# A Pattern Matching Technique for Detecting Similar 3D Terrain Segments


MOTOFUMI T. SUZUKI
National Institute of Multimedia Education
2-12 Wakaba, Mihama-ku, Chiba, 2610014
JAPAN

http://www.nime.ac.jp/

TATSUYA SHIBATA
School of Information Environment
Tokyo Denki University
Chiba New Town Campus, 2701382
JAPAN
http://www.sie.dendai.ac.jp/



***Abstract:*** 3D terrain data have been used for various systems such as navigation and landscape analysis systems. Recently, the number of 3D terrain data and databases is increasing rapidly. To handle these data effective, such systems need the ability to search 3D terrains to find matches based on shape similarity. In this paper, we propose a pattern matching technique for searching similar terrain segments from databases of 3D terrain. Since 3D terrain data are composed of three dimensional data, pattern matching computation costs are extremely high. One of the most important processes for pattern matching is the extraction of shape features from the target 3D terrain data, which requires a significant number of computations. In our system, we have applied HLAC (Higher Order Local Autocorrelation) shape features for detecting similar 3D terrain segments. The use of HLAC shape features enables fast scanning and extraction of 3D shape features by reflecting the shift invariant properties of the HLAC. Our preliminary search system successfully detects similar 3D terrain segments from databases of 3D terrains. Our pattern matching technique can be applied to various fields of research including geographical data classification and landscape visualizations.



***Key-Words:*** HLAC, Similarity Retrieval, Pattern Matching, Voxel, Terrain


## 1 Introduction

Intensive research has been conducted for similarity search methods for 3D models, and various experimental search systems have been implemented. However, most similarity search systems do not have functions that search for similar portions or segments of the 3D models. In this paper, we propose a pattern matching technique for searching for similar terrain segments from databases of 3D terrain.

Two problems occur when searching for similar portions of 3D models; (1) a segmentation problem and (2) a computing cost problem. Although humans can instantly recognize parts or segments of 3D models, computers require certain segmentation algorithms. A simple segmentation technique involves the breaking up of 3D models into tiny parts and computing all the possible combinations of segments by merging the parts. For example, 3D models can be broken up into tiny triangle patches and the patches are recomposed to form segments. The system can search for similar segments against all the recomposed segments. However, this method requires a considerable amount of time for 3D models with a large number of triangles, and an efficient searching algorithm such as a dynamic programming approach is needed. To avoid enormous computation requirements, the system can utilize several attributes rather than finding all the possible combinations of recomposed segments. For example, segmentation can be done based on several attributes such as (a) shapes, (b) colors and (c) textures. Since the algorithms detect parts and portions of 3D models based on these attributes, the number of segments required for computation is reduced.

Although the method reduces the number of computations for extracting shape features, it is still a time consuming task if the target 3D terrain is composed of large data sets. To avoid such problems, we have applied HLAC (Higher Order Local Autocorrelation) shape features to extract segmented 3D terrains. Since the HLAC shape features have shift-invariant properties, the system can reduce the number of shape feature extraction computations for scanning 3D terrain segments.

## 2 Finding similar 3D Terrain segments

In this section, (1) 3D terrain, (2) shape features, and (3) HLAC shape features are discussed.

### 2.1 3D Terrain

3D terrain data have been used for various systems, such as navigation, electrical map, and geographical analysis systems. There are many 3D terrain data samples available freely and commercially on the Internet. 3D terrain data are useful for visualizing terrain information from various viewpoints by using computer graphics techniques. There are several ways to describe 3D terrain information. For example, file formats such as USGS DEM, DTED, and BT are used. Generally, the terrains are described in forms that represent the grids of elevation points. Most of the file formats can be translated into popular 3D files by using certain file converter software programs and can be visualized by using computer graphics rendering engines. Figure 1 shows a screen shot of the 3D terrain which was translated from the USGS DEM format file for visualization.

In our experiment, 3D terrain data from the GSI [20] is used. The data are represented in DEM (Digital Elevation Model) which is a popular elevation file formats for describing terrain data. The data set used for our experiments contains terrain data of Japan. The terrain information is divided into grids and elevation information for each grid, which is described in individual files. There are several hundreds files for the entire data sets.

