A Pattern Recognition Technique for Voxel Data based on 3D Masks

MOTOFUMI T. SUZUKI, YOSHITOMO YAGINUMA, HIROSHI KATO, TSUNEO YAMADA

National Institute of Multimedia Education
2-12 Wakaba, Mihama-ku, Chiba-shi, Chiba 261-0014,
JAPAN

Abstract: This paper describes a pattern recognition technique for voxel data based on 3D HLAC (Higher order Local Autocorrelation) masks. HLAC has been used as feature descriptors for various pattern recognition applications including gesture recognition, face recognition, and 2D image retrieval. In this research, popular 2D HLAC masks are extended to 3D HLAC masks to handle 3D voxel data. Our experimental system extracts pattern features from voxel data, and the features are processed by using multiple regression analysis to recognize voxel patterns. The system successfully recognizes voxel patterns, and can count number of patterns statistically. Since HLAC masks have shift invariant characteristics, the system can recognize voxel patterns regardless of its positions and can search for the patterns quickly. The recognition system can learn voxel patterns from sample voxel data by using 3D HLAC masks and statistical analysis. Therefore, the system does not require any program modification for classifying patterns even if the target patterns are changed.

Key-Words: HLAC, 3D masks, Voxel, Shape descriptors, Pattern recognition

1 Introduction

HLAC features have been used for various pattern recognition and classification applications. For example, face recognition [3], character recognition [6], texture analysis [8], and image retrieval [9]. Important techniques related to HLAC can be found in various papers [1][2][3][7]. In these papers, autocorrelations up to the second order have been used, and they reduce the number of computations by using the shift-invariant property of the autocorrelation functions. Typically, 2D HLAC mask patterns (3 x 3) have been used for conjunctions with statistical approaches, and various multimedia data can be processed. 2D HLAC masks are suited for analyzing 2D images, but there are some limitations of analyzing 3D voxel data. Therefore, in this research, 2D masks (3 x 3) are extended to 3D masks (3 x 3 x 3) for extracting shape descriptors to handle 3D voxel data. Pattern features of voxel data are extracted using extended 3D HLAC masks, and the features are analyzed using multiple regression analysis. By using this technique, our system can detect particular patterns in voxel data which makes it possible to count the number of patterns contained in a set of voxel data. Figure 1 shows the intuitive concept of the research purposes. In the figure, each point of the cell contains the intensity value, and various intensity values form unique three dimensional patterns. Several sphere patterns are represented with high intensity values in the figures. In our research, we attempt to detect and count these spheres efficiently by applying multiple regressions to 3D HLAC features. This simple sphere pattern recognition method can be extended to a broad range of pattern recognition applications which use scientific voxel data, such as medical data and geographical data.

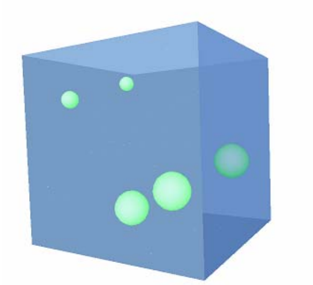


Figure 1: A cube and spheres represented as voxels

2 Pattern Classification using 3D Masks

This section describes (1) HLAC features, (2)3D masks, and (3) multiple regression analysis.

2.1 HLAC features

In our system, autocorrelations up to the second order have been used. By restricting the order of the autocorrelation functions, the number of computations is reduced. The details of HLAC features can be found in various papers [1][2][3]. Although typical HLAC local mask patterns limit the range of displacements to within a local 3 x 3 grid, we have limited the range to 3 x 3 x 3 grid in our system. The position of the mask pattern in the 3 x 3 x 3 grid is denoted by the three dimensional coordinates x, y and z. $L(x, y, z)$ denotes the mask pattern of autocorrelation function f_i. Each function f is denoted as

$$f_{0(a)} = \sum_x \sum_y \sum_z L(x-1, y+1, z+1)$$

$$f_{1(ab)} = \sum_x \sum_y \sum_z L(x-1, y+1, z+1) \times$$
$$L(x, y+1, z+1)$$

...

$$f_{13(ns)} = \sum_x \sum_y \sum_z L(x, y, z) L(x-1, y+1, z-1)$$

$$f_{14(akl)} = \sum_x \sum_y \sum_z L(x-1, y+1, z+1) \times$$
$$L(x, y+1, z) L(x+1, y+1, z)$$

$$f_{15(akm)} = \sum_x \sum_y \sum_z L(x-1, y+1, z+1) \times$$
$$L(x, y+1, z) L(x-1, y, z)$$

...

$$f_{250(ins)} = \sum_x \sum_y \sum_z L(x+1, y-1, z+1) \times$$
$$L(x, y, z) \times L(x-1, y+1, z-1)$$

The HLAC features f_i are computed by scanning the 3D voxel cube with the 3D HLAC masks. The intensity values of voxels are summed where the masks positions are non-zero value. By repeating the scanning of the 3D voxel cube, 3D HLAC features are computed, and these values represent the pattern features of the 3D voxel cube.

