

## Local Color Voxel and Spatial Pattern for 3D Textured Recognition

Hero Yudo Martono<sup>#</sup>

<sup>#</sup>*Department of Informatics and Computer Engineering, Electronics Engineering Polytechnic Institute of Surabaya,  
On Raya-ITS street , Surabaya, 60111, Indonesia  
E-mail: hero@pens.ac.id*

---

**Abstract**— 3D textured retrieval including shape, color dan pattern is still a challenging research. Some approaches are proposed, but voxel-based approach has not much been made yet, where by using this approach, it still keeps both geometry and texture information. It also maps all 3D models into the same dimension. Based on this fact, a novel voxel pattern based is proposed by considering local pattern on a voxel called local color voxel pattern (LCVP). Voxels textured is observed by considering voxel to its neighbors. LCVP is computed around each voxel to its neighbors. LCVP value will indicate uniq pattern on each 3D models. LCVP also quantizes color on each voxel to generate a specific pattern. Shift and reflection circular also will be done. In an additional way, inspired by promising recent results from image processing, this paper also implement spatial pattern which utilizing Weber, Oriented Gradient to extract global spatial descriptor. Finally, a combination of local spectra and spatial and established global features approach called multi fourier descriptor are proposed. For optimal retrieval, the rank combination is performed between local and global approaches. Experiments were performed by using dataset SHREC'13 and SHREC'14 and showed that the proposed method could outperform some performances to state-of-the-art.

**Keywords**— 3D texture; voxel pattern; Fourier descriptor; spatial pattern

---

### I. INTRODUCTION

Currently 3D texture retrieval is still an interested research because the increasing number of 3D models in a variety of shapes, colors and patterns. This causes the main focus retrieval is not only based on the shape, but also color and pattern. Previously there are some basic methods to solve 3D shape retrieval such as view-based, transform-based, graph-based which are combined with a color histogram for 3D textured retrieval. 3D textured retrieval based on pattern voxel, which has never been done yet, while generating a 3D voxel grid keep the information and at the same time is able to equalize the dimensions of various 3D models. This hypothesis appeared by observation of a 3D color voxel visually which is similar to the original one. Research related to generate voxel was already proposed in [1]. Actually some analysis of local pattern based on voxel were already exist but only used binary voxel to capture global shape not included color in such as in [2]-[5], 3D LBP [6], Lightweight binary voxel [7], MFSD [8]. Another voxel-based approach to recover full 3D shape from view based 2.5D and to recognize multiple objects from a single image are in [9], [10]. For the different reason and different use, we study about local extraction on a 3D color voxel to generate local features. In general, the proposed framework is illustrated in Fig. 1 The main contributions in this paper are as follows: (1) Local Color Voxel Pattern (LCVP). First, this

method performs 3D grid color voxel rendering by utilizing all vertices then mapping them to voxel in the related position. It is possible that a voxel will be crossed by some faces so that it obtains the color voxel by averaging all the color vertices which are crossing it. The color voxel is interpolated by three vertices on faces which are crossing it. After generating a 3D grid color voxel, the next process is analyzing local pattern of a color voxel one by one inside a cube with 3x3x3 size which covers it. A cube has 6 sides. A side will be called surface if there is at least one composition voxel equal to a non-zero. LCVP formula is calculated only on the surface of cube. LCVP has two types, first type will utilize corner voxels, and second type will utilize middle voxel. The voxel value is not used in LCVP formula directly, but it first must be converted using level color conversion equation. This will ensure that a different surface color will generate a different pattern. To produce a rotation and reflection invariance form, it performs circular shift and reflection circular shift in voxel composition sequence then choose the minimal value as LCVP value. A LCVP histogram is then built. When computing LCVP, it also builds color histogram using color on each voxel center cube. (2) Rank combination process is a process that combines two ranks from different extraction methods, in this case they are LCVP (our method), weber histogram, oriented gradient and multi Fourier descriptor method. Each rank is built by using the same query model. If there is a member in the first rank

and it is also member of second rank then it was called slice member. A sequence change will occur in the first rank. Slice member becomes the highest priority as member with the most similar to query and change the sequence in first rank. This process will be employed on each query. Using this approach, this can improve the performance.

This paper is organized as follows. Section I reviews a related work. Section II introduces the methods such as building 3D color voxel, local voxel pattern (LCVP), multi Fourier spectral descriptor (MFSD) and weber method. Section II also introduces the optimization approach using a rank combination. Section III shows some experiments, results, and comparison with the state-of-the-art. Section IV provides some conclusions.

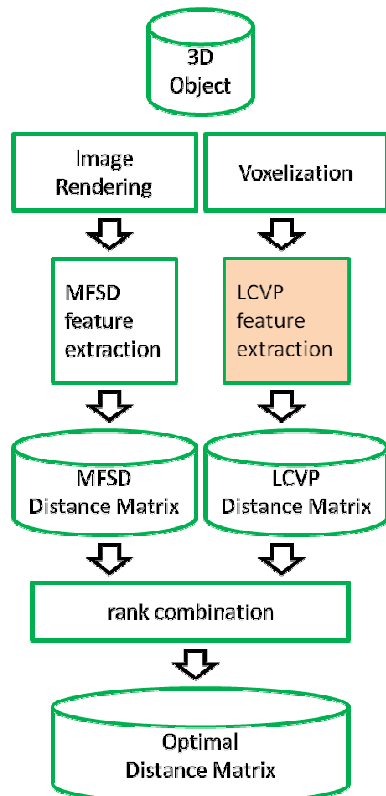


Fig. 1 Framework of proposed method for 3D textured retrieval

In the last decade, the development of 3D shape retrieval is very fast. Due to some areas such as multimedia, computer graphic, computer vision, CAD use 3D models and require a reliable 3D shape retrieval. The development of algorithm began by paying attention to the characteristics of rigid 3D shape. Based on the representation of the shape descriptor in [11], the shape matching is divided into three categories: (1) feature based methods, (2) graph-based methods and (3) geometry-based methods. Some research are in feature-based methods such as [12]-[19]. The concept of global feature-based similarity has been refined recently by comparing the distribution of global features instead of the global features directly such as [20]-[21]. Spatial maps are a representation that captures the spatial location of an object. Some researchers in this domain are [22]-[25]. Local feature based methods provide various approaches to take into account the surface shape in the neighborhood of points the boundary of the shape such as [26]-[29]. Several methods

also appear in image problem such as [30]-[36]. Graph-based methods can be divided into three broad categories according to the type of graph use: model graphs [37]-[38], Reeb Graphs [39]-[40], and skeletons [41]-[42]. Geometry based method is classified into four approaches such as view based similarity in [43], volumetric error based similarity in [44]-[45], weighted point set based similarity in and deformation based similarity in [46]-[47] and deformation based similarity in [48]-[49]. Some of these algorithms are Shape Distribution, Spherical Harmonic Descriptor, Light Field Descriptor [50], Elevator Descriptor (ED) [51] and Shape Impact Descriptor [52].

The next consideration moved to the characteristic of 3D shape non-rigid where in general some algorithms consider about local features, topological structures, isometry invariant global geometric properties, direct shape matching or canonical forms. An algorithm which considers local features and also insensitive to isometric transformation are Spin Images [53], Heat Kernel Signatures [54], salient local features (SIFTs) [55], Intrinsic Spin Images [56]. Others algorithm which considers topological structures are Multiresolution Reeb Graphs (MRGs) and skeletons matching technic. Algorithms which notice isometry invariant global geometric properties are employing Laplace-Beltrami spectra [57], eigenvalue from geodesic distance matrix [58], distribution of intrinsic distance including diffusion distance, geodesic distance, a curvature weighted distance [59] and employing canonical form [60].

