Real-time simulation of the seasonal and diurnal aspect changes in real 3D scenes

Heinz Hugli and Werner Frei

Medical Imaging Science Group, University of Southern California
4676 Admiralty Way, Suite 932, Marina del Rey, California 90291

Abstract

This paper presents the principle, the realisation and the results of a system simulating in real-time extremely realistic views of 3D scenes. The simulation considers two factors, namely the seasonal and diurnal illumination variation and the change in the atmospheric visibility. Such effects are responsible for the main aspect changes in real scenes like for instance mountainous terrains. The scene aspect simulation is based on information from a digital terrain model and albedo maps previously derived from aerial photographs. A scene model is assembled in a stage of preprocessing and is then used as the data base for the simulation. The generation of the image itself is performed in real-time. Any modification of the scene illumination and the atmospheric visibility within the scene can be entered interactively and the new aspect of the scene can be seen instantly on the display where the new images are updated at video rate. This simulation system is implemented on a display processor making an intensive use of its transform capabilities via video processor and look-up-tables. The results show typical illumination and visibility effects in various scenes simulating real natural terrain.

Introduction

In the last years, digital imagery has grown in large strides under the combined impulses of new developments in microelectronics as well as a rapidly increasing variety of applications. The earth is being measured intensively both from the space and the air. The earth resource satellite Landsat alone produces an information flow of 10^{11} bits per year. A huge amount of image information is thus available showing the earth surface in various spectral bands ranging from ultra-violet to radio frequencies. In parallel with this proliferation of real images, the development of computer generated synthetic images took place. Advances in computer graphics have led to the technique, and the development of large digital terrain models (DTM) to the practical possibility to simulate any arbitrary view of almost any site. Also, the digitization of the earth relief is still going on, making newer and better DTM available, either by digitizing uncovered reliefs or by producing DTM with an increased resolution.

As a result of these developments, information is now available both in large amount and in various types which gives a good description of the earth surface. The purpose of the work presented here is to combine all this information in order to simulate all possible aspects of a given scene, including new aspects. From a scene described analogically by its real images and intrinsically by its elevation model, images will be generated which show the scene under new aspects.

Generally, the aspect changes of a scene are numerous. In the context of this paper, we will consider those changes in the aspect of a scene which result from the diurnal and seasonal illumination effects as well as from atmospheric effects. Illumination and atmospheric effects have been intensively investigated and applied in remote sensing and computer graphics. Efforts in remote sensing were concentrated on correcting remote sensed images for illumination and atmospheric effects in order to produce the map of the intrinsic reflectances of the earth surface, the so-called albedo map. The accent in this field is on radiometric accuracy. Complex reflectance and atmospheric models are thus used which require large computation time[1,2]. In computer graphics, the aim is different: the simulations are oriented toward visual realism. The efforts in this field can roughly be divided into real-time applications where the constrain of limited computation time permits only to synthesize objects with a limited complexity, and, respectively, in high quality applications aimed at synthesizing the most realistic views of very complex objects where computation time is no object.

The originality of the present approach is to combine real-time simulation and real scenes. Indeed, the simulation is performed in real-time, based on a scene which includes real, high resolution information derived from aerial images. A concept will be used for its description. The simulation is based on the concept of a scene transformed into an image. The scene is the data base on which the transform applies and the image is its variable aspect. A characteristic of this concept is the assumption of a one-to-one geometric correspondance between scene and image. As a consequence, the transform becomes simple with the advantage that it can be performed in real-time. From the point of view of implementation however, this concept has a drawback. Because the scene has the same size as the image, the realisation of this concept requires a large amount of memory. We will show that this is no problem if the whole simulation concept is implemented within a display processor.

The present simulation can be considered as an example of how dedicated hardware can solve a computation problem requiring both fast computation and a large amount of memory. Recently, several commercial raster scan display systems evolved toward powerful display processors, adding array processing capabilities to the traditional functions of storage and display . By the description of its implementation and by the results obtained, we show that the present simulation concept fits into the structure of a typical display processor.

