EMITTER International Journal of Engineering Technology
Vol. 4, No. 1, June 2016
ISSN: 2443-1168

Multi Voxel Descriptor for 3D Texture Retrieval

Hero Yudo Martono

Electronics Engineering Polytechnic Institute of Surabaya
Raya ITS Sukolilo Surabaya, Telp. +62-(031)5947280/fax. +62-(031)5946114
hero@pens.ac.id

Abstract

In this paper, we present a new feature descriptors which exploit voxels for 3D textured retrieval system when models vary either by geometric shape or texture or both. First, we perform pose normalisation to modify arbitrary 3D models in order to have same orientation. We then map the structure of 3D models into voxels. This purposes to make all the 3D models have the same dimensions. Through this voxels, we can capture information from a number of ways. First, we build biner voxel histogram and color voxel histogram. Second, we compute distance from centre voxel into other voxels and generate histogram. Then we also compute fourier transform in spectral space. For capturing texture feature, we apply voxel tetra pattern. Finally, we merge all features by linear combination. For experiment, we use standard evaluation measures such as Nearest Neighbor (NN), First Tier (FT), Second Tier (ST), Average Dynamic Recall (ADR). Dataset in SHREC 2014 and its evaluation program is used to verify the proposed method. Experiment result show that the proposed method is more accurate when compared with some methods of state-of-the-art.

Keywords: 3D object, voxel, tetra pattern.

1. INTRODUCTION

Nowadays more 3D models are becoming available on the web and large 3D database which need efficient search techniques. Some different approaches is performed to search 3D models based on shape and color information. Some approach based on shape is in literature [1-3]. Search method usually encompass features extraction and compare them for retrieval. These features must be highly descriptive as well as invariant with respect to many transformation such as changes in the object's pose, scale or mesh resolution to distinguish between different objects. Some methods exploit 3D model based on global descriptor and local descriptor. For global descriptor, they are cord-based, moment-based, wavelet-based, voxel-based spherical harmonics and spherical wavelet. For local descriptor, they are bag of features-based, 3D shape spectrum descriptor used within MPEG-7 framework and spin-image descriptor, but some also combine both local and global descriptor. Another approach is the view-based that represent 3D objects by a set of views taken from different viewpoints and then exploit content-based image retrieval techniques to perform comparison. They are five different groups of views that are extracted from the model and then a probabilistic approach is used to find the models that maximise the a posterior probability given the query model. Some other methods are available in [8-10].Shape Retrieval Contest (Shrec) appears to evaluate the performance of existing 3D shape similarity methods, by highlighting their strengths and weaknesses, using a common test collection allowing for a direct comparison of methods. The participants of SHREC 2014 propose methods in [11-12] such as using concept of scale invariant heat kernels, multi scale area projection transform, combination of LBP, LTP, WLD, HOG [13-16], spectral graph wavelet signature, multi resolution LBP, and concept of geodesic distance.

In this paper, we propose a new and effective 3D texture retrieval system which focus exploit voxels and generate some histogram such as biner voxel histogram, color voxel histogram, distance spectral histogram, tetra pattern histogram and low frekuensi of spectral fourier transform. The rest of this paper is organized as follows. Section 2 presents related works, section 3 presents construction of 3D texture descriptor and its subsection as details. Section 4 presents how to calculate similarity measures. Section 5 describes experimental result and discussion. Finally, Section 6 concludes this presentation.

2. RELATED WORKS

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 is began by paying attention on the characteristics of rigid 3D shape. Based on the representation of the shape descriptor in they divide shape matching into three categories:(1) feature based methods, (2) graph based methods and (3) geometry based methods. Rapid development on 3D textured retrieval was triggered by algorithm contest such as SHREC'13 and SHREC'14. There are several algorithms in. M. Abdelrahman et all described a 3D shape textured method by combining a geometric using scale invariant heat kernel signature (SI-HKS)and a photometric contribution. V Garro and A Giachetti used Histogram Area Projetion (MAPT). C. Li, A. Godil, A. Ben Hamza used the spectral geometry based framework for texture 3D shape representation and retrieval. This framework is based on the eigen decomposition of the Lapclace-Beltrami operator (LBO), which provided a rich set of eigen bases that were invariant to isometric transformation. It consist of two main stages: (1) feature extraction by using spectral graph wavelett 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 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 methods that basically computed two features: a shape and a color descriptor. A shape was represented by a geodesic distance matrix (GDM) and a color was represented by an 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-dimensioanl 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 are computed on the surface embedded in the three-dimensional color space. The vertices on 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.

3. 3D DESCRIPTORCONSTRUCTION

Our proposed method retrieves 3D models on the basis of both color and shape information. Figure 1 shows the global framework. Following a quite common approach there are some main stages: perform pose normalisation into all 3D models to get same orientation. We implement method in [17-18] to perform pose normalisation. Second, we define a normalised space for voxel representation, map an arbitrary polygonal object into this space and set each voxel as its rgb value or zero. We include all vertices, and then quantize them into 64x64x64 voxels. Next we build some histogram such as biner voxel, color voxel, distance voxel, tetra pattern and fourier voxel.

