AD-A276 125

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY
2. REPORT DATE: Dec. 29, 1993
3. REPORT TYPE AND DATES COVERED: Final Report

4. TITLE AND SUBTITLE

SBIR A-93-030
Texture Library for 3-Dimensional Visualization Systems

5. FUNDING NUMBERS

DACA 76-93-C-0010

6. AUTHOR(S)

Roy Latham, Dave Marsh, P.Y.Cheng, Stuart Wells

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

C G S D Corp.
2483 Old Middlefield Way, Suite 140
Mountain View, CA 94043

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

U.S. Army Topographic Engineering Center

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

A texture library is sought to serve DoD visual simulation training needs as well as civilian commercial applications. The design effort of the present study encompasses determination of pattern requirements, identification of methods for pattern generation, establishement of image generator requirements, identification of methods for categorizing patterns, and establishment of methods for compressing and distributing patterns.

14. SUBJECT TERMS

Texture, image generators, texturing methods, visual simulation, simulation databases, compression.

15. NUMBER OF PAGES

55

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT

NSN 7540-01-280-5500

Standard Form 298 (Rev 2-89)
Prescribed by ANSI Std Z39-18
298-102

REF 9

Final Report

Texture Library for 3-Dimensional Visualization Systems

DACA76-93-C-0010

revised December 29, 1993

Prepared for:

U.S. Army Topographic Engineering Center
Contract No. DACA76-93-C-0010
Project No. W26HAJ-3096-2018
SBIR A93-030
Texture Library for 3-Dimensional Visualization Systems

CGSD

Computer Graphics Systems Development Corporation

2483 Old Middlefield Way #140
Mountain View, CA 94043-2330
415-903-4920
fax 415-967-5252

Table of Contents

List of Figures

List of Tables

1. Executive Summary

Images of terrain and other features of the environment are created in real time or near-real-time to support mission planning, mission rehearsal, simulation training, and simulation for system development. The cost of the hardware for high performance visualization is dropping at a compound rate of twenty to twenty-five percent per year as a result of general advances in computer and semiconductor technology, with the result that systems are being put to increasingly widespread use. For example, networked simulations like the Army's Close Combat Tactical Trainer (CCTT) will allow hundreds of participants to conduct tactical training in a common virtual environment.

The environments for visualization systems are constructed off-line to suit the purposes of the application. Methods for constructing databases have improved over the years, but it is safe to say that economies in database construction have not kept pace with other declining system costs. Databases are digital representations of the environment made from polygons and texture patterns. Polygons describe the three-dimensional structure of the environment, while texture patterns provide surface details. Stored texture pattern images are applied to polygon surfaces by the visualization system much like wallpaper is applied to walls.

Texture patterns, however, can include transparent cut-outs, so that outlines of objects (like trees and bushes) can be formed from polygonal structures. Generic repeating patterns are applied to terrain surfaces according to the appropriate character of the terrain.

Having a library of patterns for both objects and terrain surfaces can reduce the costs of developing databases by allowing patterns to be reused among many applications. This study has determined that a library of approximately 950 basic patterns will serve a majority of the requirements for generic patterns in the applications cited above. Non-library patterns, i.e. aerial photographs of specific locales, will still be required for some mission planning and a small fraction of simulation training requirements. Even in those applications, library patterns will be useful for three-dimensional objects and for *microtexture,* generic details of surfaces beyond the resolution of aerial photographs.

Among the specific defense applications which could use a texture library, we note the upcoming series of Combined Arms Tactical Trainers, of which CCTT is the first and all of which will require databases. We note that the Topographic Engineering Center has the responsibility for generating follow-on CCTT databases and could use a texture library directly in support of that task.

The study revealed strong advantages of library patterns beyond cost savings. For a networked simulation, having consistency of texture patterns is important. Consistent patterns will help ensure that simulators made by different manufacturers have comparable abilities to support target acquisition. The approach derived in the course of this study ensures that patterns in a library will consistently track real world colors and contrasts, and that the pattern data provided can be related to color standards so that a user can correctly calibrate a visualization system to achieve consistency with other users. The method does not attempt to provide absolute color matching of every pattern to the real world; it does provide near-absolute consistency among patterns and good matches to the real world.

Having a library of patterns also facilitates rapid construction of databases. The recommended approach to library construction facilitates fully automatic construction of databases by allowing the user to access the pattern library directly by computer using codes contained in the Interim Terrain Data format, along with information concerning the locale, the

time-of-day, and the time-of-year of the environment to be visualized. Direct computer access to the library is proposed to be supplemented by a separate human interface that provides browsing and relational database searching according to a wide assortment of descriptors.

As one might suppose, the study confirmed that it is a good deal more efficient to build a texture pattern library than to acquire patterns one at a time. Efficiencies derive from

- Locating source data (for terrain, most patterns are best derived from low altitude high resolution aerial photographs indexed by the USGS)
- Amortizing the costs of tools for image manipulation, pattern synthesis, compression and retrieval
- Specialization and training of staff
- Economies of scale in indexing and distribution

In our conception of a texture library, the user would be provided with

- Patterns: The actual pattern data stored in a prescribed format
- Descriptors: The information stored with the pattern that describes what it is
- An Index System: A cross-reference directory that allows a human user to find and retrieve patterns from the library
- A Computer Access System: A package of routines for integration into the user's software to permit automatic retrieval according to feature codes

The library will be distributed on a standard-format CD-ROM. The user would copy access programs off the CD-ROM to a personal computer for interactive searching. Routines are also provided for downloading and automatic retrieval from PC, Sun, and SGI computers. Data on the CD-ROM is stored with colors in the CIE standard color coordinate system. Patterns are compressed using JPEG, which we found to be best for the low compression ratios and high fidelity required for this application. The disk storage mechanisms will be transparent to the user, who can download patterns in a variety of selectable formats and color spaces, with the conversion managed by the access package.

There are no significant technical risks in the program, because we rely upon off-the-shelf products (like Kodak Shoebox for indexing) and upon other technology with which we have had substantial experience (CGSD produces a commercial software package that does all of the necessary color transformations.)

In what follows, we describe each aspect of the library system.

2. Requirements

Requirements for the texture library come from both the simulation and the mission rehearsal communities. The discussion below addresses those needs.

2.1 Training Simulation Texture Requirements

CGSD personnel have been involved with visual simulators since the early days of image generator technology in the 1970's, and we remain actively involved through our consulting operations and publications. The simulation industry has a continuing need for texture patterns to support visual and IR simulation. Standard colors and textures which will be produced by the texture pattern library would treat interoperability problems that currently exist between simulators for target detection ranges. Standard texture patterns would support upcoming I/ITSEC demonstrations as well as provide support for follow-on CCTT database construction planned by the Topographic Engineering Center.

Most current simulator contracts involve the generation of texture patterns to support image generation requirements. Theoretically, each contract could require the same texture pattern. With a texture pattern library, the savings are obvious. These benefits, in addition to the benefits identified below for mission rehearsal, should provide a ready market for the texture library.

2.2 Mission Rehearsal Texture Requirements

We have contacted more than a dozen individuals at organizations involved with mission rehearsal. We discussed with them their needs for texture to support mission rehearsal activities. What we have learned about mission rehearsal needs is generally in accord with what we know the needs to be in the simulation community. This is not surprising since most of the same visual cues are logically required by both groups. Indications are that rotary wing-based mission air crews tend to have the greatest need for the texture library. The fixed wing air crews fly relatively high and fast and perform adequate mission training utilizing satellite or aerial photo-originated imagery.

Rotary wing air crews, however, fly relatively low and slow so texture on objects provides speed and altitude cues necessary to perform their mission. This need is most acute when they near the target area where they hover and land. Terrain texture and texture on 3-dimensional objects provides important cues necessary for landing and low altitude maneuvering.

The Special Operations Forces (SOF) is trained to perform several missions: ordinance delivery, troop insertion/retrieval, and hostage rescue among them. SOF needs realistic real-world databases to use for mission rehearsal and they need them to be ready in 48 hours. A set of library textures, properly indexed, linked to simulation database generation software, and readily available will help simulator database personnel meet this requirement.

3. Texture Pattern Production Overview

We recommend providing users with two separate methods for obtaining patents from the library: a software-indexed interactive search method and an automatic method enabled by routines we provide for incorporation into the user's database generation software. The two methods are illustrated schematically in Figure 3-1.

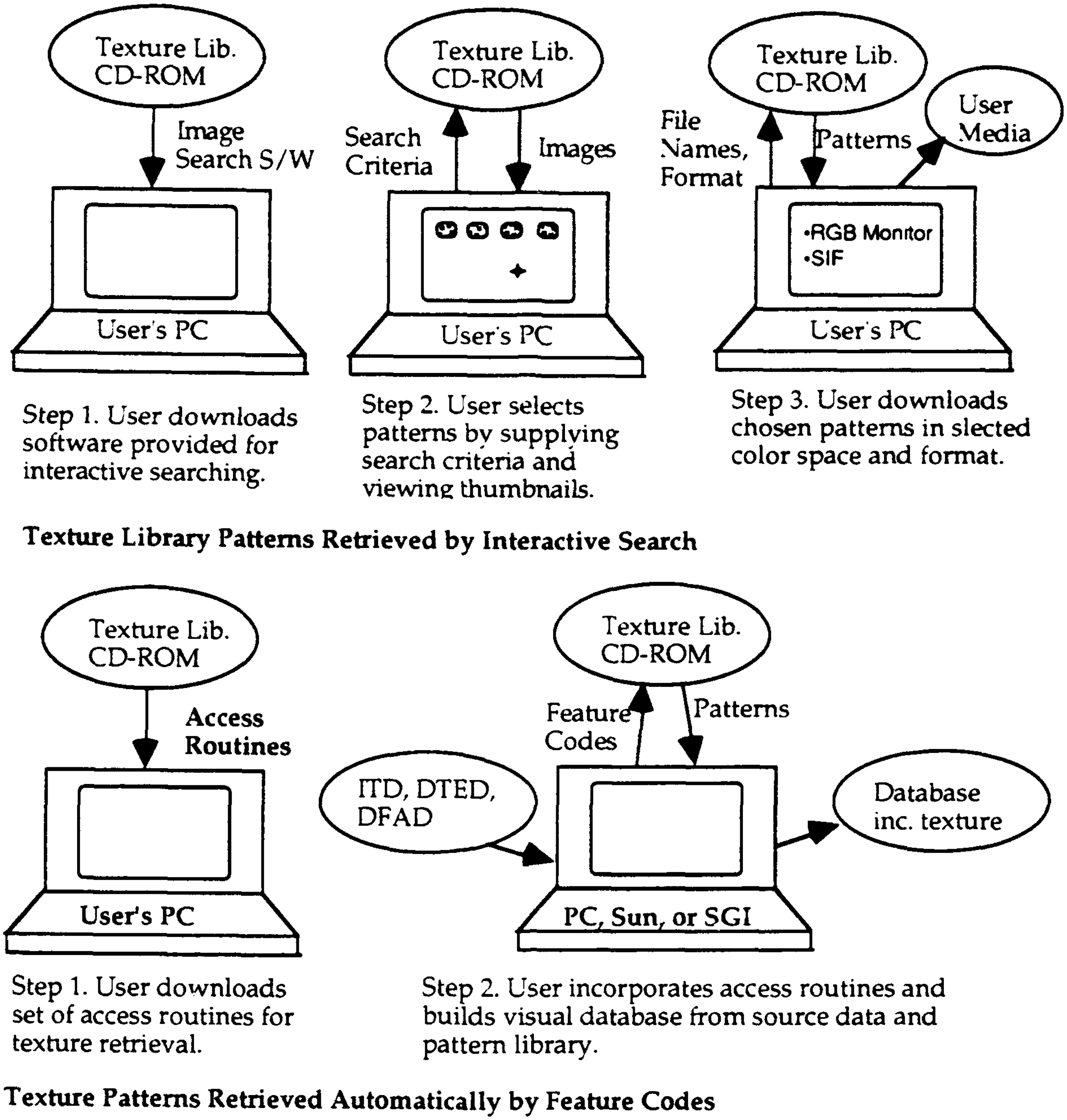


Figure 3-1. Accessing Texture Library Interactively or Automatically.

The process of producing texture patterns and making them available on CD-ROMs is somewhat involved. Figure 3-2 presents a functional overview of the process and will provide a useful reference when reading the remainder of this report. The entire process will utilize commercially available IBM PC/clone computers and peripherals. The following paragraphs present a descriptive overview of the process.

The process of making the library starts with a list of needed texture patterns. The lists of patterns we plan to implement are shown in Appendices A (visual texture patterns) and B (infrared texture patterns). These lists are discussed in detail in Section 4. Once the list of desired texture patterns is available, the Interim Terrain Database (ITD) will be searched to identify occurrences of the desired patterns. The search software we plan to write will allow searches based on a feature code or on an attribute code and value. A typical search might be used to locate a "swamp, mangrove" using the *feature code, attribute number,* and *attribute value.* Such a search could be further limited by specifying the *density measure* and the *undergrowth density category* desired.

Using the geographic location obtained from the ITD, we will purchase an aerial photograph or digitized photographic data of that location. Depending on the source data obtained, we will decide which texture pattern production technique is most appropriate for this particular texture pattern. Photographic data or digitized photographic data will allow us to produce patterns based on real world images. We expect it may, however, be impossible to obtain photographic imagery of some desired patterns. In that case, it may be more appropriate to generate some texture patterns using one of several synthetic means we have studied. It is also possible that, in some cases, more realistic patterns can be generated using synthetic techniques. If synthetic generation is used, edge matching processing must be applied to eliminate obvious seams which can occur when placing synthetically generated patterns side by side. We will make the choice on texture pattern generation technique on a case-by-case basis.

Different versions of texture patterns may be used to represent different times-of-day, times-of-year, and infrared. In some cases additional versions of a texture pattern will be produced by making modifications to the original version. Such might be the case for a texture pattern for grass. In California for instance, a grassy hillside in the spring is typically green. Later in the year the grass is brown due to the lack of rain in the summer months. In this case, a simple change to the color of the texture pattern would produce the second version.