2.2 3D Masks

The 3D mask patterns in string sequence form are shown in Figure 2. The 3D HLAC mask is a 3D solid cube which is divided into a 3 x 3 x 3 grid. It contains 27 cells and is labeled with unique alphabets as shown in the figure. Since the autocorrelation functions of the order N=0, 1 and 2 are used for HLAC features, the sizes of the string sequences are a length of 3 at most as shown in Figure 2. The figure shows all the possible combinations of 3D HLAC mask patterns. These mask patterns are applied to voxel data for extracting HLAC features.

a -++	b 0++	c +++	j -+0	k 0+0	l ++0	s -+-	t 0+-	u ++-
d -0+	e 00+	f +0+	m -00	n 000	o +00	v -0-	w 00-	x +0-
g --+	h 0-+	i +-+	p --0	q 0-0	r +-0	y ---	z 0--	A +--

N=0 (1): a
N=1 (13): ab, ak, am, an, ad, ae, aj, km, ks, kv, ms, mt, ns
N=2 (237): akl, akm, akn, ako, aks, akt, aku, akv, akw, akx, abk, abl, aln, abm, abn, abo, abc, amn, amp, amq, ams, amt, amv, amw, amy, amz, abd, ano, anp, anq, anr, ans, ant, anu, anv, anw, anx, any, anz, anA, abe, abf, abj, ack, acn, ace, adk, adm, adn, adp, adq, ade, adg, adh, adj, aek, ael, aem, aen, aeo, aep, aeq, aer, aef, aeg, aeh, aei, aej, afk, afn, agm, agn, ahm, ahn, ain, ajk, ajm, ajn, ajs, ajt, ajv, ajw, klm, kls, klv, kmn, kmo, kmp, kmq, kms, kmt, kmu, kmv, kmw, kmx, kmy, kmz, knp, kns, knv, kny, kos, kov, kpv, kpw, kqv, kst, ksu, ksv, ksw, ksx, ktv, kuv, kvw, kvx, kvy, kvz, kwy, bks, bkv, lmn, lmt, lmw, lnp, lns, lnv, lny, lpw, lst, lsw, ltv, lvw, bms, bmt, bmv, bmw, lwy, bmy, bmz, bns, bnv, bny, mns, mnt, mnu, mot, mps, mpt, mqs, mqt, mst, msv, msw, msy, msz, mtu, mtv, mtw, mtx, mty, mtz, muw, nos, nps, npt, npu, nqs, nrs, nst, nsu, nsv, nsw, nsx, nsy, nsz, nsA, ntv, nty, nuv, nuy, ost, osw, otv, psv,psw,ptv,ptw,puw, qsv, qsw, qtv, rsw, bjs, bjt, bjv, bjw, cks, ckv, cns, cnv, cny, dks, dkt, dku, dkv, dkw, dkx, dms, dmt, dns, dnt, dnu, djs, djt, djv, djw, eks, ekv, ems, emt, ens, ejs, ejt, ejv, ejw, fks, fkv, fns, gms, gmt, gns, gnt, gnu, hms, hmt, hns, ins

Figure 2: HLAC mask patterns (3 x 3 x 3) for 3D voxels in string sequence form

The intensity values of the voxels are multiplied to the masks where the cells are valid, and these values are summed as HLAC features. This process is repeated for various voxel positions, and 251 HLAC features are extracted from the voxel cube. These HLAC features have unique values and distributions by reflecting the intensity patterns of the cubes, and they can be used for content-based similarity retrieval [18][19] indices for 3D solid textures [13]. In this research, the HLAC features are used for recognition of patterns rather than indices for similarity retrievals. Not all the HLAC features are needed to classify or recognize the particular patterns, and the system needs to select relatively important HLACs for efficient computations. This HLAC selection can be done statistically using multiple regression analysis with a sufficient amount of sample voxel data. The following section examines multiple regression analysis.

