Spatial Augmented Reality User Interface Techniques for Room Size Modelling Tasks

Michael R. Marner[1] **Bruce H. Thomas**[1,2]

[1] Wearable Computer Lab, Advanced Computing Research Centre
University of South Australia,
Mawson Lakes Bvd, Mawson Lakes, South Australia 5095,
Email: michael.marner@unisa.edu.au

[2] Email: bruce.thomas@unisa.edu.au

Abstract

This paper present results of our investigations into using spatial augmented reality to improve kitchen design and other interior architecture tasks. We have developed user interface techniques for room sized modelling tasks, including cabinet layout, viewing and modifying preset designs, and modifying materials and surface finishes. These techniques are based on Physical-Virtual Tools, which consist of physical input devices augmented with projected information. These tools and techniques address key user interface issues for spatial augmented reality systems, and we discuss how they can be generalised for other applications. The techniques have been developed in the context of a demonstration application, *BuildMyKitchen*. BuildMyKitchen allows architects to design kitchen cabinets and layouts, and work with clients on the design, in an interactive spatial augmented reality environment.

Keywords: Spatial Augmented Reality, User Interfaces, Architecture.

Figure 1: Resizing the Kickboard Inset using the Resizing Wand and one half of the Resizing Tool.

1 Introduction

We have been investigating the use of Spatial Augmented Reality (SAR) (Raskar, Welch, and Fuchs, 1998) to aid in the design process (Thomas et al., 2011). This paper presents a set of SAR tools to aid in interior architecture work such as kitchen design. An example SAR design application, *BuildMyKitchen*[1] has been developed in consultation with architect Steve Kelly from the University of South Australia's School of Architecture and Design. In developing BuildMyKitchen, new user interface tools and techniques have been created. BuildMyKitchen's user interface is comprised of Physical-Virtual Tools (PVT) (Marner and Thomas, 2010) that support the designer in both the design process and designer/client meetings. The tools that comprise the user interface of BuildMyKitchen are shown in Figure 4. The tasks supported by the tools are: *Cabinet Layout*, *Design Presets*, and *Modifying Finishes*. While these techniques have been developed in the context of kitchen design, they can easily be generalised for use in other application domains and address important interaction issues for SAR systems, such as interaction techniques for room size environments and the lack of virtual projection surfaces.

This paper makes the following contributions to SAR research:

- Cabinet Layout. Physical-virtual tools that allow designers to specify dimensions and distances in a room size SAR environment. These techniques have been optimised for a range of physical movements from floor level to overhead structures, as most Virtual Environment (VE) interaction techniques are designed for the user to be in a single posture, standing or sitting.
- Design Presets. PVT user interface techniques based on a "colour swatch" metaphor for saving and loading preset designs in a SAR environment. These have been developed to allow both the designer and client to quickly access previous developed designs without the need for external technology.
- Modifying finishes using a "magic touch" metaphor based user interface technique for modifying attributes of the final design.
- BuildMyKitchen. This application represents an example exploration of how SAR can be used as part of the design process for interior architecture projects, and serves as a concept demonstrator encapsulating the ideas of the new PVT's.

This paper is structured as follows. Firstly, the motivation for exploring the use of SAR for interior architecture is described. The remainder of the paper describes the functionality of BuildMyKitchen, and shows how the system improves the design process for both the architect and client. This application provides the motivation for

[1]An earlier version of this work (Marner and Thomas, 2013) used the name "PimpMyKitchen", a name inspired by the television programme "Pimp My Ride", which involves customising and improving cars.

the specialised PVT's presented in this paper. The paper describes the new user interface tools and techniques developed to support room size modelling tasks, and how they can be generalised for use in other application areas.

2 Motivation

The motivation for investigating the use of SAR in interior architecture stems from the difficulty and complexity of this design process. SAR uses full scale white replicas (walls, cabinets, and large appliances) to provide a flexible and understandable environment to convey large scale designs. In the case of a kitchen design process, clients currently go through several steps with the designer. Firstly, the client will meet with the architect to discuss their requirements. Different lifestyles are suited to different kitchen designs, and the architect takes these requirements into consideration during the design process. Following this discussion, measurements of the available space are taken, and an initial design is created. This design is presented to the client through 2D plans and 3D rendered images on a computer. Based on these images, the client and architect discuss changes to the design. An additional task for the client is to decide on material finishes, appliances, door handles, etc. To aid in this process, kitchen builders have large showrooms with several finished kitchens. Even still, the client usually will not be able to truly visualise their new kitchen until the final installation in their home.

Figure 2: A design flaw in a completed kitchen, where the dishwasher door and pantry door collide. Damage can bee seen on the pantry door.

A key motivating goal of BuildMyKitchen is to enhance the understanding and decision making ability of the client. While architects are accustomed to visualising the end result from plans and CAD drawings, the client does not usually possess these skills. Even with high quality 3D renders, it can be difficult for the client to envision the scale of the final product. BuildMyKitchen makes it possible to preview designs at 1:1 scale early in the design process. Furthermore, using physical blanks for the kitchen cabinets provides a more intuitive environment than what is possible with a standard computer. This helps not only in deciding on the visual design of the kitchen, but also the functionality of the layout. For example, Figure 2 shows a design flaw in a completed kitchen. Here, the dishwasher door is obstructed by the door to the pantry. To access the dishwasher, the pantry door must be completely closed. Even a slight opening is enough to knock the dishwasher, resulting in damage to both the appliance and the door. These kinds of problems are difficult to predict when looking at a 3D render. Previewing the kitchen at 1:1 scale, with physical mock-ups can help in detecting these kinds of problems. Using SAR allows the projected content of the physical mock-ups to be changed quickly, making iterating the design easier. This ability becomes even more important given the rise in popularity of self-install flat-pack kitchens, where there is no trained architect and limited opportunities to preview the design before construction commences.

Another motivating goal is to provide designers and architects with tools to manipulate and visualise designs on a 1:1 scale. While these professionals are quite capable of visualising designs from drawings in full scale, SAR provides a new medium to explore designs. Discussions with professional architects (Thomas et al., 2011) indicate SAR provides an extra dimension to the design process. To facilitate the manipulation of designs in-situ of the SAR environment and not force the designer to make all their changes in a desktop CAD system, new tools and techniques are required beyond the current state of the art for SAR environments.

3 Background

We are interested in creating 1:1 previews of designs early in the design process. Hare et al. (2009) have demonstrated the importance of physical prototypes to industrial design. DisplayObjects (Akaoka and Vertegaal, 2009) bring spatial augmented reality to product design by projecting user interface controls and other elements onto simple physical mockups. Porter et al. (2010) investigated using SAR to provide interactive controls for control panels early in the design process, rather than using electronic physical controls. (Schwerdtfeger et al., 2008) have demonstrated how laser projectors can be used for quality assurance in industrial settings. WARP (Verlinden et al., 2003) shows how SAR can be used to preview materials on scale models of design mockups. SAR has also been used for maintenance and instruction. CADCast (Piper and Ishii, 2001) shows how SAR can be used to project in-situ instructions for assembly. Suganuma et al. (2008) demonstrate how SAR can be used to instruct a player in how to play Billiards, utilising an overhead camera to detect locations of billiard balls. SAR has also been used to directly project information onto patients to help doctors during surgery (Byung-Kuk Seo et al., 2007).

Our work builds on this foundation, bringing SAR technology to kitchen design and architecture. Our work is inspired by Shader Lamps (Raskar, Welch, Low, et al., 2001); we use white cabinet 'blanks', and use projectors to create the detailed appearance of the design. (Low et al., 2001) used SAR technology to build life sized dioramas for visualizing room size entities with a mixture of physical walls and virtual projected imagery. Bandyopadhyay, Raskar, and Fuchs (2001) demonstrate how SAR can be used to modify the appearance of movable objects using a tracked stylus and 'paintbrush' metaphor.

Physical-Virtual Tools (PVT) (Marner and Thomas, 2010) support interactions within a large-scale SAR environment. PVT are designed to encompass the entire user interface of the SAR applications. The user interface is based around physical tools themselves, in a similar way to Tangible User Interface systems (Ullmer and Ishii, 1997). The operation modes supported by a tool are defined by the shape of the tool itself; picking up a pencil like object will perform pencil like operations (Fitzmaurice, Ishii, and Buxton, 1995). The use of SAR allows for an understandable overloading of the tool's operation. The active mode is conveyed to the user through visual feedback projected directly onto the tool itself. No user interface controls are projected onto the artifact, walls, floor, or ceiling. The user can view and interact with the design artifact from dramatically different viewpoints, such as from different locations within a kitchen.

4 BuildMyKitchen

As previously mentioned, BuildMyKitchen was developed to support the early design process of kitchens and designer/client meetings. To support an architect when developing the initial kitchen design, *blanks* (white simple shaped light weight 3D projection substrates) representing the cabinets are placed in an environment of the shape and size of the target kitchen. One blank is shown in Figure 3. The architect decides on the basic layout of the kitchen by moving the cabinet blanks into the desired locations. The prototype workflow involves the following tasks. Architects first block out key positioning decisions, such as doors and major appliances, to optimise the efficiency of the kitchen, and then work on the next level of detail. Once the blanks are in position, the architect can focus on the layout of the cupboards, drawers, and other components. Some of these components will need to be resizable, depending on the vision of the architect; some, such as appliances, have set dimensions.

Figure 3: An example Cabinet Blank.

The second intended use case for BuildMyKitchen is during meetings where both the architect and client are present. The goal of these meetings is to come to an agreement on the layout of the kitchen, and for decisions regarding material finishes and component selection to be made. There are an overwhelming number of possible combinations of appliances, materials, handles, etc. Rather than have the client select from such a large number of choices, BuildMyKitchen allows the client to select from and customise a set of "preset" designs. These presets are created beforehand by the architect using their experience and design sense. The client chooses a preset on which to base their design and is then able to customise it using a smaller set of options. This set of options is also put together before hand by the architect.

5 Cabinet Layout

A key task for the architect is to decide on the layout of components placed into the cabinets, such as cupboards, drawers, and appliances. As previously mentioned, some of these components, such as appliances, have fixed dimensions that need to be accommodated. Others will be set by the architect based on their design expertise. Some components do not require a specific size, and the component can simply make use of whatever space is available. For example, evenly sizing the height of a set of drawers to make use of the entire height of the cabinet.

Figure 4: The tools that comprise the BuildMyKitchen user interface. Top-left: Resizing Wand, top-right: SAR Swatches, bottom-right: Two-handed Resizing Tool, bottom-left: Stylus.

In addition to components in the cabinets, other design features also need to have dimensions set by the architect. These include:

- Bench thickness. Different benchtop materials are manufactured to different thicknesses. The architect may also choose a specific thickness for the benchtop.
- Kickboard height. The kickboard is the inset material at the base of the cabinet. Choosing the height of the kickboard requires a tradeoff between a functional kickboard and maximum cupboard space.
- Kickboard inset. The inset is the amount the kickboard is set in from the face of the cabinet. Commercial kitchens often have deep insets so workers do not kick cabinets with their feet.
- Bench overhang. The overhang is how much the benchtop overhangs the rest of the cabinet.
- Component spacing. The component spacing is the horizontal and vertical space between components, such as adjacent cupboard doors. Older kitchen designs often have large spaces between components, resulting in smaller doors, where as modern designs tend to maximise the size of the doors, making it easier to see into the cupboards when the door is open.

To accommodate these requirements, BuildMyKitchen automatically places and sizes components according to their properties. This is done using the following basic component types:

- Containers. A container contains zero or more other components, and is responsible for arranging them either vertically or horizontally according to the space available.
- Doors and drawers. A door is placed into the cabinet with the handle either to the left or right. Drawers have the handle placed horizontally, centred at the top of the component.
- Appliances, such as dishwashers and ovens.
- Blank spaces, used for filling an area in the layout.

Drawers, doors, and spaces are able to have preferred dimension set. When layout out a cabinet, containers will try to give components their preferred size. Appliances have dimensions that cannot be altered by the architect, so they are given priority over other components. As containers can be nested, any layout can be produced using the right combination of containers and components. Producing the final layout is a recursive process, with each container allocating space to its children. The algorithm shown in Listing 1 produces a horizontal layout. Vertical layouts are produced in the same way, adjusting component heights instead of widths.

```
void Container::resize(float w, float h, float padding
    , Vector topLeft) {
  int elasticCount = 0;
  float availableSize = w;
  foreach (PanelComponent* component, mContents) {
    if (component->hasPreferredWidth())
      availableSize -= component->getPreferredWidth();
    else
      elasticCount++;
  }
  availableSize -= (mContents.size()-1) * padding;
  wcl::Vector currentTopLeft = topLeft;
  foreach (PanelComponent* component, mContents) {
    if (component->hasPreferredWidth()) {
      if (elasticCount > 0) {
        component->resize(component->getPreferredWidth
            (), h, padding, topLeft);
        topLeft.x += component->getPreferredWidth() +
            padding;
      }
      else {
        component->resize(w / mContents.size(), h,
            padding, topLeft);
        topLeft.x += w / mContents.size() + padding;
      }
    }
    else {
      component->resize(availableSize / elasticCount,
          h, padding, topLeft);
      topLeft.x += availableSize / elasticCount +
          padding;
    }
  }
}
```

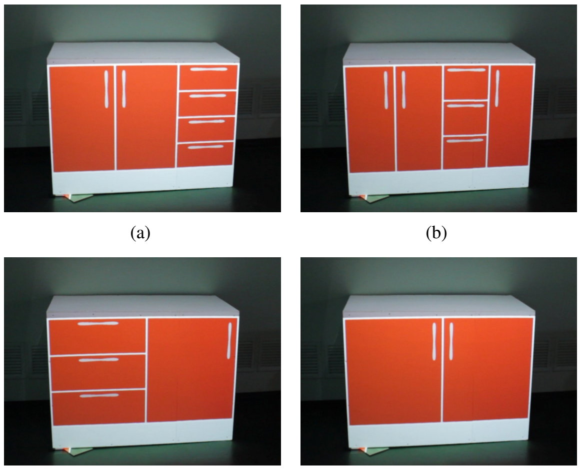
Listing 1: Algorithm for producing a horizontal cabinet layout

5.1 Creating a Cabinet Layout

The architect starts with a cabinet blank with no components added to it. They can then add components to the layout using a simple keyboard interface. As components are added, the system automatically arranges the components to fit the available space. Figure 5 shows four different layouts on the same cabinet. Once the necessary components have been added, the architect can move on to specifying dimensions to the components, as shown in Figure 6. For example, drawers used to store saucepans will need to be wider and deeper than drawers used to store cutlery. At a higher level, the architect can experiment with the overall kitchen layout by moving the cabinet blanks into different positions in the room.

5.2 Resizing Using The Two-Handed Resizing Tool

The two-handed manipulation tool has been developed for performing operations such as resizing components of the kitchen. The tools consists of two identical halves which are held in each hand. To resize a component, the user places each half of the tool against opposite edges of the component to be resized, and adjusts the distance between their hands until the desired size is obtained. This is an intuitive gesture, and is a similar motion to how people estimate and communicate distances in the real world.



(a) (b) (c) (d)

Figure 5: Example cabinet layouts.

This tool is used exclusively for resizing. Resizing in BuildMyKitchen has a single user definable attribute: the snapping mode. The user is able to choose between snapping to 1mm, 5mm, and 10mm intervals when resizing. In addition, when a set of common dimensions are used in kitchen design, the user can choose to snap to the nearest preset dimension. The snapping mode is conveyed to the user by projecting this information onto both halves of the tool. The current dimension is also projected onto the tool, allowing the architect to precisely set dimensions with little effort.

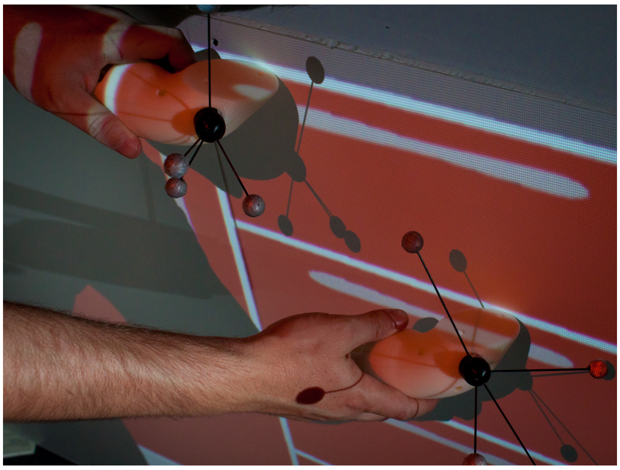
Resizing components on the surface of the blank is accomplished by placing each half of the tool on opposite edges of the component to be resized. The architect resizes the component simply by changing the distance between the two halves. Resizing the kickboard inset and the benchtop overhang is a more difficult problem, because the physical cabinet blank does not match the virtual geometry. Resizing these components is accomplished by placing one half of the tool on the component to be resized, such as the front of the bench overhang, and the other on the top of the cabinet. A line is projected onto the cabinet blank indicating the amount of overhang. The architect changes this dimension by dragging the tool along the top of the cabinet.

5.2.1 Resizing Wand

In addition to the resizing tool, a *Resizing Wand*, as shown in Figure 1, is also provided. This was added after experimenting with resizing kickboard components. As the kickboard is near the floor, using the resizing tool requires kneeling down, which quickly becomes uncomfortable. The resizing wand can be used instead of one half of the resizing tool, allowing resizing kickboard properties from a more comfortable stance. This type of tool could be extended to specifying a range of dimensions outside the user's reach, such as cabinets over benchtops and lighting fixtures on the ceiling, but these techniques require further evaluation against current out of arm reach VE interaction techniques.

6 Design Presets

While the cabinet layout process would be conducted exclusively by the architect, previewing and modifying kitchen designs is a process that involves the client. Rather



(a)

(b)

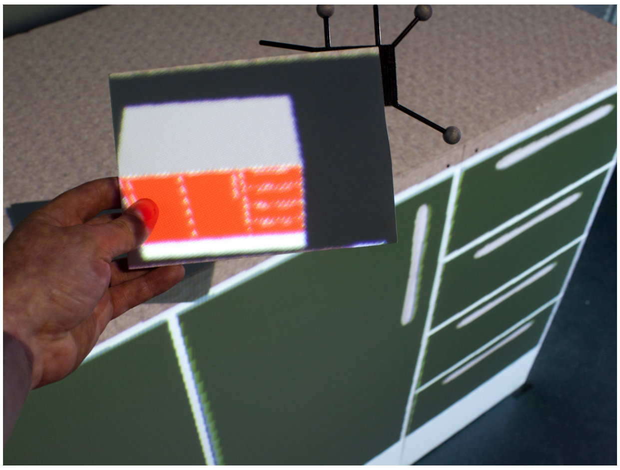
Figure 6: (a) A user resizes the height of a drawer, and (b) the thickness of the bench top.

than giving the client a 'blank canvas', they would instead be working with designs created earlier by the designer. These *Design Presets* can be a starting point from which the client can decide on their own design.

6.1 SAR Swatches

To facilitate loading and saving kitchen designs, BuildMyKitchen makes use of *SAR Swatches*. SAR Swatches are based on a metaphor adapted from the colour swatches used to choose paint colours. A SAR Swatch is a small board, and a 3D render of a kitchen design is projected onto the board. Clients can quickly evaluate different designs by holding several in their hands at once. SAR Swatches are also related to Tangible Bits (Ishii, 2008), in that the digital information is represented as a physical object.

The client can preview one of the designs simply by placing a swatch onto the cabinet. Doing this causes the kitchen design shown on the swatch to be loaded onto the cabinet blanks, and the client can experience the design at 1:1 scale on the physical surfaces. The other side of the swatch is used for saving designs. A new design can be saved to a swatch by placing it upside-down on one of the cabinets. The projected visualisation swatch will change, indicating the system is asking for confirmation (Figure 7(b)). To confirm the save operation, the user simply slides the swatch to the right. This motion will save the current kitchen design to the swatch, with the visualisation projected onto the other side of the swatch updating accordingly. The number of swatches is only limited by the number of physical swatches constructed for use; theoretically we could have thousand of swatches. They are uniquely identified by the tracking system.



(a)

(b)

Figure 7: (a) Comparing one design with another using a SAR Swatch, (b) saving a new design to a SAR Swatch by placing the swatch on the cabinet, and sliding right.

SAR Swatches provide a good example of the tradeoff between several physical tools and a single tool. Rather than individual swatches, designs could instead be previewed one at a time on a Personal Interaction Panel (PIP) (Szalavri and Gervautz, 1997) style device. The PIP could then be placed in the load area and the kitchen selected would be loaded. Alternatively, a tool for previewing could be removed entirely, and instead the user would only view designs on the full sized mock-up.

SAR Swatches were chosen because they have several advantages compared to the alternatives described above. Firstly, swatches allow the user to compare different designs simultaneously. This can improve their ability to choose a design. For example, when comparing two designs, the client may decide they prefer one design overall, but with the door handles from another design. This realisation would be difficult to come to if only one design could be seen at a time. Using several physical tools also makes it easier to quickly flick through several different designs in a more natural way than using a menu. Finally, SAR Swatches adapt a device already in use in interior design: the paint colour swatch. Kitchen swatches extend this to storing several aspects of the design on a single swatch.

SAR Swatches are interesting to consider because these tools are the first presented in this dissertation where the primary role is to hold information, not manipulate elements of the design. However, previewing a design on the swatch can be considered as one task, loading a design on the prototype as another, and saving the current design to the swatch as a third task. Each swatch has a single parameter: the kitchen design it is storing.

6.1.1 Generalising Swatches

Swatches solve the problem of loading and saving data in a SAR environment in a way that is specifically suited to the nature of SAR. Text input is difficult in large scale SAR environments because there is no logical location for a keyboard. However, for the kinds of industrial design applications discussed in this dissertation, the data to be saved is often visual in nature. Therefore, there is no need for the user to enter filenames. Instead, they can access the data in a visual manner. This is accomplished through the use of PVT. The physical SAR Swatches are treated like any other object in the SAR environment, and have virtual information projected onto them. The user is able to refer to and access data in a visual manner, and does not have to concern themselves with text entry. This simplifies the user interface by removing the keyboard, which may not have a logical location in a SAR environment.

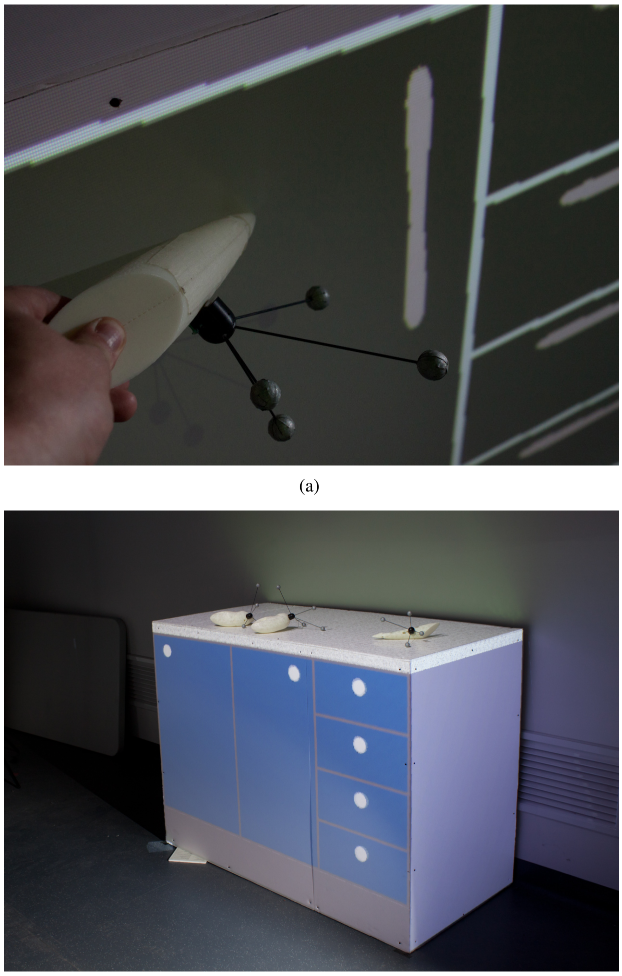
7 Modifying Finishes

The final use for BuildMyKitchen is to allow the client and architect to modify one of the presets in order to arrive at the final design. The client is able to select from several options to customise the following aspects of the design: materials (for surfaces such as benchtops, doors, kickboard, and splashback) and fixture designs (such as handles, power points, and taps).

There are of course infinite possible combinations of features. However, at this stage of the design, the number of choices is deliberately limited by the architect. Just like the kitchen presets, the designer makes available a set of options that will work together, based on their design experience. The client is able to customise their kitchen based on a smaller number of choices, rather than allowing every possible combination.

A *Magic Touch* metaphor is used for the modifying finishes tool, as this greatly simplifies the user interface. To select different options, the user taps the surface of the cabinet blank with the *Stylus*. The system cycles through the choices one at a time. The process is exactly the same for door handles and such. For these small items a selectable region is defined around the projected texture. In the case of a finish being replicated in more than one location as with a handle, the system changes all occurrences to match the selected style. For example there might be different sizes of handles for different sized doors; therefore, the change would modify the style of all the handles but with the correct sized handles for each occurrence. This mode of selection is acceptable because the number of choices is small. In the demonstration system between five and ten options are loaded for each of the changeable components. In addition, this task is fundamentally exploratory in nature. The client is interested in experimenting with different designs, and does not need to load specific finishes at this stage. However, if there were many choices, an alternate system, such as a pie menu displayed with a PIP could be employed.

Reusing the Stylus for BuildMyKitchen demonstrates how the complexity of a physical-virtual tool depends greatly on the application where it is used. Here, the tool is used for a single task: changing a property of the kitchen. This task has no attributes. While the single tool is used to change all the properties of the kitchen design, the active



(a)

(b)

Figure 8: (a) A user changes the colour of the cabinet doors using the stylus, (b) a finished kitchen design.

property is not an attribute of the tool. The property being changed simply depends on what the user touches with the Stylus.

8 Implementation

Our demonstration system is running on a standard desktop PC, with Nvidia Quadro FX 3800 graphics cards. The environment is lit using four NEC NP510W projectors, each with a resolution of 1280x800. All projectors were mounted on the ceiling, approximately 3.5 metres above the floor. One projector was centred on the cabinet blank, pointing almost vertically down. This projector primarily illuminated the cabinet top and SAR Swatches. The other three projectors were arranged in a hemisphere around the work area, pointing down at approximately 45°, giving a broad projection volume.

An eight camera Optitrack[2] tracking system is used to track the tools and cabinet blanks. BuildMyKitchen is written in C++, on top of our in house developed SAR software framework, using OpenGL.

9 Conclusions and Future Work

This paper has presented our work into using SAR for kitchen design and interior architecture. We have described our SAR design application, BuildMyKitchen,

[2]http://www.naturalpoint.com/optitrack/

and how it improves the design process for both architects and their clients. BuildMyKitchen features a PVT based user interface. We have discussed PVT-based tools and techniques developed for the application, and how they can be generalized for use in other application domains.

In the future we would like to extend BuildMyKitchen to allow functional analysis of kitchen layouts. For example, showing the available storage space once appliances, plumbing, and other components are in place. We could also optimize the functional layout by highlighting the predicted use of areas of the design, based on data on the clients actual usage patterns. In addition to these analysis tasks, we would also like to improve our graphics system. This would allow accurate previews of lighting configurations, based on models of actual light fittings.

References

Akaoka, E. and R. Vertegaal (2009). “DisplayObjects: prototyping functional physical interfaces on 3D styrofoam, paper or cardboard models”. In: *CHI 2009*. Boston, MA.

Bandyopadhyay, D., R. Raskar, and H. Fuchs (2001). “Dynamic Shader Lamps: Painting on Movable Objects”. In: *IEEE and ACM International Symposium on Mixed and Augmented Reality*, pp. 207–216.

Byung-Kuk Seo, Moon-Hyun Lee, Hanhoon Park, Jong-Il Park, and Young Soo Kim (2007). “Direct-Projected AR Based Interactive User Interface for Medical Surgery”. In: *Artificial Reality and Telexistence, 17th International Conference on*, pp. 105–112.

Fitzmaurice, G. W., H. Ishii, and W. A. S. Buxton (1995). “Bricks: Laying The Foundations for Graspable User interfaces”. In: *Proceedings of the SIGCHI conference on Human factors in computing systems*. Denver, Colorado, USA: ACM Press/Addison-Wesley Publishing Co., pp. 442–449.

Hare, J., S. Gill, G. Loudon, D. Ramduny-Ellis, and A. Dix (2009). “Physical Fidelity: Exploring the Importance of Physicality on Physical-Digital Conceptual Prototyping”. In: *Human-Computer Interaction INTERACT 2009*, pp. 217–230.

Ishii, H. (2008). “Tangible bits: beyond pixels”. In: *Proceedings of the 2nd international conference on Tangible and embedded interaction*. TEI ’08. New York, NY, USA: ACM, xvxxv.

Low, K.-L., G. Welch, A. Lastra, and H. Fuchs (2001). “Life-sized projector-based dioramas”. In: *VRST ’01: Proceedings of the ACM symposium on Virtual reality software and technology*. inproceedings. New York, NY, USA: ACM, pp. 93–101.

Marner, M. R. and B. H. Thomas (2010). “Tool Virtualization and Spatial Augmented Reality”. In: *Proceedings of the 20th International Conference on Artificial Reality and Telexistence*. Adelaide, South Australia.

Marner, M. R. and B. H. Thomas (2013). “Spatial Augmented Reality User Interface Techniques for Room Size Modeling Tasks”. In: *Poster Sessions: Proceedings of the IEEE Symposium on 3D User Interfaces*. Orlando, FL, USA.

Piper, B. and H. Ishii (2001). *CADcast: a Method for Projecting Spatially Referenced Procedural Instructions*. Tech. rep. MIT Media Lab.

Porter, S. R., M. R. Marner, R. T. Smith, J. E. Zucco, B. H. Thomas, and P. Schumacher (2010). “Spatial Augmented Reality for Interactive Rapid Prototyping”. In: *Proceedings of the 20th International Conference on Artificial Reality and Telexistence*. Adelaide, South Australia.

Raskar, R., G. Welch, and H. Fuchs (1998). “Spatially Augmented Reality”. In: *In First IEEE Workshop on Augmented Reality (IWAR98)*, 1120.

Raskar, R., G. Welch, K.-L. Low, and D. Bandyopadhyay (2001). “Shader Lamps: Animating Real Objects With Image-Based Illumination”. In: *Rendering Techniques 2001: Proceedings of the Eurographics*, pp. 89–102.

Schwerdtfeger, B., D. Pustka, A. Hofhauser, and G. Klinker (2008). “Using laser projectors for augmented reality”. In: *Proceedings of the 2008 ACM symposium on Virtual reality software and technology*. Bordeaux, France: ACM, pp. 134–137.

Suganuma, A., Y. Ogata, A. Shimada, D. Arita, and R.-i. Taniguchi (2008). “Billiard instruction system for beginners with a projector-camera system”. In: *Proceedings of the 2008 International Conference on Advances in Computer Entertainment Technology*. Yokohama, Japan: ACM, pp. 3–8.

Szalavri, Z. and M. Gervautz (1997). “The personal interaction panel - a Two handed interface for augmented reality”. In: *EUROGRAPHICS 97*. Budapest, Hungary, pp. 335–346.

Thomas, B. H., G. S. Von Itzstein, R. Vernik, S. R. Porter, M. R. Marner, R. T. Smith, M. Broecker, and B. Close (2011). “Spatial Augmented Reality Support for Design of Complex Physical Environments”. In: *Workshop on Interdisciplinary Approaches to Pervasive Computing Design*.

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

Verlinden, J., A. d. Smit, A. Peeters, and M. v. Gelderen (2003). “Development of a flexible augmented prototyping system”. In: *Journal of WSCG*.