The texture patterns discussed so far will be produced at their most detailed level; full resolution texture patterns. These full resolution patterns will now be processed to produce a number of lower Levels of Detail (LODs) for each pattern. Typically, the full resolution texture pattern will exist in a 512 x 512 array. To save image generation processing, lower LODs are used by most image generators. The lower resolution patterns we will produce include resolutions of 256 x 256, 128 x 128, 64 x 64, . . . , and 1 x 1.

The 512 x 512 size was selected as being the largest size anyone would be likely to need. It is currently supported by a number of IG manufacturers, and most vendors can use it as four contiguous maps. Typical sizes are now smaller, but we are concerned that the library meet the needs of virtually all users for many years, as trends continue to larger and larger patterns. For a specific object such as a tree, 512 x 512 only provides full resolution for less than a quarter of a workstation screen -- which does not seem excessive. It costs relatively little to provide larger patterns at the outset, but it would be expensive to go back and try to get them later.

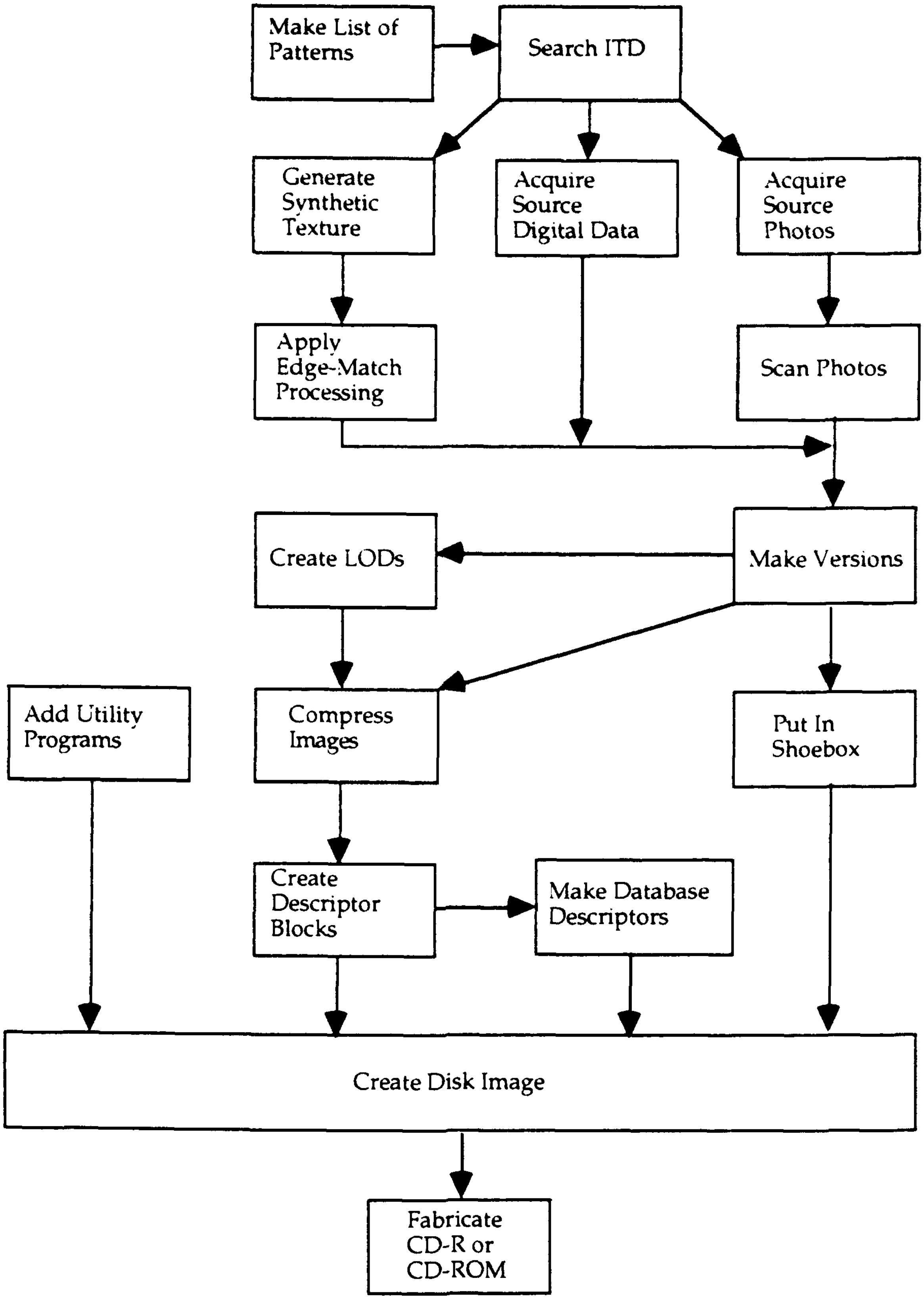


Figure 3-2. Texture Library Production Process

To save space in the master texture library and on the distribution media, the full resolution texture pattern along with all LOD images will be compressed using the JPEG standard method. Descriptive information, including all the ITD attributes of the pattern which will facilitate pattern searching, along with additional information used to maintain the texture library and provide a record of the source of each pattern are added to the texture attern file. The texture patterns along with attributes of each will also be entered into the Kodak Shoebox Image Manager software which will produce "thumbnail" images of each texture pattern to allow interactive browsing and search of the library.

A set of access subroutines which we provide to users will allow the user to automatically access the library by adding the routines to the user's software. By using these routines in conjunction with his own database generation software to automatically add texture patterns to each of the database polygons based on the feature codes and attributes of the features being modeled. The routines will allow the user to download the texture patterns in one of several popular data formats, including SGI, JPEG, TIFF, or SIF.

In the library, the texture pattern color information will be stored in CIE coordinates. The access routines will allow the user to convert the CIE color coordinates, as part of the downloading process, to RGB or other color coordinates which the user may need.

The compressed images, descriptive information, Kodak Shoebox Image Manager software, and the utility programs will be written on to the master texture library optical disk. This master disk will be kept at CGSD. At any instant, it will contain the most current texture library. When a request for a copy of the texture library is received, the contents of the master texture library optical disk will be transferred to a recordable compact disk (CD-R), using a recordable CD-ROM drive. The CD-R approach is well suited to the texture library application. It allows copies to be made on short notice using the latest texture pattern data.

The CGSD texture library approach provides government and industry with a comprehensive set of texture patterns. It does so in a way that makes copies of the most current library contents readily available and provides the user with a flexible search and download mechanisms. Key elements of our approach are:

- A library structure that will support a large library of texture patterns, including many variants for visual and infrared, time-of-day, time-of-year, and weather variations
- Microtextures to support multiple levels-of-detail (LOD)
- Header files that fully characterize the texture pattern to facilitate automatic search
- Flexible search criteria, including pattern name, material codes, colors, and appearance
- Use of commercial off-the-shelf (Kodak) software for interactive browsing and pattern search on a PC or PC clone computers
- Separate software access engine, a library of routines provided to the user that may be compiled and run on more than one specified platform with specified operating systems to access the library
- Texture patterns keyed to Interim Terrain Database (ITD) data types
- About 1000 texture patterns identified for inclusion in Phase II
- Compression for distribution using JPEG, a widely used standard method
- Distribution on Compact Disk - Recordable (CD-R) in standard format readable by CD-ROM drives on any computer
- Selectable output formats so the data can be downloaded to suit the user's preference for color space and file format

- Color coordinate conversion software included, so that patterns distributed in universal color coordinates may be adapted to any display
- A library format that can be expanded and upgraded at any time

4. Texture Patterns

A texture pattern is an array of numbers that represent the characteristics of a surface. Each point in the array corresponds to a sample of the surface properties. The samples are called *texels*, and the array of texels is called a *texture map*, a *texture pattern*, or simply a *pattern*. Each surface property provided in a texel is a texture *parameter*.

The concept of "surface properties" covers a lot of ground. The most common example of a surface property is the color of the surface, so that the array of samples tells how to color the surface in a graphics rendering. We therefore begin with color and proceed with other surface properties, including transparency, sensor parameters, microtexture, and advanced properties. Like visual imagery, sensor imagery is affected by time-of-year and time-of-day. Under sensor parameters we present the concept of versions for time-of-year and time-of-day and the effects on sensor imagery. Finally, we discuss the level-of-detail hierarchy of patterns and summarize the results for the overall data structure.

4.1 Color

Texture patterns will be derived from many different sources. There are substantial variations in the fidelity of the colors from different sources, depending on the original film, print, or sensor characteristics, how old the media is, and the conditions under which it has been stored. We want to correct for the differences among source material colors so that users can produce images that appear consistent. In some circumstances, we will have to apply false coloring to monochrome originals simply because source material in color is relatively uncommon compared to monochrome originals, and applying artificial coloration will be the only practical way to obtain certain patterns or pattern variants. A means must be found to ensure that applied coloring matches other colors. Uniform color characteristics are especially important for simulation and training applications because the ability to acquire targets depends critically upon target to background contrast. Inconsistent color or contrast among the simulation systems used in a networked exercise can give some players an unintended advantage in target acquisition, thereby invalidating the simulation results.

In meeting the requirements for uniformity of color and contrast among texture patterns, the primary concern is that they be uniform from pattern to pattern and user to user. The human visual physiology has a built-in mechanism for correcting for difference in overall variations in color, but it is acute in detecting side-by-side differences. Therefore it is more import that we have consistent colors than colors that exactly match the real world. However, it turns out to be a reasonable proposition to get a good match to the real world as well, using a technique we call *color cataloging*.

The color cataloging concept is inspired by the way the print media achieves reasonably accurate color. Specialists perform color correction using CRT displays. Remarkably, they do not use color displays. Instead, from experience, they know what ratio of the four colors of the print separation (cyan, magenta, yellow, and black) should be used to achieve pleasing results for certain objects or surfaces. For example, a major concern in much of the media is to make sure flesh tones appear reasonably correct - blue, green, or purple tints would be unacceptable. The color specialist knows the ink combinations that correspond to a great variety of different flesh tones, and makes sure that the picture is adjusted overall so that at least those critical values will come out acceptably. When those adjustments are made, other values in images fall into reasonable ranges as well.

We can use the general approach of knowing the true colors of a limited number of objects in the texture source materials, and correcting all the colors of the image in such a way that the known colors are correct. For example, suppose we have in our color catalog the true colors of the maple, oak, and pine tree foliage in mid-summer. An aerial photograph that includes trees of this type can then be corrected so that the particular colors in the catalog come out correctly, and we will assume that other colors in the images will then be reasonably correct.

How do we know that the particular oak trees in the aerial photographs actually match the generic colors in our catalog? It turns out that the colors of foliage in nature actually fall within a rather small part of the color spectrum, a result obtained in measurements done fifty years ago, and which is not difficult to verify. Secondly, we are not concerned if there are small variations between the real world and our catalog assignments. Even if the match is not absolute, we will achieve our primary goal of consistency, i.e., all the oak trees colors will match, and we will achieve our secondary goal of having reasonable, if slightly imperfect, correspondence to the real world.

For the suggested color matching approach to work, we must have a reasonably complete color catalog. Only a limited amount of absolute color data is available. For example, for our Color Science Library product, we computerized the National Bureau of Standards (now National Institute for Standards and Technology) Dictionary of Color Names which provides a starting point for quite a few materials. However, we will need to supplement the catalog with additional field measurements. These measurements are easy to make either by instrument or by comparing samples of standard colors to objects.

Before continuing with an example of the color cataloging process, let us first discuss the problem of describing and storing accurate colors. Most graphics systems compute images in terms of the red, green, and blue color components corresponding to the colors of the three phosphors of a cathode ray tube display. However, the colors of phosphors are only loosely standardized. In particular, the "official" standard phosphor colors established for broadcast television are rarely used today because more color-saturated phosphors are now available. Also, we now have a variety of display devices, including LCD flat panels, light valves, field-sequential displays, laser projectors, color printers, and film recorders, none of which have phosphor-related color primaries.

One can generally transform colors specified in phosphor RGB components to the systems used by other displays, except that some colors are out-of-gamut, i.e., they are not reproducible in one system or the other. For example, CRT colors are not completely saturated. The green phosphor is kind of a pale yellowish green compared to the brilliant green of a film image, or even more extreme, a laser projector. So if we build a texture library around the colors of phosphors, users lucky enough to have displays capable of more saturated colors will never have patterns that fully utilize the capabilities of their system.

The solution is to store colors in CIE XYZ coordinates, the International Standard that encompasses all visible colors, and therefore the capabilities of all display devices. There is a variation of CIE coordinates called xyY, where Y ("big Y") is the intensity and (x,y) are the color coordinates. The XYZ and xyY forms have the same information, but the xyY is more convenient for dealing with color and intensity as separate parameters.

Only a few users will feel comfortable dealing with xyY color coordinates. Consequently, we must provide the conversion to RGB and to other color spaces for other devices as part of the library downloading process. The general business of color space conversion is a complicated matter, but we have already developed a clean solution in our Color Science Library, a package of C-language subroutines CGSD has developed and sells as a commercial product. To illustrate how the color cataloging and conversion process works, we build a small sample.

The sample was built by matching real world objects to color samples in a Munsell Book of Color. We then used the Color Science Library to convert the data to various color spaces, including RGB and xyY. We also used the Color Science Library to find the NBS Color Name, derived from a system of stylized naming conventions that help identify colors intuitively with a measure of consistency and precision. The results are shown in Table 4.1-1 below.