2.3 Multiple Regressions

The general purpose of multiple regression analysis is to find the relationship between several independent variables and a criterion variable. In general, multiple regression analysis estimates a linear equation of the form:

$$y = a_0 x_0 + a_1 x_1 + \cdots + a_n x_n + b$$

The y variable can be expressed in terms of a constant b and slopes a_i times the x_i variables. The constant is also referred to as the intercept, and the slopes as the regression coefficients.

In this research, the aim of the system is to find patterns and count how many patterns exist in the voxel cube data. Therefore, the system needs to find the relations between these HLAC features and the number of patterns. Thus, in our statistical model, the linear equation should predict the number of spheres as a function of HLAC features, and multiple regression analysis is conducted. From the analysis results, regression functions are formed, and these functions can predict the number of patterns from HLAC features.

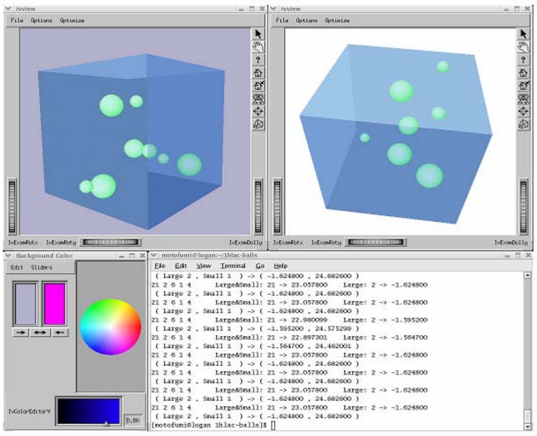
3 Experiments

This section describes, (1) a pattern recognition system, (2) pattern counting and (3) pattern classification.

3.1 A Pattern Recognition System

The experimental system is implemented by C language. A GNU gcc compiler running on a Linux operating system (Fedora Core 1) is used. The statistic software program "R" [17] is used as a software module for the multiple regression analysis. A Pentium 4 - 2.66 Mhz CPU with 1024 Mbytes of memory is used to compute HLAC features. Figure 3 shows the interface of the system which outputs a cube containing sphere patterns. Although the actual cube is represented and treated as a 64 x 64 x 64 voxel, the system outputs the cube and spheres as 3D polygonal models for fast visualization.

As described in the following sub sections, 500 sample voxel cubes are generated. Table 1 shows the approximate time needed for (a) voxel cube generation, (b) HLAC feature extraction, and (c) multiple regression analysis. All of these processes are needed as pre-computations for the system to learn patterns. Once these processes are carried out, the system needs to compute regression functions and HLAC features to determine specific patterns, all of which can be accomplished in 1 to 3 seconds.



```
 ( Large 2 , Small 1  ) -> ( -1.624800 , 24.682600 )
21 2 6 1 4      Large&Small: 21 -> 23.057800    Large: 2 -> -1.624800
 ( Large 2 , Small 1  ) -> ( -1.624800 , 24.682600 )
21 2 6 1 4      Large&Small: 21 -> 23.057800    Large: 2 -> -1.624800
 ( Large 2 , Small 1  ) -> ( -1.624800 , 24.682600 )
21 2 6 1 4      Large&Small: 21 -> 22.980099    Large: 2 -> -1.595200
 ( Large 2 , Small 1  ) -> ( -1.595200 , 24.575299 )
21 2 6 1 4      Large&Small: 21 -> 22.897301    Large: 2 -> -1.564700
 ( Large 2 , Small 1  ) -> ( -1.564700 , 24.462001 )
21 2 6 1 4      Large&Small: 21 -> 23.057800    Large: 2 -> -1.624800
 ( Large 2 , Small 1  ) -> ( -1.624800 , 24.682600 )
21 2 6 1 4      Large&Small: 21 -> 23.057800    Large: 2 -> -1.624800
 ( Large 2 , Small 1  ) -> ( -1.624800 , 24.682600 )
21 2 6 1 4      Large&Small: 21 -> 23.057800    Large: 2 -> -1.624800
 ( Large 2 , Small 1  ) -> ( -1.624800 , 24.682600 )
21 2 6 1 4      Large&Small: 21 -> 23.057800    Large: 2 -> -1.624800
 ( Large 2 , Small 1  ) -> ( -1.624800 , 24.682600 )
[motofumi@logan 1hlac-balls]$ 
```

Figure 3: A pattern recognition system

	64^3	32^3	16^3
Voxel generation	5674	1280	312
HLAC extraction	612	141	30
M.R. analysis	3.5	3.3	3.5

(Seconds)

Table 1: Time needed for processing voxel data

3.2 Pattern counting

Since the equation for the multiple regressions contains 251 explanatory variables, the same number of parameters has to be estimated. To estimate the parameters, a statistically sufficient amount of sample data is needed. Five hundred sample data are generated which is about twice the number of the explanatory variables, and it is a sufficient number of samples for estimating regression functions. The sample data are generated by using a software program which sets up a cube the size of 64 x 64 x 64 filled with voxels. Also, a random number of spheres with two different radius sizes is placed inside the cube. The program controls the positions of the spheres so that each sphere does not intersect with other spheres and the cube boundaries. Some noises are added to the cubes to emulate typical sample data. An example of sample data is shown in Figure 4. The cube is filled with 0 intensity values, and it contains large and small spheres with 1.0 intensity values where the 1.0 is the maximum value for the intensity.

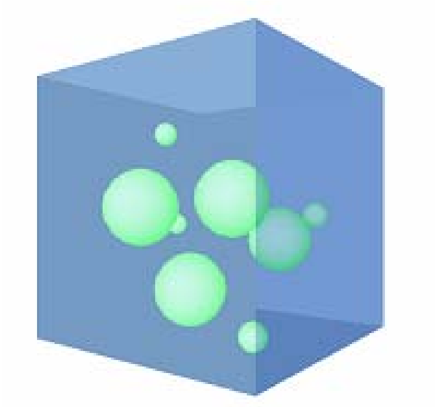


Figure 4: Sample data (Four large spheres, 4 small spheres, and 10 noise points)

A sphere generation program is executed for the 10 sets of experiments. For each experiment, the radius of large spheres is decreased, while the radius of small spheres is kept the same size. For each experiment, samples of 500 voxel cubes (64x64x64) are generated which contain a random number of spheres. Multiple regression analysis is conducted to determine the functions to predict the number of large spheres and small spheres contained in the voxel cubes. When the regression functions are found, R-squared and adjust R-squared values are examined. Both R-squared and adjusted R-squared values can be used to determine if the regression functions are statistically significant. Table 2 shows (1) R-squared values and (2) adjusted R-squared values for each experiment. When the difference between large sphere radiuses and small spheres radiuses are significant, the regression functions are able to predict the number of spheres correctly. This is because the regression functions are customized for recognizing large spheres based on HLAC features, and it is difficult to distinguish large spheres and small spheres if their radius sizes are similar.

Table 3 shows the actual number of spheres and the predicted number of spheres by the system. Since the regression functions predict the number of spheres from HLAC features, the functions return the numbers with floating-point numbers. Floating-point numbers are rounded to integer numbers to determine if the predicted number of spheres matches the actual number of spheres. As shown in the table, the system was able to predict the correct number of spheres fairly well. One important point is that the system does not have any recognition programs specialized for sphere shapes, but the system recognizes sphere shapes by learning from sample data. Also, the system counts the number of spheres without extracting portions of spheres from voxel cubes. The shift invariant features based on HLAC make possible this simple and fast recognition of sphere patterns.

	RadiusL	*RadiusS*	R^2	$Adj.R^2$
Exp. 1	10	5	0.937	0.937
Exp. 2	9	5	0.946	0.894
Exp. 3	8	5	0.831	0.829
Exp. 4	7	5	0.748	0.746
Exp. 5	6	5	0.650	0.647
Exp. 6	5	5	0.559	0.555
Exp. 7	4	5	0.506	0.502
Exp. 8	3	5	0.555	0.552
Exp. 9	2	5	0.656	0.653
Exp.10	1	5	0.753	0.751

Table 2: R-squared values and adjusted R-squared values for regression functions

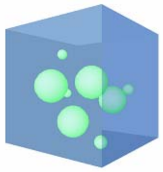
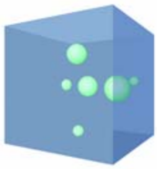
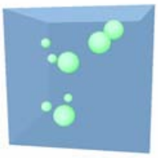
	# of spheres	Predicted # of spheres
	4	4 (4.233)
	4	4 (3.998)
	3	3 (3.238)
	4	4 (4.401)

Table 3: The number of spheres and the predicted number of spheres by the system

Figure 5 shows the four 3D HLAC masks patterns which were influential among the 251 3D HLAC masks patterns in terms of classification powers. These four masks have characteristics to describe the curvatures, moments, and directions features of spheres. Therefore, these four masks are weighted with high slope values for the regression functions.

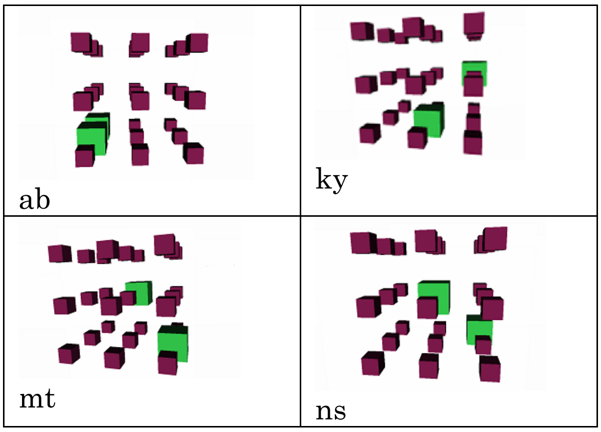


Figure 5: Four 3D HLAC masks which were influential to recognize the spheres shapes in the experiment.

3.3 Pattern Classifications

In this experiment, two different patterns, spheres and cylinders, are tested as shown in Figure 6. The conditions for the experiment are similar to the experiment described in a previous subsection. Five hundred voxel cubes are generated as sample data where a random number of spheres and cylinders are placed into the cube. The radius of spheres and the radius of cylinders are the same size, and the heights of the cylinders and the diameter of the spheres are set to the same size as shown in Figure 6. After the system learns the pattern using multiple regression analysis, 50 new voxel cubes are generated and tested to determine whether the system can correctly count the number of spheres and cylinders. Table 4 shows the pattern recognition rates for the spheres and cylinders with three different resolutions. The best resolution for the system depends on the size and type of target patterns. The 64 x 64 x 64 resolution returned the best result for sphere and cylinder recognition experiments. The system recognized two different patterns successfully with significant recognition rates as shown in Table 4.

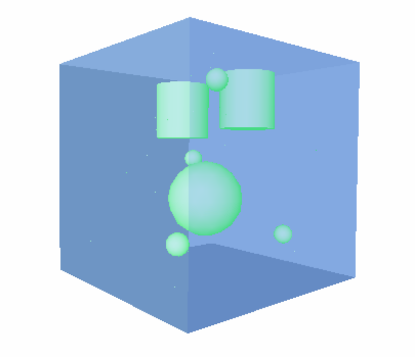


Figure 6: Pattern recognition experiments for spheres and cylinder

	64^3	32^3	16^3
Spheres	95.0 %	95.0 %	90.0 %
Cylinders	98.0 %	97.0 %	89.0 %

Table 4: Pattern recognition rates for the sphere and cylinders

4 Conclusions and Future Work

Recent advancement in computer graphics hardware allows users to visualize various scientific 3D voxel data used in the field of education, business and entertainment. Although wide ranges of 2D image pattern recognition techniques have been introduced, 3D pattern recognition technique research has not been studied extensively. This paper describes a pattern recognition technique based on 3D HLAC (Higher order Local Autocorrelation) masks for voxel data. We have implemented an experimental recognition system and a preliminary pattern recognition test shows that the system successfully could classify 3D voxel patterns. The system makes possible the detection of particular shapes or patterns in voxel data. Several interesting points can be concluded which consist of the following: (1) the recognition system can learn voxel patterns from sample voxel data by using 3D HLAC masks and statistical analysis. (2) The system does not require any extra program for classifying patterns even if the target patterns to be recognized are changed. This shows that the system can be an adaptive pattern recognizer for various patterns. (3) The system can detect patterns regardless of the pattern positions since the HLAC features are shift-invariant. (4) The computation time for the recognition is fixed regardless of whether voxel data patterns are simple or complex, because HLAC features are computed based on mask scanning of target patterns.

In our experiment, simple patterns including spheres and cylinders are tested. More complex patterns including 3D solid textures and voxelized 3D polygonal modes will be tested as a future work. Since the recognition rates of patterns vary depending on the resolution sizes of voxels, the system will be extended to flexibly select the best resolution to recognize voxel patterns.

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.

[07/2004i]