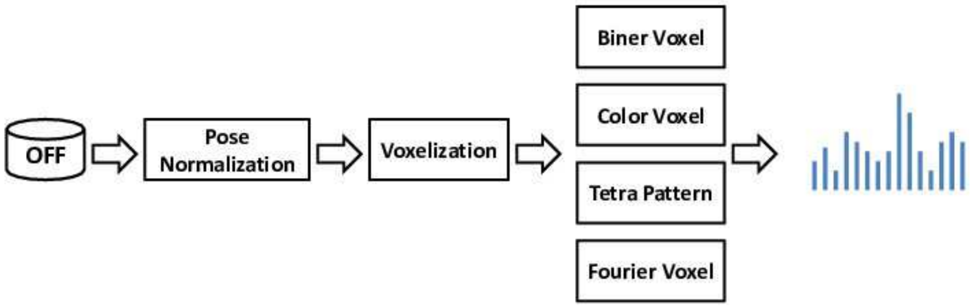


Figure 1. Architecture proposed retrieval system.

3.1 Pose Normalization

3D models have different in size, position and orientation. Because the feature is very depending on the size, position and orientation of the 3D model, then we should do pose normalization as an initial process. Pose normalization is a process of adjustment to the size, position and orientation of a 3D model in canonical space. Usually it involves computing three major principal axes perpendicular to each other by using the centroid or the maximum distance between the centroid and vertices of faces constituting a given 3D object.

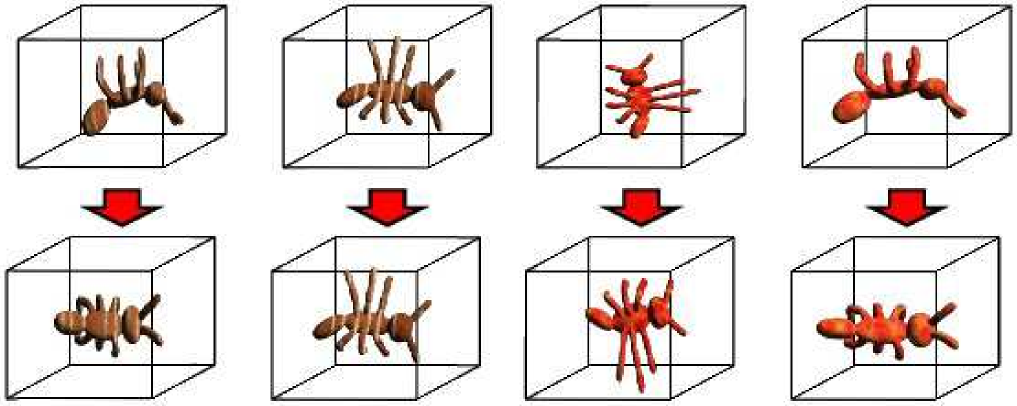


Figure 2. Example of Pose Normalization, above show original object, below show object after pose normalization.

Figure 2 shows an illustration of the importance of the pose normalization. Each 3D model has a different orientation, position, scale that affect the retrieval feature. The above is a simple illustration of the original 3D model. Pose normalization is then applied and generate 3D models with the same orientation in the bottom. In general, pose normalization is useful for reducing errors in orientation.

3.2 Construction Voxel

Voxel is a volume element representing a value on a regular grid in 3D space. By understanding voxels, we can estimate any 3D solid object. We define a normalised space for voxel representation, map an arbitrary polygonal object into this space, and set each voxel as either interpolation value (any polygon is crossed with the voxel) or zero otherwise. In our implementation, we map every vertex on the surface and then quantize them into 64 x 64 x 64 voxels. Similar method are in [17-18], but we are different on how to build voxel. They built voxel by random points on the surface while we implement all vertices. Since we need color information, so we compute interpolation to assign voxel value. The first step is to calculate the mean by involving all the vertices. There are three mean based on the coordinate axes x, y, z. Next we determine the maximum and minimum distance of a point from the mean.

Voxels size of 64x64x64 will create a cube. The mean of the vertices will be shifted to the center of the cube, so that the entire object will be included in a cube and have the same dimensions. Since each face consists of three points and has its own RGB then it will affect the color of the voxel.A color voxel will be strongly influenced by the color of the face that crossing it. The next step is to determine which color will affects voxel. For this, we use color interpolation and to make it easier. It is possible to map one face into more than one voxel. We then decide what color will be inserted into voxel. After mapping out all vertices, each voxel will be influenced by several different color vertex. Whenever the voxel is crossed by a vertex or face, then we will sum color value and sum vertex. The final color on each voxel is obtained by dividing the sum of color by the sum of points.

