Haptic Visualization of Pictograms

Dita Němcová
xnemcova@fi.muni.cz

Faculty of Informatics
Masaryk University
Brno / Czech republic

Abstract

Limited eye sight people are deprived of possibility to perceive many things, 2D raster images are one of them. Therefore we tried to mediate them this experience in some way by means of touch in our work. Consequently the question occured, how to get the third dimension from discrete image function that represents raster image and how to visualize hapticly the created relief. We achieved the most conclusive results by experimenting with pictograms, i.e. raster images containing simple plain-coloured objects only. We mention experiments realised in our HCI laboratory.

KEYWORDS: haptic visualization, raster image, image processing.

1 Introduction

In order to visualize raster images for touch we suggested several methods of three-dimensional representation. Following article describes these methods and means of visualization.

Section one discusses devices we used for haptic visualization. Next section deals with the individual methods and the last one sumarizes the results of experiments with pictograms.

2 Haptic interface

2.1 PHANToM

In the Human-Computer Interactions Laboratory the PHANToM^{TM} device was available to realize haptic interface of our program. PHANToM^{TM} is a product of the SensAble Technologies, Inc. It provides a haptic experience that reproduces human touch with real world and allow users to feel the physical properties of virtual 3D objects. We experimented with the model 1.0 (Fig. 1). Dimensions of its workspace are aproximately 13 x 18 x 25 cm (roughly extent of a human forearm). The maximum exertable force is 8.5 N.

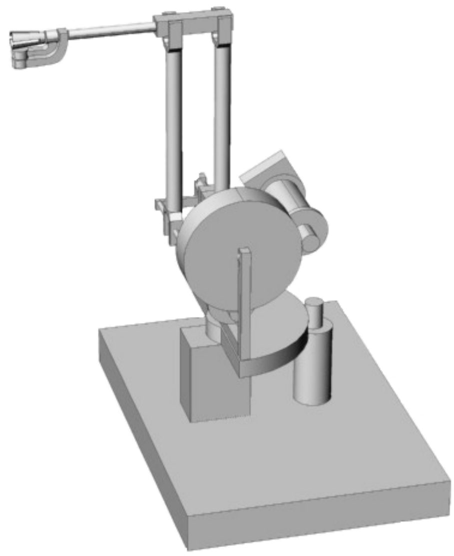

Figure 1: PHANToM 1.0

2.2 GHOST SDK

GHOST (General Haptic Open Software Toolkit) SDK is an object-oriented toolkit, written in C++, containing classes and methods for support of PHANToM haptic interface. It represents the haptic environment as a hierarchical collection of geometric objects and spatial effects (buzzing, inertia, viscosity etc.). GHOST strongly supports triangular meshes therefore we chose this representation of 3D objects.

3 Three-dimensional representation of raster images

3.1 Regular grid

We cover our raster image with regular grid of sample points (vertices of triangular meshes). In individual points we determine third coordinate according to the specific value of image function.

However, eventual triangular meshes are still too rough. Therefore we smooth triangles according to the edges detected[1] in raster image (as shown in Fig. 2).

Simple object and its final relief are shown in Fig. 3.

[1]For edge detection we can use e.g. Laplace operator, Sobel operator or some second derivative-based ones, e.g. Canny edge detector.

Figure 2: Triangle smoothing according to the edge

Figure 3: Regular grid

3.2 Adaptive meshes

To avoid disadvantages of previous approach, we use edge tracing to create adaptive meshes along the edges (Fig. 4).

If there is more than one object in the image or if this object contains other inner shapes, first we find all individual cyclic curves and then apply following procedure for each of them.

Let us suppose that input edge is represented by the list of its points and is of 1 pixel length, i.e. each point has exactly two adjacent ones.

Method:

- Choose starting point (first point in list), distance of polygon vertices (i.e. length of the side of the triangle aproximating an edge section) and direction of moving along the edge.

