Intelligent Service Robotics (2021) 14:329–344
https://doi.org/10.1007/s11370-021-00349-8

ORIGINAL RESEARCH PAPER

Investigating the importance of shape features, color constancy, color spaces, and similarity measures in open-ended 3D object recognition

S. Hamidreza Kasaei[1] · Maryam Ghorbani[2] · Jits Schilperoort[1] · Wessel van der Rest[1]

Received: 13 April 2020 / Accepted: 5 January 2021 / Published online: 12 February 2021

Abstract

Despite the recent success of state-of-the-art 3D object recognition approaches, service robots still frequently fail to recognize many objects in real human-centric environments. For these robots, object recognition is a challenging task due to the high demand for accurate and real-time response under changing and unpredictable environmental conditions. Most of the recent approaches use either the shape information only and ignore the role of color information or vice versa. Furthermore, they mainly utilize the L_n Minkowski family functions to measure the similarity of two object views, while there are various distance measures that are applicable to compare two object views. In this paper, we explore the importance of shape information, color constancy, color spaces, and various similarity measures in open-ended 3D object recognition. Toward this goal, we extensively evaluate the performance of object recognition approaches in three different configurations, including *color-only*, *shape-only*, and *combinations of color and shape*, in both offline and online settings. Experimental results concerning scalability, memory usage, and object recognition performance show that all of the *combinations of color and shape* yield significant improvements over the *shape-only* and *color-only* approaches. The underlying reason is that color information is an important feature to distinguish objects that have very similar geometric properties with different colors and vice versa. Moreover, by combining color and shape information, we demonstrate that the robot can learn new object categories from very few training examples in a real-world setting.

Keywords 3D object recognition · Open-ended learning · Robotics · Lifelong robot learning

1 Introduction

One of the primary goals in service robotics is to develop perception capabilities that will allow robots to interact with the environment robustly. Toward this goal, a robot must be able to recognize a large set of object categories accurately. Furthermore, in order to interact with human users, this process of object recognition cannot take more than a fraction of a second. In human-centric environments, the robot may frequently face a new object that visually can be either very similar or not similar to other categories. For example, consider apples and oranges categories: *what is the difference between apples and oranges?* They both fall within the class of fruits, both are edible, have a similar spherical shape, and grow on trees. Although object recognition is a typical task that is performed intuitively by human cognition, it can be quite complex when a robot has to do it.

A 3D object recognition system is composed of several software modules such as *Object Detection*, *Object Representation*, *Object Recognition*, and *Perceptual Memory*. Object detection is responsible for detecting all objects in a scene. Object representation is concerned with the calculation of a set of features for the given object. The obtained representation is then sent to the object recognition module. The target object is finally recognized by comparing its representation against all the descriptions of known objects (stored in the perceptual memory). As you can see, object representation plays a prominent role because the output of this module is used for learning as well as recognition. Moreover, the representation of an object should contain sufficient information to be able to recognize the same or similar objects seen from different perspectives. Therefore, several

✉ S. Hamidreza Kasaei
hamidreza.kasaei@rug.nl

[1] Department of Artificial Intelligence, University of Groningen, PO Box 407, 9700 AK Groningen, The Netherlands

[2] Department of Technology and Engineering, Shahrekord University, Shahrekord, Iran

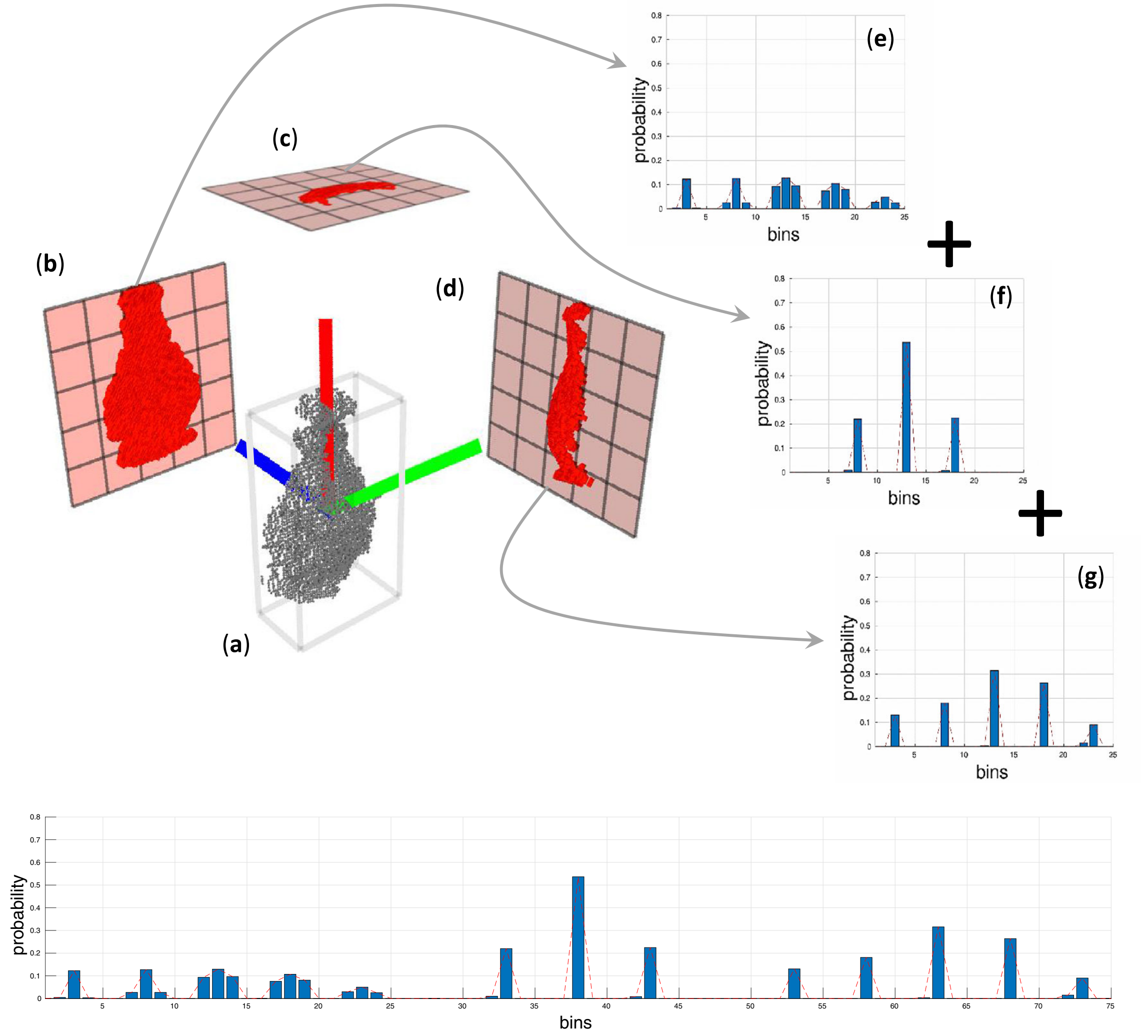

Fig. 1 Step by step visualization of the process in the GOOD object descriptor, creating a representation of a "*Vase*" object. **a** 3D point cloud of the object, its bounding box, reference frame, and three projected views; **b**–**d** three projection planes created from these views, with a *number of bins* of 5. The projections are then converted into histograms by counting the number of points falling in each bin, as shown in **e**–**g**; Finally, the GOOD object representation is created by concatenating the three histograms as visualized in **h**

important questions should be taken into account when representing an object: *which perceptual data should be used? How to represent it to the robot? Which senses would a person use to classify highly similar objects?* Arguably, we can confidently state that vision would be the most important sense, while other senses such as touch could be used for this task.

Going from this, we still do not have a definite answer on what the difference is between apples and oranges. An apple can be colored orange, while a green-colored orange could also be considered an orange. The same mutual relation goes for their shape. Taking this in mind, describing objects only by either shape or color will likely lead to confusion eventually. In this work, we assume that an object has already been segmented from a scene. The extracted point cloud of the object, containing RGB and depth data, is used to describe the shape and color of the object for distinguishing objects that have a very similar shape with a different color or vice versa. Toward this goal, we extend the Global Orthographic Object Descriptor (GOOD) [13] by adding color constancy information as an aid to improve object recognition performance. GOOD is a lightweight object descriptor that creates a convenient object representation directly from a 3D point cloud. As 3D data contains more structural information about objects, it is more robust than RGB data to the effects of illumination and shadows [22]. The required steps leading to the eventual GOOD object representation for a vase object are shown in Fig. 1. In summary, this paper contains the following main contributions:

- To develop a 3D object descriptor that represents both shape and color constancy information for a given object.
- To extensively evaluate the role of shape features, color constancy, color spaces, and similarity measures in open-ended 3D object recognition.

The remainder of this paper is organized as follows. In Sect. 2, we briefly discuss related works. The methodology for computing the object descriptor is presented in Sect. 3. Evaluation of the proposed descriptor is presented in Sect. 4. Finally, in Sect. 5, conclusions are presented, and future research is discussed.