Simulation concept

Consider figure 1 which shows how a scene is transformed into an image showing one of its aspect. The scene is described by the attribute vector A(x,y) whose components A1(x,y), A2(x,y), ... An(x,y) are specific attributes of the scene like albedo, depth, temperature, etc. These attributes are available to a processor which generates the image I(x,y) made of the color components I1(x,y), I2(x,y) and I3(x,y). The transform itself is performed by the processor according to transform rules entered into the sytem. There is a one-to-one

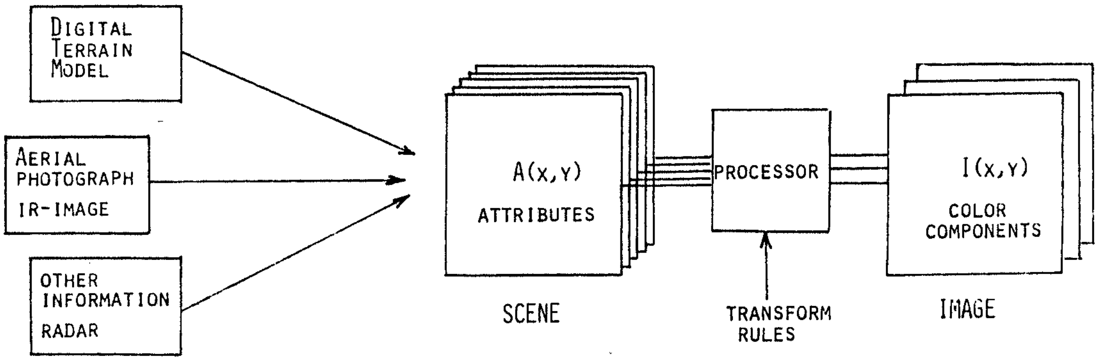


Figure 1. Concept of scene-to-image transform

geometric correspondance between the sceen and the image denoted by the common coordinates x,y in the scene and the image spaces. This characteristic makes the processing particularly simple.

A practical type of a spatial transform is to express a pixel of the image as a function of the attributes of the scene pixels which lie in its near neighborhood. For a 3*3 neighborhood, the transform is:

$$I(x,y) = T(\ A(x-1,y-1),\ A(x,y-1),\ ..\ ,\ A(x,y),\ ..\ ,\ A(x+1,y+1)\ ) \qquad (1)$$

An even simpler transform is the non-spatial transform:

$$I(x,y) = T(\ A(x,y)\ ) \qquad (2)$$

Note that a spatial transform can be converted into a non-spatial transform with an increased number of attributes, the new attributes being the former attributes shifted geometrically.

Scene description

The scene is described by its various attributes. Typical attributes of a scene on the earth surface are its spectral albedos, altitude, orientation, geometric depth, etc. Usually such a scene is not already available and it must therefore be built up prior to any other step. Suppose the available information of a given site consits of a digital elevation model and of remote sensed images. As these data are issued by different sources, they do generally not match geometrically and must therefore be registered in a first stage of preprocessing . A second stage is required to generate the spectral albedos from the original sensed data. It is called deshading and consists basically in removing the illumination and viewing effects from the original data . In a third and last stage of preprocessing, the geometrical attributes like the scene depth and orientation are computed from the digital elevation model. The eventual result of this preprocessing is a full set of attribute arrays describing the scene as it is used for the aspect simulation.

Aspect simulation

Depending on the illumination and the atmospheric conditions the scene appears under a different aspect. We are looking now at the mechanisms affecting the aspect. First, there is the illumination or shading effect. Figure 2 shows the conceptual arrangement of the light source, an element of the earth surface and the observer. The constant and collimated

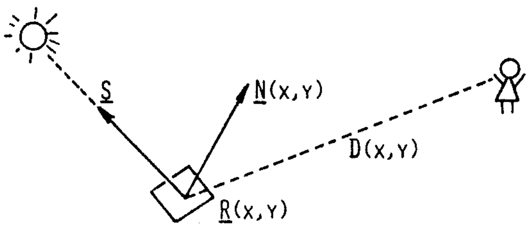


Figure 2. Geometry determining the illumination and visibility conditions

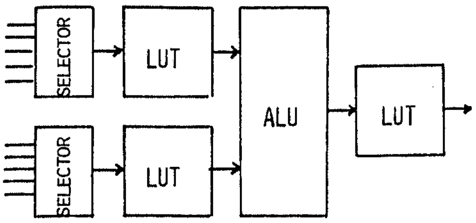


Figure 3. Basic architecture of a video processor

source located in the direction $\underline{S}$ irradiates a surface element characterized by its direction $\underline{N}(x,y)$ and its spectral albedos $Rk(x,y)$. Under the assumption of a lambertian reflection, its normalized radiance can be written as:

$$Ik'(x,y) = Rk(x,y) * \max(\, 0,\ \underline{S}.\underline{N}(x,y)\,) \qquad k = 1, 2, 3 \tag{3}$$

where the index k refers to the three color components of the image.