3.3 Color Voxel Histogram

An 3D model will be quantized by voxels 64x64x64. To distinguish 3D models, we then divide voxels based on distance and color level, shown as Table 1.

Table 1. Color voxel quantization level

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	-

3.4 Biner Voxel Histogram

Unlike color voxel, while biner voxel just set each voxel as one (any polygon that is crossed with the voxel) or zero (otherwise). We then quantize distance from centre to any voxel. It purposes to capture shape information. We also use level distance as shown Table 1 for generating biner voxel histogram.

3.5 Voxel Fourier Transform

We apply fourier transform to obtain low pass spectrum and eventually we leave 8 low-pass component. Thus, the total amount of dimensions of the voxel feature vector is 8 x 8 x 8 = 512. We implement fourier transform using this formula :

$$F_{\{p,q,s\}} = \sum_{x}.\sum_{y}.\sum_{z}.Voxel(x,y,z)\ \exp(-j2\,\pi\,(\frac{xp}{A}+\frac{yq}{B}+\frac{zs}{C}) \quad (1)$$

where A=B=C=64. We obtain 8 low-pass spectrum, thus the total amount of dimensions of the voxel feature vector is 8 x 8 x 8 = 512.

3.6 Tetra Voxel Histogram

Previously method LBP, LTP, LDP, DLEP, LTrP [19], LOCTP [20] and LOCSEP [21] motivated us to propose Voxel Tetra Pattern for 3D model textured retrieval. The LOCTP operator is obtained by computing the texture pattern over all three channels of the oppugnant color space. It is developed as a joint color-texture operator which helps to extract the uniform and non-uniform color texture features. The mechanism with which the LOCTP features is derived has been motivated by [20]. In the Oppugnant Color Texture Pattern, the local chromatic-texture operator is applied on each color channel separately. The color channels are used to collect the oppugnant color patterns in such a way that the centre pixel and neighbourhood pixels are taken from different color channels. Concept of tetra pattern is determining the relationship in terms of the intensity and directional information between the referenced pixel and their neighbors.

In this case, we implement that idea into cube by 5x5x5 voxel. We will capture tetra pattern from reference voxel at centre. Figure 4 shows, we first choose a voxel as a reference, we then involve neighbors and form a cube by size 5x5x5 and voxel reference is at the center. Next we will make a 2D plane (5x5) of the cube with 5 slices manner as presented in this Figure.Figure 5 shows an illustration of building tetra pattern. First, we already get a cube by size 5x5x5 and it has voxel reference in the center. Cube can be sliced in some direction as shown in right picture then produce 2D plane.The bottom picture is a 2D plane arrangement which viewed from the front. Its value is one of the RGB channel values