2 Related work

Three-dimensional object recognition has been under investigation for a long time in various research fields, such as pattern recognition, computer graphics, and robotics [4,14,20,28]. Although an exhaustive survey of 3D object descriptors is beyond the scope of this paper [2,6,26], we will review the main efforts.

Object representations based on just RGB data are sensitive to illuminations and shadows. Moreover, they cannot provide accurate representations of the shape of objects. To cope with aforementioned limitations, 3D data can be used to facilitate the representation of objects. Existing 3D object representation approaches are based on either global or local descriptors. As the name suggests, global descriptors represent the complete object. In contrast, local descriptors encode an object in a piece-wise manner, representing small patches of the object around specific key points, e.g., [18]. Generally, global descriptors are increasingly used in the context of 3D object recognition, object manipulation, as well as geometric categorization. These must be efficient in terms of computation time as well as the memory, to facilitate real-time performance. Some descriptors use a reference frame (RF) to compute a pose-invariant description. Therefore, this property can be used to categorize 3D shape descriptors into three categories, including (i) shape descriptors without a common reference; (ii) shape descriptors computed relative to a reference axis; (iii) shape descriptors computed relative to an RF.

Most of the shape descriptors of the first category use certain statistic features or geometric properties of the points on the surface like depth value, curvature, and surface normals to generate a description. For instance, W. Wohlkinger and M. Vincze [29] introduced a global shape descriptor called Ensemble of Shape Functions (ESF) that does not require the use of normals to describe the object. The characteristic properties of an object are represented using an ensemble of ten 64-bin histograms of angle, point distance, and area shape functions. ESF completely ignores the potential role of color information.

In contrast, the descriptors in the second and third category encode the spatial information of an object's points using an RF. In the second category, Viewpoint Feature Histogram (VFH) [25] is a well-known descriptor. It is based on another set of descriptors, the point feature histogram (PFH) [27], more specifically the fast point feature histogram (FPFH) [24]. The histogram of a PFH results from considering several angular features between the normals of pairs on the point cloud. What VFH adds to FPFH is the consideration of a viewpoint component. The direction from the viewpoint to the centroid of the object is translated to all points. The angle between this and the normal of the points constitutes the first component of the histogram. The other components of the histograms are similar to FPFH, but the pan, tilt, and yaw angles are now computed between the normals of the points and the viewpoint direction of the centroid. In the third category, we have the Global Orthographic Object Descriptor (GOOD) [13], which performs a principal component analysis on the point cloud of an object to make an unambiguous reference frame for the object. The resulting RF is then used to create three orthogonal projection of the object with respect to the X, Y, and Z axes. Each of these projections is then converted into a histogram and then combined using two statistical features, in particular the entropy and variance, to provide the final descriptor of the object. The Globally Aligned Spatial Distribution (GASD) [17] and VFH-Color [30] also fall into the third category. GASD explores the idea of forming an object descriptor containing both color and shape information. GASD represents the shape information, almost similar to the GOOD descriptor. VFH-Color combines the original VFH descriptor with the color quantization histogram. Both GASD and VFH-Color descriptors incorporate color information into the descriptor to increase their discriminative power. We refer the reader to two comprehensive surveys on local feature descriptors [5,16]. In this paper, we select one descriptor from each category, including ESF, VFH, and GOOD.

In recent studies on object recognition and grasping, much attention has been given to deep Convolutional Neural Networks (CNNs). It should be noted that there are several differences between the proposed approaches and CNN-based approaches. Deep learning approaches work well if *we have a fixed set of object categories and a massive number of examples per category that are sufficiently similar to the test images*. In real-world scenarios, these assumptions are not satisfied, and the robot needs to learn new concepts using very few training examples on-site. While deep learning is a very powerful tool, there are several limitations to use such approaches in open-ended domains. CNNs are incremental by nature but not open-ended since the inclusion of new categories enforces a restructuring in the topology of the network. In other words, the target set of classes is predefined in incremental learning, and the representation of these classes is improved over time, whereas in open-ended learning the set of classes is growing continuously. Moreover, CNNs are data-hungry approaches, and training with limited data leads to poor performance. Catastrophic forgetting is another important limitation of these approaches.

3 Proposed approach

A point cloud of an object is represented as a set of points, $\mathbf{p}_i : i \in \{1, \ldots, n\}$, where each point is described by their 3D coordinates $[x, y, z]$ and RGB information. In this work, we mainly use the GOOD object descriptor to represent the object as a histogram [8,13]. The reason why we use the GOOD rather than other 3D object descriptors is that the GOOD is a pose- and scale-invariant descriptor, and therefore suitable for 3D perception in autonomous robots. As shown in Fig. 1, this method performs a principal component analysis on the point cloud of an object to find the eigenvectors of the object. In particular, given a point cloud of an object that contains n points, the center of gravity of the object is first calculated as $\mathbf{c} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{p}_i$. The normalized covariance matrix, Σ, of the object is constructed:

$$\Sigma = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{p}_i - \mathbf{c})(\mathbf{p}_i - \mathbf{c})^T. \quad (1)$$

Then, eigenvalue decomposition is performed on the Σ, therefore, we have: $\Sigma\mathbf{V} = \mathbf{EV}$ where $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3]$ contains the three eigenvectors, $\mathbf{E} = \mathrm{diag}(\lambda_1, \lambda_2, \lambda_3)$ is a diagonal matrix of the corresponding eigenvalues and $\lambda_1 \geq \lambda_2 \geq \lambda_3$. Since the covariance matrix is symmetric positive, its eigenvalues are positive and the eigenvectors are orthogonal. In this work, the first two axes of an object's reference frame, X and Y, are defined by the eigenvectors v_1 and v_2, respectively. We define the Z-axis as the cross-product of $v_1 \times v_2$. The reference frame estimated for an example partial point cloud is shown in Fig. 2.

It is worth mentioning that over different trials, the direction of eigenvectors is not unique and has 180° ambiguity. A sign disambiguation method is used to avoid this problem. The resulting unambiguous local reference frame, centered on the object, is then used to create three orthogonal projection planes. The projections are divided up in a grid of $n \times n$ bins, which are used to compute a normalized distribution matrix by counting how many points fall within each bin. The histogram of the plane is created by stringing the rows of the matrix together. The obtained histograms corresponding to the three projections are then combined to form a single representation for a given object. The histogram appearing first in the combined histogram is the one with the highest entropy. The second one is the one with the lowest variance of the remaining two, automatically placing the remaining one in the last position.

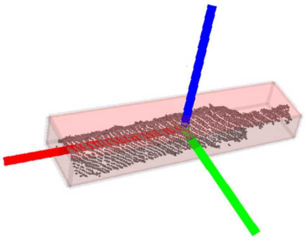

Fig. 2 Reference frame estimated for a partial view of a bottle object: the X, Y, and Z axes are represented by red, green, and blue lines. The bounding box of the object is also computed by finding minimum and maximum points in each axis (colour figure online)

The GOOD object descriptor does not contain color information. Therefore, we have decided to append color constancy information to the GOOD object descriptor by taking an average color of all points of the object. The idea of considering color constancy information is inspired by the work of Bramão et al. [1], which showed the importance of color constancy in object recognition tasks. Therefore, the integration of color constancy information of an object seems to be sufficient to improve the performance of object recognition. Color diagnostic objects will have a single dominant color that is typical for this object and could be used for the recognition of this object. Non-color diagnostic objects will not have a dominant color value and thus cannot really be used to recognize an object. Human perception and recognition, of course, do not just use color constancy information to recognize objects. However, the research by Bramão et al. [1] showed that the color diagnosticity of an object significantly influences the performance of object recognition. In most of cases, in addition to the shape properties, it is sufficient to only look at the color constancy information. Moreover, the cost of the implementation is less than using an independent texture descriptor (e.g., ORB [23]), and it is not substantially altering the shape descriptor. Given this point that only m bins are appended to the final object description for the color constancy information, it would not really affect the GOOD descriptor. It is worth to mention that the size of m depends on the color space. In most cases, m is set to three, which is much smaller than the size of the shape descriptor ($3 \times n^2$). Therefore, to avoid the dominance of the shape information, we add the parameter color weight, w. In all experiments, the obtained object representations are paired with an instance-based learning (IBL) approach (see, e.g., [21]). Therefore, a category is described by a set of known instances. An advantage of the IBL approaches is that they can recognize objects using a minimal number of experiments, and the training phase is very fast. IBL is a baseline approach to evaluate object representations. However, more advanced approaches like SVM and Bayesian [9,12] approaches can be easily adapted. Similarly, a simple baseline recognition mechanism in the form of the nearest neighbor classifier is used. In particular, IBL approaches can be seen as a combination of particular object representation, similarity measure, and classification rule. Since representing color constancy in different color spaces may lead to different results, we convert the object's color constancy into three different color spaces, including RGB, YUV, and HSV, and further discuss the procedure of combining color constancy and shape information to form a descriptor for a given object and color spaces. In a real-world scenario, the different color spaces can lead to different results when the lighting conditions in the scene change. Therefore, additional real-world experiments are performed in which an extra light source is placed underneath the camera. The results of these experiments could show a relation between the applied color space and its robustness with regard to environmental lighting conditions.