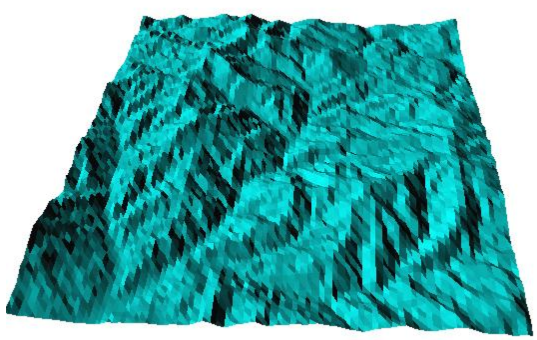

Figure 1: 3D Terrain data

### 2.2 Shape Features

Several processes are needed for extracting shape features of 3D terrains as shown in Figure 2. As a first step, 3D terrain data are converted to polygonal models. The models are then converted to voxel models for normalization purposes so that the system can support not only a DEM file format but also other terrain file formats. In our experiment, MEM file format terrains are converted to 3D polygonal model format (VRML) terrains composed of vertices and triangular faces. The polygonized terrain data are interpolated during the rendering to smooth out terrain surfaces to avoid jaggy surfaces. Once 3D terrains are polygonized, they are voxelized to fill empty volume areas. For the voxelization, the modified technique described in papers [10][11] are used to handle 3D terrain data

Once the 3D terrain data are voxelized, the system can extract HLAC shape features. HLAC shape features have been used in a wide range of software applications, including texture analysis, face recognition, and character recognition. HLAC shape features are known as shift invariant shape features and have been used for research in the field of pattern recognition and classification. The details of HLAC shape features can be found in papers [1][2][3] and [5]. Although HLAC shape features are suitable for handling 2D images, there are some limitations in using them as typical 2D HLAC mask patterns for extracting shape features of 3D voxel data. Therefore, 2D HLAC mask patterns have been extended to 3D HLAC mask patterns in order to handle 3D data more efficiently. The 3D HLAC mask patterns used in this research are shown in Figure 3. The 3D mask is represented in three dimensional boxes as shown in the Figure 3. Autocorrelation functions with an order n=0 and n=1 are used in this case. More details of 3D HLAC mask patterns can be found papers [18] and [19]. Also, electrical HLAC mask pattern files can be downloaded from the following web site (http://open.nime.ac.jp/). Intuitively, the use of 3D HLAC mask patterns is similar to the use of 2D HLAC mask patterns applied in pattern recognition research for 2D images. 2D HLAC mask patterns are often applied to 2D pixel data. However, the extensions of dimensions for the mask enable 3D HLAC masks to analyze 3D voxel data. The extracted HLAC shape features represent the characteristics of the 3D terrains. The shape features have very small file sizes, but they still describe well the characteristics of 3D terrain. Therefore, shape features are suitable for indices to search for 3D terrain data, because the system needs to analyze small files for fast computations.

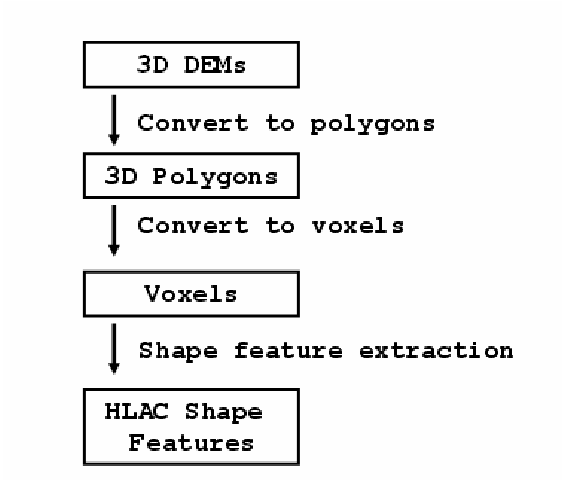


Figure 2: Processes needed for extracting shape features

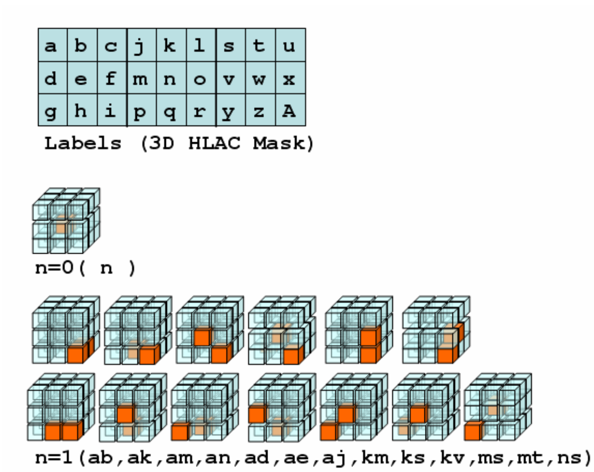


Figure 3: 3D HLAC mask patterns (n=0 and n=1)

## 2.3 Scanning of HLAC Shape Features

In this research, the system needs to recognize particular segments of 3D terrain. The system scans the entire 3D terrain volume to find segments of interest which are similar to a search key segment. Since the terrain data are described as three dimensional voxel data, the scanning computing costs are exponentially high. However, the use of HLAC shape features successfully reduces the number of computations by reflecting the sift-invariant properties of HLAC functions.

Figure 4 shows (a) a 3D terrain, (b) a window for scanning and (c) a HLAC mask. The window moves inside of the 3D terrain and extracts voxel data. When the voxel data are extracted, a HLAC mask is applied to compute HLAC shape features. These processes seem to require extremely high computation costs. However, if the window moves inside of the 3D terrain sequentially, the system needs to compute the HLAC shape features where the portion of the window moved. For instance, as shown in Figure 5, the system needs to compute only HLAC shape features of volume V3. The system subtracts the HLAC shape feature values of volume V1 from volume V, and the shape feature values are updated by adding the HLAC shape feature values of volume V2 and V3. This means that the system does not have to recalculate the entire volume V, which saves computing costs.

It is important that the movement of the scanning window is sequential so that the HLAC values of the entire volume are not cleared. This can be easily implemented to the system by using typical computer programming languages. For instance, in computer programming language, switching the starting value and ending value of "for-loops" makes possible the locations of volume move continuous. Also, it is convenient if HLAC feature values of each voxel are pre-computed and stored as a database. This enables the system to speed up the scanning processes, because the system needs only to perform one add and one subtract operation

The HLAC shape features are extracted when the scanning window moves to new positions and the shape features are stored. The stored shape features are compared to the search key shape features by using histograms. Since the similar terrain segments tend to show similar histogram values, the system can sort the terrain segments by similarity orders.

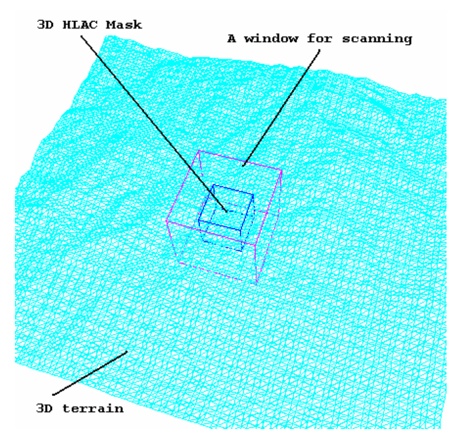


Figure 4: (a) a 3D terrain, (b) a window for scanning and (c) a HLAC mask.

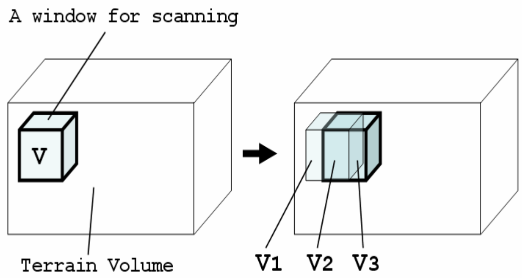


Figure 5: A window for scanning

# 3 Experiments and Results

In this section, (1) a search system, (2) computation efficiencies, and (3) examples of a similarity search are discussed.