Rapid development on 3D textured retrieval was triggered by algorithm contest such as SHREC'13 and SHREC'14. There are several algorithms in [61]. M. Abdelrahman et al described a 3D shape textured method by combining a geometric using scale invariant heated kernel signature (SI-HKS) and a photometric contribution. V Garro and A Giachetti used Histogram Area Projection (MAPT) [62]. We also participated in that contest by implementing combination methods such as local binary pattern, local ternary pattern, a histogram of oriented gradient and weber local descriptor. C. Li, A. Godil, A. Ben Hamza used the spectral geometry based framework for textured 3D shape representation and retrieval. This framework is based on the eigendecomposition of the Laplace-Beltrami operator (LBO), which provided a rich set of eigenbases that were invariant to isometric transformation. It consists of two main stages: (1) feature extraction by using spectral graph wavelet signature to capture geometry information and color histogram for texture information, (2) spatial sensitive shape comparison via intrinsic spatial pyramid matching. A.Tatsuma, M.Aono propose Multi Fourier Spectral Descriptor and Multiresolution Representation Local Binary Pattern Histogram (MRLBPH) which captured texture features of rendered image from a 3D model by analyzing multi-resolution representation using LBP. They first enclosed the 3D model within a unit geodesic sphere after normalizing the 3D via Point SVD. They got some number color buffer images rendered from 38 viewpoints. To obtain multiresolution representations, they applied a Gaussian filter with varying scale parameters to an image. S. Velasco-Forero proposed a method that basically computed two features: a shape and a color descriptor. A shape was represented by a geodesic distance matrix (GDM) [63], and a

color was represented by a CIElab color histogram. The basic idea was to compute the average Earth Mover Distance (EMD) distance between RGB histogram for two given shapes. C.-X. Xu, and Y.-J Liu. They sampled the 3D model on its surface in N-dimensional space, which includes both geometric and textural information then these sampling points are optimally clustered. Geodesic distance is computed among the points then it got shape distribution of the model. D. Girgi proposed a Textured Shape Distribution (TSD) descriptor that was a color-aware variant on classical Shape Distribution. TSD consists of the distribution of mutual distances computed between points sampled over the surface mesh representing the 3D models. TSD descriptor employs geodesic distance instead of euclidean distance, and that geodesic distance is computed on the surface embedded in the three-dimensional color space. The vertices of the surface mesh are the (L,a,b) coordinates in the CIElab color space. The conclusion of the most common approach was to combine features of shape and texture.

## II. MATERIAL AND METHOD

Starting from an early investigation that a 3D voxel still keeps its information about shape, color, and pattern, this study tries to search a good formula for texture and shape feature extraction. Actually, there are some studies related to 3D voxel, but they used binary voxel, not color voxel. This is a different point. We learn it and add new features to perform 3D grid color voxel. Completing this voxel building, we propose a novel formulation to extract local feature based on color voxel against all neighbours in a cube. To add experiment, we also implement Fourier extraction from the spatial image for capturing the global shape of 3D models.

### A. 3D Voxel

A 3D model consists of many faces (triangles), and a face consists of three vertices. Each vertex has an RGB value. Color in the face is influenced by interpolation of three color vertices. So that, in 3D grid color voxel rendering, we will compute color from each vertex. We have to know that a face can cross into one or more voxels. So we have to consider each voxel which crossed the face. The illustration is shown in Fig. 2 (a). Illustration of the calculation of face (triangle) in three dimensions is shown in Fig. 2 (b). For details of the formulation will be explained later. Vice versa a voxel can be crossed by one or more faces. If a voxel is crossed by more than one face, then it will be converted into voxel octree voxels. This illustration is shown in Fig. 2 (c).

The calculation for 3D color voxel rendering as follows: (1) find mean point on x,y,z axis; (2) calculate distance from each point to mean point; (3) find the longest distance between one point into mean point; (4) map each point into voxel; (4) update voxel with color information; (5) normalize color voxel by number points crossed it. In this experiment we use dimension as 64, it means we map 3D object into voxel size 64x64x64. By deciding to map all 3D models into one standard size, it will omit the size differences between 3D models.

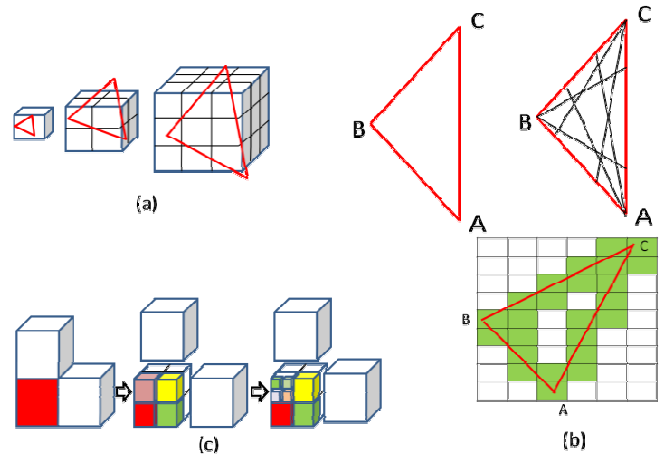


Fig. 2 Creating color voxel from 3D models

### B. Local Color Voxel Pattern

The local pattern on the 3D surface is a fundamental property of the 3D model. By capturing local pattern on the 3D surface means that it gets 3D texture information entirely. In another side, to simplify computation which consists variety vertices and faces then voxel-based approach will be chosen.

In this paper, a new approach will be proposed theoretically and computationally to discriminate 3D texture effectively. The texture on the 3D model is influenced by pattern and color. Members of the same texture class are influenced by pattern, color, and shape. The same pattern could appear on different 3D models and vice versa. Therefore the important thing for the first step is capturing pattern and distribution of color information. To distinguish color, a color histogram as the old approach will be alternative one and used to distinguish the pattern. Local color voxel pattern histogram will be built too. This work is started by constructing cube voxel by size 3x3x3 which has one center voxel. Every voxel will be computed as center cube if it meets some requirements. As an example in Fig. 3, how to create cube voxel.

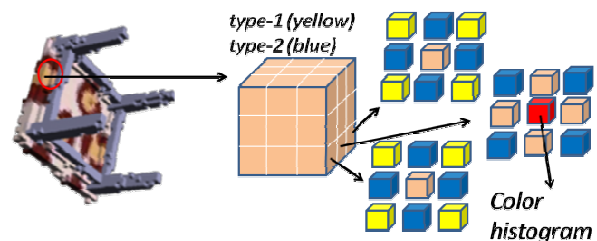


Fig. 3 Creating cube voxel with red voxel as cube center, yellow voxel for LCVP type 1 and blue voxel for LCVP type 2

Fig. 3 shows about how to generate cube voxel. Every voxel has voxel neighbours with color has potentially created cube voxel. Then by splitting this cube, it gets three squares 3x3 voxel. This approach has two different ways. To make different way is clear, each voxel will be colored with a different color such as blue and yellow. Then yellow voxel dan blue voxel will be employed by local color voxel pattern to capture cube texture. The yellow one will be

computed by local color voxel pattern type 1, and the blue one will be computed by local color voxel pattern type 2.

By this voxel (red/yellow), we first determine color scale. Dim (dimension) is voxel dimension, and scale is color scale. For example, we will build a 3D object on voxel which has 64x64x64 dimension, so we provide value for dim is 64. Scale is a number that is used to divide color, for example, the scale is 8, so color (0-255) will have 8 scale level. "dist" is distance, this represents distance a voxel from the center. The final equation as follows :

$$dist^2 = \left( \left( v_x - \frac{dim}{2} \right)^2 + \left( v_x - \frac{dim}{2} \right)^2 + \left( v_x - \frac{dim}{2} \right)^2 \right) \quad (1)$$

$$level = \frac{2 \times dist}{dim} \times scale \quad (2)$$

$$color = \frac{N \times voxel\_center}{255} \times scale \quad (3)$$

In Fig. 3, we take one voxel become a cube center and its surrounding voxels as cube composer. By splitting this cube, we get three squares of 3x3 voxels on each. Yellow voxel and the blue voxel is employed by LCVP equation to capture cube texture. The yellow voxel generates LCVP type 1 while blue voxel generates LCVP type 2. We then implement LCVP operator which is denoted as  $LCVP_{(P,R)}$ , P parameter set quantization angle. LCVP histogram is a texture feature of 3D models. We start to derive color scale and to get rotation reflection invariant by defining texture T on each cube voxel surface. On each cube voxel surface, there are two types of texture/ type of pattern provided by symbols such as  $T_1$  and  $T_2$ .  $V_c$  stands for voxel corner and  $V_m$  for voxel middle. The pattern  $T_1$  is focusing only on voxel corner (the yellow one) while the pattern  $T_2$  is focusing only on voxel middle ( the blue one). Every pattern has four value as follows:

$$T_1 = t(v_{c0}, v_{c1}, \dots, v_{cP-1}) \quad (4)$$

$$T_2 = t(v_{m0}, v_{m1}, \dots, v_{mP-1}) \quad (5)$$

The local pattern on 3D voxel surface is influenced by color (0-255) and its composition. So the color scale will be used than original color for computation's efficiency. N Scale value will be set empirically and be used to compute local color voxel pattern. Each cube consists of six sides (surfaces) where each surface has 9 voxels (3x3). Employing all voxels in one pattern will obtain very wide pattern and become ineffective for computation. That is the reason why it will be split into two types. Type one employs corner voxels while type two employs middle voxels. This splitting will not reduce accuracy even will make it faster and more effective. Each color on voxel is determined as follows :