In RGB space, often the most popular color space, colors are made up of red, blue, and green channels, having a range of values [0, 255]. We get the RGB values for all points of the object and calculate the summation of each channel values separately. We then get the average colors of the object by dividing the obtained red, green, blue values by the number of points of the object. Finally, since the shape information is normalized, i.e., having a range from 0 to 1, we also normalize the obtained color values to be in the range of [0, 1], by diving each color to 255. The obtained values are then appended to the shape description of the object.

YUV space is mainly used for television transmission and represents a color by three components, one channel for luminance and two channels for chrominance. The Y component determines the brightness of color, which is referred to as luminance. The U and V component determines the color itself, also called chroma. The value of Y ranges from 0 to 255, while the value of U and V ranges −128 to +127. The YUV values can be derived from the RGB values using the following linear transformation:

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.168 & -0.331 & 0.500 \\ 0.500 & -0.418 & -0.0813 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 0 \\ 128 \\ 128 \end{bmatrix}, \tag{2}$$

where 128 is added to the U and V component to make sure that each of the YUV components ranges in [0, 255]. Afterward, the obtained colors are normalized and appended to the histogram of the object as done for the RGB color space.

HSV color space was developed to take into consideration how humans view color, where H stands for hue, S stands for saturation, and V stands for value. In particular, it describes a color (hue) in terms of the saturation (shade) and value (brightness). The hue components represent the angle, which has a range of [0, 360] degrees. The saturation component describes the percentage of gray in a particular color and value works in conjunction with saturation and describes the brightness or intensity of the color, with a range from [0, 100] percent. The RGB value of every detected point is converted to HSV; this is done using the minimum and maximum value of the normalized RGB value of the point in the point cloud. It is worth to mention that the final object descriptor is formed the same way as the other two color spaces. The normalized HSV color dissimilarity, d, of two object views, p and q, can be computed using the following equations:

$$\begin{aligned} h &= \frac{\min(abs(h_q - h_p),\ 360 - abs(h_q - h_p)}{180.0}, \\ s &= abs(s_q - s_p), \\ v &= \frac{abs(v_q - v_p)}{255.0}, \\ d &= h + s + v, \end{aligned} \tag{3}$$

As stated earlier, after forming the object descriptor containing both color and shape information, we use the *color weights* parameter to set how important the difference in color is of the two compared object representations. We are doing this because in the representation of the object, the number of bins representing the shape of the object is much more than the number of bins representing the color information (675 bins vs. 3 bins), and hence the shape information will largely dominate the decision. Therefore, we calculate the difference by using a weighted distance function, as shown below:

$$D(p, q) = (1 - w) \times d_s(p, q) + w \times d_c(p, q), \tag{4}$$

where d_s is the difference in the shape space, d_c is the difference in the color space, and w is the color weight, which is a value between 0.0 and 1.0.

4 Result and discussion

Three types of experiments were carried out to evaluate the proposed approach. It should be noted that in addition to the GOOD descriptor [13], two popular state-of-the-art 3D object descriptors including, VFH [25] and ESF [29] were evaluated, which are available in the Point-Cloud Library.[1] We compare the obtained results and use the best configuration as the system's default configuration in the second

[1] http://pointclouds.org/.

Table 1 System configurations that obtained *best* object recognition accuracy

No.	Descriptor	#bins	Distance function	Accuracy	Time (s)
1	GOOD	15	Manhattan	0.97	3.351
2	GOOD	15	Gower	0.97	3.450
3	GOOD	15	Sorensen	0.97	3.512
4	GOOD	15	Motyka	0.97	3.666
5	GOOD	15	Euclidean	0.97	4.078
6	GOOD	15	Cosine	0.97	4.297
7	GOOD	15	Dice	0.97	4.380
8	GOOD	25	χ^2	0.97	4.219
9	GOOD	30	Bhattacharyya	0.97	5.771
10	GOOD	35	Bhattacharyya	0.97	6.116
11	GOOD	35	χ^2	0.97	5.348
12	ESF	—	Manhattan	0.97	8.055
13	ESF	—	Sorensen	0.97	7.983
14	ESF	—	Neyman	0.97	7.983
15	ESF	—	Bhattacharyya	0.97	8.055
16	ESF	—	Euclidean	0.97	8.174

round of experiments (open-ended evaluation). In the following subsections, we have investigated the importance of shape information and similarity measures using an extensive set of offline evaluations and considered the importance of color constancy and color spaces in a broad set of open-ended assessments.

4.1 Classical offline evaluation using the restaurant object dataset

For this round of experiments, we have used the restaurant object dataset since it has a small number of classes (10 categories) with a significant intra-class variation that is suitable for performing extensive sets of experiments. The parameter of the selected object descriptors must be tuned to provide a good balance between recognition performance, memory usage, and processing speed. The descriptiveness of the GOOD descriptor was evaluated with varying *number of bins*, n, ranging from 5 to 50 with an interval of 5. For the VFH descriptor, we performed a parameter sweep on the *normal estimation radius* parameter, ranging from 2 to 10 cm with an interval of 2 cm, to find the value which resulted in the highest accuracy. The ESF object descriptor does not have any parameters to be optimized. Furthermore, the choice of the similarity measure has an impact on the recognition performance. Since the selected object descriptors represent an object as a normalized histogram, the dissimilarity between two histograms can be computed by different distance functions. We refer the reader to a comprehensive survey on distance/similarity measures provided by Cha [3]. In this work, during the selection of the distance functions, care was taken to select functions that are dissimilar to each other. This policy will increase the chance that different distance functions lead to different results. Based on these considerations, the following 14 functions have been explored: *Euclidean, Manhattan*, χ^2, *Pearson, Neyman, Canberra, KL divergence, symmetric KL divergence, Motyka, Cosine, Dice, Bhattacharyya, Gower,* and *Sorensen*. We refer the reader to [3] to check the mathematical equations. We therefore performed a total of $224 = (10 \times 14) + (5 \times 14) + 14$ tenfold cross-validation experiments to obtain best configuration for each method. All combination of parameters that obtained the *best accuracy* is summarized in Table 1. By comparing all experiments, it is clear that the GOOD with 15 bins and a Manhattan (city-block) distance function configuration obtained the best performance in terms of *accuracy* and *computation time*. The Manhattan distance function is in the L_p Minkowski family and has very low computational expenses. The accuracy of the proposed system with this configuration was 0.97. A complete experiment (including both learning and recognition phases) using this configuration took 3.351 s. The following results are computed using this configuration unless otherwise noted.

Although a large number of bins provides more details about the point distribution, it increases computation time, memory usage, and sensitivity to noise. The descriptiveness of VFH was not as good as the other descriptors. VFH with the radius parameter set to 6cm and *Canberra* distance function resulted in the best performance with a 0.94 accuracy followed by the same radius parameter and a *Motyka* function which resulted in an accuracy of 0.93. One crucial observation is that for VFH, there is a significant drop in performance when the normal estimation radius becomes too small or too large. It was observed that ESF performed well on all distance functions, always having a precision greater than 0.95.

Several factors influence the computation time of an instance-based object recognition approach. They include (i) number of training instances, (ii) distance function, and (iii) object descriptor. In this round of experiments, the number of instances was the same for all of the approaches. In the case of distance functions, it is worth mentioning that a distance function requires the same amount of time regardless of the input objects, i.e., constant time O(1). Therefore, distance functions should be compared in terms of mathematical complexity and not time complexity. By comparing the mathematical equation of the selected distance functions, it is clear that the Manhattan function is the optimum one. To evaluate an object descriptor's computation time, we randomly select 20 objects and measure the average computation time required to generate representations for the given objects. We observed that the GOOD achieved the best performance, which was around 10 and 44 times better than ESF and VFH, respectively. The underlying reason is that the GOOD works directly on 3D point clouds and requires neither triangulation of the object's points nor surface meshing [8]. Therefore, we use the GOOD descriptor as the basis of the proposed model in the remaining experiments.

4.2 Open-ended evaluation using RGB-D object dataset

In this round of experiments, we explore the importance of color constancy and color spaces. To evaluate the performance of object recognition approaches in an open-ended domain, Kasaei et al. [10] have recently adopted a teaching protocol which simulated the simultaneous nature of learning and recognition. The main idea is to emulate the interactions of a robot with the surrounding environment over long periods. The teaching protocol determines which examples are used for training the algorithm, and which are used to test the algorithm. This protocol is based on a Test-then-Train scheme, which can be followed by a human user or by a simulated user. We develop a simulated teacher to follow the protocol and to autonomously interact with the system using ***teach***, ***ask***, and ***correct*** actions. In this experiment, the robot initially has zero knowledge, and the training instances become gradually available according to the teaching protocol.