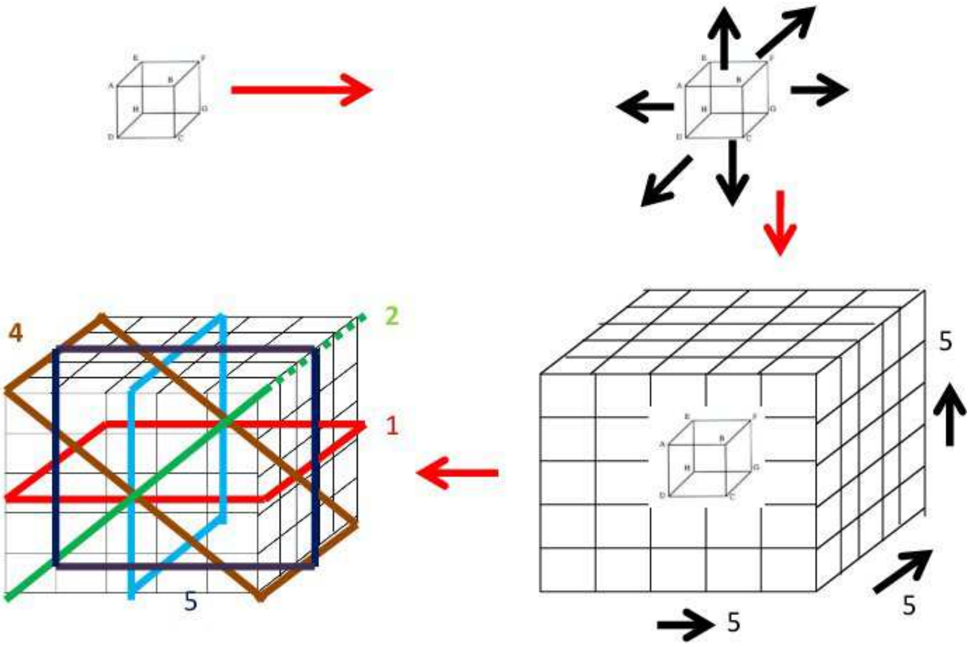
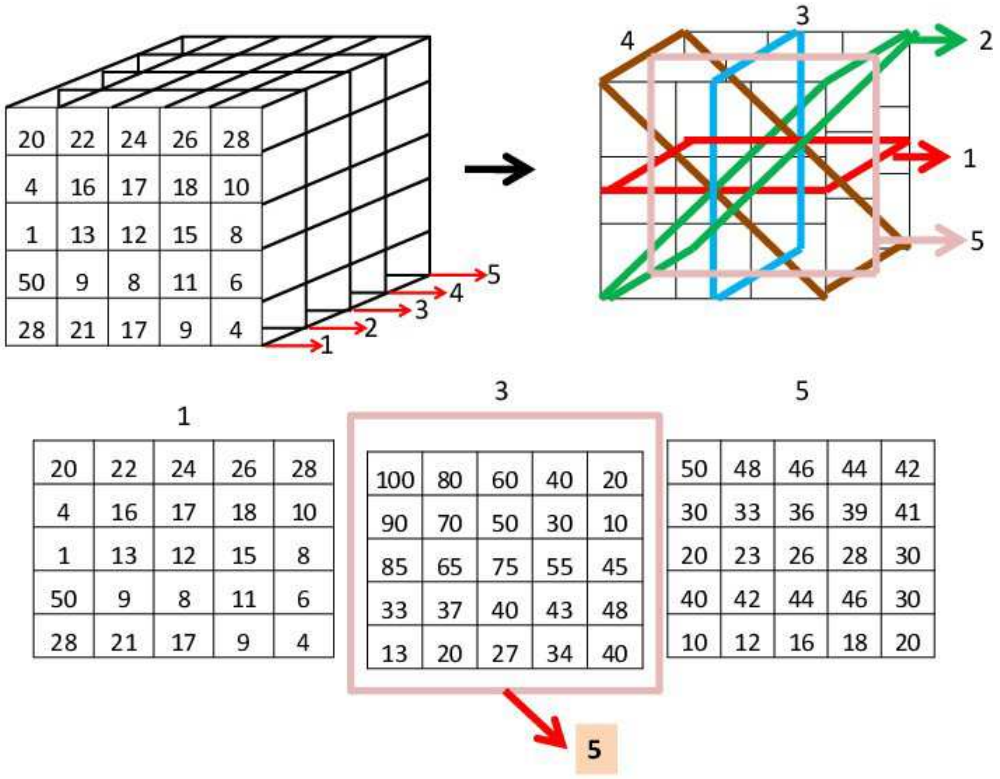


Figure 4. Building spectral voxel tetra pattern.