$$v_{cp} = \frac{n \cdot voxel_{cp}}{255} \quad (6)$$

$$v_{mp} = \frac{n \cdot voxel_{mp}}{255} \quad (7)$$

$$LCVP_1 = \sum_{p=0}^{P-1} v_{cp} \cdot n^p \quad (8)$$

$$LCVP_2 = \sum_{p=0}^{P-1} v_{mp} \cdot n^p \quad (9)$$

Where the color value  $v_{cp}$   $p = 0, \dots, P-1$  is correlated with the color value scale of voxel corner to p(), while the color scale  $v_{mp}$  is correlated with the value of the middle corner to p. We transform color voxel against a scale to get color scale voxel, by defining n factor, then get a unique number that characterizes the spectral texture of the cube. Fig. 4 show how to get LCVP in rotation reflection invariant value, we choose minimum value between patterns  $(P_1, P_2, P_3, P_4)$  to get rotation invariant and choose minimum value between patterns  $(P_5, P_6, P_7, P_8)$  to get reflection invariant value. For final histogram, we add two histograms on each correlated bin. Final formula to get this as follows:

$$LCVP_{rot} = \min_{RO} R_{rotation} (LCVP_{(P_1, P_2, P_3, P_4)}) \quad (10)$$

$$LCVP_{ref} = \min_{RO} R_{reflection} (LCVP_{(P_5, P_6, P_7, P_8)}) \quad (11)$$

$$LCVP = \min(LCVP_{rot}, LCVP_{ref}) \quad (12)$$

$$hist[LCVP] = hist[LCVP_1] + hist[LCVP_2] \quad (13)$$

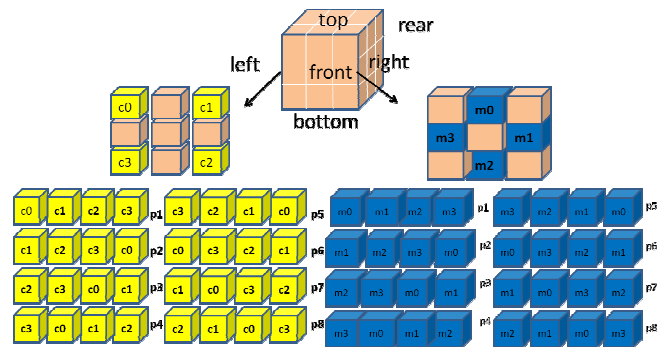


Fig. 4 Obtaining rotation and reflection invariance form by calculating minimum value from eight different patterns

Where color value  $v_{cp}$   $p=0,1,\dots, P-1$  is correlated with scale value color of corner voxel, while color scale  $v_{mp}$  is correlated with a scale value of the middle corner. Symbol  $\min_{RO} R_{rotation}$  (minimal Rotation) indicates a minimal value which is chosen as the minimal value after rotation process, similarly symbol  $\min_{RO} R_{reflection}$  indicates minimal value after reflection process.

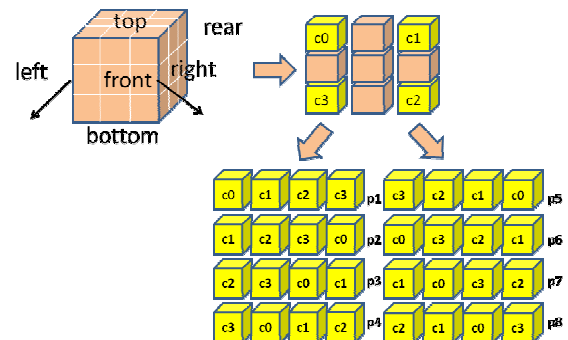


Fig. 5 Local Voxel Pattern Type 1

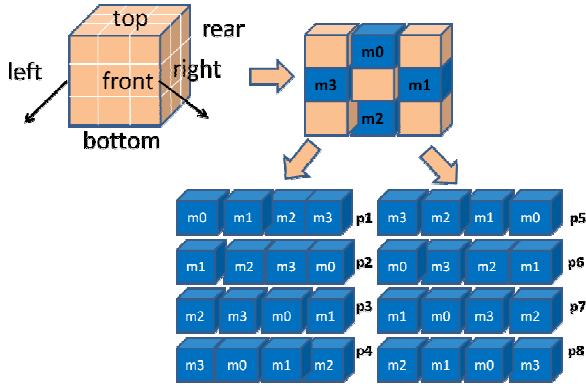


Fig. 6 Local voxel pattern type 2

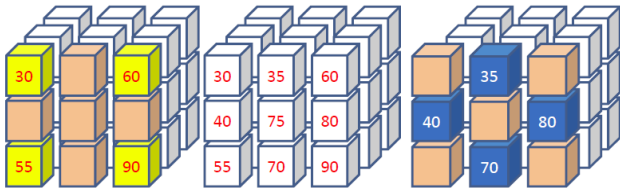


Fig. 7 Example how to calculate LCVP from one side cube. In the middle is original one, left side is LCVP type 1, and right side is LCVP type 2

For better understanding, we give an example shown in Fig. 7. First, we get an original cube (white voxel) in the middle. We then split into type one (left side) and type two (right side). The next process is computing each voxel scale such as  $v_{c0}, \dots, v_{c3}$  and  $v_{m0}, \dots, v_{m3}$

$$v_{c0} = \frac{10 \times 30}{255} = 1 \quad v_{m0} = \frac{10 \times 35}{255} = 1$$

$$v_{c1} = \frac{10 \times 60}{255} = 2 \quad v_{m1} = \frac{10 \times 80}{255} = 1$$

$$v_{c2} = \frac{10 \times 90}{255} = 3 \quad v_{m2} = \frac{10 \times 70}{255} = 1$$

$$v_{c3} = \frac{10 \times 55}{255} = 1 \quad v_{m3} = \frac{10 \times 40}{255} = 1$$

We then compute local color voxel pattern both type one and type two. Each type has eight patterns and chooses the minimal one. Finally, we add each result to build LCVP histogram. The formula as follows :

$$LCVP1_{p1} = 1 \times 10^0 + 2 \times 10^1 + 3 \times 10^2 + 2 \times 10^3 = 2,321$$

$$LCVP1_{p2} = 2 \times 10^0 + 3 \times 10^1 + 2 \times 10^2 + 1 \times 10^3 = 1,231$$

$$LCVP1_{p3} = 3 \times 10^0 + 2 \times 10^1 + 1 \times 10^2 + 2 \times 10^3 = 2,123$$

$$LCVP2_{p1} = 1 \times 10^1 + 3 \times 10^1 + 2 \times 10^2 + 1 \times 10^3 = 1,231$$

$$LCVP2_{p2} = 3 \times 10^0 + 2 \times 10^1 + 1 \times 10^2 + 1 \times 10^3 = 1,123$$

$$LCVP2_{p3} = 2 \times 10^0 + 3 \times 10^1 + 1 \times 10^2 + 1 \times 10^3 = 1,132$$

$$LCVP_1 = \min(LCVP1_{(p1,p2,p3)}) = 1,232$$

$$LCVP_2 = \min(LCVP2_{(p1,p2,p3)}) = 1,123$$

$$Histogram[LCVP_1] ++$$

$$Histogram[LCVP_2] ++$$

Because every pattern has four members (m1,m2, m3, m4), so LCVP will also have four value. We have to choose a minimal value on every pattern. LCVP<sub>1</sub> is LCVP which focus on corner voxel while LCVP<sub>2</sub> is LCVP which focus on the middle corner. The two histograms built are LCVP<sub>1</sub> and LCVP<sub>2</sub>. Every time we calculate LCVP on each voxel, we will add grade on histogram LCVP<sub>1</sub> and histogram LCVP<sub>2</sub>.

### C. Multi Fourier Spectral Descriptor

For shape features, a method called Multi Fourier Spectral Descriptor (MFSD) in [8] was chosen. It uses the low-level Fourier spectra concept to build shape features. This process is started by reading 3D model then performing pose normalization. This process is absolutely needed because every 3D models have different size and orientation while the solution is based on image processing. Pose normalization is definitely very important. One kind of pose normalization method is based on point SVD. 2D image rendering is implemented after pose normalization and generating three kinds of images such as silhouette image, contour image, and depth buffer image. This also implements Periphery Enhanced Image (PEI) to enhance the peripheral shape of all images. All three images are converted into polar coordinate first before performing Fourier transform. It then gets low-frequency spectra from each image transformation. The combination of four different Fourier spectra is called multi Fourier spectra descriptor. These features are used in this experiment. This process can be seen in Fig. 7.