Table 4.1-1 A Sample Color Catalog

Object	Munsell Color	NBS Color Name	RGB (monitor)	xyY
Decking, moderately weathered redwood	10YR4.5/4	moderate yellowish brown	35, 43, 74	0.4085, 0.3891, 15.56
Flower (bract), bougainvillea	10RP 4/12	strong purplish red	97, 3, 35	0.4789, 0.2717, 12.00
Flower, "potato plant"	2.5P 4/10	strong violet	41, 18, 84	0.2619, 0.1903, 12.00
Flower, daisy	5Y 8.5/12	vivid yellow	232, 174, 0	0.5035, 0.4745, 68.41
Flower, rose	5R 4/16	vivid red	110, 0, 2	0.6039, 0.2978, 12.00
Foliage, agave, leaf center	5GY 4/4	moderate olive green	27, 35, 9	0.3538, 0.4284, 12.00
Foliage, agave, leaf edge (10% of area)	5Y 8.5/8	light yellow	218, 176, 29	0.4117, 0.4347, 68.41
Foliage, bamboo, green	5GY 5/6	moderate yellow green	42, 60, 9	0.3663, 0.4614, 19.77
Foliage, bamboo, yellow	10Y 8/10	strong greenish yellow	168, 164, 3	0.4190, 0.4790, 36.17
Foliage, cedar tree	7.5GY 5/4	moderate yellow green	42, 52, 29	0.3274, 0.3994, 19.77
Foliage, common fig tree, autumn	5Y 7/10	strong yellow	93, 43, 16	0.4509, 0.4696, 43.06
Foliage, myrtle tree	7.5GY 3.5/3	moderate olive green	19, 24, 14	0.3235, 0.3912, 9.0
Foliage, oleander	5GY 4/4	moderate olive green	27, 35, 9	0.3538, 0.4284, 12.00
Foliage, olive tree	5GY 6.5/5	moderate yellow green	91, 101, 29	0.3661, 0.4192, 36.17
Fruit, banana	7.5Y 9/10	brilliant greenish yellow	231, 205, 13	0.4201, 0.4622, 76.54
Fruit, persimmon	2.5YR 6/14	vivid orange	193, 37, 2	0.5488, 0.3922, 30.05

Grass, lawn 1	5GY 4/4	moderate olive green	27, 35, 9	0.3538, 0.4284, 12.00
Grass, lawn 2	5GY 5/6	moderate yellow green	42, 60, 9	0.3663, 0.4614, 19.77
Paving, concrete conglomerate	2.5Y 7/3	grayish yellow	137, 105, 69	0.3600, 0.3655, 43.06

Two objects, the rose and daisy flowers, had colors too vivid for a standard workstation monitor to reproduce. The closest RGB approximation, in each case, has one of the color components zero, a sign of an extreme case of color saturation. The rose, in fact, is outside of the range of color chips reproduced in the Munsell Book of Color, and we had to guess the Munsell coordinates by extrapolating the saturation an extra step. The CIE xyY coordinates, however, are defined for all visible colors and describes the flower colors accurately. Although we do not know how many out-of-gamut colors will occur in the real texture library, the utility of using xyY coordinates for the library storage is demonstrated.

Note the color similarities of the various green foliage types. Three of our small sample of foliage types turned out to match the Munsell 5GY 4/4 color chip. Other types are not too different. We suspect the close grouping of the colors has something to do with the color of the common ingredient -- chlorophyll. In any case, the similarities mean that we will have a much easier time correcting the colors of aerial photographs than we would if the variations were greater, because it is easier to identify a cataloged color in each image. Even if a foliage type is misidentified, the error would be small.

Overall, the table shows why CIE xyY coordinates, despite their universal applicability and precision, are not readily interpreted. The xyY coordinates of the lawn, for example are (0.3663, 0.4614, 19.77) which means little to most people. The RGB coordinates (42, 60, 9) are easier to interpret, but the NBS color name "moderate yellow green" eliminates most of the mystery. Thus, keeping the NBS name with each designation is a good way to facilitate common sense checking for errors. (Incidentally, the Y coordinate in xyY has an easy interpretation: 19.77, for example, means that the surface reflects 19.77% of the light that falls upon it. The Y coordinate is therefore the intensity element of the monochrome version of the pattern.)

CGSD's Color Science Library is, to our knowledge, the only library of subroutines commercially available to perform the color conversions shown in the sample color catalog. Traditionally, the conversions have been done interactively with reference to published tables.

Providing data in standard universal color coordinates does not guarantee that the colors will appear the same in the user's display device, but it is a necessary precondition. The users must correctly calibrate their local processing and display system to preserve the color accuracy.

4.2 Transparency

Along with color, we must include transparency, sometimes called *alpha*, as a surface property. Transparency is most often used to cut out irregular shapes to represent tree foliage, but it is also used for cloud layers, for smoke, and occasionally for windows.

4.3 Sensor Parameters

Another important surface property, it turns out, is temperature. Military users of the pattern libraries will want to use them to simulate operations with infrared sensors that make images of the heat patterns. At this point, perhaps half of simulation training is done using sensor imagery rather than visual simulation, so it would be a major omission to leave out IR patterns. On the other hand, temperature patterns depend upon the heat characteristics of each scene element and the time history of heat applied to or generated by each element. A thorough study of IR simulation taking all the variables into account is way beyond our scope.

A reasonable objective is to take into account the four seasons of the year and four times-of-day for each season. We need to represent the four seasons for visual patterns as well. The four times-of-day we have selected for the IR patterns are 0400, 1000, 1600, and 2200. These are key points in the daily heating and cooling cycle, so that a user desiring to do so could reasonably interpolate patterns between these points. Weather effects like wind and rain lower the temperature differences more rapidly that the cooling cycle for a clear day, so we will let the user lower the contrast to reflect weather conditions. Finally, patterns are affected by climate, because, for example, in the Northern Hemisphere the temperature of the ground stays colder in northern regimes than in southern. We will simply assume the climate that is most ordinarily associated with a pattern, like the temperate zone for oak trees and the tropics for palm trees.

4.4 Microtexture

Another surface property we want to include is an index to a microtexture pattern. A microtexture pattern is a more fine-grained pattern used to fill in detail when the main pattern runs out of detail. For example, a geospecific pattern derived from an aerial photograph might have a highest-resolution of one meter. An observer close to the pattern would ordinarily see no detail below that resolution; the pattern is smoothly interpolated between the texels. Microtexture is a pattern on a finer scale that is used to provide generic detail at resolution less than the main pattern can support. Microtexture patterns include grass, sand, pebbles, and road surfaces.

To implement microtexture, the main texture pattern includes an index number to a table of microtextures. We will allow 256 microtextures to be referenced from a main texture pattern by storing an eight-bit code as one of the texel parameters. A table of the codes in the description of the pattern would then give the full reference to the microtexture. For example, a photographically derived pattern might have the code 153 stored in one of its texels. Code 153 in the descriptor table would reveal the location of the microtexture pattern for "coarse grass."

Microtexture patterns are currently implemented in real time image generators manufactured by Silicon Graphics, Inc., by Martin Marietta (formerly GE Aerospace), and others. It should be easily accommodated by software-based mission rehearsal systems.

4.5 Other Parameters

There are many other surface properties that could be included in texture patterns. Some computer graphics software supports "bump mapping" which uses a pattern of surface orientation data to simulate bumps on a surface for different lighting conditions. Others support reflective properties, for shiny or highlight effects. We will rule out all these properties from our library, at least for the time being, because we do not expect anyone in our intended group of users to want them. We will leave an extension mechanism open for the future.

Some users may want monochrome visual patterns rather than full color. Generally, if we are storing color in XYZ we will convert the three color components to an equivalent monochrome reflectivity when the user requests a monochrome version to be downloaded. If storage is in xyY, then a conversion will be made to three color components for downloading.

4.6 Level-of-Detail Hierarchy

Nearly all graphics systems using texture patterns require a hierarchy of pattern resolutions, called a *zoom pyramid* or sometime, erroneously, a *MIP-MAP*. (A MIP-MAP is really a scheme for computing the storage address of a zoom pyramid, not the pyramid itself.) Each level of the zoom pyramid is computed from the previous higher-resolution level. Each level has one-quarter the number of texels as the previous. The lowest resolution of the pyramid has only one texel, the average of the whole pattern. When rendering an image, the different levels of the pyramid are used to prevent aliasing; polygons viewed distantly or at oblique angles receive lower-detail versions of the pattern.

We could require users to compute the lower-resolution versions of the pattern from a single version supplied in the library. However, we will include all the versions in the library anyway. First, including all the versions takes only one-third more disk space than just the highest resolution; the quartered sizes yield $1/4 + 1/16 + 1/256 + \ldots = 1/3$. Second, we want to allow every user to have available identically-computed versions of the pyramid so that they can be matched across a networked simulation, if desired. There are many ways of filtering the data to get from one level of the pyramid to the next, and just averaging four neighboring points is not the best way. Third, computing the pyramid can be time-consuming, and since the storage burden is not too great we might as well save the user computation time. And finally, if using the library solves many problems, we may in the future have users who do not have the software to compute the pyramid. So all things considered, we might just as well include it.

4.7 Summary

In summary (see Figure 1), visual patterns have up to five parameters: three color components, transparency, and microtexture code. Infrared sensor patterns have up to three parameters: temperature, transparency, and microtexture code. Both visual and sensor patterns have up to four versions, corresponding to each of the four seasons, and sensor patterns have versions for four times-of-day for each of the four seasons. All patterns have versions for each level-of-detail in the zoom pyramid. The format of the texture pattern database provides for all versions of the texture patterns. We recommend including the texture patterns selectively.

A list of visual texture patterns is provided in Appendix A. It contains over 400 texture patterns for fixed objects plus 256 texture patterns to be applied to moving objects such as vehicles (aircraft, tanks, ships, and personnel). The list was derived from a combination of sources: MIL-I-89014 (Military Specification, Interim Terrain Data (ITD)/Planning Interim Terrain Data (PITD)), United States Geographic Service (USGS) photographs, books having photographs of military vehicles, and our experience with similar databases. The list is divided into a number of categories which roughly correspond to those in MIL-STD-89014 with some categories added and modified.

When deciding which patterns to include, we considered only the appearance of the patterns. We intend to provide enough unique patterns to allow reasonable flexibility while excluding rarely-used patterns from the database. For example, for "Swamp, Mangrove", the ITD allows for variations GR1 (general roughness category), UGD (undergrowth density category), and DMT (density measure). If all possible combinations were used, thousands of different patterns would be produced just for mangrove swamps. After some study, we concluded that GR1 had the least effect on the appearance of the mangrove swamp and that nine texture patterns using different combinations of UGD and DMT would provide adequate flexibility in the choices available for mangrove swamp texture patterns without having too many. Similar considerations were made for the other proposed texture patterns. While we did make specific choices to limit our efforts for phase II, there is no practical limit to the expansion possible with the approach being considered. New patterns can be added when necessary.

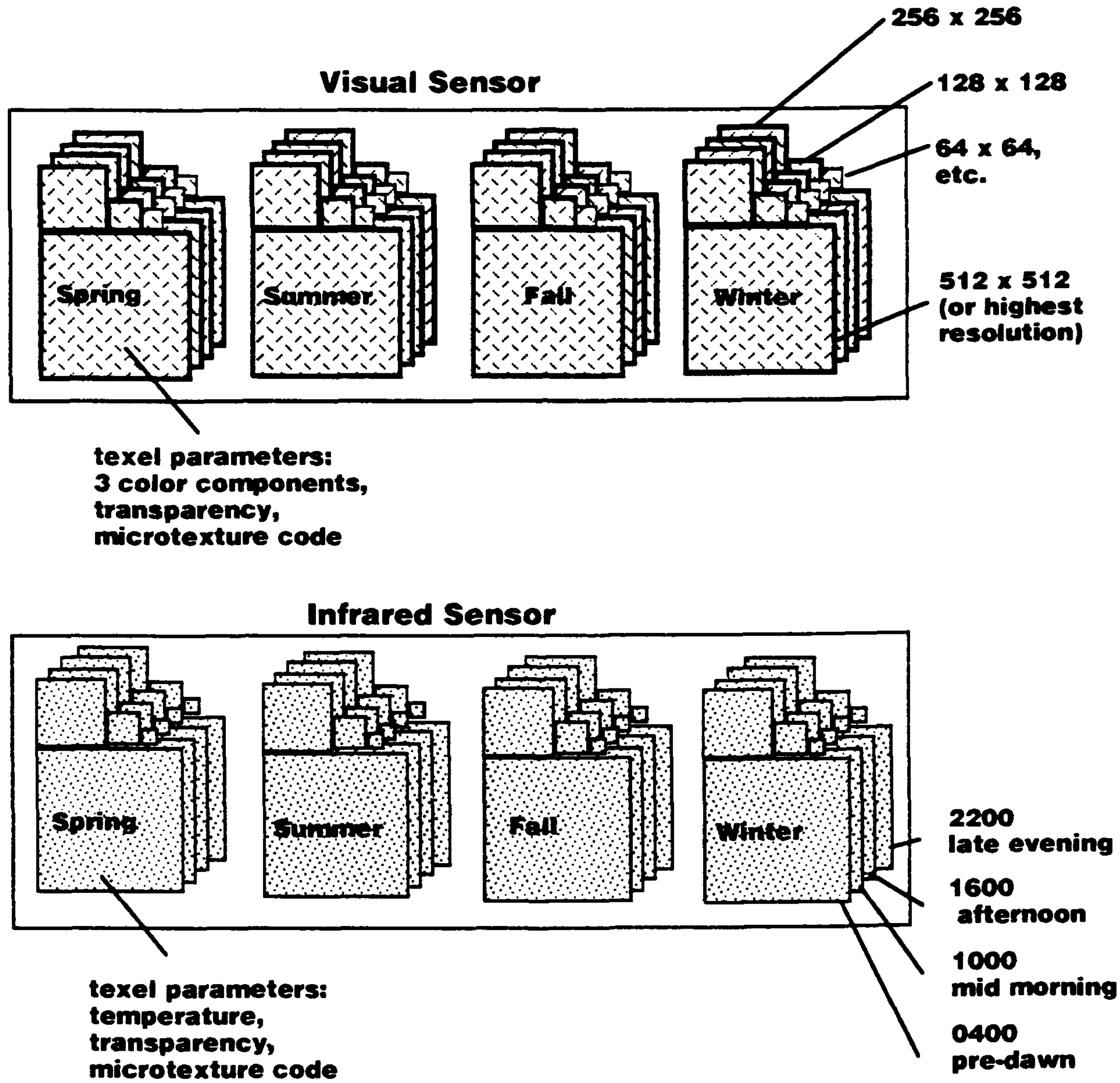


Figure 4.7-1 The data included in a set of texture pattern variants for a particular case. Most patterns would not require so many variants, but the library is structured to accommodate them.

The list is tabulated according to: **Pattern Index, Variety, Description,** and **Attribute(s). Pattern Index** is a reference number to separate one group of patterns from another. It is provided to allow easy reference. **Variety** is the number of patterns associated with the indexed pattern. **Description** is the name of the indexed pattern sometimes augmented with descriptive information. The **Attribute(s)** is the variable(s) that has the most affect on the appearance of the texture pattern. For example, we show a list of vegetation texture patterns. On the first entry we suggest producing 12 different "Built-up area" texture patterns by incorporating various combinations of building density and building types. We plan to make the exact choice for each pattern during the Phase II effort.

