Since 1993, the authors have been developing ViSurf, a three-dimensional virtual surface modeler with force feedback [1][2]. The system has two kinds of point-contact type force feedback device; a 6 D. O. F. Cartesian type manipulator [3][4], which has been developed by the authors, and a PHANToM Haptic Interface (SensAble Devices, Inc., U.S.A.). A user can trace or deform a B-Spline or a triangular surface with a force feedback device and can feel it.

In a realtime haptic virtual reality system like ViSurf, it is essential to separate haptic rendering process from others, e.g. graphic rendering and geometry changing, since haptic rendering requires at least 1,000 Hz device control latency to achieve precise and stable haptic representation of geometry. Graphic rendering and changing geometry, e.g. inserting a knot into a B-Spline surface, require much lower refreshing rate (~30 Hz) and longer time that can affect the quality of haptic rendering.

To solve this problem, ViSurf has process architecture shown in Fig. 1 since 1993. Haptic rendering process has no geometry database and haptic rendering is done based on feature library using features sent by the modeler. Communication traffic is minimized by exchanging feature ID and minimum feature geometry data. We call this process architecture based on features Feature-Based Haptic Rendering (FBHR) [5]. FBHR is highly extensible and efficient.

2. Feature
A feature is a unit of inter process communication and can be categorized into two groups: data and commands. The data feature includes local haptic characteristics of virtual objects, such as local geometry (e.g. shape and position) and material (haptic texture, stiffness, and so on). Fig. 2 shows shape feature examples. Solid lines are edges and dotted lines show the infinite extent. In 1993, there were only two shape features: None and Plane. Fan and Blade features were introduced to show edges and corners of a surface more accurately. Haptic renderer renders features instead of the true geometry.

Feature pre-fetching also improves haptic rendering quality. Modeler can predict cursor movement in the

**Abstract**: Since haptic rendering requires at least 1 kHz device control latency to obtain high haptic fidelity, haptic rendering process must be separated from others in realtime virtual reality system with haptic/force feedback. Feature-Based Haptic Rendering gives process architecture for load balancing. Haptic rendering is done based on features using feature library. Feature is local haptic characteristics of geometry and commands. Features can be cached and pre-fetched for interpolation, which gives high extensibility to the method.

**Key Words**: Haptic Rendering, Virtual Surface Modeler, Feature, Virtual Reality
Collision detection
Feature selection
Geometry handling
Graphic rendering
Get position data
Haptic rendering based on features

Async. TCP/IP socket communication using "feature"

Haptic Renderer (PC)
- Get position data
- Haptic rendering based on features

Graphic Renderer & Modeler (GWS)
- Collision detection
- Feature selection
- Geometry handling
- Graphic rendering

Feature Library

Geometry Database

Haptic feedback device

Fig. 1 Load Balancing Architecture of Feature-Based Haptic Rendering

next several frames and send features in advance. Feature caching and pre-fetching are effective in shape tracing mode.
Temporal interpolation is necessary in shape deformation mode, since the shape changes as the pushing by the user and the haptic renderer should discard older features and use newer ones.

3. Protocol
FBHR provides a communication protocol to exchange features. A feature will be split into one or more fixed-length data packets, from which the receiver reconstructs the original feature. Packets are sent over a single asynchronous communication line, which is a TCP/IP socket in our current implementation. The header part of each packet contains a feature ID, serial packet number, and mode flags. The body part contains geometry data of the feature.

4. Result
A virtual surface modeler with force feedback has been implemented with FBHR, whose modeling and graphic rendering run on a SGI Indigo2 Elan (200 MHz R4400) and haptic display is PHANToM (6 DOF input and 3 DOF force output) on a PC (200 MHz Pentium Pro). They communicate over 10Base-T EtherNet. A 5 - 10 kHz haptic rendering latency has been achieved with 20-35 Hz communication and graphics refreshing rate, which gives high haptic rendering quality of a virtual surface.

5. Discussion
Similar haptic process architecture was presented as intermediate representation [6][7] which originally is equal to a plane feature. The recovery time algorithm [7] is an example of temporal interpolation of plane features.
Shape Approximation Device [8][9] is a hardware implementation of shape features. It is a carefully designed collection of edges, corners, flat and curved surfaces, which a user touch with a 3 DOF probe.
Compared to related work above, FBHR is a more unified and extensible method. FBHR has a single communication line, while intermediate representation by Mark et al. uses two. FBHR divides haptic rendering into feature rendering and feature interpolation. Each feature is so simple that it can keep high rendering latency, and a variety of interpolation functions can be developed. Since FBHR is a software implementation, it is safer, more extensible, and more flexible than hardware.

6. Future Work
Haptic volume rendering is a possible application of FBHR. Shape features can be constructed as local isosurfaces by the Marching Cubes algorithm [10], to which any material feature can be attached. Physically correct and non-penetratable haptic rendering of a thin, elastic object is then possible using proxy technique [11], which is important for surgical simulation but is difficult with the voxel density field algorithm [12].
We are now developing more features, their rendering algorithms, and interpolation functions. Extending FBHR to multiple-contact type force feedback device is a challenging issue. It will be possible by introducing an object momentum feature.
Another problem to be solved is feature selection when the cursor speed is very high. Feature pre-fetching is
expected to be a solution, however, sawlike geometry possibly increases the communication traffic and affects haptic rendering quality. Switching between shape feature and haptic texture is needed. To proceed our research, human haptic characteristics should be investigated as well, since very little is known about our sense of touch yet.

References