From Low-Cost Depth Sensors to CAD: Cross-Domain 3D Shape Retrieval via Regression Tree Fields

Yan Wang[1], Jie Feng[2], Zhixiang Wu[2], Jun Wang[3], Shih-Fu Chang[1,2]

[1] Dept. of Electrical Engineering, Columbia University
{yanwang, sfchang}@ee.columbia.edu
[2] Dept. of Computer Science, Columbia University
jiefeng@cs.columbia.edu, zw2229@columbia.edu
[3] IBM T. J. Watson Research Center
wangjun@us.ibm.com

Abstract. The recent advances of low-cost and mobile depth sensors dramatically extend the potential of 3D shape retrieval and analysis. While the traditional research of 3D retrieval mainly focused on searching by a rough 2D sketch or with a high-quality CAD model, we tackle a novel and challenging problem of cross-domain 3D shape retrieval, in which users can use 3D scans from low-cost depth sensors like Kinect as queries to search CAD models in the database. To cope with the imperfection of user-captured models such as model noise and occlusion, we propose a cross-domain shape retrieval framework, which minimizes the potential function of a Conditional Random Field to efficiently generate the retrieval scores. In particular, the potential function consists of two critical components: one unary potential term provides robust cross-domain partial matching and the other pairwise potential term embeds spatial structures to alleviate the instability from model noise. Both potential components are efficiently estimated using random forests with 3D local features, forming a *Regression Tree Field* framework. We conduct extensive experiments on two recently released user-captured 3D shape datasets and compare with several state-of-the-art approaches on the cross-domain shape retrieval task. The experimental results demonstrate that our proposed method outperforms the competing methods with a significant performance gain.

1 Introduction

Shape-based retrieval and analysis of 3D models is an important research topic in computer vision, graphics, and computational geometry due to the wide applications in many domains such as archeology, architecture, medical imaging, and computer-aided design (CAD). In the past two decades, extensive efforts have been made to design effective 3D shape retrieval algorithms [1]. The existing work is mainly focused on two search scenarios, i.e., search by sketch [2][3] (Figure 1(a)) and search with CAD models as query input [1] (Figure 1(b)).Along

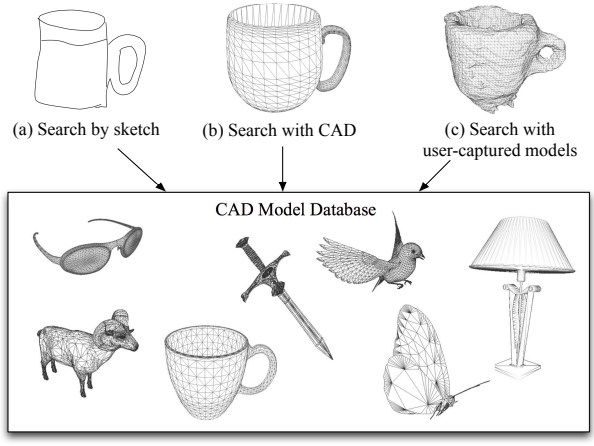

Fig. 1. Different 3D shape retrieval scenarios: (a) search by sketch; (b) search with CAD; and (c) cross-domain search with user-captured models from low-cost sensors.

with the advances of low-cost depth sensors such as Microsoft Kinect, PrimeSense sensors, and the newly revealed mobile depth sensor from Google [4], there is tremendous growth of user-generated 3D data, which promotes the study of a new *cross-domain* retrieval problem, i.e., *search with user-captured models*, where the users capture potentially noisy depth data and images of the object to their interest, and then use reconstructed 3D models as queries to find similar 3D shapes from a large collection of high-quality CAD models as illustrated in Figure 1(c). Such a cross-domain scenario also promotes new applications for 3D shape retrieval, such as high-quality 3D scanning, manipulation and printing.

Note that the existing methods for search with CAD models are often specifically designed for high resolution models with a well-controlled level of quality, which differ from the 3D models captured with low-cost sensors in several aspects. First, the user-captured models often contain a significant level of noise generated in either the capturing or the reconstruction process. Second, the generated model in uncontrolled environment is often incomplete due to various reasons like occlusions or partial views. Hence, this new retrieval scenario with user-captured models brings significant challenges in various aspects of shape analysis and retrieval, including 3D shape descriptor extraction, model representation and matching.

More specifically, existing 3D shape retrieval approaches generally follow two popular frameworks, local feature matching with optional spatial verification [5][6][7][8][9] and the Bag-of-Feature scheme [10][11], both of which require effective 3D local features. Although great progress has been made in 3D feature design, such as spin-image based descriptor [6], MeshDOG/MeshHOG [7], Heat Kernel Signature (HKS) [8][11], and Intrinsic Shape Context (ISC) descriptor [9], these low-level shape features highly rely on the quality of the 3D models and

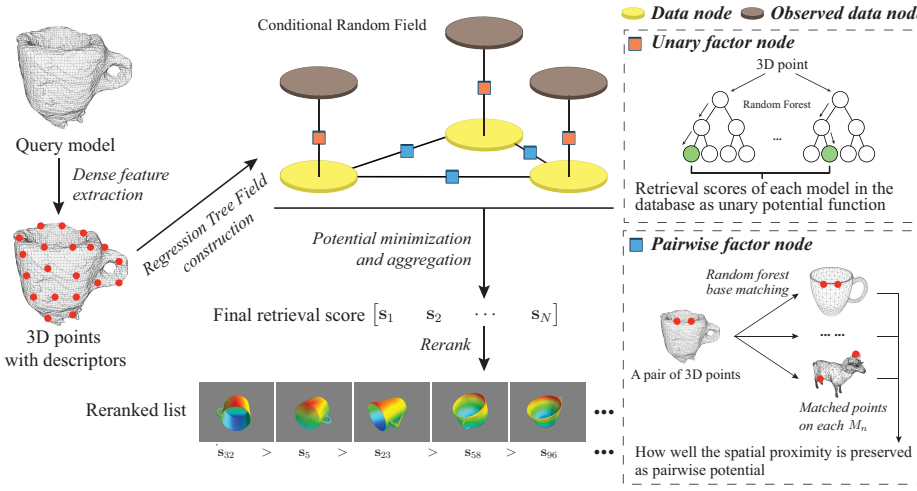

Fig. 2. Framework of our cross-domain 3D shape retrieval based on Regression Tree Fields. Best viewed in color.