The idea is that the simulated teacher introduces a category to the robot using three randomly selected object views. The robot creates a model for that category based on these instances. Afterward, the teacher picks a never-seen-before object view and tests the robot to see if it has learned the category, and learning this category does not interfere with the previously learned categories. This is done by asking the robot to recognize unseen object views of the currently known categories. When the robot makes a misclassification, the teacher will provide feedback with the correct category. This way, the robot adjusts its category model using the mistaken instance. The simulated teacher estimates the recognition accuracy of the robot using a sliding window of size $3n$ iterations, where n is the number of categories that the robot has already learned. If the number of iterations it took since the last time the agent learned a new category is less than $3n$, all results are used. If the recognition performance of the agent is higher than the protocol threshold, τ, the simulated teacher introduces a new category. It is worth mentioning that the original protocol suggests to set the τ to 0.67, meaning the object recognition accuracy is at least twice the error rate. In our experiment, we set the protocol threshold to 80% since we aim to force the robot to learn and recognize object categories more precisely. This way, object recognition accuracy is at least four times better than error. Furthermore, considering such a high threshold not only makes it harder to learn new object categories, but also highlights the importance of combining color and shape information in open-ended learning scenario. This relatively high protocol threshold also allows for robustness tests, as configurations that are still able to learn many categories can be considered to be more robust. If the agent could not meet this protocol threshold after a certain number of iterations (e.g., 100), a breakpoint is encountered. This way, the simulated teacher can state that the agent cannot learn any more categories. The agent may learn all existing categories before reaching the breaking points. In such cases, it is no longer possible to continue the protocol, and the evaluation process is halted. In the reported results, this is shown by the stopping condition, "*lack of data*."

4.2.1 Dataset and evaluation metrics

In this round of experiments, we use the Washington RGB-D dataset [15]. This dataset is known as one of the largest available 3D objects datasets and consists of 51 categories with 250.000 views of 300 objects. When an experiment is carried out, learning performance is evaluated using several measures [7,19,21], including: (i) the number of learned categories (NLC) at the end of the experiment, an indicator of *how much the system was capable of learning*; (ii) the number of question/correction iterations (QCI) required to learn those categories and the average number of stored instances per category (AIC), indicators of *time and memory resources required for learning*; (iii) global classification accuracy (GCA), computed using all predictions in a complete experiment, and the average protocol accuracy (APA), i.e., average of all accuracy values successively computed to control the application of the teaching protocol. GCA and APA are indicators of *how well the system learns*.

Table 2 Summary of evaluation using shape information

No.	w	QCI	NLC	AIC	GCA	APA
1	0.0	648.10 ± 196.76	18.90 ± 4.38	11.78 ± 1.35	0.74	0.84

Table 3 Summary of evaluation in RGB color space

No.	w	QCI	NLC	AIC	GCA	APA
1	0.1	922.20 ± 459.24	24.10 ± 7.53	11.60 ± 1.84	0.76	0.84
2	0.2	1217.70 ± 669.51	31.80 ± 11.66	10.84 ± 1.40	0.78	0.84
3	0.3	1881.60 ± 555.00	44.70 ± 8.10	11.28 ± 1.27	0.80	0.841
4	0.4	1751.80 ± 477.00	45.70 ± 7.04	10.19 ± 1.19	0.81	0.85
5	0.5	1656.20 ± 260.22	49.40 ± 4.72	8.92 ± 0.80	0.82	0.85
6*	0.6	1632.50 ± 153.28	**51.00 ± 0.0**	8.30 ± 0.77	0.84	0.86
7*	0.7	1509.50 ± 104.62	**51.00 ± 0.0**	7.55 ± 0.57	0.85	0.86
8*	0.8	1452.30 ± 76.15	**51.00 ± 0.0**	7.07 ± 0.47	**0.86**	0.87
*9**	*0.9*	*1410.20 ± 43.18*	*51.00 ± 0.0*	*6.79 ± 0.38*	*0.86*	*0.88*
10	1.0	1257.10 ± 609.35	33.30 ± 10.84	10.50 ± 1.38	0.79	0.85

Best result highlighted by bold and italic values
*Stopping condition was "*lack of data*"

4.2.2 Results

Since the order of introducing the categories may have an effect on the performance of the system, ten experiments were carried out for each of *shape-only*, *color-only* ($w = 1.0$), and *nine combinations of shape and color* in three mentioned color spaces, i.e., $w \in \{0.1, 0.2, \ldots, 0.9\}$, resulting in 330 experiments. This is due to the nature of IBL approaches that the recognition of new objects relies on all the previously learned objects. For example, if the teacher introduces a red apple right after a red tomato (both a red color and a similar shape), it would be harder to recognize this new object than when a banana was introduced after the red tomato (different color and different shape). Detailed summaries of the obtained results are reported in Tables 2, 3, and 4 and depicted in Figs. 3, 4, 5, 6, and 7. For all results, boxplots are added to show the variation of obtained results for each configuration based on *minimum*, *first quartile*, *median*, *third quartile*, and *maximum* performances. Line plots are also added to display the *average* number of learned categories as a function of color weight.

One important observation is that considering color constancy information significantly improved object recognition performance. It was found that the performance of the agent is improved by increasing the level of color weight in all color spaces. Notably, the agent learned all 51 categories in all color spaces when the color weight was in the range of $0.6 \leq w \leq 0.9$. It is worth to mention, in this range, all experiments concluded prematurely due to the "*lack of data*," i.e., no more categories available in the dataset, indicating the potential for learning many more categories. Moreover, it was observed that neither the agent with *color-only* nor the one with *shape-only* configurations could learn all categories in all of the experiments.

On closer inspection, we can see that the combination of HSV color and shape model resulted in a better performance in all levels of color combination, as clearly shown in Fig. 3. By comparing all approaches, it is also visible that the agent learned all categories faster in HSV space than in other color spaces. It can also be concluded that shape+HSV ($w = 0.9$) obtained the best GCA and APA with stable performance. In contrast, the performance of the agent with *shape-only* ($w = 0.0$) configuration was the worst among the evaluated configurations. In the case of *color-only* ($w = 1.0$), the best performance was obtained in HSV color space, where the agent on average learned 42.30 categories, and YUV and RGB spaces achieved the second and third places by learning on average 40.20 and 33.30 categories, respectively.

Figure 4 illustrates *"how fast"* the learning occurred in each of the experiments while shedding light on the number of learned categories (blue lines). It shows the number of question/correction iterations (QCI) required to learn a certain number of categories. We can see that, on average, the longest experiments were observed with shape+YUV, when the w parameter was set to 0.5. The shortest ones were observed with shape+HSV with $w = 0.8$. It should be noted that the agent with shape+HSV ($w = 0.8$) configuration was able to learn all 51 categories in all experiments, while the experiments with shape+YUV ($w = 0.5$) were stopped due to reaching the *break point* condition after leaning 50 categories on average (see Tables 4, 5). In the case of shape+RGB, the best performance of the agent was achieved when the w was set to 0.5. With this shape+RGB configuration, the agent on average learned all categories using 1410.20±43.18

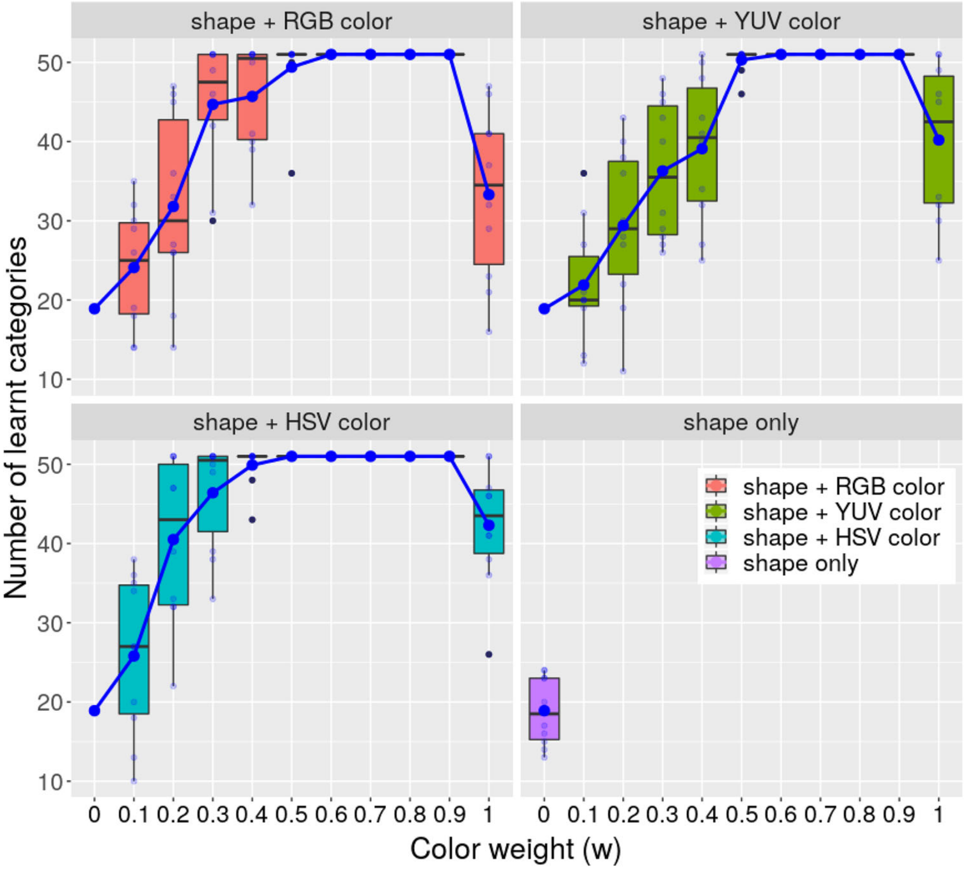

Fig. 3 Summary of open-ended evaluation of all approaches; These plots show the number of learned categories versus color weight for all experiments in four different spaces. Boxplots represent the distribution of obtained results for each configuration based on *minimum*, *first quartile*, *median*, *third quartile*, and *maximum* performances. The blue lines represent the *average* number of learnt categories as a function of color weight (colour figure online)