## 3.1 A Search System

The experimental system is implemented by C language. A GNU gcc compiler running on a Linux operating system (Fedora Core 2) is used. A Pentium 4 - 2.66 GHz CPU with 1024 Mbytes of memory is used to compute HLAC features. Figure 6 shows the interface of the system. In this system, users can choose a 3D terrain segment as a search key, and the system looks for similar 3D terrain segments based on similarity orders by scanning the shape features of 3D terrains. The retrieved 3D terrain segments can be visualized from various viewpoints by using buttons and a mouse as shown in Figure 6. Similarities of the retrieved 3D terrain segments are marked by using colored spheres. The color of the spheres shows similarity values in which dark spheres indicate extremely similar, while light spheres indicate less similar against the search key.

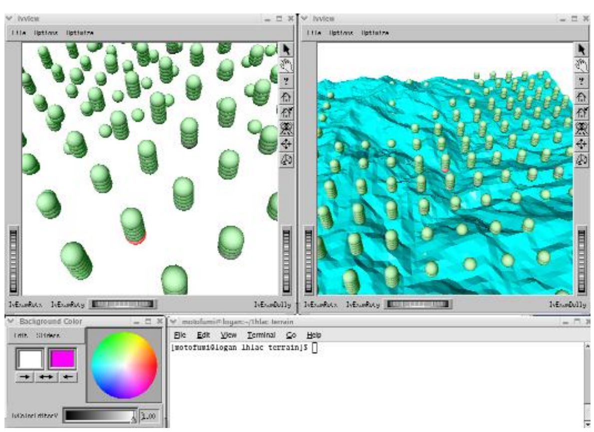

Figure 6: A search system for 3D terrain data

## 3.2 Computation Efficiencies

Since the HLAC shape feature scanning processes are conducted sequentially and a small portion of voxels are computed, computation costs remain low compared to computing the entire volume of voxels. As shown in Figure 7, in the case for the window size of 15x15x15, HLAC shape feature computation for the entire volume requires 70488 steps of multiplication. However, if the portion of volume is computed, only 23495 steps of multiplication, one addition and one subtraction are needed.

We have compared time efficiencies of (a) simple sequential scanning and (b) sequential scanning described in the previous sub-section as shown in Figure 8. For the (a) simple sequential scanning, HLAC shape feature values of the scanning windows are refreshed, and HLAC shape feature values are recomputed whenever the scanning window moves to new positions. In contrast, the (b) sequential scanning described in the previous sub-section computes HLAC shape features for portions of the scanning window which moves. In this case extra add and subtract operations are needed. However, these extra operation computing costs are insignificant in comparison to computing the entire HLAC shape feature of the scanning window. As shown in Figure 8, the terrain scanning method (a) requires more computation time compared to the (b) method when the size of scanning windows is large.

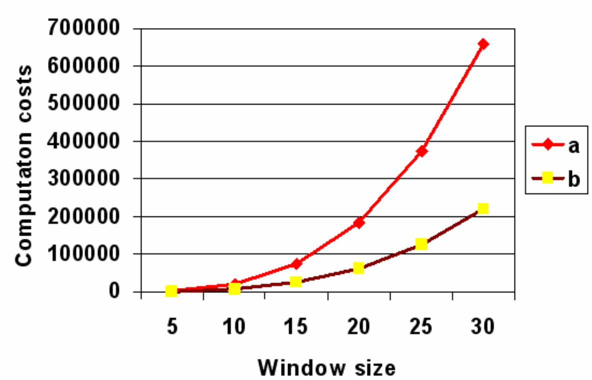


Figure 7: HLAC computations with various window sizes.

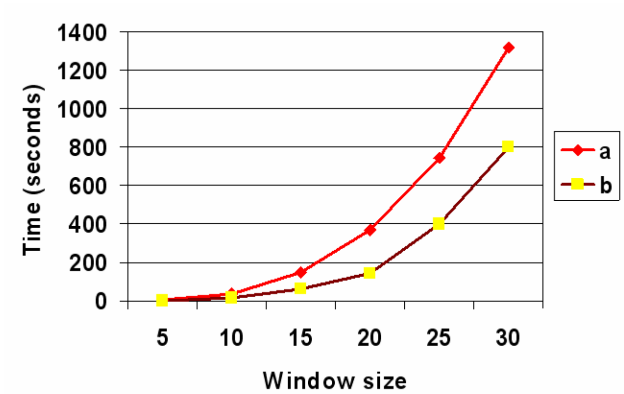

Figure 8: Time needed for scanning methods (a) and (b)

### 3.3 Examples of a Similarity Search

Figures 9a, 9b, and 9c show examples of a similarity search for 3D terrain segments. In the figures, 3D terrain segments similar to a search key are indicated by sphere markers. The colors of the spheres indicate similarity, and the black spheres indicate extremely similar, while the white spheres indicate less similar. The search keys are indicated by the label 'k' in the figures. In the examples, a terrain data size of 64x64x64, and a scanning windows size of 16x16x16 are used for the system evaluation. The system successfully searches for similar 3D terrain segments regardless of the shape of the search key terrain segments.

