Copyright 2000 IEEE, Proc. of the International Symposium on Robotics with Applications, June 2000.

GRAPHICAL AND HAPTIC INTERACTION WITH LARGE 3D COMPRESSED OBJECTS

Krasimir Kolarov
Interval Research Corp., 1801-C Page Mill Road, Palo Alto, CA 94304
Kolarov@interval.com

Abstract

The use of force feedback can greatly enhance our interaction with large 3D environments and objects. By combining haptics and graphics we can explore and manipulate objects defined by meshes consisting of hundreds of thousands of polygons. To do that feasibly even on modern computers, we need to develop compression methods that allow for using different levels of resolution and detail in interacting with the 3D objects. Such methods should allow us to zoom in on the parts of the objects being manipulated while keeping the rest of the mesh in compressed form. This paper introduces an approach for solving this problem combining some of our recent results in force feedback, haptic interaction and innovative methods for 3D data compression. There are a number of potential applications in the areas of personal robotics, entertainment, visualization and 3D exploration.

1. Introduction

Haptic interfaces are devices that stimulate the sensory capabilities within our hands. The word "haptic" has to do with tactile sense of touch and describes an information processing perceptual system that takes its inputs from receptors in our skin. An important characteristics of the human haptic system is that it relies on action to stimulate perception, it can both sense and act on the environment (as opposed to the purely sensory nature of human vision and audition). That ability enhances the sense of immersion in the environment that is important in VR type applications.

The goal of haptic interfaces is to get as close as possible to the human tactile capabilities. The human tactile sensory system distinguishes vibrations up to 1 KHz. The area of haptics covers a number of issues including: actuators, sensors, tactile feedback interfaces, control and physical modeling. In this paper we will concentrate on 3D haptic interfaces. The particular haptic device used is the PhanTom force feedback manipulator from Sensable Technologies.

In the recent years, 3D haptics and graphics interaction has become an active area of research. In particular [5,6] introduce a powerful framework and language HL (for Haptic Language) that allows for quick "haptization" of arbitrarily complex 3D objects defined as polygonal meshes. Examples include objects consisting of hundreds of thousands of polygons. Motion is incorporated through the implementation of simple dynamics simulations. Recent work in that area has incorporated sound and collision algorithms.

We would like to be able to build a system that allows for haptic and graphic interaction with complex environments which include a number of (possibly) large and complicated non-stationary objects. Since we are aiming at real-time interaction, it is clear that the haptics models described above need to be integrated with some degree of compression. In particular the part of the environment that we are not immediately interacting with, can be kept in a compressed form.

Realistic haptic interaction with virtual objects requires very high force feedback update rate. To achieve such data rate we need highly efficient haptic representation and rendering and fast model computations. That requires well designed compression algorithms for high dimensional objects as well as matching of the haptic display to the haptic capabilities of the human user.

Copyright 2000 IEEE, Proc. of the International Symposium on Robotics with Applications, June 2000.

2. Compressing data defined on 3D objects

2.1 Problem definition

With the fast advent of the Internet and our abilities to manipulate 3D geometrical models, there has been a resurgence of research in compact representation of complicated 3D objects. The majority of work has concentrated on building efficient 3D models of the **geometry** of the objects. [3] recently addressed the issue of compact representation (compression) of the **data** defined on the surfaces of arbitrary 3D objects. That paper (as well as [2,4]) introduced a technique for compression and expansion of functions defined upon a 3D object using a second generation wavelet transform and a modified zerotree bit-encoding scheme.

This method models the data to be compressed as a function. In computer graphics applications for example, that data can be the texture of the object, which will be represented as one or more scalar functions defined on the object's geometry. The surface of the 3 dimensional object can be represented as a 2-manifold and the method compresses scalar (or vector) functions defined on 2-manifolds. While the traditional data compression literature deals with simple manifolds, we aim at higher-dimensional or complex geometries such as spheres or general polytopes.

2.2 Transform data compression

The transform compression of a function involves three steps: a transformation of the function, quantization, and entropy encoding. Only the quantization step actually discards information - the transformation and encoding steps have inverses and are reversible.

Step 1) the function is subjected to a reversible linear transformation in order to concentrate most of its entropy (i.e., information) into a low dimensional subspace, thus simplifying its description. A wide variety of transformation techniques are currently in use, including the discrete cosine transform (DCT) and wavelet transforms.

