Tool Virtualization and Spatial Augmented Reality

Michael R. Marner* Bruce H. Thomas†

University of South Australia

ABSTRACT

This paper presents two continua for classifying and comparing the user interfaces of virtual and augmented reality systems. We are particularly interested in user interfaces consisting of physical-virtual tools. The Within System Tool Virtualization Continuum compares the level of task overloading for tools in a single application. Application designers can use this continuum to aid in developing the tools required for a system's user interface. The Inter-System Tool Virtualization Continuum is used to compare the relative complexity of the user interfaces and tools between different systems. We have analyzed and placed several tools and systems from our previous work and from other researchers onto the continua to allow them to be compared.

Keywords: Spatial Augmented Reality, User Interfaces, Tool Virtualization Continuum.

Index Terms: H.5.2 [Information Interfaces and Presentation]: Graphical User Interfaces—Input Devices and Strategies; I.3.6 [Computer Graphics]: Methodology and Techniques—Interaction Techniques

1 INTRODUCTION

Tool virtualization is the overloading of one tool with a number of functions and attributes. For example, a pen can be overloaded with a number of different colors and tips. In the case of a physical pen the ink cartage and the tip of of the pen may be replaced. The ability to quantify the property of tool virtualization is of particular importance for physical tools in virtual environments.

This paper will explore tool virtualization for interactive Spatial Augmented Reality (SAR) applications, and in particular our Physical-Virtual Tools (PVT) [8], to support interactions within large-scale SAR environments. Sets of PVT, augmented with projected graphics, are useful for SAR systems, because different types of interactions are suited to tools with different form factors. For example, a pistol shaped tool is suited for *just out of arms reach* interaction, such as airbrushing onto an object, or for pointing. However, a stylus is more suitable for annotation directly onto an object. We have previously created dedicated tools suitable for different types of interactions. Each tool supports multiple tasks, with the active task shown by changing the projection onto the tool.

In this paper we describe two Tool Virtualization Continua (TVC) for ranking tools and the applications that use them.

The **Within System Tool Virtualization Continuum** compares the level of task overloading for tools in a single application. Application designers can use this continuum to aid in developing the tools required for a system's user interface, by comparing the different complexities of the overloaded tools. Suppose a selection of different physical tools are available for the application architect to choose or design themselves. Based on the complexity of tasks and the application, the number and styles of physical tools with

*e-mail: michael.marner@unisa.edu.au

†e-mail:bruce.thomas@unisa.edu.au

20th International Conference on Artificial Reality and Telexistence (ICAT2010), 1-3 December 2010, Adelaide, Australia

ISBN: 978-4-904490-03-7 C3450 ©2010 VRSJ

Figure 1: The Tablet tool during the color selection task.

appropriate attribute overloading will be determined. In the case of designing a SAR-based PVT, the continuum provides a means of determining which tools would have the most complex virtualization. When designing the physical shape of the tool, the ability to display virtual information on the tool itself needs to be taken into account.

The **Inter-System Tool Virtualization Continuum** compares the entire sets of tools between systems. This enables the user to understand to what extent a system virtualizes its tools compared to other systems, and to gauge the relative complexities of their user interfaces.

These continua are useful to application designers, aiding them in:

1. developing user interface toolkits for SAR applications; in particular, assessing application requirements in terms of our TVC will help designers to decide on the number and nature of tools required,
2. quantifying and ranking the level of virtualization of the set of tools in one application, and
3. comparing the complexity of user interfaces between applications.

The remainder of this paper is organized as follows: the remainder of this section describes SAR and our motivation for using the technology, followed by a description of our PVT concept. Section 2 gives an overview of the related work. Section 3 describes our two continua for classifying tools and the systems that use them. Section 4 provides concrete examples of using the continua, by classifying tools we have created previously as well as work from other researchers. Finally, we conclude, with a look towards future work.

