The Visual Computer (2020) 36:5–17
https://doi.org/10.1007/s00371-018-1597-4

ORIGINAL ARTICLE

3D indirect shape retrieval based on hand interaction

Erdem Can Irmak[1] · **Yusuf Sahillioğlu**[2]

Published online: 11 September 2018

Abstract
In this work, we present a novel 3D indirect shape analysis method which successfully retrieves 3D shapes based on hand–object interaction. To this end, the human hand information is first transferred to the virtual environment by the Leap Motion controller. Position-, angle- and intersection-based novel features of the hand and fingers are used for this part. In the guidance of these features that define the way humans grab objects, a support vector machine (SVM) classifier is trained. Experiments validate that SVM results are useful for retrieval of 3D shapes. We also compare the retrieval performance of our method with an interaction-based indirect method based on the Data Glove controller as well as a direct method based on 3D shape distribution histograms. These comparisons reveal different advantages of our method, which are (i) being lower-cost and more accurate compared to the Data Glove, and (ii) being more discriminative compared to a direct approach. We finally note that our algorithm is rigid-motion invariant and able to explore databases of arbitrarily represented 3D shapes.

Keywords Indirect shape analysis · 3D shape retrieval · Leap Motion · Interaction-based shape analysis · Data Glove

1 Introduction

Over the last few years, there is a growing demand for the analysis and retrieval of 3D shapes in areas such as computer-aided design, molecular biology, medicine, geometry modeling, computer animation, and video games. Searching 3D shapes in huge model databases is becoming an essential task to enhance design and discovery processes in a time-efficient manner. To achieve 3D shape searching easily and efficiently, related shape retrieval and analysis approaches need to be developed. Although text-based searching methods can be used if the text queries of the 3D shapes are well linked with the models, searching 3D shapes using only text query is not very practical in large databases. Features that are extracted directly from the form of the 3D object can also be preferred in the context of example-based direct searching, which, however, brings up the issue of finding the right 3D shape features for an accurate classification. Humans can easily classify objects from their surroundings according to their functionality, but for computers, it is unclear how much information is needed to detect their functionality. Without observing its complete functionality, problems may arise while classifying the 3D objects. Features are generally insufficient to capture the functionality.

We address the problem of recognizing and retrieving 3D digital objects that are geometrically similar but functionally different. Such objects regularly appear in our daily lives and consequently in the digital lives. This fact constitutes our main motivation for designing this system. The following design decisions are taken:

– Distinguishing geometrically similar but functionally different shapes, e.g., cylinder versus pencil, with conventional feature-based direct analysis methods is not sufficiently robust as they are defined based on purely geometric and/or topologic information. A better design choice is to guide the process via live user feedbacks which, however, takes away the attractiveness of a fully automatic system that is convenient for, e.g., batch processing. We consequently design an interaction-based system free of direct geometric/topologic features and user interventions.

✉ Erdem Can Irmak
erdem.can.irmak@gmail.com; e150247@metu.edu.tr

Yusuf Sahillioğlu
ys@ceng.metu.edu.tr

[1] Game Technologies Department, Middle East Technical University, Ankara, Turkey

[2] Computer Engineering Department, Middle East Technical University, Ankara, Turkey

– For the interaction agent, we prefer a non-rigid hand agent over a body agent since it interacts with small indoor objects in a more natural way.
– For the hand interaction, we preferred the Leap Motion sensor over the Data Glove sensor since it is cheaper and able to provide a larger variety of features which in turn renders it more accurate.
– For the method input, we admit a large domain of 3D object representations including clean manifold meshes, non-manifold meshes, meshes of arbitrary genus, polygon soups, and point clouds. Also, rotation and translation differences in the input do not affect our rigid motion invariant method.

2 Related work

There are various methods that have been proposed for 3D shape retrieval and analysis since the early 1990s. This section discusses and reviews some of the most significant papers from this domain by dividing them into three categories: rigid, non-rigid, and interaction-based retrieval and analysis methods.

2.1 Rigid shape retrieval and analysis

A shape is considered to be rigid if it can be observed under rigid transformations, which are translation and rotation. Methods based on rigid-motion invariant features such as volume–surface ratio and Fourier transform of the volume exist in the literature for rigid shape retrieval task [1–3]. Osada et al. [4] introduce a novel method that computes features extracted from 3D rigid shapes. This approach aims to decrease the computational complexity by comparing only 3D shape distributions. The fundamental idea of this work is generating a shape function using different approaches. Shape functions are used to create shape distribution histograms which are the main elements to measure similarities.

Another work for rigid shape retrieval is shown by Paquet et al. [5]. According to that article, 2D and 3D shape descriptors can be extracted from MPEG-7 images of the shapes. In this work, bounding box information is extracted from images and is used for categorizing 3D shapes. This approach naturally extends to 2D image retrieval [6].

Using Zernike invariants as 3D shape descriptors is yet another method for 3D rigid shape retrieval [7]. This algorithm aims to classify 3D shapes according to their general categories by utilizing the 3D Zernike descriptors using Canterakis' work [8].

Similarity-based 3D shape retrieval by Chen et al. [9] is another rigid shape retrieval method. The main idea of this approach is that if two 3D shapes are similar, then they should be viewed similar from certain view points. To this effect, Chen et al. introduce a novel method based on the Light Field Descriptor to describe 3D shapes. Using Light Field Descriptors, they extract 3D shape features from camera views at different angles. This work aims to reduce the feature size and decrease the complexity of the retrieval process.

In general, global rigid shape features are suitable for most of the 3D shapes, but some shapes become distinctive according to their local features, which are defined locally around shape points. According to Shilane et al. [10] similarity of two shapes can be found using their set of local descriptors. However, this work also states that finding local features in local shape retrieval is a highly expensive process. Because of this, they introduce a new method for selecting the most distinctive local features. Local features are selected from several regions for each 3D shape, and their retrieval performance is computed using multivariate Gaussian distributions. This method uses only important local features because using all local features in 3D shapes results in longer retrieval times.

With the advancements in computational hardware and developments of large-scale public repositories, new techniques have been introduced in rigid retrieval [11–13]. Thus, multi-view-based approaches became another essential technique for 3D shape analysis and retrieval. Bai et al. [14] present a new method that obtains projective images from different angles and collects features of the 3D models using GPU acceleration. Also, reducing the time complexity helps to work with large model databases. Gao et al. [15] introduce another view-based method that constructs hyper-graphs using 2D views. Most of these approaches have predefined camera array settings. Using constraint-free camera arrays [16], on the other hand, improves matching accuracy.

A different line of thought for rigid shape retrieval is introduced by Leifman et al. [17] with their interactive relevance feedback mechanism. Such a feedback helps user influence the search results, which in turn enables retrieval of semantically similar rigid shapes. This work also utilizes a novel descriptor that captures the characteristics of the geometry and topology of the model.

