# Star Interpolator – A Novel Visualization Paradigm for Graphical Interpolators

Darrell Gibson
Faculty of Science & Technology
Bournemouth University
Dorset, UK
dgibson@bournemouth.ac.uk

Richard Polfreman
Faculty of Arts and Humanities
University of Southampton
Hampshire, UK
r.polfreman@soton.ac.uk

## ABSTRACT

This paper presents a new visualization paradigm for graphical interpolation systems, known as Star Interpolation, that has been specifically created for sound design applications. Through the presented investigation of previous visualizations, it becomes apparent that the existing visuals in this class of system, generally relate to the interpolation model that determines the weightings of the presets and not the sonic output. The Star Interpolator looks to resolve this deficiency by providing visual cues that relate to the parameter space. Through comparative exploration it has been found this visualization provides a number of benefits over the previous systems. It is also shown that hybrid visualizations can be generated that combine the benefits of the new visualization with the existing interpolation models. These can then be accessed by using an Interactive Visualization (IV) approach. The results from our exploration of these visualizations are encouraging and they appear to be advantageous when using the interpolators for sound designs tasks.

## Author Keywords

Sound, synthesizer, interpolation, visualization, interface, sound design

## CCS Concepts

• **Human-centered computing → Visualization design and evaluation methods** • Human-centered computing → Usability testing • Human-centered computing → Empirical studies in HCI

## 1. INTRODUCTION

A challenge when designing sounds with a synthesizer is how to configure a large number of synthesizer parameters to create a certain audio output, i.e. how to translate sonic intent to parameter values. Although having direct access to every parameter (*one-to-one* mapping) gives fine control over the sound, it increases the complexity of the sound design process. As a result, practitioners often rely heavily on experience or the use of preset sounds and samples. An alternative approach is to map a smaller number of control parameters to a larger number of synthesizer parameters (*few-to-many* mapping) in order to simplify the process [1]. Graphical interpolation systems offer such a mechanism by providing a two-dimensional graphical

---

[1] Comparative examples of the sonic outputs generated with the different interpolation models can be accessed with the following two links: https://youtu.be/KiT2wXujrv4 https://youtu.be/E_l1XdX-E80

control pane where markers that represent "known" preset synthesizer sounds can be positioned. Interpolation can then be used to generate new parameter values in-between the specified locations by moving an interpolation cursor within the parameter space. Interpolating between presets of parameters can facilitate smooth sonic transitions and the discovery of new settings that blend the characteristics of two or more existing sounds. The sonic outputs are a function of the presets, their location within the interpolation space, the relative position of the interpolation point and the interpolation model used to calculate the influence of each preset [1]. The importance of mapping strategies has been previously recognized in the design of electronic musical instruments [1], [2], but other than recognizing the ability to find new sounds, little consideration has been given to a sound design context. Work by the authors aims to bridge this gap.

A variety of distinct graphical models have been used in the past for parameter interpolation [3 - 11], which present the user with different levels of visual feedback. A number of these have been previously re-implemented [12] by the authors and Figure 1 shows a range of the different visualizations. Through this earlier work it has been shown that each of these models provides a unique sonic pallet, even when populated with the same preset sounds at identical locations.[1] Following this work a wider evaluation of the visual characteristics of each interface has now been undertaken.
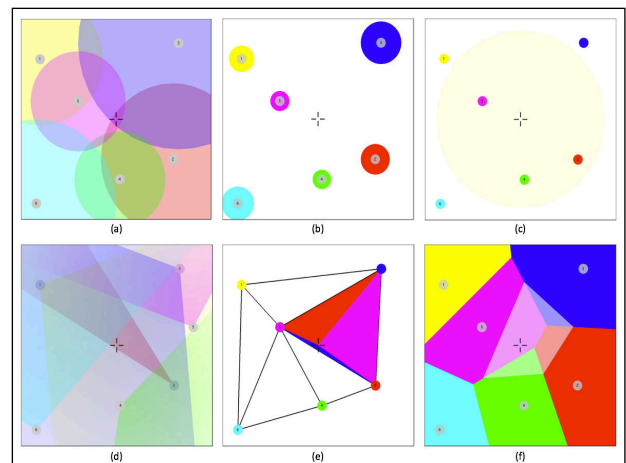