The following table summarizes the categories chosen as well as the number of visual texture patterns we propose developing for each category in Phase II of the program:

Table 4.7-1 Visual Texture Pattern Summary

Texture Category	**Total Texture Pattern Quantity**
Vegetation	163
Natural Surface Materials	75
Surface Drainage	17
Transportation	54
Obstacles	8
Trees/Shrubs (profile)	8
Construction Materials	63
Environmental	18
Camouflage	256
Microtexture	28
Total	**690**

Vehicles use a variety of camouflage patterns. Most countries have multiple camouflage patterns for a given vehicle type - particularly ground vehicles. Not only can different standard camouflage patterns be applied for different seasons of the year and environment but individual military units sometimes apply non-standard patterns. While some standardization has occurred for NATO countries, a wide variety still exists between countries. To scope the task of providing camouflage patterns we have therefore proposed providing up to 64 patterns for each camouflage subcategory as shown in Appendix A.

As the simulation observer moves toward a textured object, the resolution of some textures is insufficient to provide the proper visual cues. As a result, most image generators have the ability to provide microtextured surfaces when the viewing range to selected polygons is very short. For example, microtexture could be employed to simulate detailed vegetation, or fine detail in gravel or sand. A list in Appendix A summarizes the microtexture patterns we have identified.

We reviewed the list "Visual Texture Patterns and Categories" shown in Appendix A in order to produce a list of IR texture patterns. We decided that a single IR pattern for each visual texture pattern type would be appropriate for implementation on Phase II of this contract. The IR texture patterns we recommend including are listed in Appendix B. The list recommends 276 patterns.

Choice of the exact IR texture patterns to be produced for the texture library will be the subject of discussions during Phase II of the contract. We recommend providing texture patterns for common military vehicles, perhaps those currently modeled for SIMNET and the Army's Close Combat Tactical Trainer, now under development.

5. Source Material Acquisition

Texture patterns can be generated using interactive or synthetic generation methods. These methods will be discussed in detail in Section 6 of this report. Basically, interactive methods used to make texture patterns require that we obtain the digitized photographic data of real life objects or produce the digitized photographic data from hard copy photographs. If color photographs are available we will use them; use of monochrome photographs requires that we add false color to the digitized image. Although we can correct for some perspective distortions, the preferred photographs are those that present the texture pattern in the same plane as the photograph to avoid the need for perspective corrections. Many of the texture patterns we plan to produce represent patterns seen on the ground as viewed from the air. Source material for many interactively produced texture patterns therefore consists mainly of aerial photographs or digitized aerial photographs. Other non-aerial photographs of certain objects such as tree profiles and some construction materials will also be needed. There are only a few of these and they can be obtained from books or by taking the photographs ourselves.

Synthetic generation requires knowledge of what a particular texture pattern should look like. In most cases, therefore, synthetic texture pattern generation also requires that we obtain a photograph of the desired pattern so that we can use it as a reference. It is not as important that the image be in the correct perspective since it is only used as a reference. The need for both interactive and synthetic texture generation methods is, therefore, to find sources of aerial photographs, preferably color, that would provide a large number of possible texture patterns. The United States Geological Survey (USGS) in Menlo Park, CA is one such source.

We visited the USGS on several occasions during the study. In conversations with USGS personnel we learned, the USGS offers black and white photographs, true color photographs, IR photographs of various areas of the US as well as digitized photographs of some areas. We learned that various groups of photographs in each of these categories were obtained by USGS at different times since photographic missions were funded by different projects for various agencies. The more recent photographs are typically higher quality than the older ones.

By far, the most plentiful photographs in the Menlo Park office were monochrome photographs of California. They ranged in scale from 1:12,500 to 1:80,000. Copies of most black and white photographs were available for review. Color photographs (1:12,500 scale) were also available of limited areas but were not available for review due to their cost. We ordered two color photographs to use experimentally. One covers a residential area and the other a rural area, both on the San Francisco peninsula. Infrared photographs (1:12,500 to 1:58,000 scale) of limited areas are also available from USGS.

USGS personnel provided us with a list with over 300 additional aerial photography sources covering the western U.S., Hawaii, and Guam. We called several of these agencies and learned some of them have only monochrome and some have both color and monochrome. The scale of some of these commercially available photographs (1:2,000) is more appropriate for some of the texture patterns we plan to produce. The combination of the USGS and commercial suppliers will provide us with ample aerial photographic source material for most patterns. Aerial photographs for remaining texture patterns, if any, will be obtained from similar photograph services elsewhere in the U.S. or around the world.

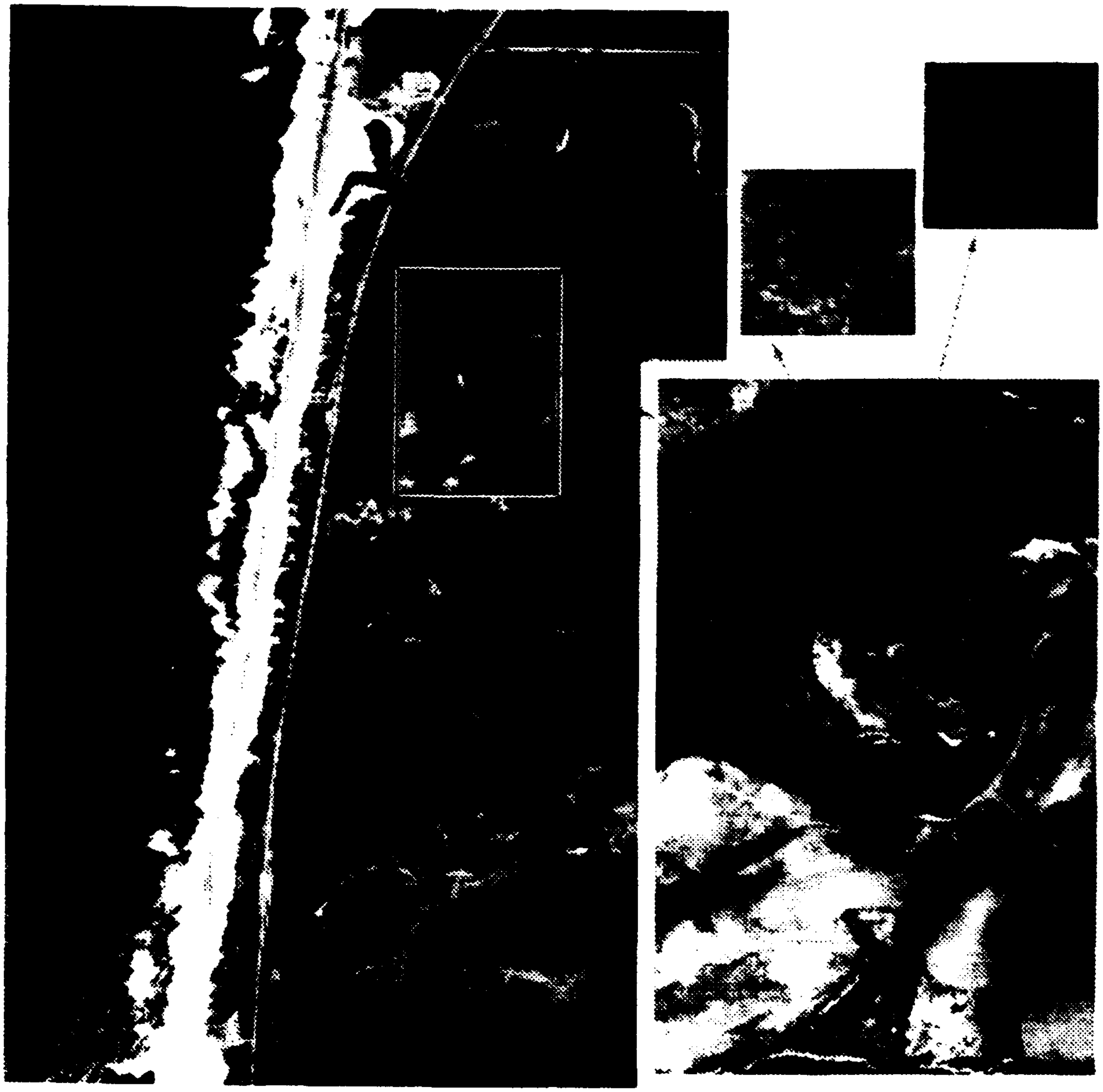


Figure 5-1 Texture patterns will most often be obtained from aerial photographs indexed by the U.S. Geological Survey. This example shows textures we obtained from successive magnifications of a photograph, starting with an approximately 4" x 8.5" section of a 9" x 9" color original of an area near Half Moon Bay, California. More detail can be obtained from photographs at lower altitudes and by using higher resolution scanning.

The task is to identify and select the correct photographs for the texture patterns we need. We will write software which will, based on the feature code and attributes for the desired texture pattern, search the Interim Terrain Database (ITD) for occurrences of that feature. We expect most searches will result in multiple matches. When a match occurs, the geographic location of the feature will be determined. Using the geographic location obtained from the ITD, we will purchase an aerial photograph or digitized photographic data of that location from one of the agencies mentioned above.

The visual texture patterns identified in Appendix A are a comprehensive subset of the massive variety of patterns possible in the ITD. This approach for collecting source material will insure that the resulting texture patterns correlate directly with the ITD. The source material acquisition process is universal; it will apply to all terrain texture patterns. With only one process involved, the software developed to support source material acquisition will be minimized; it will not be necessary to have several methods each with its own procedure and supporting software. It will also mean that training time and time spent supervising CGSD personnel will be reduced since only one process will need to be learned and used.

6. Texture Generation Methods

We have identified six texture generation methods, two interactive, and four synthetic. Each of the six could be used as a part of the library-building process.

6.1 Texture Generation with Photographic Source Data

To generate texture patterns from photographic source data, the image must first be digitized by use of an image scanner. Once an appropriate image is stored in the computer, it can be manipulated to produce the desired texture pattern. Corrections can be made for such things as perspective, color, contrast, and brightness. A second method involves obtaining the photographic image data in digital form. The USGS and other government agencies offer photographic imagery in digital form. However the image data is obtained, it must ultimately exist in digital form for the texture library. Subsequent processing methods are the same for both images obtained in digital form and those we convert to digital form.

Some texture patterns, like trees, are single objects surrounded by transparency. Many patterns, however, are designed to be repeated to cover large areas, the way identically-patterned floor tiles are repeated to cover a floor. For repeated patterns, matching the pattern across the four edges is a major concern in the texture generation process.

One way is to touch up the pattern interactively so the four edges match. For example, a pattern might be scaled to cover a square region of, say, an urban area, so that a road bounds each edge. If the pattern is carried exactly to the middle of the road on each edge, the tiled patterns would mate forming full width roads. This method may work for fields bounded by hedgerows and for other types of agriculture patterns as well. Generic patterns may also be made by overlapping the seams with feathered edges, then recutting the pattern to the tile size.

An old trick that works, at least to some extent, with many patterns is to mirror the pattern left-to-right and then mirror the pair again top-to-bottom (Fig. 6.1-1). This makes a group-of-four-in-one tile that always matches. The limitation is that lines crossing the edges diagonally become chevrons in the four-in-one version.

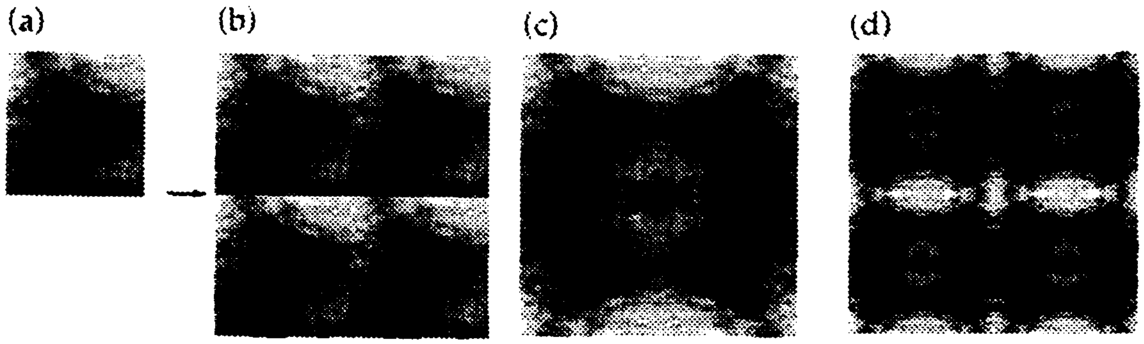


Figure 6.1-1 Mirroring a pattern makes a version which can be tiled without discontinuity, but the method can introduce unintended large-scale features

The method is therefore most useful with patterns having lines that are either horizontal or vertical with respect to the pattern edges. For example, any pattern of crops in which the crop furrows are aligned with the pattern edges is a candidate for the method. Large scale features also form circular or diamond patterns that are sometimes unwanted.

6.2 Synthesized Texture Generation with Digital Source Data

Patterns can also be synthesized so that the edges automatically match. There are four approaches to the synthesis process.

One way is to use a synthesis program that places the statistical properties of the pattern under user control. The user adjusts the properties and colors of the pattern iteractively until the result is subjectively acceptable for the intended purpose. The user might look at a photograph as a guide in judging the acceptability. We wrote a simple program to explore this concept, based on a "turbulent texture" algorithm described in the literature (Perlin, Ref. 45 in Bibliography). Our experience (Fig. 6.2-1) revealed that this approach is more promising than it might seem; it is possible to get realistic looking textures. It seems particularly well-suited to patterns with high spatial frequencies, like rough water.

A second method is to draw profiles of the color component patterns interactively in the horizontal and vertical directions of the texture tile. The drawing program automatically mirrors the starting profile at the edges so the user can accurately achieve continuity for the tile to be replicated. We wrote a program to experiment with this method, and found it works well with patterns having lower frequency components. Patterns from drifting sand are an example.