Step 2) the coefficients of the transformed function are quantized, i.e., scaled and truncated appropriately. Algebraically, this amounts to a projection of the function into one of the low dimensional subspaces in which the entropy is concentrated. Such a step discards the entropy perpendicular to the projection - hopefully information that will not be required by an application. For example, if an image is being compressed for later viewing by the human perceptual system then information about high spatial frequencies can be attenuated or discarded on the grounds that such information will be imperceptible. Likewise in audio compression for human listening, information at frequencies above 20 KHz. may be discarded as imperceptible.

Step 3) the remaining coefficients after the quantization step are encoded using a suitable entropy encoding technique. Not all possible values of the coefficients are equally likely nor are all sequences of coefficients equally likely. Entropy encoding exploits statistics about the data to use shorter codes for the more frequent situations and longer codes for the less frequent situations. In particular, the transformation and quantization steps produce many zero values so that a major task of an entropy compression technique is to code the geometric location of these zeros. These locations are heavily correlated by location and scale and are compactly encoded by a quadtree-like technique known as *zerotrees* [7,9].

2.3 Summary of the compression method

The method in [3,4] starts by choosing a base complex (e.g a cube or an icosahedron) as a coarse initial model of the 3D object. This base complex is recursively subdivided (by subdividing the triangles comprising the mesh representation of the object) to produce a refined triangular mesh with subdivision connectivity. A tree structure, called a G-tree, is created in which each node of the tree structure represents a cell of the triangular mesh.

Copyright 2000 IEEE, Proc. of the International Symposium on Robotics with Applications, June 2000.

Overall we model a 2-manifold by a triangulation with subdivision connectivity, determine function values to attach to each vertex, and determine a method to calculate wavelet transforms on this structure. If the function is suitably smooth the norm of the wavelet transform will be concentrated at a small number of vertices.

Second generation wavelets (described in [8]) for the specified function are calculated and scaled, the wavelet coefficients being defined at the vertices in the triangular mesh and at the vertex correspondent nodes of the G-tree. Using a modified zerotree encoding scheme, the G-tree is processed threshold by descending threshold, depth-first, left-to-right, outputting bits indicative of significant G-tree nodes and the corresponding coefficient bits. This results in a bit plane by bit plane embedded encoding.

The decoding algorithm inputs bits according to the modified zerotree scheme into the G-tree structure, refining the wavelet coefficients. Descaling and an inverse second generation wavelet transform completes the synthesis of the original data function.

3. Haptic Manipulation of Large Compressed 3D Data Sets

3.1 Structure of the compressed bitstream

As a result of the compression method described in the previous section we build a coded file representing the data we want to interact with. The structure of the coded file is as follows: We begin with a preamble that details the base complex, the subdivision method, the number of subdivision levels, and the scaling of the wavelet coefficients. The preamble information is sufficient to reconstruct the G-tree and the decoder initializes by reconstructing the G-tree from the preamble. Following this are a sequence of bit planes, each representing the difference between the current approximation (with threshold T) and the next approximation (with threshold $T/2$). Each bit plane is a sequence of two sub-strings. The first of these are the *navigation* bits which describe the spatial location in the G-tree of the coefficients significant at this bit plane. The second are the bit plane coefficient bits (*refinement bits*).

The navigation bits for a bit plane result from a depth-first, left-to-right walk of the G-tree. However, terminal zerotrees are elided. This compactly encodes zerotrees, greatly reduces the encoding of leading zeros, and accounts for the bulk of the lossless compression of the method. The actual encoding method of the navigation bits proposed in [3] is very efficient. The tree walk proceeds by a sequence of binary decisions which guide the walk while avoiding zerotrees. The encoder outputs as navigation bits each binary predicate as the decision is made. The decoder then performs exactly the same G-tree walk as the encoder by using exactly the same algorithm, with the exception of using the next navigation bit as the next binary predicate.

The refinement bits for a bit plane contain one bit per coefficient significant at that bit plane. As a coefficient becomes significant at the bit plane corresponding to its significance, it is added to the end of a list of significant coefficients. The appropriate bit of each coefficient is read out by the encoder to form the refinements bits for the current bit plane. Coefficients are represented in sign-magnitude form, and since the position of the leading one is known it need not be read out. The sign bit is read out in its place.

