

# Haptic Display of Visual Images \*

Yunling Shi and Dinesh K. Pai  
Department of Computer Science  
University of British Columbia  
Vancouver, B.C. V6T 1Z4  
Canada  
*e-mail:* {ylshi|pai}@cs.ubc.ca

## Abstract

*We describe an implemented system for touching 3D objects depicted in visual images using a two dimensional, force-reflecting haptic interface. The system constructs 3D geometric models from real 2D stereo images. The force feedback at each position is computed to provide the sensation of moving a small ball over the surface of the 3D object.*

## 1 Introduction

We describe a system that allows users to feel a three dimensional shape, using stereo camera images of the shape as input and a haptic interface for force display. Such a system has several possible applications in virtual reality and telepresence. For instance, using only stereo cameras mounted on a mobile robot, it will be possible to move around in a real environment and feel the objects found there with our hands. It can also serve to display visual information to the visually impaired.

Haptic interfaces are devices for synthesizing mechanical impedances; they enable users to interact with 3-D objects in a virtual environment and feel the resulting forces. A typical system with a haptic interface consists of a real-time simulation of a virtual environment and a manipulandum (handle) which serves as the interface between the user and the simulation. The operator grasps the manipulandum and moves it in the workspace. Based on feedback from the sensors, the simulation calculates forces to output with the actuators. These forces are felt by the operator through

the manipulandum, making it seem to the operator as if (s)he is actually interacting with the physical environment. Combined with visual and auditory displays, haptic interfaces give users a powerful feeling of presence in the virtual world.

The geometric models of objects in current virtual environments are typically simplistic, artificially created CAD models. These models are expensive, require a long time to construct, and lack fine details as well. In our work, we construct the geometric models using real stereo images. In related work, [PR96] describes a system for interacting with multiresolution curves derived from single images. Our approach can be considered a type of virtualized reality [Kan95], because it starts with a real world and virtualizes it.

The rest of the paper is organized as follows. In Section 2, we describe the stereo reconstruction algorithm we use, based on [IB94]. This approach handles large occlusions well. In Section 3, we describe the computation of the force feedback based on user interactions. The results are described briefly in Section 4.

## 2 Stereo Reconstruction

We use correlation-based stereo for reconstructing the 3D shape. Typical methods such as [Fua93] keep reliable disparity information (which is inversely proportional to depth) derived from correlation between the left and right images, and expand this set by filling and smoothing; these algorithms tend to have mismatches when there are plain surfaces (such as most of the background) and repeated objects along the epipolar line. Our approach is based on [IB94] and [Cox92]. [Cox92] describes an algorithm that optimizes a maximum likelihood cost function, subject to ordering and occlusion constraints. Intille and Bobick [IB94] assigned costs to different disparities of each pixel based on disparity space image (DSI) and occurrence of occlusions.

---

\*Supported in part by NSERC, BC Advanced Systems Institute, and IRIS NCE. The authors would like to thank V. Hayward, J. J. Little, D. G. Lowe for their help.

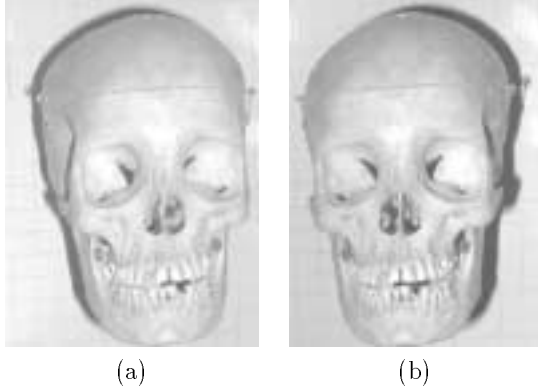


Figure 1. Stereo Image Pair

## 2.1 Method Outline

The Disparity Space Image (DSI) is a data structure for stereo matching that can be generated in the following way: select the  $i^{th}$  scanline of the left image and right image, respectively. The difference between windows along the scanline of left and right image is

$$W^L(x, d, w_x, w_y, c_x, c_y) = \sum_{s=-c_y}^{(w_y-c_y)} \sum_{t=-c_x}^{(w_x-c_x)} [(I_R(x+t, i+s) - M^L(x, i)) - (I_L(x+d+t, i+s) - M^R(x+d, i))]^2$$

Where  $w_x \times w_y$  is the size of the window,  $(c_x, c_y)$  is the location of the center of the window, and  $M^L, M^R$  are the means of the windows. In a pair of real uncropped stereo images,  $d$  is in the range of  $[0, d_{\max}]$ , where  $d_{\max}$  is the maximum possible disparity.  $d_{\max}$  will not be less than zero because the object in the left image is always to the right of which in the right image [IB94].

