Improving 3D Object Detection and Classification Based on Kinect Sensor and Hough Transform

T.J. Mateo Sanguino, and F. Ponce Gómez
Dep. Electronic Engineering, Computer Systems and Automatics
University of Huelva (UHU), Huelva, Spain
tomas.mateo@diesia.uhu.es, francisco.ponce@alu.uhu.es

Abstract— **Hough Transform has been successfully applied to a variety of image processing problems in recent years. This papers presents a novel approach for detecting and classifying 3D objects by using the generalized Hough method and the Kinect™ sensor. Our algorithm considers feature points and color spectra as two interleaved processes to cooperatively recognize objects in a 2.5D fashion. With this strategy, the algorithm automates the image pre-processing operations regardless of scenes (i.e., particle cleaning, hole filling, particle eroding, and object dilating) and reduces the processing load over the sensor's point cloud for 3D object classification. Extensive experiments applied —but not limited— to recognition between different and similar objects, occlusion, and perspective change analyzing fitness and processing time show that the 2.5D approach makes feasible 3D object recognition for applications with video information.**

Keywords— 3D object; classification; depth camera; detection; Hough Transform; image processing; Kinect; pattern recognition

I. INTRODUCTION

In order for an image-based system to autonomously interact in real-world environments (i.e., complex, dynamic and uncertain), the system should recognize and understand its surrounding. Typically, vision systems rely on some way of feature detection and categorization to provide —to itself or to others— information about 2D/3D objects. In this field, a wide sort of vision techniques such as appearance-based, feature-based and histogram-based methods are available, among others [1]. But even a method with a very accurate detection algorithm can be ineffective to be performed in real-time for 3D object recognition. The majority of authors propose fairly precise algorithms but they rarely address the computational cost or optimal parameter settings comprehensively [2]. Therefore, these concerns still remain a major challenge in 3D image processing for real-time purposes [3].

The motivation for this work came as a result of the research described in [4], [5] about a navigation assistance system for a telepresence robot based on augmented virtuality. Although the aim was to provide dependents with improved accuracy and reduced mental workload in teleoperation tasks, we experienced that long-term errors in the robot's odometry —due to wheel slips by obstacles and parquet flooring— decreased the users' perception on the robot's environment. To address this issue, we designed a vision-based support system for mobile robot localization using Kinect™ and a dynamic particle filter [6]. From the research, we found that while the algorithm efficiently worked in most of sceneries (i.e., localization at corners, corridors, windows, and landmarks), it was not so suitable in other environments providing less information (e.g., localization at long blank walls).

With the idea of providing a complementary method to assist the global robot localization when the environment's information is lacking, a preliminary strategy for 3D object recognition is addressed in this paper. We start from the assumption that the characteristic information about an object in an image is habitually contained in its color and contour shape [7]. Object information is commonly obtained from high cost devices such as industrial digital cameras [8], stereovision systems [9] or laser range finders [10]. Nevertheless, they must be combined with other devices to obtain RGB-D information for multi-task purposes. Alternatively, various authors have proposed the use of depth cameras (e.g., Kinect™) as a vision subsystem for several fundamental problems in robotics: navigation, localization, and obstacle detection [11], [12]. Although the benefits turn Kinect™ into a very suitable low-cost solution, it has not yet been extensively used for 3D object recognition —in contrast to range sensors— until overcome some important limitations (i.e., worse calibration and depth resolution, narrower horizontal FOV, and higher amount of sensor readings) [13]. These have implications for the processing time and accuracy in 3D object classification with video streaming.

To assist the global robot localization, we propose a strategy for 3D object classification based in integrating shape and color information from Kinect™. The operation is performed with a novel approach consisting in simplifying the Generalized Hough Transform (GHT) to 2D object sections and using the color spectra as decision criterion. Thus, it is possible to reduce the processing requirements for the point clouds from Kinect™ and compensate —with color features— possible lacks of information in shapes (e.g., occlusion and perspective change). This strategy requires computing fewer points than a complete 3D surface and improves the discrimination of similar objects by color information. So, the paper is structured as follows. The next section provides a brief review on Hough Transform (HT) examples present in literature and discusses their pros and cons. Section III outlines the fundamentals of the generalized HT and its application to object recognition is introduced. Section IV

978-1-4673-7751-5/15/$31.00 ©2015 IEEE

presents the methodology followed to combine the HT method and the Kinect™ sensor for 3D object classification. Then, the experimental results on different case studies are explained. Finally, the paper discusses the findings and limitations.

II. Related Work

The HT has been typically used in image processing as a means to exploit the duality between analytic geometry and feature points —of that geometry— to detect objects [14]. The algorithm was firstly patented in 1962 by P. Hough as a method for recognizing straight lines in bubble chamber photographs and later extended by several authors [15]. Beyond the detection of simple straight lines, the method for detecting instances of complex shapes in an image (i.e., ellipses, circles, and polygons) became a generalized version of the HT [16]. This has been successfully applied to other areas of interest such as the geometric texture measurement for mammographic lesion detection [17]. However, classical HT methods do not scale well for many unknown parameters (i.e., high dimensional spaces). This led to further research into new approaches (e.g., probabilistic, SVM, supervised learning, etc.), thus providing an alternative to improve the performance of conventional HT methods [18]. In these fields, the combination of different interleaving processes for object detection and localization has already demonstrated to benefit from each other and to improve the total performance [19]-[21].

Advances in the generalized HT also led researchers to address 3D object recognition. For instance, a method to extract 3D features by a volumetric vision system was used to detect glasses' frames. However, it stands for an ad-hoc solution centered in applying some geometric constrains and obtaining a plane in which 3D features were concentrated [9]. Closer to the background of our paper, scan matching algorithms in the Hough domain have been mainly exploited for robot localization [22]. As another example, a 3D version of the algorithm presented in [22] was designed for fast global estimate with limited precision [23]. As the main difference, this approach is focused in the correlation of data from a pre-built map and range sensors, which has demonstrated the improvement of feature distributions instead of extracting line features as conventional HT methods [24]. Moreover, approaches based on point clouds for arbitrary shapes have been proposed to generalize 3D object detection [10]. Nevertheless, these methods are still computationally expensive and the development of cost-reduction strategies require further research on Hough processing (i.e., better engineering of the algorithm, tuning to range-finder data, and faster implementations).