**Figure 1 Visualizations for Different Graphical Interpolator Models**

The first interpolator, shown in Figure 1a, is based on the *nodes* object in Max [11], where each preset is represented as a numbered circular node and the interpolation is performed where

the nodes intersect. In areas where there is no overlap between nodes the corresponding preset sound will be generated. Due to a transparency component in the visualization, it clearly shows the "regions-of-interest" where a new interpolated sound will be generated. The next interpolator (Figure 1b) is based on the INTERPOL control window for the SYTER system [4]. This uses a gravitational model where each preset is a *planet* the size of which determines its *gravitational force* and so influence in the interpolation space – larger planets have stronger influence than smaller planets. Placing the cursor on a planet then results in the corresponding preset sounding. Interpolation is performed in the free-space, based on the cursor distance to the planets and their gravitational force, resulting in new sonic outputs. The third interpolator (Figure 1c) uses an Inverse Weighted Distance (IWD) model [8], where just the distance to the preset locations is used. To further distinguish it from the gravitational model, when implemented it was enhanced by providing a cursor radius so that only presets within the visual circle will be included in the interpolation calculations. The interpolator shown in Figure 1d, uses a light model, where each preset corresponds to a lamp that emits a beam of light and the interpolation is performed where the beams intersect [5]. Each lamp has controls for the angle, aperture and extent of the beam so a directional aspect can be included for each influence. The light beams give a visual representation of the corresponding preset's "field-of-influence". The original lamp model also included an optional "background" preset, so that even when the cursor is only in one beam it would still be interpolated relative to the background. In addition, different coloured lamps represented different subsets of interpolated values, allowing a multi-layered approach, e.g. controlling different sonic features with each colour. Although these are unique features, in this realisation they were not included so as to allow only the visual model to be compared to the others, without the inclusion of additional features. The next interpolator (Figure 1e) uses the preset locations to form a Delaunay triangulation in the interpolation space [9]. The triangulation then determines which three presets are included in the interpolation by using the vertices of the containing triangle. The interpolation is then calculated using Barycentric coordinates [13], giving a weighting for the presets dependent on the cursors relative distance to each. In this implementation, the relative weightings of each preset are shown as a triangulation between the cursor point and the presets. The colours of this sub-triangulation correspond to the containing presets showing the relative proportions of each preset included in the interpolated sound. The final interpolator, shown in Figure 1f, uses *natural neighbour interpolation*. Here a Voronoi tessellation is constructed where each polygon represents a preset [6]. The interpolation is then calculated between the presets that are natural neighbours of the cursor location. The weightings are determined based on the area that would be "stolen" from each neighbouring preset, if a new polygon were inserted into the tessellation at the cursor position. This is shown as a transparent "ghost" polygon centred at the cursor so the relative proportion of each preset in the interpolation is displayed.

In Figure 1 each of the interpolators has been populated with identical presets, placed at matching locations within the space. However, as each interpolator uses a different model to generate the preset weightings, each generates a different sonic output for the same cursor locations. For example, Figure 2 shows the weightings for each interpolator from Figure 1, with the same cursor position. As can be seen, the relative proportions of the presets are very different between the interpolators, resulting in significantly distinct sonic outputs. It should also be noted that the position of the preset objects (node, planet, cursor range, lamp or point) and their size/range have an impact on the preset weightings. The light model also provides a directional aspect

for the weightings through each lamp's chosen angle and aperture. For example, as shown in Figure 3, changing these for just two lamps within the original layout creates very different weightings even though all the locations and ranges remained identical.
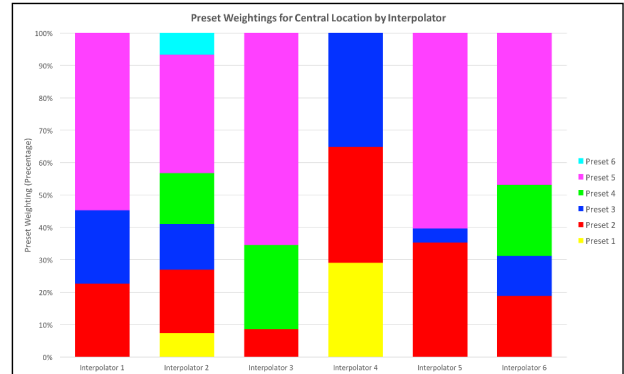