A third method is to analyze a photograph to find the spatial frequency distribution of a base pattern, and then reconstruct a similar pattern using the derived spatial frequencies. The idea is to filter out the high frequencies introduced by the tile edges, then recreate a tile. Starting with the four-in-one version of the pattern sometimes is helpful because it provides at least first-order continuity over the edges to start with.

We also studied the production of texture patterns by using fractals. The technique involves controlling a few variables which are acted upon by the fractal-based software. Experimentation revealed that small changes in the input variables produced large unpredictable changes in the resultant pattern. While the technique produces patterns quickly, the unpredictable results make the technique less useful than expected. Nevertheless, we may find some applications where the technique is useful.

The interactive processes have the advantage of potentially being the most realistic since they are based most directly upon photographs. However, photographs are not always available for every version of a pattern desired. For example, an aerial photograph may show a pattern with long shadows corresponding to a particular time of morning or afternoon. A user might generate a shadowless "high noon" version by manipulating a synthesizing program with reference to the photograph. Consequently, we expect alternative methods to each be useful.

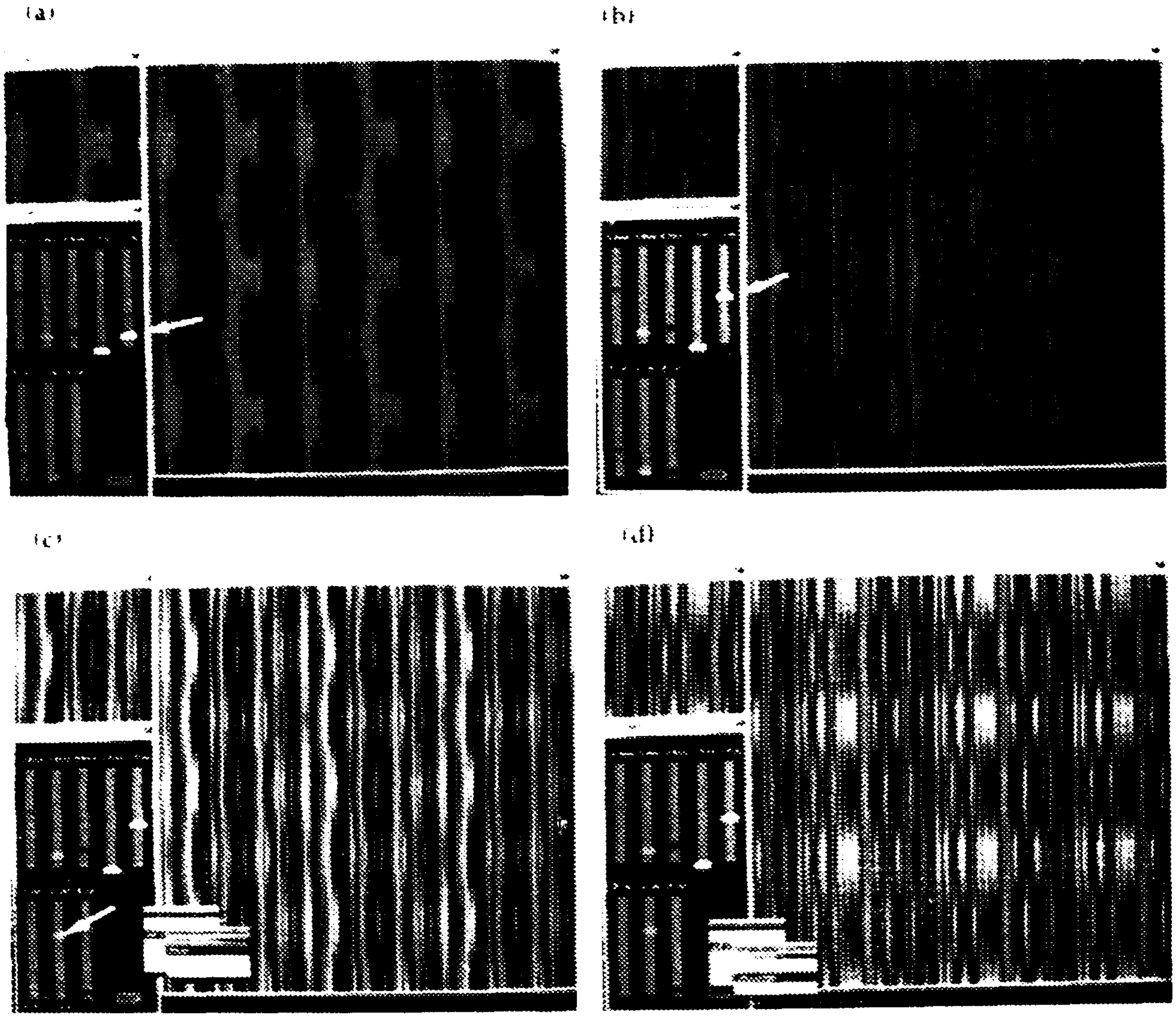


Figure 6.2-1. Texture synthesis program. Screen photographs of the Silicon Graphics workstation program show shaping of pattern using control panel and menus. Both horizontal and vertical frequencies are [illegible] (a); horizontal frequency is increased in (b). Color [illegible] variation is boosted in (c), increasing contrast. Interpolation method is changed from linear to sinusoidal in (d).

6.3 Color Matching

Aerial photography provides an excellent source for geometric texture pattern data. Aerial photography also presents a fairly accurate representation of the color of the texture pattern on the ground but only if it is viewed from the aircraft taking the photo. The color would not match the color of the real-world object if viewed from a short distance. If this distorted color was used with an image generator, the image generator would introduce additional haze effects proportional to the observation range from the texture pattern. This "double attenuation" would produce an erroneous impression. Something must therefore be done to correct the texture pattern color for the library.

In order to assure that the colors used in the texture library are accurate and consistent we plan to measure real-world object colors using the Munsell Book of Colors and produce a short catalog of standard colors. During texture pattern production texture pattern colors would be adjusted to match the closest item in the catalog of standard colors. The colors obtained from the Munsell Book of Colors will be converted to CIE coordinates and stored with the texture pattern. This process will insure that texture pattern colors are both accurate and consistent.

6.4 Visual Version Creation

Creation of visual versions of the texture patterns involves both modification of existing images and the production of new images from new source material. The method used will depend on both the pattern needed and whether existing patterns are suitable for modification. Camouflage patterns, for instance, often vary only in color, not in shape. It is therefore likely that at least some new camouflage texture patterns could be created by modifying the colors of existing patterns. It will certainly be necessary to acquire new source material for many other patterns. The choice will be made on a case by case basis.

6.4.1 Visual Time-of-Day

Texture patterns for times-of-day (TOD) other than "high noon" need to have shadows. This means new source data having texture patterns with shadows must be used. In many cases, it may be possible to use the same source data for morning and afternoon by controlling the orientation of the texture pattern. In other cases new source material will be needed.

6.4.2 Visual Time-of-Year

In many cases colors can be re-mapped to provide adequate visual time-of-year (TOY) versions. This might be true of selected crops, terrain, and evergreen trees. Deciduous trees and most crops would, however, require different texture patterns produced from different source material.

6.5 Infrared Version Creation

High fidelity infrared (IR) texture patterns for terrain will, if eventually implemented, require knowledge of material the object is made of but also the effects for time-of-day, time-of-year, weather, and for vehicles, current vehicle activity. Since IR images reflect the current heat of objects in the image, the recent "heat history" of objects is a concern. For fixed objects, the heat history is primarily affected by time-of-day, time-of-year, and weather. Weather effects contrast in IR imagery - imagery for a clear day will differ from one for a rainy day. High fidelity IR texture patterns for vehicles will require knowledge of recent vehicle activity, i.e., has the engine been running or not, and the location in the vehicle of heat sources such as the engine, gun barrel, and treads.

We have had preliminary discussions with Technology Service Corporation and McDonnell Douglas Corporation relative to the generation of IR texture patterns. Discussions with personnel involved with the TABILS program at Eglin AFB has revealed that most of the IR imagery they have is classified. While this does not rule out TABILS as a source of IR image data for approved users, it does make access impossible for uncleared users.

We have decided to include only basic, unclassified, IR versions in the initial library. These will be predicted by modifying visual texture patterns. The subject of high fidelity IR texture pattern generation requires substantial (out-of-scope) study to treat it comprehensively. Only a very few users need high fidelity IR texture patterns.

6.6 Summary

Through our study we learned that multiple techniques for generating texture patterns will be required. The methods we plan to use for texture pattern generation will be both comprehensive and flexible. Each technique has its own advantages and is best suited for certain types of texture pattern generation. For example, high spatial frequency texture patterns for such things as concrete, water, gravel, and grass might best be generated synthetically. Other texture patterns for irregular features such as a residential neighborhood, industrial complex, or a railroad will be generated using photographically based imagery. In addition to generating TOD, TOY, and IR versions from new source material, some versions will be generated by modification of the original version of the texture pattern. The flexibility offered by this suite of techniques will allow us to produce the best possible texture patterns in the least amount of time.

7. Formats

7.1 Input/Output

CGSD conducted a user survey relating to the format of the library data rather than its content. We sent our survey to eleven vendors of image generators and received responses from six. The point of the survey is to make sure that there is nothing major overlooked in the selection of distribution media, format, or computer host. We learned several things from the survey: The most preferred format for texture patterns is an integer array with all others (SIF, MultiGen Flight, SGI-RGB, and TIFF) as close seconds. All respondents currently support 8 bits/color component. The preferred platform for the texture library is SGI/UNIX with IBM PC and Sun/UNIX as a second choice. All respondents prefer a power of two up to 512 x 512 for pattern size.

Based on the survey results, we are planning to allow patterns to be downloaded in any of at least four formats: integer array, SGI-RGB, TIFF and SIF (SIF is JPEG compressed). Download software provided with the distribution will format the patterns to suit the users preference during the process of downloading. This variety of output formats directly addresses the stated needs of the potential texture pattern library user community.

7.2 Storage

There are two modes for accessing the library. One mode is for browsing and interactive searching by a user on line. The second mode is for direct computer access by a user program building a database from multiple sources.

We have chosen to use an IBM/clone PC as the platform for browsing and interactive searching of the texture library. Our user survey indicates this choice will meet the needs of the potential texture pattern library user community. Beyond that, the IBM/clone PC is a very cost effective solution and it permits us to use a wide variety of third party software. We have chosen Kodak Shoebox Image Manager as the texture pattern storage and retrieval software. It has the ability to store images in PICT, TIFF and EPS file types. Only the miniature "thumbnail" images will be stored in the Kodak Shoebox software. The thumbnail images which are used for browsing with Kodak Shoebox are independent of the large, full resolution texture pattern files.

The large, full resolution texture pattern files are compressed with JPEG software on the distribution media. Using only modest compression, we expect to store the entire library of nearly 1000 patterns on a quarter of a single recordable CD.

The CD format is an industry standard that can be read by virtually any computer that supports CD-ROM. While it is not practical to put the browsing software on every computer (Kodak supports PC's and Mac's), we propose providing a C-language source library that user's can compile and run as part of their own database building program.

8. Image Header Information

8.1 Statistics

To provide additional search criteria for the texture library, image statistics such as the dominant, secondary, and average color of the texture pattern will be added to the header information for the pattern. Pattern horizontal and vertical geometric repetition rates will be included for the same reason.

8.2 Text Description

The text description is a caption for the pattern. This caption will appear with the thumbnail images presented with the Kodak Shoebox image manager software. Typically it would be the MIL-STD-89014 (ITD) description if appropriate. Examples include "bamboo/wild cane", "railroad, single track", "swamp, mangrove", or "volcanic".

8.3 Descriptors

The descriptor data is used by the library access programs to facilitate finding and retrieving the right texture data. For example, if all the versions of the pattern (which we will call a *pattern set)* are available in the library, for a 512 x 512 pattern we will have ten levels of resolution x four seasons x (1 visual + 4 IR times-of-day) = 200 patterns to keep track of for a single set. Rarely will all the versions be generated, and not all are necessary (we do not need four seasonal versions of a palm tree, for example, and many patterns have no transparency). Also, some texture map parameters may be identical, such as the transparency parameters in each of the IR sensor versions (but the IR transparency is not the same as the visual transparency). The descriptor can equate the parameter maps so we do not have to store identical copies. The descriptor information must include which of the patterns in the pattern set are available, which parameter patterns are available and which are identical, and where the available data are stored on the library disc.

The descriptor data will be stored contiguously on the storage media to speed up the search process. Addresses in the descriptor data will point to the patterns stored separately.

Note that if each parameter requires one byte for storage, a full set of levels of detail for a visual pattern (with parameters R,G,B, and transparency) will take (512 x 512) pixels x 4 bytes/pixel x 1.333 for all levels of detail = 1.4 MB of storage without compression. (Note that all levels-of-detail of a pattern comprise 1 + 1/4 + 1/16 + . . . = 1.333 times the storage of the highest resolution alone.) With a modest compression ratio of 5:1, the storage required per visual pattern is 1.4 / 5 = 280K bytes. IR patterns have one or two parameters (depending on whether there is transparency), and so will not take more than 140K bytes. There are 650 MB on a single CD-ROM. We are planning to have 276 IR patterns plus about 700 visual patterns in the basic library, which will require 276*0.14 + 700*0.28 = 239 MB. This easily fits on one CD-ROM, and would accommodate an average of about two additional varieties per pattern as well. There is no extreme penalty for using more than one disc, but it will be easier to search the library if there are fewer discs.