2.2 Non-rigid shape retrieval and analysis

A shape is regarded as non-rigid if it can be observed under rigid transformations and an additional bending transformation, e.g., articulated poses of a human avatar. Non-rigid 3D shape retrieval methods are getting significant research attention with the increasing popularity of the recent trends in multimedia contents such as prerendered images, motion captures, computer animations, video games and interactive applications. 3D shapes such as human body and hand models that appear in different poses by using different joint data are widely employed in both virtual and real environments [18,19]. Categorization and analysis of these similarly structured but distinctly posed 3D shapes are needed. These

non-rigid models can be inaccurately classified as different shapes if we use rigid shape analyzing techniques.

Non-rigid shape retrieval is considered more challenging than the rigid case due to the increased degree of freedom in the database models to be searched. A non-rigid approach is based on a similarity calculation between 3D shapes using a novel technique called topology matching [20]. It suggests a method for finding similarities between non-rigid polyhedral models using multi-resolution Reeb graphs, which represent functions based on the pose-invariant geodesic distances over the shape. Another topology-matching method is skeleton-based shape matching [21], in which skeleton of the volume is first created and then indexed to 3D shape databases.

Many investigations have also been trying to employ the geodesic distance of non-rigid 3D models for shape matching. Jain et al. [22] suggest a retrieval approach by comparing eigenvectors of the geodesic affinity matrices of the 3D shapes. On the other hand, in Reuter et al. [23] eigenvectors of the Laplace–Beltrami operator are used. A recent method based on geodesic distance preservation establishes non-rigid retrieval of 3D shapes from 2D sketch queries [24].

Another approach is based on the exact differences between 3D non-rigid shapes. It is a reliable solution, but because of the required direct match between shapes [25], it has high computational complexity. The prominent examples of this approach is based on Gromov–Hausdorff distances [26,27]. Deformation-based methods [28,29] tackle the same non-rigid retrieval problem by bringing the shapes to be compared into a detail-preserving canonical pose [30].

Hierarchical multi-resolution approaches [31,32] as well as bag-of-words approaches [33,34] also fit well to the task of 3D non-rigid shape retrieval.

Besides these approaches, recently, deep learning methods are emerged for fast solutions to many related applications ranging from 3D pose estimation [35] to 3D scene generation [36]. Network architectures constructed in the spirit of residual learning bring solutions to the retrieval problem as well. Xie et al. [37] introduce deep shape descriptor using a novel discriminative deep auto encoder which is insensitive to deformations. This method implements a multi-scale shape distribution and use it as the auto encoder. Afterward, they utilize the Fisher discrimination criterion on the neurons in the hidden layer, and in the final stage, these neurons are concatenated to create a shape descriptor which can be used for 3D shape retrieval and classification. The main drawback of this, and many other deep learning approaches, is the tedious training process.

2.3 Indirect shape retrieval and analysis

Shape retrieval methods discussed thus far analyze the 3D shapes based on their geometric or topological features. These features are extracted directly from the shape. In recent years, new 3D shape analysis approaches are developed. Interaction-based shape retrieval methods are based on how external agents interact with the shape of the surface. This type of retrieval approach has some benefits upon other conventional retrieval methods. The most significant advantage of this class of methods is that 3D shape functionalities can be conveniently discovered by the object's interaction with, e.g., human body and hand. The other benefit of the interaction-based retrieval is that shape retrieval is not affected by the defects of the shape.

Liu et al. [38] introduce a method where shape features are not computed directly from the shape itself. It rather uses external agents which are deformable 3D shapes. This work aims to map external models to 3D shapes correctly. Using external model's position and orientation information, Liu et al. match objects with their probabilistic information. This method has some drawbacks. Firstly, 3D shapes must be placed upright correctly and scaled according to its external models. Also, the models should be appropriate to enable accurate alignments with the predefined agents that are not allowed to be reposed.

Kim et al. [39] propose a new shape analysis method that reposes a human agent on the human-made objects in order to extract the semantic and functionality of the 3D shapes. After the learning process, when a 3D shape is used as an input, the framework searches to find the proper human pose with small energy according to affordance model [40]. They also extract contact points and kinematic parameters of the 3D shape.

Kaick et al. [41] introduce another interaction-based method that contains contextual descriptors. They aim to define the functionality of the objects in a geometric manner. The contextual descriptors here are called the interaction contexts. Normally, other works extract functionality of the shape indirectly. On the other hand, interaction contexts define functionality of the objects explicitly. Interaction context collects the geometric data between the center object and the peripheral objects. After that, it constructs a hierarchical structure to define the interaction relations between the shapes.

Another fundamental topic for 3D shape classification is object reasoning and their affordances. Instead of using shapes, names, types, or colors to categorize shapes, objects can be classified by defining which function that object is used for. For example, basketball object can be labeled as a rollable object or apple can be defined as an eatable object. According to Zhu [42], the knowledge base approach gives a new direction to the classification methods by using the functionalities of the objects. Knowledge base is a graph like structure that holds entities to define the functionality of the object. It consists of various object attributes and entities. Attributes consist of three different types; visual attributes (e.g., color), physical attributes (e.g., size, weight) and cate-

gorical attributes (e.g., cat is an animal). Affordances provide a moderate representation to represent the objects, allowing objects to be recognized even if they have never been seen before. Analysis of objects can be strengthened by establishing a link between attributes and affordances.

Bar-Aviv and Rivlin [43] introduce a novel approach to classify 3D shapes using their functional usage. It argues that the classification can be achieved through simulation of actions. In order to validate this idea, the ABSV: agent-based simulated vision approach tries to imitate the way humans perform certain classification tasks. In the ABSV method, a corresponding virtual agent is assumed to exist for any functionality. For instance, a corresponding virtual human model should exist for the seatability function. By embodying that model in the system, they classify objects as chairs by trying to make the virtual human seat on them.

3 Proposed method

We propose a supervised learning approach to analyze and retrieve 3D shapes using support vector machine classifier and their interactions with a digital hand. Our method is based on the analysis of how people are grabbing a 3D object.

In particular, we implement two different analysis tools for our grab-based retrieval task. The first tool captures the hand features using the Data Glove device. We obtain the hand data using the real-world objects. This tool is used to compare accuracy with the proposed software. Our proposed tool is implemented using the Leap Motion device, which captures the hand data using virtual objects. We estimate the performance of our method using tenfold cross-validation. Our training set consists of nine labeled objects that have different functionalities. Every object can be held by only one hand.

The organization of the algorithms implemented for this work is shown in Fig. 1 for the Leap Motion application and Fig. 2 for the Data Glove application.