**Figure 2 Comparison of Interpolator Preset Weighting's for Identical Preset and Cursor Positions**

In this example, the angle of lamp 2 (present in all output weightings in the original layouts) was increased by 30 degrees and the angle of lamp 6 (in the original layouts only present in the output weighting for Interpolator 2) was reduced by 20 degrees. As a result of these relatively small adjustments to the interpolation space, the weightings change substantially, with preset 2 dropping to zero and preset 6 being included with a relative weighing of 32%.
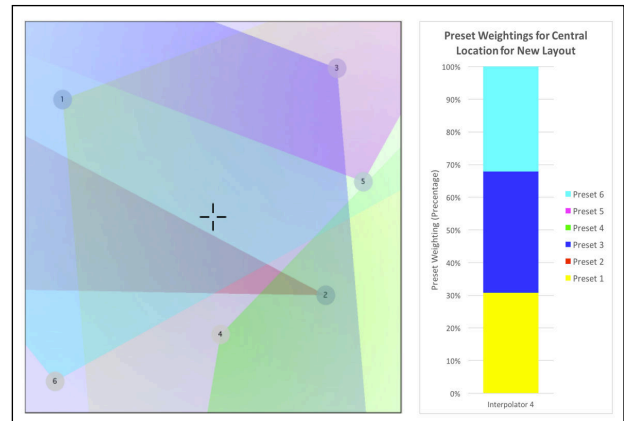


**Figure 3 Comparison of Interpolator Preset Weighting's for Identical Preset and Cursor Positions**

## 2. INTERPOLATION VISUALISATION

From the different interpolator implementations detailed above, it was possible to evaluate the visual cues provided by each interface and a summary of this is presented in Table 1. From this comparison, it becomes apparent that the Nodes and Light models are very similar intersecting models, except for the addition of the angular settings available in the light model. From previous work [14], it appears that one of the most important aspects for a graphical interpolator is the identification of regions-of-interest where interpolation is actually performed. As can be seen in the comparison table, most of the interpolator visualizations explicitly show this with some form of visual cue. However, for both the Radius-Based IWD and Voronoi Tessellation the graphics only imply where the interpolation is performed rather than explicitly showing a defined area. For the Radius-Based IWD, it is areas of free-space where more than one preset can be contained within the cursor radius. Similarly, for the Voronoi Tessellation it is areas of free-space, but this time restricted to areas where the cursor has more than one natural neighbor. Closely

aligned to this is the display of which presets are contributing to the current interpolation. For the intersecting models the presets included are shown by those intersecting the cursor position. The Gravitational and Triangulation models result in a constant, fixed number of presets (all and three, respectively) being included in the interpolation. The Radius-Based IWD explicitly shows the presets included in the interpolation as those within the cursor radius. Finally, the Voronoi Tessellation includes all presets that are natural neighbors of the cursor position. That is, the preset polygons that are adjacent to the cursor's ghost polygon. In the past, others have provided a visual cue for exactly which presets are included in the interpolation by drawing guidelines that connect all included presets to the cursor [6], [7]. This technique provides a direct representation of all presets included in the interpolation and could potentially be included in any interpolation system. However, there may be certain situations where the use of guidelines would detract from other aspects of the visual display.

Linked to being able to understand where the interpolation is performed is being able to interpret a preset's range and so its relative influence within the interpolation. With the first four interpolators, the weightings are implied primarily through a distance component and this may not always be obvious, especially when there is a secondary component, such as size, to take into account. The other two interpolators show a visualization that directly relates to the relative proportions of the contributing presets. In some of the other implementations shading and/or colour interpolation have been used to imply the weightings of presets through the interpolation space [3], [5], [6] & [7]. Although this could be added to any of the interpolation systems it may not always be desirable as it might detract from some other visual aspects, such as being able to clearly see regions-of-interest. A better solution would seem to be the area-based representation as occurs with Triangulation and Tessellation systems. The fact this is possible with both is no surprise as geometrically the two are the duals of each other, because for every Triangulation a unique corresponding Tessellation can be constructed and vice-versa [15]. However, the Voronoi Tessellation does have the benefit of extending to the full area of the space, whereas Triangulation creates a boundary defined by the outermost presets. A similar solution could be provided for the other interpolators if an additional bar-graph display is included that shows the relative weightings of the current interpolation output.