tend to be sensitive to model noise that is often encountered with low-cost depth sensors. Furthermore, neither of these two frameworks explicitly address the challenge of partial models, resulting in degenerated performance in cross-domain 3D retrieval. For instance, previous study shows that the Scale-Invariant Heat Kernel Signature (SIHKS) achieves a high retrieval accuracy with CAD model queries [11], but significantly degrades for user-captured model queries [12]. To address these issues, spatial consistency checking has been used in both 2D [13] and 3D [10] cases. But the existing spatial consistency checking approaches such as pairwise feature quantization [10] and RANSAC [13] are still insufficient to handle the severe challenges associated with user-generated low-quality partial models, as observed in [12]. This is because the spatial consistency checking is often heuristic, and merely acts as a preprocessing or postprocessing, without principled optimization considering both feature similarity and spatial constraints.

To address the above two challenges, in this paper, we propose a robust and effective cross-domain shape retrieval approach by encoding local geometric structures in a Conditional Random Field (CRF), with a learned similarity measurement for robust feature matching. In particular, we build a CRF on the 3D points of the query model. Random forests are exploited to estimate rough similarity efficiently, thus to determine the unary potential. The geometric structures around each 3D point are embedded in the pairwise potential in a novel way, formulating the overall framework as a variant of *Regression Tree Field* [14], as show in Figure 2. Compared with the earlier approaches such as the Bag-of-Feature scheme and the existing partial matching algorithms, the proposed Regression Tree Field approach utilizes rich geometric information (instead of traditional pairwise spatial relationship checking) to compensate ill effects from model noise and incompletion. We evaluate our approach using two empirical

study cases for cross-domain shape retrieval: a) the Querying with Partial Models dataset from SHREC '09 [15]; and b) the Low-Cost Depth Sensing Camera data from SHREC '13 [12], both of which contain noisy 3D models reconstructed from low-cost depth sensors. The experimental results clearly demonstrate the superior performance of the proposed method, compared with several state-of-the-art 3D shape retrieval approaches.

The remainder of the paper is organized as follows. Section 2 presents a brief review of the related work. In Section 3, we give the details of the proposed *Regression Tree Field* based cross-domain shape retrieval method. The experimental results and comparison studies are reported in Section 4, followed by our conclusions and discussions in Section 5.

2 Related Work

As discussed earlier, most of the 3D shape retrieval and search methods can be grouped into the following two major categories: a) search by sketch; b) search with CAD models. Below we briefly review the representative approaches in each category. Detailed survey papers of shape retrial methods can be found in [1][16].

Search by sketch: As shown in Figure 1 (a), one first sketches a 2D projection of a 3D object and then uses the sketch as the query example to find similar 3D objects in a shape database, often containing CAD 3D models. Due to the simplicity, various techniques have been developed to retrieve 3D models whose 2D images match the query sketch. For instance, Funkhouser *et al.* used a variant of the 3D sphere harmonics to develop a shape search engine that accepts sketches as queries [2]. Yoon *et al.* employed suggestive contours and diffusion tensor fields to improve the robustness against shape and pose variance that often occurs in the user sketched images [17]. More recently, Shao *et al.* utilized a combination of contour-based representation and dense 2D matching to develop a robust approach that could perform partial matching between a query sketch and 3D models [18]. In summary, the sketch-based framework is still a popular choice for 3D shape retrieval and the influential SHape REtrieval Contest (SHREC) specifically has a sketch-based contest track. A comprehensive review on this topic is available in [16].

Search with CAD: The setting of search with CAD often requires the query sample to be a complete or partial CAD model. There have been two popular directions regarding to this task. One of them is to design powerful 3D shape signatures that can capture the intrinsic geometric information of the CAD models, with the motivation that the query and the database samples are essentially the same type of 3D models. To this end, various local features have been developed to describe the local geometry of 3D models, including Mesh-HoG as a 3D extension of the SIFT feature [7], Heat Kernel Signature [8][11], and Intrinsic Shape Context [9]. Realizing the sensitivity to model noise for those local descriptors [15], researchers also proposed to use high-level topological features [19][20], or aggregate low-level features to mid-level representations such as the extended Bag-of-Words model [10][21] and graph correspondences [22].

Another direction is to map 3D models to a set of views, each of which can be represented using 2D descriptors [23][24]. Although such multi-view shape descriptors can benefit from the discriminative power of mature 2D features such as SIFT, they often overlook the important spatial information and suffer from expensive computational cost due to the matching of a large number of views.

Finally, the recent rapid growth of consumer 3D models promotes the study of a new shape search scheme, i.e, search with consumer models, which explores cross-domain shape retrieval using models generated from low-cost depth sensors to query CAD model database. Representative efforts include the "Querying with Partial Models" track in SHREC '09 [15] and "Low-Cost Depth Sensing Camera" track in SHREC '13 [12]. However, the evaluation of existing shape retrieval methods on these two test benchmarks shows unsatisfactory performance due to the challenging issues of model noise and incompletion. Therefore, it motivates us to design robust and accurate cross-domain shape retrieval techniques which can compensate the low quality of the consumer models.