### D. Weber Local Descriptor and Histogram Oriented Gradient

The next method is using weber method. As explained in [35], it has two components: differential excitation ( $\zeta$ ) and orientation ( $\theta$ ). How to build weber histogram is illustrated in Fig. 8 and Fig. 9. The first step, we read 2D image in one viewpoint. Gray image is chosen to generate one channel rather than color image because global shape information does not need a color image. In every single pixel, we will compute excitation value and orientation value. In excitation, there are two filters while in orientation gradient will be generated. The first filter in excitation calculates the differences between its neighbours and the center point as follows:

$$\zeta = \arctan\left(\frac{Vs_{11}}{Vs_{10}}\right) \quad (14)$$

$$\theta = \arctan\left(\frac{Vs_{11}}{Vs_{10}}\right) \quad (15)$$

where Vs<sub>11</sub> and Vs<sub>10</sub> are the output of the filter in Fig. 9. In some previous papers, the idea of depicting an image by histogram has been used in the biologically plausible vision system. Motivated by this idea, we tried to compute excitation and orientation on every pixel against its neighbors.

We also implement Histogram of Oriented Gradient in (HoG) [36] as one of features extraction method that can be

implemented for capturing shape. This function meets our need. By using this method, we can receive shape features. The HoG representation has some advantages which can captures edge or gradient structure that is characteristic of local shape. This method is ever implemented to detect human. In practice on this paper, we divide image window into small spatial regions (“cells”), for each cell accumulating a local 1-D histogram of gradient directions or edge orientation over the pixels of the cell.

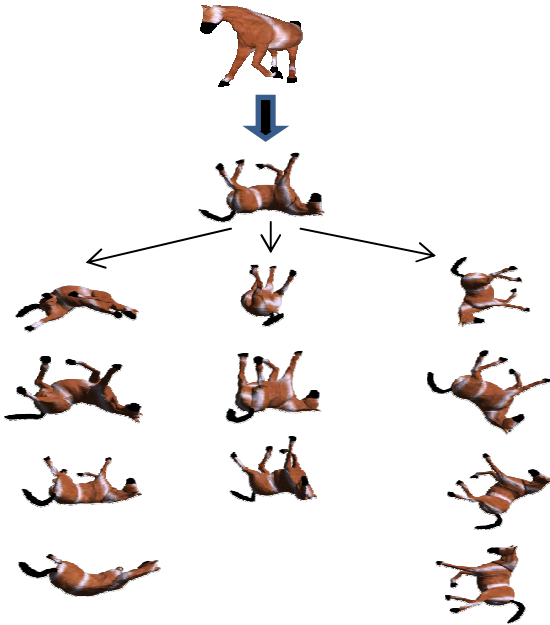
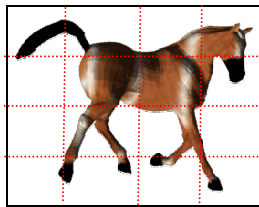


Fig. 8 Different point of view 3D models will generate different contour image

$$V_{s\_00} = \sum_{i=0}^{P-1} (X_i - X_c) \quad (16)$$



1	1	1
1	-8	1
1	1	1

Vs\_00

1		-1

Vs\_01

1	1	1
1	-8	1
1	1	1

Vs\_10

1		-1

Vs\_11

Fig. 9 Mapping image for weber and oriented histogram

Fig. 9 describes how to calculate oriented gradient on each image. We divide one image into some cells. Before computing gradient, a color image is converted into the grayscale to focus only one model one color channel. The equation of oriented gradient is similar with filtering. There

are two experiments conducted on this method. The first used image in grayscale and second used image after the canny filter. The goal is computing only on its periphery.

### E. Color Image Histogram

3D models not only come with shape features but also come with texture features which represented by a different color on each model. To capture different texture, we also can utilize color composition as texture features. We use a simple color histogram which will quantize color based on distance from center pixel and range intensity. The following Table 1 describes our color quantization.

TABLE I  
Color Quantization

Quantitation level	Color	Distance
1	0-25	0-3
2	26-50	4-7
3	51-75	8-11
4	76-100	12-15
5	101-125	16-19
6	126-150	20-23
7	151-175	24-27
8	176-200	28-31
9	201-225	-
10	226-255	-

For final descriptors, we combine all histograms to be one package. The result of our shape and texture descriptor formula as follows :

$$d = w_{(d)}d_{(d)} + w_{(s)}d_{(s)} + w_{(c)}d_{(c)} + w_{(v)}d_{(v)} \quad (17)$$

each part represents a feature of weber histogram, histogram of oriented gradients, spatial Fourier transform histogram and color histogram. This is a different approach from LCVP and MFSD. We will compare it with another approach.

### F. Building Dissimilarity Matrix

Dissimilarity Matrix (DM) is built by comparing between all 3D models features. The obtained value represents the similarity between each 3D model. The smaller value is the more similar ones. For the first experiment, there are three features such as local color voxel pattern (LCVP), color histogram (CH) and multi Fourier spectral descriptor (MFSD). For the intentionally, this study will combine two DM both LVP and CH by normalizing first but not for MFSD. We will perform rank combination between normalizing (LCVP and CH) and MFSD next. For generating DM, we use Manhattan distance as follows:

$$d_{LCVP}(x_i, x_j) = \sum_{a=1}^n |x_{ia} - x_{ja}| \quad (18)$$

$$d_{weber}(x_i, x_j) = \sum_{a=1}^n |x_{ia} - x_{ja}| \quad (19)$$

$$d_{MFSD}(x_i, x_j) = \sum_{a=1}^n |x_{ia} - x_{ja}| \quad (20)$$

$$d_{normal}(x_i, x_j) = \frac{d_{LCVP}(x_i, x_j)}{\max(d(i, 0, \dots, M-1))} w_a + \frac{d_{weber}(x_i, x_j)}{\max(d(i, 0, \dots, M-1))}$$

(21)

Symbol  $d(xi, xj)$  (distance between object  $i$  and object  $j$ ) indicates the difference between object  $i$ -th and object  $j$ -th. If they are similar, then their distance is closed to zero otherwise closed to one. Every object will have  $n$  features. So we compare all features between each 3D objects. From those notations, we can conclude that we have three distance similarity between object based on LCVP, weber, and MFSD. For first calculation, we build dissimilarity matrix based on LCVP and weber. After that, we get the normal value between them. Finally, we get two dissimilarity matrix (DM), one is from LCVP and weber and secondly is from MFSD.

### G. Rank Combination

We choose the first optimization by using a rank combination approach. Many rank combination approach can be cast as the problem of assigning scores to classes based on the ranks they receive from multiple constituent classifiers. Once assigned, then classes can be ordered based on their scores, generating a combined ranking. We use the following notation. There are  $K$  target classes  $t^1 \dots t^K$  and a collection of  $J$  component classifier algorithms  $a^1 \dots a^J$ . For any particular query, the output of algorithm  $a^k$  is  $r^{(j)} = (r^{(j)}(1), \dots, r^{(j)}(K))$ , with  $r^{(j)}(K)$  being the rank assigned by algorithm  $a^k$  to class  $t^j$ . A score function, for each class maps the rankings to a scalar

$$S(\theta) \equiv f_{\theta} (r^{(1)}, \dots, r^{(j)})$$

where  $\theta$  is a class in  $t^1 \dots t^K$ . As score functions are ultimately used to generate new rankings, they have the important property of being invariant to monotonic transformation.

$$(R^*(k) \leq R^*(k') \iff S(k) \geq S(k') \iff g(S(k)) \geq g(S(k'))$$

where  $g$  is monotonically increasing and  $R^*$  denotes the combined ranking derived from the class scores. In [64], they proposed the Mixed Group Ranks (MGR) score function which generalizes the Best Rank and Linear scores. It is a weight linear sum of the minimum rank of all subset of classifiers.

$$S(\theta) = - \sum w_A \min r^{(j)}(\theta) \quad (22)$$

The MGR score function combines the democratic voting aspect of the Linear and Borda Scores with the emphasis on confident rankings of the Best Rank score. In [64], they describe the general category of score function that embodies these characteristics. Score functions that are both monotonic and quasiconvex (in the ranks assigned to a class) prefer lower ranks to bigger ranks and assign greater influence to smaller ranks. With non-negative weights,  $w_A \geq 0$ , MGR is both monotonic and quasiconvex, embodying these score properties.

