SRI International

Technical Note 515 • January 1992

OVERVIEW OF THE SRI CARTOGRAPHIC MODELING ENVIRONMENT

Prepared by:

Andrew J. Hanson
Senior Computer Scientist

Lynn Quam
Senior Computer Scientist

Artificial Intelligence Center
Computing and Engineering Sciences Division

APPROVED FOR PUBLIC RELEASE: DISTRIBUTION UNLIMITED

Originally published in Proceedings: Image Understanding Workshop, April 6-8, 1988; distributed by Morgan Kaufmann Publishers, Inc., San Mateo, CA 94403 (Approved for Public Release--Distribution Unlimited). This research was supported by the Defense Advanced Research Projects Agency under Contract No. MDA903-86-C-0084.

Overview of the SRI Cartographic Modeling Environment*

Andrew J. Hanson and Lynn H. Quam

Artificial Intelligence Center
Computer and Information Sciences Division
SRI International

Abstract

The SRI Cartographic Modeling Environment has been created to support research on interactive, semiautomated, and automated computer-based cartographic activities. The underlying image manipulation capabilities are provided by the SRI ImagCalc^{TM} system. The cartographic features and data that can be entered include multiple images, camera models, digital terrain elevation data, point, line, and area cartographic features, and a wide assortment of three-dimensional objects. Interactive capabilities include free-hand feature entry, altering features while constraining them to conform to the terrain and lighting geometry, adjustment of feature parameters,

and the adjustment of the camera model to display the scene features from arbitrary viewpoints. Cartographic features are depictable either as wire-frame sketches for interactive purposes or as texture-mapped renderings for realistic scene synthesis. High-quality simulated scenes are created by texture-mapping images onto terrain data and adding renderings of cartographic features using depth-buffering and antialiasing techniques. Motion sequences can be created by choosing a series of camera models and rendering the simulated appearance of the scene from each viewpoint.

1 Introduction

Among the cartographic applications upon which computer-based techniques can have a substantial impact are the following:

*Originally published in Proceedings: Image Understanding Workshop, April 6–8, 1988; distributed by Morgan Kaufmann Publishers, Inc., San Mateo, CA 94403 (Approved for Public Release—Distribution Unlimited). This research was supported by the Defense Advanced Research Projects Agency under Contract No. MDA903-86-C-0084.

- **Cartographic compilation.** Manual cartographic compilation is expensive and traditional techniques limit the ways in which constraints and previously-compiled information can be incorporated into the process. The task of compilation requires the construction of world-registered three-dimensional models from a body of sensor data and geometric information sources. Such information is now used not only for the creation of paper-based cartographic products, but also for the establishment of digital databases that are used for two-dimensional cartographic depictions and three-dimensional tactical simulations. Without automation, the increasing demands for timely, detailed cartographic information will vastly exceed the resources available.

- **End-user consumption of cartographic products.** A growing number of end-users need to interact directly with digital data, without requiring the use of paper media. These same end-users must be able to select, customize, and update the distributed data to meet rapidly changing situations or unforeseen requirements. These needs have led to the concept of the "Deployable Digital Database," in which digital media replace paper media for information storage and retrieval.

- **Generation of mission planning and training scenarios.** Mission planning and training can be greatly enhanced by the ability to visualize spatial environments. This can be accomplished within a computer graphics system by generating synthetic static views and motion sequences incorporating detailed cartographic feature models and texture maps of real imagery projected onto the features and terrain.

The SRI Cartographic Modeling Environment has been developed as a sophisticated experimental research tool for exploring three-dimensional modeling tasks. While the system has many capabilities in common with other Computer Aided Design (CAD) systems, it is particularly well-suited for evaluating computer-based approaches to the cartographic applications noted above. It supports a variety of interactive facilities for creating, editing, interacting with, viewing, and rendering three-dimensional models of world objects, cartographic features, and terrain. The modeling capabilities may be used independently or in conjunction with multiple, metrically calibrated digital images. Interaction with geometric models is characterized by intuitive simplicity and by innovative techniques for exploiting geometric and data-driven constraints in the manipulation process. Synthetic views of a scene may be constructed from arbitrary viewpoints using terrain and feature models in combination with texture maps acquired from aerial imagery. Figure 1 gives a schematic overview of the system's organization and capabilities.

