Neurocomputing 151 (2015) 612–619

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Hypergraph based feature fusion for 3-D object retrieval

Fanglin Wang [a], Jialiang Peng [b,*], Yongjie Li [c]

[a] School of Computing, National University of Singapore, Singapore
[b] Information and Network Administration Center, Heilongjiang University, No. 74 Xuefu Road, Nangang District, Harbin, China
[c] Department of Computer Engineer, University of Naval Engineer, Wuhan, China

ARTICLE INFO

Article history:
Received 15 November 2013
Received in revised form
8 February 2014
Accepted 28 March 2014
Available online 22 October 2014

Keywords:
3-D object retrieval
Hypergraph learning
Dense Kerne LBP
Feature fusion

ABSTRACT

In view-based 3D object retrieval, each object is represented by a set of image views. 3D object retrieval becomes a group matching problem under such definition. Recent works have shown the effectiveness of hypergraph learning that computes the distance between 3D objects by solving a hypergraph structure problem. However, the single feature used in most of state-of-the-art works is often not sufficient to describe a 3D object. In this paper, we propose a feature fusion method based on hypergraph for 3D object retrieval. Besides the frequently used Zernike moments feature, we propose a Dense Kernel Local Binary Feature (DKLBP) feature for 3D object view description. A feature fusion method is proposed under the hypgraph framework. Experiments are conducted on the popular ETH-80 and National Taiwan University 3D model datatsets. Extensive experimental results show that the proposed approach has made significant performance improvement compared to other competitive approaches in recent works.

1. Introduction

In the past few decades, the development of modeling and digitizing technologies has made the 3D model generation process much easier. The emergence of new hardware makes 3D information acquisition more easier [1]. A large number of 3D models have been created and available for different applications such as virtual reality, CAD, and entertainment [2]. This leads to a rise of the 3D model retrieval systems. Efficient and effective 3D model retrieval algorithms are highly desired and attracted the attention of researchers [3–6].

The goal for 3D object retrieval is to search for 3D models that are similar to the query model. Most of the early 3D object retrieval methods are based on 3D model information (model-based methods) [7]. The model-based methods can be divided into two categories: geometry-based approach and visual similarity-based approach. The geometry-based approach concentrates on the shape-based or topology-based matching. It utilizes the vertices or polygons distribution, or topological structures of 3D model to measure the model similarity. The visually similarity-based approach proposes visual features to represent 3D models. For example, LightField Descriptor (LFD) is introduced to represent 3D models in [8]. LFD computes 10 silhouettes obtained from the vertices of a dodecahedron over a hemisphere. However, the early methods have many limitations. The main limitation is that 3D model information is not easy to obtain from real objects. These limitations severely affect the practical applications of these methods. With the rapid development of camera technology, we can now easily acquire images for 3D objects. Thus, view-based 3D model retrieval has become popular. View-based methods can represent 3D models using a set of 2D images. Compared to the methods based on 3D model information, it has following two advantages [9]:

- Firstly, it does not require 3D model information for the object.
- Secondly, it can benefit from the existing image processing technologies due to the importance of visual analysis in this kind of methods.

There are mainly two phases for view-based 3D model retrieval: modeling representation and matching. Many efforts have been devoted to the 3D model representation, such as Elevation Descriptor (ED) [10], Compact Multi-View Descriptor (CMVD) [11], and Bag-of-Region-Words (BoVF) [5]. However, these approaches usually utilize single feature and the comparison scheme for them highly depends on the view generation methods. 3D model matching approaches focus on the distance measure between models. As each object is represented by a set of views, the matching between two objects can be formulated as a many-to-many matching mode. There are different matching techniques that have been exploited. The Hausdorff distance and Sum distance have been widely used [10]. The Bayesian probabilistic method has been employed to retrieve similar 3D objects in

* Corresponding author.
E-mail address: jialiangpeng@hlju.edu.cn (J. Peng).

http://dx.doi.org/10.1016/j.neucom.2014.03.090
0925-2312/

adaptive view clustering [12]. Weighted Bipartite Graph Matching (WBGM) has also been investigated in [13] by modeling the views of two 3D objects into a bipartite graph. Recently, Gao et al. in [14] construct a hypergraph to model the 3D object relationship.

In this paper, we focus on view-based 3D model retrieval and tackle the model representation problem. Instead of using single feature in most of recent works, we apply multiple features to enhance the representation capability. In addition, we propose the Dense Kernel LBP feature for view image description, which is able to maintain the structural information effectively. The framework is also proposed for the multiple feature fusion based on hypergraph. In the framework, multiple hypergraphs are constructed for each feature and fused with different weights. The weighted hypergraph model is then solved. We conduct experiments on ETH-80 and NTU datasets to demonstrate the effectiveness of the proposed method. Comparisons with the-state-of-the-art work of single feature and hypergraph based methods demonstrate the superior performances of the proposed method in 3D object retrieval.

The rest of this paper is organized as follows. Section 2 briefly introduces the review of related works. Section 3 provides the visual features for 3D object retrieval including Zernike moments and Dense Kernel LBP feature. Section 4 is devoted to presenting the hypergraph-based feature fusion in detail. The experimental results are provided in Section 5. We conclude this paper in Section 6.