Based on MGR concept, we implement this approach by comparing two ranks (first is LCVP rank and second is MFSD rank) which is built by employing two different dissimilarity matrix (DM). This process is started by

building two ranks using the same query. Suppose that  $r^{LCVP} = r^{LCVP}(1), \dots, r^{LCVP}(K)$ , with  $r^{LCVP}(k)$  being the rank assigned by algorithm LCVP to class  $t^k$ . We then examine whether there is a same member situated in both ranks. If no one than there is no update in the first rank, otherwise this member will be the highest priority in the first rank. Final result of first rank will update LCVP distance matrix. In Fig. 10, it is shown how a rank be generated by both LCVP method and MFSD method using the same query model. The sequence number of models based on distance is shown on each rank. There are some sliced members, then comparison is performed and finally a final rank established. This result will be used to update distance matrix LCVP. The formula is shown as follows:

$$\begin{aligned} r^{LCVP}(i, 1..K) &\leftarrow DM^{LCVP}(i, 1..M) \\ r^{MFSD}(i, 1..K) &\leftarrow DM^{MFSD}(i, 1..M) \\ r^{MFSD}(k) &= r^{LCVP}(1..K) \\ r^{LCVP}(n) &= r^{MFSD}(n) \\ DM_{LCVP}(i, n) &= 0.01 \times n \end{aligned} \quad (23)$$

where  $r^{LCVP}(i, 1..K)$  is rank result within  $K$  numbers of query  $i$ . This result generated by computing dissimilarity matrix of each model in LCVP DM. While  $r^{LCVP}(i, 1..M)$  is list of dissimilarity matrix between query  $i$  to all models.  $r^{LCVP}(k)$  is  $k$ -th rank result of LCVP. In Fig. 10, we try to present our combination rank. First we build LCVP rank based on LCVP dissimilarity matrix (DM) using a query. At the same time, we also build MFSD rank based on MFSD DM using the same query. By looking into MFSD rank result one by one until end, we try to find the same answer from LCVP rank from the first until the end. Suppose we find the one similar answer, we then shift into first order for final answer. By using this final order, we recompute LCVP DM. We also try to capture our algorithm for combining rank.

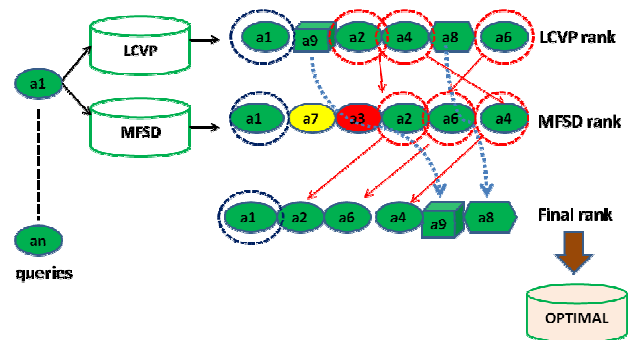


Fig. 10 Optimization by combining two ranks from two different type of features.

### III. RESULT AND DISCUSSION

In this section, some experiments will be performed. To show the performance of this system, it used a standard dataset from Shape Retrieval Contest (SHREC'13) [65] and SHREC'14 [66]. Dataset of SHREC'13 has 240 models of 3D objects grouped in 10 geometric classes and 33 texture

classes. Dataset of SHREC'14 has 572 models of 3D objects grouped in 16 geometric classes and 48 texture classes. The whole dataset is separated into two-level ground truth: if two models share only shape they will be called relevant, if they share both shape and texture then will be called highly relevant.

The evaluation process has been determined by using several evaluation measures such as Average precision-recall curves, Nearest Neighbor (NN), First Tier (FT), Second Tier (ST) and Average Dynamic Recall (ADR). At the end of experiments, we will compare this proposed to state-of-the-art by using standard measurement and evaluation. This evaluation code is coming from SHREC organizer. It was done to make sure, that all evaluation will be examined fairly.

### A. Experimental Setup

For the experiments testing, we adapt to the rules established by the committee in SHREC 2014 on track Retrieval and Classification on Textured 3D Objects. We also use code generating performance which is provided by the committee. This is done to assure fairly comparison. Detailed information can be found in [12]. The performance of this method has been evaluated according to the following relevance scale. If a retrieved object shares both shape and texture with the query, it is included in a highly relevant group otherwise if it shares the only shape, it is included in the relevant group. The evaluation process has been based on the following evaluation measures such as Average precision-recall curves, Nearest Neighbour (NN), First tier, Second tier (ST) and Average Dynamic Recall (ADR). Average precision-recall curves. Precision is the fraction of retrieved items that are relevant to the query. The recall is the fraction of the items relevant to the query that are successfully retrieved. Being A the set of all the relevant objects and B the set of all the retrieved object. The formula of precision-recall is defined by :

$$Precision = \frac{|A \cap B|}{|B|}, Recall = \frac{|A \cap B|}{|A|} \quad (24)$$

Nearest Neighbour, First tier, and Second tier. These evaluation measures aim at checking the fraction of objects in the query's class also appearing within the top k retrieval. Average dynamic recall. The idea is to measure how many of the items that should have appeared before or at a given position in the result list actually have appeared. The average dynamic recall (ADR) at a given position averages this measure up to that position. Precisely, for a given query let A be the set of highly relevant classified items, and let B be the set of relevant items. The ADR is computed as:

$$ADR = \frac{1}{n} \sum_{i=1}^n \frac{r_i}{i} \quad (25)$$

### B. Relevant

First, we perform global features to captures geometric information by using multi Fourier spectral descriptor (MFSD), it extracts features and generates dissimilarity matrix between each model. Fig. 11 shows partial result using dataset SHREC'13. Left side is a query and the right

one is answers. The first query is human by using our proposed methods; we can retrieve the similar one such other humans with a different pose. Similarly, the second and third query, queries are tree and tree again. We can get answers with similar shapes. In Fig. 12, we also show partial result using dataset SHREC'14. Left sides are queries (pliers and scissors), and right sides are the answers. Even some result is failed, but in approximately, this approach gives a good result.