3 Approach

To address the partial matching problem for 3D shape retrieval using noisy models captured by low-cost depth sensors, we here propose to use a potential minimization formulation on a Conditional Random Field (CRF) defined on the query model, where the potential functions are efficiently estimated through random forest prediction. This forms a variant of Regression Tree Field [14], with a difference that the potential is not learned fully jointly, resulting in more affordable training and testing time for larger-scale shape retrieval. Below, we will first introduce the notations, and then illustrate the potential function design, followed by our efficient method to determine the potential functions.

3.1 Background and Notations

Assume we are given a database consisting of N 3D mesh models $\{M_n\}_{n=1}^{N}$ with n as the index of models, and a possibly incomplete and noisy user-captured model M_q as the query. The goal of a cross-domain shape retrieval engine is to return a ranked list of the 3D models in the database, such that the models ranked higher are more similar to the query.

In our formulation, we first construct a conditional random field on M_q with an undirected graph representation $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Here we specifically use the 3D points in M_q as the vertices $\mathcal{V} = \{v_i\}_{i=1}^{|\mathcal{V}|}$ with $|\mathcal{V}|$ being the cardinality, and the edges $\mathcal{E}$ are the connections between nearby 3D points in an ϵ-ball manner. For a 3D point v_i in M_q, we compute the Scale-Invariant Spin Image (SISI) [25] to represent the local geometry of a *3D patch* centered at v_i. The calculated 128-dimensional SISI descriptor is used as the *observation* $\mathbf{x}_i$ of the CRF. Besides the observation $\mathbf{x}_i$, each vertex is also associated with a continuous vector $\mathbf{y}_i \in \mathbb{R}^N$ as the output variable conditioned on $\mathbf{x}$, where the n-th element $(\mathbf{y}_i)_n$ denotes the partial matching score between the i-th patch of the query model

M_q and the n-th CAD model in the database. Compared to the standard CRF setting that often has a scalar as the output variable, in our CRF construction process, we have the output variable as a N-dimensional vector indicating the partial similarity between the 3D patch and each CAD model in the database. In the following, by designing the objective potential function to encode both the *shape similarity* and *geometric consistency*, we expect the inferred $\mathbf{y}$ to be a discriminant indicator for measuring the partial similarity between 3D patches and CAD models, while being robust to model noise and model incompletion in the cross-domain shape retrieval task.

3.2 Formulation

With the undirected graph model $(\mathcal{V}, \mathcal{E})$ and the associated random variables $\mathbf{x}, \mathbf{y}$, we can model the conditional distribution of the CRF. In particular, the optimal similarity scores $\mathbf{y}$ can be derived through minimizing the following potential function in a logarithmic form as

$$\mathbf{y}^* = \underset{\mathbf{y}}{\operatorname{argmin}} \log \Psi\big(\mathbf{y} \mid \mathbf{x}\big). \tag{1}$$

With the assumption that the conditional distribution obeys the Markov property with respect to the graph, the potential function $\Psi\big(\mathbf{y} \mid \mathbf{x}\big)$ can be further decomposed as a *unary* term Ψ_u defined on each vertex and a *pairwise* term Ψ_p defined on each pair of connected vertices,

$$\log \Psi(\mathbf{y} \mid \mathbf{x}) = \lambda \sum_{v_i \in \mathcal{V}} \log \Psi_u(\mathbf{y}_i \mid \mathbf{x}_i) + (1-\lambda) \sum_{(v_i, v_j) \in \mathcal{E}} \log \Psi_p(\mathbf{y}_i, \mathbf{y}_j \mid \mathbf{x}_i, \mathbf{x}_j), \tag{2}$$

where the coefficient λ is a parameter weighting contributions from these two terms. Note that the above two terms reflect important properties for shape retrieval. The unary term Ψ_u provides an robust estimation of similarity scores solely considering the local shape of the individual 3D patches, namely *shape similarity*. The pairwise term Ψ_p aims to further refine the scores by enforcing *geometric consistency* among neighbor patches. Through combining these two terms, our method can handle cross-domain partial matching with the unary term, while being less sensitive to model noise and incompletion due to the embedded geometric consistency in the pairwise term.

A natural concern of this formulation is the scalability, especially given that the optimization in Equation (1) may involve hundreds of variables with thousands of dimensions. But as we will show shortly, by exploring the sparsity of the problem and use discriminative random forests, inference on such CRFs can be very efficient and scalable to large-scale datasets.

Unary Potential. Following the standard practice of CRFs, the unary potential is used to penalize the variable $\mathbf{x}$ being far away from a rough estimation $\tilde{\mathbf{y}} = f(\mathbf{x})$. In particular, the unary potential is defined as a quadratic loss,

$$\log \Psi_u(\mathbf{y}_i \mid \mathbf{x}_i) = \frac{1}{2}(\mathbf{y}_i - f(\mathbf{x}_i))^T(\mathbf{y}_i - f(\mathbf{x}_i)). \tag{3}$$

Here the function $f : \mathbf{x} \in \mathbb{R}^{128} \rightarrow \mathbf{y} \in \mathbb{R}^N$ is a discriminative regressor which efficiently estimates similarity scores between a 3D patch in M_q and a database model $M_n, n = 1, \cdots, N$. We employ random forests as an ensemble learning method to build an efficient regression process, as discussed in Section 3.3.

Pairwise Potential. As a key difference from standard CRF formulation, the pairwise potential in our approach utilizes all the models in the database to help embed the local geometric structures. Intuitively, for a pair of neighbor vertices $(v_i, v_j) \in \mathcal{E}$ from the query model M_q, their corresponding vertices $v_i^n\prime, v_j^n\prime$ in a similar database model M_n should also be close by. Otherwise it indicates the spatial proximity of the neighbor vertices (v_i, v_j) is violated in the process of matching against the model M_n, and therefore M_n is not a spatially consistent candidate to the query. We define the pairwise term as

$$\log \Psi_p(\mathbf{y}_i, \mathbf{y}_j | \mathbf{x}_i, \mathbf{x}_j) = \sum_{n=1}^{N} \|\mathbf{v}_i^n\prime(\mathbf{x}_i) - \mathbf{v}_j^n\prime(\mathbf{x}_j)\|_2 \cdot (\mathbf{y}_i)_n (\mathbf{y}_j)_n. \tag{4}$$

Recall that $\mathbf{y}_i$ and $\mathbf{y}_j$ are the retrieval scores of the 3D patches $\mathbf{x}_i, \mathbf{x}_j$ in the query against all database models, with $(\mathbf{y}_i)_n$, $(\mathbf{y}_j)_n$ being the similarity scores against a database model M_n. Here, $\mathbf{v}_i^n\prime(\mathbf{x}_i)$ and $\mathbf{v}_j^n\prime(\mathbf{x}_j)$ are the 3D coordinates of the matched vertices in model M_n corresponding to 3D patches $\mathbf{x}_i$ and $\mathbf{x}_j$, respectively. Thus $\|\mathbf{v}_i^n\prime(\mathbf{x}_i) - \mathbf{v}_j^n\prime(\mathbf{x}_j)\|_2$ measures the Euclidean distance between two matched vertices in the model M_n. For well matched vertices $\mathbf{v}_i^n\prime(\mathbf{x}_i)$ and $\mathbf{v}_j^n\prime(\mathbf{x}_j)$, they are nearby with a small Euclidean distance, which indicates their similarity scores to the query patches will be less penalized recall we wish to minimize the potential function. On the contrary, if the matched vertices $\mathbf{v}_i^n\prime(\mathbf{x}_i)$ and $\mathbf{v}_j^n\prime(\mathbf{x}_j)$ are not spatially close to each other, indicating a large spatial distance, their similarity scores $(\mathbf{y}_i)_n, (\mathbf{y}_j)_n$ to the query patches will be suppressed since our objective is to minimize the above potential function. It is also worth noting that each model in the database is checked separately in the pairwise term computation, which does not require any pose estimation or calibration, thus being more reliable against sensor noise and incomplete models.

In practice, the straightforward local feature matching to find the corresponding vertices $\mathbf{v}_i^n\prime, \mathbf{v}_j^n\prime$ is unreliable under sensor noise. Therefore we further use random forests to robustly determine the vertex correspondences, as will be introduced in Section 3.3.

Inference. Let us define a matrix $\mathbf{V} \in \mathbb{R}^{N \times N}$ with its element V_{ij} calculated as $V_{ij} = \sum_{n=1}^{N} \left(\|\mathbf{v}_i^n\prime - \mathbf{v}_j^n\prime\|_2\right)$. Then, the pairwise potential can be written in a compact matrix form as

$$\log \Psi_p(\mathbf{y}_i, \mathbf{y}_j) = \mathbf{y}_i^T V_{ij} \mathbf{y}_j. \tag{5}$$

Hence, the overall log pairwise potential is represented as

$$\sum_{(v_i, v_j) \in \mathcal{E}} \log \Psi_p(\mathbf{y}_i, \mathbf{y}_j) = \mathbf{y}^T V \mathbf{y}, \tag{6}$$

where $\mathbf{y} \in \mathbb{R}^{N|\mathcal{V}|}$ is the concatenation of all the column vectors $\mathbf{y}_i$, and V is a blockwise matrix with $|\mathcal{V}| \times |\mathcal{V}|$ blocks, each as V_{ij}. Substituting Ψ_u and Ψ_p

in Equation 2 by the above derivations, we can derive the objective potential function in a quadratic form as,

$$\mathbf{y}^* = \underset{\mathbf{y}}{\operatorname{argmin}} \log \Psi(\mathbf{y} \mid \mathbf{x}) = \underset{\mathbf{y}}{\operatorname{argmin}} \quad \Big(\frac{1}{2}\mathbf{y}^T H\mathbf{y} - \mathbf{c}^T\mathbf{y}\Big), \tag{7}$$

where we have

$$H = \lambda I + (1-\lambda)V$$
$$\mathbf{c} = \lambda\tilde{\mathbf{y}} = \lambda f(\mathbf{x}).$$

$\mathbf{y}$ and $\tilde{\mathbf{y}}$ are the column concatenation of $\mathbf{y}_i$ and $\tilde{\mathbf{y}}_i$ (c.f. Unary Potential above) respectively. However, the above quadratic problem is not necessary to be convex since H might not be positive semi-definite in practice. Therefore we use the stationary point that gives the solution to the linear system $H\mathbf{y} = \mathbf{c}$ as an approximate solution. Because H is high dimensional, it is computationally prohibitive to directly compute the analytical solution to the linear system. Following Regression Tree Fields [14], we use the conjugate gradient descent to obtain the solution efficiently in an iterative manner. In addition, since H is often sparse, the inference procedure is fairly efficient, which usually ends in 10 iterations within 0.1 seconds on a desktop i7 CPU.

After computing the locally optimal solution $\mathbf{y}^* = \{(\mathbf{y}_i^*)_n\}$ $(1 \le i \le |\mathcal{V}|, 1 \le n \le N)$, we can derive the final ranking score to a query model as $\mathbf{s}_n = \sum_{i=1}^{|\mathcal{V}|}(\mathbf{y}_i^*)_n$, which will be used for reranking.

However, in order to make this framework fast enough for real applications, two critical problems remain unresolved: a) to efficiently obtain the rough estimation of the similarity scores $\{\tilde{\mathbf{y}}_i = f(\mathbf{x}_i)\}$ for the unary term; and b) to perform efficient matching of (v_i, v_j) against every model in the database to determine the pairwise potential term. Below we present our choice by using random forests to accomplish these tasks in a sub-linear testing time.

