

# DeskCube: Using Physical Zones to Select and Control Combinations of 3D Navigation Operations

Michael Glueck, Sean Anderson, Azam Khan

Autodesk Research  
210 King Street East,  
Toronto, ON, Canada M5A 1J7  
{firstname.lastname}@autodesk.com

**Keywords:** Augmented reality, Input device, 3D navigation.

## Abstract

We present the DeskCube, a new passive input device, together with a space-division scheme using physical above-the-surface zones to select and control the desired 3D navigation operations that gives users simple scene-in-hand control over the virtual 3D world.

## 1. INTRODUCTION

Navigating around objects of interest in 3D environments can be difficult for novice users. In addition, most 3D software packages require two-handed input to control virtual cameras in three dimensions. The non-dominant hand uses keyboard hotkeys to switch between standard interaction modes, such as pan, zoom, and orbit. The dominant hand uses the mouse to control the selected operation. Combining multiple degrees-of-freedom (DOF) into a single input device could reduce user effort to control the viewpoint of a 3D model, but it has been found [Hinckley 1994] that users generally have difficulty in controlling so many dimensions simultaneously. To address this problem, our design includes a high DOF input device that uses physical above-the-surface zones to select and control the desired operations. We present the DeskCube, a new passive input device, together with a space-division scheme that allows camera controls to be manipulated to give users scene-in-hand control over the virtual 3D world.

## 2. RELATED WORK

Augmented reality has long been used to integrate the physical and digital realms. ARTag [Fiala 2005] is a library that uses computer vision to translate real-world camera coordinates to virtual camera coordinates using fiducial markers, essentially creating a mapping from the real world to the digital world.

Several multiple degree-of-freedom devices have been proposed to help users interact with 3D scenes. Recently, Woods et al. [2003] used the ARToolkit [Kato 2003], an earlier system similar to ARTag, to create a low-cost six-degree-of-freedom “mouse” using the information extracted

from the toolkit to position a camera in 3D space. This gave users a camera-in hand interaction metaphor.



**Figure 1.** DeskCube configuration at a demonstration booth.

## 3. DESIGN CONSIDERATIONS

The design of the DeskCube was inspired as a tangible analogue to the ViewCube [Khan 2008], an interface widget to aid a user in orientation awareness in a 3D scene. Each side of the DeskCube would represent one of the six canonical scene views: top, bottom, left, right, front, and back. The DeskCube is intended to rest on the desk, in front of a web camera, while not in use (see Figure 1).

When designing the DeskCube, several considerations needed to be met. The size of the DeskCube needed to be the right size to be comfortably grasped. We also discovered that the weight of the tangible device was a factor in how users manipulated early prototypes. We noticed early paper prototypes, which were lighter, would be manipulated very quickly and irregularly, and that additional weight from heavier materials, such as plastic, encouraged users to interact with the DeskCube more smoothly and consistently. Since the ARTag library produces better results when motion is less disruptive, this helped improve interaction between the DeskCube and the 3D software.

Since the DeskCube was expected to be handled by a user directly, we needed to make sure that at least one fiducial marker was visible at all times. There is a tradeoff between the number of markers one can present on each face and the size of the markers. While Fiala [2005] recommends arrays of markers for improved performance, the decreased size of the markers also decreases correct recognition of targets, especially when they are blurred by motion. We found that four markers per face sufficiently prevented errors due to markers being occluded by a user's fingers.

#### 4. 3D NAVIGATION SCHEMES

Three 3D navigation schemes were investigated, allowing users to access different combinations of panning, zooming, and orbiting operations. To support these operations, we partitioned the space around the DeskCube into *action zones*, allowing users to access different navigation operations by moving the DeskCube. In all schemes, a central zone was designated as a *static zone*, where no navigation operations were activated.

Two different methods of transferring user input to navigation operations were used: position control and rate control. Position control directly maps user input to the movement in the 3D scene, such as moving the mouse left to pan the scene left. Rate control, on the other hand, is a mapping where the position of the input maps to the velocity of the movement in the 3D scene, for example when pushing a joystick left to pan the scene left [Zhai 1995].