This system extends the two-dimensional capabilities of the ImagCalc^{TM} image manipulation system to three-dimensional space. The geometric parameters of cartographic features and camera models are defined in a three-dimensional world coordinate system. Each image has an associated camera model that defines a ray in space for each pixel in the image. These camera models are used to project world coordinates to image coordinates; when a world coordinate corresponding to a point in the image is required, the camera models also serve to project a ray from the image to the world, so that its intersection with terrain and object models can be found.

Among the potential applications of the system we note the following:

- Studying human interface techniques for enhancing productivity in the computer environment.
- Developing improved specifications for future, more specialized application environments.
- Serving as a flexible background environment in which to develop and test semiautomated and automated cartographic analysis techniques.
- Performing one-of-a-kind tasks such as creating a unique set of models to be used in a simulated image, or making a demonstration videotape of a simulated mission scenario.

The distinguishing features that set the Cartographic Modeling Environment apart from more conventional CAD systems include:

- Registration of multiple data sources, including stereographic or multiple images, terrain elevation models, and three-dimensional object models, to the same world coordinate system. This capability is unique in that it permits object model entry to be driven by *sensor* data such as actual images.
- Use of lighting models, terrain elevation data, and other geometric knowledge to constrain and facilitate data entry. The exploitation of *constraints* in the interactive modeling process potentially increases the efficiency of the human operator.
- Registration of local coordinate systems to UTM's, latitude-longitude, and other cartographic coordinate representations. The use of real-world coordinate systems enables the system to exploit specific world knowledge, e.g., by computing the sun position for a particular location at a particular time of day.

The remainder of this document is organized as follows: Section 2 describes the basic facilities that are available in the system for the interactive user; Section 3 gives an overview of the internal structures of the system that would be used by a programmer to build applications within the environment; Section 4 outlines the hardware and software characteristics of the current implementation. The final Section concludes with brief summary.

2 Interacting with the System

The Cartographic Modeling Environment can be used in two very different ways. This Section is devoted to a description of the interactive tasks that can be carried out on the system with little or no reference to the programming environment; no significant knowledge of the computer system itself is required, since all operations are carried out by pointing to the graphics display with the mouse and performing simple menu operations with the mouse and/or keyboard. In the subsequent Section, we describe the programming interface to the system for those who need to go beyond the facilities provided in the interactive user interface.

In this Section, we first discuss the data entry modes and cartographic display capabilities that are supported. Next, we describe the nature of the user interface, the types of data sources used, and the components of an interactive scene view. Finally, we give an outline of some of the interactive feature manipulation activities possible within the system.

2.1 Data Entry Modes

The system supports three major approaches to the entry of cartographic data:

- **Manual.** A human operator can directly interact with the computer system in many ways, ranging from viewing a library of images and sketching cartographic features to creating synthetic scene views. In this mode, the user typically adjusts the position and parameters of a feature using direct visual feedback from the underlying image data sources.

- **Semiautomated.** The computer has the ability to perform certain operations to enhance the human user's effectiveness substantially while remaining under the human's direct, interactive control. Examples of such operations are the display of registered world points on multiple images, no two of which need be fusible as stereographic pairs, the display of illumination rays intersecting any world point, and running local feature-extraction operators that depend upon being given a reasonable starting cue.

- **Fully Automated.** Very few cartographic compilation procedures can be fully automated using present technology. However, as automated techniques mature, it is apparent that they will need to co-exist with the manual and semiautomated compilation methods. Furthermore, even fully automated techniques need an environment in which the human initiator of the analysis process can interact with and select the domains and goals of automated operation. Thus we see our system as a natural framework in which to perform the staged transition from manual to automated computer-based cartographic analysis.

2.2 End-User Data Display Capabilities

Facilities similar to those required by end-users are supported as well. Among the specific viewer-oriented capabilities are:

- **Data Selection.** The user can choose a customized subset of the features available, thereby uncluttering the display to make his task more efficient.

- **View Manipulation.** A data set can be viewed using an arbitrary camera model, whose position, orientation, and internal parameters are controllable as those of an aerial observation platform might be.