3.3 Efficient Estimation of Potential Functions

To achieve fast estimation of the similarity score $\{\tilde{\mathbf{y}}_i = f(\mathbf{x}_i)\}_{i=1}^{|\mathcal{V}|}$ in the unary potential term, we propose to use the random forest method to carry out a regression process. The training data contains all the extracted features of 3D patches from the database models as inputs, and the indices of the associated model as discrete responses. For the random forest, each decision tree is trained recursively using the standard information gain algorithm with the linear classifiers for data splitting. Finally, each leaf node in a decision tree receives a score vector $\mathbf{p}_l = [p_{l1}, \cdots, p_{ln}, \cdots, p_{lN}]$ measuring the frequencies of the patches of a specific 3D model falling in that leaf with the element computed as

$$p_{ln} = \frac{\#\text{ of training examples from the model } n}{\#\text{ of training examples}}, n = 1, \cdots, N.$$

Here $l = 1, \cdots, L$ is the index of the decision tree with L being the number of decision trees in the random forest.

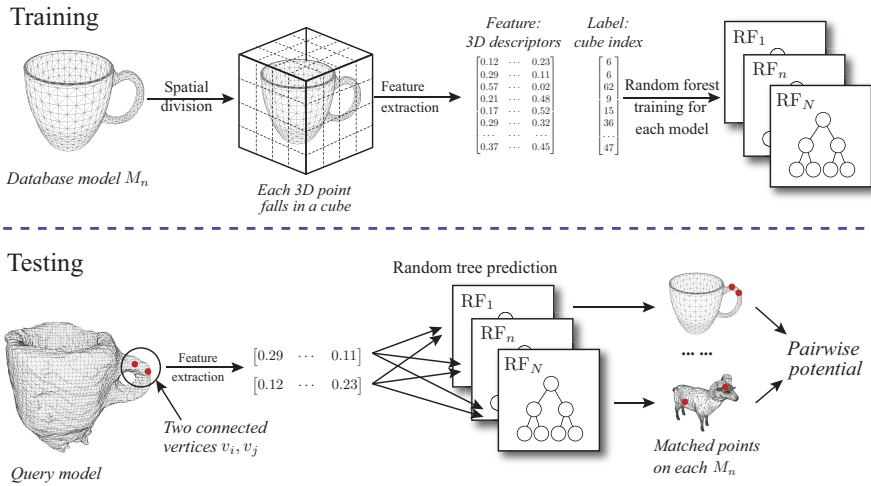

Fig. 3. Illustration of the efficient estimation of the pairwise potential term using random forests.

Given a feature vector $\mathbf{x}_i$ from a 3D patch in the query model, we first conduct examination from the root node to leaf nodes through all the decision trees in the trained random forest. The rough estimation of the similarity scores between a patch in the query model $\mathbf{x}_i$ and the CAD models are computed via averaging the recomputed score $\mathbf{p}_l$ on the retrieved leaf nodes as $\tilde{\mathbf{y}}_i = \frac{1}{L}\sum_{l=1}^{L}\mathbf{p}_l$. Compared to the traditional way that computes the similarity score by performing exhaustive matching between features, the regression method utilizes a discriminative decision model that can capture the underlying distributions of the features, resulting more robust estimation against model noise. In addition, the random forests method also benefits from the computational efficiency with a sub-linear time complexity, that can be further sped up for handling large scale applications through easy parallel implementations.

To estimate the pairwise potential term, it is necessary to find the best matched patch in a CAD model M_n for a query patch $\mathbf{x}_i$ to derive the corresponding vertex $v_i^n\prime$. Here we propose to again employ random forests to perform fast matching in a classification manner, with the framework shown in Figure 3. In particular, we set 3D bounding boxes on each CAD model and partition the model into $d \times d \times d$ voxels, each of which contains a set of 3D vertices. Here d is often set as a small value, such as $d = 4$ in our experiments. Then we use those partitioned vertices as training data to build a random forest *for each model* with the leaf node generating the prediction of which voxel the query patch falls into. The random forests are trained in the same manner by using the information-gain based algorithm. Then for a given query patch $\mathbf{x}_i$, we can quickly retrieve a small voxel in M_n that could contain similar patch, and adopt the center of that voxel as the matched vertex $v_i^n\prime$. Providing random forests empirically provide testing time of $\mathcal{O}(\log C)$, in which C is the class number, the total time

cost for matching a query patch with all the CAD models is $\mathcal{O}(N \log d)$, significantly faster than exhaustive matching with the time cost as $\mathcal{O}(N|\mathcal{V}|)$. In our experiments, we observe that such a random forest based matching achieves fast yet accurate matching results in practice. For instance, for a database with 720 models and a query with 500 points, it only requires less than 0.2 second on modern i7 CPUs to accomplish the matching procedure, where 80% of the matched results are the nearest vertices.

In summary, we formulate the cross-domain search as a potential minimization problem on a CRF, whose potential functions are dynamically determined from random forests, forming a variant of Regression Tree Field. The two challenges of sensor noise and model incompletion are resolved with the random forest based similarity computation and pairwise geometric consistency checking, which will be demonstrated quantitatively and qualitatively with experiments on real consumer models.

4 Experiments

To provide quantitative performance evaluation of the proposed cross-domain shape retrieval approach, we conduct experiments on two benchmarks from the well-known SHape REtrieval Contest (SHREC). The first dataset is from the Querying with the Partial Models track in the SHREC '09 [15], which consists of incomplete and noisy models captured from desktop 3D scanners. The second dataset contains query 3D models generated by Microsoft Kinect sensor that were used in the SHREC '13 [12]. Below we describe the details of the datasets, experimental settings, and evaluation results.

4.1 Datasets

The dataset from the Querying with Partial Models track of SHREC '09 is specifically designed to explore the frontier of 3D shape retrieval techniques in handling incomplete and possibly noisy query samples. It consists of a set of 720 high-quality CAD models as the database for querying. The CAD models are from 40 categories such as *bird, fish, mug* and *car* with 18 models for each category. In addition, it has two query sets, including a set of high-quality incomplete samples cropped from CAD models, and a set of user-captured models obtained with a desktop 3D scanner. Here we use the user-captured query set since it well represents the common challenges of cross-domain shape retrieval, such as surface noise and model incompletion due to self-occlusion. Examples of the physical objects used to capture the models are shown in Figure 4 (a), with the user-captured models shown in Figure 4 (b).