20	22	24	26	28
4	16	17	18	10
1	13	12	15	8
50	9	8	11	6
28	21	17	9	4

1

20	22	24	26	28
4	16	17	18	10
1	13	12	15	8
50	9	8	11	6
28	21	17	9	4

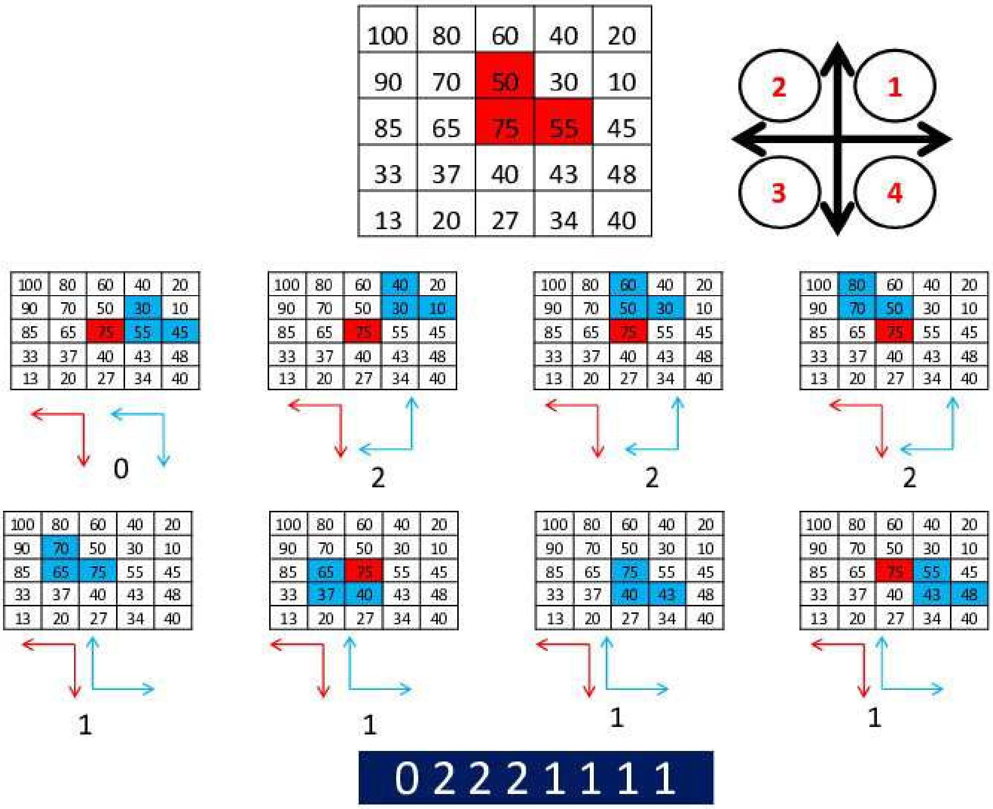
3

100	80	60	40	20
90	70	50	30	10
85	65	75	55	45
33	37	40	43	48
13	20	27	34	40

5

50	48	46	44	42
30	33	36	39	41
20	23	26	28	30
40	42	44	46	30
10	12	16	18	20

Figure 5. Spectral voxel decomposition.



100	80	60	40	20
90	70	50	30	10
85	65	75	55	45
33	37	40	43	48
13	20	27	34	40

Figure6. Computing spectral voxel tetra pattern.

Figure 6 shows an illustration creating tetra pattern for each referenced voxel. We separate all patterns into four parts based on the direction of reference voxel. For example, we start with the direction of reference voxel one. The reference voxel is 75 and its neighbors are 50 and 55. Because of 75 is greater than 50 and 55 so the direction is 3. Next we compute all the neighbors and get pattern from all points of the neighbors. Tetra pattern is coded 0 when it is equal to direction of reference voxel, otherwise coded in the direction of neighborhood reference voxel. Using the same analogy, tetra pattern are calculated for reference voxel having direction 2,3,4.

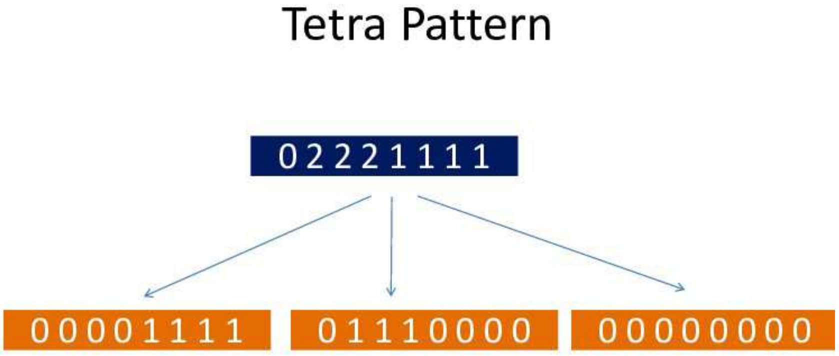


Figure 7. Separation tetra pattern

Finally, in Figure 7, we get tetra pattern as "0222 1111" with direction "3" for a reference voxel. We then separate it into three binary patterns as follows. The first pattern is obtained by keeping "1" where the tetra pattern value is "1" and "0" for other values, the we get "0000 1111". Similary, the other two binary pattern "0111 0000 " and "0000 0000 " are computed for tetra pattern values "2" and "4". In the same way, tetra patterns for reference voxel having direction 2,3, and 4 are computed. Thus with four tetra pattern, we will get 12 binary patterns.

4. SIMILARITY MEASURE

The next step is calculating the similarity between 3D objects based on their descriptors and which described in the previous section. We use a general standard 3D retrieval system where given a set of queries (Q) and the other set of candidate 3D models (M). For the experiment, we compute the similarity between the query with all existing 3D models using their descriptors. The approachment is taken by comparing between descriptors from the query to all existing 3D models separately and is computed by the Manhattan distance method, using this formula:

$$d(x,y) = \{\sum . \,|\, f\left(x_{\{i\}}\right) - f\left(y_{\{i\}}\right)\} \tag{2}$$

where x and y are two 3D models and f(xi) is to describe the i-th of feature x. We have compared the use of Manhattan distance, euclidean distance, and correlation to get dissimilarity between 3D objects. We choose Manhattan distance or L1 norm. Finally, how to determine the dissimilarity distance between objects is performing L1 norm. By using our descriptor, we have 4 different types of spectral features in the distance calculation. During this calculation, we use the sum of the weights from all the existing features and produce a shape descriptor distance. The results of our shape descriptor formula as follows :

$$distance = w_{\{d\}}d_{\{a\}} + w_{\{s\}}d_{\{b\}} + w_{\{c\}}d_{\{c\}} + w_{\{v\}}d_{\{d\}} \tag{3}$$