In the first step of both applications, the hand descriptors and the interaction attributes are gathered from the Leap Motion controller and the Data Glove controller, respectively. Then, the set of relevant features is extracted from the data

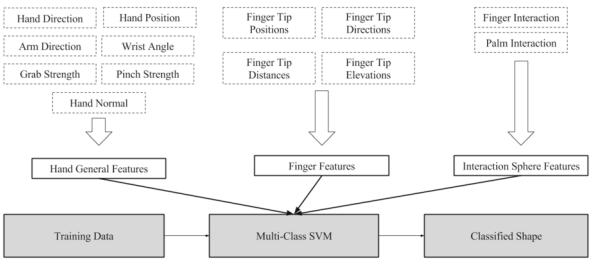

Fig. 1 Leap Motion application pipeline

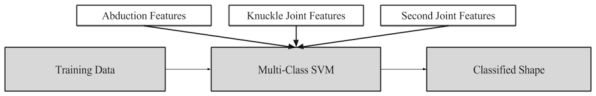

Fig. 2 Data Glove application pipeline

that are acquired by these devices. After the feature extraction process, training and test data are converted into the proper formatted file for the support vector machine (SVM). During the data collection process, these two controllers provide different types of data, and these data are collected at various times. For this reason, extracted features are used separately in the machine learning process. In the preprocessing part, data are scaled and offset. Afterward, the best SVM kernel is selected according to the data type of the feature. With grid search approach, the best kernel parameters are found for each approach. Cross-validation method using the best kernel parameters gives the best result for this dataset. Finally, a multi-class SVM is applied to the extracted features from both applications.

3.1 Features for the leap motion application

In the Leap Motion application part of this work, different types of features of the hand are used to classify 3D shapes accurately. These features are mostly extracted directly from the hand and computed by the Leap Motion API. We combine some of the API features into novel features as well as define a different feature that calculates the interaction between the hand and the virtual object to enhance the retrieval performance.

During the process of sample acquisition, we observe that most of the values that are calculated by the Leap Motion device are very accurate. However, in some situations, Leap Motion device may produce unsatisfactory results. We found out that light source angle, intensity, and type are some of the reasons for the poor results; therefore, we took the samples during daylight to prevent data distortion. Also, if the fingers are standing next to each other, Leap Motion device may not get the data of the fingers correctly. Moreover, hand orientation is another factor for capturing finger data properly. Thus, some of our virtual objects are aligned differently to capture all fingers correctly. Background objects cause another problem for obtaining the hand data. For example, if a part of the human body or real-life object enters the Leap Motion camera frame, the device may not recognize the hand appropriately. Therefore, users are asked to sit on the chair when application samples were taken, and the objects that may affect Leap Motion camera are removed from the room.

All features used in the Leap Motion application are explained in the following list of 11 items. Note that, some items represent a set of features in this list. The actual number of distinct features is indeed 119. While the first five items

are obtained directly from the API, we design and propose the other six as novel contributions of this work.

1. *Position and direction of fingertips* These features are the translation and orientation of the fingers in world coordinates. Fingertip positions are denoted as F_i for $i = \{1, 2, 3, 4, 5\}$. Fingertip directions are the unit normals of the corresponding tips.
2. *Hand and arm direction* Vectors that show hand and arm directions are unit direction vectors in world coordinates. Direction vectors are shown using D_h and D_a for hand and arm, respectively.
3. *Hand normal and center* Hand normal is a unit vector that is perpendicular to the palm plane and pointing down from the palm center in world coordinates. Hand center is an approximate position of the palm region in the world coordinates.
4. *Pinch and grab strength* Pinch strength shows how much the hand joints are close to the predefined pinch pose. Additionally, grab strength shows closeness of the hand joint values to the predefined grab pose. These values are defined between zero and one.
5. *Sphere center and radius* Sphere center value defines the center position of the imaginary sphere that is located around the palm such that it gets smaller as the pose turns into a fist. Radius refers radius of the defined sphere.
6. *Distance of fingertips* It is a distance value between all fingertips calculated as:

$$D_{ij} = \left|\left|F_i - F_j\right|\right|, \quad i, j = 1, \ldots, 5 \tag{1}$$

7. *Wrist angle* This feature is based on the angle between the normalized arm and hand direction vectors defined as:

$$\alpha = \mathrm{acos}\left(D_\mathrm{h} \cdot D_\mathrm{a}\right) \tag{2}$$

8. *Angle difference between initial and current joints* This feature uses the rotation of each finger joint in local coordinates. These joint variables are shown as J_{ij}, where $i = \{1, 2, 3, 4, 5\}$ is the finger ID and $j = \{1, 2, 3\}$ is used for the joint ID. When the application starts, initial quaternion values of the hand in reference null pose is recorded to calculate every joint angle. These joint variables are shown as I_{ij} with the same indexing as J_{ij}. To record the current sample, which is essentially the difference between the reference and the current pose, conjugated angle θ for each joint is calculated as follows, where A_0 is the angle component of the currently computed quaternion A_{ij}:

$$A_{ij} = J_{ij} \cdot \mathrm{conj}\left(I_{ij}\right) \tag{3}$$
$$A_{ij} = [A_0\ A_1\ A_2\ A_3] \tag{4}$$
$$\theta = 2 \cdot \mathrm{acos}\left(A_0\right) \tag{5}$$

9. *Interaction points* Interaction points consist of 64 small spheres that reside on the inner surface of the hand and indicate whether the hand touches the 3D shape or not. When spheres intersect the virtual object, they become active. Each sphere is attached to a part of the hand according to the position and rotation information gathered from the Leap Motion API and if that part of the hand moves, the spheres move accordingly. Sphere interaction status is recorded 60 times per second through intersection operations. To get a sample, the application calculates an average of the interaction point values in one second over all participants. During the data collection phase, interaction points might lead to inaccurate results because of the lack of feedback. To prevent inconsistent interaction points feature data, a visual feedback mechanism is implemented which shows the interaction between the virtual object and the hand (Fig. 3-right). We also show the interaction points which are created according to the average interaction data over all objects and the digital hand, green being the highest interaction and red the lowest (Fig. 3-left).
10. *Fingertip direction and distance from hand center* This feature is the unit direction from the hand center C to each fingertip as well as the corresponding distances, given as:

$$D_i = F_i - C, \quad i = 1, \ldots, 5 \tag{6}$$
$$M_i = ||D_i||, \quad i = 1, \ldots, 5 \tag{7}$$

11. *Fingertip elevation from hand center* Fingertip elevation is the angle between the hand plane normal and the vector from the hand center to each fingertip. It is calculated for one tip via:

$$u = F_i - C \tag{8}$$
$$N = (A,\ B,\ C) \tag{9}$$
$$\alpha = \frac{|A \cdot u_1 + B \cdot u_2 + C \cdot u_3|}{\sqrt{A^2 + B^2 + C^2} \cdot \sqrt{u_1^2 + u_2^2 + u_3^2}} \tag{10}$$

Note that, since all of our features, except the Interaction Points, are computed on the digital hand, our method is quite flexible in dealing with databases of 3D shapes, may it be represented as a clean manifold mesh in arbitrary topology, non-manifold mesh, polygon soup, or a point cloud. Interaction Points feature processes the hand as well as the object but still enforces no constraints on the representation of the object.