2. Related work

Lighting Field Descriptor (LFD) [8] captures representative views from the vertices of a dodecahedron over a hemisphere. The Zernike moments and Fourier descriptors are employed as a views features. Shih et al. have proposed a descriptor called Elevation Descriptor (ED) [10] for 3D model retrieval. Six range views are captured to describe the original 3D object with the altitude information of the 3D model, and the matching similarity of 3D models is based on the ED features. Compact Multi-View Descriptor (CMVD) [11] first captures multiple views by the camera array that is set at the 18 vertices of a 32-hedron, and then a set of 2D rotation-invariant shape descriptors, based on the Polar-Fourier Transform, Zernike Moments and Krawtchouk Moments, is produced as the final set of descriptor vectors. The Bag-of-Visual-Features (BoVF) method has also been investigated in view-based 3D object retrieval, for example in [15], dense SIFT features are extracted and visual words are generated. The distribution of visual words is used to describe the 3D model. These methods generally focus on model feature representations. They usually utilize single feature, and the fusion of multiple features is hardly no exploited for 3D object retrieval.

Despite the feature representation, distance estimation between 3D objects is also very important. Adaptive Views Clustering (AVC) [12] first captures 320 initial views for each 3D model, and the representative views are selected among them. Then, the Bayesian probabilistic method is employed for 3D model retrieval. A camera constraint-free view based 3D object retrieval algorithm (CCFV) is proposed in [16]. Gaussian models are used to represent the 3D feature distribution. A positive and negative matching model are trained by positive and negative matching samples, respectively. The distance between two 3D models is calculated based on the likelihood of positive matching model and dislikelihood of negative matching model. The methods often rely on the one-to-one view matching. Recently, a hypergraph model is investigated in [14]. Multiple hypergraphs are generated to capture the higher order relationship of 3D objects at different granularities. But it still ignore the multiple feature fusion.

The fusion of multiple features has shown its effectiveness in many different applications. In this paper, we exploit multiple features that have been proven useful in traditional image retrieval. Hypergraph model has been employed and solved to combine different features.

3. Visual features

This section introduces the visual features we use for 3D object retrieval. It should be noted our algorithm can be extended to more kinds of features like SIFT (Scale-Invariant Feature Transform) [17] or HOG (Histograms of Oriented Gradient) [18].

3.1. Zernike moments

Zernike moments descriptors have been widely studied for tasks like image recognition [19] and 3D object retrieval [14,12]. Zernike moments are a class of orthogonal moments and can overcome the shortcomings of information redundancy present in the popular geometric moments features. Based on the underlying scheme, Zernike moments are rotation invariant and can be easily constructed to an arbitrary order. In this paper, we use 49 dimensional Zernike moments features as in [14].

3.2. Dense Kernel LBP feature

Here we propose a new feature, named Dense Kernel Local Binary Pattern feature, DKLBP for short, for object representation of each view. DKLBP takes advantage of the excellent capability of representing textures of LBP feature. LBP has been proved effective in many computer vision missions [20]. To extract DKLBP, similar to the HOG feature [18], an image window is firstly divided into N overlapped blocks. Then for image block i, we first extract uniform LBP (ULBP) feature for each individual pixel and then compute the normalized histogram $\mathbf{B}_i$ whose bins correspond to ULBP patterns. After extract all histogram vectors of each block, the dense histogram vector is obtained by concatenating all the block histogram vectors into one high dimensional vector $\mathbf{B} = (\mathbf{B}_1, \ldots, \mathbf{B}_N)$.

During the normalization of block histogram $\mathbf{B}_i$, instead of frequently used L1 norm, $\mathbf{B}_i$ is computed by square root normalization of the original kernel histogram $\mathbf{H}_i = (q_i^1, \ldots, q_i^M)$ because of the better performance, i.e., $\mathbf{B}_i = \sqrt{\mathbf{H}_i / \|\mathbf{H}_i\|_1}$, where M is the number of bins. Here the normalized kernel histogram q_i^u at block i with bin u is computed according to

$$q_i^u = \frac{1}{C} \sum_{k=1}^{n} \delta(f(\mathbf{p}_i^k) - u) k(\|\mathbf{p}_i^k - \overline{\mathbf{p}}_i\|^2) \tag{1}$$

where n is the number of pixels in the i-th block, $\delta(\cdot)$ is the Dirac Delta function, $\mathbf{p}_i^k$ is the spatial position of k-th pixel in current block and $\overline{\mathbf{p}}_i$ is the block center, $f(\mathbf{p}_i^k)$ is the function mapping the pixel at $\mathbf{p}_i^k$ to a LBP pattern and $k(\cdot)$ is a kernel as used in [21] to weight a pixel according to the distance from itself to the block center. The kernel function can be Gaussian or Epanechnikov function, etc. Here we use Gaussian kernel:

$$k(\|\mathbf{y}\|^2) = \frac{1}{\sqrt{2\pi\sigma}^d} \exp\left(-\frac{\|\mathbf{y}\|^2}{2\sigma^2}\right) \tag{2}$$

where C is a normalization constant and can be calculated as

$$C = \sum_{k=1}^{n} k(\|\mathbf{p}_i^k - \overline{\mathbf{p}}_i\|^2) \tag{3}$$

which can be calculated in advance. Compared to the dense ULBP feature used in [22], DKLBP brings spatial information in some degree by adding a weight for each pixel by function $k(\cdot)$ during