Second, there is the visibility effect. Scattering and absorption in the atmosphere modify the visibility of the scene. The apparent radiance perceived by the observer is given by[3]:

$$Ik''(x,y) = Ikoo + (\, Ik'(x,y) - Ikoo\,) * \exp(\, -Zk * D(x,y)\,) \tag{4}$$

where:
- $Ik'(x,y)$ is the zero-range radiance
- $Ikoo$ is the radiance of the horizont
- Zk is the scattering coefficient
- $D(x,y)$ is the scene depth

again, the index k refers to the three color components of the image.

These equations will now be further transformed to fit the context of scene to image transform described earlier. The overall transform can be expressed by:

$$Ik''(x,y) = T(\, N1(x,y),\ N2(x,y),\ Rk(x,y),\ D(x,y)\,) \tag{5}$$

which states that the scene which is described by the attributes N1, N2, R1, R2, R3 and D is transformed into the images I1, I2, I3. This is for color images. The case of monochrome images is given by k=1 only.

Note that the orientation vector $\underline{N}$ appears only in its two components N1 and N2. This is possible because the orientation has only two degrees of freedom. It is done formally by setting $\underline{N} = (\, N1,\ N2,\ 1\,)$. Note also that since the orientation is derived from the scene surface it is in fact a spatial function of the DTM in the sense of equation (1). This is a typical case where the property described above is used i.e. where the spatial transform was reduced to a non-spatial transform.

Display processor

Modern display processors are well adapted to implement the concept of the scene-to-image transform as depicted in figure 1. Their storage, processing and display capabilities make them very attractive in that context. Their storage capability includes several banks of fullsize image memories and their processor runs at video speed, transforming the information available in the memory banks into image data which are then directly displayed. Thus a full new image is computed in a 1/30 of a second.

A display processor is a specialized array processor whose processing principle is based on two mechanisms. The first one controls the data flow which runs from the memory banks through the video processor into the image display. Data access is strictly sequential and the transfer rate is constant as well as fixed by the video rate. In short, the data flow is invariable. The second mechanism applies to the data transform within the video processor. Data entering the processor run through transform units where they are processed. As the units are programmable, a change in the transform function is immediately applied to the whole data set. The very large data set is thus transformed in real-time.

So far, a simulation concept has been set up and its implementation with a display processor has been proposed. The implementation will now be described.

Implementation

After preprocessing, the various attribute arrays are loaded into the memory banks of the display processor. A host computer receives the aspect parameters and generates the transform rules which are loaded into the display processor. The aspect simulation is running. Before we go to the result, we will analyse the transform mechanisms i.e. how the equations (3), (4) and (5) are typically implemented in a video processor. After a brief description of the video processor, three different cases of implementation will be shown.

Video processor

The video processor is the processing unit within a display processor. Many different types of video processors are found in the various systems available on the market. However they all exhibit the same basic architecture shown in figure 3. Two input paths are fed through look-up tables (LUT) into an arithmetic and logic unit (ALU) which is the source for the output path running through a third LUT. Each input path may be connected to either memory bank by mean of a selector and the output drives either directly the display or is fed back into a memory bank. Usually, the ALU can add and subtract. Multiplication and division require a logarithmic conversion in the input path and an exponential conversion in the output path, a transform for which the 3 LUT are used. Very generally, the LUT are used whenever a non-linear transform is required.

Case 1: illumination

The product of albedo and shading in equation (3) is solved using the ALU and the LUT's as a multiplicator. Beyond that, the shading transform, which is a non-linear function of two variables must be computed. This can be done by coding two orientation components of the direction vector N into a single memory bank and performing the transform by LUT[6]. Both the shading and the logarithmic transform are thus computed with the same input LUT which works very fine indeed. There is with certain display processors a limiting factor to this method of computing the shading: coding 2 components of N into one memory bank may cause relatively large quantization steps and, as a result, unwanted contouring in the image.

Case 2: visibility

Equation (4) transforms the 2 attributes Ik'(x,y) and D(x,y) which are therefore applied to the inputs of the video processor. Here again, the product of the derived functions of both attributes is to be performed, for which the logarithmic and exponential convertions are used. A particularity of this implementation is that it requires to check the argument of the logarithm in order to prevent it from getting negative.