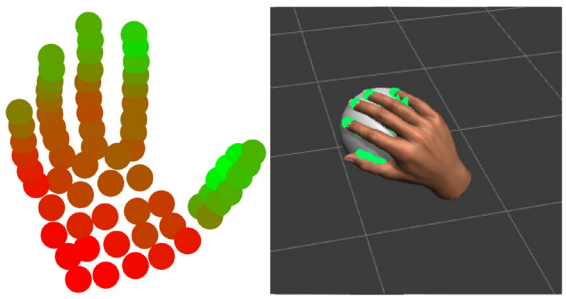

Fig. 3 Interaction points obtained by averaging the results of all the objects (left) and Leap Motion feedback component (right)

3.2 Features for the data glove application

In this application, since the users interact only with the real-world objects, only the features supported by the Data Glove is used. The Data Glove API provides hand values from 14 different points. Ten of these values are the angles of the finger joints (two per finger), and the remaining four are the angles between the five fingers. The values that are transmitted from the glove can be between zero and one, as well as between zero and 4096 [44].

3.3 Experiments

3.3.1 Object selection

Many different objects are used in the Leap Motion and Data Glove applications developed for this work. In total, nine different objects are employed. Three of them are chosen as objects with primitive shapes, and the other objects are selected as objects which are regularly used in daily life. Every object has a real and a virtual version. Real objects are employed in the Data Glove application, and virtual objects are for the Leap Motion application. Real and virtual objects are not identical, but their dimensions and shapes are the same. As seen in Figs. 4 and 5, the virtual and real versions of these objects are cup, sphere, cylinder, mouse, pencil, phone, quadrangular, scissor, and tablet.

3.3.2 Data collection

The data in our user study are obtained from user-defined static hand motions. To this effect, two different data-capturing processes are built and nine different shapes are found. The first experiment uses the real objects and is based on the Data Glove controller. In order to perform the second experiment that is based on the Leap Motion controller, these objects are created virtually using their real-world references as shown in Fig. 6.

The participants to our user study are provided with the hardware and software they needed for both experiments. In every experiment, firstly, the necessary information is given to each participant to let them perform the sampling process properly. After that introduction, their data are captured for each real-world or virtual object. Finally, this hand information is processed and used in support vector machine for object classification.

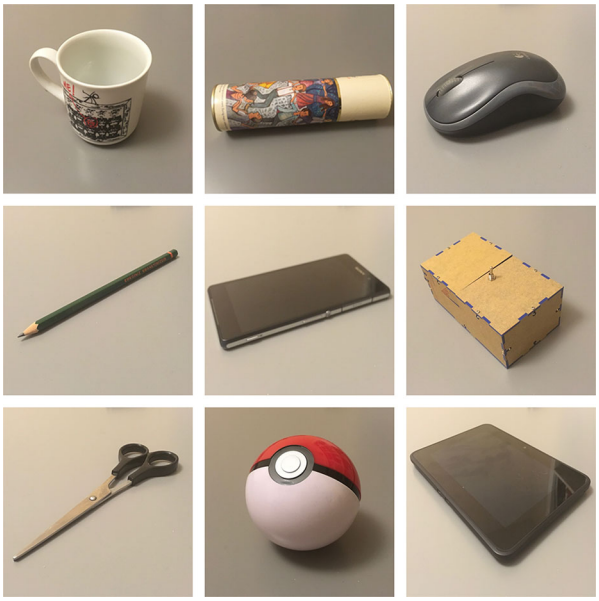

Fig. 4 Real-world objects that are used in the Data Glove application: cup (top left), cylinder (top middle), mouse (top right), pencil (middle left), phone (center), quadrangular (middle right), scissor (bottom Left), sphere (bottom middle), and tablet (bottom right)

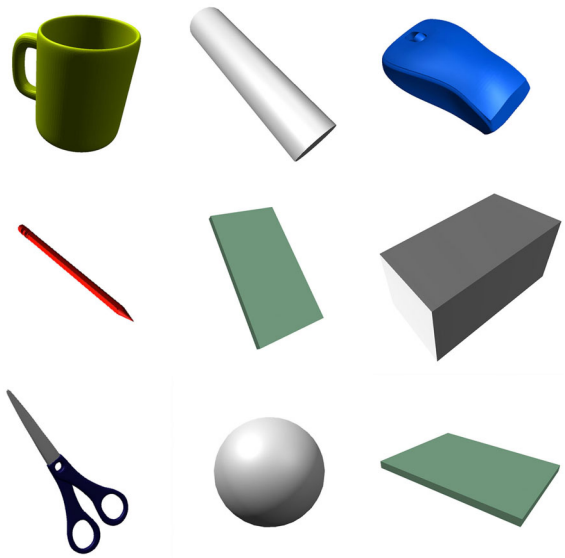

Fig. 5 Virtual objects used in the Leap Motion application: cup (top left), cylinder (top middle), mouse (top right), pencil (middle left), phone (center), quadrangular (middle right), scissor (bottom Left), sphere (bottom middle), and tablet (bottom right)

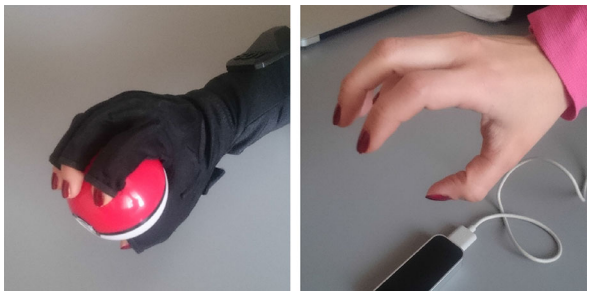

Fig. 6 Examples of capture processes using Leap Motion and Data Glove controllers

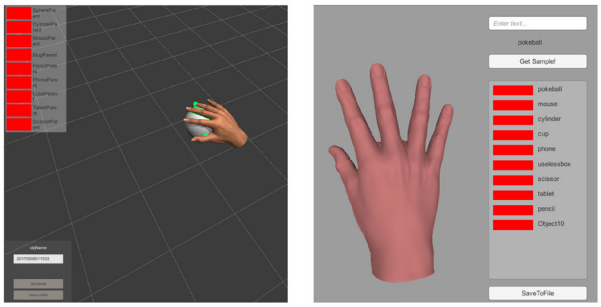

Fig. 7 Screenshots from our Leap Motion (left) and Data Glove applications (right)

3.3.3 Participants and experiment area

For this work, 26 participants are volunteered from Ankara METU Area. The volunteers are 21 male and five female participants within age range between 24 and 46, and their average age is 32. The candidates are chosen from various professions. On the other hand, to avoid technical bias, we choose most of them from computer-based jobs, such as software developers or 3D–2D graphics artists.

The participants average daily computer usage is 6 hours (the least is 4, and the most is 10). 13 participants reported that they used the Leap Motion controller before. None of the participants used data glove device before. Moreover, the other five stated that they used human–computer interaction devices before, e.g., Microsoft Kinect or Nintendo Wiimote. In the experiments, all participants used their right hands even if they are left handed.

The experiments are performed in a closed environment. The samples were taken during the daytime because Leap Motion gives better results under natural light. A relatively quiet environment is created so that users are not distracted or experiment is not interrupted by external factors.