**Table 1. Interpolator Visualization Comparison**

| | Interpolator 1 | Interpolator 2 | Interpolator 3 | Interpolator 4 | Interpolator 5 | Interpolator 6 |
|---|---|---|---|---|---|---|
| Visual Model | Nodes | Gravitational | Radius-Based IWD | Light | Delaunay Triangulation | Voronoi Tessellation |
| Minimum Interpolation Requirement | Two Nodes (Overlapped) | Two Planets | Two Presets (within Cursor Radius) | Two Lamps (Overlapped) | Three Presets | Two Presets |
| Presets Included in Interpolation | Intersecting Nodes | All | Presets within Cursor Radius | Intersecting Light Beams | Presets at Cursor Containing Triangle's Vertices | Natural Neighbors |
| Field-of-Influence (Preset Range) | Shown by Node Size | Implied by Planet Size (Strength) Across All Free-Space | Implied by Cursor Radius Size | Shown by Extent of Lamp Beam | Implied by Area of Adjacent Triangles | Implied by Area of Polygons Between Natural Neighbors |
| Region-of-Interest (Interpolation Space) | Area of Node Intersections (Overlapped Node Colours) | Free-Space minus Planet Surface (White Space) | Free-Space with More Than One Preset within Cursor Radius (Shaded Cursor Region) | Area of Light Beam Intersections (Overlapped Light Beam Colours) | Free-Space within Triangulation Mesh (Triangles within Mesh) | Free-Space with More Than One Neighbor (All Polygon Surfaces Between Neighbors) |
| Preset Weightings (When Included in Interpolation) | Implied by Cursor Position in Intersection and Relative Distance to Node Center | Implied by Relative Distance to Planet and Size | Implied by Relative Distance to Presets within Cursor Radius | Implied by Cursor Position in Intersection and Relative Distance to Lamps | Shown by Relative Area of Triangles Between Cursor and the Containing Presets | Shown by Relative Area Covered by Ghost Polygon |
| Preset Recall | Non-Intersecting Area of Node (If Available) | Planet Surface | One Preset Marker with No Other Presets within Cursor Radius (If Available) | Non-Intersecting Area of Light Beam (If Available) | Preset Marker | Preset Marker |

## 3. STAR INTERPOLATION

One thing that is common with each visualization presented is that they all relate directly to the interpolation model and do not relate to the parameter space or the eventual sonic output. One previous solution has provided a visual representation of the sound signals at locations within the space, known as a spike code, that gives a time/frequency plot [16]. Although this does provide a guide to sonic differences between locations the spikes can be difficult to interpret and do not relate to the parameters and their adjustment within the space.

Given the large number of parameters that most synthesis engines possess and their sizeable data range, to solve this problem a high-dimensional visualization is required that will work in the interpolation context. Radial based plots were chosen as the center point provides a precise location for the represented preset's position within the interpolation space. Although area-based radar plots are a popular form of multi-dimensional visualization, they were not chosen here as area increases quadratically rather than linearly, which could result in users thinking small parameter changes are more significant than they actually are. In addition, the area views may interfere with the user's ability to interpret the other visual cues shown in Table 1. Our proposed solution is to use a glyph-based display, known as a star plot [17], which provides a representation of individual parameter values while also providing minimal interference with other visual aspects. With this method, each of the preset parameters is represented as a

"beam" within the star and the beam's length is proportional to the parameter value within the preset. Each beam is centered at the preset location and angularly distributed within a unit circle to show the relative value of each parameter. (Typically, with plug-in based synthesis engines, continuous parameters are normalized for external control to have a numerical range 0.0 – 1.0, but if this is not the case, scaling can be applied. At this stage, only plug-in based synthesis engines have been used so this has not been explored any further.) The star plot gives a pictorial representation for the parameter values in each preset in the interpolation space. With this the order of parameters has a large impact on its ability to effectively communicate the information. The order is defined by the synthesis engine's parameter list, as this generally matches the signal-flow through the engine. Once the user selects the desired interpolation parameters, each star plot is constructed with these in the defined order. This provides a logical order for the parameters and allows parameter values to be directly compared between the chosen presets. All the non-selected parameters remain "locked" at their non-interpolated/last values and are not shown in the plots to aid the reading of the parameters that are being interpolated. As can be seen in the example shown in Figure 4, each preset has different parameter values giving each star a unique beam arrangement.
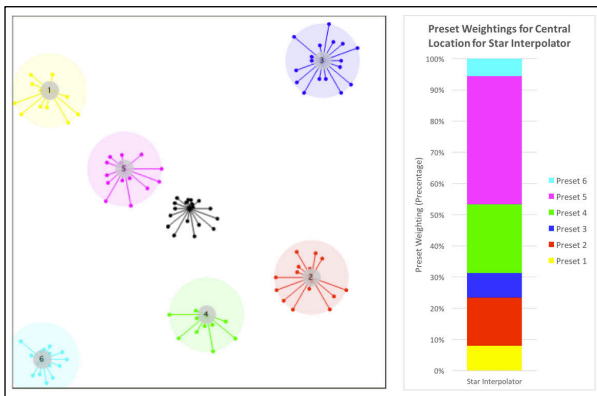


**Figure 4 Star Interpolation Visualization with Same Preset Layout and Eighteen Interpolation Parameters**

In this example, eighteen parameters (of the 149 available in this synthesis engine) have been selected for interpolation. In this case, the interpolation is realized as a generalized IWD model [8] to generate the preset weightings shown. If a different set of parameters is selected the new star plots are generated and displayed. In this way, it is possible to represent different numbers of parameters using the same mechanism. It can also be seen that the normal interpolation point's cross-hair cursor, used in the other interpolators, has also been replaced with a star plot, the shape of which is updated in real-time as the interpolation is performed. As the cursor is moved to coincide with one of the preset locations so the cursors star becomes the same as the preset's star. In this way, the interpolation point provides direct visual feedback on the parameter values that provide the biggest contribution to the current sonic output. Moreover, when moving the interpolation point within the parameter space this visualization also provides feedback on which parameter value changes are producing different sonic outputs. To provide the user with a more detailed picture of these parameter value changes, a separate larger viewer is provided that shows the parameter names, their position within the star plot and their current numerical values. An example of the cursor viewer is shown in Figure 5. This viewer is also updated in real-time to offer the user instant visual feedback that when combined with interpolation space and the real-time sonic output, provides a powerful platform for sound design tasks.
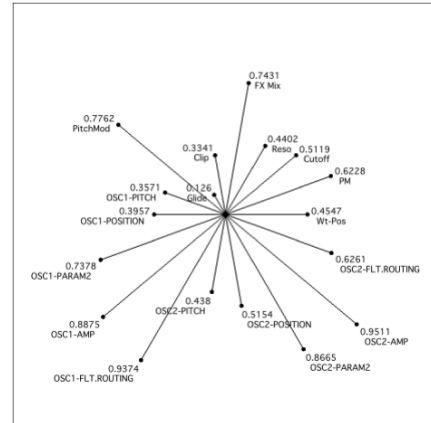


**Figure 5 Star Interpolation Point Parameter Visualizer**

An example of the full Star Interpolator interface is shown in Figure 6 that incorporates all of the features covered. As can be seen, the relative weightings of the interpolation are displayed via a bar-graph.
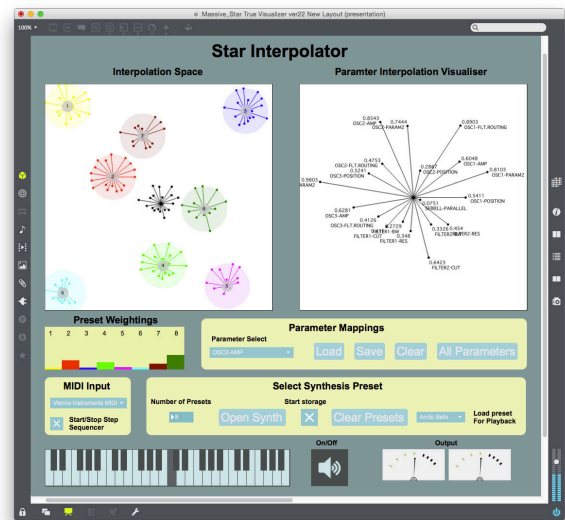


**Figure 6 Star Interpolator Full Interface**

A 3D version of the Star Interpolator was considered by expanding the model to three dimensions as others have previously done [8]. The use of a 3D model results in two different aspects that require consideration: having a representation of a 3D space allows the 3D arrangement of presets within the space and also provides the possibility of having 3D glyphs representing the presets. The use of a 3D space provides users with additional degrees-of-freedom when using the interface. Additionally, the use of 3D glyphs provides a greater area for displaying a large number of parameters. These two things are actually independent of each other, but have both been considered here. However, on implementation, challenges with this strategy became apparent. With a 3D version of a star plot it is harder to compare parameters and see the relative values directly, as the 3D projection makes it difficult to determine depths without additional visual cues. Moreover, the view of the parameters will depend on the viewer's perspective within the 3D space. This can be seen in Figure 7(a) which shows a 3D star plot, of 25 parameters, where all parameters have identical values, in this case 1.0. As can be seen it is difficult to determine the parameter values due to their different positions within three dimensions. Figure 7(b), shows the same star and parameter values from a different orientation giving a different perspective on the parameters. This makes it hard to identify the position of individual parameters within the star and to be able to compare