Case 3: multi-processing

The full transform of equation (5) cannot be computed directly with the video processor of figure 3. In fact, its implementation requires to cascade equations (3) and (4) which is not possible with a single ALU. This is a typical problem where the transform is too complex to be performed by a single processor. Generally, two solutions are available: the multi-processor and the multi-pass method. The multi-processor method consists of using several video processors to perform the task. The case of equation (5) would require therefore to cascade two video processors, the first for the illumination transform, the second for the visibility of the display processor transform. The other method, multi-pass processing, performs the transform iteratively, using the processor first for the illumination transform, then for the visibility transform. It makes use of the write-back capability, in which the output of the processor is written back into a memory bank where it is available as a new attribute for the next processing pass.

Results

The concept described above is implemented on a Grinnell GMR 270 display processor controlled by a Data General Eclipse AP/130 as a host. Prior to the simulation, the host

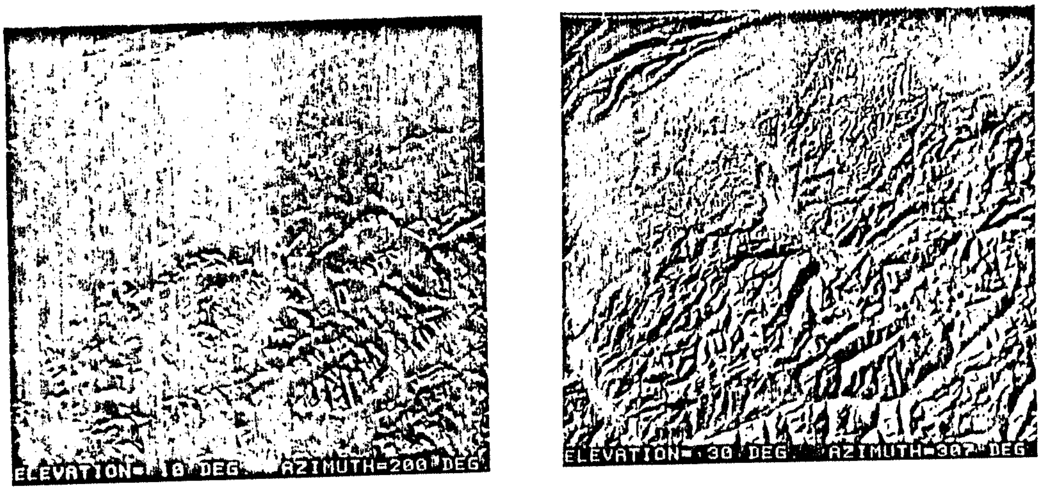
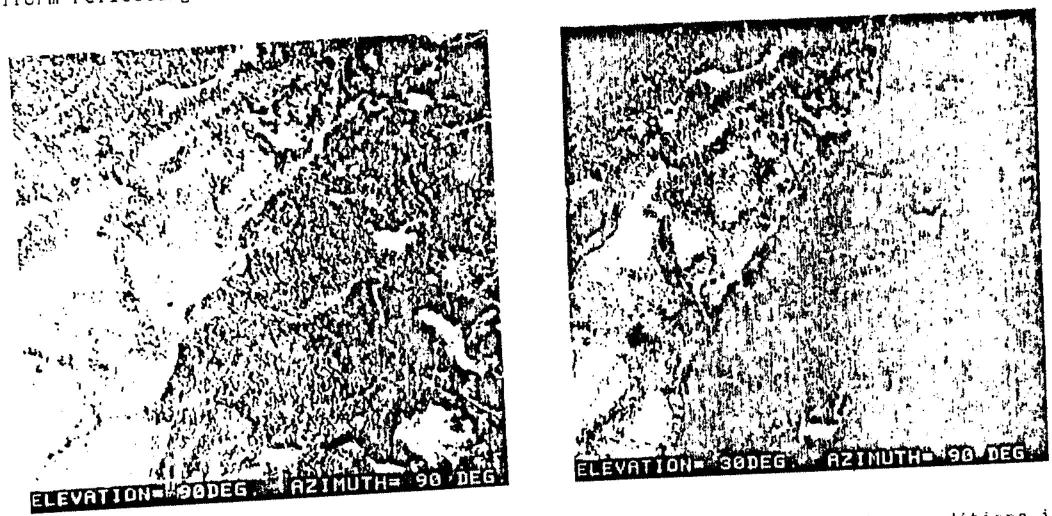