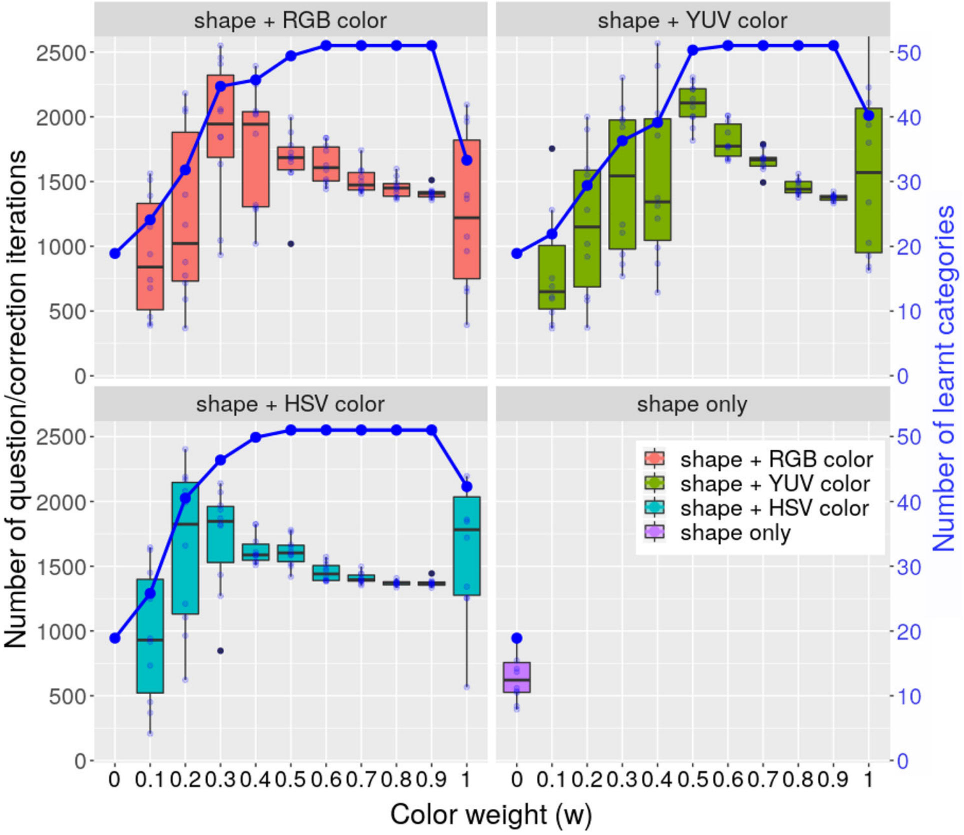

Fig. 4 Summary of open-ended evaluations: these graphs show the number of question/correction iterations (QCI) required to learn a certain number of categories as a function of color weight. The blue lines also represent the average number of learned categories in different combinations of color and shape (colour figure online)

Table 4 Summary of evaluation in YUV color space

No.	w	qci	nlc	aic	gca	apa
1	0.1	802.30 ± 443±21	21.90 ± 7.49	11.53 ± 1.72	0.75	0.84
2	0.2	1183.20 ± 572.72	29.40 ± 10.13	11.64 ± 1.43	0.77	0.84
3	0.3	1507.00 ± 587.95	36.30 ± 8.87	11.62 ± 1.53	0.79	0.84
4	0.4	1524.10 ± 655.25	39.10 ± 9.31	10.65 ± 1.79	0.80	0.84
5	0.5	2095.50 ± 161.95	50.30 ± 1.64	10.95 ± 0.85	0.81	0.84
6*	0.6	1817.70 ± 138.34	**51.00 ± 0.0**	9.31 ± 0.76	0.82	0.85
7*	0.7	1659.90 ± 84.90	**51.00 ± 0.0**	8.32 ± 0.47	0.84	0.86
8*	0.8	1455.10 ± 58.64	**51.00 ± 0.0**	7.23 ± 0.35	0.85	0.87
9*	***0.9***	***1375.50 ± 26.95***	***51.00 ± 0.0***	***6.58 ± 0.31***	***0.87***	***0.88***
10	1.0	1568.30 ± 664.25	40.20 ± 9.56	10.55 ± 1.71	0.80	0.84

Best result highlighted by bold and italic values
*Stopping condition was "*lack of data*"

Table 5 Summary of evaluation in HSV color space

No.	w	QCI	NLC	AIC	GCA	APA
1	0.1	958.80 ± 523.49	25.80 ± 10.08	10.89 ± 1.69	0.77	0.84
2	0.2	1639.30 ± 618.87	40.50 ± 10.33	11.01 ± 1.18	0.80	0.84
3	0.3	1717.40 ± 407.80	46.40 ± 6.92	9.90 ± 0.98	0.81	0.84
4	0.4	1628.00 ± 114.81	49.90 ± 2.60	8.7079 ± 0.62	0.83	0.85
5*	0.5	1608.50 ± 113.54	**51.00 ± 0.0**	8.12 ± 0.56	0.84	0.86
6*	0.6	1454.20 ± 71.34	**51.00 ± 0.0**	7.17 ± 0.45	0.85	0.87
7*	0.7	1406.20 ± 43.20	**51.00 ± 0.0**	6.73 ± 0.36	**0.87**	0.88
8*	0.8	**1369.60 ±19.66**	**51.00 ± 0.0**	6.46 ± 0.29	**0.87**	0.88
9*	***0.9***	***1371.90 ± 28.90***	***51.00 ± 0.0***	***6.42 ± 0.33***	***0.87***	***0.89***
10	1.0	1624.00 ± 513.38	42.30 ± 7.66	10.34 ± 1.78	0.81	0.85

Best result highlighted by bold and italic values
*Stopping condition was "*lack of data*"

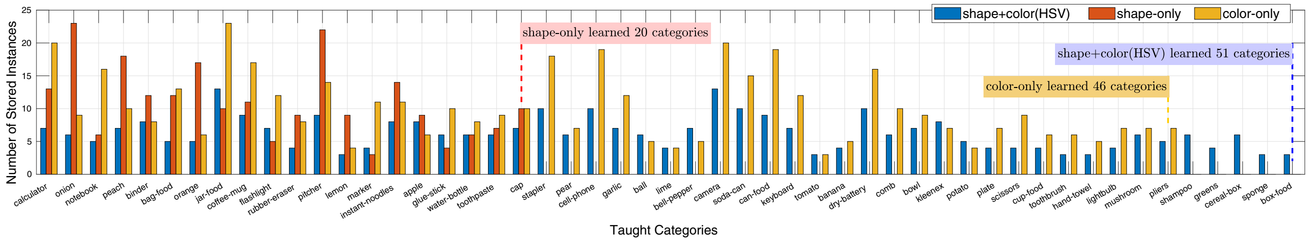

Fig. 5 This graph shows the number of instances stored in the models of all of the categories in three system configurations: shape-only, color-only, and shape+HSV ($w = 0.9$). Each bar represents the three instances provided at the introduction of the category, together with any instances that had to be corrected somewhere along the experiment. *Onion*, *jar-food*, and *camera* were the most difficult categories for *shape-only*, *color-only*, and *shape+color* configurations, respectively, i.e., requiring the largest number of instances. It should be noted that categories that were introduced near the end of the experiment have been tested less, which is clearly visible in a general trend for fewer instances to be included for categories appearing later. The agent learned 20, 46, and 51 categories with shape-only, color-only, and shape+color (HSV) configurations, respectively. It is worth to mention that the shape+color experiment finished due to *lack of data* condition, showing the potential to learn many more categories

question/correction iterations. It was also observed that the longest experiments were continued for 1881.60 ± 555.00 question/correction iterations with shape+RGB ($w = 0.3$) configuration and the agent on average was able to learn 44.70 ± 8.10 categories (see Table 3).