The DSI of  $i^{th}$  scanline in the left image, for example, is generated by

$$DSI_i^L(x, d, w_x, w_y) = \begin{cases} \min W_i^L(x, d, w_x, w_y, c_x, c_y) & 0 \leq x-d < N \\ NaN & \text{otherwise} \end{cases}$$

Figure 1<sup>1</sup> shows the pair of test images. They have a large disparity (maximum disparity of 60) comparing with their size (240×360).

Figure 2(a) shows the left disparity space image for the 75<sup>th</sup> scanline (the forehead) of the test image pair shown in Figure 1. The darker parts of the DSI correspond to better matches. When a textured region on

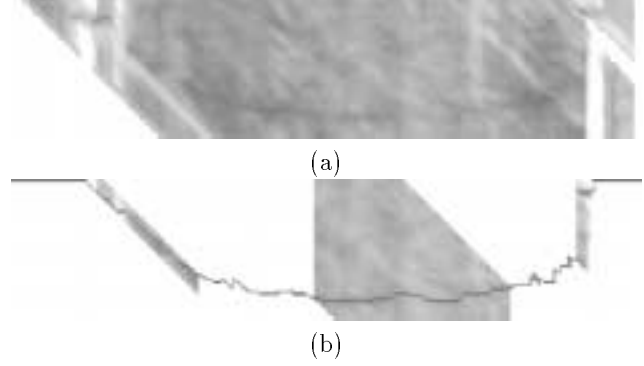


Figure 2. Disparity Space Image

the left scanline slides across the corresponding region in the right scanline, a line of “good” match can be seen in the  $DSI_i^L$ .

The best stereo match is constructed by finding the least cost path from the left end to the right end in the DSI. By assuming the ordering rule and occlusion rule [IB94] [Cox92], the jumps of disparity at each point can only go in three ways: in the left DSI, moving from left to right, jumping up and jump down and right; in the right disparity image, moving from left to right, jumping down and jumping up and right. The occlusion and ordering constraints greatly reduce the search space. Since noise could change the costs of the pixels, and thus change the minimum cost path through the disparity space image, paths are forced to pass those ground control points which are highly reliable. Dynamic programming is used to reduce the complexity of the least cost path searching along the scanline in DSI. The computed disparity path for one scanline of the algorithm described as above is shown in Figure 2(b).

## 3 Haptic Rendering

We now describe the haptic rendering of the reconstructed shape (a surface in 3D), using a two degree-of-freedom (2-dof) haptic interface (see Figure 3). 2-dof haptic interfaces are not the most natural for interacting with 3D objects, but there are compelling reasons for using them as haptic displays. They are considerably less complex, less expensive, and smaller than higher degree of freedom devices, and likely to remain so; they are now easily available on the market; finally, they map directly onto mouse interaction paradigm familiar to users and supported by all computers.

<sup>1</sup>Data courtesy of Dr. Doug Cochrane, Head of Section of Surgery at Vancouver Children’s Hospital, and Dr. J. J. Little.

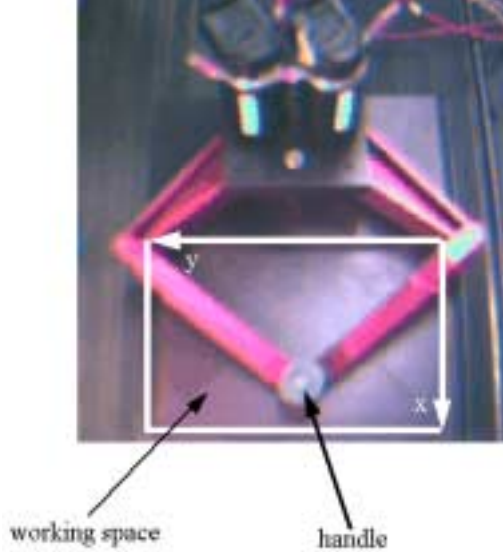


Figure 3. Pantograph

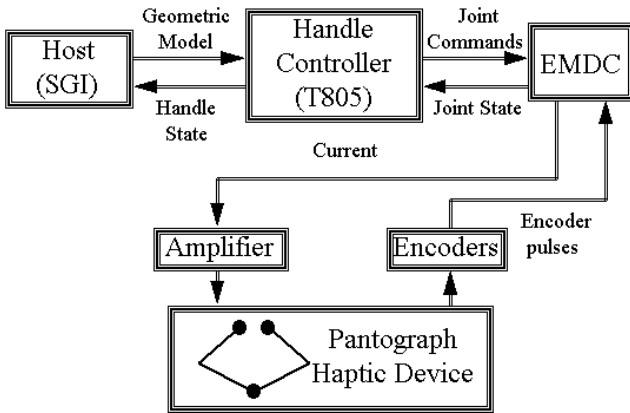


Figure 4. System Connections

### 3.1 Haptic Interface

The haptic interface is a 2-dof Pantograph haptic interface designed by Hayward, et.al. [HCLR94] (see Figure 3). It is controlled by a microcontroller and a small on-board network of transputers developed in our lab (see [SP96] for details). Figure 4 shows the system block diagram. The device behaves essentially like a mouse with force feedback, with a workspace of  $10\text{cm} \times 16\text{cm}$ .

### 3.2 Haptic Force Model

We model the haptic interaction as the act of pushing a small ball over the surface, under the influence of gravity. The force felt is proportional to the gradient

of the surface at the current position of the manipulum. Therefore, the user feels small forces on flat, horizontal surfaces, and large forces on vertical slopes. This model appears to be intuitively clear to users.

Specifically, let the surface be  $z = z(x, y)$  and let gravity act along the negative Z-axis. The gradient vector is defined as  $\nabla z = (\frac{\partial z}{\partial x}, \frac{\partial z}{\partial y})^T$ . Let  $v = (\dot{x}, \dot{y})^T$  be the velocity of the manipulum. Then

$$f = -k_g \nabla z - k_v v \quad (1)$$

where  $k_g$  and  $k_v$  are parameters. Note that, aside from the part that is proportional to the gradient, a velocity dependent damping term is added to increase the stability of the system.

Since the surface is already sampled on a regular grid, the partial derivatives are computed using Sobel operators which combine the vertical and horizontal differencing operations with some smoothing to reduce the effects of noise or very local texture. A  $5 \times 5$  Sobel operator is convolved with the disparity image in horizontal and vertical directions, respectively.

Stable haptic interaction requires high sampling rates from the controller – for instance our controller samples positions and applies forces to the user’s hand at 500Hz. To achieve this in real time, we precompute the disparity gradients over the entire image to form a discrete gradient force image.

### 3.3 Surface Smoothing

The depth map of the reconstructed surface is shown in Figure 5(a); brighter areas of the image are closer. Since the algorithm processes each scanline separately, horizontal spikes are generated by expanding the mismatches in this direction. These result in large, undesirable gradient forces and detract from haptic realism.

Removing this noise by applying a low-pass filter to the depth map is not desirable since this tends to wash out the entire force map, including crucial features. Fortunately, these noises are different from ordinary noises in that they are only in one direction and are thin, abrupt changes. We adopt a robust estimation method in the vertical direction to smooth the surface. We compute the distribution of the depth values and replace those contribute to the tail of outliers of the distribution with a weighted average of their reasonable neighbors. Also, if a mismatch occurs, its neighbors along the horizontal scanline tend to be mismatched too. For this reason, the sub-scanline centered on the noise point is also checked and adjusted. The result is shown in Figure 5(b).

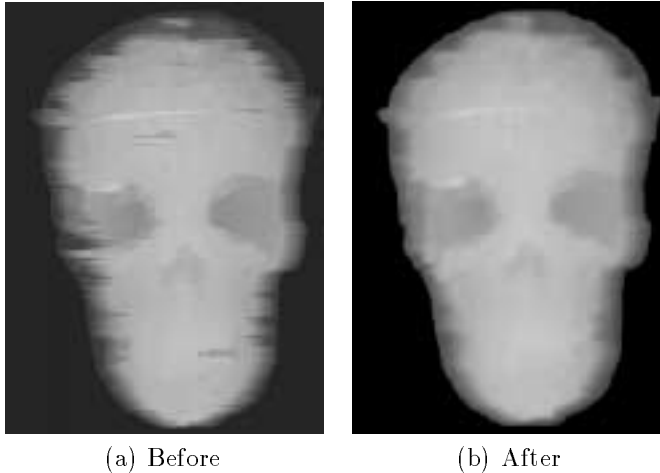


Figure 5. Surface Smoothing

## 4 Results

We have implemented a haptic virtual environment based on the methods developed in the previous sections. The interaction occurs using a visual display of the shape and a haptic interface.

The size of the sample image is  $240 \times 360$ . A force image of  $120 \times 180$  is used for rendering the force feedback. The image is mapped to a  $10\text{cm} \times 15\text{cm}$  area in the middle of the workspace.

A graphical display of the 3D reconstructed shape is also provided to the user, textured with front view of the image after subsampling for real-time performance. The front view is generated from the texture and depth information of both left and right image. A small marble which corresponds to the manipulandum is also displayed on the surface (see Figure 6).

As the user holds the handle of Pantograph and moves over the workspace, (s)he can feel as if (s)he is pushing a small marble on the surface of the skull; the visual display the marble is updated at 60Hz without any perceptible latency. It is difficult to describe the experience on paper; our informal experiments to date indicate that users can detect not only the obvious contour changes, such as the location of the eye-sockets, but also the subtle changes such as the wire on the skull's forehead and its teeth. Unfortunately, due to inherent limitation of reconstruction from real image data, a tradeoff is made between the disparity changes and the smoothness, and fine textures are hardly perceptible.

In summary, our system demonstrates the automatic generation of realistic haptic interactions directly from stereo visual images. Such a system has interesting applications in telepresence, telerobotics, and in displays

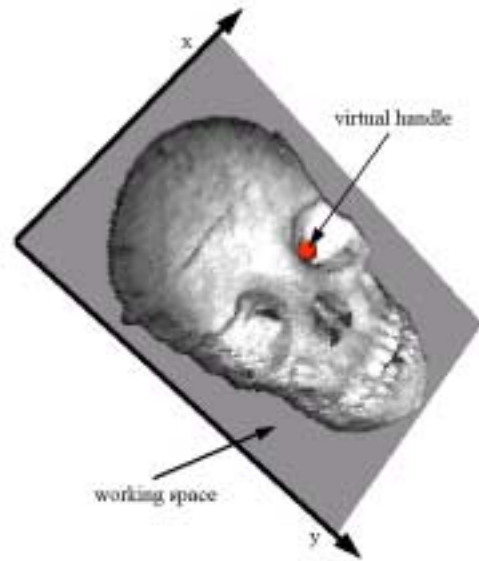


Figure 6. 3D Reconstructed Object

for the visually impaired.

## References

- [Cox92] I. J. Cox. Stereo without disparity gradient smoothing: A bayesian sensor fusion solution. In *British Mach. Vis. Conf.*, pages 337–346, 1992.
- [Fua93] P. Fua. A parallel stereo algorithm that produces dense depth maps and preserves image features. *Mach. Vis. and App.*, 1993.
- [HCLR94] V. Hayward, J. Choksi, G. Lanvin, and C. Ramstein. Design and multi-objective optimization of a linkage for a haptic interface. In *Proc. of the Fourth Workshop on Adv. in Robot kinematics*, 1994.
- [IB94] S. Intille and A. Bobick. Disparity space images and large occlusion stereo. In *ECCV'94 Proc.*, Stockholm, Sweden, 1994.
- [Kan95] T. Kanade. Virtualized reality: Concepts and early results. In *IEEE Workshop on Representations of Visual Scenes*, Boston, 1995.
- [PR96] D. Pai and L.-M. Reissell. Touching multiresolution curves. In *Proceedings of ASME Fifth Annual Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, 1996.
- [SP96] J. Siira and D. Pai. Haptic texturing - a stochastic approach. In *Proceedings of IEEE International Conference on Robotics and Automation*, volume 1, pages 557–562, 1996.