5. EXPERIMENT AND ANALYSIS

To assess the performance of proposed 3D model retrieval, we use a standard SHREC 2014 dataset. SHREC 2014 dataset consists of 572 mesh models grouped in 16 shape class and 13 texture class . The aim of SHREC is to evaluate the performance of existing 3D shape similarity methods, by highlighting their strengths and weaknesses, using a common test collection allowing for a direct comparison of methods. For the experimental setup, we adjust to the rules established by the committee in SHREC 2014 track: Retrieval and Classification on Textured 3D Models, including how to conduct an evaluation. For the purposes of fair comparison, we use the evaluation code from the committee. Detailed information can be found in this literature[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 query, then it is highly relevant, if it shares only shape, it is considered marginally relevant otherwise it is not relevant. The evaluation process has been based on the following evaluation measures: Average precision-recall curves, Nearest Neighbor (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. 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:

$$Precission = \frac{|A \cap B|}{|B|}\ , Recall = \frac{A \cap B}{|A|} \tag{4}$$

Nearest Neighbor, First tier and Second tier. These evaluation measures aim at checking the fraction of models 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:

$$ADR = \frac{1}{B}\sum_{i=1}^{B}\frac{r_i}{i} \tag{5}$$

We have conducted an experiment using a SHREC 2014 dataset and compared with the results of other participants that become state-of-the-art. They are such as Scale Invariant Heat Kernel, Color Data and Histograms of Area Projection Transform, Spectral geometry Consist of spectral graph wavelet signature for capture geometry and color histogram for the texture, Multiresolution Representation Local Binary Pattern histogram, Color and Shape descriptor by using the geodesic distance matrix. Some others define each color histogram using Earth Mover Distance, Geometry and Color Features based on geodesic distance, Textured Shape Distribution (TSD) by using the geodesic distance matrix. They also build a color histogram using the CIELAB space for representing color.

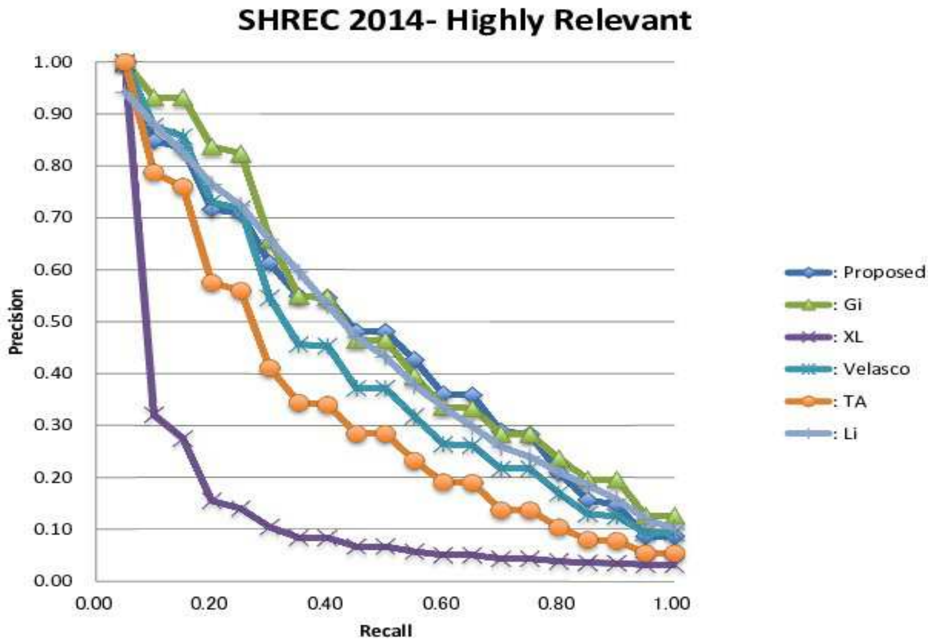


Figure 8. Precission-recall curve on the SHREC 2014 model dataset

We have conducted an experiment using a SHREC 2014 dataset and compared with the results of other participants that become state-of-the-art . We use this datasets because it has more model and diversity. The result can be seen in Table II and Figure 8. This proves that the proposed method could be a good alternative method. This method becomes the fourth on the Nearest Neighbor evaluation, the second on the First tier evaluation, the second on the Second tier evaluation and the third on ADR evaluation. Although it is not the best but our computation is simpler than the winner. This is because we do not use the geodesic matrix which requires a complete computing on time and requires more memory.In other words, our method uses less time and memory with similar performance. It means we provide another alternative, where our method can be used to solve the problem of 3D model retrieval based on shape and texture with the advantages in time and memory, but performance closed to the maximum results.

Tabel 2. Result Evaluation Comparison

Shrec 2014	NN	FT	ST	ADR
Proposed	0.717	0.439	0.576	0.368
G1	0.813	0.455	0.590	0.383
Ve	0.735	0.396	0.540	0.342
GG	0.722	0.432	0.557	0.368
LBG	0.676	0.412	0.565	0.353
TA	0.563	0.336	0.456	0.294
HA	0.486	0.278	0.362	0.260
AEF	0.123	0.228	0.351	0.206
XL	0.108	0.148	0.192	0.159

In Figure 8 shows the precision-recall curve. Our method is very closed to the winner. The experimental results illustrate that our method is good enough for retrieving 3D shape textured model.