III. Basic Principles

In this section a generalization of the HT and its main properties are introduced. The key idea of the HT is to extract features from geometric figures by mapping the input space (i.e., image space) into a more fitting parameter space (i.e., Hough space). For instance, straight lines can be expressed by their following analytic geometric form:

$$y = mx + b \tag{1}$$

where y and x are variables in the Cartesian plane, m is the slope of the line, and b is the y-intercept. The straight line can be drawn for each pair (x_1, y_1), …, (x_n, y_n), which in practice stand for infinite points defined in $\mathbb{R}^2$. The same geometric form can be depicted in the Hough space through a simple transformation considering only two parameters (m, b) per line (Fig. 1a).

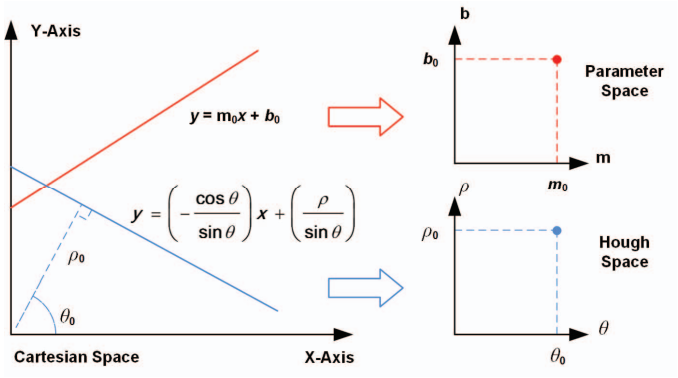
Different parameterizations for representing a line in the image space can be found. As deduced from Eq. (1), computational issues may happen since slopes for vertical lines can go to infinity. Thereby, the polar representation is typically used:

$$y = \left(-\frac{\cos\theta}{\sin\theta}\right)x + \left(\frac{\rho}{\sin\theta}\right) \tag{2}$$

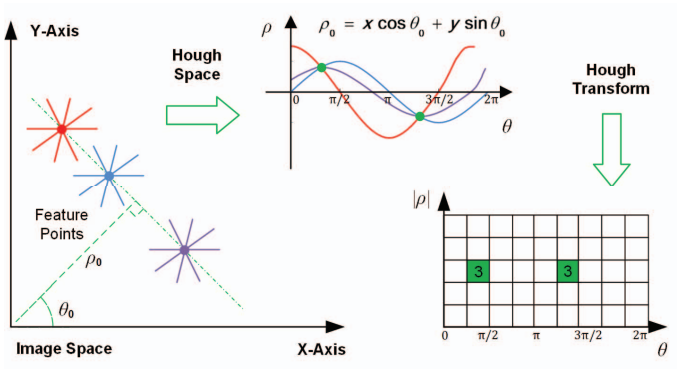
$$\rho = x\cos\theta + y\sin\theta \tag{3}$$

where ρ stands for the distance between the coordinate origin and the point (x, y), whilst θ is the angle between the perpendicular vector to the line and the origin intersection. This transformation allows mapping a straight line to a unique pair (θ, ρ) in the Hough domain (i.e., a point) when $\theta \in [0, \pi)$ and $\rho \in \mathbb{R}$, or $\theta \in [0, 2\pi)$ and $\rho \geq 0$.

Conversely, it is assumed using this concept that a point with coordinates (x_0, y_0) in the Cartesian space can be described by a set of straight lines with pairs (θ_1, ρ_1), …, (θ_n, ρ_n) crossing that point. These can be equally expressed by Eq.



(a)



(b)

Figure 1. Correspondence between the Cartesian space and the Hough space: (a) transformation of straight lines, and (b) transformation of feature points from an object countour

(3) and accordingly depicted in the Hough space through a sine wave (Fig. 1b). Since this idea can be extended to consider the various feature points forming an object in the image space (e.g., its contour), the problem is then reduced to analyze the relation of the sine waves in the Hough space. This is easily done by discretizing the parameter space into bins (θ, ρ), putting a vote in every bin for each wave crossing other ones, and finally finding the bins that have the most votes. This method is indeed equivalent to extract features in the image space.

IV. METHODOLOGY

This section describes the strategy carried out to optimally parametrize the generalized HT method for the Kinect™ sensor, shows the image processing steps for 3D object detection and classification, and explains the algorithm operation.

A. Implementation with Kinect™

The method proposed for the detection and classification of 3D objects was developed for Kinect™ using LabVIEW™ 2011 and math nodes as part of the MathScript RT module. Currently, various wrapper libraries exist to utilize the functionalities of Kinect™ for that programming environment [25]. We have chosen the one developed by the University of Leeds, UK, called Kinesthesia [26]. This toolkit allows direct access to the SDK 1.5 released by Microsoft whose main features are full control of the Kinect™ functions, high stability, and depth measurements directly given in centimeters without requiring scaling or calibration transformations, among others. Kinect™ incorporates a depth detector to the RGB camera, which can be used in multi-task mode. Because of some important features which confer greater capability to detect singular objects as the horizontal and vertical FOVs (57º and 43º, respectively), and depth images in real time (640x480 pixels at 30 fps or 1280x1024 pixels at lower frame rate) we took Kinect™ into consideration to be used in this research. However, other solutions to improve results based on the combination of high-quality cameras and laser range sensors —although more expensive— may be similarly applicable.

B. Parameter Optimization

Typically, real time systems require a higher level of demand in terms of performance compared with other purposes. With this premise, a previous study on the parameters influencing the Hough Transform was carried out before setting our algorithm for video streaming. The piece of code for the GHT was isolated and tested through Matlab®. Three case studies were considered to compute fitness and time processing using figures with varying difficulty: *i*) simple straight line, *ii*) square polygon with medium complexity, and *iii*) face outline with complex shape. The testing bench was performed in the image space with B&W test pictures (24-bits, 270x300 pixels) and consisted in varying the number of points going through a shape and the number of lines crossing each point:

$$Cost = \frac{P}{\Delta\psi} \tag{4}$$

where P stands for a finite set of points $\{p_j\}$ in the input space and $\Delta\psi$ is the angle interval for each line through $\{p_j\}$. This means that the complexity of the GHT algorithm is a quadratic function that increases as shapes include more information. As expected, the computation time decreases for less points and higher angle intervals. On the contrary, the cost rapidly increases for more points and lower angle intervals.