parameter values between presets. In light of these difficulties, a 3D display has not been pursued further at this time. Potentially allowing the user to rotate their viewing position within the space as well as providing additional visual cues (colour/shadowing) could aid the clarification of depth.
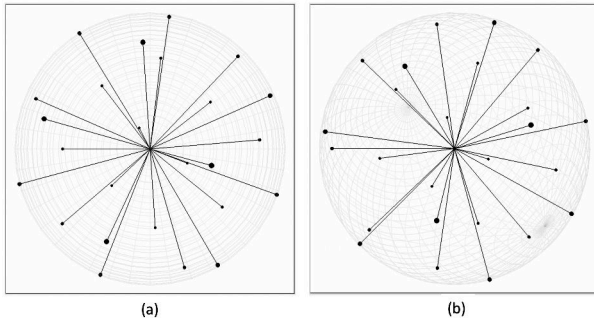


**Figure 7 3D Star Plots with 25 Parameters all with Value of 1.0 (a) Positioned at *x*=0.0, *y*=0.0 & *z* =0.0, (b) Orientated +30 degrees rotation on *x*-axis and *y*-axis**

## 3.1 Star Interpolator Exploration

The Star Interpolator was initially used in back-to-back sound design comparisons with the six different visualizations presented in Section 1. This was done by populating it with the same presets and layout as the other interpolators. The key difference between the Star Interpolator and the others is that none of the other interpolators provide any guidance on parameter differences between the presets, whereas the Star Interpolator makes these explicit. Moreover, since the cursor star shows the parameters for the current sonic output from the interpolator, it helps in gaining an understanding of the complex relationship between the parameter values and the audible output. Moving the cursor within the parameter space then allows the user to visually see which parameters are changing and by how much. This can be done to the level of individual parameters or groups. When this is combined with the sonic output, it provides a powerful mechanism for understanding the sonic palette that the interpolation space is providing. Furthermore, the real-time update of the interpolation star plot allows the users to not only establish values for the parameters, but also gain a feel for the rate-of-change of the parameters when moving between locations within the space. This was found to be very useful when trying to establish desirable parameter mappings to use in the interpolator. During our use of the system it was discovered that a good sound design strategy was to initially start with all the preset parameters mapped to the interpolator. An example is shown in Figure 8 of the Star Interpolator with all parameters mapped, in this case 149.
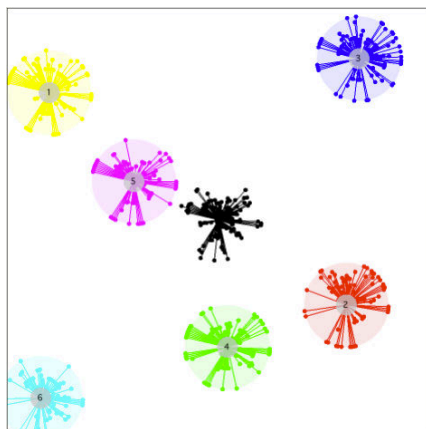


**Figure 8 Star Interpolation Visualization with All Preset Parameters Mapped**

Although the star plots become very crowded with this number of beams, the visuals still provide useful information as it is possible to

instantly identify which parameters are not changing between the selected preset sounds and so are not affecting variations in the sonic output. The mappings can then be modified to remove these parameters and simplify the sound design process. With the refined star plots it is then possible to recognize which parameters values/changes are producing the most significant impact to the sonic outputs and an iterative refinement approach can be adopted in the sound design task. In this, parameters can be selectively removed to establish their sonic footprint and if the results are found to be unsatisfactory they can always be reintroduced to recover the desirable sonic manipulation. Through this approach it is possible to identify regions within the interpolation space that generate specific audible characteristics. This was found to be particularly valuable when trying to design sonic expressions by moving the cursor between different locations within the parameter space. This process is not easy with any of the original interpolator visualizations and although their parameter mappings can also be changed, it is very difficult from the sonic output alone to identify which parameters are generating the difference between location changes within the space.