1.1 Spatial Augmented Reality

Spatial augmented reality uses digital projectors to project perspective-correct computer generated images and information onto objects in the real world. This is in contrast to other Augmented Reality (AR) research which uses hand-held or head worn displays for providing virtual information. SAR benefits from the natural depth cues and passive haptic feedback provided by physical objects. Rather than adding purely virtual objects to the environment, SAR projections change the appearance of physical surfaces. The display technology of SAR is embedded in the environment; the users are not required to wear or carry equipment to see the augmented view. This makes SAR useful for collaborative applications and interaction techniques requiring both hands. The nature of SAR is not without limitations. SAR systems require physical surfaces to project onto; it is not easy to place a virtual object in 'mid air' in a SAR environment. As with other AR display techniques, rendering view-dependent effects requires tracking the user. In collaborative tasks, these effects would only appear correct from the specific tracked viewpoint of a single user.

Therefore, SAR systems need to be tailored to take advantage of the benefits of SAR, while avoiding the disadvantages. SAR can not replace other display technologies for applications with the restrictions described above. However, when the computer generated graphics involve modifying the appearance of physical objects, and where view dependent rendering for multiple users is not required, the benefits of SAR make it a compelling choice. The types of SAR systems considered for this paper range from small table top systems, where the user interacts with objects on the table, to larger room size systems with one or more large artifacts being augmented with computer generated images. We also want to support collaborative tasks. For example, an industrial designer and client modifying and annotating a physical mock-up in a design meeting. The users should be able to move around unencumbered, allowing them to view and interact with augmented objects from different angles.

1.2 Physical-Virtual Tools

When designing user interfaces, the designer needs to make compromises between having many single purpose tools and a single tool that is used for all interaction. An example in the physical tool world is mechanics' tools. A standard set of wrenches are included in their toolbox, with an individual wrench for every size of bolt. Alternately, a shifting spanner could be used. Dedicated wrenches are better at tightening bolts, but a shifting spanner is more flexible, as it can be used on any size bolt. Choosing one of these options is a trade off between many single purpose tools, and one flexible tool that is not as well suited to any one task. An example in the virtual domain is a virtual ink pen. Different pens could be color coded to allow the user to pick up a pen and know what color it will draw with. SMART Technologies[1] provides this capability, while a number of systems have allowed the user to change the color of the ink virtually with a single physical pen device.

Our PVT[8] are designed to encompass the entire user interface for SAR applications. Key properties of SAR systems are the user's unencumbered view of the physical world not having to wear or hold the display device. With this in mind, the user interface is based around physical tools to be manipulated. The operations supported by a tool are defined by the shape of the tool itself; employing a pencil shaped tool will perform pencil like operations. SAR allows for the projection of information onto the tool itself, highlighting the overloading of the tool's operation. The active mode of operation is conveyed to the user through visual feedback projected directly onto the tool itself. This removes the need for user interface controls to be projected onto the walls, floor, ceiling, or other objects in the environment. The user can view and interact with the system from dramatically different viewpoints, such as from either side of a car.

[1]http://smarttech.com

2 Background

Our work on physical-virtual tools is closely related to graspable and tangible user interfaces (TUI). Graspable user interfaces [2], are systems where user interaction is accomplished by manipulating physical objects that are *handles* to virtual controls. This concept is extended by Ullmer and Ishii [17], who define tangible analogues to standard graphical user interface elements, such as icons, windows, and controls. TUI have been used for a variety of applications including landscape analysis [9] and urban planning [18]. TUI concepts are compelling for interactive SAR applications, since SAR also relies on physical objects as projection surfaces. The key difference between the work in TUI and our work is TUI applications typically use separate tools for each task. For example, Urp has separate tools for measuring distances and previewing reflections. The appearance of the tools is not modified by the system itself. Our work aims to reduce the number of tools required by meaningfully overloading tools with several tasks and attributes, and changing the appearance of the tool depending on its current state.

Our work is also related to *props*, which are commonly used in virtual reality applications as physical handles to virtual objects and controls [3]. Previous research has shown being able to physically touch virtual objects enhances the user experience [4], and physical handles can help with operations such as rotating virtual objects [19]. Props have been used for 3D modeling tasks. For example, Spray Modeling [5] uses a physical airbrush prop to spray virtual material onto a model. The airbrush is similar to a PVT used for a single task, but the appearance of the tool is not augmented with extra information. Surface Drawing [12] uses household kitchen tongs as props for moving and scaling virtual objects. The main difference between this and our work is rather than using props as stand ins for tools, we want to design the tools specifically for certain types of tasks, then overload their functionality and provide feedback to the user by altering the projection. At the single tool end of the spectrum is the Virtual Tricorder [20]. This physical tool is used in VR systems for all tasks the user needs to perform. As the virtual tricorder is used in immersive VR applications, tool's physical appearance is completely replaced by computer graphics. The Personal Interaction Panel [16] is another tool used for all interactions. The panel is a blank tablet, with its appearance provided by the AR or VR system.