With the aim of finding (P, $\Delta\psi$) to obtain the best fitness with the lowest computational time we analyzed the data based on the minimum Euclidean distance. Considering the best solution as that with least distance regarding its origin, we describe this concept as follows:

$$s(P,\Delta\psi) = \min\left[\sqrt{T^2(P,\Delta\psi) + \left[\frac{1}{F(P,\Delta\psi)}\right]^2}\right] \tag{5}$$

where T and F stand for two dimensions depending on (P, $\Delta\psi$) who describe the time processing and the fitness score, respectively. From this, an optimal tradeoff between processing time and fitness was obtained for P = [50, 100] and $\Delta\psi$ = [5, 10].

C. Image Pre-Processing

Typically, an image pre-processing technique as the one herein implemented consists of the following steps: *i*) intensity threshold selection from the HSI plane for all objects to be detected, *ii*) noise reduction by particle cleaning within the image, *iii*) object contour enhancement by hole filling, *iv*) object contour eroding to clear defects in object shapes, and *v*) object dilating to recovery imperfections on shapes. Although the image pre-processing can get high achievements, the various parameters above mentioned depend on the particular scene and therefore it is not possible to generalize the process. Moreover, humans can take long time to recognize at a glance the elements populating a scene depending on the background, which is unfeasible in practice. Therefore, objects to be

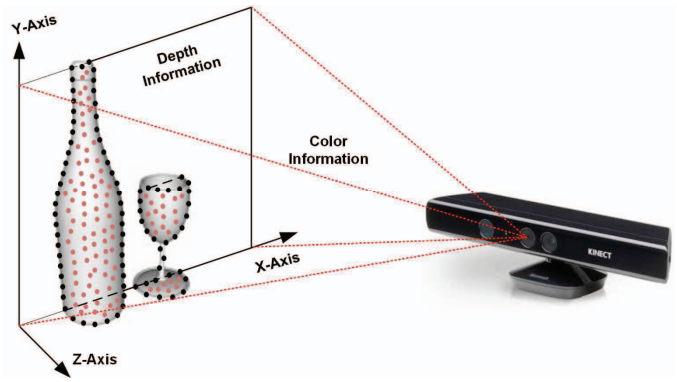


Figure 2. Approach for 3D object classification by combining 2D section and color information obtained from Kinect™

detected within an arbitrary scene should be previously discriminated from background to increase achievement.

With this aim, we have proposed to automate the process of object detection by using a depth camera. We have used the Microsoft Kinect™ device to take advantage from its distance sensor to capture elements within an uncertain scene during the pre-processing stage. To do this, a user-defined parameter (i.e., *Distance_Threshold*) is used to select the range to objects and thus discriminating the rest of the scene for the subsequent classification stage. In this fashion, a 2D perpendicular section is obtained from 3D objects and those within the image placed at a distance different from that desired are separated. So, the detection focuses only on objects at the distance established. This requires computing fewer points than a whole 3D surface (Fig. 2).

With this simple approach we have obtained several advantages. On the one hand, the automation process generalizes the object detection regardless of background, foreground and illumination, thus avoiding the pre-processing operations for the subsequent segmentation (i.e., particle cleaning, hole filling, particle eroding, and object dilating). On the other hand, this expedites the detection of 3D objects by the generalized HT providing a simple 2D approximation (i.e., 2.5D classification). Figure 3 shows the several steps involved. Firstly, the RGB image is recorded by the Kinect™ camera (32-bits, VGA resolution) and then converted to the HSI space. Secondly, the depth information is collected by the Kinect™ sensor (11-bits) and displayed in grayscale (16-bits, VGA resolution). The distance information is applied to the input image due to the similar correspondence —once offset is calibrated— and the objects within the scene are then discriminated (8-bits, VGA resolution). The Canny filter is applied —on the binary image— to detect contours around the objects and finally, the result after removing the scene is shown in Figure 3e (32-bits, VGA resolution).

The Canny filter was used as a well suitable method to detect edges from objects within an image even under conditions of poor signal-to-noise ratios. The operation consists in processing images to avoid the influence of noise by using a Gaussian filter. As the edge of an object can be pointing to different directions, the Canny algorithm uses four filters to detect horizontal, vertical and diagonal lines. Thus, the angle and direction can be determined through several iterations based on edge detection operators.

D. Detection Algorithms

The process of object detection has been approached by combining two complementary algorithms focused on shape recognition and color matching. To this end, our method obtains a Hough Spectrum (HS) and a Color Spectrum (CS) of the objects within Figures 3d and 3e, respectively. Spectra are computed for both the reference object attempted to find and the input image from the Kinect™ sensor, and later compared by means of a cross-correlation operator to estimate solutions.

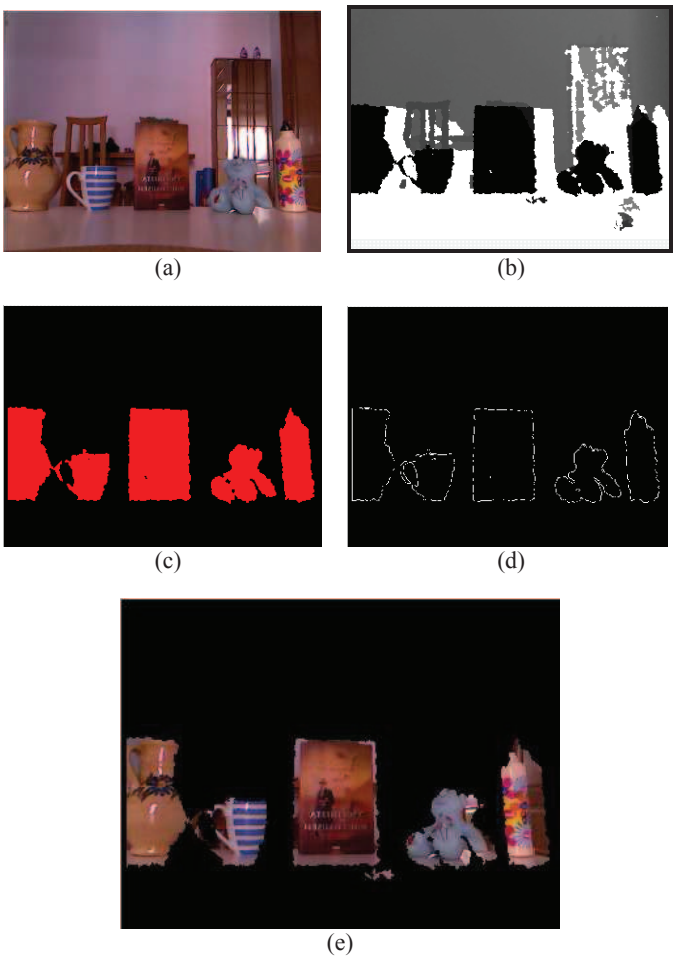


Figure 3. Image pre-processing technique by using Kinect™: (a) RGB image, (b) depth image, (c) binary image, (d) Canny filter, and (e) segmentation

Figure 4 shows a simple way of performing the HT algorithm in Matlab®. Notice that lines are numbered only for illustration purposes. We provide the input image already pre-processed and an edge detection function is applied (i.e., the Canny filter used herein). Then, the algorithm works as follows. Line 1 determines how many elements from the binary image belong to the object shape. Line 2 stands for the maximum size of the Hough matrix. Line 3 defines the angle interval of the lines crossing each point in the image space. Line 4 creates a two dimensional array consisting of the angles (θ) and the distances (ρ) from each line to the origin of the image plane. Finally, lines 5 to 12 perform the Hough Transform in whose inner for loop is located the accumulator.

The HT algorithm stands for a voting method which obtains — as a result — a histogram in the parameter space (θ, ρ). However, a formula to compare a reference object with an input image becomes necessary. The HS algorithm comes to provide a well-suited method —computed from the HT— due to its facilities for object recognition. HS is introduced as a global searching, multi-modal and non-iterative method able to work in unstructured environments [22]. In addition, HS is invariant to translations and rotated by rotations —therefore very robust to sensor or object variations— as well as provides

an acceptable response to limited information about the points forming a shape (e.g., partial occlusion and point truncation).

Although other existing algorithms may be used (e.g., likelihood), we propose one that exploits the already computed HT. The HS is computed by applying a translational invariant transformation (**g**) to the HT. Since the input space $i(\mathbf{s})$, $\mathbf{s} \in \mathcal{S}$ is mapped to the parameter space $\mathrm{HT}\{i\}(\mathbf{p})$, $\mathbf{p} \in \mathcal{P}$, the rigid transformation of the $\mathcal{S}$ space ($T = 0$, $R = \varphi$) is traduced —in practice— as a translation of the $\mathcal{P}$ space in the θ direction (i.e., along the x-axis of the Hough space):

$$\mathrm{HS}_{\mathbf{g}}\{i\}(\theta) = \mathbf{g}[\mathrm{HT}\{i\}(\theta,\cdot)] \tag{6}$$

$$i'(\mathbf{s}) = i(R\cdot\mathbf{s} + T) \tag{7}$$

$$\mathrm{HT}'(\theta,\rho) = \mathrm{HT}(\theta+\varphi,\rho) \tag{8}$$

$$\mathrm{HS}_{\mathbf{g}}\{i\}(\theta) = \mathrm{HS}_{\mathbf{g}}[i'](\theta+\varphi) \tag{9}$$

Many functions to define **g** could be used; so we will use the energy of a sequence $x[n]$ in the discrete domain as proposed in [22]:

$$\mathbf{g}[f] = \textstyle\sum_n |x[n]|^2 \tag{10}$$

where n is a finite length sequence depending on the bins used to define the HT histogram for $\theta = [0, \pi)$. Figure 5 depicts a simple script to perform the HS algorithm in Matlab®. Therein, lines 3 to 7 are used to compute the energy of a sequence as described in Eq. (10).

E. Classification Algorithm

The following step of the process consists in comparing the spectra of the objects obtained from the Kinect™ video and that spectra from a database. The HS stands for a ranking method (i.e., multi-modal solution) very suitable for cross-correlation of its data series (i.e., the input and the reference spectra). The cross-correlation operator is computationally efficient and provides a huge potential method to know the level of similarity regarding the objects to localize. Let **f** and **g** be discrete functions (e.g., the array of values of the HS from the sensor and the reference data), then the cross-correlation for a finite length sequence is defined as:

$$CC_{\mathbf{fg}}(n) = \sum_{m=-N}^{N} \mathbf{f}(m)\cdot\mathbf{g}(m+n) \tag{11}$$

where m and n stand for the delay between functions. That way, it is possible to estimate the offset between two spectra by computing the cross-correlation through Eq. (11). Figure 6 shows the algorithm to implement the cross-correlation operator in Matlab® (line 22). As a result, a set of hypotheses are produced based on local maxima which maximum value stands for the best fitness (lines 26 and 27).