Another benefit of the Star Interpolator was found to be in the selection of presets and the setup of their layout within the interpolation space. This is often a process of trial-and-error in selecting presets that possess desirable sonic characteristics and then randomly exploring different arrangements of them within the space. However, with the star plots providing the identification of changing parameters it is possible to leave the interpolation point stationary and then individually move each preset towards and away from the interpolation point to hear the contribution that the mapped parameter changes provide. Repeating this successively provides a mechanism for constructing an interpolation space that provides the sonic changes which are desirable for the specific sound design task in-hand.

## 3.2 Star Interpolation Visualization

Despite the advantages covered in the previous section, the Star Interpolator has not been designed as a new method of interpolation, but as a different form of visualization for existing methods. For example, the star interpolation visualization could be applied to all the different interpolation models presented in Section 1. In this way, the generation of the preset weightings remains the same, but the visuals directly relate to the parameter changes and so the sonic changes being heard. However, as already covered in Section 2, the original visualizations still provide useful cues, such as, regions-of-interest, that have been shown to be beneficial to users in sound design tasks [14]. Therefore, an Interactive Visualization (IV) paradigm can be adopted, where the user can choose the visualization displayed between the original visualization, Star Interpolator or a hybrid visualization which combines both. Figure 9 shows an example of a hybrid visualization for the nodes interpolator that includes the star representation.
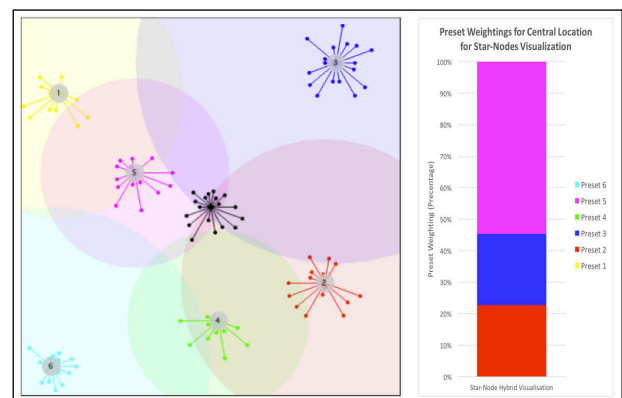


**Figure 9 Hybrid Node-Star Visualization with Same Preset Layout and Interpolation Parameters**

Here the preset layout and interpolation parameters are the same as was shown in Figure 4, but the interpolation model is the intersecting circles provided by Nodes so, as shown, the weightings generated match those from the original Nodes model.

In the current version, the alternate visualizations can be accessed by the users with hotkeys on the computer keyboard. In this way, the user can select when and how the different visualizations are used. However, the hybrid visualization, shown in Figure 9, provides all of the visual cues provided by the original nodes model, but also provides the additional cues provided by the star visualization. Through our exploration, it has been found that the hybrid visualization provides excellent detail of not only where within the space the sound will change (intersections in this case) and which parameter changes are providing sonic changes, but also when changes are not occurring and exactly when a preset is being recalled (i.e. has 100% weighting). For example, for the system shown in Figure 9, with Nodes as the interpolation model, the preset sounds can be recalled by placing the cursor on a node area where no intersection exists, if available. With complex preset layouts, these areas may not be easily apparent, especially when the sonic output is long evolving sounds. However, with the hybrid visualization it is much easier to see, as when the cursor is moved within non-intersecting regions all of the star beams remain static.

## 4. CONCLUSIONS AND FUTURE WORK

The original graphical interpolators provide a number of visual cues as well as a corresponding audio output. Although there is some linkage, albeit subtle, between the two, they do not directly correspond to each other. While one of the typical functions of a visual interpolator has been to conceal or abstract the user from the details of an underlying sound synthesis process, allowing them to focus purely on the sonic changes induced through navigation of the interpolation space, the Star Interpolator visualization provides a powerful mechanism for obtaining additional visual cues that do directly relate to the sonic output obtained from the interpolator, still without necessarily having to understand the technical details of the underlying synthesis algorithm. The real-time relationship between the visuals and the corresponding audible output offers a combined audio-visual cue that has the potential to provide more efficient navigation of the space, as has been found with other systems [18]. It also appears to offer further assistance in determining desirable parameter mappings when undertaking sound design tasks. Moreover, this visualization potentially offers a platform for beginners to become familiar with synthesizer programming techniques and understanding the complex relationships between changes to multiple parameters and the resulting sound. Adopting the IV approach means that users that do not want the additional cues provided by this visualization can choose through their interactions what is displayed and when, providing increased user flexibility.