Our investigations into interactive SAR applications are inspired by Shader Lamps [10], a SAR technology that utilizes digital projectors to augment physical objects with computer generated images. Physical objects are represented in the system as textured 3D models. Calibrated projectors are used to project the virtual model onto the physical object. Projecting onto white objects allows different materials to be simulated, as well as other effects such as non photo-realistic rendering [11]. The Shader Lamps system has been extended to allow the user to paint onto physical objects using a tracked brush [1]. The system projects the paint color onto the tip of a brush prop, and a color palette at a fixed location onto the workbench. A tracked stylus has also been used in a SAR system for programming motion paths for industrial robots [21]. A 2D GUI is provided on a tablet PC for controlling the application. Kurz et al. [6] implemented techniques for laser pointer tracking and interaction using a pan/tilt/zoom camera. Gesture based techniques enable action at a distance in SAR environments. SAR has been used in medical applications, where a tracked stylus is used to virtually mark surgical targets onto a patient, with the marks projected digitally [14]. Manufacturing has benefited, where laser projectors mark weld points onto car parts [13].

3 The Tool Virtualization Continua

In this section we describe our two continua for classifying and comparing individual tools used in SAR systems, and for comparing the systems that make use of these tools. We discuss how ap-

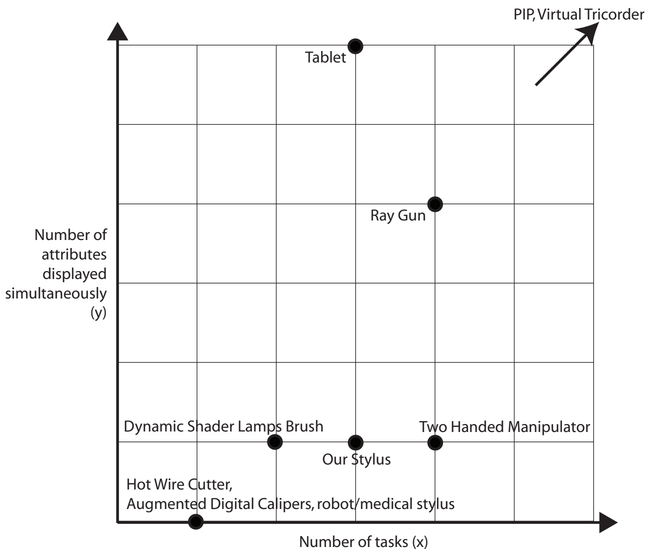

Figure 2: The Within System Tool Virtualization Continuum, showing the positions of several tools.

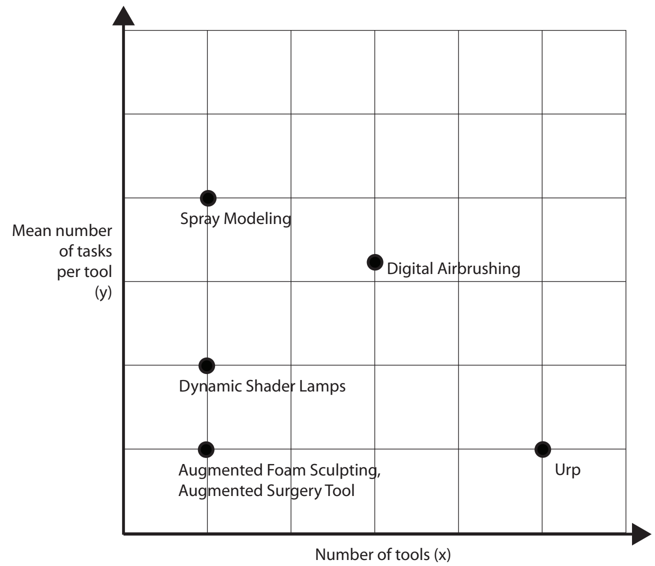

Figure 3: The Inter-System Tool Virtualization Continuum. Several applications have been placed on the graph.

plication designers can use our TVC to help when designing SAR applications. In particular, our TVC help to determine the number of tools required, and the nature (physical shape) of these tools.

