Recognition of Confusing Objects for NAO Robot

Thanh-Long Nguyen, Didier Coquin(✉), and Reda Boukezzoula

LISTIC Laboratory, Polytech Annecy-Chambery,
University of Savoie Mont-Blanc, 74940 Annecy-le-vieux, France
{thanh-long.nguyen,didier.coquin,reda.boukezzoula}@univ-smb.fr

Abstract. Visual processing is one of the most essential tasks in robotics systems. However, it may be affected by many unfavourable factors in the operating environment which lead to imprecisions and uncertainties. Under those circumstances, we propose a multi-camera fusing method applied in a scenario of object recognition for a NAO robot. The cameras capture the same scenes at the same time, then extract feature points from the scene and give their belief about the classes of the detected objects. Dempster's rule of combination is then used to fuse information from the cameras and provide a better decision. In order to take advantages of heterogeneous sensors fusion, we combine information from 2D and 3D cameras. The results of experiment prove the efficiency of the proposed approach.

Keywords: Object recognition · NAO robot · Uncertainty · Evidence theory · Camera fusion

1 Introduction

With the very fast development of high technologies, robotics is now more and more important to human life. Specifically, vision processing is one of the most focused areas, which helps a robot increase its ability to learn in explored environments. This work considers a scenario in which a NAO robot can recognize previously learned objects by fusing multi-camera to increase the quality of recognition and reduce uncertainties and imprecisions. We first have a look at how the other works have dealt with object recognition, then propose a solution for the considered case.

In fact, the problem of recognizing an object has been addressed for several decades. The number of methodologies is huge up to now; each of them tried to prove their strengths and overcame the weaknesses of the preceding solutions. For instances, Berg et al. [1] used Geometric Blur approach for feature descriptors and proposed an algorithm to calculate the correspondences between images. The query image was then classified according to its lowest cost of correspondence to the sample images. Besides that, Ling and Jacobs [2] introduced the term "inner-distance"as the length of the shortest path between

J.P. Carvalho et al. (Eds.): IPMU 2016, Part I, CCIS 610, pp. 262–273, 2016.
DOI: 10.1007/978-3-319-40596-4_23

landmark points within the shape silhouette. The inner-distance was used to build shape representations and they helped to obtain good matching results. For some texture-based approaches, [3] proposed a texture descriptor based on Random Sets and experimentally showed that it outperformed the co-occurrence matrix descriptor. Decision tree induction was used in that work to learn the classifier. Another example can be found in [4] where color and texture information were both used in an agricultural scenario to recognize fruits. On the other hand, some context-based methods like [5–7] considered contextual information surrounding the target objects. These information come from the interaction among objects in the scene and they help to disambiguate appearance inputs in recognition tasks. Similarly successful, the methods based on local feature description like SIFT [8] and SURF [9] have received many positive evaluations and have been widely applied [10–13]. SIFT extracts keypoints from object to build feature vectors. We then calculate the matching (using Euclidean distance) between an input object and the ones in database to find the best candidate class. After that, the agreement on the object and its location, scale, and orientation are determined by using a hash table implementation of the Generalized Hough Transform. In a different manner, SURF uses a blob detector based on the Hessian matrix to find interest points, then it calculates the descriptor by using the sum of Haar wavelet responses. Finally, by comparing the descriptors obtained from different images, the matching pairs can be found.

For the purpose of collecting spatial information about the detected objects, and avoiding imprecision of 2D images under non-ideal lighting conditions like outdoor environment, some works concentrated on 3D object recognition. In [14], an extended version of the Generalized Hough Transform was used in 3D scenes. Each point in the input cloud votes for a spatial position of the object's reference point and the accumulating bin with the maximum votes indicates an instance of the object in the scene. In [15,16], the 3D extensions of SIFT and SURF descriptor also gave positive recognition results. In addition, Zhong [17] introduced a new 3D shape descriptor called Intrinsic Shape Signature to characterize a local/semi-local region of a point cloud. This descriptor uses a view-independent representation of the 3D shape to match shape patches from different views directly, and a view-dependent transform encoding the viewing geometry to facilitate fast pose estimation. On the contrary, [18,19] considered the use of point pairs for the description and the feature matching is then done by implementing a hash table. Recently, the SHOT descriptor [20] has emerged as an efficient tool for 3D object recognition [21,22]. Indeed, the descriptor encodes histograms of basic first-order differential entities (i.e. the normals of the points within the support), which are more representative than plain 3D coordinates about the local structure of the surface. After defining an unique and robust 3D local reference frame, it is possible to enhance the discriminative power of the descriptor by concerning the location of the points within the support, from that describing a signature.

It is clear that all of the above mentioned approaches have experimentally shown good results in object recognition. Nevertheless, many of them did not

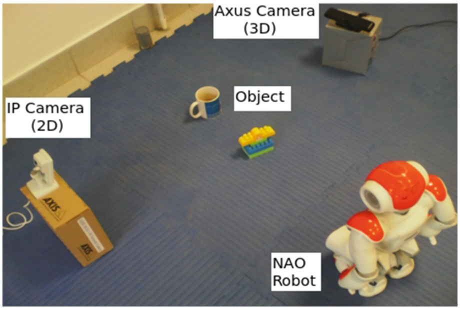


Fig. 1. Multi-camera helps NAO robot recognize objects.

focus on the problem of uncertainty and imprecision which might come from the quality of data and sensors, the lighting conditions, the viewing angles to the objects and particularly, *the similarity among confusing objects*. Therefore, in this work we propose to use multi-camera to recognize objects which have many similarities. The proposed method is implemented in a NAO robot due to our development in a robotics project, however it is not restricted to any other kind of vision-based platform. In order to take advantage of both 2D and 3D recognitions, we use not only a 2D camera of the NAO robot but also another 2D IP Axis camera and another 3D Axus camera; Fig. 1 shows the multi-camera environment where the robot is requested to recognize objects. The fusion of these three heterogeneous sensors brings additional advantages for each one because the NAO camera and the IP camera give characteristics about the 2D features of the detected objects whereas the Axus camera provides depth information. We propose an evidential classifier based on Dempster-Shafer theory (or Evidence theory) [23] for each camera, then we combine them in decision level in order to give more reasonable results of object recognition.

The outline of the paper is as follow. First, we describe our approach step-by-step in Sect. 2, then we give an illustrative example in Sect. 3. Section 4 shows our results of experiment to validate the approach, finally Sect. 5 gives the conclusion.

2 Our Recognition Approach

2.1 An Evidential Classifier for Each Camera

Processing Flow: Figure 2 shows the flow of classification by each camera. First, an input image in 2D or 3D form is captured based on the type of camera sensor. For the NAO camera and the IP camera (2D), the input data is 640×480 images; for the Axus camera (3D), the input images are in form of Point Cloud since we implement 3D processing by using the PCL library [24]. To focus on the classification, we use only one instance of object appearing in the captured scene.

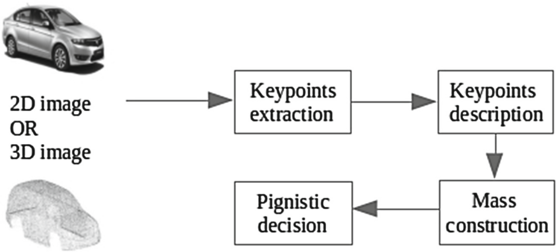


Fig. 2. Evidential classifier for each camera

First, interest points (or key points) of the object in the scene are extracted. In an image, an interest point can be described as a point that has rich information about local image structure around it, and these points characterize well the patterns in the image. After that, we use methods of descriptor to build a feature vector for each interest point. We use the word "feature points" for the interest points that have been described by the descriptor. The methods of descriptors used in this work are SURF [9] for 2D data and SHOT [20] for 3D data according to their strong properties as explained above. From the set of feature points acquired, we build a mass function which describes the camera's degree of belief about the classes of detected object. Thereafter, a decision is made by choosing the class with the maximum pignistic probability. The processing flow is described with more detail later.

Evidence Theory in the Scenario: Suppose the robot has to recognize an object that can be only in one of N classes, i.e. the space of discernment is:

$$\Omega = \{O_1, O_2, ..., O_N\} \tag{1}$$

Then we have the power set which contains the subsets of the space of discernment:

$$2^{\Omega} = \{\{\emptyset\},\ \{O_1\},\ \{O_2\},\ ...,\ \{O_N\},\ \{O_1 \cup O_2\},\ ...,\ \{O_1 \cup O_N\},\ ...,\ \{\Omega\}\} \tag{2}$$

In Evidence Theory, we have to determine a mass function which describes the degree of belief for all possible hypotheses in the power set. This function satisfies:

$$\begin{gathered} m : 2^{\Omega} \rightarrow [0,1] \\ \sum_{H \in 2^{\Omega}} m(H) = 1 \end{gathered} \tag{3}$$

To illustrate the proposed approach, we consider a simple case in Fig. 3 where we suppose that there are three classes of object: A, B and C. For the sake of explanation, we assume that we have only one training image for each class. With an input image which contains a set X of feature points of object, our mission is to decide the appropriate class for X. The basic idea is that each

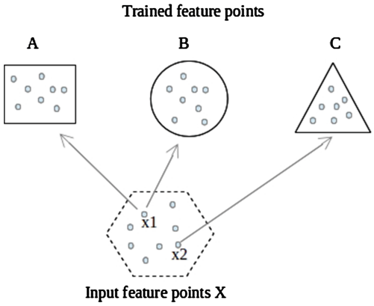


Fig. 3. Illustration of the idea. Each input feature point votes for a hypothesis.

feature point $x_i \in X$ will vote for a hypothesis $H \in 2^{\Omega}$ based on its matching to the training images. In Fig. 3, the feature point x_1 matches both images of class A and B, so we accumulate one vote for the hypothesis $H = \{A \cup B\}$. Similarly, the feature point x_2 votes for $H = \{C\}$. By doing the same principle for all the feature points of X, we can construct all elements of the mass function after doing a normalization step. Due to the need of clear explanation in a scientific work, the step of defining the matching and constructing mass function will be mathematically described thereafter.

Construction of Mass and Decision: First, let us denote $\varDelta(p_i, p_j)$ the normalized distance between two feature points p_i and p_j; the shorter the distance is, the more similar the two feature points are.

$$\varDelta(p_i, p_j) \in [0, 1] \tag{4}$$

In order to decide the matching between a feature point p_i^X of an input image X (X can also be understood as the set of feature points for the input image) and a training image M whose class is $O_j \in \Omega$, we use the idea in [25]. We will find the two nearest neighbours of p_i^X in M, called $p_{i_1}^M$ and $p_{i_2}^M$ (the feature points in M are previously extracted in the training phase). We suppose that $p_{i_1}^M$ is closer to p_i^X than $p_{i_2}^M$ i.e. $\varDelta(p_i^X, p_{i_1}^M) \leq \varDelta(p_i^X, p_{i_2}^M)$. After that, we define a matching function between the feature point p_i^X of an input image X and the model M :

$$\delta(p_i^X, M) = \begin{cases} 1, & \text{if } \varDelta(p_i^X, p_{i_1}^M) \leq \alpha \text{ and } \frac{\varDelta(p_i^X, p_{i_1}^M)}{\varDelta(p_i^X, p_{i_2}^M)} \leq \beta \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

where α and β are two user-defined parameters such that $0 \leq \alpha, \beta \leq 1$. The former guarantees that the distance between p_i^X and its most similar feature point found in M is small enough whereas the latter helps to avoid false matching. In this work, we choose $\beta = 0.8$ as suggested in [25], and we add $\alpha = 0.25$ in order to reduce noise. Indeed, these two parameters help us to find a strong and distinctive matching between the feature point p_i^X and its closest feature point in M. If $\delta(p_i^X, M) = 1$, we then say that p_i^X is matched to the training image M, i.e. matched to the class $O_j \in \Omega$ of M and vice versa. In the same way, we can find all the matches of the feature points in the input image X to the training image M.

For now, we define the *matching between* X *and the class* O_j by considering all the matches between feature points p_i^X in X and the class O_j. In the case that the class O_j has several training images M_k, we choose the training image M_{max} that has the maximum number of matches to X according to Eq. (5).

$$\delta^{max}(p_i^X, O_j) = \delta(p_i^X, M_{max}) \tag{6}$$

Table 1 shows an example illustrating the matches between input feature points and the output classes. A cell $c(p_i^X, O_j)$ implies the matching between the feature point p_i^X of X and the class O_j, $i = 1, 2, ...R_X$ - number of feature points in X, $j = 1, 2, ...N$ - number of classes. If the cell is red, it means that the feature point p_i^X matches the class O_j (i.e. $\delta^{max}(p_i^X, O_j) = 1$), otherwise not matched.

After we determine the matching between the input feature points and the output classes, we can construct the mass function as follow. Each feature point p_i^X will vote for a hypothesis in the power set such that the hypothesis is composed of the classes that match p_i^X. Mathematically, let's define a hypothesis-voted function that calculates the accumulated votes for each hypothesis:

$$accVote(X, H) = \sum_{p_i^X \in X} \phi(p_i^X, H), \quad H \in 2^{\Omega} \tag{7}$$

where $\phi(p_i^X, H)$ is a function indicating the matching between the feature point p_i^X and every element class in H:

$$\phi(p_i^X, H) = \begin{cases} 1, & \text{if } \sum_{O_j \in H} \delta^{max}(p_i^X, O_j) = |H| \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

where $|H|$ be the cardinality of H and $\delta^{max}(p_i^X, O_j)$ was already explained above. Indeed, $\phi(p_i^X, H)$ indicates whether a feature point p_i^X matches every element class in the hypothesis H or not, and $accVote(X, H)$ calculates the number

Table 1. Matching between the feature points of input image X and the classes

	p_1^X	p_2^X	p_3^X	...	$p_{R_X}^X$
O_1	🟥			...	
O_2		🟥		...	🟥
O_3	🟥		🟥	...	🟥
...	...	...	...	...	...
O_N		🟥		🟥	

of feature points in X that matches every element class in H. After that, we calculate the mass function based on the hypothesis-voted function:

$$m^X(H) = \frac{accVote(X,H)}{G^X} \tag{9}$$

where G^X is the normalization factor that guaranties the condition in Eq. (3):

$$G^X = \sum_{H \in 2^{\Omega}, H \neq \emptyset} accVote(X,H) \tag{10}$$

It is worth noting that, in this work we assume that the class of object in the input image X is only in Ω, so we put $m^X(\emptyset) = 0$.

Once we have constructed the mass function, we can give decision about the class of the object. Since the maximum of belief is too pessimistic and the maximum of plausibility is too optimistic, we choose the class which has the maximum pignistic probability [26]:

$$BetP^X(O_j) = \frac{1}{1 - m^X(\emptyset)} \sum_{O_j \in H} \frac{m^X(H)}{|H|} \tag{11}$$

2.2 Fusion of Cameras

Base on the Evidence theory, each camera gives a decision about the classification of the detected object. In addition, by using Dempster's rule of combination [23], we can integrate information from multi-camera in order to give a better decision. Usually, the rule is defined for two sources, however it is enough to ensure a trivial extension to many sources due to its associativity and commutativity:

$$m_{comb}(\emptyset) = 0$$

$$m_{comb}(H) = \frac{\sum_{H_1 \cap H_2 \cap \ldots \cap H_S = H} m_1(H_1) m_2(H_2) \ldots m_S(H_S)}{1 - K}, H \in 2^{\Omega}, H \neq \emptyset \tag{12}$$

where S is the number of information source (i.e. number of cameras, 3 in this experiment) and:

$$K = \sum_{H_1 \cap H_2 \cap \ldots \cap H_S = \emptyset} m_1(H_1) m_2(H_2) \ldots m_S(H_S) \tag{13}$$

Finally, the decision about the class of the detected object can be made by using pignistic probability as in Eq. (11).

3 Illustrative Example

In this section, we provide an example to illustrate the proposed approach. Suppose that we want the robot to recognize an object in a captured scene with three classes in the space of discernment, that means:

$$\Omega = \{O_1, O_2, O_3\} \tag{14}$$

so there are 8 possible hypotheses in the power set:

$$2^{\Omega} = \{\{\emptyset\}, \{O_3\}, \{O_2\}, \{O_2 \cup O_3\}, \{O_1\}, \{O_1 \cup O_3\}, \{O_1 \cup O_2\}, \{\Omega\}\} \tag{15}$$

For simplicity, we suppose that for each class, we have only 1 training image. Assuming that the NAO camera captures the scene X and it found 10 feature points in the input image X_{NAO}. For each of those input feature points, we find two nearest neighbours feature points in each training image. After that, we use Eqs. (4), (5), and (6) to construct the matching between the input image and each class. Table 2 shows an example of the matching found. Each cell describes the matching between a feature point and a class; if $\delta^{max}(p_i^{X_{NAO}}, O_j) = 1$, the cell is red, otherwise white. The last row indicates the hypothesis voted by the associating feature point.

Table 2. Matching between the input image X^{NAO} and the classes

	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	p_9	p_{10}
O_1	🟥		🟥	🟥			🟥		🟥	
O_2		🟥			🟥	🟥			🟥	🟥
O_3				🟥			🟥	🟥		🟥
Vote for:	O_1	O_2	O_1	$O_1 \cup O_3$	O_2	O_2	$O_1 \cup O_3$	O_3	$O_1 \cup O_2$	$O_2 \cup O_3$

From Table 2, we have determined the strength of each hypothesis in the power set. Table 3 then shows the accumulated vote for each hypothesis which is calculated by Eqs. (7) and (8). Each cell in the table is the value of $\phi(p_i^{X_{NAO}}, H), H \in 2^{\Omega}$. Remind that if $\phi(p_i^{X_{NAO}}, H) = 1$, it means that the feature point $p_i^{X_{NAO}}$ votes for the hypothesis H. According to Eq. (10), we have $G^{X_{NAO}} = \sum accVote = 1 + 3 + 1 + 2 + 2 + 1 + 0 = 10$. From these information, we calculate the mass values as in the last column by using Eq. (9).

After that, we assume that we use not only the NAO camera but also another IP camera (2D) and another Axus camera (3D). By doing the same steps, we can obtain two mass vectors output from the two additional sensors. Table 4 shows example values of these mases. Additionally, we also calculate the combination of the masses using Dempster's rule (m_{comb}) and transform it to the pignistic probability ($BetP$) for each of singleton hypothesis. The last column is the final decision from the fusion of three cameras, which recognizes that the detected object belongs to the class O_1.

Table 3. Accumulated vote for each hypothesis

$H \in 2^{\Omega}$	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	p_9	p_{10}	accVote	Mass value
$\emptyset$	0	0	0	0	0	0	0	0	0	0	0	0.00
O_3	0	0	0	0	0	0	0	**1**	0	0	1	1/10
O_2	0	**1**	0	0	**1**	**1**	0	0	0	0	3	3/10
$O_2 \cup O_3$	0	0	0	0	0	0	0	0	0	**1**	1	1/10
O_1	**1**	0	**1**	0	0	0	0	0	0	0	2	2/10
$O_1 \cup O_3$	0	0	0	**1**	0	0	**1**	0	0	0	2	2/10
$O_1 \cup O_2$	0	0	0	0	0	0	0	0	**1**	0	1	1/10
Ω	0	0	0	0	0	0	0	0	0	0	0	0/10

Table 4. Mass values from there camera sensors

Hypothesis	m_{NAO}	m_{IP}	m_{Axus}	m_{comb}	$BetP$	Decision
$\emptyset$	0.00	0.00	0.00	0.00		
O_3	0.10	0.23	0.21	0.22	0.23	
O_2	0.30	0.17	0.12	0.26	0.27	
$O_2 \cup O_3$	0.10	0.08	0.00	0.00		
O_1	**0.20**	**0.32**	**0.09**	**0.49**	**0.50**	$\boldsymbol{O_1}$
$O_1 \cup O_3$	0.20	0.13	0.13	0.02		
$O_1 \cup O_2$	0.10	0.00	0.39	0.01		
Ω	0.00	0.07	0.06	0.00		

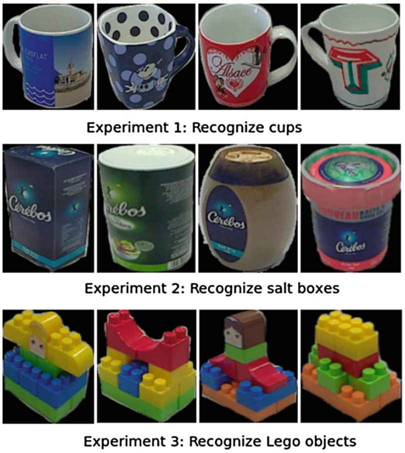
4 Experiments

As mentioned previously, the concentration of this work is how to resolve uncertainties and imprecisions during the object recognition process of the NAO robot. For that reason, we did three experiments, each of them contains a set of confusing objects as shown in Fig. 4. In the first set, there are 4 cups which can cause uncertainty in their spatial structures for the 3D camera to recognize. Conversely, the second experiment contains 4 boxes that have similar brand information on their surface, which may limit the recognition of the 2D cameras. Finally, we tested with 4 Lego bricks which are considered to have difficulties for both 2D and 3D cameras, in the third experiment.

For the training phase, we trained two images for each object with each camera in different view points. We then manually removed the background in these images in order to have only the model objects. For the test phase, NAO robot is requested to recognize an object appearing in front of it and say the result to human. The two cameras (IP and Axus) are on the two sides of the robot to help it improve the recognition. These three cameras capture the scene at the same time whenever the robot wants to recognize the object in the scene.

To focus on the work of recognition, the image region containing the object is restricted in order to avoid the noises in scene. For each of the three experiments, we did 32 recognition tests with different objects of 4 classes (so 8 tests for each object). The tested objects were turned around and put in different angles to the cameras in each test for the reason of challenging uncertainty.

Table 5 shows the results of experiment which is the comparison between the recognition rate of each camera (using the proposed classifier individually) and the fusion of three cameras. Remind that the rate for each camera cannot be high due to the confusing between similar objects and the objects are turned around each time of test. The fifth column is the result when we fuse the three cameras by using a simple voting based on majority: each camera gives its own recognition result based on the proposed classifier, then we choose the output class that is voted by the largest number of cameras. The last column shows the result of using Dempster-Shafer combination for the three cameras, which outperforms the majority voting to improve the recognition rate in average.


Experiment 1: Recognize cups

Experiment 2: Recognize salt boxes

Experiment 3: Recognize Lego objects

Fig. 4. Confusing objects used in the experiment

Table 5. Experiment result

Camera	NAO (2D)	IP (2D)	Axus (3D)	Majority voting fusion	Dempster-Shafer fusion
Experiment 1	78 %	88 %	75 %	100 %	**97** %
Experiment 2	72 %	72 %	91 %	91 %	**97** %
Experiment 3	59 %	59 %	69 %	72 %	**84** %
Average	69.67 %	73 %	78 %	87.67 %	**92.67**%

5 Conclusion

The work in this paper focuses on how to resolve uncertainties and imprecisions in object recognition for a NAO robot. Since the robot may face difficulties during its visual operation due to lighting conditions, viewing angles and the quality of camera, we propose to add more cameras in order to improve the recognition rate. Each camera extracts feature points from the captured scene, then provides a mass function based on the matching between the input and the training images. After that, Dempster's rule of combination is used to fuse information from these cameras. As can be seen, the approach is generalized for both 2D and 3D cameras, and the experiment work gives positive results, which prove the advantage of the fusion. Our future works will consider a more complex scenario where the NAO robot can build a semantic map based on the recognition approach used in this work.

References

1. Berg, A.C., Berg, T.L., Malik, J.: Shape matching and object recognition using low distortion correspondences. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005, vol. 1, pp. 26–33. IEEE (2005)
2. Ling, H., Jacobs, D.W.: Shape classification using the inner-distance. IEEE Trans. Pattern Anal. Mach. Intell. **29**(2), 286–299 (2007)
3. Perner, P.: Cognitive aspects of object recognition-recognition of objects by texture. Procedia Comput. Sci. **60**, 391–402 (2015)
4. Arivazhagan, S., Shebiah, R.N., Nidhyanandhan, S.S., Ganesan, L.: Fruit recognition using color and texture features. J. Emerg. Trends Comput. Inf. Sci. **1**(2), 90–94 (2010)
5. Galleguillos, C., Rabinovich, A., Belongie, S.: Object categorization using co-occurrence, location and appearance. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2008, pp. 1–8. IEEE (2008)
6. Murphy, K., Freeman, W.: Contextual models for object detection using boosted random fields. In: NIPS (2004)
7. Wolf, L., Bileschi, S.: A critical view of context. Int. J. Comput. Vis. **69**(2), 251–261 (2006)
8. Lowe, D.G.: Object recognition from local scale-invariant features. In: The proceedings of the Seventh IEEE International Conference on Computer vision, 1999, vol. 2, pp. 1150–1157. IEEE (1999)
9. Tuytelaars, T., