

# Partial Matching for Real Textured 3D Objects using Color Cubic Higher-order Local Auto-Correlation Features

A. Kanezaki<sup>1</sup>, T. Harada<sup>1</sup> and Y. Kuniyoshi<sup>1</sup>

<sup>1</sup>Laboratory for Intelligent Systems and Informatics, Department of Mechano-Informatics, Graduate School of Information Science and Technology, The University of Tokyo, Japan

---

## Abstract

*In recent years the need for the retrieval of real 3D objects is growing more and more. However, if the 3D models are obtained without the use of special equipment such as engineered environments or multi-camera systems, they are often incomplete, making retrieval difficult. On the other hand, real models often include rich texture information, which can compensate for the limited shape information. In this paper we present new 3D shape features which take into account an object's texture. We demonstrate the retrieval performance of these features on a dataset of real textured objects and a real color 3D scene.*

Categories and Subject Descriptors (according to ACM CCS): I.4.7 [Image Processing and Computer Vision]: Feature Measurement—Feature representation

---

## 1. Introduction

Recently the state of the art of 3D scanning has dramatically advanced, allowing us to obtain precise 3D models of various objects with associated textures. It has also become possible to obtain 3D descriptions of large scenes. The availability of such precise 3D models motivates the research problem of real 3D object retrieval.

There are various important considerations which must be taken into account when approaching the real 3D object retrieval problem. First, the ability to perform partial matching is essential. Real 3D models are often incomplete because they are measured from a few directions only. Also, when searching for a query object in a 3D scene, partial matching is inevitable because objects are rarely segmented in the scene obtained with a 3D scanner. Furthermore, the matching method needs to be fast, in order for it to be applicable to large databases or large 3D scenes. If the matching is to be performed against a changeable dataset such as a 3D scene of daily environment, then the feature extraction process should also be fast. Secondly, whereas artificial 3D models can be textured at users' own will, real 3D models have their unique textures which contain rich information about the color and visual "feel" of the objects' surfaces. This texture information can supplement shape information and provide an addi-

tional clue for retrieval. Thus, real 3D object retrieval techniques should take into account texture information as well as shape information.

Although various 3D shape descriptors have been proposed up to now, few of them are appropriate for partial matching. REXT [Vra03] and SHT [KFR03] require spatial maps, and therefore it is difficult to apply these techniques to partial matching. It is also difficult to apply the well-known histogram-based descriptor technique 3DHT [ZP02], because it requires normal distances from the centers of mesh triangles to the origin. LightField [CTSO03], a 2D image-based descriptor is also difficult to apply since it requires multi-view images. To execute partial matching, [MBF06] makes use of structural descriptors while [MDTS07] extracts meaningful segments from 3D objects. In those approaches, however, only the parts whose segments match the predefined sub-parts of models in a database can be retrieved. [JH99] and [GCO06] succeeded in partial matching for real 3D objects of various segments, but the process of shape descriptor computation and matching is computationally expensive. Note that the descriptors mentioned above don't take texture information into consideration.

In this paper, we propose "Color Cubic Higher-order Local Auto-Correlation (CCHLAC) Features" which accumu-

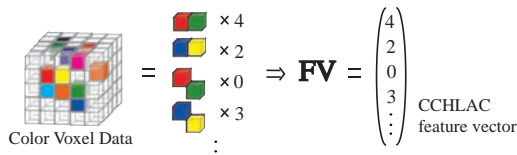
late local descriptors of both shape and texture patterns of objects' surfaces, and then we use these descriptors to match the query object against models of real textured 3D objects. These features can be extracted from partial models of any shape or size and their computation is fairly fast. Moreover, the feature vector of any part of the target model can be calculated simply by summing up the feature vectors of its subparts. This additive property allows high-speed matching of partial models of various sizes.

## 2. CCHLAC features

Proposed CCHLAC features are extension of CHLAC features [KO04]. CHLAC features are integrals of local autocorrelation of 3D voxel data. In our approach, CCHLAC features are computed by measuring the autocorrelation function of the 3D target object at specific points, represented by local patterns. Local descriptors are represented by the co-occurrence of their shape and colors. Because the feature vector consists of the sums of each pattern, it is insensitive to minor change, noise and loss of data. Moreover, these features enable partial matching of various sized models by choosing an appropriate integral interval.

### 2.1. Outline

The concept of CCHLAC features extraction is illustrated in Fig 1. A given element of the feature vector is calculated by summing the response of a particular local pattern over all the color voxels. Local patterns are expressed by the relative position of neighboring voxels, for example, two voxels lying in a row. By summing the response of each pattern, we obtain a rough measure of some property of an object's surface. The voxels of the local patterns are also differentiated by color, so that information about texture patterns can also be obtained. Because of this color sensitivity, the features can be used to discriminate between objects with the same shape, if their textures differ.



**Figure 1:** Illustration of CCHLAC features. The resulting feature vector consists of the sums of local patterns of color voxel data.

### 2.2. Extraction Process

In order to extract CCHLAC features, the color voxel data corresponding to the 3D model must be found. First, the bounding box of the target model is divided into voxel grid of a certain size, so that each voxel has property of being occupied by the object or not. We use the notation  $p(\mathbf{x}) = 1$  if a voxel whose position is  $\mathbf{x}$  is occupied, and  $p(\mathbf{x}) = 0$  otherwise. Secondly, RGB color property is added to the voxel

data. We represent them as  $r(\mathbf{x})$ ,  $g(\mathbf{x})$  and  $b(\mathbf{x})$ . Then each set of values is binarized using a certain threshold (127 in this paper). This process is effective in achieving robustness to changes in light intensities. Finally a voxel status  $\mathbf{f}(\mathbf{x})$  is represented as:

$$\mathbf{f}(\mathbf{x}) = \begin{cases} {}^t(r(\mathbf{x}) & 1-r(\mathbf{x}) & g(\mathbf{x}) & 1-g(\mathbf{x}) & b(\mathbf{x}) & 1-b(\mathbf{x})) & (p(\mathbf{x})=1) \\ {}^t(0 & 0 & 0 & 0 & 0 & 0) & (p(\mathbf{x})=0) \end{cases}$$

Then  $\mathbf{f}(\mathbf{x})$  is categorized into 9 patterns of voxel status (Fig 2). Since  $\mathbf{f}(\mathbf{x})$  contains not only RGB values but also their reverse values, the function is able to differentiate between low RGB value voxels and empty voxels. CCHLAC features are the integral of  $\mathbf{f}(\mathbf{x})$  or correlations of  $\mathbf{f}(\mathbf{x})$  between neighboring voxels. They are calculated by following equations:

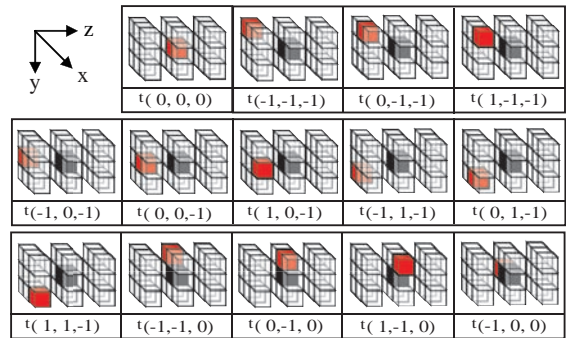
$$\mathbf{z} = \int \mathbf{f}(\mathbf{x}) d\mathbf{x}, \quad (1)$$

$$\mathbf{z}(\mathbf{a}_n) = \int \mathbf{f}(\mathbf{x}) {}^t\mathbf{f}(\mathbf{x} + \mathbf{a}_n) d\mathbf{x}, \quad (2)$$

Displacement vectors  $\mathbf{a}_n$  have 14 patterns (Fig 3). CCHLAC feature vector consists of all  $\mathbf{z}$  above excluding redundant elements, and then its dimension is 486.

State Variable	Voxel State	State Variable	Voxel State
$(0, 0, 0, 0, 0, 0)$	Empty	$(1, 0, 1, 0, 0, 1)$	Yellow
$(0, 1, 0, 1, 0, 1)$	Black	$(0, 1, 1, 0, 1, 0)$	Cyan
$(1, 0, 0, 1, 0, 1)$	Red	$(1, 0, 0, 1, 1, 0)$	Magenta
$(0, 1, 1, 0, 0, 1)$	Green	$(1, 0, 1, 0, 1, 0)$	White
$(0, 1, 0, 1, 1, 0)$	Blue	Others	× forbidden

**Figure 2:** Patterns of voxel status.



**Figure 3:** Patterns of displacement vectors. Position of red voxel is  $\mathbf{x} + \mathbf{a}_n$  while position of black voxel is  $\mathbf{x}$ .

## 3. Matching Method

### 3.1. CCHLAC features Cache

It is important for 3D features to enable matching of partial models of any size in real 3D object retrieval. It is also essential that the time required for the matching process should

be short. By choosing an appropriate integration interval, the CCHLAC features can be calculated from any region in each 3D model. Furthermore, the full feature vector of a partial model can be calculated simply by summing up the feature vectors of its sub-parts. Another merit of CCHLAC features for partial matching is that they are translation invariant. Because of this, the feature vector is insensitive to small linear position changes, so fast rough scanning should not degrade the results too much.

As a pre-processing step for object retrieval, we calculate and store the CCHLAC features of subdivided parts of models in a database. For example, suppose that the color voxel data of every object is  $300 \times 300 \times 300$  grid and CCHLAC features are computed from subdivisions respectively which consist of  $30 \times 30 \times 30$  voxels. Then  $10 \times 10 \times 10$  CCHLAC feature vectors at most are obtained. In the retrieval process, given a query part of a certain size, e.g.  $40 \times 20 \times 80$  voxels, we first choose an appropriate matching area size. This matching area size is chosen to be larger than the size of the query part and an integral multiple of the size of the subdivisions of the database models. In this case it becomes  $60 \times 30 \times 90$  grid. The similarity measure is simply the dot product of the query's CCHLAC feature vector with the corresponding feature vector of the given model with the same area. The matching process is fast because the CCHLAC features of each matching model area can be calculated by simply adding up some of the precomputed and cached feature vectors (in this case  $6 (= 2 \times 1 \times 3)$  vectors).

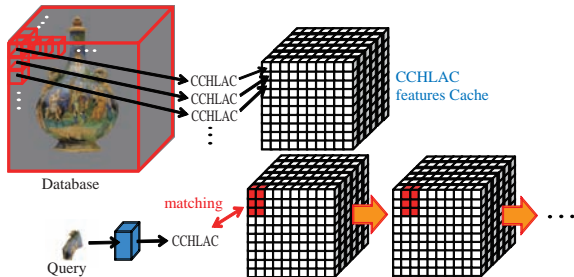


Figure 4: Matching method using CCHLAC features cache.

### 3.2. Achieving Robustness to Rotation

In principle, any feature based matching scheme can fail if the feature vector produced by an object changes significantly when the object is rotated. There are two basic approaches to solving this problem. One is to use a rotation-invariant descriptor, and the other is to rotate objects before the matching process. We opt for the latter approach, repeating the matching process with various postures of the query model. The feature vector of an object which is rotated by 90 degrees can be obtained rapidly through a simple exchange of the elements of the feature vector in the initial posture. This is possible because each displacement vector in Fig 3 is equivalent to another, rotated by 90 degrees. This method is

used together with smaller rotations of the object (e.g. 30 and 60 degrees) to obtain a set of poses which are tested against the model database. To obtain tight matching, the step size of the rotation should be small, but this increases computation time, and good matching can be obtained with reasonable step sizes.

### 4. Retrieval with color 3D real models

The performance of the proposed approach on the part-in-a-whole matching task was evaluated using the ‘‘Telecom Paris Image-based Digitized 3D Models Archive’’ [ES04], a dataset of real textured 3D objects. For this experiment we used the first 50 models. Every object is normalized with respect to scaling and converted into  $300 \times 300 \times 300$  color voxels. In this voxelization, we detected collisions of all triangles and voxels. 10 query parts (Fig 5) are obtained from the models, rotated to a random posture at the range from 0 to 359 degrees, and used as input to the matching procedure.

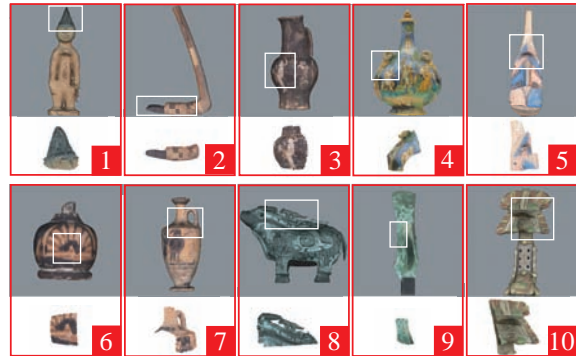
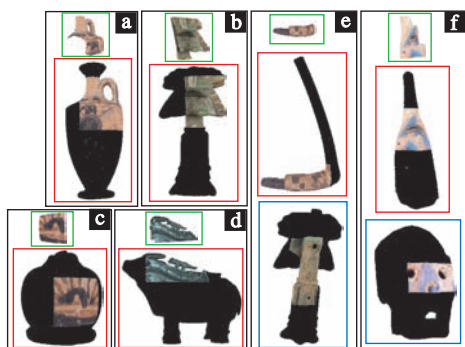


Figure 5: 10 query parts.

### 4.1. Results

Examples of the first retrieved model for a given query object are shown in Fig 6. The matching procedure generally ranks the correct model first, though the retrieval fails with some initial postures. Average correct answers in 300 trials are shown in Table 1. The numbers in the first row are first-tiers, percentage that the correct model is ranked as the first, the numbers in the second row are second-tiers, percentage that the correct model is ranked as the first or the second, and the numbers in the third row are third-tiers, percentage that the correct model is ranked within the third. We observe that a query part which has a strong color characteristic is correctly retrieved even if its size is rather small (e.g. No.9). On the other hand, mistakes sometimes occur if there is another model in the database with a similar shape (see (e) in Fig 6) or texture (see (f) in Fig 6). Table 2 shows the time required for computing the CCHLAC features of the 50 objects in the database (as a pre-processing step), for computing the CCHLAC features of 504 different postures of a query part, and for matching the query part against all sub-parts of the 50 database models. The reported computation time is obtained

using a Pentium IV 3.4 GHz with 1.0 GB of main memory and a C++ implementation.



**Figure 6:** Examples of first retrieved parts. Query parts are in green rectangles, correct answers are in red rectangles, and wrong answers are in blue rectangles.

**Table 1:** Average correct answers

No.	1	2	3	4	5	6	7-10
FT(%)	59.3	72.3	79.3	89.7	93.0	99.7	100
ST(%)	68.7	82.3	91.7	99.7	96.7	100	100
TT(%)	74.3	84.0	99.7	100	99.0	100	100

**Table 2:** Time of feature extraction and retrieval

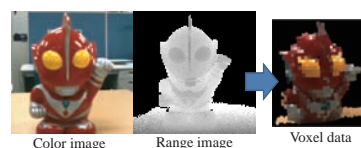
Process	Time(sec)
Feature extraction (database)	8.64
Feature extraction (1 query data)	0.31
Matching to all sub-parts	5.21

## 5. Search demonstration in 3D scene

In this section we demonstrate the results of using the system to find an target object in our lab room. In this experiment, the size of a voxel is  $10\text{mm} \times 10\text{mm} \times 10\text{mm}$ . The input data (Fig 7) is a sub-part of the query object measured from one direction, the size of which is  $35 \times 32 \times 24$  voxels. The target 3D scene is also partial and its size is  $334 \times 209 \times 216$  voxels. We use subdivisions of size  $10 \times 10 \times 10$  voxels. The search result is shown in Fig 8. The matching reports the exact position where the query object actually is. Note that the query object is on the different place in the map from the place where its input data was measured, and its lower body is occluded. The time required for the search is 6.07 sec by using a Pentium IV 3.4 GHz with 1.0 GB of memory and a C++ implementation.

## 6. Conclusions

In this paper, we proposed new features which describe the co-occurrence of shape and colors of an objects' surface. These features enable efficient partial matching of real textured 3D objects. Its retrieval performance was evaluated



**Figure 7:** Color image and Range image of the query object



**Figure 8:** Result of exploration. The discovered area is shown in the picture in the upper right.

with a database of real textured 3D models, and the possibility of using the system as a method of searching for objects in the real world was demonstrated. It is expected that our proposed approach will contribute to the real 3D object retrieval.

## References

- [CTSO03] CHEN D. Y., TIAN X. P., SHEN Y. T., OUHYOUNG M.: On visual similarity based 3D model retrieval. *Computer Graphics Forum* 22, 3 (2003), 223–232.
- [ES04] ESTEBAN C. H., SCHMITT F.: Silhouette and stereo fusion for 3D object modeling. *Computer Vision and Image Understanding* 96, 3 (2004), 367–392.
- [GCO06] GAL R., COHEN-OR D.: Salient geometric features for partial shape matching and similarity. *ACM Trans. on Graphics* 25, 1 (2006), 130–150.
- [JH99] JOHNSON A. E., HEBERT M.: Using spin images for efficient object recognition in cluttered 3D scenes. *IEEE Trans. Pattern Anal. and Mach. Intell.* 21 (1999), 433–449.
- [KFR03] KAZHDAN M., FUNKHOUSER T., RUSINKIEWICZ S.: Rotation invariant spherical harmonic representation of 3D shape descriptors. In *Proc. SGP* (2003).
- [KO04] KOBAYASHI T., OTSU N.: Action and simultaneous multiple-person identification using cubic higher-order local auto-correlation. In *Proc. ICPR* (2004), vol. 4, pp. 741–744.
- [MBF06] MARINI S., BIASOTTI S., FALCIDIENO B.: Partial matching by structural descriptors. In *Proc. Content-Based Retrieval* (2006).
- [MDTS07] MADEMLIS A., DARAS P., TZOVARAS D., STRINTZIS M. G.: On 3D partial matching of meaningful parts. In *Proc. ICIP* (2007).
- [Vra03] VRANIC D. V.: An improvement of rotation invariant 3D-shape descriptor based on functions on concentric spheres. In *Proc. ICIP* (2003).
- [ZP02] ZAHARIA T., PRETEUX F.: Shape-based retrieval of 3D mesh models. In *Proc. ICME* (2002).