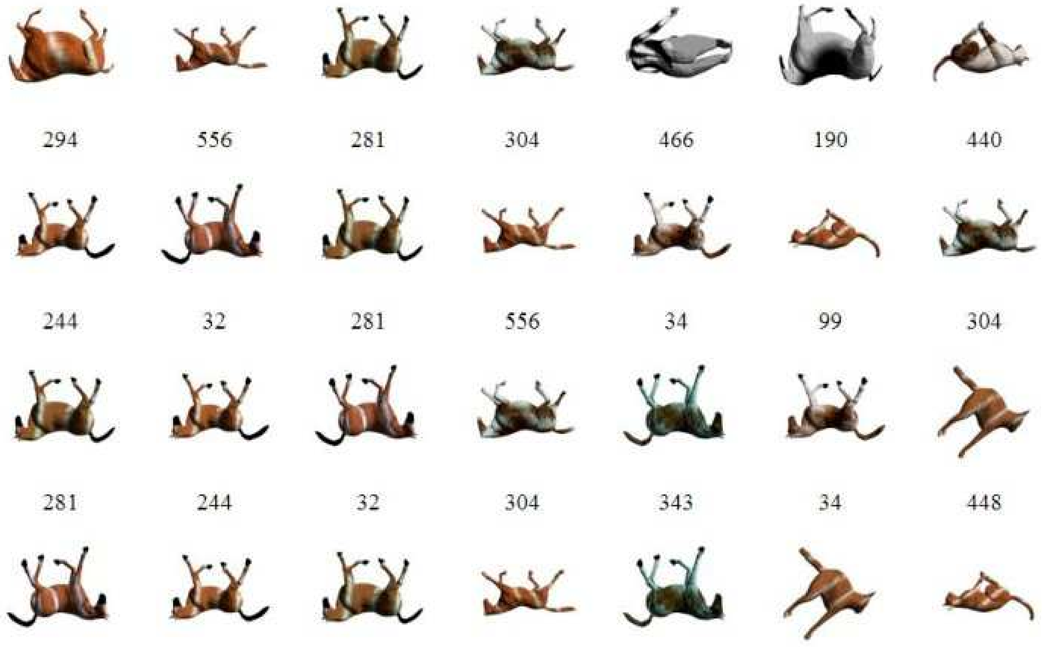


Figure 9. Example of 3D retrieval system with horse query on left and result on right

To show the results of our research, we try to capture a part of them. In Figure 9 shows that the left side is query and the next right column is the result. This is a final result which obtained by sorting its shape, color and texture. Numbers below the model is number model that is randomized by the committee. In that Figure, the query is a horse that has a certain style. We can see that the first answer to this query is a horse, but in the some next answer are other objects appear where its shape was deliberately similar to the query object. In Figure 10 shows another example of a query, it is a vase. Actually, several vases on dataset have variety of shape, color and texture. So the result is determined by a composite shape, color and texture. We see that the trend of results is shape then color or texture on the first priority, then shape on the next and the last answers may be not compatible with the query.

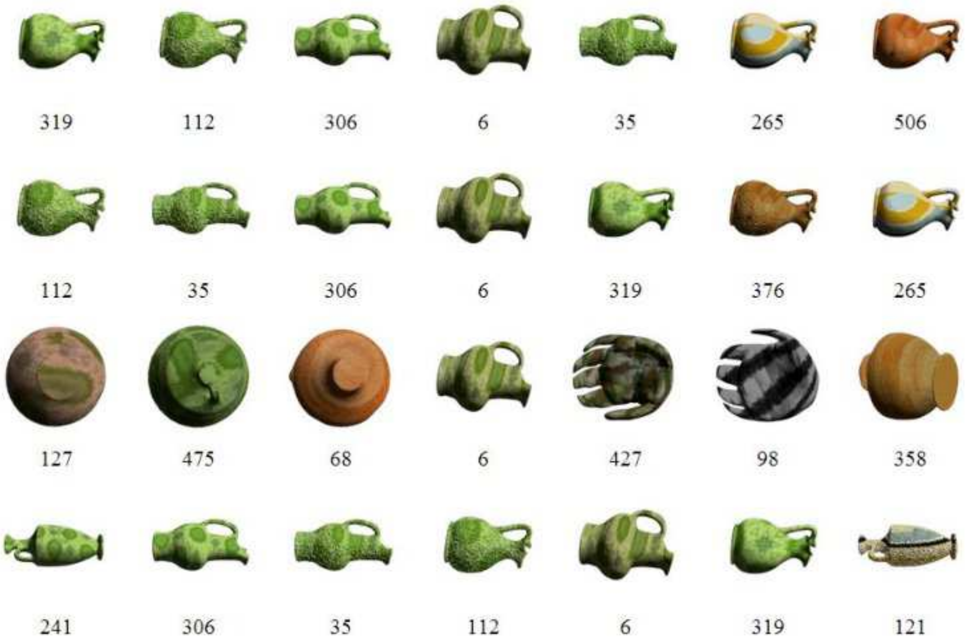


Figure 10. Example of 3D retrieval system with vase query on left and result on right

6. CONCLUSION

In this paper, a novel feature descriptors which focus exploring voxel has been proposed. We generate some features to retrieve 3D models based on shape, color and texture. We build biner voxel histogram, color voxel histogram, spectral tetra pattern histogram and spectral fourier transform histogram. Linear combination of these features to be effective descriptors. We test these features using SHREC dataset 2014 and outperform some participant. Generating these features is very fast. The implementation of the proposed scheme can be applied to various fields in domain 3D processing.

Acknowledgements

The authors would like to thank DIKTI (Indonesian Goverment for Higher Education) for scholarship.

REFERENCES

Journal:

[1] Lian, Z., Godil, A., Bustos, B., Daoudi, M., Hermans, J., Kawamura, S., ... & Ohkita, Y.. **A comparison of methods for non-rigid 3D shape retrieval**.*Pattern Recognition*, *46*.1 (2013), 449-461.

[2] 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.

[3] Bimbo, Alberto Del, and Pietro Pala. "**Content-based retrieval of 3D models.**"*ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 2.1 (2006): 20-43.

[4] Sun, Jian, Maks Ovsjanikov, and Leonidas Guibas. "**A Concise and Provably Informative Multi-Scale Signature Based on Heat Diffusion.**"*Computer graphics forum*. Vol. 28. No. 5. Blackwell Publishing Ltd, 2009.

[5] Reuter, Martin, et al. "**Laplace–Beltrami eigenvalues and topological features of eigenfunctions for statistical shape analysis.**" *Computer-Aided Design*41.10 (2009): 739-755.

[6] Bronstein, Alexander M., et al. "Shape google: **Geometric words and expressions for invariant shape retrieval.**" *ACM Transactions on Graphics (TOG)* 30.1 (2011): 1.

[7] Giachetti, Andrea, and Christian Lovato. "**Radial symmetry detection and shape characterization with the multiscale area projection transform.**"*Computer Graphics Forum*. Vol. 31. No. 5. Blackwell Publishing Ltd, 2012.

[8] Li, Chunyuan, and A. Ben Hamza. "**Intrinsic spatial pyramid matching fordeformable 3d shape retrieval.**" *International Journal of Multimedia Information Retrieval* 2.4 (2013): 261-271.

[9] Li, Chunyuan, and A. Ben Hamza. "**A multiresolution descriptor for deformable 3D shape retrieval**." *The Visual Computer* 29.6-8 (2013): 513-524.A

[10] Li, Chunyuan, and A. Ben Hamza. "**Spatially aggregating spectral descriptors for nonrigid 3D shape retrieval: a comparative survey**." *Multimedia Systems*20.3 (2014): 253-281.

[11] Cerri, Andrea, et al. "**SHREC'13 track: retrieval on textured 3D models**."*Proceedings of the Sixth Eurographics Workshop on 3D Object Retrieval*. Eurographics Association, 2013.

[12] Biasotti, Silvia, et al. "**SHREC’14 track: Retrieval and classification on textured 3D models.**" *Proceedings of the Eurographics Workshop on 3D Object Retrieval*. 2014.

[13] Dalal, Navneet, and Bill Triggs. "**Histograms of oriented gradients for human detection.**" *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*. Vol. 1. IEEE, 2005.

[14] Ojala, Timo, Matti Pietikäinen, and Topi Mäenpää. "**Multiresolution gray-scale and rotation invariant texture classification with local binary patterns**."*Pattern Analysis and Machine Intelligence, IEEE Transactions on* 24.7 (2002): 971-987.

[15] Zhao, Yang, et al. "**Completed robust local binary pattern for texture classification.**" *Neurocomputing* 106 (2013): 68-76.

[16] Chen, Jie, et al. "**WLD: A robust local image descriptor.**" *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 32.9 (2010): 1705-1720.

[17] 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.

[18] Vranic, Dejan, and Dietmar Saupe. "**3D shape descriptor based on 3D Fourier transform.**" (2001).

[19] Murala, Subrahmanyam, R. P. Maheshwari, and R. Balasubramanian. "**Local tetra patterns: a new feature descriptor for content-based image retrieval**."*Image Processing, IEEE Transactions on* 21.5 (2012): 2874-2886.

[20] 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.

[21] Reddy, A. Hariprasad, and N. Subhash Chandra. "**Local oppugnant color space extrema patterns for content based natural and texture image retrieval.**" *AEU-International Journal of Electronics and Communications* 69.1 (2015): 290-298.