3.1 Within System TVC

The Within System TVC (WS-TVC) helps designers of SAR applications to make decisions on the number of tools required for an application, and the physical nature of these tools. The continuum considers two factors: tasks and attributes. A *task* is the physical interaction required to achieve a goal. For example, the act of drawing a predefined shape on an object. Each task can have zero or more user changeable *attributes*. Selection is an example of a task requiring no attributes. In the shape drawing example, attributes may include fill color, stroke color and style, and shape geometry.

Designers can use the WS-TVC when deciding on the number and type of tools required for a system. The designer must first define the tasks and attributes for each task the system will support. Tasks that contain similar types of interaction are suitable for consideration for overloading on a single tool. For example, virtual spray painting and a laser pointer both involve pointing at an object, and therefore would require a tool with a similar form factor. These two tasks are ideal for consolidation onto a single tool. When deciding whether to have a separate task, or add an attribute to an existing task, the designer should look at the interaction required to complete the task. If the two potential tasks require identical interactions, then a single task should be used, and an attribute added. For example, ‘drawing red’ requires identical interaction from the user as ‘drawing blue’, so a single ‘draw’ task should be added with a color attribute.

The WS-TVC is shown in Figure 2. The tools listed in the figure are described in Section 4. It contains two axes. A tool’s position on the x-axis is defined by the number of tasks a tool has been overloaded with, and its position on the y-axis is defined by the number of attributes for the task with the most attributes. The x-axis tells the designer the relative burden of the tool. There is more cognitive load required in swapping the active task, and recalling the active task as the number of tasks a tool supports increases. In addition, the physical design of the tool will be less ideal for any particular task as the number of supported tasks increases, as the design must accommodate different kinds of interaction. The y-axis places design constraints on the physical tool, since as the number of attributes increases, more information must be shown simultaneously, requiring a larger projection area.

This leads to the two competing factors on placing a system on the continuum, number of *physical tools* verses the number of *attributes* and *tasks* overloaded on each *physical tool*. In the case of the pen, there are a number of standard attributes that can be overloaded including color, line thickness, and tip shape. The tip shape is good example of how the continuum allows the developer to decide between two factors. On the one hand virtually overloading a single physical pen with different tool tips requires less physical tools to accommodate all tasks, while on the other hand drawing with different tip shapes on the pen *feels* different to the user. The developer must make a design decision on the best outcome for the system based on the requirements given to them. Virtual pens can do different actions to physical pens, such as line patterns, gradient line colors, automatic straightening, and automatic arrowheads.

The above pen example describes different attributes for a single physical tool that had the same basic functionality, a hand-held drawing instrument. However, the tool could also be overloaded with completely different tasks. For example the pen device could also perform selection tasks. There are many different modes of performing selections, such as: select, copy, paste, duplicate, cut, or resize.

3.1.1 Using the WS-TVC

Once a designer has decided on the base number of tools, they should be placed on the WS-TVC and evaluated. As a tool’s position in the x-axis increases, the designer should consider adding another tool with a *different* physical design. The new tool should be designed for a subset of the original tool’s supported tasks. As a tool’s position on the y-axis increases, the designer should consider adding another tool with an *identical* physical design to remove the need for some of the attributes. For example, changing from a ‘predefined shape’ tool with geometry and fill color attributes, to separate tools for each geometric shape, so each tool only requires the color attribute.

3.2 Inter-System TVC

Where the WS-TVC compares the individual tools that make up a system, the Inter-System TVC (IS-TVC) is used to compare the

Figure 4: The Ray Gun Tool.

Figure 5: The Stylus Tool.

relative complexity of user interfaces among different systems. The IS-TVC, shown in Figure 3, illustrates the transition in the x-axis from a single virtualized tool responsible for all user input to having dedicated tools for each task, and on the y-axis the transition of mean number of tasks per tool from a single task to every task mapped onto each tool.