- **Creation of Synthetic Views.** Once an area of interest, a data set, and a camera view have been selected, the system can create synthetic views to show the operator how the area might look in real life from the chosen viewpoint.

We remark that softcopy views constructed using such tools have many potential advantages over hardcopy cartographic products.

2.3 User Interface Components

The interactive environment consists of the following components:

- **Graphics Screen.** The graphics screen of the workstation is divided into a number of windows. Each window holds a stack of views, and each view consists of a camera model, an optional image corresponding to the camera model, and a set of cartographic feature databases. Wire frame models of the cartographic features are overlaid on the image, if present, using the associated camera model to project from world coordinates to window coordinates.

  The features themselves are displayable either in terms of wire frames or texture-mapped renderings. The wire frames are easily moved around in real time and have special sensitive regions used as interactive handles for the mouse functions described in the next paragraph. When realistic but static grey-scale images of the scene are desired instead of real-time manipulation, each feature has an alternative display method that renders texture-mapped faces into a depth buffer with antialiasing.

- **Mouse Pointing Device.** A variety of operations are accessible by pressing either the left, middle, or right mouse buttons while some combination of the four state-modifying shift keys (Control, Meta, Super, and Hyper) is held down. Depending on the operation, the movement of the mouse cursor may serve to change a parameter of the object. When an operation involves several steps, a sequence of prompts for mouse and keyboard actions appears on the screen. Two of the mouse operations, the *Create Object* function and the *View Menu*, are specifically for the creation and manipulation of three-dimensional scene structures. The remainder of the operations available directly from the mouse interface, shown in Table 1, are standard ImagCalc operations that deal mainly with the world of two-dimensional screens and digitized image arrays. A superset of the ImagCalc mouse-driven operations can be invoked from an optional pull-down menu bar on the main display screen. These operations are detailed in the ImagCalc manual [Quam, 1988] and will not be described here.

When the mouse cursor touches a sensitive area of an object's interactive depiction, the menu of operations available to the mouse immediately switches to one appropriate to the object. A typical set of such operations is shown in Table 2 for a simple cube-shaped object.

- **Keyboard.** The keyboard is available for entering text information. Typical operations in the interactive mode would involve typing in parameters on the menu attached to a specific object to modify its current state or configuration, or specifying a file name for loading a data set. Simple Lisp operations typed on the keyboard are interpreted in the Lisp Listener window described below.

- **Lisp Listener.** The Lisp Listener is a window in which keyboard input is passed to the Lisp interpreter and executed. Many mouse operations on windows return the data structure pointed at to the Lisp Listener. In this way, one can assign symbols to individual images and cartographic objects to facilitate their later examination and manipulation.

2.4 Information Sources

Among the types of information that the system is designed to handle are the following:

- **Images.** Digitized photographs or similar sensor data from geographic areas relevant to modeling and feature extraction tasks. Associated with the images should be supplementary data such as geographic coordinates, camera models (or equivalent sensor-to-world transformations), illumination information, exact times that allow reconstruction of the sun position for outdoor scenes, sensor characteristics, sensor response information, and atmospheric properties at the time the images were generated. We note that, at the current time, we are aware of no standard sources of data that include the necessary atmospheric or sensor response information; this lack severely handicaps any efforts that might be made to correct the photometry for atmospheric and film processing effects.

- **Cartographic Products.** Existing cartographic products, either in the form of digital feature data or digitized images of hardcopy products, can be used in the same way as images to provide a world-registered context for the entry or editing of additional cartographic information. Existing low-resolution data can be extremely effective in guiding a high-resolution feature acquisition task.

- **Digital Terrain Elevation Data.** When available, terrain elevation data should be registered to the corresponding images. This permits the effective use of three-dimensional constraints during feature entry, the construction of accurate synthetic images, and the generation of cartographic products containing elevation information.

- **Compiled Feature Data.** Cartographic feature data bases are needed for the generation of cartographic products meeting the needs of tasks for which an image alone is insufficient. Ideally, the system should support the interpretation of precompiled feature data bases, the editing of any item in such a data base, and the interactive entry of new data base features.