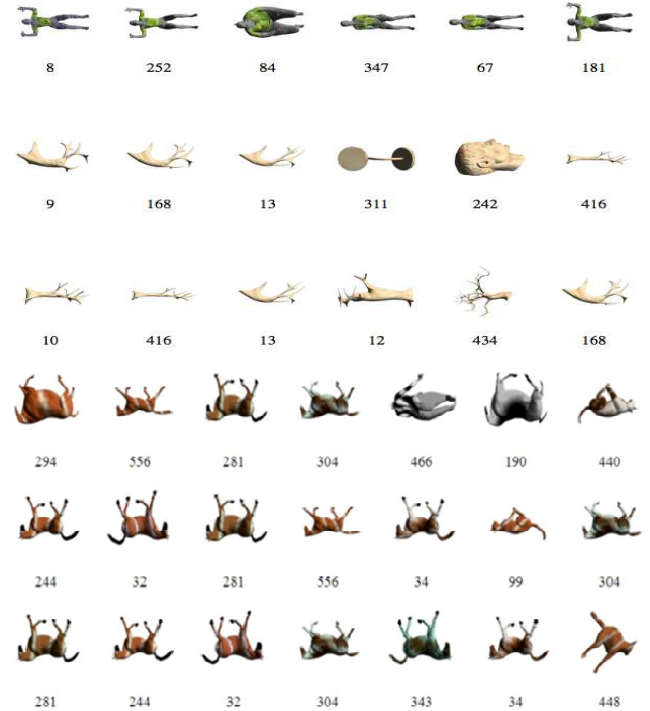


Fig. 11 Example of 3D retrieval system in relevant mode using dataset SHREC'13

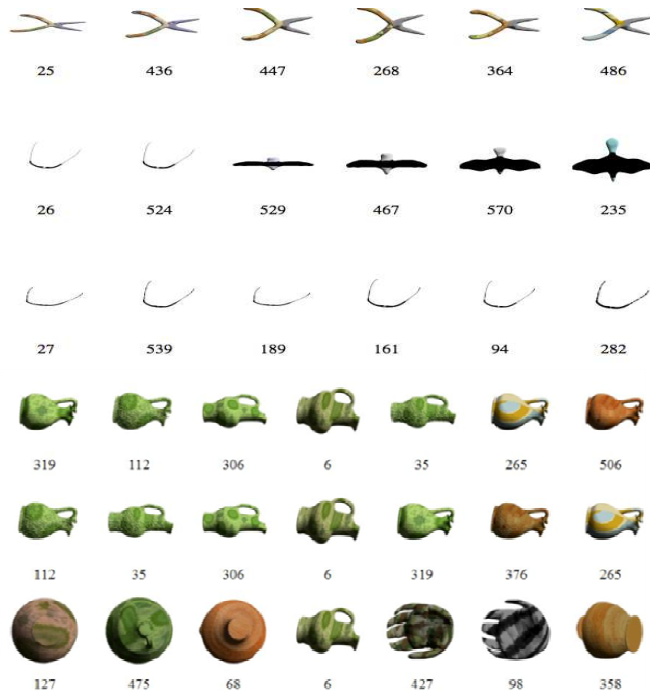




Fig. 12 Example of 3D retrieval system in relevant mode using dataset SHREC'14

### C. High Relevant

Second, we perform our local features called LCVP, then combine with MFSD features using a rank combination. In Fig. 13, we show partial result using dataset SHREC'13. Left side is a query and the right one is answers. The first query is round table, by using our proposed methods we can retrieve the similar one such round table. Similarly, the second and third query, queries are an ant and round table again. Now we can get answers with similar in shapes and texture. In Fig. 14, we also show partial result using dataset SHREC'14. Left sides are queries (study lamp, vase, face sculpture) and right sides are the answers. Once again, we get answers with similar in shapes and texture. We can compare additional performance we get after combine LCVP and MFSD

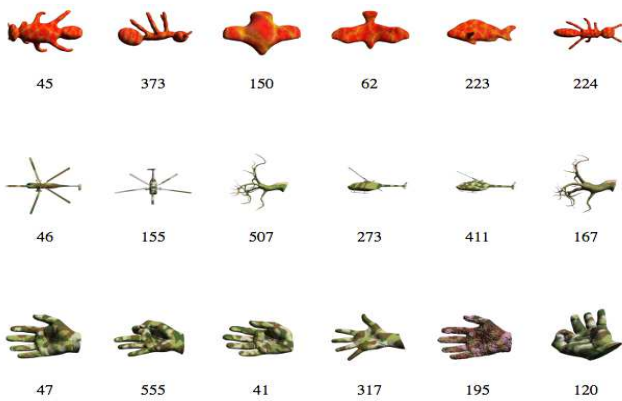


Fig. 13 Example of 3D retrieval system in high relevant mode using dataset SHREC'13

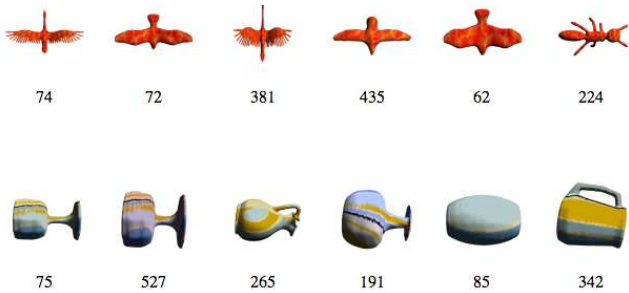


Fig. 14 Example of 3D retrieval system in high relevant mode using dataset SHREC'14

TABLE I  
SHREC'13 RELEVANT

Method	NN	First T	Second T
Proposed	0.996	0.694	0.807
A1	0.963	0.603	0.681
A2	0.958	0.603	0.229
Be	0.083	0.135	0.229
Zh	0.342	0.238	0.353
Gi	0.971	0.574	0.715
G1	<b>1.000</b>	<b>0.708</b>	<b>0.873</b>
G2	0.992	0.575	0.708
G3	0.983	0.632	0.801

V1	0.871	0.422	0.583
V2	0.996	0.480	0.606
V3	0.971	0.476	0.634

The result from other participant is obtained from the paper in [65] for SHREC'13 and paper in [66] for SHREC'14. The description of each method also can be found in a paper [65].

TABLE III  
SHREC'13 HIGH RELEVANT

Method	NN	First T	Second T
Proposed	<b>0.981</b>	<b>0.884</b>	<b>0.941</b>
A1	0.515	0.553	0.710
A2	0.508	0.561	0.730
Be	0.019	0.175	0.209
Zh	0.174	0.135	0.214
Gi	0.788	0.658	0.748
G1	0.417	0.526	0.799
G2	0.898	0.733	0.893
G3	0.519	0.579	0.772
V1	0.807	0.511	0.633
V2	0.879	0.764	0.904
V3	0.909	0.733	0.863

As we can see in Table 1 and Table 2 as a result of the experiment when using dataset SHREC'13. Table 1 shows the result in the relevant category. It means that the retrieval system is only focusing on shape. G1 is the best method because it uses geometric calculation using geodesic metric and eigenvalue. This method is best, but it needs time more. We have already tried this method but the computation time needs high specification machine, while our method no needs complex computation. Even though our method is not the best, but the result is still closed to the best Table 2 shows the result on the high relevant category. It means that the retrieval system is focused on shape and texture. The combination of LCVP and MFSD is our proposed method become the best in this category.

In this dataset, it consists of 240 models. The fact is that the 3D model in the same texture class mostly have similar geometry, while the texture is affected by the difference in color so that it can be said that the dataset is not too complex. Our approach is good in this case because there is a geometry element to determine the shape and color element to determine texture between models. Fig. 15 shows the composition of all participants approach in precision recall evaluation .

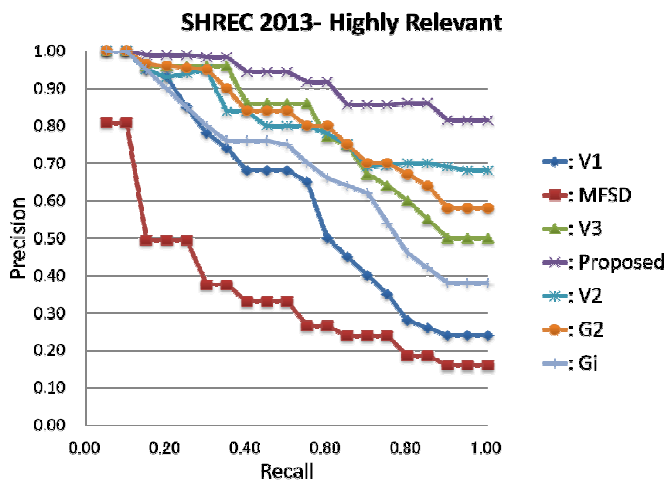


Fig. 15 Comparison result on SHREC 2013 High Relevant

Next experiment is conducted by using SHREC 2014 dataset then comparing with other participants method on that event. For comparison, we refer to a paper in [66], and the result can be seen in Table 3 dan Table 4. Gi method is proposed by D. Giorgi from National Research Council (Italy). This method is called Textured Shape Distribution (TSD). It is a color-aware variant on classical Shape Distribution. TSD consist of the distribution of mutual distances computed between points sampled over the surface mesh representing the 3D models. TSD employs geodesic distance instead of Euclidean, and the geodesic distance is computed on the surface embedded in three-dimensional color space. Ve method is proposed by S. Velasco-Forero (ITWM Fraunhofer Institute, Germany). He also computes the geodesic distance matrix in the mesh information. He uses a spectral representation of the geodesic distance as a descriptor. He also uses color information in RGB space as texture descriptor and finally combines both of them. GG method is proposed by V. Garro and A. Giachetti from the University of Verona (Italy). He computed textured mesh difference based on Histogram of the Multiscale Area Projection Transform (MAPT). This method is based on a spatial map that encodes the likelihood of the points inside the shape of being centers of spherical symmetry. LBG method is proposed by C. Li, A. Godil (NIST, USA) and A. Ben Hanza (Cordia University, Canada). They employ the spectral geometry for textured 3D shape representation and retrieval. This method is based on the eigendecomposition of the Laplace-Beltrami Operator (LBO) which provides a rich set of eigenbases that are invariant to isometric transformation. TA method is proposed by A. Tatsuma, M. Aono and C Sanada (Toyohashi University of Technology). They propose the multi-resolution Representation Local Binary Pattern Histogram (MRLBPH). This method enclosed a 3D model within a unit geodesic sphere after normalizing the 3D model via Point SVD and rendered image as 38 viewpoints. HA is proposed by Hero Yudo and M. Aono (Toyohashi University of Technology). They employ some methods such as LBP and LTP in spatial space. AEF is proposed by M. Abdelrahman, M. El-Melegy and A. Farag from the University of Louisville (USA). He generates shape descriptor by employing scale invariant heat kernel signature. For texture descriptor is based on the color

histogram in RGB space. XL method is proposed by C.-X Xu and Y. J. Liu from the Tsinghua University. They propose a sketch-based method, which belongs to the 3D image manners but applies to some specific cases such as 3D CAD design process.