Likewise, the color histograms for both the reference object and the input image from Kinect™ are obtained by counting of votes obtained for each component in the HSI space. This method has been implemented by high-end functions included in the IMAQ Vision module for

```
Input Elements after Canny operator (x, y); resolution of image from depth sensor (sx, sy)
Output Hough Matrix (HM)
1  totalpix = length(x);                          %gather all pixels with value to 1
2  maxrho = round(sqrt(sx^2 + sy^2));             %determine size for HM
3  interval = 5;                                  %define interval for theta
4  HM = zeros(2*maxrho,180/interval);             %define 2-dim. HM
5  for cnt = 1:totalpix                           %run loop for all of the pixels to 1
6  |  cnt2 = 1;                                   %initiate variable
7  |  for theta = 0:interval*pi/180:pi-pi/180     %turn line for pixel
8  |  |  rho = round(x(cnt).*cos(theta) + y(cnt).*sin(theta));
   |  |                                          %compute range to origin and round
9  |  |  HM(rho+maxrho,cnt2) = HM(rho+maxrho,cnt2) + 1;
   |  |                                          %compute Hough accumulator
10 |  |  cnt2 = cnt2 + 1;                        %increment variable
11 |  end                                        %end when all rotations are made for each pixel
12 end                                           %end when go through all pixels to 1 in shape
```

Figure 4. Pseudocode for the Hough Transform (HT)

```
Input Hough Matrix (HM)
Output Hough Spectrum (HS)
1 [s1,s2] = size(HM);                            %return size of Hough matrix
2 HS = zeros(1,s2);                              %define 2-dimensional HS
3 for i = 1:s1                                   %run each value of theta
4 |  for j = 1:s2                                %run each value of rho
5 |  |  HS(j) = HS(j) + HM(i,j)*HM(i,j);         %compute seq. energy
6 |  end                                         %end when run all values for rho
7 end                                            %end when run all values for theta
8 HS = HS / max(HS);                             %normalize
```

Figure 5. Pseudocode for the Hough Spectrum (HS)

```
Input HS for input image (HSi); HS for reference object (HSr)
Output Fitness; Local_Maxima
1  n = size(HSi,2);                              %return size of dimension 2 of HSi
2  CC = zeros(1,n);                              %define 2-dimensional cross-correlation
3  maxphi = 180/interval, minphi = 1;            %define range for spectra
4  mx = 0, my = 0;
5  for i = 1:n
6  |  mx = mx + HSi(i);
7  |  my = my + HSr(i);
8  end
9  mx = mx/n, my = my/n;
10 sx = 0, sy = 0;
11 for i = 1:n
12 |  sx = sx + (HSi(i) - mx) * (HSi(i) - mx);
13 |  sy = sy + (HSr(i) - my) * (HSr(i) - my);
14 end
15 denom = sqrt(sx*sy);                          %see line 24
16 for phi = minphi:1:maxphi
17 |  for theta = 1:n
18 |  |  delay = theta - phi; %def. similarity as a function of delay
19 |  |  if delay < 1
20 |  |  |  delay = maxphi + delay;
21 |  |  end
22 |  |  CC(phi) = CC(phi) + (HSi(theta)-mx)*(HSr(delay)-my);
   |  |                                          %compute cross-correlation of HSi and HSr
23 |  end
24 |  CC(phi) = CC(phi) / denom;  %normalize cross-correlation
25 end
26 Fitness = max(CC)                              %find best fitness
27 Local_Maxima = find(CC == fitness)*interval %find indexes of nonzero elements
```

Figure 6. Pseudocode for the cross-correlation of the HS

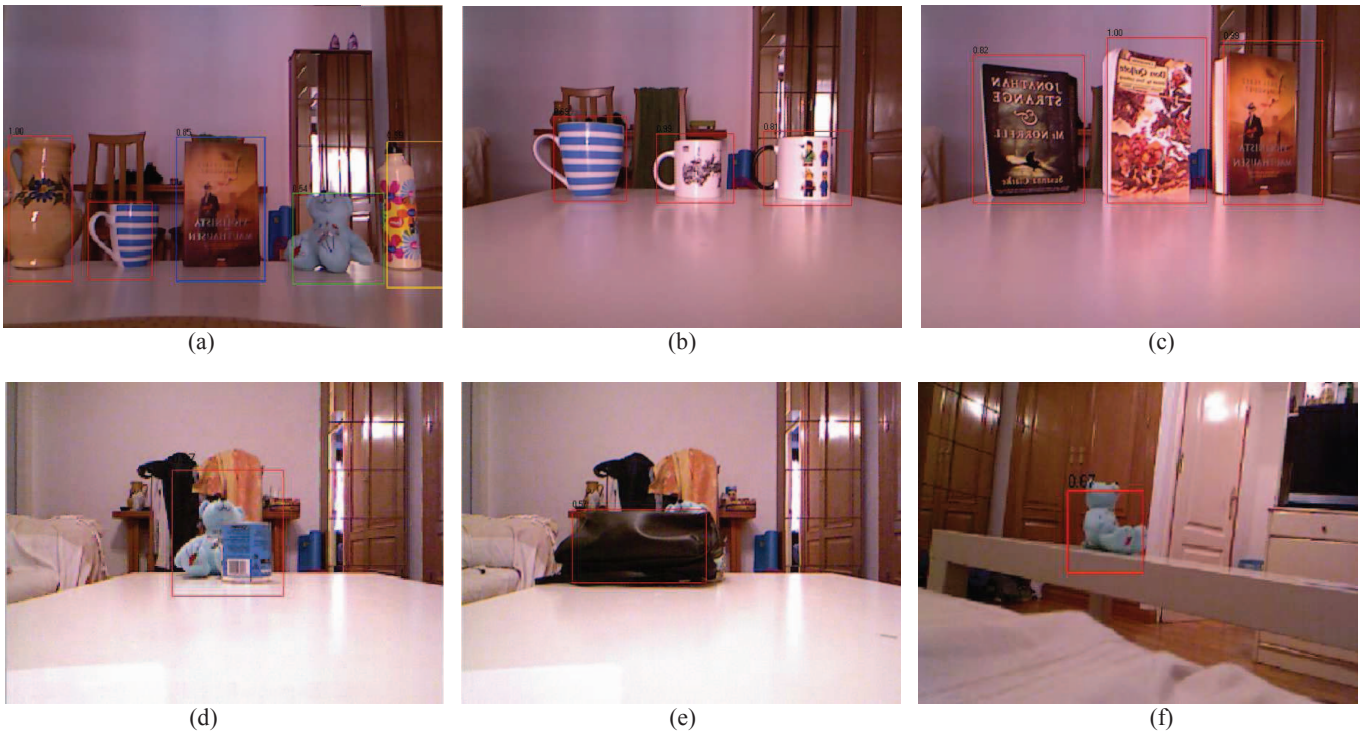


Figure 7. Results of the 2.5D classification algorithm considering different case studies: (a) recognition between different objects, (b) and (c) recognition between similar objects, (d) and (e) recognition under partial and total occlusion, and (f) recognition under perspective change

LabVIEW™. Specifically, the *IMAQ Color Learn* function has been used to extract the color features of an image and return the CS found in a ROI defined by the Canny filter (i.e., the object). Subsequently, the CS is normalized and the *IMAQ Color Match* function finds the match between the color content of multiple regions in the image and that defined by the CS reference.