- **Feature Models.** In order to support the entry of new features, a library of predefined prototype feature models is required. Such models would be used either by an analyst generating a distributable feature data base product, or by an end user updating an existing data base. Feature models must support interactive modes that allow them to be easily moved to correct geographic positions and adjusted with respect to their internal parameters.

2.5 View Components

Each view in the stack on a window consists of a fundamental set of components tying together information sources such as those described in section 2.4. The components are:

- World-to-view coordinate transformation;

- Cartographic object databases;

- Optional image, which must correspond to the coordinate system if present;

- Optional terrain elevation model.

The basic operations available on the view include the functions on the *View Menu*, summarized in Table 3. To begin working with a view, one must either create a bare transform from the *View Menu*, or manually load an image file and set up the required items on the resulting viewpoint in the window. Once a view has been established, one can perform such operations as adding, editing, deleting, and cloning objects, copying a viewpoint to a different window without an image, changing the simulated camera position, acquiring texture maps for object faces, and generating simulated scene views.

2.6 Feature Manipulation

Features are instantiated either by cloning existing features or by selecting from a pop-up menu of feature types invoked by *Create Object*. The object types currently available on this menu are summarized in Table 4. Each feature contains a set of sub-features (such as polyhedral vertices, edges, and faces) that may be sensitive to the mouse pointing device. Whenever the mouse is pointing sufficiently close to a sub-feature, it is highlighted to indicate feature selection, and a prompt window on the screen is updated to indicate the name of the feature and the menu of operations available using the left, middle, and right mouse buttons.

A feature's geometric parameters can be dynamically modified using a variety of operations that exploit continuous mouse-to-screen feedback. For example, one of the operations uses the mouse to drag an object around on the surface of the surface of the terrain model associated with the view. As the mouse changes the position of the object in the window, a corresponding ray from the camera is intersected with the terrain model to determine the world position of the object. All visible views containing the feature are continuously updated to reflect changes in the object's parameters.

There are several types of "utility" objects that provide specialized user interface capabilities. Among these are:

- **Camera model objects.** These are graphical features that allow the user to interactively modify the parameters of a camera.
- **Sun ray object.** The sun ray lets the user control the position of the sun in a view. The ray is adjusted and then set to determine the way in which illumination is taken into account in rendering operations.
- **DTM Mesh.** A mesh representing a small patches of the digital terrain model can be moved interactively around the scene to help the user visualize the local terrain characteristics.

Views with new camera models can be created by copying existing views and modifying their camera models. Initially, these new views have no associated aerial image and only the wire frame depicitions of the feature databases are visible. Images can be generated for new camera models by texture mapping an existing image onto the terrain model and rendering the feature databases.

3 Programming Tools

In order to create new scenarios and test the feasibility of new concepts, one must be able to manipulate and augment the internal system data structures. Here we give a brief summary of some of the structures and the associated programming tools and utilities.

The features manipulated by the system are all implemented as instances of *flavors* in an object-oriented programming environment. This means, in particular, that many distinct object types can share or inherit fundamental data fields and *messages*, which are functions that act intelligently within the data structures of a particular object instance.

The fundamental object types used by the Cartographic Modeling Environment, along with a few of the most critical messages they handle, are summarized below:

- **Viewpoint.** The viewpoint, also referred to in this document as a *view*, is a fundamental ImagCalc flavor that holds the information describing each displayable thing on a window's stack. In ImagCalc, the Viewpoint contains such structures as digital images, graphs, and curve plots. In the Cartographic system, the Viewpoint relates perspective transformations representing camera models to corresponding images, elevation models, and databases of cartographic objects.

- **Image.** An image is a data structure containing digitized image data. The image formats supported include binary images, 8-bit grey-scale images, integer and floating point images, and multiple-band images such as color images. The image object handles the message `:display-image` that chooses the best way to display it on the current window, dithering when necessary. Image pixels can be accessed and altered with the `:iref`, and `:iset` messages.

- **Perspective Transform.** This family of objects includes both orthographic transforms and 4 × 4 homogeneous perspective transform matrices that relate a world coordinate system to a particular pixel in the window. Film digitization parameters are incorporated when the window pixel corresponds to an actual digital image. By sending the messages `:project-to-world` and `:project-to-view` to a transform instance, one can achieve any desired forward or inverse coordinate transformation.