TABLE III  
SHREC'14 RELEVANT

Method	NN	First T	Second T
Proposed	<b>0.979</b>	<b>0.466</b>	<b>0.582</b>
AEF1	0.732	0.408	0.521
AEF2	0.735	0.408	0.521
GG1	0.711	0.240	0.324
GG2	0.740	0.265	0.365
GG3	0.822	0.344	0.469
HA1	0.721	0.271	0.382
HA2	0.736	0.281	0.387
HA3	0.737	0.277	0.386
LBG2	0.840	0.402	0.553
LBG3	0.804	0.374	0.512
LBG4	0.909	0.430	0.559
TA	0.963	0.436	0.562
Ve1	0.902	0.404	0.512
Ve2	0.918	0.398	0.499
Ve3	0.909	0.372	0.469
XL	0.348	0.175	0.271
Gi1	0.895	0.259	0.374
Gi2	0.906	0.279	0.388
Gi3	0.958	0.383	0.504

TABLE IVV  
SHREC'14 HIGH RELEVANT

Method	NN	First T	Second T
Proposed	0.748	<b>0.462</b>	<b>0.629</b>
Gi2	<b>0.894</b>	0.365	0.448
Gi1	0.890	0.324	0.401
Gi3	0.813	0.455	0.590
Ve1	0.735	0.396	0.540
TA	0.563	0.336	0.456
HA2	0.468	0.278	0.362
HA1	0.424	0.265	0.350
Ve3	0.336	0.275	0.369
XL	0.108	0.149	0.192
GG2	0.722	0.432	0.557
GG1	0.696	0.404	0.530
LBG2	0.676	0.412	0.565
GG3	0.665	0.384	0.504
Ve2	0.593	0.338	0.469
LBG3	0.512	0.306	0.406
HA3	0.462	0.272	0.358
LBG4	0.394	0.325	0.437
AEF2	0.123	0.228	0.351
AEF1	0.098	0.226	0.350

We also can see at Table 3 and Table 4 as a result of experiment when using dataset SHREC'14. This dataset consists of 572 models. In relevant criteria, our approach reaches the best one while in high relevant criteria only on the first tier and second tier our proposed is number one. In Fig. 16, we can see the composition of all participants approach in precision recall evaluation.

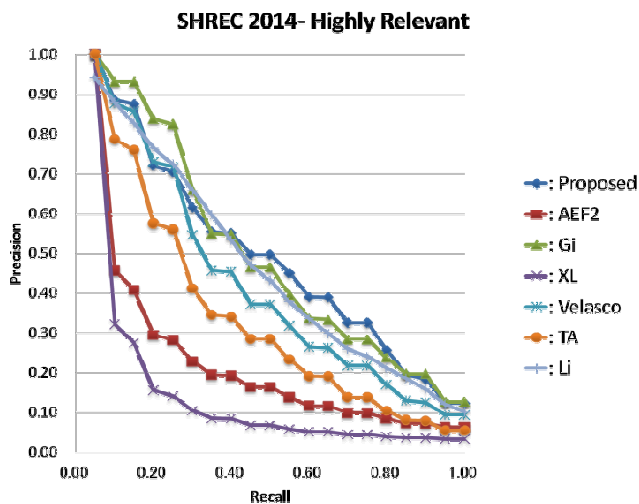


Fig. 16 Comparison result on SHREC'2014 high relevant

#### IV. CONCLUSION

In this paper, a novel approach based on local color voxel pattern called LCVP is proposed. This approach will extract 3D textured features by considering pattern on a voxel to its neighbors. To increase performance, aggregation with some spatial pattern such as Weber descriptor, oriented gradient, and multi Fourier spectral descriptor are utilized. Linear combination is chosen to cumulate each function. One process which should not be overlooked is pose normalization. This process gives a big impact to determine the most appropriate position for capturing the image. This problem still leaves some opportunities to increase performance.

The combination of all approach is also added with a rank combination between them to generate a more satisfactory result. To verify our approach, this approach has been experimentally evaluated with evaluation standards. According to the experiments, it is shown that this approach outperforms to state of the art. In the future, we wish to improve the proposed methods by improving how to do pose normalization and also consider about non-rigid objects.

#### ACKNOWLEDGMENT

The author would like to thank DIKTI (Indonesian Higher Education) for the research funding.

#### REFERENCES

- [1] Baert, Jeroen, Ares Lagae, and Ph. Dutré. "Out-of-Core Construction of Sparse Voxel Octrees." *Computer Graphics Forum*. Vol. 33. No. 6. 2014.
- [2] Novotni, Marcin, and Reinhard Klein. "3D Zernike descriptors for content based shape retrieval." *Proceedings of the eighth ACM symposium on Solid modeling and applications*. ACM, 2003.
- [3] Funkhouser, Thomas, et al. "A search engine for 3D models." *ACM Transactions on Graphics (TOG)* 22.1 (2003): 83-105.
- [4] Kriegel, Hans-Peter, et al. "Using sets of feature vectors for similarity search on voxelized CAD objects." *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*. ACM, 2003.
- [5] Kriegel, H-P., et al. "Effective similarity search on voxelized CAD objects." *Database Systems for Advanced Applications, 2003.(DASFAA 2003)*. Proceedings. Eighth International Conference on. IEEE, 2003.

- [6] Paulhac, Ludovic, Pascal Makris, and Jean-Yves Ramel. "Comparison between 2D and 3D local binary pattern methods for characterisation of three-dimensional textures." *International Conference Image Analysis and Recognition*. Springer Berlin Heidelberg, 2008.
- [7] Matsuda, Takahiro, Takahiko Furuya, and Ryutarou Ohbuchi. "Lightweight binary voxel shape features for 3D data matching and retrieval." *Multimedia Big Data (BigMM), 2015 IEEE International Conference on*. IEEE, 2015.
- [8] Tatsuma, Atsushi, and Masaki Aono. "Multi-Fourier spectra descriptor and augmentation with spectral clustering for 3D shape retrieval." *The Visual Computer* 25.8 (2009): 785-804.
- [9] Wu, Zhirong, et al. "3d shapenets: A deep representation for volumetric shapes." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015.
- [10] Xiang, Yu, et al. "Data-driven 3d voxel patterns for object category recognition." *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2015.
- [11] Tangelder, Johan WH, and Remco C. Veltkamp. "A survey of content based 3D shape retrieval methods." *Multimedia tools and applications* 39.3 (2008): 441-471.
- [12] Zhang, Cha, and Tsuhan Chen. "Efficient feature extraction for 2D/3D objects in mesh representation." *Image Processing, 2001. Proceedings. 2001 International Conference on*. Vol. 3. IEEE, 2001.
- [13] Paquet, Eric, et al. "Description of shape information for 2-D and 3-D objects." *Signal processing: Image communication* 16.1 (2000): 103-122.
- [14] Corney, Jonathan, et al. "Coarse filters for shape matching." *IEEE Computer Graphics and Applications* 22.3 (2002): 65-74.
- [15] Kolonias, Ilias, et al. "Fast content-based search of VRML models based on shape descriptors." *IEEE Transactions on Multimedia* 7.1 (2005): 114-126.
- [16] Kazhdan, Michael, et al. "A reflective symmetry descriptor for 3D models." *Algorithmica* 38.1 (2004): 201-225.
- [17] Kang, Sing Bing, and Katsushi Ikeuchi. "Determining 3-D object pose using the complex extended Gaussian image." *Computer Vision and Pattern Recognition, 1991. Proceedings CVPR'91., IEEE Computer Society Conference on*. IEEE, 1991.
- [18] Zhang, Cha, and Tsuhan Chen. "Indexing and retrieval of 3D models aided by active learning." *Proceedings of the ninth ACM international conference on Multimedia*. ACM, 2001.
- [19] Elad, Michael, Ayellet Tal, and Sigal Ar. "Content based retrieval of VRML objects—an iterative and interactive approach." *Multimedia 2001*. Springer Vienna, 2002. 107-118.
- [20] Osada, Robert, et al. "Shape distributions." *ACM Transactions on Graphics (TOG)* 21.4 (2002): 807-832.
- [21] Ohbuchi, Ryutarou, et al. "Shape-similarity search of three-dimensional models using parameterized statistics." *Computer Graphics and Applications, 2002. Proceedings. 10th Pacific Conference on*. IEEE, 2002.
- [22] Ankerst, Mihael, et al. "3D shape histograms for similarity search and classification in spatial databases." *International Symposium on Spatial Databases*. Springer Berlin Heidelberg, 1999.
- [23] Vranic, Dejan V., Dietmar Saupe, and Jörg Richter. "Tools for 3D-object retrieval: Karhunen-Loeve transform and spherical harmonics." *Multimedia Signal Processing, 2001 IEEE Fourth Workshop on*. IEEE, 2001.
- [24] Yu, Meng, et al. "3D model retrieval with morphing-based geometric and topological feature maps." *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*. Vol. 2. IEEE, 2003.
- [25] Kazhdan, Michael, Thomas Funkhouser, and Szymon Rusinkiewicz. "Rotation invariant spherical harmonic representation of 3 d shape descriptors." *Symposium on geometry processing*. Vol. 6. 2003.
- [26] Shum, Heung-Yeung, Martial Hebert, and Katsushi Ikeuchi. "On 3D shape similarity." *Computer Vision and Pattern Recognition, 1996. Proceedings CVPR'96, 1996 IEEE Computer Society Conference on*. IEEE, 1996.
- [27] Zaharia, Titus, and Françoise J. Preteux. "3D-shape-based retrieval within the MPEG-7 framework." *Photonics West 2001-Electronic Imaging*. International Society for Optics and Photonics, 2001.
- [28] Chua, Chin Seng, and Ray Jarvis. "Point signatures: A new representation for 3d object recognition." *International Journal of Computer Vision* 25.1 (1997): 63-85.
- [29] Körtgen, Marcel, et al. "3D shape matching with 3D shape contexts." *The 7th central European seminar on computer graphics*. Vol. 3. 2003.

