose the *Tunnels*. For example, shape distortion might need more complex shapes than boxes to be discernible. Combinations of parameters should also be chosen carefully. Indeed, modulation of texture may not be compatible with modulation of color or transparency. Therefore, we need to define rules for the creation of *Tunnels* so that their efficiency is maximized.

Acknowledgements

References

[1] C. Ahlberg and B. Shneiderman. The alphaslider: a compact and rapid selector. In *Proceedings of the SIGCHI conference on Human factors in computing systems: celebrating interdependence*, CHI '94, pages 365–371, New York, NY, USA, 1994. ACM.

[2] C. Appert and J.-D. Fekete. Orthozoom scroller: 1d multi-scale navigation. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, CHI '06, pages 21–30, New York, NY, USA, 2006. ACM.

[3] M. Beaudouin-Lafon. Instrumental interaction: an interaction model for designing post-wimp user interfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '00, pages 446–453, New York, NY, USA, 2000. ACM.

[4] B. B. Bederson, J. D. Hollan, A. Druin, J. Stewart, D. Rogers, and D. Proft. Local tools: an alternative to tool palettes. In *Proceedings of the 9th annual ACM symposium on User interface software and technology*, UIST '96, pages 169–170, New York, NY, USA, 1996. ACM.

[5] S. Benford, C. Greenhalgh, T. Rodden, and J. Pycock. Collaborative virtual environments. *Commun. ACM*, 44:79–85, July 2001.

[6] E. A. Bier, M. C. Stone, K. Fishkin, W. Buxton, and T. Baudel. A taxonomy of see-through tools. In *Proceedings of the SIGCHI conference on Human factors in computing systems: celebrating interdependence*, CHI '94, pages 358–364, New York, NY, USA, 1994. ACM.

[7] M. Billinghurst, S. Baldis, L. Matheson, and M. Philips. 3d palette: a virtual reality content creation tool. In *Proceedings of the ACM symposium on Virtual reality software and technology*, VRST '97, pages 155–156, New York, NY, USA, 1997. ACM.

[8] D. A. Bowman and L. F. Hodges. An evaluation of techniques for grabbing and manipulating remote objects in immersive virtual environments. In *SI3D '97: Proceedings of the 1997 symposium on Interactive 3D graphics*, pages 35–38, New York, NY, USA, 1997. ACM.

[9] T. Coduys and G. Ferry. Iannix - aesthetical/symbolic visualisations for hypermedia composition. In *Proceedings of the 1st Sound and Music Computing Conference (SMC04)*, 2004.

[10] A. Cohé, F. Dècle, and M. Hachet. tbox: a 3d transformation widget designed for touch-screens. In *Proceedings of the 2011 annual conference on Human factors in computing systems*, CHI '11, pages 3005–3008, New York, NY, USA, 2011. ACM.

[11] B. D. Conner, S. S. Snibbe, K. P. Herndon, D. C. Robbins, R. C. Zeleznik, and A. van Dam. Three-dimensional widgets. In *Proceedings of the 1992 symposium on Interactive 3D graphics*, I3D '92, pages 183–188, New York, NY, USA, 1992. ACM.

[12] C. G. Healey. Formalizing artistic techniques and scientific visualization for painted renditions of complex information spaces. In *In Proceedings International Joint Conference on Artificial Intelligence 2001*, pages 371–376, 2001.

[13] A. Hunt and R. Kirk. Mapping strategies for musical performance. *Trends in Gestural Control of Music*, pages 231–258, 2000.

[14] A. Kulik, A. Kunert, C. Lux, and B. Frhlich. The pie slider: Combining advantages of the real and the virtual space. In A. Butz, B. Fisher, M. Christie, A. Krger, P. Olivier, and R. Thern, editors, *Smart Graphics*, volume 5531 of *Lecture Notes in Computer Science*, pages 93–104. Springer Berlin / Heidelberg, 2009.

[15] R. W. Lindeman, J. L. Sibert, and J. N. Templeman. The effect of 3d widget representation and simulated surface constraints on interaction in virtual environments. In *In Proceedings of IEEE VR 2001*, pages 141–148. IEEE Press, 2001.

[16] M. Mcguffin, N. Burtnyk, G. Kurtenbach, and K. Street. Fast sliders: Integrating marking menus and the adjustment of continuous values. graphics interface. In *Proceedings of Graphics Interfaces*, pages 35–42, 2002.

[17] A. Webb and A. Kerne. The in-context slider: a fluid interface component for visualization and adjustment of values while authoring. In *Proceedings of the working conference on Advanced visual interfaces*, AVI '08, pages 91–99, New York, NY, USA, 2008. ACM.