Here are other descriptor data that accompanies each pattern set:

- **ID number**: a unique integer identifier for each pattern set
- **Name**: unique alphanumeric string naming the pattern
- **Key descriptors**: descriptive words for key word searching
- **Pattern dimensions**: the size of the pattern, specified by the number of texels along each of two edges at the highest resolution, all patterns are squares or rectangles with each edge having a number of texels that is an integer power of two
- **Scale**: the texel spacing in meters, all texels are on a square grid
- **Feature codes**: a set of code numbers that establishes correspondence with DMA, Project 2851 (SIF), or DIS standards
- **Category tags**: to facilitate searching, a set of one-bit tags that categorize the object such as natural/manmade, moving/stationery, terrain/feature, generic/specific
- **Projection:** whether the pattern was derived for ground mapping (like an aerial photograph to be applied to three-dimensional terrain) or for surface mapping (like a brick pattern applied to a wall surface)
- **Location**: if generic, the associated climate zone, if any; if specific, the latitude and longitude bounds of the pattern
- **Characteristics**: automatically computed dominant, secondary, and average colors, degrees of variation
- **Time-of-year**: the month (1-12) the source data was produced
- **Time-of-day:** the time (1-24) the source data was produced
- **Shadow angle:** to aid in searching and to facilitate choosing patterns with similarly directed shadows, the angle of the shadow relative to the vertical axis of the pattern
- **SIF defined header information:** to be included to meet the potential needs of access control and security

We will define extension mechanisms. For example, some patterns might have "destroyed" versions to reflect damage from military operations or other versions not anticipated at the outset. Other patterns may form an animated sequence, perhaps for explosions or for sea state for example. Also, a mechanism will be included to accommodate different texel parameters. This seems straightforward, because we already have the task of keeping track of which parameters and patterns from our basic selection are available. Adding more possibilities to the list should be easy and would add to flexibility to adapt to unforeseen requirements.

In studying texture patterns, we have noticed that many patterns have a direction of "grain" or orientation. For example, the direction of field furrows, the direction of tree rows, and the direction of erosion patterns may need to be aligned with other terrain features. Accordingly, if a pattern has a grain, the orientation will be included in the header information. Also, although patterns without shadows are often preferred, sometimes shadows do appear. The orientation of the shadow relative texture pattern axes should also be included in the header information. All other things equal, it is best to have all shadows aligned away from the simulated sun direction in the final application, and including the data will allow the user to make the alignment automatically.

An additional group of descriptor data called *library maintenance data* also accompanies each texture pattern. This data is used primarily by those who generate or modify the texture patterns. It will facilitate configuration control of the texture pattern library.

- **Pattern creation date:** automatically entered by the computer
- **Photographic source:** identifies the source of the pattern such as a particular photograph in a particular book or the identification number of an aerial photograph
- **Corrections:** quantitatively identifies the corrections made to photo-based images (perspective, contrast, color, etc.)
- **Pattern generation software:** identifies the software used to generate a synthetic pattern
- **Input data set:** stimulus data for synthetic pattern generation software

Since we are unable to predict all the ways a user might wish to search the texture library, a highly flexible search scheme is desirable. The descriptors listed here will allow users to search the texture pattern library with a large variety of parameters used alone or in combination with each other. A user could, for instance, conduct a combined-key search for patterns having a certain feature code, shadow angle, ground roughness, and a particular primary color; or he could make a single-key search for a pattern when his only concern is the primary color.

The library maintenance data permits the user to trace library contents back to its source. Such information will be useful when maintaining the library. Suppose a latent defect such as poor contrast, wrong color, or a scratched photographic source image for a pattern was discovered after the texture library had been produced. It will be possible, using this descriptor information, to recreate the texture pattern from the original source material, whether photo-based or synthetic, while correcting or avoiding the defect.

9. Compression Techniques and Software

9.1 Image Compression Requirements

Our estimates show that without compression, a 1000 pattern texture library would not quite fit on a single CD-ROM, although depending upon the pattern sizes it might barely fit. Discs are inexpensive to reproduce, but the problem is that with any expansion of the library, users would have to change discs to complete a pattern search. With modest compression ratios (of 5:1), the library will have substantial expansion capacity within the bounds of a single disc. Using JPEG compression/ decompression software at a 5:1 compression ratio, the patterns appear virtually identical to the originals.

The Simulator Interchange Format (the Project 2851 standard for flight simulator databases) calls out use of JPEG compression. JPEG is not ideally-suited to the application because the compression and decompression times are by design about equal, although actually from our timing, decompression is about two to three times faster than the compression. We would prefer high compression times and fast decompression because compression is only performed once, when the library is made. However, as a practical matter we discovered that all the decompression times are so fast, typically under 20 seconds, that decompression time is not the dominant concern. A desire to conform to the standards, together with JPEG's high quality at low compression ratios, ultimately lead us to recommend it for library use. The considerations leading up to that selection are detailed in this section.

An image compression and decompression scheme is required for efficient storage and retrieval of texture maps from the image database. There are several important requirements the compression/decompression algorithm must meet for candidacy in the architecture. These requirements are listed in decreasing order of importance:

a) Maintain almost perfect reconstructed image quality. Image artifacts introduced, if any, must be almost imperceptible. This must be true for synthetically generated as well as natural images.

b) The algorithm must successfully operate on 24 bit xyY color image data, 32 bit color data with 8 bit transparency, and 8 bit monochrome data (infrared imagery).

c) Fast image retrieval and reconstruction must be possible. In this context decompression times for a 512 x 512 x 24 bit image of 10 seconds seems acceptable.

d) Image compression performance should be such that a PC software solution is viable and compression times of less than 5 minutes for a 512 x 512 x 24 bit image must, at a minimum, be achievable.

e) Hardware support for compression and decompression is a desirable commercially available option.

Interpretation of these requirements suggests:

- A reversible compression scheme be used or a nonreversible scheme which maintains extremely high image quality at low to medium compression ratios e.g. compression from 3:1 to 6:1. Since reversible compression schemes have difficulty achieving greater than 3:1 compression for generalized color imagery, nonreversible compression is preferred.

- For single frame images the most successful intraframe compression/decompression schemes which operate on color and monochrome data include adaptive differential pulse code modulation (ADPCM), vector quantization, fractal compression, adaptive hybrid coding, and block by block transform coding [3]. Of the most successful block by block transform coding schemes the defacto standard JPEG compression scheme [4] is the most frequently used. Since ADPCM degrades image quality rapidly after compression ratios of 2-3:1, it is not considered viable for this application.
- Requirements (c) and (d) above suggest a non-symmetric scheme where fast decompression is desired. Of the schemes mentioned above vector quantization and fractal compression [1] fit this profile the best with algorithms which are heavily weighted towards computationally costly compression in order to facilitate simple and fast decompression. JPEG is also non-symmetric but decompression still necessitates an inverse discrete cosine transform and is therefore more costly when decompressing than either fractal or vector quantization.

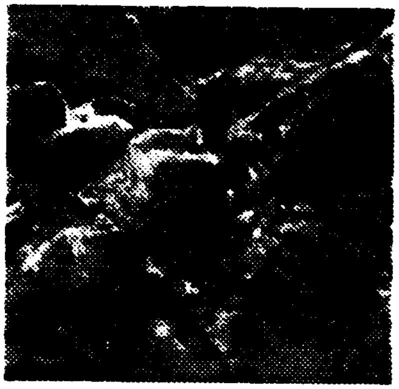
Both Fractal and JPEG algorithms achieve the desired compression and decompression performance requirements stipulated. Vector quantization (VQ) compression is computationally costly for a PC-based solution. More importantly, vector quantization even at low compression ratios can introduce annoying blocking artifacts when a distinctive single area in the image has no close code book entry [6]. For these reasons VQ is not considered further.

The requirement to retrieve image data suggests slightly higher compression ratios e.g. 6:1 if image quality can be maintained and decompression is sufficiently fast for this application. Both JPEG [5] and Fractal [2] approaches have known commercial hardware compression solutions. JPEG decompression hardware is also available [5].

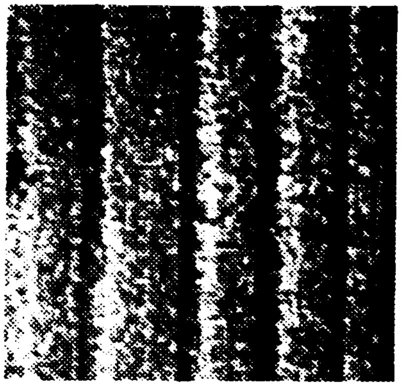
9.2 Evaluation System Description

The image compression and decompression evaluation was undertaken on an IBM clone 80486i/66Mhz system supporting a 24 bit, 640 x 780 color frame buffer. For the algorithm evaluation the POEM Images Incorporated package from Iterated Systems, Inc. was obtained and installed. This package provides among other features Fractal and JPEG compression/decompression software support. The software further supports the input and output of image data in various formats and the ability to adjust the compression ratio for both algorithms under investigation. Using the package's tile image capabilities original and compressed images can be viewed simultaneously side by side.

Two image data sets were selected for the algorithm comparison. The first set contained 12, 24 bit TIFF format natural images from a wide variety of sources. This set included images of residential housing, forestry, aerial images of an inner city, and head and shoulder scenes. The second set contained 4 texture maps (Fig. 9.2-1) consisting of rock formations, water and foliage, typical of imagery which will make up the texture database.

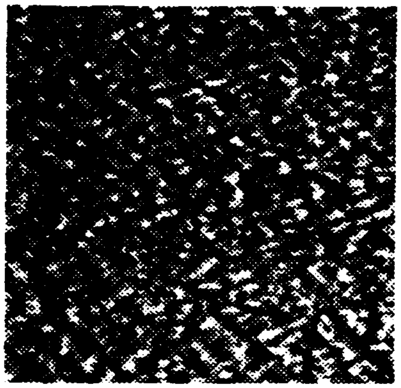


Rock, sandstone microtexture



Orchards

Rock, vertical strata



Water, large river

Figure 9.2-1 Texture patterns used for evaluating compression schemes

The rationale for using the first set of natural scenes was to establish the form of image degradation, if any, from the algorithm and to help distinguish the possibility of misleading artifacts thrown up as a consequence of a particular image texture sample during algorithm evaluation.

9.3 Algorithm Comparison

The JPEG and Fractal image compression/decompression schemes will be compared for their abilities to:

a) Achieve extremely high quality at low compression ratios.

b) Achieve compression and decompression speeds which allow implementation via a low cost PC software solution. A bias is towards an algorithm which has extremely high performance decompression.

c) Has low implementation/acquisition costs.

9.3.1 *Fractal Image Compression*

Image quality from the Fractal compression scheme was extremely good at compression ratios up to 10:1. From 10:1 to 20:1 there was a marked decline in image quality. Below 10:1 image compression distortion was however still visible and took on two distinct forms. The first noticeable distortion was blurring which occurred in areas of high detail. The second form of visible distortion was the loss of small structural detail. As an example of the later distortion, cracks in rock formations which were continuous in the original appeared discontinuous in the compressed and then reconstructed image. Image detail which was small but particularly distinct in terms of contrast, also appeared "blocky". This blocking distortion appeared to introduce artificial detail that was not visible in the original image.

Software compression and decompression times for various image sizes and CPU's for Fractal compression are given below, as are software decompression times.

Table 9.3.1-1 Software-only versus hardware-assisted compression times

Image Size	CPU	Software Only	FTC-III Hardware
		(min:sec)	(min:sec)
320 x 200	80486/50	0:10 - 0:25	0:03 - 0:09
	80486/33	0:15 - 0:42	0:03 - 0:11
	80386/33	0:34 - 1:36	0:04 - 0:14
640 x 400	80486/50	1:07 - 2:04	0:10 - 0:40
	80486/33	1:37 - 3:23	0:13 - 0:47
	80386/33	3:48 - 7:39	0:16 - 1:06
640 x 768	80486/50	2:07 - 4:10	0:19 - 1:13
	80486/33	3:04 - 6:49	0:25 - 1:32
	80386/33	7:50 - 15:28	0:32 - 2:19

Table 9.3.1-2 Full color versus color map software decompression times

Image Size	CPU	24-bit	8-bit color map
		(seconds)	(seconds)
320 x 200	80486/50	<1	<1
	80486/33	<1	1
	80386/33	1	2
640 x 400	80486/50	1	2
	80486/33	2	3
	80386/33	4	7
640 x 768	80486/50	2	6
	80486/33	4	10
	80386/33	7	17

The compression and decompression times noted in the table are well within acceptable limits for this application, even when the compression is done entirely in software.

The fractal compression and decompression software from Iterated Systems used in this evaluation is available for product use and a software developers kit can be purchased from the company. There is also hardware compression support if the software compression times for extremely large texture map databases was inconvenient.

9.3.2 *JPEG Compression*

JPEG, which stands for the Joint Photographic Equipment Group, has become an industry standard compression scheme. The approach is based on two algorithms cascaded together to provide variable compression via user selection and gracefully decreasing quality with an increasing compression factor. The first step performs an 8 pixel by 8 pixel block discrete cosine transform to convert correlated image pixels into transform coefficients which are uncorrelated and which compact the image information into typically a few low frequency harmonics. These harmonics are then quantized into a sample stream which can be transmitted or stored using considerably fewer bits than the original pixel based image. This first step achieves high compression but is of course non-reversible and at the expense of image distortion to a degree dependent on the compression ratio and image type. JPEG then uses reversible arithmetic coding or Huffman coding to achieve a further but usually much smaller additional compression factor.

At low compression ratios this algorithm introduces very little if any distortion but this is image dependent. For the texture images considered for use in this project little distortion is envisaged. At higher compression ratios this scheme introduces visible blocking distortion at 8 by 8 pixel block boundaries and overall blurring due to the truncation within the algorithm of high frequency harmonics.

Over a wide range of compression ratios from 3:1 to 25:1 the JPEG scheme was evaluated on both image test sets. The algorithm performed exceptionally well at compression ratios below 10:1 as expected and little, if any, noticeable distortion was observed. At various compression ratios above 10:1 and dependent on the degree of high frequency detail, blocking and blurring distortion was, as anticipated, visible. In nearly all cases both for textured images as well as images from the first test image set, the quality from the JPEG algorithm was superior to that of the fractal compression scheme.

Using the evaluation system described, the following compression and decompression timings were recorded for a 512 x 512 x 24 bit color image at 10:1 approx. compression ratio:

Table 9.3.2-1 Compression/decompression times for JPEG and Fractal methods

	JPEG	Fractal
Compression	25 secs.	98 secs.
Decompression	14 secs.	3 secs (approx.).

JPEG compression and decompression times fall within the requirements for performance. The decompression timings, however, favor the fractal algorithm.

9.4 Recommendation

Both fractal and JPEG compression met the majority of essential requirements. Both schemes met the desired compression and decompression speed requirements and both had hardware support if it were desired. Quality assessments at various low to medium (3:1 to 10:1) image compression ratios showed that JPEG compression introduced the least visibly distracting artifacts for the image sets considered. At very high compression ratios (25:1 to 50:1) Fractal compression appeared to produce better quality. The blocking distortion at 8x8 pixel boundaries introduced by JPEG rendered the texture maps unusable, while the blurring distortion obvious in the Fractal scheme was overall less displeasing. Since this application calls for higher quality at lower compression ratios, JPEG compression is the preferred solution.