- [30] Ojala, Timo, Matti Pietikainen, and Topi Maenpaa. "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns." *IEEE Transactions on pattern analysis and machine intelligence* 24.7 (2002): 971-987.
- [31] Tan, Xiaoyang, and Bill Triggs. "Enhanced local texture feature sets for face recognition under difficult lighting conditions." *IEEE transactions on image processing* 19.6 (2010): 1635-1650.
- [32] Murala, Subrahmanyam, R. P. Maheshwari, and R. Balasubramanian. "Local tetra patterns: a new feature descriptor for content-based image retrieval." *IEEE Transactions on Image Processing* 21.5 (2012): 2874-2886.
- [33] Zhang, Baochang, et al. "Local derivative pattern versus local binary pattern: face recognition with high-order local pattern descriptor." *IEEE transactions on image processing* 19.2 (2010): 533-544.
- [34] Jacob, I. Jeena, K. G. Srinivasagan, and K. Jayapriya. "Local oppugnant color texture pattern for image retrieval system." *Pattern Recognition Letters* 42 (2014): 72-78.
- [35] Chen, Jie, et al. "WLD: A robust local image descriptor." *IEEE transactions on pattern analysis and machine intelligence* 32.9 (2010): 1705-1720.
- [36] Dalal, Navneet, and Bill Triggs. "Histograms of oriented gradients for human detection." *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. Vol. 1. IEEE, 2005.
- [37] Hoffmann, Christoph Martin. "Geometric and solid modeling." (1989).
- [38] Li, Xuetao, et al. "Decomposing polygon meshes for interactive applications." *Proceedings of the 2001 symposium on Interactive 3D graphics*. ACM, 2001.
- [39] Biasotti, Silvia, et al. "3D shape matching through topological structures." *International Conference on Discrete Geometry for Computer Imagery*. Springer Berlin Heidelberg, 2003.
- [40] Hilaga, Masaki, et al. "Topology matching for fully automatic similarity estimation of 3D shapes." *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. ACM, 2001.
- [41] Sundar, Hari, et al. "Skeleton based shape matching and retrieval." *Shape Modeling International, 2003*. IEEE, 2003.
- [42] Iyer, Natraj, et al. "A Reconfigurable 3D Engineering Shape Search System: Part I—Shape Representation." *ASME 2003 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. American Society of Mechanical Engineers, 2003.
- [43] Serre, Thomas, Lior Wolf, and Tomaso Poggio. "Object recognition with features inspired by visual cortex." *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. Vol. 2. IEEE, 2005.
- [44] Novotni, Marcin, and Reinhard Klein. "A geometric approach to 3D object comparison." *Shape Modeling and Applications, SMI 2001 International Conference on*. IEEE, 2001.
- [45] Sánchez-Cruz, Hermilo, and Ernesto Bribiesca. "A method of optimum transformation of 3D objects used as a measure of shape dissimilarity." *Image and Vision Computing* 21.12 (2003): 1027-1036.
- [46] Dey, Tamal K., Joachim Giesen, and Samrat Goswami. "Shape segmentation and matching with flow discretization." *Workshop on Algorithms and Data Structures*. Springer Berlin Heidelberg, 2003.
- [47] Tangelder, Johan WH, and Remco C. Veltkamp. "Polyhedral model retrieval using weighted point sets." *International journal of image and graphics* 3.01 (2003): 209-229.
- [48] Jacobs, David. "Determining the similarity of deformable shapes." *Vision Research* 15.38 (1998): 2365-2385.
- [49] DeCarlo, Douglas, and Dimitris Metaxas. "Shape evolution with structural and topological changes using blending." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20.11 (1998): 1186-1205.
- [50] Chen, Ding-Yun, et al. "On visual similarity based 3D model retrieval." *Computer graphics forum*. Vol. 22. No. 3. Blackwell Publishing, Inc, 2003.
- [51] Shih, Jau-Ling, Chang-Hsing Lee, and Jian Tang Wang. "A new 3D model retrieval approach based on the elevation descriptor." *Pattern Recognition* 40.1 (2007): 283-295.
- [52] Mademlis, Athanasios, et al. "3D object retrieval using the 3D shape impact descriptor." *Pattern Recognition* 42.11 (2009): 2447-2459.
- [53] S. M. Metev and V. P. Veiko, *Laser Assisted Microtechnology*, 2nd ed., R. M. Osgood, Jr., Ed. Berlin, Germany: Springer-Verlag, 1998.
- [54] J. Breckling, Ed., *The Analysis of Directional Time Series: Applications to Wind Speed and Direction*, ser. Lecture Notes in Statistics. Berlin, Germany: Springer, 1989, vol. 61.
- [55] S. Zhang, C. Zhu, J. K. O. Sin, and P. K. T. Mok, "A novel ultrathin elevated channel low-temperature poly-Si TFT," *IEEE Electron Device Lett.*, vol. 20, pp. 569–571, Nov. 1999.
- [56] M. Wegmuller, J. P. von der Weid, P. Oberson, and N. Gisin, "High resolution fiber distributed measurements with coherent OFDR," in *Proc. ECOC'00*, 2000, paper 11.3.4, p. 109.
- [57] R. E. Sorace, V. S. Reinhardt, and S. A. Vaughn, "High-speed digital-to-RF converter," U.S. Patent 5 668 842, Sept. 16, 1997.
- [58] (2002) The IEEE website. [Online]. Available: <http://www.ieee.org/>
- [59] M. Shell. (2002) IEEEtran homepage on CTAN. [Online]. Available: [http://www.ctan.org/tex-archive/macros/latex/contrib/supported/IEEEtran/FLEXChip Signal Processor \(MC68175/D\), Motorola, 1996](http://www.ctan.org/tex-archive/macros/latex/contrib/supported/IEEEtran/FLEXChip Signal Processor (MC68175/D), Motorola, 1996).
- [60] "PDCA12-70 data sheet," Opto Speed SA, Mezzovico, Switzerland.
- [62] A. Karnik, "Performance of TCP congestion control with rate feedback: TCP/ABR and rate adaptive TCP/IP," M. Eng. thesis, Indian Institute of Science, Bangalore, India, Jan. 1999.
- [63] J. Padhye, V. Firoiu, and D. Towsley, "A stochastic model of TCP Reno congestion avoidance and control," Univ. of Massachusetts, Amherst, MA, CMPSCI Tech. Rep. 99-02, 1999.
- [64] *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification*, IEEE Std. 802.11, 1997.
- [65] Cerri, Andrea, et al. "SHREC'13 Track: Retrieval on Textured 3D Models." *3DOR*. 2013.
- Biasotti, Silvia, et al. "SHREC'14 track: Retrieval and classification on textured 3D models." *Proceedings of the Eurographics Workshop on 3D Object Retrieval*. 2014.