
CAPTIVE: A Cube with Augmented Physical Tools

Arpan Chakraborty

North Carolina State University
Raleigh, NC 27603 USA
achakra@ncsu.edu

Ryan Gross

North Carolina State University
Raleigh, NC 27603 USA
rbgross@ncsu.edu

Shea McIntee

North Carolina State University
Raleigh, NC 27603 USA
ssmcinte@ncsu.edu

Kyung Wha Hong

North Carolina State University
Raleigh, NC 27603 USA
khong@ncsu.edu

Jae Yeol Lee

Chonnam National University
Gwangju, South Korea
jaeyeol@jnu.ac.kr

Robert St. Amant

North Carolina State University
Raleigh, NC 27603 USA
stamant@ncsu.edu

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the Owner/Author.

Copyright is held by the owner/author(s).

CHI 2014, Apr 26 - May 01 2014, Toronto, ON, Canada
ACM 978-1-4503-2474-8/14/04.
<http://dx.doi.org/10.1145/2559206.2581340>

Abstract

This paper describes a tangible 3D user interface called CAPTIVE, a Cube with Augmented Physical Tools, for exploration of 3D information. The design of CAPTIVE is founded on the concept of tool use, in which handheld tool objects are used to modify the properties or appearance of target objects. The user of CAPTIVE holds a physical wireframe cube that contains virtual objects in one hand, in the other a pointing device, its tip visually augmented to reflect its function as a tool. On the display the user watches the immediate, direct effects of actions with the tool. In the current prototype, routines for handling cube manipulation and an augmented haptic pointing device have been separately implemented, but integration and refinement remain to be done. In this paper we describe our vision of the system and preliminary testing carried out to date.

Author Keywords

Augmented reality; tangible computing; 3D user interfaces; tool-based interaction.

ACM Classification Keywords

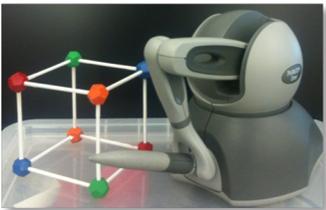
H.5.1. Information interfaces and presentation (e.g., HCI): Artificial, augmented, and virtual realities.

CAPTIVE hardware



CAPTIVE consists of the following devices connected to a computer:

- Toshiba 14-inch ultra-portable mobile LCD monitor, with 1366×768 resolution, mounted on a stand.
- Logitech C920 camera with full HD 1080p video at 30 fps, 20-step autofocus and Carl Zeiss optics.
- GeoMagic Touch (formerly Sensable Phantom Omni) haptic pointing device with 6 degrees of freedom and force feedback, pictured below.



Introduction

3D and higher-dimensional information has been of interest in visualization research for decades, and some consensus has been reached on the basic categories of abstract visualization tasks: information can be selected, explored, reconfigured, recoded, abstracted, filtered and connected [10]. Interaction techniques for accomplishing these tasks, however, receive “little emphasis in visualization research” [3].

We are developing a system called CAPTIVE, a Cube with Augmented Physical Tools, for exploration of 3D information. The current prototype is configured as a desktop video see-through augmented reality system. User manipulations occur behind a monoscopic display, with the camera mounted to the back of the display and focused on the user's hands. A stereo configuration, with interaction in front of the display, is under development; the focus of this paper is on the monoscopic configuration.

The cube in CAPTIVE, unlike the solid cubes often used in AR systems, is a physical wireframe 10 cm on each side, with white edges and vertices colored red, orange, green, and blue. CAPTIVE software builds on OpenCV and OpenGL. OpenCV's image processing routines are used on video frames to identify vertex locations by color; these 2D coordinates are converted to 3D locations using known point correspondences and camera parameters, such as focal length and field of view. The cube's pose is then filtered to smooth out detection noise and to resolve ambiguities. The resulting transformation enables the system to superimpose virtual objects on the display to appear within the cube. The software currently runs in close to

real time for detecting the position and orientation of the cube.

Presently, only cube-tracking has been implemented and performance-tested. We envision rich interaction with the system via a collection of additional tools. One usage scenario under active development is selection using a pointing tool. Users hold the cube in one hand and a pointing device in the other; when the tip of the pointing device is brought into the display region, it is visually augmented to reflect its function as a tool. The tool can extend into the cube, to interact with the data projected inside. A magic lens [1] and a cutting plane [6] are among other planned tools.

Tool use

CAPTIVE is founded on the concept of tool use. Tool use is a central concept in HCI; in the technical literature, almost any function of an interactive system—even the entire system—can be described as a tool. We adopt a more restrictive definition, based on our work on animal tool use [7]:

*Tool use is the exertion of control over a freely manipulable object (the tool) with the goal of altering the properties of another object or medium (the target) via a dynamic mechanical interaction, or mediating the flow of information between the tool user and the environment.*¹

CAPTIVE features several aspects of tool use, common in physical manipulation tasks but unusual in

¹ The “freely manipulable” criterion eliminates immobile surfaces as tools, and the “mechanical” criterion eliminates communicative actions such as gestures.