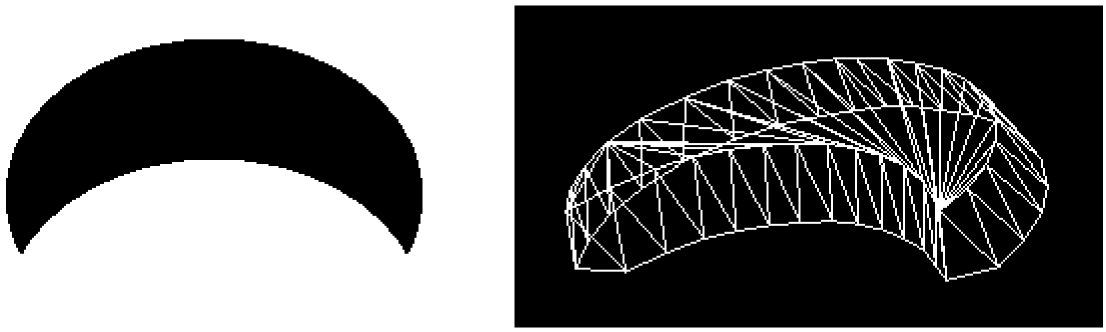

Figure 4: Adaptive triangular meshes

- Trace edge in chosen direction and search for polygon vertices. For each two found adjacent vertices determine the third one to create triangle altogether.
- Repeat previous for each point of the curve.

In the end we triangulate eventual area of the polygon to get continuous triangular meshes.

3.3 Negative edge relief

To recognize shapes in the raster image the edges of objects are very important. That is why we tried to create relief not of whole object area but of edges only (see Fig. 5).

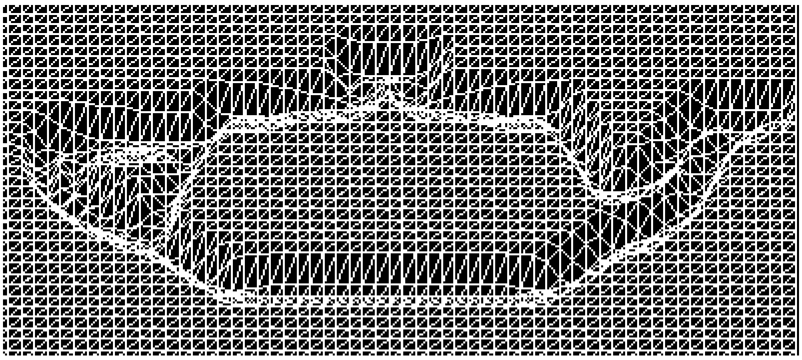

Figure 5: Negative edge relief

The best solution is negative relief because the valley created in such way allows user to follow easily the edge and not to lose haptic contact.

The edge itself is located in the lowest level of the relief. The depth coordiate of surrounding points should come near to it along some continuous curve to reach good haptic feeling. We used sine function to interpolate coordinates (Fig. 6).

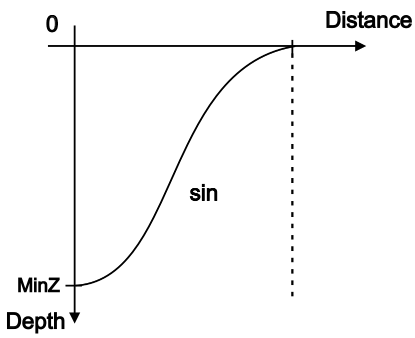

Figure 6: Relief profile – sine interpolation

The negative edge must be of certain length for user to recognize it. The edges of very narrow objects cannot be well distinguishable and can be merged into one negative area. We can use scaling to zoom details and get better notion of image. Negative edge is also useful for visualization of graphs of math functions. In raster images language graph is a curve consisting of discrete function values. We examined this approach for functions with smooth continuous graph like sine (Fig. 7). Picture on the right shows how is this relief influenced by changing function argument (from x to $2x$).