Implementation costs also favor JPEG compression since a JPEG compression and decompression software suite is available in the public domain at no cost. This package has been evaluated and would, after performance tuning, be easily integrated into the proposed architecture. The royalties associated with Fractal compression, although not high for large volumes, is a detracting factor. The recommendation is therefore for the use of JPEG compression in this application.

References

[1] M. F. Barnsley and L. P. Hurd, "Fractal Image Compression", AK Peters, Ltd., Wellesley, Massachusetts, 1993.

[2] POEM Images Incorporated, Iterated Systems, Inc. 5550-A Peachtree Pkwy. Suite 650, Norcross, GA 30092.

[3] A. K. Jain, "Fundamentals of Digital Image Processing", Prentice Hall Inc., Englewood Cliffs, NJ 07632, 1989.

[4] Penne Baker, "JPEG Still Image Data Compression Standard", Van Nostrand, 1992.

[5] JPEG Hardware, Storm Technology, 1861 Landings Dr., Mountain View, CA 94043.

[6] Allen Gersho and Robert M. Gray, *Vector Quantization and Signal Compression*, Kluwer Academic Publishers, Boston, 1992.

10. Classification and Retrieval

This section of the report addresses two user needs: the need to be able to interactively browse through the texture library and retrieve texture patterns, and the need to access the patterns automatically from the users database development software.

10.1 Categorization

The principle behind our access system is to characterize the data by descriptors in many different ways and to provide a means of selecting patterns by combinations of descriptor selection criteria. Once a pattern is identified, it can always be accessed again by reference to its unique ID code.

There are four broad categories of descriptors used for selection: keywords, feature codes, classification tags, and appearance properties.

The keyword category contains alphanumeric information. It includes the name of the pattern, such as "pine tree" and gives associated keywords such as "forest, temperate zone, coniferous, foliage, vegetation." The name and keywords allow a pattern or group of patterns to be selected the same way a literature search is conducted to find a publication.

Feature codes relate the texture pattern to categories established by the Defense Mapping Agency, Project 2851 (Simulation Interchange Format), and the Distributed Interactive Simulation Standard (which has codes for vehicles and weapons). The objective is to permit software working with databases derived from one or more of these standards to automatically select and assign texture patterns from the libraries.

Classification tags are designed to facilitate a hierarchical search, something like searching a library of books classified according to the Dewey Decimal System. For example, one could call up the class of objects that satisfy [natural.stationery.feature.generic] that which would be mainly trees and shrubs. Looking in that class one might find a pattern for an alder tree that is appropriate for a particular application, even though one would not think of it by name.

Appearance properties characterize patterns by automatically computed statistics. For example, one might look for all the patterns that had prominent horizontal streaks. Sometimes patterns can be usefully applied in situations that were never originally contemplated. A "streaky" pattern originally intended for striated rocks might, for example, be applied with a color change to suit a type of roof.

Searches may be conducted by concurrently selecting combinations of characteristics from more than one of the broad categories. For example, the keyword "building" might be used with the appearance property for "predominantly gray" to find a class of gray buildings. For human conducted searches, the access routine could be set to show "postage stamp" versions of the patterns for further selection and investigation, or the search could produce the pattern names. Computer searches would return a list of the ID codes of patterns meeting the criteria. If there were no match the computer would have to try again with a broader criteria, and if there were more than one it could either narrow the selection criteria or make a random choice from the list.

We considered a number of commercial products for storage and retrieval of texture images as shown below.

Table 10.1-1 Image Management Software Products We Evaluated

Company	Product
Aldus	Fetch
Cricket Associates	CA dbFast
Eastman Kodak	Shoebox Image Manager
Imspace Systems Corp.	Kudo Image Browser
Microsoft	Access
Microsoft	FoxPro
MultiAd Services	MultiAd Search
North Point Software	CompassPoint

We have reviewed product literature for the listed products and have discussed features with company representatives. We found that most products have limited search capabilities. Since Kodak Shoebox appears not to have this limitation, we purchased the software for further evaluation.

The Kodak product is well-designed and flexible. Important for our application, it has one hundred user programmable search fields. We implemented a small (about 20 image) database and experimented with the search capabilities. They seem completely adequate. One minor disadvantage is that the user interface layout is not customizable to the particular application, so that, for example, we cannot design a custom window to expedite texture pattern searches. The standard set-up they have is every bit as functional as would be a custom one, but not quite as easy to use.

Another minor limitation of the Kodak software is that only PICT and TIFF files can be retrieved directly for "full screen" viewing. Thumbnail views are stored as part of the index files, so in fact the original data files are not required at all for searching and thumbnail previewing. Texture pattern files will not be larger than 512 x 512, so having 128 x 128 thumbnails seems fine for previewing. However, when it comes time to download the original files, names of the identified files would have to be passed to another program to perform any custom decompression. The download format (e.g. TIFF, JPEG, or SIF) and color space would be specified as well. Most users will only want small numbers of files, so this seems acceptable. Users wanting high speed access to large numbers of patterns will, we believe, not be browsing, but will use direct software access.

General purpose database programs like Microsoft Access have provisions for making custom menus, and for keying external processes like decompression. The general purpose programs have very limited image display capabilities, however. Overall, the Kodak product seems like the best compromise in a commercial product for our application.

10.2 Search and Retrieval Software

Search and retrieval software will be written which will allow the user to search and directly access the full resolution texture patterns according to DMA material code or any other database descriptor. In the interactive mode, the user will be able to interactively select and retrieve the selected full resolution texture pattern by entering the filename obtained from the Kodak Shoebox image manager. Shoebox software is a relational database used to facilitate search. If the user wanted to integrate this search and retrieval software with his own database software, so that search and retrieval was automatic, the routines for this software will be included on the distribution media. Integration of these routines with user database generation software would greatly reduce the time needed to produce a complete textured database. It would seem to be essential for any polygonal database system used by the Special Operation Forces (SOF) due to their need to produce databases in 48 hours.

11. Distribution Media and Production

11.1 Media

Our user survey confirmed our belief that CD-ROM is an acceptable media for texture library distribution. CD-ROM has gained wide acceptance in the computer industry as a reliable media capable of storing large amounts of data. Currently available CD-ROM drives are relatively inexpensive and have acceptable access and data transfer rates. Recently lowered media costs favors the purchase of a drive capable of producing individual read-only CD-ROMs, one at a time. The production drive would be attached to one of CGSD's personal computers. The CD-ROMs so produced could be read by any standard CD-ROM drive or player.

We plan to maintain the master texture library on an optical disk connected to one of the CGSD computers. As the texture library grows, the contents of the master disk will be updated. At any given time the master disk will contain the most up-to-date texture library. When a request for a copy of the library is received, library data is read from the master optical and written to a read-only CD-ROM. The advantage of the approach is that only the required quantity of CDs need to be produced at a given time and no more. This avoids having CDs produced in quantities which risk becoming obsolete before they are sold.

11.2 Production Costs

Mass production CD-ROMs are made by injecting substrate material into a mold. The molding process results in a long narrow spiral track consisting of reflective flat areas (called lands) and non-reflective indents (called pits). The advantage of this production technique is that cost per CD is low. The disadvantage is that a large quantity is required for each production, and the total cost of production is thus high.

The write-once technology we plan to use starts with a totally reflective disk media. The variety of CD-ROM that allows the user to record data on it is called a recordable compact disk, sometimes referred to as a CD-R. These disks permit the user to write data only once by use of a recordable CD-ROM drive. A laser is used to distort the surface of the disk making it non-reflective where a pit is desired. The end result is a long narrow spiral track consisting of lands and pits similar to that described above. Both types of CDs can be played on the same equipment. The initial device cost of a recordable CD-ROM drive is $3,995 (Pinnacle Micro). Recordable CD-ROM drives are sold with the necessary formatting software. Media cost for suitable CDs is $35 each in small quantities.

The CD-R approach is well suited to the texture library application. It allows copies to be made on short notice using the latest texture pattern data. Additionally, it allows disks to be produced one at a time so that there is no risk of producing disks that become obsolete before they are sold.

Bibliography

The following references were encountered during the course of our investigations. Collectively, they comprise a complete introduction to the subject of texture patterns in computer generated imagery.

1. Adelson, Edward H., and Peter J. Burt, "Image Data Compression With The Laplacian Pyramid", Psychology Dept., New York University and Electrical Computer, and Systems Engineering Dept., Rensselear Polytechnic Institute, Troy, NY, CH1595-8/81 IEEE.
2. Beck, J., "Textural Segmentation, Second Order Statistics and Textural Elements", Biol. Cybern., Vol. 48, 1983, pp 125-130.
3. Blinn, James F., and Martin E. Newell, "Texture and Reflection in Computer Generated Images", University of Utah, ACM. Vol. 19, No. 10, Oct. 76, Association for Computing Machinery.
4. Bunker, Marvin, Richard Economy, James Harvey, "Cell Texture - Its Impact on Computer Image Generation", General Electric Co., Eagle Technology.
5. Burge, R. E., and J. K. Wu, "An Adaptive Transform Image Data Compression Scheme Incorporating Pattern Recognition Procedures", Department of Physics, University of London, CH1595-8/81, IEEE.
6. Burt, P. J., and E. H. Adelson, "The Laplacian Pyramid as a Compact Image Code", IEEE Trans. on Communications, Vol. COM-31 No. 4, 1983, pp 523-540.
7. Cabral, Brian, Nelson Max, and Rebecca Springmeyer, "Bidirectional Reflection Functions from Surface Bump Maps", UC Davis/Lawrence Livermore National Laboratory, Computer Graphics, Volume 21, No. 4, July 1987, ACM.
8. Cahill, P. T., and R. J. R. Knowles, " Texture Analysis of Uniform Fields: A Prerequisite to Tissue Signature Identification Analysis of Nuclear Medicine Images", The New York Hospital-Cornell Medical Center, N.Y., N.Y., CH 1595-8/81 IEEE.
9. Cheng, Pi-Yun, and Nicholas Szabo, "Building Digital Image Generator Objects With Multiple Textured Planes", Schlumberger/Benson and Link Flight Simulation Division.
10. Conners, Richard W., and Charles A. Harlow, "A Theoretical Comparison of Texture Algorithms", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-2, No. 3, May 1980.
11. Cosman, Michael A., "Mission Rehearsal Modeling", Evans and Sutherland Computer Corporation, Salt Lake City, UT, Image VI Conference Scottsdale, AZ 14-17, 92.
12. Crow, Franklin C., "Summed-Area Tables for Texture Mapping", Computer Sciences Laboratory, Xerox Palo Alto Research Center, Computer Graphics, Vol. 18 No. 3, July 1984, ACM.
13. D'Astous, F., and M. E. Jernigan, "Texture Discrimination Based on Detailed Measurements of the Power Spectrum", Proc. 7th International Conference on Pattern Recognition, Montreal, 1984, pp 83-86.
14. "Data Format Description, Software Systems Flight Data Bases", Software Systems, Format Revision 9.0, Updated November, 1990.
15. Davis, L. S., and M. Clearman, "An Empirical Evaluation of Generalized Co-occurrence Matrices", IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. PAMI-3 No. 2, 1981, pp 214-221.

16. Demko, Stephen, et. al., "Construction of Fractal Objects with Iterated Function Systems", Georgia Institute of Technology, SIGGRAPH Conference Proceedings Vol. 19 No. 3, 1985 ACM.

17. Dunham, Douglas, et. al., "Creating Repeating Hyperbolic Patterns", Department of Mathematical Sciences, University of Minnesota, and Department of Mathematical Sciences, University of Chicago, Computer Graphics, Volume 15, No. 3, July 1981, ACM.

18. Faugeras, O. D., and K. W. Pratt, "Decorrelation Methods of Texture Feature Extraction", IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. PAMI -2 No. 4, 1980, pp 323-332.

19. Findley, David A., "Visual Database for Orbital Flight", McDonnell Aircraft Co., St. Louis, MO.

20. Gagalowicz, Andre, and Song De Ma, "Sequential Synthesis of Natural Textures", Computer Vision, Graphics, and Image Processing 30, 289-315 (1985).

21. Gardner, Geoffrey Y., "Simulation of Natural Scenes Using Textured Quadric Surfaces", Grumman Aerospace Corp., Bethpage, NY., 1984 ACM 0-89791-138-5/84.

22. Gu, Tao, and Julius T. Tou, "A Group-Linking Classifier", Center for Information Research, University of Florida, Gainesville, FL, CH1595-8/81 IEEE.

23. Haralick, R. M., K. Shanmugan and I. Dinstein, "Textural Features for Image Classification", IEEE Trans. Systems, Man and Cybernetics, Vol. SMC-3 No. 6, 1973, pp 610-621.

24. Hooks, John T., Garth J. Martinsen, Venkat Devarajan, "On 3-D Perspective Generation From A Multi-Resolution Photograph Mosaic Data Base", LTV Aerospace and Defense Co., Image V Conference, Phoenix, AZ, 19-22 June 1990.

25. Ikonomopoulos, A., and M. Unser, "A Directional Filtering Approach to Texture Discrimination", Proc. 7th *International* Conference on Pattern Recognition, Montreal, 1984, pp 87-89.

26. "Interim Terrain Data (ITD/Planning Interim Terrain Data (PITD)", METRIC, MIL--89014, Amendment 3, 11 August 1992.

27. Keller, R. M. Crownover and R. Y. Chen, "Characteristics of Natural Scenes Related to the Fractal Dimension", IEEE Trans., Pattern Analysis and Machine Intelligence, Vol. PAMI-9 No. 5, 1987, pp 621-627.

28. Jules, B., "A Theory of Preattentive Texture Discrimination Based on First Order Statistics of Textons", Biol. Cybern., Vol. 41, 1981, pp 131-138.

29. Kajiya, James T., and Timothy L. Kay, "Rendering Fur with Three Dimensional Textures", California Institute of Technology, Pasadena, CA, Computer Graphics Volume 23, No. 3, July 1989, ACM.