System operation

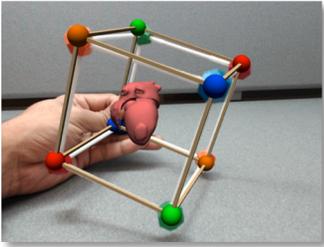


Figure 1. Augmenting a model of the human heart within the cube.
Model source: *BodyParts3D*
<http://lifesciencedb.jp/bp3d/>

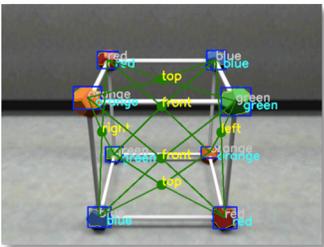


Figure 2. Intermediate results showing detected markers and faces

interactive computing. It supports asymmetrical bimanual interaction, with task components distributed to the hands to support coordinated action. Actions are spatially collocated with their visual effects, in a control/display isospace, unlike some existing systems [2][4]. Visual constraints are relevant: the entire hand/tool/target configuration can be translated and rotated to make relevant information visible. Further, visual attention focuses on the endpoint of the tool and its effect on the target. This has implications for how the tool extends the internal representation of the body. Mechanical constraints providing passive haptic feedback are another critical factor, with the right hand potentially being braced against the left and with the tabletop supporting one hand or both (bimanual and environmental stabilization). Actions under these constraints produce proprioceptive feedback as well as visual feedback: intrinsic proprioceptive information is additively combined with extrinsic visual information to minimize error.

We believe that 3D interaction may be improved by the combination of three specific aspects of tool use: (a) spatial collocation of user actions and their effects, (b) physical constraints for bimanual/environmental stabilization, and (c) a greater role for proprioception.

A hypothetical usage scenario

Imagine the use of CAPTIVE by Sally, a prototypical user, in an introductory class in anatomy or medical diagnostics, for the task of examining a 3D heart model.

Sally picks up the cube containing the heart and orients it, as shown in **Figure 1**. Some points on the model are visually highlighted; Sally picks up the pointer and

activates the *probe* mode. She touches the augmented tip of the probe to one of the points, and a label appears with information about that region.

Sally decides to look at a specific region in more detail. She switches to the *magnifying glass*, which changes the augmented tip of the pointer to a circular region and allows her to see a higher magnification.

To view a slice through the heart model, Sally uses a *cutting plane*, which extends tip of the pointer in two dimensions. She rotates the cube to an orientation appropriate for the plane, and she adjusts the angle of the pointer to study the desired cross-section.

Of these actions, only the cube manipulation has been implemented. In the remainder of this paper we describe how cube manipulation works in CAPTIVE.

Cube tracking

The motivation for using a hollow cube lies in its ability to act as a container and reference frame for exploration of 3-dimensional data within a constrained volume. This rules out the use of planar markers on cube faces that would block access to the interior space. We employ colored vertices as point markers that can be detected and used to track the cube's pose through arbitrary movements. At any given time, only a few vertices need to be visible for a successful detection. This makes the system robust to partial occlusion and allows the cube to be rotated without any angle limitations. The 8 cube vertices are assigned 4 colors with an ordering that ensures no face has any repeated colors. This reduces face ambiguity. The detection and tracking procedure applied to each incoming video frame (image) is outlined below.

Summary of results

The following trends can be gleaned from the results:

- In general, cube pose error increases with faster frame rates (i.e. faster movement).
- Orientation error when *translating* is very low (~ 0.06 rad = 3.4 deg); position error when *rotating* is also low (~ 2.6 cm) – this is expected, but still an important validation.
- Position error when translating the cube climbs up (4.3 cm avg., 6.8 cm max.); similarly, orientation error when rotating is high (0.4 rad = 22.9 deg avg., 0.66 rad = 37.8 deg max.).
- The system remains fairly usable up to 30 fps, i.e. movement speeds of about 8 cm/s, 1.57 rad/s.

Note: By playing back input videos at different frame rates, we are able to simulate translational speeds of 2.67 - 13.33 cm/s and rotational speeds of 0.52 - 2.62 rad/s.

Algorithm: TrackCube(image)

1. Find colored markers in input image.
2. For each 4-element ordered subset (sequence) of markers with distinct colors, check if it matches a known face ordering.
3. For each valid face found in step 2, find a cube transform (translation, rotation) that matches the observed 2D marker locations with known 3D vertex positions using Levenberg-Marquardt optimization².
4. If a single valid transform is found, return that as current cube pose.
5. If two or more valid transforms are found, first reject any outliers. From the remaining (consistent) transforms, return the one closest to the mean. If no consistent transforms remain (ambiguous pose), return one of the original valid transforms that is closest to the last known cube pose.
6. If no valid transforms were found (not enough markers, no valid faces, etc.), retain last known cube pose but flag as tracking failure.