We see the increase in the y-axis as an increase of cognitive load of a total system. On average, each tool is required to support more tasks, and as such requires a more complicated mental mapping when changing tasks. While an increase in the x-axis increases the complexity of physical workspace to support more physical tools. This increase in complexity would impact the size and organization of the workspace to accommodate the tool set and the time required to physically change tools.

3.2.1 Using the IS-TVC

IS-TVC allows one to compare the complexity of the user interface between systems and visualize their relative cognitive load and complexity of physical workspace. We propose the complexity of user interface increases as the system moves away from the origin of the graph. Systems in the lower left of the IS-TVC would have less complex user interfaces. While moving into the upper right of the continuum would have the most complex user interfaces. Currently most systems would be towards the left hand side of the continuum but spread vertically. TUI research is exploring the regions more to the right of the continuum.

4 Example Tools

This section describes several of the physical-virtual tools we have developed for our SAR systems, and places them on the WS-TVC. We will give tools a rank in the form (x,y), where *x* is the tool's task rank, and *y* is the tool's attribute rank. We will also discuss tools that have been developed by other researchers, and where possible place these tools on the WS-TVC as well. The tools' positions on the continuum are illustrated in Figure 2. For implementation details of the applications built using these tools, please see our previously published physical-virtual tools [8] and augmented foam sculpting [7] papers.

4.1 Ray Gun

The *Ray Gun* (Figure 4) is a pistol shaped device held in the user's dominant hand. The top of the device is flat, providing an area for projection. This form factor is suited to tasks that involve pointing at objects. The ray gun was intentionally designed with a large flat area for projection, allowing for more overloaded tasks and more complex attributes to be shown.

We have previously used the Ray Gun for the following overloaded tasks:

- Airbrushing onto objects. This task has several user changeable attributes: paint color, spray angle, brush hardness, and paint flow.
- Virtual laser pointer. This task has a single user changeable attribute: The color of the laser dot.
- Drawing custom stencils on the tablet. This task has no user changeable attributes.
- Command entry on the PIP. This task also has no user changeable attributes.

Therefore, we give this tool a rank of (4,4). As this tool has a high task rank, we have needed to modify the appearance of the tool depending on the active task. In airbrush mode, an arc is projected onto the tool filled with the current paint color. As with a physical airbrush, the further one holds the airbrush away from the spray surface, the wider the painted area. The angle of the arc represents the spray angle of the brush when painting. We have chosen this representation so the user can quickly see the brush mode. When painting, the top of the tool lights up with the paint color, indicating a paint operation is in progress. This gives the user feedback that painting is occurring, even if the user has the device pointed away from any objects. In laser pointer mode, the projection onto the device changes to an arrow pointing towards the tip of the tool, indicating laser pointer mode. We could have provided a tool for each of these functions; however as the form factors of the tools would be similar we have chosen to combine the functions onto a single virtualized tool. We do not change the appearance of the tool for command entry and stencil drawing, as the tool reverts back to its previous state once these tasks are complete.

4.2 Stylus

The stylus (Figure 5) is an easy to hold pen like device, with a small flat area suitable for projection. This tool has been used to compliment the Ray Gun. While it would have been possible to use the system exclusively with the Ray Gun, the stylus has been provided for tasks where a pistol grip is less suitable for the type of interaction required.

The tasks we have used the stylus for include:

- Annotation onto design artifacts. This task requires a single attribute: the color of the pen.

- Drawing custom stencils on the tablet.

- Command entry on the PIP. Again, this task also has no user changeable attributes.

Note that two of the tasks supported by the Stylus were also supported by the Ray Gun. We made this decision so the user could work with whatever tool they were currently holding, and so two users could work at the same time with different tools.

We give the Stylus a rank of (3,1). The low attribute rank is important for this tool. It is designed to be held in the hand like a pen, and therefore can only provide a small area for projection. As our system grows more complex, more attributes will be added, such as line style and thickness, increasing the attribute rank. We foresee that eventually there will be too many attributes and another tool will need to be added, taking over some of the functionality.