30. Kjell, Bradley P., and Charles R. Dyer, "Edge Separation and Orientation Texture Measures", Computer Sciences Department, University of Wisconsin, CH2145-1/85, IEEE.

31. Knutson, H. and G. H. Granlund, "Texture Analysis using Two Dimensional Quadrature Filters", IEEE Computer Society Workshop on Computer Architecture for Pattern Analysis and Image Database Management, CAPA, 1983, pp 206-213.

32. Krueger, Wolfgang, "Intensity Fluctuations and Natural Texturing", Hochschule der Kuenste, Berlin Art + Com Projekt, Computer Graphics, Volume 22, No. 4, July 1988, ACM.

33. Laws, K. I., "Texture Energy Measures", Proc., Image Understanding Workshop, 1979, pp 47-51.

34. Lewis, J. P., "Algorithms for Solid Noise Synthesis", Computer Graphics Laboratory New York Institute of Technology, Computer Graphics, Volume 23, No. 3, July 1989, ACM.

35. Linnet, L. M., "Multi-texture Image Segmentation", Ph.D. Thesis, Heriot-Watt University, Edinburgh, 1991.

36. Mandlebrot, B. B., "Fractals: Form, Chance and Dimension", Freeman and Co. San Francisco, 1977.

37. Marcelja, S., "Mathematical Description of the Responses of Simple Cortical Cells", J. Opt Soc. Am., Vol. 70 No. 11, 1980, pp 1297-1300.

38. Moshell, J. Michael, et. al., "Dynamic Terrain Databases For Networked Visual Simulators", University of Central Florida, Orlando, FL, VI Conference Scottsdale, AZ 14-17 July 92.

39. Nullmeyer, Robert T., Philip D. Bruce, Capt. Michael T. Conquest, "Using The MH-53J Weapon System Trainer/Mission Rehearsal System: Initial Assessments and Lessons Learned", Armstrong Laboratory Aircrew Training Research Div., 7 October 1992.

40. Oppenheim, A. V., and J. S. Lim, "The Importance of Phase in Signals", Proc. IEEE, Vol. 69, 1981, pp 529-541.

41. Paterson, David N., "Fractal Analysis and Reconstruction of Textures for Visual Simulation", Proceedings of the International Training Equipment Conference, May 4-6, 1993.

42. Peachey, Darwyn R., "Solid Texturing of Complex Surfaces", Department of Computational Science, University of Saskatchewan, Saskatoon, Canada, Volume 19 No. 3. 1985 ACM.

43. Peitikainen, M., and A. Rosenfeld, "Edge Based Texture Measures", Proc. 6th International Conference on Pattern Recognition, Munich, 1982, pp 298-300.

44. Pentland, A. P., "Fractal Based Description of Natural Scenes". IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. PAMI-6.

45. Perlin, Ken, "An Image Synthesizer", SIGGRAPH, 1985 Proceedings, Volume 19, No. 3, ACM.

46. Perlin, Ken, and Eric M. Hoffert, "Hypertexture", Computer Graphics, Volume 23, No. 3, July 1989, ACM.

47. "Proposed IEEE Standard Draft Standard Coordinating Committee, "IEEE Standard for Information Technology-Protocols for Distributed Interactive Simulation Applications, Version 2.0, Third Draft", Contract Number N6133k9-91-C-0091 STRICOM, DMSO, May 28, 1993.

48. Qing-Yun, Shi, "A Method for the Design of Binary Tree Classifiers", School of Electrical Engineering, Purdue University, W. Lafayette, IN, CH1595-8/81 IEEE.

49. Rearick, T. C., "A Texture Analysis Algorithm Inspired By A Theory of Preattentive Vision", Lockheed- Georgia Co., CH2145-1/85 IEEE

50. Reeves, William T., "Particle Systems-A Technique for Modeling a Class of Fuzzy Objects", Computer Graphics, Vol. 17 No. 3, July 1983, ACM 0-89791-109-1/83 Vol. 2 No. 2

51. Sims, Karl, "Artificial Evolution for Computer Graphics", Thinking Machines Corporation, Computer Graphics, Volume 25, No. 4, July 1991 ACM.

52. Smith, Alvy Ray, "Planar 2-Pass Texture Mapping and Warping", Pixar, Computer Graphics, Volume 21, No. 4, July 1987, ACM.

53. Standard Coordinating Committee, "IEEE Standard for Information Technology-Protocols for Distributed Interactive Simulation Applications", IEEE Std. 1278-1993.

54. Target and Background Information Library System (TABILS) Users Manual", Eglin AFB, FL 32542-5000., not dated.

55. "Technical Report Summaries", Defense Technical Information Center, AO 169-030.

56. Turk, Greg, "Generating Textures on Arbitrary Surfaces Using Reaction-Diffusion", Computer Graphics, Volume 25, No. 4, July 1991, ACM.

57. van Wijk, Jarke J., "Spot Noise Texture Synthesis for Data Visualization", Netherlands Energy Research Foundation ECN, Computer Graphics, Volume 25, No. 4, July 1991, ACM.

58. Voss, R. F., "Random Fractal Forgeries", in Fundamental Algorithms for Computer Graphics, Springer-Verlag, Berlin, 1985, pp 805-835.

59. Weinberg, Richard, "Parallel Processing Image Synthesis and Anti-Aliasing", Computer Science Department, University of Minnesota, Minneapolis, MN, Computer Graphics, Volume 15, No. 3, ACM 0-8971-045, 1/81.

60. Witkin, Andrew, and Michael Kass, "Reaction-Diffusion Textures", Computer Graphics, Volume 25, No. 4, July 1991, ACM.

61. Zucker, S. W., and D. Terzopoulos, "Finding Structure in Co-occurrence Matrices for Texture Analysis", Computer Graphics and Image Processing, Vol. 12, 1980, pp 286-308.

Appendix A Visual Texture Pattern List

Tables A-1 through A-10 show lists of texture patterns and categories that CGSD plans to implement in Phase II of the SBIR contract. The patterns were selected using MIL-STD-89014 as a guide. In the tables where the variety of texture patterns is more than one, texture patterns of a type will vary based on the variable(s) shown in the Attribute(s) column. Codes shown in the attribute(s) column of the tables are from MIL-STD-89014 and have identical meaning. In many cases, the texture patterns correspond one-to-one with the features (feature codes) shown in MIL-STD-89014. Other texture patterns are suitable for more than one feature. In some cases, multiple texture patterns are used to represent a single feature.

Table A-1 Texture Patterns for Vegetation

Pattern Index	Variety	Description	Attribute(s)
1	12	Built-up area	Building density Building types
2	12	Water	Wind velocity Color
3	3	Wetlands	GR1 - ground roughness category
4	3	Bare ground	GR1 - ground roughness category
5	3	Wet, dry, or terraced crops	GR1 - ground roughness category
6	3	Shifting cultivation	GR1 - ground roughness category
7	7	Agriculture area	Plowed field, Low row crop (beans) High row crop (corn), Hay field Orchard trees, Rice paddy, or Cotton
8	9	Orchard/plantation (deciduous)	DMT - Density measure (% tree cover) UGD - Undergrowth density category
9	9	Orchard/plantation (coniferous/evergreen)	DMT - Density measure (% tree cover) UGD - Undergrowth density category
10	9	Orchard/plantation (mixed)	DMT - Density measure (% tree cover) UGD - Undergrowth density category
11	9	Orchard/plantation (palm)	DMT - Density measure (% tree cover) UGD - Undergrowth density category
12	3	Vineyard/hops	GR1 - ground roughness category
13	3	Grassland, pasture, meadow	GR1 - ground roughness category
14	3	Grassland w/scattered trees	GR1 - ground roughness category
15	6	Brushland/scrub	GR1 - ground roughness category BDC - brushland density category
16	3	Bamboo/wild cane	GR1 - ground roughness category
17	9	Coniferous/evergreen forest	DMT - Density measure (% tree cover) UGD - Undergrowth density category
18	9	Deciduous forest	DMT - Density measure (% tree cover) UGD - Undergrowth density category
19	9	Mixed forest	DMT - Density measure (% tree cover) UGD - Undergrowth density category
20	3	Marsh/bog	GR1 - ground roughness category
21	9	Swamp, deciduous	DMT - Density measure (% tree cover) UGD - Undergrowth density category
22	9	Swamp, coniferous/ evergreen	DMT - Density measure (% tree cover) UGD - Undergrowth density category
23	9	Swamp, mixed	DMT - Density measure (% tree cover) UGD - Undergrowth density category
24	9	Swamp, mangrove	DMT - Density measure (% tree cover) UGD - Undergrowth density category

Table A-2 Texture Patterns for Natural Surface Materials

Pattern Index	Variety	Description	Attribute(s)
1	3	Permanent snow fields	GR1 - general roughness category
2	9	Gravel	GR1 - general roughness category Color
3	9	Sand	GR1 - general roughness category Color
4	9	Silt, clay	GR1 - general roughness category Color
5	6	Peat/organic soils	GR1 - general roughness category Color
6	6	Evaporates	GR1 - general roughness category Color
7	9	Rock outcrop	GR1 - general roughness category Color
8	3	Volcanic	GR1 - general roughness category
9	3	Stone	Color
10	18	Rocky field	Rock density Rock size Rock color

Table A-3 Texture Patterns for Surface Drainage

Pattern Index	Variety	Description	Attribute(s)
1	1	Covered drainage	
2	7	Bank of: Canal/channelized stream/irrigation canal/drainage ditch/ intermittent stream/ephemeral stream/ perennial stream/stream subject to tidal fluctuations	MCC - material composition category
3	1	Stream bank vegetation	
4	1	Braided water	
5	7	Gorge sides	MCC - material composition category

Table A-4 Texture Patterns for Transportation

Pattern Index	Variety	Description	Attribute(s)
1	1	Railroad, single track, narrow gauge	
2	1	Railroad, single track, normal gauge	
3	1	Railroad, single track, broad gauge	
4	1	Railroad, dual track, narrow gauge	
5	1	Railroad, dual track, normal gauge	
6	1	Railroad, dual track, broad gauge	
7	1	Railroad, single track, narrow gauge, transparent background	
8	1	Railroad, single track, normal gauge, transparent background	
9	1	Railroad, single track, broad gauge, transparent background	
10	1	Dismantled railroad	
11	1	Cart track	
12	6	All weather hard surface highway, single lane	WDD - Width (Decimeters) MCC - Material composition category
13	6	All weather hard surface highway, dual lane	WDD - Width (Decimeters) MCC - Material composition category
14	6	All weather loose surface highway, single lane/fair weather loose surface highway, single lane	WDD - Width (Decimeters) MCC - Material composition category
15	3	Road bridge - steel trestle	WDD - Width (Decimeters)
16	3	Railroad bridge - steel trestle	WDD - Width (Decimeters)
17	7	Bridge span	MCC - Material composition category
18	9	Tunnel, road/railroad	WDD - Width (Decimeters) MCC - Material composition category
19	1	Smudged runway	
20	1	Runway centerline stripes	
21	1	Runway touchdown zones	

Table A-5 Texture Patterns of Obstacles

Pattern Index	Variety	Description	Attribute(s)
1	1	Dragon teeth	
2	1	Pipeline	
3	1	Wall/fence	
4	1	Volcanic dike	
5	1	Crossing point	
6	1	Moat	
7	1	Escarpment	
8	1	Hedgerow	

Table A-6 Texture Patterns for Trees/Shrubs (profile)

Pattern Index	Variety	Description	Attribute(s)
1	1	Deciduous tree	
2	1	Conifer tree	
3	1	Palm tree	
4	1	Shrub	
5	1	Scrub/brush	
6	1	Cactus	
7	1	Forest wall	
8	1	Hedge wall	

Table A-7 Texture Patterns for Construction Materials

Pattern Index	Variety	Description	Attribute(s)
1	12	Wood	
2	3	Masonry - brick, block, stone	Color
3	3	Stone	Color
4	3	Stone slabs	Color
5	1	Thatched	
6	1	Corrugated metal	
7	1	Tar and gravel	
8	1	Wood shingle/shake	
9	3	Asphalt shingle	Color
10	3	Tile	Color
11	15	Window	Style
12	4	Door - personnel	Style
13	4	Door - equipment	Style
14	1	Concrete	
15	1	Prestressed concrete	
16	1	Reinforced concrete	
17	1	Asphalt	
18	5	Steel girder	Type

Table A-8 Texture Patterns for Environmental

Pattern Index	Variety	Description	Attribute(s)
1	18	Cloud patterns	Cloud type

Table A-9 Texture Patterns of Camouflage

Pattern Index	Variety	Description	Attribute(s)
1	64	Land vehicles - all countries	TBD
2	64	Sea vehicles - all countries	TBD
3	64	Air vehicles - all countries	TBD
4	64	Life forms - all countries	TBD

Table A-10 Texture Patterns for Microtexture

Pattern Index	Variety	Description	Attribute(s)
1	1	Bare/cleared	
2	1	Bedrock	
3	1	Clay	
4	1	Concrete	
5	1	Earthwork	
6	1	Evaporates	
7	1	Gravel	
8	1	Masonry (stone/brick)	
9	1	Paved	
10	1	Prestressed concrete	
11	1	Reinforced concrete	
12	1	Rock, rocky	
13	1	Sand	
14	1	Silt	
15	1	Soil	
16	1	Steel	
17	1	Stone	
18	1	Volcanic	
19	1	Wood	
20	6	Leaves	Type
21	3	Grass	Type

Appendix B Infrared Texture Pattern List

The infrared texture patterns we recommend including in the texture library are summarized below.

Table B-1 Texture Patterns and Categories

Pattern Category	Number of IR Texture Patterns
Vegetation	24
Natural Surface Materials	10
Surface Drainage	5
Transportation	21
Obstacles	8
Trees/Shrubs (profile)	8
Construction Materials	18
Environmental	1
Microtexture	21
Vehicles/personnel	20 ground vehicles, 5 views each 5 aircraft, 6 views each 5 ships, 5 views each 1 life form 5 views
Total	**276**

Phase I Study Final Report

TEXTURE LIBRARY

FOR 3-DIMENSIONAL VISUALIZATION SYSTEMS

revised
29 December 1993

CGSD

Computer Graphics Systems Development Corporation

2483 Old Middlefield Way #140
Mountain View, CA 94043-2330
415-903-4920
fax 415-967-5252