- **Camera-Model Objects.** This object type is the interactive handle by which perspective transform objects (i.e., camera models) can be accessed and modified by the user. It accepts a family of messages supporting the modification of its three-dimensional position and orientation, as well as its focal length and piercing point.

- **Digital Terrain Elevation Model.** Elevation models are arrays of world elevation values in a designated local coordinate system. The model includes a transformation object that translates from the array grid coordinates to the horizontal world coordinates.

- **Two-dimensional Objects.** These are objects such as text that are strictly related to the display window, and move only in the two dimensional window space.

- **Three-dimensional Objects.** This class of objects includes broad families of objects that are displayable and movable in the three-dimensional simulated world. They accept the `:draw-on-view` message to make a wire-frame depiction, and the `:render-on-view` message to generate a texture-mapped view of the object. The two major subclasses in this family are *planar-faced* objects, whose faces are true planes, and *smooth-shaded* objects, whose faces are only arbitrarily-tesselated representations of a smoothly-interpolatable heuristic or mathematical surface.

- **Three-dimensional Curves.** This family of objects corresponds to roads, boundaries, and delineations of various types. In addition to the usual motion and depiction capabilities, these objects can have individual vertices added, deleted, or edited independently.

Rendering facilities form a separate important class of capabilities to which the programmer has access. The operations available here include

- **Buffered line-drawing.** This allows complex objects to be moved with minimal flicker because the erasure step is buffered to occur at the same time as the next draw.

- **Z-buffering.** Three-dimensional scenes may have arbitrarily complex configurations of objects intersecting each other and the terrain. Depth buffering is provided to correctly handle these situations when rendering a simulated scene.

- **A-buffering.** A substantial improvement in the A-buffer approach to antialiasing [Carpenter, 1984] has been developed especially for this system. This facility removes jagged edges and unrealistic artifacts of the uncorrected rendering procedure by computing subpixel contributions of all visible faces affecting the pixel.

- **Terrain Calc Plot.** Texture mapping of an image onto a digital terrain model is handled by the Terrain Calc system. Among the unique techniques incorporated into this system to make the rendering as realistic as possible are the use of a multiresolution image hierarchy and the dynamic selection of local texture map resolutions.

4 Overview of the System Configuration

The current implementation of the Cartographic Modeling Environment is a research-oriented system that is intended primarily to be a tool for feasibility studies, rather than a fully-supported software product.

The system is implemented in a combination of Common Lisp and Zetalisp with Flavors on Symbolics Lisp Machines running Symbolics Genera Release 7 system software. Image data and feature files can be stored on any file system on the network that supports a CHAOSNET file transfer protocol. Support is included for Symbolics black-and-white consoles, for the Symbolics CAD frame buffer systems, and for the Symbolics "Hi-Res" system. The system displays grey-scale images on the black-and-white consoles using dithering, and can display 24-bit color images on color systems with only 8 bits of memory using a color dithering technique. Experimental support is also available for the Tektronix SGS-420 stereographic display system, which is used in conjunction with the Symbolics CAD buffer to provide double-buffered 512 × 512 stereo displays that refresh at 60Hz per eye, 120Hz overall. The system has not been transferred to other hardware and software configurations at this time; the feasibility of such a conversion effort is expected to depend strongly upon the availability of high-performance graphics systems, the development of correspondingly capable window system standards, and the availability of high-performance object-oriented extensions to Common Lisp.

This article is an initial version of the first of three manuals that are being written to document the system. The documentation planned at this time consists of the following:

- **Overview of the SRI Cartographic Modeling Environment.** A brief overview of the system and its capabilities [this document].

- **Users' Guide to Interactive Use of the SRI Cartographic Modeling Environment.** A description of the tasks a system user can perform interactively, using the mouse and keyboard to issue commands.

- **Programmers' Guide to the SRI Cartographic Modeling Environment.** A detailed description of the system's internal structure for users needing to customize the system to their own needs and build new capabilities within the context of the environment.

In addition, a series of videotapes is planned: an overview presentation will illustrate the basic nature and capabilities of the system and will be a companion to the *Overview* manual; a family of shorter tapes dealing with specific applications within the environment will be produced from time to time.