Figure 4. Simulation of aspect changes due to variation of the illumination conditions in a uniform reflecting mountainous terrain (Switzerland)

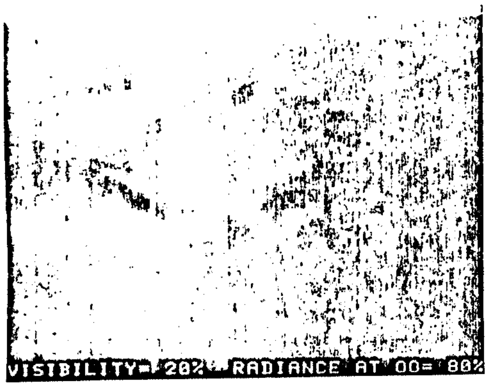


5. Simulation of aspect changes due to variations of the illumination conditions in a mountainous terrain with variable reflectances (Redondo Peak, New Mexico)

controls the loading of the scene into the memory banks. During the simulation itself, the illumination and the visibility parameters are varied manually by mean of a trackball and entered into the host which generates the transforms and the commands to be transfered to the display processor.

Figure 4 shows the real-time simulation of aspect changes due to illumination effects. The scene has a constant albedo and only the shading part of equation (3) is therefore active here. The transform uses a single LUT and the two orientation components N1 and N2 are coded with only 4 bits each. The quantization effects were reduced to a minimum by taking advantage of the property specific to digital terrain models to have a limited slope range and therefore also a limited range for N1 and N2.

Figure 5 shows the real-time simulation of aspect changes of a scene with natural albedo due to illumination effects. The scene has two attributes: albedo and orientation. It is transformed according to equation (3). The albedo map was derived from an aerial



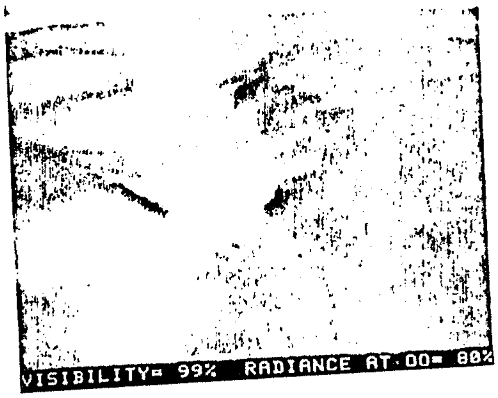


Figure 6. Simulation of aspect changes due to variations both of the illumination and visibility conditions in a perspective view of a uniform reflecting mountainous terrain. (Redondo Peak, New Mexico)

photograph during a phase of preprocessing which includes registration and deshading.

Figure 6 shows the real-time simulation of aspect changes due to visibility effects in a perspective view of a terrain. The scene has two attributes, namely orientation and scene depth, and it is transformed according to equation (5) with a constant albedo. Here, the required preprocessing was to generate the perspective view of the scene. This was done by mean of an image synthesis package based on a depth buffer hidden surface algorithm, which is the only algorithm producing both the image array as well as the depth array of the given scene.

Summary and conclusion

The aspect changes of a scene caused by illumination and visibility changes were simulated using a display processor to implement the real-time part of the processing. The real advantage of this simulation concept are its versatility to implement various effects, its very short response time and its ability to generate high resolution images without varying the computation time.

Aknowledgements

This work is a spin-off of the environmental remote sensing work performed for the Departement of Energy under contract No. DE-AS03-76-SF00113.

References

1. Horn B.K.P., 'Hill Shading and the Reflectance Map', Proc. IEEE, Vol. 69, No. 1, Jan. 1981
2. Hugli H. and Frei W., 'Correcting for Anisotropic Reflectances in Mountainous Sites', Symp. on Machine Processing of Remotely Sensed Data, Purdue, June 1981
3. RCA, 'Electro-Optics Handbook', RCA 1974, p.93
4. Frei W., Shibata T. and Chen C.C., 'Fast Image Matching with False Fix Protection', Intern. Conf. on Pattern Recognition, Miami, Dec. 1980
5. Driscoll T. and Walker Ch., 'Evolution of Image Processing Algorithms from Software to Hardware', SPIE Conf. Los Angeles, Feb. 1981, Volume 271
6. Sloan K.R. and Brown C.M., 'Color Map Techniques', Computer Graphics and Image Processing, Vol. 10, No. 4, Aug. 1979, p. 297-317