Figure 2 illustrates the procedure showing detected colored markers and estimated faces. The computed cube pose is smoothed to reduce detection noise and used to display a virtual cube and any other augmented object within it, as shown earlier in **Figure 1**.

Performance evaluation

In order to characterize the space of operations possible with CAPTIVE, we first need to measure the limits it imposes on interactions. We are primarily interested in the maximum speed of movement that does not result in a loss of tracking ability. For this purpose we use a virtual cube modeled with realistic

lighting and shadows to generate recorded videos with known cube positions and velocities³. For comparing the observed pose sequence with ground truth we use Dynamic Time Warping (DTW). DTW aligns the cube position values (in cm.) and orientation values (in radians) with corresponding ground truth to give us a cumulative error measure, using simple Euclidean distance as basis. This is normalized so that the resultant error values are roughly in original units, which makes performance comparison more meaningful (note: due to the warping nature of DTW, exact units cannot be deduced).

DTW results are shown in **Table 1** (position errors) and **Table 2** (orientation errors) for the different input videos used. These videos individually cover translation and rotation in all 3 axes, at 5 different speeds each. In the translation videos, the cube moves 8 cm along each indicated axis, and in the rotation videos the cube turns 360 degrees on the respective axis.

Motion	Axis	Video frame rate (fps)					Mean
		10	20	30	40	50	
translate	X	2.18	2.93	3.54	5.41	6.65	4.14
	Y	2.16	3.38	3.91	4.95	5.42	3.96
	Z	2.53	3.75	5.38	5.79	6.80	4.85
rotate	X	1.99	2.08	2.49	2.83	3.48	2.57
	Y	2.08	2.29	2.60	2.86	3.26	2.62
	Z	1.94	2.21	2.90	3.09	3.37	2.70

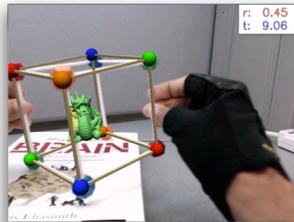
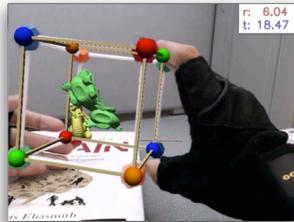
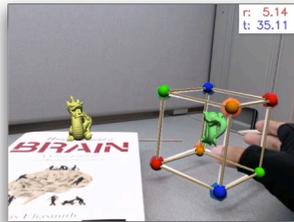
Table 1. DTW position errors (cm.)

² As implemented in OpenCV's solvePnP() routine.

³ Virtual cube animations created using Blender – an open source 3D modeling program available at www.blender.org

Docking a dragon

Informal testing with the CAPTIVE cube: the user matches a target object's position and orientation.



Model source: Stanford University Computer Graphics Laboratory

Motion	Axis	Video frame rate (fps)					Mean
		10	20	30	40	50	
translate	X	0.04	0.05	0.06	0.07	0.09	0.06
	Y	0.02	0.03	0.04	0.05	0.06	0.04
	Z	0.04	0.06	0.09	0.10	0.11	0.08
rotate	X	0.04	0.05	0.06	0.07	0.12	0.07
	Y	0.49	0.57	0.55	0.56	0.53	0.54
	Z	0.48	0.56	0.63	0.66	0.63	0.59

Table 2. DTW orientation errors (radians)

The main reason for high orientation error (esp. at high rotational speeds) is that face visibility changes frequently. A comparison between the observed cube orientation and ground truth is shown in **Figure 5**. The noticeable lag is primarily due to pose smoothing and is more pronounced at higher speeds. Although it contributes to cumulative error, it does not affect interaction too much once users get accustomed to it; turning off smoothing results in jitter caused by detection noise which is considerably more distracting.

Planned interaction tasks

Evaluation using recorded virtual videos, as above, tests the system in an isolated, controlled environment. In order to see how well CAPTIVE performs in a more

realistic setting, we have designed a set of demonstrative tasks for users to carry out using the system. These tasks encompass the broader function areas that we wish to enable and test overall usability. Formal experimentation on them remains to be done.

Docking a dragon

This 6 DoF docking task [11] consists of moving a dragon model augmented within the cube (the *cursor*) to match the position and orientation of an identical static model of a different color (the *target*) placed randomly in the scene that does not move with the cube. We are interested in recording the time to completion as well as cumulative translations and rotations performed to get a measure of efficiency.