5 Summary

The current implementation of the SRI Carographic Modeling Environment has many capabilities that are of interest for evaluating computer-based approaches to digital cartography. In addition, the system is well-suited for interacting with three-dimensional simulated environments and for performing general computer-aided three-dimensional modeling and rendering tasks. Future development of the system's capabilities will include the following areas:

- Supporting symbolic representations of scene relationships in the style of semantic networks.
- Improving computer-assisted feature entry and constraint exploitation capabilities.
- Extending scene simulation to additional types of sensors.
- Exploring techniques for high-speed manual or semiautomated entry of high-resolution feature data.

Open problems for research in the context of this system include such tasks as simulation of scenes at arbitrary times of day, supporting irregular and topologically complex terrain data, and incorporating the notions of time and physical constraints into the scene generation facilities to support complex animation requirements.

References

L. Carpenter, "The A-Buffer, an Antialiased Hidden Surface Method" Computer Graphics 18, pp. 103–108 (SIGGRAPH, 1984).

A.J. Hanson, A.P. Pentland, and L.H. Quam, "Design of a Prototype Interactive Cartographic Display and Analysis Environment," Proceedings of the Image Understanding Workshop, pp. 475–482 (February 1987).

L.H. Quam, "ImagCalc Users' Manual," SRI Technical Manual, 1988.

references

Shift Keys	**Mouse Left**	**Mouse Middle**	**Mouse Right**
none	Select	Show Selected	Menu
C	% Reposition	Zoom In	Zoom Out
M	Recenter	Copy To Here	Move To Here
MC	Cycle Stack	Kill Top	Kill Stack
S	X-Slice	Y-Slice	Histogram
S C	Set Tandem	Show Tandem	Clear Tandem
SM	Fast Scroll	Fast Zoom In	Fast Zoom Out
SMC	Rev Cycle	Describe	Pop Multiple
H	Window	Scroll Window	Magnify Pane
H C	Print Window	Plot 3d	Inspect Stack
H M	Scale Rotate	Stretch	Negate
H MC			
HS	Flip-X	Flip-Y	Rotate
HS C	Recenter Others	Create Object	View Menu
HSM			
HSMC	Edit Viewpoint	Push Displayed	

Table 1: The view operations available directly from the mouse when no object is selected. To get access to the functions on each line, the corresponding state-modifying keys must be held down while the mouse buttoned is depressed. The symbols are *C* for Control, *M* for Meta, *S* for Super, and *H* for Hyper.

Shift Keys	Mouse Left	Mouse Middle	Mouse Right
none	Move UV @Z	Move Z	Menu
C	Move UV on DTM	Move W	Move Z
M	XY Sizes	XY Rotation	Z Size
MC	Drop Z	Sun Z	Drop W
S	UV Roll	Reorient	W Rotation
S C	Set Texture	Remove Texture	
SM	Azimuth-Elevation	Z Rotation	Z' Rotation
SMC	Set Face Texture	Remove Face Texture	
H	Clone	Delete	Drop/Obj
H C			
H M	Open Vertices	Close Object	Change Superior
H MC			
HS	Blank	Render	Refresh Views
HSM			
HS C			
HSMC			

Table 2: The functions available directly from the mouse when a cube object is selected. To get access to the functions on each line, the corresponding state-modifying keys must be held down while the mouse buttoned is depressed. The symbols are *C* for Control, *M* for Meta, *S* for Super, and *H* for Hyper.

Database Operations	View Operations
Configure Databases	New View Transform
Add a Database	Render Terrain
Add All Databases	Render Objects
Remove a Database	Refresh Window
Sensitize a Database	
Hide a Database	
Expose a Database	

Table 3: *View Menu.* The model-oriented operations performable on a view.

Flat-Surface Objects	**Curve-Based Objects**
Box	Flight Path
Building	Open Curve
House	Closed Curve
	Ribbon
Curved-Surface Objects	**Mensuration Objects**
Half Cylinder	Crosshair
Cylinder	Epipolar Crosshair
Superellipse	3D Axes
Superquadric	Ruler
SuperSketch Object	
Utility Objects	**Group Objects**
3D Text	Composite Object
DTM Mesh	Cartographic Database
Camera	
Sun Ray	

Table 4: *Create-Object Menu.* The classes of objects that may be instantiated.