Finally, the classification of objects is performed by combining the fitness scores from the HS algorithm and the CS function as follows:

$$\mathbf{w} = \alpha \cdot (\mathbf{w}_{CS}) + (1 - \alpha) \cdot (\mathbf{w}_{HS}) \quad (12)$$

where $\alpha : [0, 1] \in \mathbb{Q}$ stands for the contribution of color information to the object classification, and $\mathbf{w}_{CS}$ and $\mathbf{w}_{HS}$ are the fitness scores for the CS and HS, respectively. Depending on the object complexity (i.e., contour richness and color depth) a different α value may be set during the algorithm execution. Let $\mathbf{w}_{CS}$ and $\mathbf{w}_{HS}$ be functions computed as follows:

$$\mathbf{w}_{CS} = \max[\text{norm}(CC(CSi, CSr))] \quad (13)$$

$$\mathbf{w}_{HS} = \max[\text{norm}(CC(HSi, HSr))] \quad (14)$$

where CSi and CSr are the color spectra of the input image and the reference object being compared, and HSi and HSr are the Hough spectra of the input image and the reference object likewise.

V. Experimentation

This section shows a comprehensive study on the 2.5D classification algorithm considering four case studies: *i*) recognition between different objects, *ii*) recognition between similar objects, *iii*) recognition under occlusion, and *iv*) recognition under perspective change (Fig. 7). Experiments were completed with an Intel® Core™ i7 (2.6 GHz, 16 GB RAM) and evaluated by measuring the fitness score ($\mathbf{w}$) and processing time ($\boldsymbol{\tau}$) over 50 iterations.

With the aim of performing the first test, five objects with strong differences on shape and color were included in the same scene: vase, cup, book, bear, and bottle (Fig. 7a). The contribution of fitness was set to $\alpha = 0.5$, which means that both Hough and color information were considered in the same proportion. Table I shows the results for each series when comparing an object respect to the others. Results show an average fitness score of true positives between $\mathbf{w} = 92.1\%$ and 97.9%, whilst the false positive for the rest of the objects was observed between $\mathbf{w} = 44.5\%$ and 80.5% (the latter not shown in Table I). We found significant differences between positive and non-positive matches ($53.4\% \geq \Delta\mathbf{w} \geq 11.6\%$), where the closest false positive was due to the similarity in color spectrum between the vase and the book; this in turns depends on α selection. With the aim of obtaining a statistic for model comparison, a ROC analysis was carried out [27]. Figure 8 shows the area under the curve (AUC) for which a sensitivity of TPR = 0.941 and a specifity of TNR = 0.335 were obtained [28]. Moreover, the average execution time per iteration varied between $\boldsymbol{\tau} = 0.361 \pm .012$s and $0.404 \pm .032$s. Considering a frame rate of 30 fps for the Kinect™ sensor (32-bits, 640x480 pixels), this suggests that the 2.5D algorithm is feasible for applications of object classification with video information.

The second experiment was intended to study the algorithm's performance when objects with very similar characteristics shared the same scene. Tests were divided into two groups: a set of cups, and a set of books (Fig. 7b and 7c). On the one hand, we consciously chose for the first series two very similar cups and one with slightly different features. Notice that the object intended to detect is the cup at the middle (i.e., Cup 2). The correlation of both shape and color were studied, but focusing more on color differences because it was beforehand known that we were facing similar objects (α = 0.7). As shown in Table II, the two white cups obtained an average fitness score of $\mathbf{w} \geq 91.4\%$, whilst the blue cup —due to its slightly oval contour and blue stripes pattern— obtained an average fitness score of $\mathbf{w}$ = 75%. The ROC analysis found a sensitivity of TPR = 0.736 and a specifity of TNR = 0.461 (Fig. 8). Moreover, the experiment revealed that the execution time per iteration decreased. This suggests that the algorithm's computational cost improved —as expected— when fewer and similar objects had to be processed. On the other hand, the second series was focused on study the features of morphologically identical books. A classification with higher contribution in color fitness would have been trivial due to the higher information of book covers, so we decided to set α = 0.1 to test the algorithm's response. Although the selected book —the one at the middle— was successfully recognized with an average fitness score of $\mathbf{w}$ = 100%, the high similarity of the other books provided an estimate very close ($6.7\% \geq \Delta\mathbf{w} \geq 0.7\%$). Nonetheless, the ROC analysis obtained a sensitivity of TPR = 0.875 and a specifity of TNR = 0.421 (Fig. 8). This indicates that the algorithm is useful in multi-modal applications where several objects have to be simultaneously detected.

The third experiment was focused to study the algorithm's performance when objects were occluded in the scene. Tests were divided into two groups: partial occlusion and total occlusion with α = 0.5 (Fig. 7d and 7e). Firstly, a can was placed between the bear and the field of view of Kinect™ leaving approximately 30% of the bear completely hidden (Table III). We observed a response higher than $\mathbf{w}$ = 91% with an execution time of $\boldsymbol{\tau}$ = 0.152±.019s, which suggests the soundness of the algorithm even under conditions of partial occlusion. On the contrary, the second test showed a poor response of the algorithm under total occlusion. This result provides a helpful feedback and suggests that hypotheses with $\mathbf{w} \leq 60\%$ may be discarded —in these conditions— for a positive matching.

Finally, the fourth experiment consisted in studying the algorithm's behavior when an object was observed in a different perspective regarding the information hold in the database (Fig. 7f).We found that the fitness score strongly decreased as expected to $\mathbf{w}$ = 52.8% when the bear was captured in a lateral position (α = 0.5). However, some of the features —mainly to color and shape information to a lesser extent— restrained the response (Table IV). This means that in situations where the Hough Transform can be penalized, the algorithm may be successfully compensated with color spectrum.