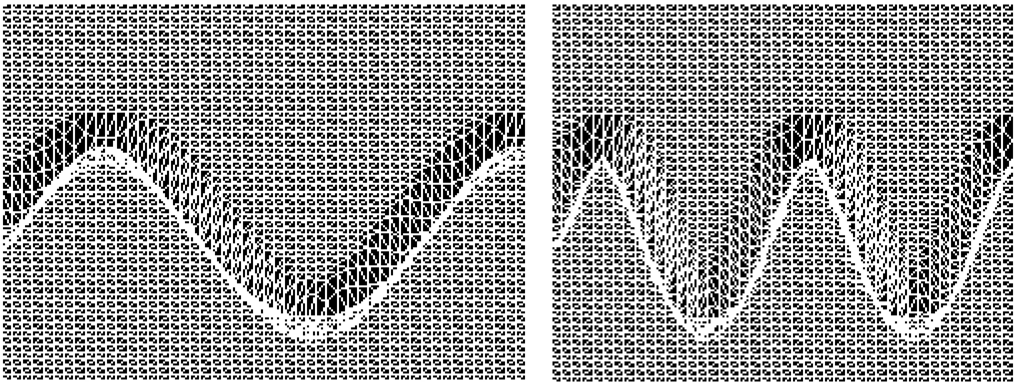

Figure 7: Sine function graph using negative edge relief

3.4 Heightfield

Following method is possible to apply to any general raster image. However, the results of recognition of image content are not as conclusive as applying to simple objects. The information available to get depth coordinate is the image function again. The x and y coordinate of each pixel is known, accordingly we try to replace its brightness by corresponding height. In case of colour raster image we used the RGB (Red-Green-Blue) colour model where we added up the individual colour components. Then the z coordinate increases from black $[0.0, 0.0, 0.0]$ to white $[1.0, 1.0, 1.0]$, black is the lowest level of relief.

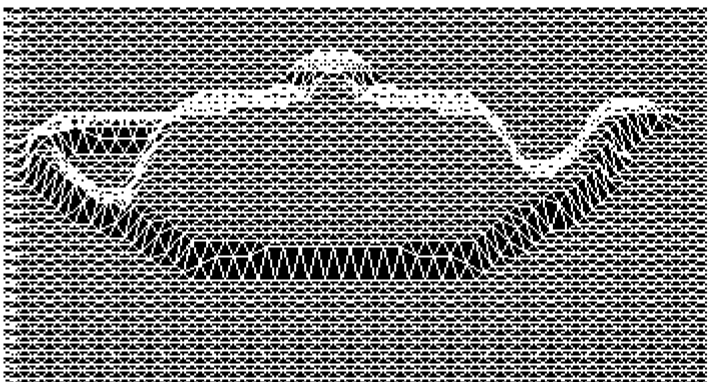

Figure 8: Heightfield from bitmap

The smoothing of adjoining triangles is very important for acceptable haptic feeling. But, as shown in Fig. 9, if we smooth out the sharp transition edges we lose some information about the edges of objects and reduce the probability to recognize object.

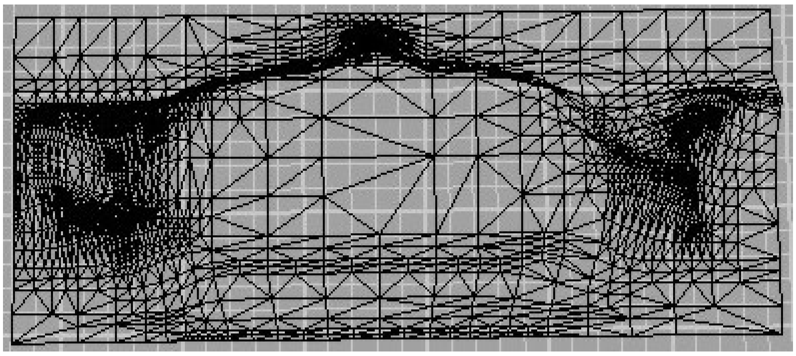

Figure 9: Smoothing of triangles in heightfield (created in Rhinoceros NURBS modeller)

4 Comparison of different methods used to process raster image

We achieved different success of recognition of image content using the suggested methods.

For experiments we used program written in C++ (for Windows NT), OpenGL and GLUT. Source code of the program is platform-independent to facilitate the portability to IRIX based platforms as well. Structure of the program is shown in Fig. 10.

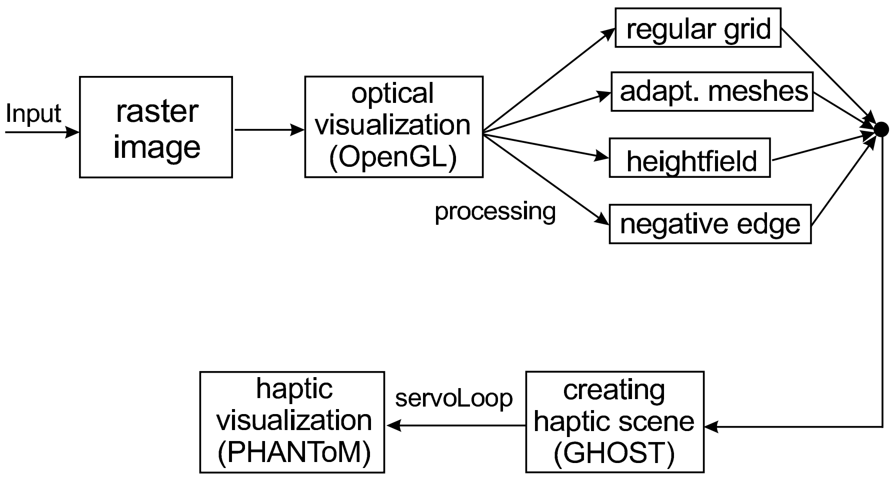

Figure 10: Structure of the program used for experiments

Regular grid was found unusable in practice. Because relief rises from very approximate area, outline of relief is so rough that the shape can not be recognized.

Adaptive meshes seems to be the best method for haptic visualization of pictograms. Edges of the object are distinct enough and transitions among adjacent triangles smooth enough, therefore object shape is well perceivable. Only height of relief is of importance. If the height is too small, user (who has no visual information about his location in relief) can lose haptic contact with relief edge.

Heightfield method gives better results for general raster images than for pictograms. The disadvantage of pictograms is that the edges are sharp defined. Heightfield method does not take into consideration these edges and therefore the shape is partly misshapen. It makes recognition difficult. If we have the same raster image with interpolated triangles and without interpolation, success of recognition is different as well. In case of no interpolation edges are at least partly preserved.

Negative edge relief gives good results for pictograms which do not contain too narrow parts.

5 Conclusion

We suggested and tried out several methods to hapticly visualize simple raster images in order to mediate a new experience. The third dimension was derived from the brightness value of discrete image function for individual pixels. We found out that aproximation should not disturb original object edges to facilitate the recognition of image content. It turned out that the method with adaptive triangular meshes gives the best results in haptic recognition of pictograms.

We also suggested some ways how to use these methods in teaching limited eye sight people (and not only them). The negative edge relief is very useful to hapticly visualize graphs of functions, adaptive meshes allow to recognize the letters of alphabet. The presented results are based on experiments with only 2 individual subjects (including author). In future we plan to test described methods on more subjects systematically.

Acknowledgments

This work was supported by a research grant, from Grant Agency of Czech Republic, Contract No. GACR 201/98/K041.

References

[1] Bedřich Beneš Jiří Sochor, Jiří Žára. *Algoritmy počítačové grafiky.* Vydavatelství ČVUT.

[2] Roger Boyle Milan Šonka, Václav Hlaváč. *Image Processing, Analysis, and Machine Vision.* Brooks/Cole Publishing Company, 1999.

[3] William K. Pratt. *Digital Image Processing, Second Edition.* John Wiley & Sons, Inc., 1991.

[4] John C. Russ. *The Image Processing Handbook, Second Edition.* CRC Press, Inc., 1995.

[5] Inc. SenSable Technologies. *GHOST SDK Version 2: Programmer's Guide.* 1998.