From the review that has been undertaken so far it appears that the Star Interpolator visualization offers a number of benefits, but to further understand these and to measure their effectiveness, formal usability tests will be undertaken using a similar methodology to that previously used by the authors for the evaluation of graphical interpolators [14], [19]. This previous work has shown that the visual feedback provided by an interpolator interface changes how users interact with the interpolator [14]. This has subsequently been shown to impact on the use of an interface during a sound design task, with interfaces that provide more visual feedback resulting in wider exploration of sonic outputs, faster speed of movement and improved accuracy at locating a specific sound [19]. Moreover, these interfaces also result in a high perceived usability. Through applying a similar methodology to the Star Interpolator, it should be possible to empirically test the usability of this new visualization paradigm through a comparative approach.

## 5. REFERENCES

[1] A. Hunt, M. Wanderley, M. Paradis. The Importance Of Parameter Mapping. *In Electronic Instrument Design. Journal of New Music Research,* Vol. 32, Iss. 4, pp 429–440, 2003.

[2] D. Van Nort, M. Wanderley, P. Depalle. Mapping control structures for sound synthesis: functional and topological perspectives. *Computer Music Journal,* Sep;38(3):6-22, 2014.

[3] J.J. van Wijk and C.W. van Overveld. Preset based interaction with high dimensional parameter spaces. *In Data Visualization,* 2003.

[4] J.F. Allouis. The SYTER project: Sound processor design and software overview. *In Proceedings of the 1982 International Computer Music Conference (ICMC).* Ann Arbor, MI: Michigan Publishing, University of Michigan Library, 1982.

[5] M. Spain and R. Polfreman. Interpolator: a two-dimensional graphical interpolation system for the simultaneous control of digital signal processing parameters. *Organised Sound.* Aug 1;6(02):147-51, 2001.

[6] R. Bencina. The metasurface: applying natural neighbour interpolation to two-to-many mapping. *In Proceedings of the 2005 conference on New interfaces for musical expression,* 2005 May 1 (pp. 101-104). National University of Singapore, 2005.

[7] O. Larkin. INT.LIB–A Graphical Preset Interpolator For Max MSP. ICMC'07: *In Proceedings of the International Computer Music Conference,* 2007.

[8] T. Todoroff and L. Reboursière. 1-d, 2-d and 3-d interpolation tools for max/msp/jitter. *In Proceeding of International Computer Music Conference 2009,* ICMC'09, 2009.

[9] C. Drioli, P. Polotti, D. Rocchesso, S. Delle Monache, K. Adiloglu, R. Annies and K. Obermayer. Auditory representations as landmarks in the sound design space. *In Proc. of Sound and Music Computing Conference,* 2009.

[10] M. Marier (2012). Designing Mappings for Musical Interfaces Using Preset Interpolation. *In Conf. on New Interfaces for Musical Expression* (NIME 12), 2012.

[11] nodes. Max Reference, Cycling 74, 2020.

[12] D. Gibson and R. Polfreman. A Framework for the Development and Evaluation of Graphical Interpolation for Synthesizer Parameter Mappings. *In Proceedings of Sound and Music Computing Conference,* 2019.

[13] C. Goudeseune, Interpolated mappings for musical instruments. *Organised Sound,* 7, 85-96. 2002.

[14] D. Gibson and R. Polfreman. A Journey in (Interpolated) Sound: Impact of Different Visualizations in Graphical Interpolators. In *Proceedings of the 14th International Audio Mostly Conference: A Journey in Sound,* (pp. 215-218). ACM, September 2019.

[15] D. T. Lee and B. J. Schachter. Two algorithms for constructing a Delaunay triangulation. *International Journal of Computer & Information Sciences,* 9(3), 219-242, 1980.

[16] K. Adiloglu, C. Drioli, P. Polotti, D. Rocchesso, S. Delle Monache. Physics-Based Spike-Guided Tools for Sound Design. *Proceedings of the 13th International Conference on Digital Audio Effects,* Graz, Austria, September 2010.

[17] M. O. Ward. Multivariate data glyphs: Principles and practice. In *Handbook of data visualization* (pp. 179-198). Springer, Berlin, Heidelberg, 2008.

[18] M. Gröhn, T. Lokki, and T. Takala. Comparison Of Auditory, Visual, and Audiovisual Navigation In a 3D Space. *ACM Transactions on Applied Perception (TAP),* 2(4), 564-570, 2005.

[19] D. Gibson and R. Polfreman. Analyzing Journeys in Sound: Usability of Graphical Interpolators for Sound Design. *Personal and Ubiquitous Computing,* 2020.