The decoder reads the refinements bits and distributes one bit to each significant coefficient. When the decoder is finished the coefficients can be attached to their proper nodes in the G-tree.

3.2 Integration of the compression and Haptics environments

The topic of this paper is interactive 3D visualization and 3D haptic manipulation. To that goal, the library of haptics routines HL described in [5, 6] complements the graphical routines available through GL. In addition, the coded file in 3.1 can be integrated with the 3D haptic modules in HL in order to interact both haptically and graphically with the compressed data. To manipulate

Copyright 2000 IEEE, Proc. of the International Symposium on Robotics with Applications, June 2000.

haptically complicated 3D objects with millions of patches, there is a need of multi- resolution representation and compression of the data. That involves zooming in on the part of the object that have been probed with the haptic device while keeping the rest of the object in sufficiently recognizable low resolution.

We have developed an interface that connects the compression code for functions (e.g. elevation) defined on manifolds, with the HL library. In relation to that, we have incorporated stereo and texture display for the compression data generated by the SW software [2]. The haptic interface used in our simulations is the PhanTom.

The haptics implementation is done by adding a new set of triangle callback methods, based closely on the "Draw" methods (C++ classes to visualize the mesh) used for the visual representation. The idea is that any function can be bound to a manifold (or tree of triangles). Once it is applied, it goes through every tree of triangles in the representation and calls the supplied function for every triangle it encounters. A "haptic draw" triangle routine is bound and it applies the function to the manifold to haptically draw the manifold. The callback of the function is applied automatically whenever the manifold changes.

Using that method we can feel the 3D object through any phase of multi-resolution representation. The haptics runs at all the resolutions. Depending on the complexity of the object, the collision algorithm can process different amount of triangles (comprising the representative mesh). For example, at the highest resolution 6, a globe (8 cm wide and stationary, used in the example described in the next section) contained 81920 polygons with the update rate being ~2000 Hz. We believe that with faster processor and code optimization, we can process more levels (of subdivision) to more than 300K polygons!

The system can also integrate dynamics using simplified equations of motion, i.e. using only diagonal terms where each animation term is basically independent (that covers models for most of the currently used haptics objects). The limitation lies in models with too many degrees of freedom (deformable models, ones generated from sensed information).

When building integrated systems, we need to remember that while human's haptics exceeds 500 Hz, the visual sensory limit is 30 Hz and the human motor output is only 10 Hz. One potential solution suggested in the literature is to use proxy blending, i.e. create an actual proxy and another proxy and use blending between them. That might allow for support for object-object collision and eventually a full dynamic support for systems with small dof. There are different options for blending including visco-elastic or other mechanical models.

3.3 Haptic Resolution / Compression Norms

Another aspect of our haptic work is understanding human haptics capabilities and perception. That involves research and development of vector norms for evaluating “haptic resolution” similar to the ones developed in the visual and graphic resolution fields. A commonly used notion in human vision is the "contrast sensitivity function" (CSF) - the variation of contrast sensitivity as a function of spatial frequency. Because the human visual system is more sensitive to image contrast than absolute illumination intensity, contrast sensitivity better represents the human visual capabilities. The CSF enables efficient displays of object geometry and surface texture. It has also been very useful for image compression since it defines which frequencies of the image can not be perceived by the human vision and thus can be compressed out.

In addition, the importance of different scaling norms for evaluating compressed data have been recognized and studied in the visual world. In particular the L1 norm has been found to be most appropriate in matching the characteristics of the HVS (Human Visual System) [1]. Similar research is necessary in identifying the relevance of L0, L1, L2 scaling norms for better compression of haptic data. In parallel to the CSF we need to determine the haptic sensitivity of human subjects to spatial detail. Similarly to the "image contrast" notion in vision, we need to

Copyright 2000 IEEE, Proc. of the International Symposium on Robotics with Applications, June 2000.

determine the limits of the human haptic perception in terms of "force contrast". That type of knowledge can be used for efficient compression and rendering of haptic information in virtual reality systems. It also helps understanding the human haptic abilities and haptic system.

4. Implementation and results

Among the 3D geometrical objects the sphere presents a simple, albeit challenging example that can be represented with various level of detail (i.e. with various sizes triangular meshes). Spherical meshes [8] can be used to represent the Earth example and compress the topographic function that is the elevation (wrt. sea level) of the Earth [3,4]. This function is approximated by the ETOPO10 data set which samples the Earth every 10 arc minutes (satelite data for 1.5 million points approximately 11 miles apart available from NOAA).