Tracing a radial tree branch

Here the user is presented with two radial trees [9], with their root nodes colored differently, displayed within the cube with considerable visual overlap. The task requires the user to trace a highlighted leaf node back to the root of the tree it belongs to. Fine rotational manipulation is critical for success.

Point-object selection

This is a simple point-object selection task requiring the use of a pointing device (GeoMagic Touch) in addition

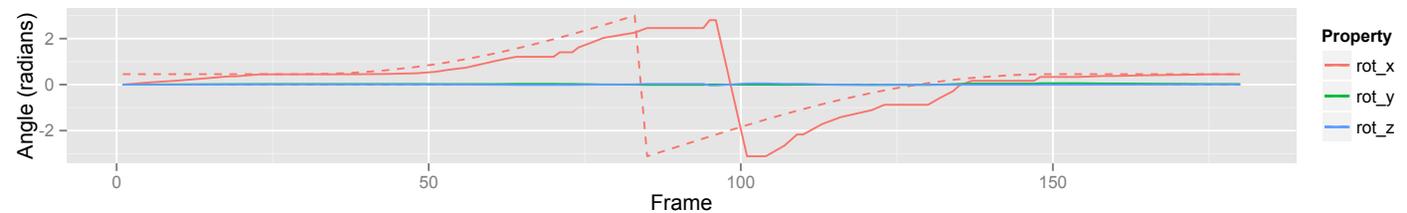


Figure 3. Comparing observed cube orientation (solid line) and ground truth (dashed line) during X-axis rotation

to the cube⁴. A set of small spheres representing some, possibly random, distribution is displayed within the cube. Each trial consists of a random sphere being highlighted and the user having to select it using the pointing device. The control condition will require the cube to remain static, while the bimanual condition will allow users to move both the cube and pointing device.

To ascertain the advantage offered by CAPTIVE, we plan to compare user performance against identical tasks carried out with standard input (mouse, keyboard) and display devices. In addition to quantitative measures of performance, the overall usability of the system and immersiveness will be gauged using a post-completion survey.

Conclusion

CAPTIVE represents a promising direction for 3D interaction, in its combination of ideas from tangible computing, augmented reality, and high-DOF input. The interactions that CAPTIVE is designed to support are a natural match for our experience and skills in using tools. We envision its use, in the long term, as part of a natural user interface for exploring 3D data.

References

- [1] Bier, E. A., Stone, M. C., Pier, K., Buxton, W., and DeRose, T. D. Toolglass and magic lenses: the see-through interface. In *Proc. Computer Graphics and Interactive Techniques*, ACM Press (1993), 73–80.
- [2] Block, F., Schmidt, A., Villar, N., Gellersen, H.W., Towards a Playful User Interface for Home

Entertainment Systems. In *Proc. of EUSAI'04*, Springer Verlag (2004), 207–217.

- [3] Elmqvist, N., Moere, A. V., Jetter, H.-C., Cernea, D., Reiterer, H., and Jankun-Kelly, T. Fluid interaction for information visualization. *Information Visualization* 10, 4 (2011), 327–340.
- [4] Frohlich, B., Plate, J., The cubic mouse: a new device for three-dimensional input. In *Proc. CHI 2000*, ACM Press (2000), 526–531.
- [5] Guiard, Y. Asymmetric division of labor in human skilled bimanual action: The kinematic chain as a model. *Journal of Motor Behavior* 19 (1987), 486–517.
- [6] Hinckley, K., Pausch, R., Goble, J. C., and Kassell, N. F. Passive real-world interface props for neurosurgical visualization. In *Proc. CHI 1994*, ACM Press (1994), 452–458.
- [7] St. Amant, R., and Horton, T. E. Revisiting the definition of tool use. *Animal Behaviour* 74, 4 (2008), 1199–1208.
- [8] Sarlegna, F. R., and Sainburg, R. L. The effect of target modality on visual and proprioceptive contributions to the control of movement distance. *Experimental Brain Research* 176, 2 (2007), 267–280.
- [9] Stavness, I., Lam, B. and Fels, S. pCube: a perspective-corrected handheld cubic display. In *Proc. CHI 2010*, ACM Press (2010), 1381–1390.
- [10] Yi, J. S., Kang, Y. a., Stasko, J. T., and Jacko, J. A. Toward a deeper understanding of the role of interaction in information visualization. *IEEE Trans. Vis. Comp. Graphics* 13, 6 (2007), 1224–1231.
- [11] Zhai, S., and Milgram, P. Quantifying coordination in multiple DOF movement and its application to evaluating 6 DOF input devices. In *Proc. CHI 1998*, ACM Press (1998), 320–327.

⁴ This commercial device was chosen to reduce sources of localization error; we plan to replace this with an untethered pen tracked using a similar scheme as the cube.