As another popular low-cost depth sensor, Microsoft Kinect is used to build 3D models using multiple range images [26]. Compared with single range image based 3D models like the SHREC '09 dataset, the Kinect-captured models tend to be more noisy due to non-smooth surfaces, and also with lower resolutions. In our experiments, we adopt the dataset from the Low-Cost Depth Sensing

(a) (b) (c)

Fig. 4. Illustration of the physical objects and the user-captured 3D models from the benchmark dataset: a) Physical objects used to generate the 3D models for the SHREC '09 dataset (Figure cited from [15]); b) An incomplete query model of the SHREC '09 dataset captured by a 3D desktop scanner; c) A noisy and low-resolution query model of the SHREC '13 dataset captured by the Microsoft Kinect.

Camera track of the SHREC '13 [12], which contains a total of 192 Kinect models. Note that the original test in the SHREC '13 is designed for 3D retrieval with both queries and database containing Kinect models. To test the cross-domain performance, here we use the CAD models from the SHREC '09 dataset as the database and use the 192 Kinect models from the SHREC '13 as the query set. Figure 4 (c) demonstrates an example of the used Kinect models.

4.2 Experiment settings

We conduct two types of empirical studies. On the SHREC '09 dataset, we provide quantitative performance evaluations and compared with several representative 3D shape retrieval methods. Since the query dataset from the SHREC '13 has no ground truth category information, we simply design qualitative evaluation by demonstrating the retrieval results.

For quantitative comparison, we compare with popular methods on CAD model retrieval and several approaches achieving state-of-the-art performance in the cross-domain contest track, including one 3D feature-based approach [10] and two 2D view-based approaches [15]. For our method, we also evaluate a variant that only uses the unary term without the pair-wise term of spatial consistency. Below we briefly describe the settings for each compared method.

- **Shape Google** [10]: We implement the Shape Google's approach [10], a shape retrieval approach for CAD models. For a fair comparison, we use the same Scale-Invariant Spin Image feature [25] as in our approach. A codebook with the size 10000 is built using the Approximate KMeans method [13].
- **CMVD-Depth** [15]: Achieving the best precision-recall in the SHREC '09 contest, the Compact Multi-View Descriptor (CMVD) extracts global 2D descriptors from the depth maps rendered from different views. The retrieval ranking is derived based on the minimum ℓ_1 distances between the signatures of the query and that of the database model.

Approach	MAP	NDCG
Shape Google	0.188	0.506
CMVD-Depth	0.193	0.521
CMVD-Binary	0.203	0.511
BF-GridSIFT	0.219	0.532
RTF-Unary	0.281	0.591
RTF	**0.315**	**0.611**

Table 1. The computed MAP and NDCG on SHREC '09.

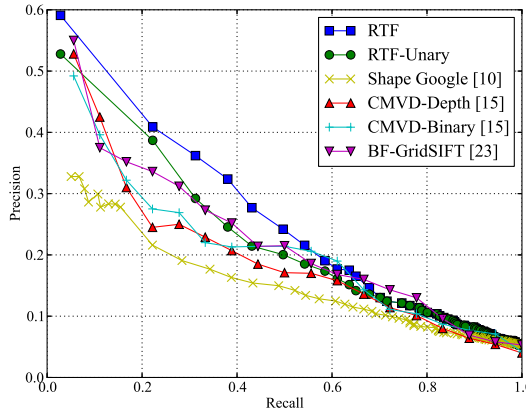

Fig. 5. Precision-recall curves of the evaluated approaches on the SHREC '09 dataset.

- **CMVD-Binary** [15]: **CMVD-Binary** is another approach with strong perform on the consumer model retrieval task in the SHREC '09 [15]. Different with the **CMVD-Depth** method that renders depth images, **CMVD-Binary** renders binary masks of the model to achieve computational efficiency and robustness against model noise.
- **BF-GridSIFT** [27]: As a state-of-the-art approach for both generic and user-captured model 3D shape retrieval, **BF-GridSIFT** first performs pose normalization to the models, and then renders depth maps from uniformly distributed views. Then the Bag-of-Feature scheme is employed to aggregate the extracted 2D dense SIFT descriptors. In the retrieval stage, KL-Divergence is used to compute a non-symmetric distance between the query sample and a database model.
- **RTF-Unary**: It is a simplified version of the proposed Regression Tree Field (RTF) based approach, which only considers the unary term by setting $\lambda = 1$ in Equation 2 and 7. Note that the **RTF-Unary** approach is equivalent to only using the computed similarity score from partial matching with random forests to perform ranking.
- **RTF**: The proposed Regression Tree Fields (RTF) approach. In the implementation of both RTF based methods, i.e., the **RTF-Unary** and the **RTF**, we use 128 trees with the depth 12 in the unary term. For the pairwise term in the **RTF** method, we apply bounding boxes to partition each model into 64 voxels ($d = 4$) and build a random forest with four trees with the height as 6. The coefficients balancing the two potential terms is set as $\lambda = 0.9$ uniformly across all the experiments.