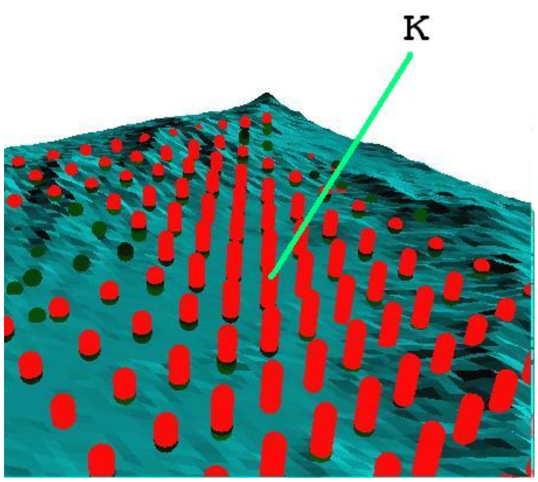

Figure 9a: Similarity search 1

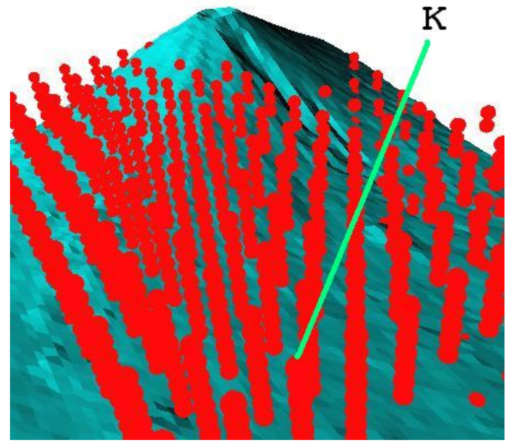

Figure 9b: Similarity search 2

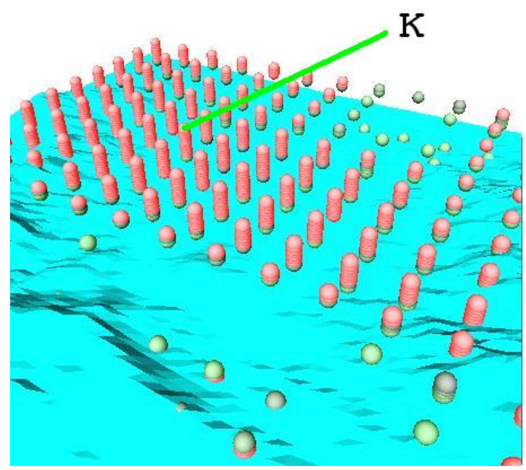

Figure 9c: Similarity search 3

## 4 Conclusion

3D terrain data are becoming more commonly used in various fields reflecting recent developments of high speed computers. Pattern matching and similarity retrieval techniques are essential for 3D terrain databases and systems. One of the most important processes for pattern matching is the extraction of shape features from 3D terrain data. However, the pattern matching computation costs are extremely large for 3D data, and huge computations are required. In our system, we have applied HLAC (Higher Order Local Autocorrelation) shape features for finding similar 3D terrain segments. By using HLAC shape features, our preliminary system enables fast scanning and extraction of 3D shape features by reflecting the shift invariant properties of the HLAC. Our preliminary experiment showed that the system successfully detects similar 3D terrain

segments from the databases of 3D terrain. Also, the system searches for similar 3D terrain fairly fast by using our sequential scanning techniques.
In this research, the system was implemented to run HLAC shape feature computation for a single CPU. As a future work, we plan to extend the system to support multiple CPUs for high speed computations because HLAC shape feature computations are easily parallelized.