4.3 Hot Wire Cutter

The *Hot Wire Cutter* (Figure 6) was developed for Augmented Foam Sculpting [7]. This is an example of a highly specialized tool, used for a single task: cutting through a piece of foam while simultaneously simulating the cut operation on a virtual model. No customizable attributes are available to the user. Therefore, this tool would receive a rank of (1,0).

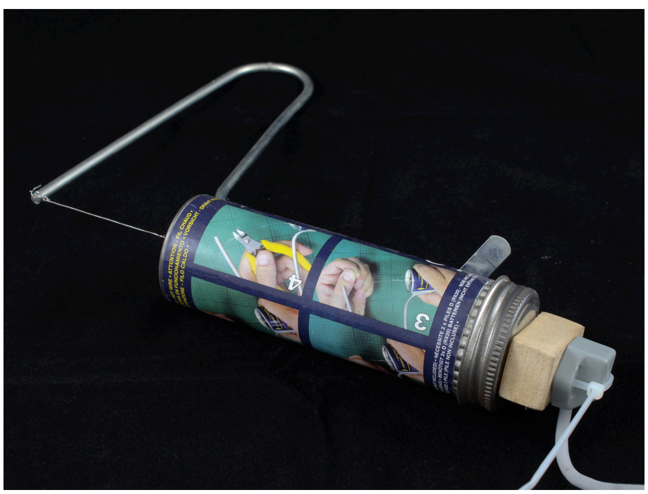

Figure 6: The Hot Wire Cutter.

4.4 Tablet

The tablet (Figure 1) is a physical board approximately the size of an A4 piece of paper. Our tablet can be considered a PIP [16] style tool, being used for a variety of tasks:

- Stenciling for the airbrush. This task has two attributes: the stencil shape and whether the stencil is normal or inverted.

- A target for drawing new stencils. Here we can consider the stencil shape being drawn as an attribute.

- Command entry. The tablet is used to change attributes for other tasks through buttons and controls projected onto it. The number of attributes can therefore be considered as the sum of attributes for the other tasks.

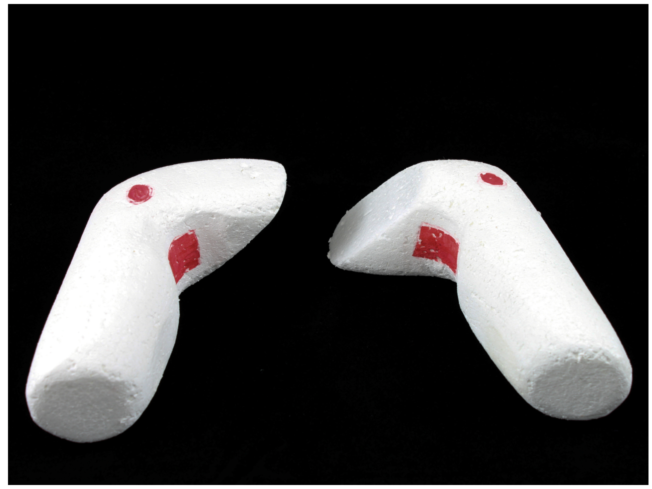

Figure 7: Mockup of the two handed manipulation tool. Red shows button locations

The command entry task allocated to this tool makes it difficult to place in the continuum, as the position depends on a specific application. However, if we restrict our classification to the airbrushing application, the tablet would receive a rank of (3,6). We consider the tablet closest to a single virtual tool, as it supports the greatest number of tasks. In theory a PIP could support any user interface controls that appear on a traditional desktop. Our tablet supports a number of buttons and different control devices.

4.5 Possible Future Tools

We have created several general purpose and specialized tools for our SAR systems. However, we plan on extending our toolbox with additional devices.

A *two handed manipulator*, shown in Figure 7, will allow two handed interactions. This is another example of a tool that can be heavily overloaded with a number of tasks. We foresee this tool being used to position, rotate, and scale virtual elements in the system. In addition, this tool could be used for tasks such as placing predefined shapes onto objects. One hand would define the center point of the shape to add, with the relative position of the other hand defining the size and orientation of the shape. Based on this prediction, we can give this tool a rank of (4,1).

Augmented Digital Calipers could be used for precise measurement in a SAR environment. Such a tool would be useful, as current real time tracking technologies do not provide the high precision and accuracy that is possible with digital calipers. This is an example of a specialized, single purpose tool, and as such would receive a rank of (1,0).