Figure 5 represents the exact number of stored instances per category for shape-only, color-only (HSV), and shape+HSV ($w = 0.8$). By comparing the obtained results, it can be concluded that the agent with shape +HSV configuration not only stored much fewer instances per category but also

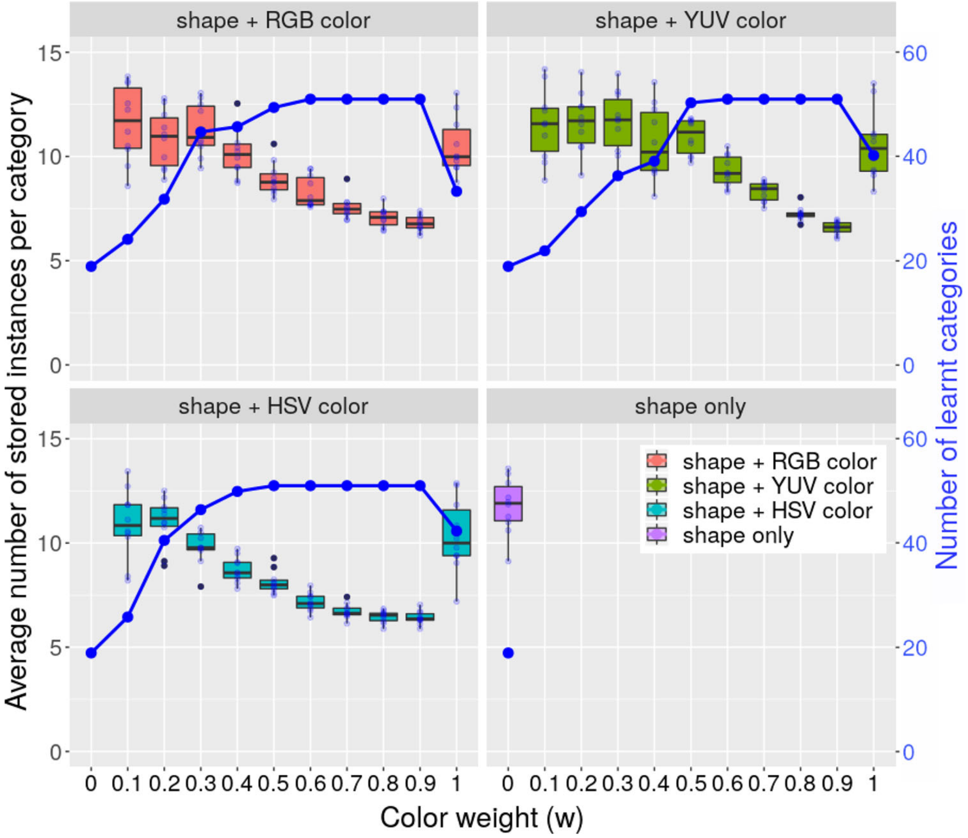

Fig. 6 Summary of open-ended evaluations: these graphs represent the average number of stored instances per category and the average number of learned categories at the end of experiments as an indicator of how much memory each approach uses to learn a certain number of categories. The blue lines display the average number of learned categories as a function of color weight (colour figure online)

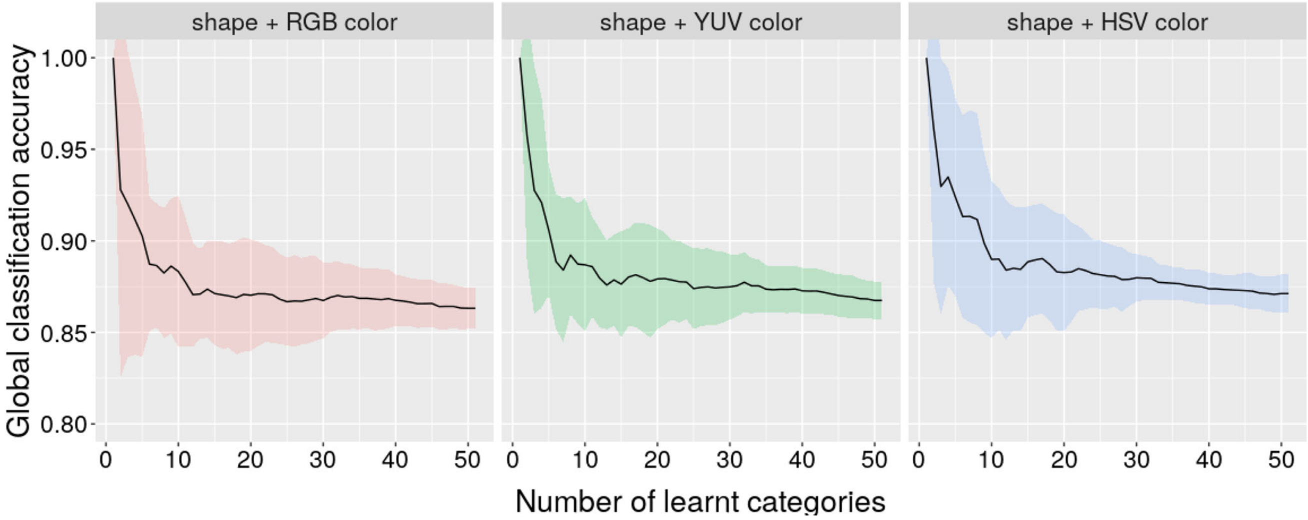

Fig. 7 These graphs show the global classification accuracy as a function of the number of learned categories in three different color spaces. In all these experiments, color weight was set to 0.9

it could learn more categories as well. Figure 6 provides a detailed summary of the obtained results concerning the average number of stored instances per category (AIC) as a function of color weight. By comparing all approaches, it is clear that shape+HSV, shape+YUV, and shape+RGB on average stored less than seven instances per category to learn all categories, while *shape-only* and *color-only* required more than 10 instances per categories to learn 18.90 and 33.30 categories, respectively. The shape+HSV ($w = 0.9$) configuration on average stored smallest number of instances per category (see Table 5).

Figure 7 shows the global classification accuracy obtained by the best combination of shape and color as a function of the number of learned categories in three different color spaces

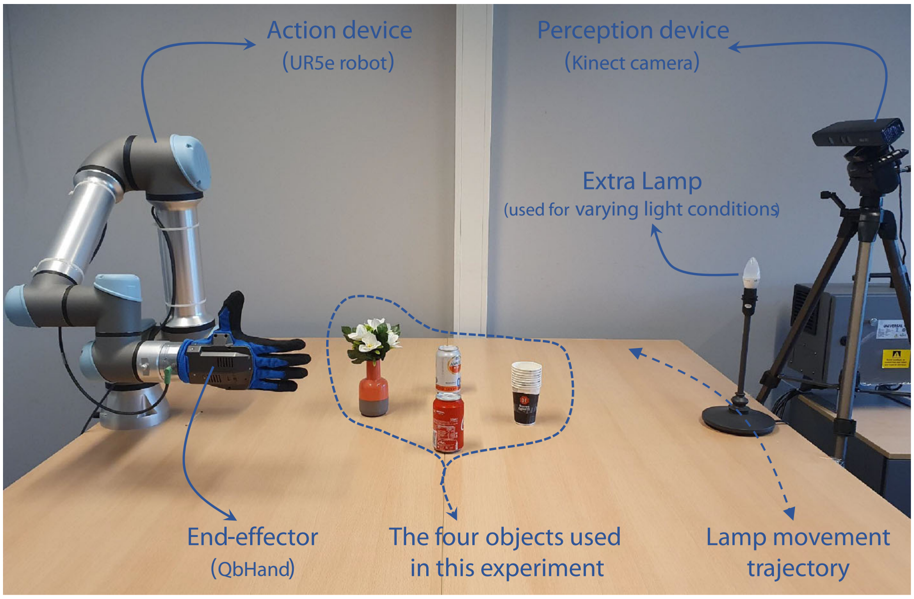

Fig. 8 Our experimental setup consists of a computer for human–robot interaction purposes, a Kinect sensor, and a UR5e robotic-arm as the primary sensory-motor embodiment for perceiving and acting upon its environment. We also use an extra lamp to test the performance of the system under varying light conditions

(i.e., the best configuration in each color space is highlighted by the bold and italic in respective tables). One important observation is that accuracy decreases in all approaches, as more categories are introduced. This is expected since a higher number of categories known by the system tends to make the classification task more difficult.

4.3 Real-robot experiment

To show the strength of the proposed approach, we carried out a real-robot experiment in the context of the *serve_a_coke* scenario. We have integrated the proposed approach into the cognitive robotics system presented in [11]. In this experiment, a table is in front of a Kinect sensor, and a user interacts with the system. There is one instance of four object categories on the table: *CokeCan*, *BeerCan*, *Cup*, and *Vase*. This is a suitable set of objects for this test, since there are objects with very similar shapes and different colors (*CokeCan*, *BeerCan*, and *Cup*) and also objects with very different shapes and similar colors (*CokeCan* and *Vase*). The experimental setup is shown in Fig. 8. It consists of a computer for human–robot interactions, a Kinect sensor for perceiving the environment, and a Universal Robot (UR5e) for manipulation purposes.

Figure 9 presents some snapshots of this experiment. It is worth mentioning that a constraint has been applied to the Z-axis of objects, which forces its initial direction to be similar to the direction of the table's Z-axis. At the start of the experiment, the set of categories known to the system is empty, and therefore, the system recognizes all table-top objects as *Unknown* (Fig. 9a). It should be noted that the robot can learn about a set of categories in advance from batch data (i.e., dataset of observations with ground truth labels), and improves its knowledge in active and on-line manners. A user interacts with the system by teaching all object categories. The system conceptualizes them using the extracted object views and recognizes all objects properly (Fig. 9b). In this task, the robot must be able to detect the pose of objects as well as to recognize the label of all active objects. Afterward, it has to grasp the *CokeCan* object (Fig. 9c, d) and transport it on top of the *Cup* object and serve the drink (Fig. 9e). The robot finally returns to the initial pose (Fig. 9f). It was observed that the proposed object descriptor is capable of providing distinctive global features for recognizing geometrically similar objects with different colors and vice versa. This evaluation also illustrates the process of learning object categories in an open-ended fashion.