3.3.4 Process

At the beginning of the hand data collection process, we gave a tutorial regarding how to use both of these applications. After that process, we show the 3D virtual or real-world shapes that are used in both experiments. It is the vital issue that example of how the users should hold the objects is not given. On the other hand, for functional objects only, users are informed that they should grab items as if they are using the objects.

When a user starts any of our two applications, namely the Data Glove and Leap Motion applications, his/her name is first entered through the graphical user interface. Subsequently, the appropriate application is opened according to the controller type (Fig. 7). When the user holds the object, the hand information is saved with the help of the submit button in the application. If the hand information is successfully saved, the status field corresponding to that object turns green, and this process continues for each object. Each saved user file contains information about which objects are stored in the saved file along with the corresponding hand information, and how many samples are taken from the participant. After the samples are taken from all users, they are exported in a suitable file format. If the feature is a scalar value, then it is inserted as a single entry, and if it is a vector value, the value is added as a separate feature for each dimension.

3.4 Shape retrieval

In order to apply multi-class support vector machine (SVM), all vectors are classified according to their corresponding 3D shapes. To obtain the classification results, first, SVM calculates the outcome of every 3D shape. In other words, if we have N different objects that will be used in SVM, then $N(N-1)/2$ binary SVMs are used to find a result for each 3D shape pair. Each result of these SVMs is used as a point of a certain shape, and the object that has the maximum number of points is selected as the output of the classification.

In this work, nonlinear radial basis function (RBF) is used to train the feature vectors in SVM. To find the best RBF variables, the grid-based search method is applied for the Leap Motion and Data Glove applications. For every RBF parameters (C, γ), a range of values is selected, and a basic grid is created for every (C, γ) pair. Then, SVM with RBF is applied with these values repeatedly until the best results are found. Furthermore, to find the best results, tenfold cross-validation approach is used for both application features. In this cross-validation, the original data is randomly divided into tenfold. Each fold contains prelabeled data special to the application at hand. We then train the model using every fold except for one, which we call the test fold. We essentially test to see how well the model is doing on this test fold. We record the number representing its performance and repeat the whole process with a new test fold and nine other training folds. In the end we have 10 performance numbers which we average into a single estimation with the optimal RBF variables.

4 Results and discussion

We examine hand–object interactions from two different viewpoints: via Data Glove (Sect. 4.1) and Leap Motion (Sect. 4.2) and promote the results of the latter as our proposed solution. We also provide a comparison with a direct analysis method which is not interaction-based at all (Sect. 4.3).

In the Data Glove solution, the real-world objects are captured using the Data Glove controller, and the samples are collected with the aid of the software which is implemented for this work. In this stage, hand–object interaction samples which are captured from Data Glove application are analyzed with the help of the support vector machine (SVM), and eventually these objects are categorized with the result.

In the Leap Motion solution, the samples are taken from the users by providing the objects in the virtual environment with the help of Leap Motion controller. The objects used in this phase have the same dimensions and shapes as the objects utilized in the Data Glove part. Also, SVM is used in the same way as that part to categorize the shapes.

In the sequel, the analysis and categorization results computed by these solutions are shown, and a detailed comparison is made between their results. The accuracy results found in this section are computed by the ratio of the true positive predictions to all predictions.

4.1 Data glove results

To measure the performance of the approach that is used in Data Glove controller application, we capture a dataset of the hand–object interaction information. The Data Glove dataset contains data of 9 different real-world objects which can be seen in Fig. 4. This data are captured from 20 different people. Each object is captured twice for every participant, and $9 \times 20 \times 2 = 360$ samples are obtained in total.

The features obtained from the Data Glove controller are very few due to the limited capabilities of the device. For this reason, only 14 different features are extracted from the samples obtained from the Data Glove. These features are the normalized joint angles between the Knuckle joints and the Second joints for every finger, hence a total of $5+5 = 10$ features. Also, the horizontal angle between the fingers, 4 in total, are used as features in this phase, and these features are called as Abduction angles.

Table 1 shows the results obtained from the Data Glove application using the classification algorithm in Sect. 3. Each row of the confusion matrix shown represents the number of instances in the actual class while each column represents the number of instances in the predicted class. This matrix demonstrates that there is no significant high rate of false-positive values.

Table 1 Confusion matrix for Data Glove when all 14 features are in use

	O1	O2	O3	O4	O5	O6	O7	O8	O9
O1: Sphere	27	0	3	0	0	5	4	0	1
O2: Mouse	0	32	0	0	4	0	3	0	1
O3: Cylinder	2	0	34	0	0	2	1	0	1
O4: Cup	0	0	1	36	0	0	0	0	3
O5: Phone	0	0	0	0	31	0	2	4	3
O6: Cube	4	0	2	0	2	27	4	1	0
O7: Scissor	1	1	1	1	1	2	25	2	6
O8: Tablet	0	0	0	0	2	1	2	32	3
O9: Pencil	0	1	0	0	1	0	4	5	29

Table 2 Performance of Data Glove when different sets of features are in use

Feature set	Accuracy (%)	Train + test time (s)
Abduction	60.55	6.12
Knuckle	46.94	6.21
Second	42.50	6.49
Abduction + knuckle	65.83	7.79
Knuckle + second	61.94	5.92
Abduction + second	66.94	6.25
Abduction + second + knuckle	75.83	7.82

Last row shows the case where all the sets, hence all 14 features, are in use

When all features are used, the retrieval accuracy is about 75%, which shows that majority of the shapes are recognized (last row in Table 2). Also, we have obtained an adequate result of 0.72 kappa statistics. If the features are narrowed down to the only abduction angles, the accuracy decreases to 60.55%. When only knuckle joint features are used in the SVM to categorize the shapes, the performance decreases dramatically to as low as 46.94%. These results show that the objects cannot be separated using only these features. Additionally, the confusion matrix of the second joints gives similar results with the knuckle joints with 42.5% accuracy.

An interesting observation is that sets of features capture different properties of the hand, and by combining them together, it is possible to improve the retrieval accuracy, e.g., by combining knuckle, second, and abduction feature sets, an accuracy of about 65% can be reached. Results obtained based on different combinations of feature sets are shown in Table 2 along with the Train + test time. Note that, once the model is set after training and testing, queries are responded instantly.

Table 3 Confusion matrix for Leap Motion when all 119 features are in use

	O1	O2	O3	O4	O5	O6	O7	O8	O9
O1: Sphere	49	1	0	0	0	2	0	0	0
O2: Mouse	0	50	2	0	0	0	0	0	0
O3: Cylinder	1	1	50	0	0	0	0	0	0
O4: Cup	0	0	0	48	0	0	1	1	2
O5: Phone	0	0	0	3	23	0	0	26	0
O6: Cube	3	1	0	0	0	47	0	1	0
O7: Scissor	0	0	0	1	0	1	50	0	0
O8: Tablet	0	0	0	0	32	0	0	20	0
O9: Pencil	0	0	0	2	1	0	0	0	49

Table 4 Performance of Leap Motion when different sets of features are in use

Feature set	Accuracy (%)	Train + test time (s)
General	74.14	14.14
Fingers	75.35	26.8
Interaction points	60.00	29.81
General + fingers	78.98	25.6
General + interaction points	73.53	32.81
Fingers + interaction points	77.97	40.5
General + fingers + interaction points	80.00	68.24

Last row shows the case where all the sets, hence all 119 features, are in use

4.2 Leap motion results

The dataset for Leap Motion tests consists of information about nine different virtual 3D shapes which are shown in Fig. 5. Our tests are performed on 26 different people. Each object is captured two times for every people, and $9 \times 26 \times 2 = 468$ samples are collected in total.

The features that are captured from the Leap Motion controller are very distinct thanks to controller's flexible interface. 119 different features are extracted from the samples obtained from the Leap Motion controller and these features are grouped into three main categories as General features, Finger features, and Interaction Points features. The former constitutes non-finger features listed as items 2, 3, 4, 5, and 7 in Sect. 3.1, and the last one represents the set of 64 interaction features (item 9). Items 1, 6, 8, 10, and 11 make up the Fingers features.

The confusion matrix in Table 3 shows the results obtained by using all the features extracted from our Leap Motion application. All the features revealed a result of 80% correctness (last row in Table 4), which in general shows the correct classification of the objects. A value of 0.92 kappa statistics also verifies that the results are satisfactory. There is, however, a misclassification between the phone and the tablet objects. These two objects cannot be adequately classified by SVM, because they are very similar to each other in the form of grabbing. If these two objects are assumed to be a single class, a result of more than ninety percent can be obtained. Also, during the data collection process, users have already confused these objects due to their visual similarities in the absence of texture mapping (see Fig. 5), which made them treat the objects wrongly, e.g., used the tablet as a phone. This led to inaccurate interaction data and explains the corresponding low scores.

A few cube samples in Table 3 are incorrectly classified as sphere objects, despite the high accuracy in general. The main reason of this misclassification is the similarity of the Interaction Points. As seen in Fig. 8, there is a high interaction value in the distal and middle phalanx areas of the fingers, and medium-sized interactions can be seen in the areas where metacarpal and proximal phalanx bones join.

The results obtained by using only the General features, instead of using all the features, also contains frequent incorrect recognitions between phone and tablet objects. An overall accuracy rate of 74% is obtained with General features only.

4.2.1 Results based on interaction points

In order to emphasize the promoted solution of combined features, which comes with 80% accuracy, we in this section focus on the results based on a single feature set, namely the Interaction Points.

Data obtained using Interaction Points only are prone to confusions and results in an accuracy of 60% and a value of 0.55 kappa statistics. The mouse object, for instance, has high false-positive values due to its eight times miscategorizations as cup objects. As demonstrated in Fig. 8—top left and top right for these two objects, there is interaction in thumb, index, and middle fingers at high quantity, and also there is interaction at the points where metacarpals intersect with these three fingers.

The cylinder object presents a high correct categorization rate, showing 46 of the 52 samples as true positives, which is still inferior to the 50/52 result of our promoted combined solution in Table 3—row 3. This object has a high relatively percentage of interaction on the distal phalanx bones (Fig. 8—top middle).

The cube object, on the other hand, has a high rate of misclassifications as pencil and scissors objects. Thumb, index, ring and middle fingers play a major role in the hand interaction of the cube object (Fig. 8—middle right), which is similar to the pencil and scissor objects (Fig. 8—middle left and bottom left). Notice also the lack of interaction with the metacarpals (palm and finger connections) common to both these three objects.

Fig. 8 Hand–object interaction points for every object. Layout is consistent with that of Fig. 4

When looking at the accuracy rates of phone and tablet shapes, there is a large number of false-positive results as in the other Leap Motion analyses. As shown in Fig. 8—center and bottom right, the thumb and index fingers show a high level of interaction on these two objects and the same rate of interaction does not appear in the other fingers and the palm. 24 of the 52 samples in the phone object are consequently miscategorized as tablets, and 19 of the 52 tablet objects are misguessed as phones. As in the other feature sets, combined or alone, the Interaction Points feature set cannot distinguish the difference between the phones and the tablets.

When all of the Interaction Points for each shape are examined, we observe in general that the objects do not interact with the parts of the metacarpal near the wrist, and therefore these areas do not contribute to the classification at all. During the interaction, phalanx bone regions play a major role in classification. Besides, the thumb finger can be seen as the common part in every interaction.

4.2.2 Results based on other feature combinations

Similar to Sect. 4.2.1, we now examine the results when a pair of feature sets is used. We observe in this scenario that accuracy slightly increases (Table 4). With a combination of feature sets of General and Fingers, the accuracy increases to 78.98%, and the kappa value reaches to 0.76 which is a strong value. In these results, an unexpected false positive value is not encountered except for the tablet and phone objects. Likewise, General and Interaction Points feature set combination has a good result with 73.53%. In this combination, in addition to the tablets and phones, it can be seen that the spheres and cube objects are not satisfactorily classified. Fingers and Interaction Points feature set pair shows a result that is similar to the combination of General and Fingers with a result of 77.97%. In these results, there is an unexpected false-positive result of classification between the pencil and the cup.

Another feature set combination is achieved via the leave-one-out method with the results in Table 5. When compared to the tenfold cross-validation experiment (Table 3), there are no significant true positive value changes in the confusion matrices. Furthermore, the confusion matrix of this experiment shows that the distribution of the false-positive values is similar to the tenfold cross-validation experiment. Using all the features sets found on Leap Motion experiment, a similar accuracy result of 80.17% is achieved compared to our proposed tenfold cross-validation technique. Despite the similar performance in terms of accuracy, leave-one-out method is about 12 times slower than our tenfold cross-validation solution as it divides the data into much more folds of much smaller sizes during training and testing.

In summary, using different feature variety in the promoted Leap Motion application and the competitor Data Glove application has led to different accuracy results with a five percent difference favoring our system. In the Data Glove application, we can only access the angles between finger joints, on the other hand, Leap Motion API gives positional information of the hand and fingers alongside the joint information. Therefore, we can extract new features using the Leap Motion data, e.g., the Interaction Points. Also, in the data collection phase with Data Glove, we noticed that the collected device output is not precise compared to Leap Motion. Leap Motion application, however, demonstrates unsuccessful results in phone and tablet classification in contrast to the Data Glove application. Note finally that, although queries are responded instantly in both applications, our promoted solution takes more training and testing time due to the larger number of features in use. The difference is linear in the number of features, i.e., Data Glove with 14 features is about 8.5 times faster than Leap Motion with 119 features.

4.3 Direct retrieval versus indirect retrieval

In addition to the interaction-based classification experiments, the success of our approach is also verified when compared to a direct shape retrieval method based on 3D shape distribution histograms [45]. Measurement of the distance between a fixed point and random points on the surface is used in this work as the histogram descriptor. As seen in Fig. 9, if the objects have different geometric forms, shape histograms can be used for classification. However, there is no satisfaction in classifying objects that are similar in geometry but have different functionalities and classes. Shape histograms show that cylinder objects give similar

Table 5 Confusion Matrix for Leap Motion when all 119 features are in use under leave-one-out cross-validation

	O1	O2	O3	O4	O5	O6	O7	O8	O9
O1: Sphere	48	1	0	0	0	2	0	1	0
O2: Mouse	0	51	1	0	0	0	0	0	0
O3: Cylinder	4	0	48	0	0	0	0	0	0
O4: Cup	0	0	0	46	1	0	2	1	2
O5: Phone	0	0	0	2	28	1	0	21	0
O6: Cube	3	0	0	0	1	48	0	0	0
O7: Scissor	0	0	0	0	0	1	51	0	0
O8: Tablet	0	0	0	0	25	0	0	27	0
O9: Pencil	0	0	0	2	1	0	0	0	49

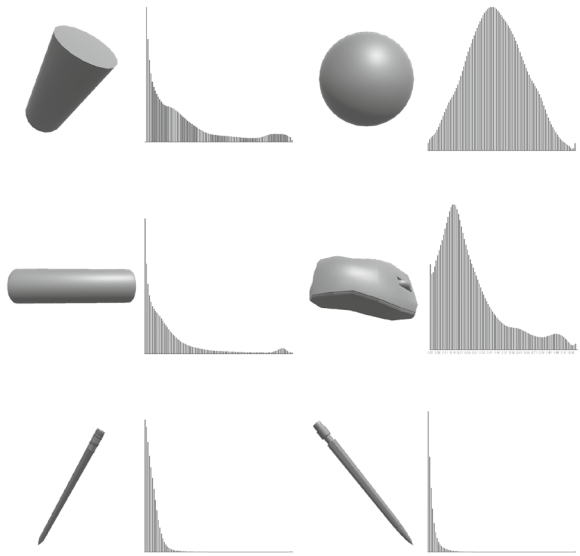

Fig. 9 D1 shape distributions of six shapes. In each plot, the horizontal axis shows the normalized distance, and the vertical axis represents the probability of that distance being between two shape points

results with the pencil shapes. On the other hand, in Leap Motion application, we get a successful accuracy without a false-positive result in the classification between pencil and cylinder (Table 3). Note that cylinder can be replaced with many other practical objects in similar geometric forms but different functionalities, such as batteries, toiler paper rolls, aerosol cans, and candles. With this work, we can see that such objects with similar geometries but different functionalities can be successfully classified according to how they are grabbed.

5 Limitations

Although this work is algorithmically complete and has achieved its purpose, there is room for improvement in the experimentation part. Firstly, for each object class, one type of each object was used during the experiment. A more comprehensive evaluation could be carried out by populating the object set with various versions of these models. Secondly, Data Glove experiments could be repeated with virtual objects to be fully compatible with the Leap Motion tests. Finally, the number of participants could be increased for more generic results.

6 Conclusion and future work

In this work, we developed a novel 3D rigid shape retrieval algorithm based on indirect analysis paradigm. In contrast to body–object interactions popular in the indirect analysis approaches, we utilize a novel hand–object analysis framework which presents difficulties as well as opportunities that are specific to this new problem. Our approach is also fundamentally different from yet another popular technique, retrieval by direct analysis. We in the end show that our method successfully predicts objects groups with 80% accuracy. Our algorithm coupled with a cheap device like Leap Motion achieves more accurate results than a comparative algorithm run on the expensive Data Glove equipment. Experiments show that our indirect approach to the retrieval problem distinguishes certain objects classes much better than a comparative direct approach.

Our algorithm works on existing feature descriptors and our novel feature descriptors obtained using the information of how to grab 3D objects in the correct way. These features define which parts of the 3D shapes and the digital hand model interact with each other. Fed into the support vector machines, our features produced promising retrieval results in our learning-based framework. These features do not constraint the representation of the input 3D database models in any way.

Our results can serve as a guide to describe how hand parts are involved in the grabbing action. In particular, our results show which parts are the most and least important and demonstrate user habits while grabbing an object. These findings extend well to future work, such as time-varying analysis where the motion of the hand is also important, e.g., in robotics. Yet another fruitful research direction can be the addition of the second hand to the process which should increase the recognition scope. We finally point out the possibility of replacing the user control on our digital hand agent with a fully automatic control through artificial intelligence.

Acknowledgements This work has been supported by TUBITAK under the project EEEAG-115E471.