4.6 Tools From Other Researchers

So far we have explored our TVC in respect to our own previous research. In this section we revisit several of the tools discussed in Section 2 and attempt to place them on the WS-TVC. We also look at the systems as a whole, placing them on the IS-TVC.

Dynamic Shader Lamps [1] provides a tracked paintbrush for digitally paint onto physical objects. This tool has two tasks: painting and color selection, with a single attribute: the active color. Therefore we can place the paintbrush at (2,1) on the WS-TVC. Dynamic Shader Lamps as a whole would be placed at (1,2) on the IS-TVC. The stylus tools used for industrial robot programming [21] and marking surgical targets [14] would both be placed at (1,0) on the WS-TVC, as they are used for single tasks with no user changeable attributes. The robot programming system would be placed at

(2,2) on the IS-TVC, as it also uses a tablet PC for command entry, while the surgery application would be placed at (1,1) on the IS-TVC.

Spray Modeling [5] uses a single tool for its functionality: the airbrush. The airbrush is responsible for three tasks: line drawing, volume spraying, and air spraying. This gives the tool a task rank of 3. It is unclear from the paper exactly how many attributes were defined for each task, so we are unable to give the tool an attribute rank. It is also worth noting that the tool was not augmented with information based on the active task. The application as a whole would be placed at (1,3) on the IS-TVC, showing a heavily overloaded tool used for all interaction.

Urp [18] is a TUI system utilizing five tools: clock, distance, reflection, wind, and camera. This system is a good example of having many single purpose tools. As such, it is placed at (5,1) on the IS-TVC, with each tool receiving a score of (1,0) on the WS-TVC.

The Personal Interaction Panel (AR/VR) [15], Virtual Tricorder (VR) [20], and standard computer mouse are examples single tools that are used for all interaction. It is difficult to place these tools exactly on the WS-TVC, as their position would depend on the system in which they are being used. However, we can say that both the attribute rank and task rank for these tool is high. This is because the tools are designed to be used for all user interaction within a system. Along the same lines, we can classify systems using these tools as being in the upper left region of the IS-TVC.

5 Conclusion

This paper has presented two continua for measuring virtualization and task overloading of tools in virtual environments. The WS-TVC compares individual tools used in a single system. It features two axes: task rank and attribute rank. Application designers can use this continuum to aid in making decisions regarding the number and physical design of the tools required for a system. The IS-TVC is used to compare the relative complexity of user interfaces among multiple systems. This two axis continuum considers the number of tools in a system, and the mean number of tasks per tool.

We have compared several of the tools we have developed for the industrial design domain by analyzing their uses and placing them on the WS-TVC. We have also compared our work against selected previous research, by placing tools on the WS-TVC and systems as a whole on the IS-TVC. We have described how the WS-TVC can aid designers when designing systems. In the future we would like to investigate whether general optimal numbers of tools and levels of task overloading exist. This would further aid designers, as we would be able to divide our continua into “acceptable” and “unacceptable” regions.

Acknowledgements

The authors wish to thank the Wearable Computer Lab of the University of South Australia for collaboration on this project. We would also like to thank the Mawson Institute of the University of South Australia for their support in setting up the Spatial Augmented Reality Lab.

References

[1] D. Bandyopadhyay, R. Raskar, and H. Fuchs. Dynamic shader lamps: Painting on movable objects. In *IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 207–216, 2001.

[2] G. W. Fitzmaurice, H. Ishii, and W. A. S. Buxton. Bricks: Laying the foundations for graspable user interfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 442–449, Denver, Colorado, United States, 1995. ACM Press/Addison-Wesley Publishing Co.

[3] K. Hinckley, R. Pausch, J. C. Goble, and N. F. Kassell. Passive real-world interface props for neurosurgical visualization. In *Proceedings of the SIGCHI conference on Human factors in computing systems: celebrating interdependence*, pages 452–458, Boston, Massachusetts, United States, 1994. ACM.

[4] H. Hoffman, A. Hollander, K. Schroder, S. Rousseau, and T. Furness. Physically touching and tasting virtual objects enhances the realism of virtual experiences. *Virtual Reality*, 3(4):226–234, 1998. 10.1007/BF01408703.

[5] H. Jung, T. Nam, H. Lee, and S. Han. Spray modeling: Augmented reality based 3D modeling interface for intuitive and evolutionary form development. In *International Conference on Artificial Reality and Telexistence*, 2004.

[6] D. Kurz, F. Hantsch, M. Groe, A. Schiewe, and O. Bimber. Laser pointer tracking in Projector-Augmented architectural environments. In *International Symposium on Mixed and Augmented Reality*, pages 19–26, 2007.

[7] M. R. Marner and B. H. Thomas. Augmented foam sculpting for capturing 3D models. In *IEEE Symposium on 3D User Interfaces*, Waltham Massachusetts, USA, 2010.

[8] M. R. Marner, B. H. Thomas, and C. Sandor. Physical-Virtual tools for spatial augmented reality user interfaces. In *International Symposium on Mixed and Augmented Reality*, Orlando, Florida, 2009.

[9] B. Piper, C. Ratti, and H. Ishii. Illuminating clay: a 3-D tangible interface for landscape analysis. In *Proceedings of the SIGCHI conference on Human factors in computing systems: Changing our world, changing ourselves*, pages 355–362, Minneapolis, Minnesota, USA, 2002. ACM.

[10] R. Raskar, G. Welch, K. Low, and D. Bandyopadhyay. Shader lamps: Animating real objects with Image-Based illumination. In *Rendering Techniques 2001: Proceedings of the Eurographics*, pages 89–102, 2001.

[11] R. Raskar, R. Ziegler, and T. Willwacher. Cartoon dioramas in motion. In *NPAR '02: Proceedings of the 2nd international symposium on Non-photorealistic animation and rendering*, page 7ff, New York, NY, USA, 2002. ACM. inproceedings.

[12] B. Schwerdtfeger, D. Pustka, A. Hofhauser, and G. Klinker. Using laser projectors for augmented reality. In *Proceedings of the 2008 ACM symposium on Virtual reality software and technology*, pages 134–137, Bordeaux, France, 2008. ACM.

[13] B. Schwerdtfeger, D. Pustka, A. Hofhauser, and G. Klinker. Using laser projectors for augmented reality. In *Proceedings of the 2008 ACM symposium on Virtual reality software and technology*, pages 134–137, Bordeaux, France, 2008. ACM.

[14] B. Seo, M. Lee, H. Park, J. Park, and Y. S. Kim. Direct-Projected AR based interactive user interface for medical surgery. In *Artificial Reality and Telexistence, 17th International Conference on*, pages 105–112, 2007.

[15] Z. Szalavri and M. Gervautz. The personal interaction panel - a two handed interface for augmented reality. pages 335–346, Budapest, Hungary, 1997.

[16] R. Szeliski and H. Shum. Creating full view panoramic image mosaics and environment maps. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 251–258. ACM Press/Addison-Wesley Publishing Co., 1997.

[17] B. Ullmer and H. Ishii. The metaDESK: models and prototypes for tangible user interfaces. In *Proceedings of the 10th annual ACM symposium on User interface software and technology*, pages 223–232, Banff, Alberta, Canada, 1997. ACM.

[18] J. Underkoffler and H. Ishii. Urp: a luminous-tangible workbench for urban planning and design. In *Proceedings of the SIGCHI conference on Human factors in computing systems: the CHI is the limit*, pages 386–393, Pittsburgh, Pennsylvania, United States, 1999. ACM.

[19] C. Ware and J. Rose. Rotating virtual objects with real handles. *ACM Trans. Comput.-Hum. Interact.*, 6(2):162–180, 1999. 319102.

[20] M. M. Wloka and E. Greenfield. The virtual tricorder: a uniform interface for virtual reality. In *Proceedings of the 8th annual ACM symposium on User interface and software technology*, pages 39–40, Pittsburgh, Pennsylvania, United States, 1995. ACM.

[21] M. Zaeh and W. Vogl. Interactive laser-projection for programming industrial robots. In *Mixed and Augmented Reality, 2006. ISMAR 2006. IEEE/ACM International Symposium on*, pages 125–128, 2006.