Most object recognition systems are sensitive to lighting conditions and require a fair amount of time and effort to calibrate. We performed another real-time demonstration to evaluate the robustness of the proposed approach under various lighting conditions. As shown in Fig. 8, we placed a lamp underneath the Kinect camera as an extra source of illumination and changed its position through the experiment. Figure 10 shows a sequence of snapshots of the behavior of the system in this experiment. It was observed that by combining shape features and color constancy information, we achieved a good level of robustness against varying lighting condition. In particular, the system could recognize all objects correctly under various lighting conditions (see Fig. 10a–e). Some misclassification also happened. Figure 10f shows a snapshot where the robot incorrectly recognized the *BeerCan* and the *CokeCan* objects as *Vase*, and correctly recognized the *Vase* and the *Cup* objects. A video of these experiments is available online at: https://youtu.be/a9cL21sf6Qc

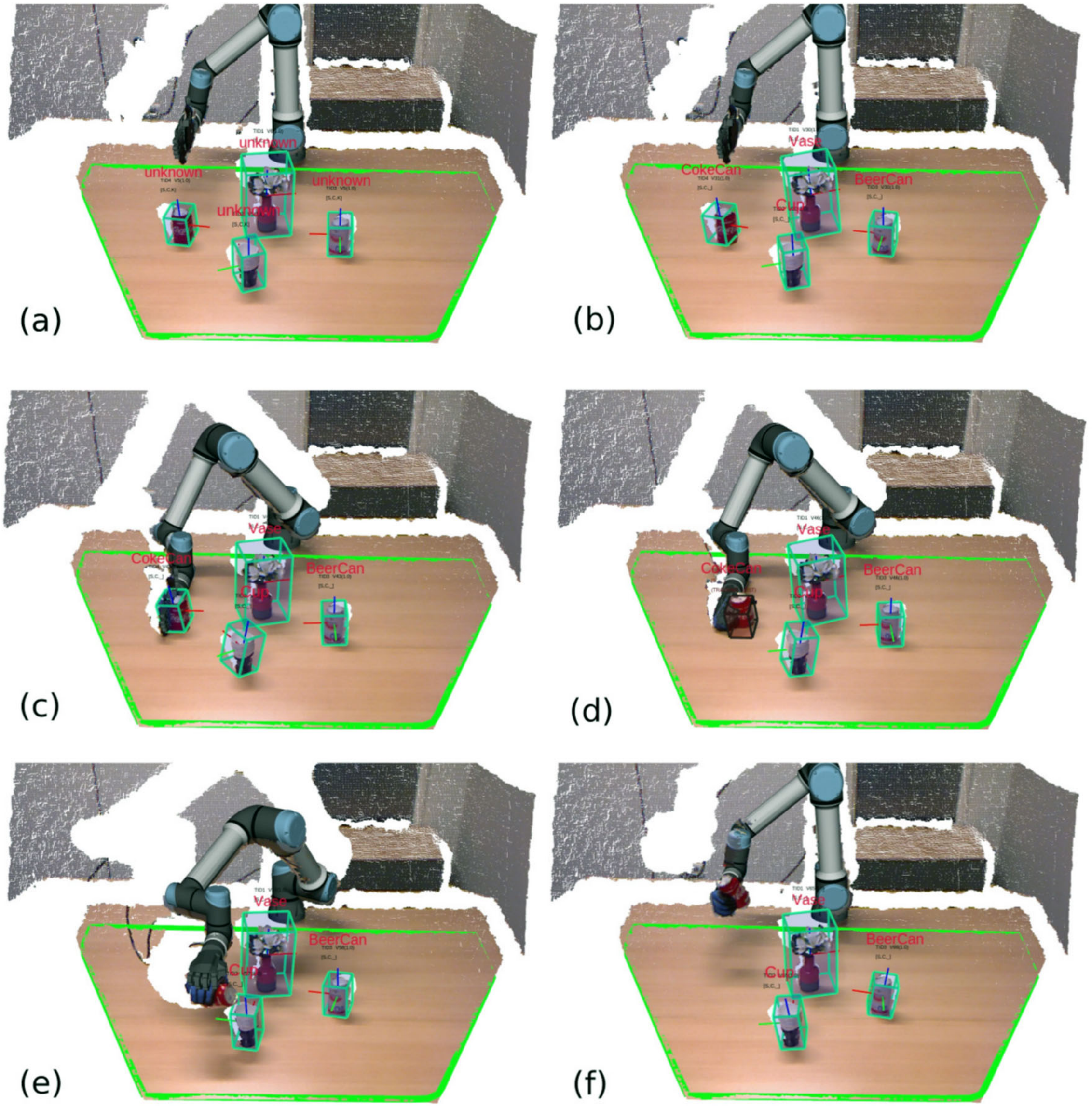

Fig. 9 System performance during the *serve_a_coke* scenario; **a** Initially, the system starts with no knowledge of any object. The posture of the UR5e arm in each state is also visualized. The table is then detected, as shown by the green polygon. Afterward, the object candidates are detected and highlighted by different bounding boxes. The local reference frame of each object represents the pose of the object as estimated by the object tracking module. **b** A user then teaches all the active objects to the system, and all objects are correctly recognized, i.e., the output of object recognition is shown in red on top of each object. **c** The robot then finds out the *CokeCan* object and goes to its pre-grasp area and **d** picks it up first from the table. **e** The robot retrieves the position of *Cup* first, and then moves the *CokeCan* on top of the *Cup* and serves the drink. **f** Finally, the robot goes back to the initial position

5 Conclusion

In this article, we have investigated the importance of shape features, color constancy information, and similarity measures in open-ended 3D object recognition. Toward this goal, an instance-based 3D object category learning and recognition has been developed, which can be seen as a combination of a memory system, an object representation, a similarity measure, and a nearest neighbor classifier. We have selected three state-of-the-art global 3D shape descriptors, namely GOOD [13], ESF [29], and VFH [25], which provide a good trade-off between descriptiveness, computation time, and memory usage and are suitable for real-time robotic application. Besides, a multitude of distance functions has been implemented to measure the similarity of two object views. Accordingly, 224 system configurations have been examined in offline settings.

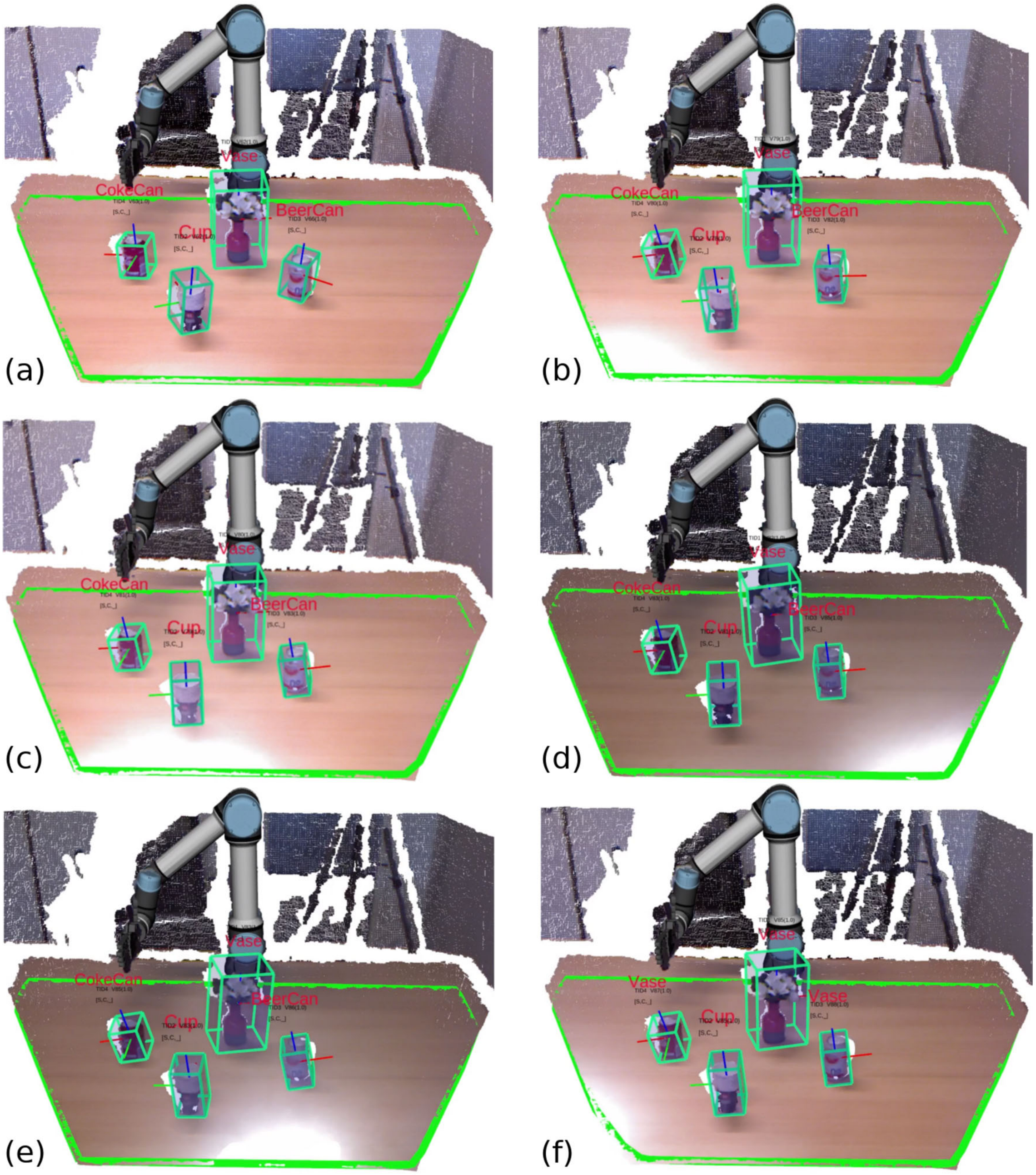

Fig. 10 System performance under various lighting conditions: **a**–**e** the robot could recognize all objects correctly under varying light conditions; **f** some misclassifications also happened where the robot could not recognize *BeerCan* and *CokeCan* objects correctly

The offline experiments have been performed to optimize the parameters of selected shape descriptors and investigate the importance of similarity measures. It was observed that the combination of the GOOD descriptor ($number\_of\_bins = 15$) and the $Manhattan$ function led to the best result in terms of both accuracy and computation time. We then investigated the importance of color information in an open-ended learning setting. In particular, we have added the color constancy information of an object to its shape description. A set of 330 open-ended experiments has been performed in three popular color spaces including: RGB, YUV, and HSV. In this round of experiments, we adopted a teaching protocol to incrementally evaluate the performance of the system concerning several characteristics, including descriptiveness, scalability, and experiment time.

Experimental results show that the overall classification performance obtained with the proposed shape+color approach is clearly better than the best accuracies achieved with the *color-only* and *shape-only* methods. In particular, by setting the color weight parameter in the range of $0.6 \leq w \leq 0.9$ in all color spaces, the agent could learn all categories in all experiments with stable performance. This might suggest that there are reliable color differences between categories and similar color values within categories in the Washington RGB-D dataset [15]. This is not always the case in the real-world environment. Furthermore, it was observed that the performance of the agent with color-only setting ($w = 1$) was better than the shape-only configuration ($w = 0$). This might be caused by a data bias in the dataset. Concerning computational time (QCI), the best result was obtained with shape+HSV ($w = 0.9$), followed by the shape+YUV with the same w. It was also observed that the agent could learn new categories from very few examples in an incremental and open-ended manner. A real demonstration was also carried out to show the usefulness of the proposed method.

Although the addition of color information to the object representation improved the performance of object recognition, the number of bins representing the color constancy information was greatly outnumbered by the number of bins dedicated to the shape of the objects. The color information had a small role in the resulting histogram since only the color constancy of the object was used. In the continuation of this work, we would like to investigate the possibility of integrating color information in a concrete manner. Furthermore, separate distance functions could be used to estimate the similarity of objects in terms of shape and color information.

References

1. Bramão I, Faísca L, Petersson KM, Reis A (2012) The contribution of color to object recognition. In: Advances in object recognition systems, p 16
2. Carvalho L, von Wangenheim A (2019) 3D object recognition and classification: a systematic literature review. Pattern Anal Appl 22(2):1243–1292
3. Cha SH (2007) Comprehensive survey on distance/similarity measures between probability density functions. City 1(2):1
4. Girshick R, Donahue J, Darrell T, Malik J (2014) Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 580–587
5. Guo Y, Bennamoun M, Sohel F, Lu M, Wan J (2014) 3Dobject recognition in cluttered scenes with local surface features: a survey. IEEE Trans Pattern Anal Mach Intell 36(11):2270–2287
6. Hana XF, Jin JS, Xie J, Wang MJ, Jiang, W (2018) A comprehensive review of 3D point cloud descriptors. arXiv preprint arXiv:1802.02297
7. Kasaei SH, Lopes LS, Tomé AM (2018) Coping with context change in open-ended object recognition without explicit context information. In: 2018 IEEE/RSJ international conference on intelligent robots and systems (IROS), IEEE, pp 1–7
8. Kasaei SH, Lopes LS, Tomé AM, Oliveira M (2016) An orthographic descriptor for 3D object learning and recognition. In: 2016 IEEE/RSJ international conference on intelligent robots and systems (IROS), IEEE, pp 4158–4163
9. Kasaei SH, Oliveira M, Lim GH, Lopes LS, Tomé AM (2018) Towards lifelong assistive robotics: a tight coupling between object perception and manipulation. Neurocomputing 291:151–166
10. Kasaei SH, Oliveira M, Lim GH, Seabra Lopes L, Tomé AM (2015) Interactive open-ended learning for 3D object recognition: an approach and experiments. J Intell Robot Syst 80(3):537–553. https://doi.org/10.1007/s10846-015-0189-z
11. Kasaei SH, Shafii N, Lopes LS, Tomé AM (2019) Interactive open-ended object, affordance and grasp learning for robotic manipulation. In: 2019 IEEE/RSJ international conference on robotics and automation (ICRA), IEEE, pp 3747–3753
12. Kasaei SH, Sock J, Lopes LS, Tomé AM, Kim TK (2018) Perceiving, learning, and recognizing 3D objects: an approach to cognitive service robots. In: Thirty-second AAAI conference on artificial intelligence
13. Kasaei SH, Tomé AM, Seabra Lopes L, Oliveira M (2016) GOOD: a global orthographic object descriptor for 3D object recognition and manipulation. Pattern Recogn Lett 83:312–320. https://doi.org/10.1016/j.patrec.2016.07.006
14. Kim T, Jeong M, Kim S, Choi S, Kim C (2019) Diversify and match: a domain adaptive representation learning paradigm for object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 12456–12465
15. Lai K, Bo L, Ren X, Fox D (2011) A large-scale hierarchical multi-view RGB-D object dataset. In: 2011 IEEE international conference on robotics and automation, pp 1817–1824
16. Leng C, Zhang H, Li B, Cai G, Pei Z, He L (2018) Local feature descriptor for image matching: a survey. IEEE Access 7:6424–6434
17. Lima JPSdM, Teichrieb V (2016) An efficient global point cloud descriptor for object recognition and pose estimation. In: 2016 29th SIBGRAPI conference on graphics, patterns and images (SIBGRAPI), IEEE, pp 56–63
18. Logoglu KB, Kalkan S, Temizel A (2016) CoSPAIR: colored histograms of spatial concentric surflet-pairs for 3D object recognition. Robot Autonom Syst 75:558–570
19. Lopes LS, Chauhan A (2007) How many words can my robot learn? An approach and experiments with one-class learning. Interact Stud 8(1):53–81
20. Martinez-Martin E, Del Pobil AP (2017) Object detection and recognition for assistive robots: experimentation and implementation. IEEE Robot Autom Mag 24(3):123–138
21. Oliveira M, Lopes LS, Lim GH, Kasaei SH, Tomé AM, Chauhan A (2016) 3D object perception and perceptual learning in the race project. Robot Auton Syst 75:614–626
22. Regazzoni D, de Vecchi G, Rizzi C (2014) RGB cams vs RGB-d sensors: low cost motion capture technologies performances and limitations. J Manuf Syst 33(4):719–728. https://doi.org/10.1016/j.jmsy.2014.07.011

23. Rublee E, Rabaud V, Konolige K, Bradski G (2011) Orb: an efficient alternative to sift or surf. In: 2011 International conference on computer vision, IEEE, pp 2564–2571
24. Rusu RB, Blodow N, Beetz M (2009) Fast point feature histograms (FPFH) for 3D registration. In: 2009 IEEE international conference on robotics and automation, IEEE, pp 3212–3217
25. Rusu RB, Bradski G, Thibaux R, Hsu J (2010) Fast 3D recognition and pose using the viewpoint feature histogram. In: 2010 IEEE/RSJ international conference on intelligent robots and systems, pp 2155–2162
26. Rusu RB, Cousins S (2011) 3D is here: point cloud library (PCL). In: 2011 IEEE international conference on robotics and automation, IEEE, pp 1–4
27. Rusu RB, Marton ZC, Blodow N, Beetz, M (2008) Learning informative point classes for the acquisition of object model maps. In: 2008 10th International conference on control, automation, robotics and vision, IEEE, pp 643–650
28. Ullrich M, Ali H, Durner M, Márton ZC, Triebel R (2017) Selecting CNN features for online learning of 3D objects. In: 2017 IEEE/RSJ international conference on intelligent robots and systems (IROS), IEEE, pp 5086–5091
29. Wohlkinger W, Vincze M (2011) Ensemble of shape functions for 3D object classification. In: 2011 IEEE international conference on robotics and biomimetics, IEEE, pp 2987–2992
30. Zrira N, Hannat M, Bouyakhf EH (2017) Vfh-color and deep belief network for 3d point cloud recognition. In: Iberian conference on pattern recognition and image analysis, Springer, pp 445–452

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.