TABLE I. RESULTS ON FITNESS AND PROCESSING TIME FOR OBJECT RECOGNITION BETWEEN DIFFERENT ONES

Object	$\mathbf{w}_{max}$	$\mathbf{w}_{min}$	$\mathbf{w} \pm \mathbf{dev}$	τ_{max}	τ_{min}	$\tau \pm$ dev
Vase	0.964	0.840	0.932±.024	0.384	0.353	0.361±.012
Cup	0.987	0.974	0.979±.017	0.413	0.366	0.396±.012
Book	0.970	0.950	0.955±.042	0.541	0.373	0.404±.032
Bear	0.944	0.814	0.921±.044	0.568	0.371	0.392±.032
Bottle	0.973	0.941	0.967±.015	0.538	0.374	0.395±.030

TABLE II. RESULTS ON FITNESS AND PROCESSING TIME FOR RECOGNITION BETWEEN SIMILAR OBJECTS

Object	$\mathbf{w}_{max}$	$\mathbf{w}_{min}$	$\mathbf{w} \pm \mathbf{dev}$	τ_{max}	τ_{min}	$\tau \pm$ dev
Cup 1	0.799	0.669	0.75±.030	0.345	0.231	0.264±.020
Cup 2	0.960	0.921	0.951±.042			
Cup 3	0.960	0.845	0.914±.009			
Book1	0.990	0.820	0.933±.042	0.488	0.278	0.327±.031
Book2	1	0.998	1±.004			
Book3	0.997	0.979	0.993±.030			

TABLE III. RESULTS ON FITNESS AND PROCESSING TIME FOR RECOGNITION UNDER OCCLUSION

Object	$\mathbf{w}_{max}$	$\mathbf{w}_{min}$	$\mathbf{w} \pm \mathbf{dev}$	τ_{max}	τ_{min}	$\tau \pm$ dev
Bear (30%)	0.931	0.893	0.918±.006	0.224	0.114	0.152±.019
Bear (95%)	0.670	0.563	0.582±.027	0.472	0.228	0.281±.008

TABLE IV. RESULTS ON FITNESS AND PROCESSING TIME FOR RECOGNITION WITH CHANGE OF PERSPECTIVE

Object	$\mathbf{w}_{max}$	$\mathbf{w}_{min}$	$\mathbf{w} \pm \mathbf{dev}$	τ_{max}	τ_{min}	$\tau \pm$ dev
Bear	0.579	0.425	0.528±.034	0.327	0.127	0.172±.020

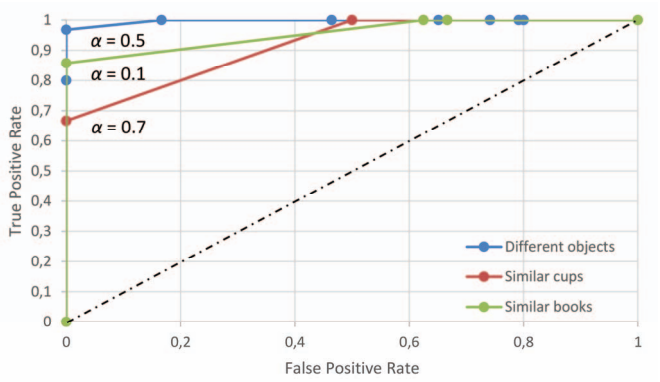


Figure 8. ROC plots produced for experiments depicted in Fig. 7a-7c

VI. CONCLUSSIONS

This paper presented a 2.5D classification algorithm —based on Hough Transform and color spectra— for object recognition by means of Kinect™. The aim was to take advantage of the depth sensor facility: *i*) to automate the image pre-processing operations for the subsequent segmentation regardless of arbitrary scenes (i.e., particle cleaning, hole filling, particle eroding, and object dilating) and *ii*) to simplify the generalized HT for 3D object recognition from video information. To this end, we addressed a comparative analysis on different case studies for recognition between different objects, similar objects, occlusion, and

perspective change —extensible to others— by analyzing representative performance metrics and ROC curves.

On the one hand, we found a high fitness score and sensitivity (**w** = 92.1%, TPR = 0.941, and TNR = 0.335) in sceneries including objects to be classified among different ones (i.e., strong differences on shape and color). In other sceneries including objects intended to be classified between similar shapes, we also found a high fitness score with similar sensitivity and specifity (up to **w** = 100%, TPR = 0.875, and TNR = 0.421). This suggests that the algorithm is suitable in multi-modal applications where an object have to be detected between others within the same scene. In mono-modal sceneries, the algorithm behaved very robustly in conditions of partial occlusion (30%), thus keeping its success rate (**w** > 92%). Although the response worsened for objects experiencing a total occlusion (**w** ~58%) and a perspective change (**w** ~52%) as expected for the HT, the tests suggest that the algorithm can be strongly compensated with color information in situations where the HT is penalized by lack of information on shape. On the other hand, the methodology followed to size the number of points and lines defining the object shapes demonstrated to play a key role, thus optimizing the execution time for video streaming purposes with P = [50, 100] and $\Delta\psi$ = [5, 10]. The algorithm showed different computational cost depending on the number of objects within the scene and the complexity of their shapes. In general, the algorithm responded better in mono-modal situations where average times were better than in multi-modal searching schemes.

Despite of the advantages of Kinect™, the camera involved limiting issues: *i*) shorter range, *ii*) narrower horizontal FOV, and *iii*) worse depth accuracy compared with laser range finders. Moreover, we observed from the experience that the pseudorandom beam pattern projected over objects would cause —in the same series— a jitter on fitness scores up to 9%. In addition, the distance threshold used to discriminate objects from the scene stands for a user-defined parameter that limits the automated operation for autonomous systems. In this sense, future efforts and experiments will be addressed to generalize the conclusions and enhance the algorithm's capabilities for 3D object classification.