References

1. Funkhouser, T., Min, P., Kazhdan, M., Chen, J., Halderman, A., Dobkin, D., Jacobs, D.: A search engine for 3D models. ACM Trans. Gr. **22**(1), 83–105 (2003)
2. Passalis, G., Theoharis, T., Kakadiaris, I.: Ptk: a novel depth buffer-based shape descriptor for three-dimensional object retrieval. Vis. Comput. **23**, 5–14 (2007)
3. Tangelder, J., Veltkamp, R.: A survey of content based 3D shape retrieval methods. Multimed. Tools Appl. **39**(3), 441–471 (2008)
4. Osada, R., Funkhouser, T., Chazelle, B., Dobkin, D.: Shape distributions. ACM Trans. Gr. **21**(4), 807–832 (2002)
5. Paquet, E., Rioux, M., Murching, A., Naveen, T.: Description of shape information for 2-D and 3-D objects. Signal Process. Image Commun. **16**(1–2), 103–122 (2000)
6. Zheng, Y., Neo, Y., Chua, T., Tian, Q.: Toward a higher-level visual representation for object-based image retrieval. Vis. Comput. **25**, 13–23 (2009)
7. Novotni, M., Klein, R.: Shape retrieval using 3D Zernike descriptors. Comput. Aided Des. **36**(11), 1047–1062 (2004)
8. Canterakis, N.: 3D Zernike moments and Zernike affine invariants for 3D image analysis and recognition. In: 11th Scandinavian Conference on Image Analysis, pp. 85–93 (1999)
9. Chen, D., Tian, X., Shen, Y., Ouhyoung, M.: On visual similarity based 3D model retrieval. Comput. Graph. Forum **22**(3), 223–232 (2003)
10. Shilane, P., Funkhouser, T.: Selecting distinctive 3D shape descriptors for similarity retrieval. In: Proceedings of IEEE International Conference on Shape Modeling and Applications, SMI 2006, p. 18 (2006)
11. Wang, P., Liu, Y., Guo, Y., Sun, C., Tong, X.: O-CNN: octree-based convolutional neural networks for 3D shape analysis. ACM Trans. Gr. **36**(4), 1–11 (2017)
12. Gao, Y., Yang, Y., Dai, Q., Zhang, N.: Representative views re-ranking for 3d model retrieval with multi-bipartite graph reinforcement model. In: Proceedings of the 18th ACM International Conference on Multimedia, pp. 947–950 (2010)
13. Gao, Y., Wang, M., Shen, J., Dai, Q., Zhang, N.: Intelligent query: open another door to 3d object retrieval. In: Proceedings of the 18th ACM International Conference on Multimedia, ACM, pp. 1711–1714 (2010)
14. Bai, S., Bai, X., Zhou, Z., Zhang, Z., Tian, Q., Latecki, L.: GIFT: towards scalable 3D shape retrieval. IEEE Trans. Multimed. **19**(6), 1257–1271 (2017)
15. Gao, Y., Wang, M., Tao, D., Ji, R., Dai, Q.: 3-D object retrieval and recognition with hypergraph analysis. IEEE Trans. Image Process. **21**(9), 4290–4303 (2012)
16. Gao, Y., Tang, J., Hong, R., Yan, S., Dai, Q., Zhang, N., Chua, T.: Camera constraint-free view-based 3-D object retrieval. IEEE Trans. Image Process. **21**(4), 2269–2281 (2012)
17. Leifman, G., Meir, R., Tal, A.: Semantic-oriented 3d shape retrieval using relevance feedback. Vis. Comput. **21**, 865–875 (2005)
18. Lian, Z., Godil, A., Bustos, B., Daoudi, M., Hermans, J., Kawamura, S., Kurita, Y., Lavoué, G., Nguyen, H., Ohbuchi, R., Ohkita, Y., Ohishi, Y., Porikli, F., Reuter, M., Sipiran, I., Smeets, D., Suetens, P., Tabia, H., Vandermeulen, D.: A comparison of methods for non-rigid 3D shape retrieval. Pattern Recognit. **46**(1), 449–461 (2013)
19. Sipiran, I., Meruane, R., Bustos, B., Schreck, T., Li, B., Lu, Y., Johan, H.: A benchmark of simulated range images for partial shape retrieval. Vis. Comput. **30**(11), 1293–1308 (2014)
20. Hilaga, M., Shinagawa, Y.: Topology matching for fully automatic similarity estimation of 3D shapes. In: Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques, pp. 203–212 (2001)
21. Sundar, H., Silver, D., Gagvani, N., Dickinson, S.: Skeleton based shape matching and retrieval. In: Proceedings of SMI 2003: Shape Modeling International, pp. 130–139 (2003)
22. Jain, V., Zhang, H.: A spectral approach to shape-based retrieval of articulated 3D models. CAD Comput. Aided Des. **39**(5), 398–407 (2007)
23. Reuter, M., Wolter, F., Peinecke, N.: Laplace-spectra as fingerprints for shape matching. In: SPM '05 Proceedings of the 2005 ACM Symposium on Solid and Physical Modeling, vol. 1, no. 212, pp. 101–106 (2005)
24. Sahillioğlu, Y., Sezgin, M.: Sketch-based articulated 3d shape retrieval. IEEE Comput. Gr. Appl. **37**(6), 88–101 (2017)
25. Sahillioğlu, Y., Yemez, Y.: Minimum-distortion isometric shape correspondence using em algorithm. IEEE Trans. Pattern Anal. Mach. Intell. **34**(11), 2203–2215 (2012)
26. Memoli, F., Sapiro, G.: A theoretical and computational framework for isometry invariant recognition of point cloud data. Found. Comput. Math. **5**(3), 313–347 (2005)
27. Bronstein, A.M., Bronstein, M.M., Guibas, L.J., Ovsjanikov, M.: Shape google: geometric words and expressions for invariant shape retrieval. ACM Trans. Gr. **30**(1), 1 (2011)
28. Sahillioğlu, Y.: A shape deformation algorithm for constrained multidimensional scaling. Comput. Gr. **53**, 156–165 (2015)
29. Sahillioğlu, Y., Kavan, L.: Detail-preserving mesh unfolding for non-rigid shape retrieval. ACM Trans. Gr. **35**(3), 27 (2016)
30. Pickup, D., Liu, J., Sun, X., Rosin, P., Martin, R., Cheng, Z., Lian, Z., Nie, S., Jin, L., Shami, G., Sahillioğlu, Y., Kavan, L.: An evaluation of canonical forms for non-rigid 3d shape retrieval. Gr. Models **97**, 17–29 (2018)
31. Hu, J., Hua, J.: Salient spectral geometric features for shape matching and retrieval. Vis. Comput. **25**, 667–675 (2009)
32. Li, C., Hamza, A.: A multiresolution descriptor for deformable 3d shape retrieval. Vis. Comput. **29**, 513–524 (2013)
33. Toldo, R., Castellani, U., Fusiello, A.: The bag of words approach for retrieval and categorization of 3d objects. Vis. Comput. **26**, 1257–1268 (2010)
34. Lavoué, G.: Combination of bag-of-words descriptors for robust partial shape retrieval. Vis. Comput. **28**, 931–942 (2012)
35. Fan, Q., Shen, X., Hu, Y.: Detail-preserved real-time hand motion regression from depth. Vis. Comput. **34**, 1145–1154 (2018)
36. Abbasi, A., Kalkan, S., Sahillioğlu, Y.: Deep 3d semantic scene extrapolation. Vis. Comput. (2018). https://doi.org/10.1007/s00371-018-1586-7
37. Xie, J., Fang, Y., Zhu, F., Wong, E.: Deepshape: deep learned shape descriptor for 3D shape matching and retrieval. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 07, pp. 1275–1283 (2015)
38. Liu, Z., Xie, C., Bu, S., Wang, X., Han, J., Lin, H., Zhang, H.: Indirect shape analysis for 3D shape retrieval. Comput. Gr. **46**, 110–116 (2015)
39. Kim, V.G., Chaudhuri, S., Guibas, L., Funkhouser, T.: Shape2pose: human-centric shape analysis. ACM Trans. Gr. **33**(4), 120 (2014)
40. Gibson, J.: The theory of affordances. In: Perceiving Acting, and Knowing, pp. 127–142 (1977)
41. Hu, R., Zhu, C., van Kaick, O., Liu, L., Shamir, A., Zhang, H.: Interaction context (icon): towards a geometric functionality descriptor. ACM Trans. Gr. **34**(4), 83:1–83:12 (2015)
42. Zhu, Y., Fathi, A., Fei-Fei, L.: Reasoning about object affordances in a knowledge base representation. In: Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 8690 LNCS, pp. 408–424 (2014)
43. Bar-Aviv, E., Rivlin, E.: Functional 3D object classification using simulation of embodied agent. In: Proceedings of the British Machine Vision Conference, pp. 32.1–32.10 (2006)
44. 5DT: 5DT Data Glove Ultra Series User Manual

45. Osada, R., Funkhouser, T., Chazelle, B., Dobkin, D.: Matching 3D models with shape distributions. In: International Conference on Shape Modeling and Applications, pp. 154–166 (2001)

Erdem Can Irmak obtained his bachelor's degree from the Computer Engineering, Middle East Technical University in 2011. He graduated from Game Technologies master program in Middle East Technical University under the supervision of Yusuf Sahillioğlu in 2017. His research is centered on 3D shape classification based on interaction hand–object interaction.

Yusuf Sahillioğlu is an Associate Professor of Computer Engineering at Middle East Technical University, Turkey. His research interests are digital geometry processing and computer graphics. Please see http://www.ceng.metu.edu.tr/~ys for details.