To measure the performance, we adopt the semantic category information to evaluate the retrieved results. In particular, we treat the models from the same category as *relevant* and the models from different categories as *irrelevant* to compute two quantitative measurements as the evaluation protocols. First, we

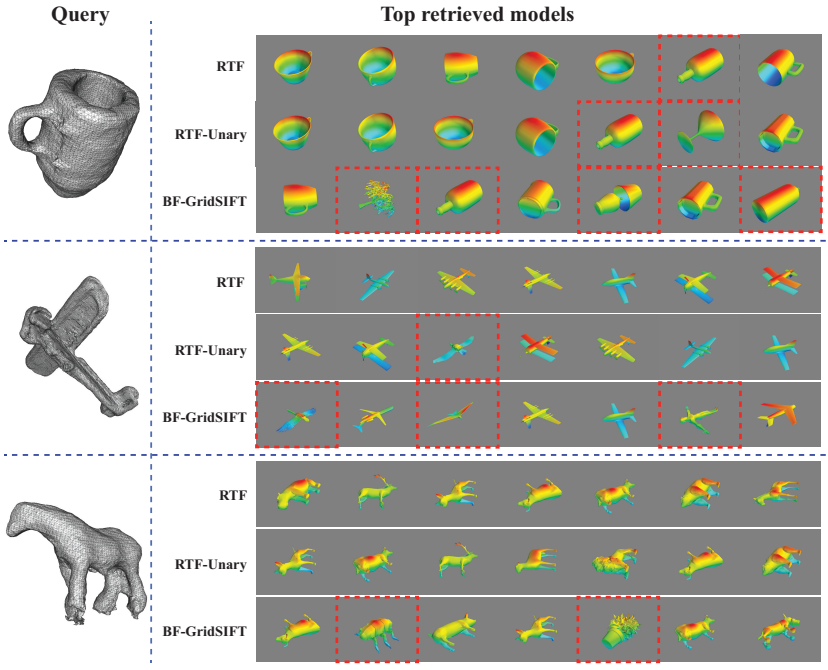

Fig. 6. Examples of the top results of the cross-domain shape retrieval, where the database contains CAD models from the SHREC '09 dataset and the query models are the user-captured models with the Microsoft Kinect. From the top to the bottom, the query models are *Mug*, *Airplane*, and *Quadruped*. And for each query, the three rows show the results from **RTF**, **RTF-Unary**, and **BF-GridSIFT** respectively. The results highlighted by red bounding boxes indicate the irrelevant 3D models.

compute the Mean Average Precision (MAP) that measures the average precision scores across all queries [28]. Second, we employ the popular evaluation criteria, the Normalized Discounted Cumulative Gain (NDCG) that is defined as

$$\mathrm{NDCG} = \frac{\sum_{n=1}^{N} \frac{\mathrm{Relevant}_n}{\log_2(n+1)}}{\sum_{n=1}^{N} \frac{1}{\log_2(n+1)}},$$

where $\mathrm{Relevant}_n$ is 1 when the nth sample is relevant to the query, otherwise 0. By assigning larger weights to the results ranked higher, the NDCG favors high-ranked relevant instances because they are more important for user experience. Below we report the results for both quantitative and qualitative evaluations.

4.3 Results

For the results on the SHREC '09 dataset, we report the MAP and NDCG for all the compared methods in Table 1, with the performance of **CMVD-Depth**, **CMVD-Bianry** and **BF-GridSIFT** cited from [15].

It is clear to see that the proposed **RTF** method achieves the highest performance among all the compared methods. Note that the pairwise term brings a significant performance improvement compared with **RTF-Unary** – a 12% gain in MAP. Although only exploring a single unary potential term, the **RTF-Unary** method achieves the second best performance in the SHREC '09 dataset. This is because the unary potential term derives cross-domain partial matching based similarity retrieval, which is suitable for addressing the model incompletion and noise issues on this dataset. Note that the methods adopting multiple views such as the **BF-GridSIFT** and the **CMVD-Depth** perform stronger than the single-view method **Shape Google**, which might be also due to the model incompletion issue on this data. In addition, we also plot the precision-recall curves for all the methods in Figure 5, which further confirms the clear performance gain of the proposed methods. Finally in terms of computational cost, on a desktop PC with an i7 3.0GHz CPU, the proposed method requires less than one second to perform the retrieval process in a database containing 720 objects, significantly faster than other compared methods.

On the SHREC '13 dataset, we present the qualitative evaluation by demonstrating the top retrieved 3D models in Figure 6. In particular, we compared the results of the two variants of our methods, i.e., the **RTF** and the **RTF-Unary**, and a strong competitor method the **BF-GridSIFT**. From Figure 6, it is clear to see that the **RTF** method outperforms the other two methods by generating semantically consistent 3D models for both simple object like *mugs* and complicated object like *planes*.

5 Conclusions

This paper addresses an emerging cross-domain shape retrieval problem, where the query samples are captured by users using low-cost depth sensors and the database contains conventional high-quality CAD models. To tackle the challenging issues like noise and incompletion of the user-captured models, we present a novel retrieval method that explores the unique power of Regression Tree Fields. In particular, we formulate our objective as a minimization problem of the CRF potential function, which contains a unary term measuring the similarity of cross-domain partial matching and a pairwise term with embedded geometric consistency. Both of these two terms are determined using efficient random forest algorithms. We conduct extensive empirical studies on two benchmark datasets from the well-known SHape REtrieval Contest (SHREC). The results clearly corroborate the superior performance of the proposed method, compared with other representative shape retrieval algorithms. Our future directions include introducing online random forest training algorithms [29] to avoid the necessity of retraining when adding new models, and also extending the proposed method to explore cross-domain 3D shape recognition and classification.

References

1. Tangelder, J.W., Veltkamp, R.C.: A Survey of Content based 3D Shape Retrieval Methods. Multimedia Tools and Applications **39**(3) (2008) 441–471
2. Funkhouser, T., Min, P., Kazhdan, M., Chen, J., Halderman, A., Dobkin, D., Jacobs, D.: A Search Engine for 3D Models. ToG **22**(1) (January 2003) 83–105
3. Zeleznik, R.C., Herndon, K.P., Hughes, J.F.: SKETCH: An Interface for Sketching 3D Scenes. In: ACM SIGGRAPH 2007 Courses. (2007)
4. Google Inc.: Project Tango. http://www.google.com/atap/projecttango/
5. Funkhouser, T., Kazhdan, M., Shilane, P., Min, P., Kiefer, W., Tal, A., Rusinkiewicz, S., Dobkin, D.: Modeling by Example. ToG **23**(3) (2004) 652–663
6. Johnson, A., Hebert, M.: Using Spin Images for Efficient Object Recognition in Cluttered 3D Scenes. Pattern Analysis and Machine Intelligence, IEEE Transactions on **21**(5) (May 1999) 433–449
7. Zaharescu, A., Boyer, E., Varanasi, K., Horaud, R.: Surface Feature Detection and Description with Applications to Mesh Matching. In: Proc. of CVPR. (2009) 373–380
8. Sun, J., Ovsjanikov, M., Guibas, L.: A Concise and Provably Informative Multi-Scale Signature Based on Heat Diffusion. Computer Graphics Forum **28**(5) (2009) 1383–1392
9. Kokkinos, I., Bronstein, M.M., Litman, R., Bronstein, A.M.: Intrinsic Shape Context Descriptors for Deformable Shapes. In: Proc. of CVPR. (2012) 159–166
10. Bronstein, A.M., Bronstein, M.M., Guibas, L.J., Ovsjanikov, M.: Shape Google: Geometric Words and Expressions for Invariant Shape Retrieval. ToG **30**(1) (February 2011)
11. Bronstein, M., Kokkinos, I.: Scale-Invariant Heat Kernel Signatures for Non-Rigid Shape Recognition. In: Proc. of CVPR. (2010) 1704–1711
12. Machado, J., Ferreira, A., Pascoal, P.B., Abdelrahman, M., Aono, M., El-Melegy, M.T., Farag, A.A., Johan, H., Li, B., Lu, Y., Tatsuma, A.: Shrec'13 track: Retrieval of objects captured with low-cost depth-sensing cameras. In: Proc. of EuroGraphics 3DOR. (2013) 65–71
13. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Object Retrieval with Large Vocabularies and Fast Spatial Matching. In: Proc. of CVPR. (2007)
14. Jancsary, J., Nowozin, S., Sharp, T., Rother, C.: Regression Tree Fields - an Efficient, Non-Parametric Approach to Image Labeling Problems. In: Proc. of CVPR. (2012) 2376–2383
15. Dutagaci, H., Godil, A., Axenopoulos, A., Daras, P., Furuya, T., Ohbuchi, R.: SHREC'09 Track: Querying with Partial Models. In: Proc. of Eurographics 3DOR. (2009) 69–76
16. Li, B., Lu, Y., Godil, A., Schreck, T., Bustos, B., Ferreira, A., Furuya, T., Fonseca, M.J., Johan, H., Matsuda, T., Ohbuchi, R., Pascoal, P.B., Saavedra, J.M.: A Comparison of Methods for Sketch-Based 3D Shape Retrieval. CVIU **119** (2014) 57 – 80
17. Yoon, S.M., Scherer, M., Schreck, T., Kuijper, A.: Sketch-based 3D Model Retrieval Using Diffusion Tensor Fields of Suggestive Contours. In: Proc. of ACM Multimedia. (2010) 193–200
18. Shao, T., Xu, W., Yin, K., Wang, J., Zhou, K., Guo, B.: Discriminative Sketch-based 3D Model Retrieval via Robust Shape Matching. Computer Graphics Forum (PG) (2011) 2011–2020

19. Tung, T., Matsuyama, T.: Topology dictionary for 3d video understanding. TPAMI **34** (August 2012) 1645 – 1657
20. Huang, P., Hilton, A., Starck, J.: Shape similarity for 3d video sequences of people. IJCV **89**(2-3) (2010) 362–381
21. Pickup, D., Sun, X., Rosin, P.L., Martin, R.R., Cheng, Z., Lian, Z., Aono, M., Ben Hamza, A., Bronstein, A., Bronstein, M., Bu, S., Castellani, U., Cheng, S., Garro, V., Giachetti, A., Godil, A., Han, J., Johan, H., Lai, L., Li, B., Li, C., Li, H., Litman, R., Liu, X., Liu, Z., Lu, Y., Tatsuma, A., Ye, J.: SHREC'14 track: Shape retrieval of non-rigid 3d human models. In: EG 3DOR. (2014)
22. Wang, C., Bronstein, M., Bronstein, A., Paragios, N.: Discrete minimum distortion correspondence problems for non-rigid shape matching. In: Proc. of International Conference on Scale Space and Variational Methods in Computer Vision. (2012) 580–591
23. Chen, D.Y., Tian, X.P., Shen, Y.T., Ouhyoung, M.: On Visual Similarity Based 3D Model Retrieval. Computer Graphics Forum (EuroGraphics) **22**(3) (2003) 223–232
24. Daras, P., Axenopoulos, A.: A 3d shape retrieval framework supporting multimodal queries. IJCV **89**(2-3) (2010) 229–247
25. Darom, T., Keller, Y.: Scale-Invariant Features for 3-D Mesh Models. TIP **21**(5) (May 2012) 2758–2769
26. Izadi, S., Kim, D., Hilliges, O., Molyneaux, D., Newcombe, R., Kohli, P., Shotton, J., Hodges, S., Freeman, D., Davison, A., Fitzgibbon, A.: KinectFusion: Real-time 3D Reconstruction and Interaction Using a Moving Depth Camera. In: Proc. of UIST. (2011) 559–568
27. Ohbuchi, R., Osada, K., Furuya, T., Banno, T.: Salient Local Visual Features for Shape-Based 3D Model Retrieval. In: Proc. of Shape Modeling and Applications. (June 2008) 93–102
28. Yilmaz, E., Aslam, J.A.: Estimating average precision with incomplete and imperfect judgments. In: Proc. of CIKM. (2006)
29. Ben-Haim, Y., Tom-Tov, E.: A streaming parallel decision tree algorithm. JMLR **11** (March 2010) 849–872