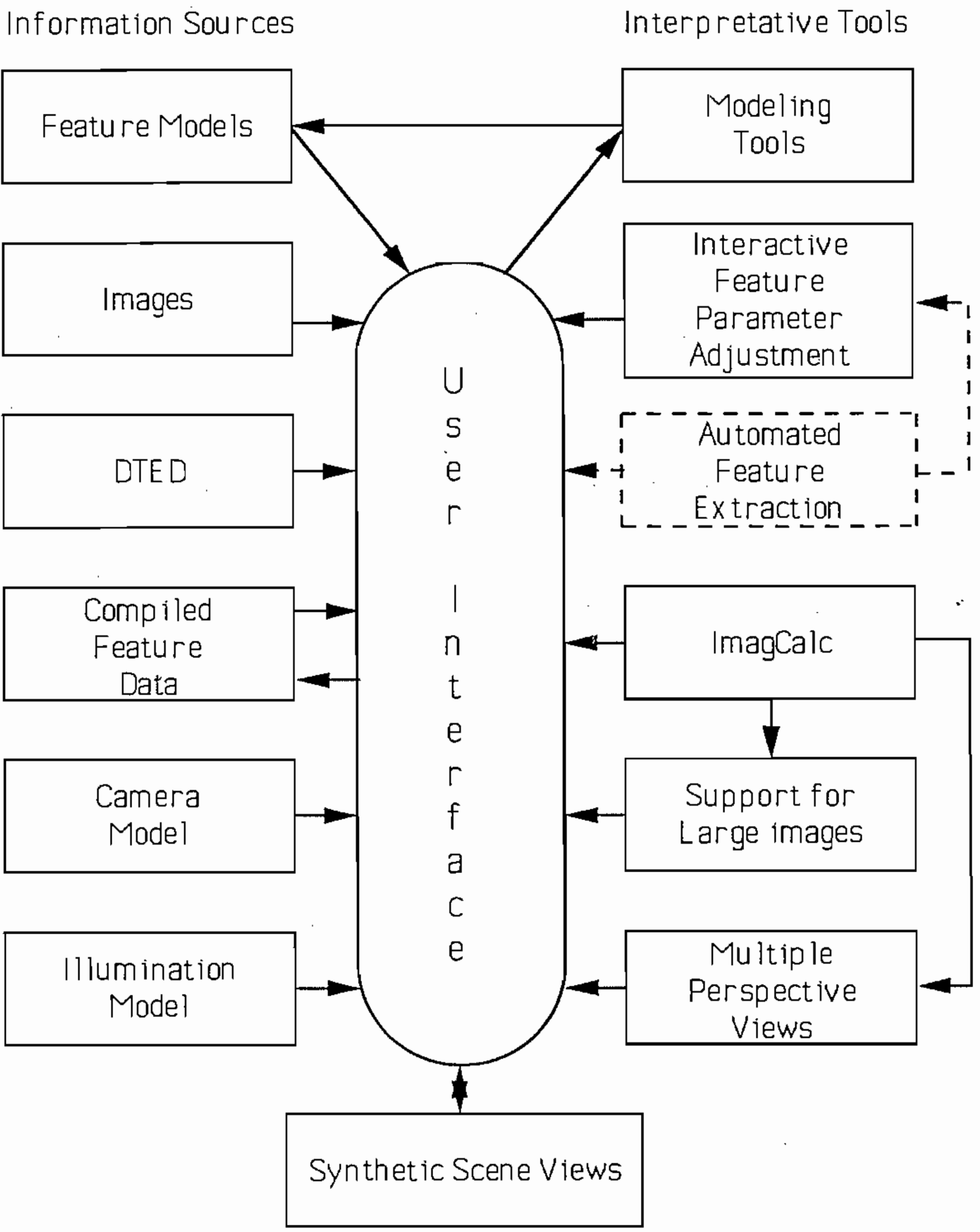

Figure 1: Block diagram of the major components of the SRI prototype cartographic analysis and display system.