***Acknowledgements***

This research was supported by research grants from the National Institute of Information and Communications Technology (NICT) and a grant-in-aid from the Ministry of Education, Science, Sports and Culture, Japan (#EYS-B(2)-15700115).

***References***

[1] T. Kurita, N. Otsu, and T. Sato, Proceedings of 11th International Conference on Pattern Recognition, Vol.II, pp.213-216, 1992.
[2] N. Otsu and T. Kurita, A new scheme for practical flexible and intelligent vision systems, Proceedings of IAPR Workshop on Computer Vision, Tokyo, pp.431-4352, 1998.
[3] K. Hotta, T. Kurita and T. Mishima, Scale Invariant Face Detection Method using Higher-Order Local Autocorrelation Features extracted from Log-Polar Image, Third IEEE International Conference on Automatic Face and Gesture Recognition, pp.70-75, 1998.
[4] F. Goudail, E. Lange, T. Iwamoto, K. Kyuma and N. Otsu, Face Recognition System using Local Autocorrelations and Multi-scale Integration, IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 18, No. 10, pp.1024-1028, 1996.
[5] V. Popvici and J.P. Thiran, Higher Order Autocorrelations for Pattern Classification, Proceedings of the International Conference on Image Processing, 2001.
[6] J.A. MacLaughlin and J. Raviv, Nth-Order Autocorrelations in Pattern Recognition, Information and Control, Vol.12, pp.121-142, 1968.
[7] T. Kurita and S Hayamizu, Gesture Recognition Using higher order local autocorrelation features of PARCOR, IEICE Trans. Inf. & Syst., Vol.E86-D, No.4, April, 2003.
[8] D. Chetverikov and Z. Foldvari, Affine-Invariant Texture Analysis in Machine Vision, World Scientific, 2000, Series in Machine Perception and Artificial Intelligence.
[9] M. Kubo, Z. Aghabari, K.S Oh, and A. Makinouchi, Image Retrieval by Edge Features Using Higher Order Autocorrelation in a SOM Environment. IEICE Trans. Info and Sys., Vol. E86-D, No.8, Aug. 2003.
[10] J. Huang, R. Yagel,V. Filippov, Y. Kurzion, An Accurate Method for Voxelizing Polygon Meshes, ACM 1998 Symposium on Volume Visualization, 1998, 119 – 126.
[11] Tomas Akenine-Moller, Fast 3D Triangle-Box Overlap Testing, Journal of graphics tools 6(1):29-31, 2001
[12] Ken Perlin, An image synthesizer, Computer Graphics, (SIGGRAPH 1985 Proceedings), Volume 19, pp.287—296, 1985.
[13] D. S. Ebert, F. K. Musgrave, D. Peachey, K. Perlin, and S. Worley, Texturing and Modeling, A Procedural Approach, ISBN 0-12-228760-6, Academic Press, 1994.
[14] K. S. Fu and S. Y. Lu. Computer generation of texture using a syntactic approach, Computer Graphics (SIGGRAPH 1978), Vol.12, pp.147-152, 1978.
[15] S. Haruyama and B. A. Barsky, Using stochastic modeling for texture generation, IEEE Computer Graphics and Applications, 4(3) pp.7-19, 1984.
[16] B. J. Schacter and N. Ahuja, Random Pattern Generation Processes, Computer Graphics and Image Processing, Vol.10, pp.95-114, 1979.
[17] P. Dalgaard, Introductory Statistics with R, Springer-Verg, ISBN 0-387-95475-9, 2002.
[18] M. T. Suzuki, Y. Yaginuma, and N. Osawa, A HLAC Shape Descriptor Extraction Method for 3D Solid Textures, The World Scientific Engineering Academy and Society Transaction on Computers, Issue 3, Volume 3, pp.768--773, ISSN:1109-2750, 2004.
[19] M. T. Suzuki, Y. Yaginuma, and N. Osawa, A Similarity Evaluation Method for 3D Models by Using HLAC Mask Patterns, The World Scientific Engineering Academy and Society Transaction on Computers, Issue 3, Volume 3, pp.713--718, ISSN:1109-2750, 2004.
[20] The Geographical Survey Institute, Japan http://www.gsi.go.jp/