The DeskCube uses position control for orientation changing operations, such as orbit and look. As the DeskCube is rotated, the change in degrees about its axis is mapped directly to the degree change in the scene camera. Rate control is used for operations such as panning and zooming. Leaving a static zone and moving the DeskCube into an active zone acts as a clutch to engage the operation associated with that zone. The farther into the zone the DeskCube is moved the more quickly the operation is applied. The operation can be stopped by moving the DeskCube back into the static zone.

#### 4.1. Orbiting with Panning and Zooming

In our first navigation scheme, we wanted to support all three standard 3D navigation methods: panning, zooming, and orbiting. We used the scene-in-hand [Ware and Osborne 1990] interaction metaphor to guide our design of the DeskCube navigation, meaning that if the user was viewing the DeskCube from the top-left, then they should also be viewing the 3D scene from the top-left on the monitor.

We divided the space into three primary zones, which would control zooming behavior, and subsequently divided the middle zone into nine secondary zones to control panning direction. The middle area of the primary and secondary zones was a static zone where neither zooming nor panning occurred. Rotating the DeskCube in any of these regions accessed the orbiting operation simultaneously to either zooming or panning (see Figure 2).

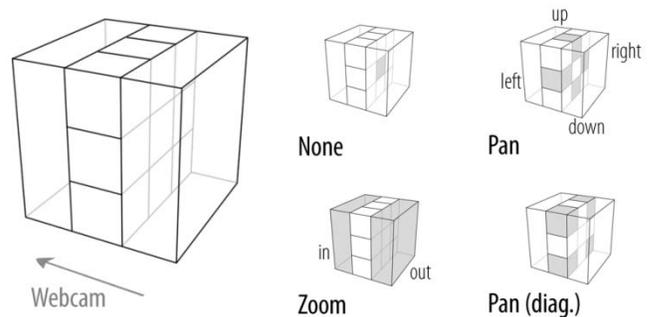


Figure 2. Schematic of modal operations for first scheme. Note that orbiting was always available in this scheme.

The main drawback to supporting all of these interaction techniques was that the DeskCube would need to be held in the static zone if the user did not want zooming or panning to occur. In order to allow the user to put the DeskCube down without effecting the camera position, we needed to provide a method of modally turning on and off the panning functionality. This was achieved through a keyboard hotkey. Unfortunately, this moved away from our original DeskCube and mouse interaction paradigm. However, we found that without some kind of modal switch, we would only be able to support a subset of these operations.

#### 4.2. Looking with Zooming and Limited Panning / Modal Orbiting

In our second navigation scheme, we instead approached the interaction technique in terms of egocentric and exocentric usage. We defined the table top area as an exploratory zone and the area above the table as an orientation zone. The table top zone was divided into nine zones that allowed users to zoom in and out, and pan left and right, as well as a combination of zooming and panning operations. Changing the orientation of the DeskCube in any of these zones would result in a looking operation either

to the left or right. If the DeskCube was lifted, only then would orbiting be available, while panning and zooming would be disabled altogether. This scheme allows users to explore a space, such as the interior of a room, by simply moving the DeskCube around the surface of their table top, and reorient themselves globally by lifting the DeskCube from the table top (see Figure 3).

While we felt this was an interesting interaction technique for the exploration of the interior of spaces, it removed the DeskCube from the same conceptual space as the ViewCube, which was designed purely for the exocentric orientation around objects from the outside.

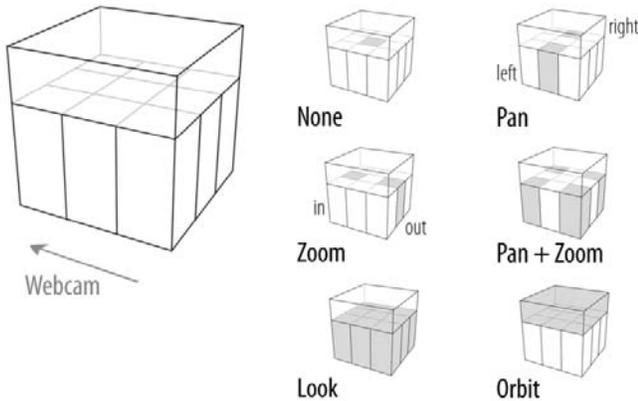


Figure 3. Schematic of modal operations for second scheme.

### 4.3. Orbiting with Zooming

In our third navigation scheme, we simplified the interaction technique by removing panning altogether. We also returned to the scene-in-hand metaphor. In addition, we drew inspiration from another feature of the ViewCube: selection sensitivity. In this case, if an object is selected, the ViewCube would represent the orientation of that object, which could be different from the global scene. This actually extends the metaphor to an object-in-hand technique. Selection and deselection of objects was performed by mouse clicks and used to navigate between selected objects. For example, in the city scene (see Figure 1), clicking on a building would reposition the user with respect to that object. If no objects were selected, then by default all scene geometry would be selected.

When a new object was selected, the camera position would be animated toward to the selected object and the camera would be turned according to the current DeskCube orientation. The camera would also be placed a fixed distance away from the object initially, as a factor of the bounding radius of the object. The user could then zoom in or zoom out from the object to achieve different views (see Figure 4).

This scheme provided the user with constrained interaction with objects found in the scene. Conceptually, it can be thought of as moving along the surface of a sphere

centered on the selected object (turntable), roll, and changing the size of the sphere interactively; offering users four degrees-of-freedom.

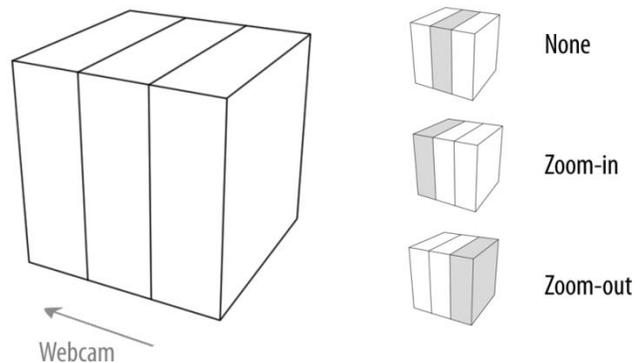


Figure 4. Schematic of modal operations for third scheme. Note that orbiting was always available in this scheme.

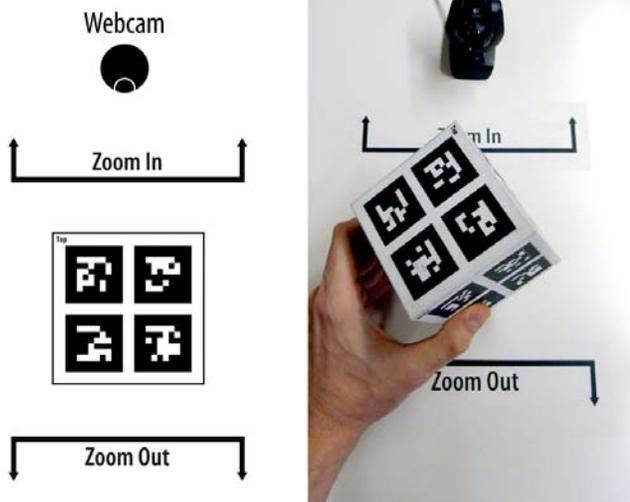
## 5. IMPLEMENTATION

The DeskCube system was implemented using the third interaction scheme. The ARTag library was used to map the DeskCube position and orientation from real-world to digital coordinates, and written as a plug-in to an existing 3D graphics engine. We used a Logitech QuickCam Pro 9000 to record the DeskCube. To ensure consistent readings, we disabled most of the automatic features (focus, white balance, exposure), in favor of a static manual calibration. The final DeskCube prototype measured 7.5cm along each side and was a hollow plastic cube with ARTag fiducial markers attached via stickers to each face.

The DeskCube and web camera system was set up on the left-hand side of the computer desk so that a right-handed user would be able to use the mouse and DeskCube simultaneously: controlling orientation and zoom with the left hand, while being able to select objects for inspection using the mouse with the right hand. To provide some stability to the zooming feature, we developed a zoned control system, whereby stickers attached to the desk would indicate tolerance zones where zooming in and zooming out would begin (see Figure 5). The zoom in region was placed 15cm away from the web camera, and the zoom out region was placed 30cm away from the web camera.

In the area between these two lines, no zooming would occur. Moving the DeskCube into the zoom-in region and zoom-out regions would perform those operations, respectively. The zooming speed was determined by the proximity of the DeskCube to the camera, increasing exponentially as it approached the webcam. Zooming was limited to the bounding radius of the object to prevent a user from entering the selected object on the near side, and a factor ten times the bounding radius on the far side.

The 3D graphics application displayed a model of downtown Toronto, Canada. Individual buildings were selectable (see Figure 1).



**Figure 5.** Layout of DeskCube regions on desk and position of webcam: schematic (left) and actual (right).

We used two methods to smooth out the DeskCube interaction: (a) averaging across samples from the ARTag library and (b) detecting when a DeskCube face was oriented orthogonally to the web camera.

Samples were read from the web camera at 30 frames per second and were averaged across the last 5 samples recorded. This was sufficient to smooth out the interaction and produced minimal lag when using the DeskCube.

An angular threshold was set for when the cube approached an orthogonal orientation. Once the cube was within 15 degrees of a face towards the web camera, the orientation of the user in the viewing application would start to pull towards the orthogonal orientation, like a magnet. This snapping behaviour allowed for more precise interaction by users to view canonical view directions, and also removed any jitter from perpendicular views of the cube, where small angular differences in samples from one web camera frame to the next could result in oscillating orientation changes in the scene.

## 6. DISCUSSION

A number of considerations for the design space of the DeskCube have been shown. Each one has strengths in some aspects but weaknesses in others. While we have implemented these scenarios and developed preferences for certain combinations, external testing is still needed to discover user performance and preference. To this end, we have conducted an informal pilot study to gather initial user responses and to observe the levels of difficulty in learning and using the DeskCube (see Figure 1). We installed the system at a public event held for users of 3D design

software. At times we actively demonstrated the intent and use of the system while at other times we observed passers-by attempting to use the system. Approximately 200 people used the DeskCube system over a period of three days. Generally, users quickly understood the operation of the DeskCube and were able to carefully control the virtual camera. Furthermore, users were excited about this form of 3D control and several suggested that the system would be well suited to the presentation of 3D content for their clients. We were told that the use case of the DeskCube as a navigation aid while giving a presentation was appealing because it was easy enough to operate that the clients themselves could “take control.” As integrated webcams in laptops become ubiquitous, the DeskCube would be readily usable without additional equipment set-up. An architect could present a concept model of a building to a client, and use the DeskCube to offer different exterior views of the structure, without using the mouse or keyboard. This could facilitate a more natural presentation style and allow the architect to engage the client directly, without the distraction of interacting directly with the computer.

## 7. FUTURE WORK & CONCLUSION

Future research will focus on the design of a formal experiment to evaluate the design space of the DeskCube. We expect that further insights will be discovered leading to both improvements and an expanded design space. Also, a pilot could be undertaken to explore the DeskCube as a 3D presentation controller. In summary, we have shown the DeskCube, a new interaction device and space-division scheme to give users simple control over an urban 3D scene.

## References

- Fiala, M. 2005. ARTag, a fiducial marker system using digital techniques. In Proc. CVPR'05, Vol. 2, 590-596.
- Hinckley, K., Pausch, R., Goble, J.C., Kassell, N.F. 1994. A survey of design issues in spatial input. ACM UIST, 213- 222.
- Kato, H., Billingham, M., Poupyrev, I., Imamoto, K., Tachibana, K. 2000. Virtual object manipulation on a table-top AR environment. In Proc. ISAR'00, 111-119.
- Khan, A., Mordatch, I., Fitzmaurice, G., Matejka, J., and Kurtenbach, G. 2008. ViewCube: a 3D orientation indicator and controller. In Proc. I3D'08, 17-25.
- Ware, C., and Osbourne, S. 1990. Exploration and virtual camera control in virtual three dimensional environments. In Proc. I3D'90. 175-183.
- Woods, E., Mason, P., and Billingham, M. 2003. MagicMouse: an inexpensive 6-degree-of-freedom mouse. In Proc. GRAPHITE '03, 285-286.
- Zhai, S. 1995. Human performance in six degree of freedom input control. Ph.D. Thesis, University of Toronto.