The base complex approximating the sphere used in [3,4] is an icosahedron which is further subdivided eight times yielding 655362 vertices and 1310720 triangles. The vertices of this complex were then radially projected onto a concentric geoid (vertices average about a 22 arc minute separation - about 25 miles).

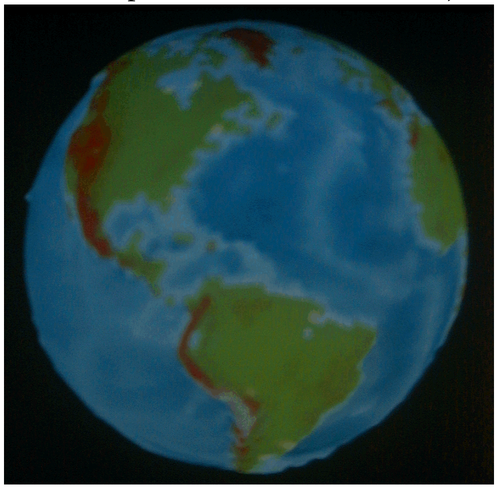
Figure 1 The compressed 3D stereo Etopo-10 data

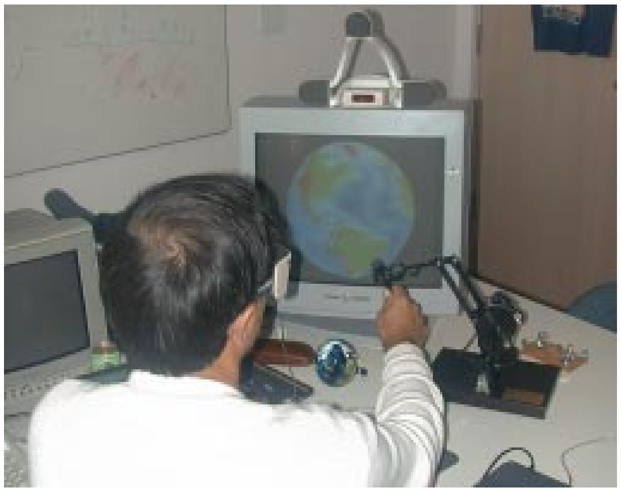
Figure 2 The experimental 3D haptics/graphics rendering setup

The resulting data set represents the function to be compressed in a discrete form. This data was then wavelet transformed using linear lifting and compressed at various ratios. For visualization purposes, the compressed data at different resolution was decoded and inverse transformed to obtain again values for each coefficient in the mesh. That compressed/decompressed data is then rendered in stereo for visual representation. Figure 1 shows the elevation data in stereo (mapping the wavelet values as scaled distances from the center of the sphere). The figure illustrates the data in color using different color hues to represent different elevation ranges ("pseudo-color").

The same processed wavelet coefficients are fed to the haptic processing system using the routine in the HL language (see [5,6] for HL). That allows the user to explore and manipulate both haptically and in stereo graphics the compressed data in real time. Figure 2 illustrates the experimental setup including stereo glasses connected to a SGI machine running the visual rendering. The haptic rendering is computed on a PC running Linux, using the PhanTom arm from Sensable as a haptic interface. The system includes an (optional) stereo head tracker which allows for changes in the graphic views as the user moves around in the workspace.

The compressed Earth data visually look exactly as what we would expect for good compression given the resolution that is achieved. In addition to the perceptual evaluation (visual and haptic) of the compressed objects, we can compute an analytical measure of the compression quality described by the peak signal-to-noise ratio (PSNR) for the different compression ratios (see Figure 3). For compression ratios up to 500:1, the error is undistinguishable.

Copyright 2000 IEEE, Proc. of the International Symposium on Robotics with Applications, June 2000.

Our approach is applicable for any 3D objects, not necessarily ones topologically equivalent to a sphere. Figure 4 shows an example of a mechanical part with holes (courtesy of the University of Washington) with mapped Earth elevation data using the mid-point subdivision method.

Since all the triangles are kept in RAM for interactive visualization, different base manifolds allow different number of subdivision levels. The Earth manifold (starting with an icosahedron) can be subdivided 8 times, an image (starting with a square) allows 10 levels, while the mechanical part in Figure 4 can only be subdivided twice before saturating the memory.

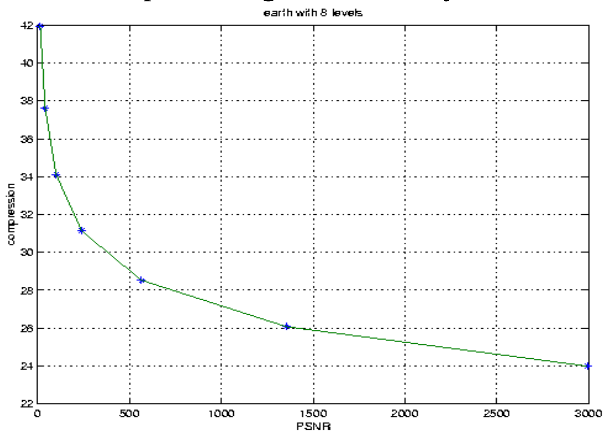

Figure 3. Compression performance for the Earth example

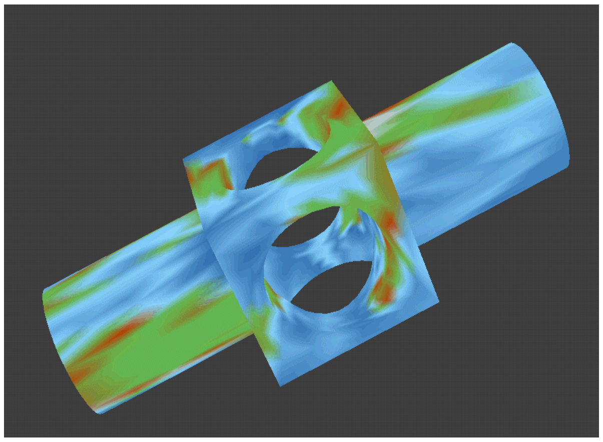

Figure 4. Elevation data mapped on a Mechanical part

5. Applications

There are a number of interesting applications for haptic/graphics interfaces, in particular in the areas of medicine, entertainment, telerobotics. Enhancement of graphical user interfaces through haptics will enable users to feel where the buttons on their programs are. In computer games haptics allows for engaging touch interactions in a cost-sensitive market. Using haptics in simulations for training humans will enable users to perform tasks that require sensorimotor skills (surgery, army training). Haptics can enrich interactions with computer-generated 3D data - a benefit for users of CAD/CAM, data visualization, engineering and scientific applications.

References:

[1] R. A. DeVore, B. Jawerth, and B. J. Lucier, "Image compression through wavelet transform coding", *IEEE Trans. Inform. Theory*, vol. 38, pp. 719-746, March 1992

[2] Kolarov, K. "Representation, Optimization and Compression of High Dimensional Data" World Multiconference on Systemics, Cybernetics and Informatics SCI'98/ ISAS'98, Orlando, 1998.

[3] K. Kolarov and W. Lynch. Compression of functions defined on surfaces of 3D objects, *Proceedings of the Data Compression Conference*, Snowbird, Utah, pages 281--291, March 1997.

[4] K. Kolarov and W. Lynch. Wavelet Compression for 3D and Higher-Dimensional Objects, *Proc. of SPIE Conference on Applications of Digital Image Processing XX*, Volume 3164, San Diego, CA, July 1997.

[5] D. Ruspini, K. Kolarov and O. Khatib. The Haptic Display of Complex Graphical Environments, *Computer Graphics Proceedings, Annual Conference Series, SIGGRAPH'97*, Los Angeles, California, August 1997.

[6] D. Ruspini, K. Kolarov and O. Khatib. Haptic Interaction in Virtual Environments, *Proc. of the IEEE International Conference on Intelligent Robots and Systems IROS'97*, Grenoble, France, September 1997.

[7] A. Said and W.A. Pearlman. A new fast and efficient image codec based on set partitioning in hierarchical trees, IEEE Transactions on Circuits and Systems for Video Technology, 1997.

[8] P. Schröder and W. Sweldens. Spherical wavelets: Efficiently representing functions on the sphere, *Computer Graphics Proceedings, (SIGGRAPH 95)*, pages 161--172, 1995.

[9] J.M. Shapiro. Embedded image coding using zerotrees of wavelet coefficients, 41(12):3445--3462, IEEE Trans. Signal Process, 1993.