REFERENCES

[1] P. Azad, T. Asfour, and R. Dillmann, "Combining Appearance-based and Model-based Methods for Real-Time Object Recognition and 6D Localization," in Proc. Intern. Conf. Intelligent Robots and Systems (IEEE/RSJ), pp. 5339–5344, 2006.

[2] N. Bayramoğlu, and A.A. Alatan, "Segmentation driven semantic information inference from 2.5D data," in IEEE 17th Signal Processing and Communications Applications Conf. (SIU), pp. 604–607, 2009.

[3] P. Piccinini, A. Prati, and R. Cucchiara, "Real-time object detection and localization with SIFT-based clustering," Image and Vision Computing, vol. 30, pp. 573–587, 2012.

[4] T.J. Mateo Sanguino, J.M. Andújar Márquez, T. Carlson, and J.d.R. Millán, "Interaction and evaluation of an augmented virtuality assistance system for teleoperated robots," in IEEE Intern. Symp. Robotic and Sensors Environments (ROSE), pp. 19–24, 2012.

[5] T.J. Mateo Sanguino, J.M. Andújar Márquez, T. Carlson, and J.d.R. Millán, "Improving Skills and Perception in Robot Navigation by an Augmented Virtuality Assistance System," J. Intelligent & Robotic Syst., vol. 76, no. 2, pp. 255-266, 2014.

[6] T.J. Mateo Sanguino, F. Ponce Gómez, "Evaluation of a Dispersion-based Adaptive Strategy using Kinect™ and Dynamic Particle Filter," in 17th Intern. Conf. Information Fusion (Fusion), pp. 1–8, 2014.

[7] B. Ommer, "The Role of Shape in Visual Recognition," Shape Perception in Human and Computer Vision, pp. 373–385, 2013.

[8] S. Sternig, T. Mauthner, A. Irschara, P.M., Roth, and H. Bischof, "Multi-camera multi-object tracking by robust hough-based homography projections," in IEEE Intern. Conf. Computer Vision Workshops (ICCV), pp. 1689–1696, 2011.

[9] H. Wu, G. Yoshikawa, T. Shioyama, S. Lao, and M. Kawade, "Glasses frame detection with 3D Hough transform," in Proc. 16th Intern. Conf. Pattern Recognition, vol. 2, pp. 346–349, 2002.

[10] K. Khoshelham, "Extending Generalized Hough Transform to Detect 3D Objects in Laser Range Data," in ISPRS Workshop on Laser Scanning and SilviLaser, vol. XXXVI, no. 3, pp. 206–210, 2007.

[11] J. Stowers, and M. Hayes, "Quadrotor Helicopter Flight Control Using Hough Transform and Depth Map from a Microsoft Kinect Sensor," in Conf. Machine Vision Applications (MVA), pp. 352–356, 2011.

[12] D. Dube, and A. Zell, "Real-time plane extraction from depth images with the Randomized Hough Transform," in IEEE Intern. Conf. Computer Vision Workshops (ICCV), pp. 1084–1091, 2011.

[13] S. Zug, F. Penzlin, A. Dietrich, T.T. Nguyen, and S. Albert, "Are laser scanners replaceable by Kinect sensors in robotic applications?," in IEEE Intern. Symp. Robotic and Sensors Environments (ROSE), pp. 144–149, 2012.

[14] P.V.C. Hough, "Method and means for recognizing complex patterns," U.S. Patent 3,069,654, 1962.

[15] R.O. Duda, and P.E. Hart, "Use of the Hough Transformation to Detect Lines and Curves in Pictures," Comm. ACM, vol. 15, pp. 11–15, 1972.

[16] D.H. Ballard, "Generalizing the Hough Transfor to Detect Arbitray Shapes," Pattern Recognition, vol. 13, no. 2, pp. 111–122, 1981.

[17] M. Zhang, B. Jaggi, and B. Palcic, "Hough spectrum and geometric texture feature analysis," in Proc. 12th IAPR Intern. Conf. Pattern Recognition, vol. 2, pp. 583–585, vol. 2, 1994.

[18] R.S. Stephens, "Probabilistic approach to the Hough transform", Image and Vision Computing, vol. 9, no. 1, pp. 66–71, 1991.

[19] B. Leibe, A. Leonardis, and B. Schiele, "Robust object detection with interleaved categorization and segmentation," Intern. J. Computer Vision, vol. 77, no. 1-3, pp. 259–289, 2008.

[20] A. Lehmann, B. Leibe, and L. V. Gool, "PRISM: Principled Implicit Shape Model," in British Machine Vision Conf. (BMVC), 2010.

[21] J. Knopp, M. Prasad, G. Willems, R. Timofte, and L. Van Gool, "Hough Transform and 3D SURF for Robust Three Dimensional Classification," Lecture Notes in Computer Science, vol. 6316, pp. 589–602, 2010.

[22] A. Censi, L. Iocchi, and G. Grisetti, "Scan Matching in the Hough Domain," in Proc. IEEE Int. Conf. Robotics and Automation (ICRA), pp. 2739–2744, 2005.

[23] A. Censi, and S. Carpin, "HSM3D: feature-less global 6DOF scan-matching in the Hough/Radon domain," in IEEE Intern. Conf. Robotics and Automation (ICRA), pp. 3899–3906, 2009.

[24] G. Grisetti, L. Iocchi, and D. Nardi, "Global hough localization for mobile robots in polygonal environments," in Proc. of IEEE Intern. Conf. Robotics and Automation (ICRA), vol. 1, pp. 353–358, 2002.

[25] OpenKinect, Tech. Report, 2011. http://openkinect.org; https://github.com/OpenKinect/libfreenect

[26] C. Norman, D. Clark, and B. Cotter, "Kinesthesia - a Kinect based rehabilitation and surgical analysis system," Tech. Report, 2012. http://decibel.ni.com/content/docs/DOC-20973

[27] T. Fawcett, "An Introduction to ROC Analysis," Pattern Recognition Letters, vol. 27, no. 8, pp. 861–874, 2006.

[28] P.A., Flach, J. Hernandez-Orallo, and C. Ferri, "A coherent interpretation of AUC as a measure of aggregated classification performance," in Proc. 28th Int. Conf. Machine Learning (ICML), pp. 657–664, 2011.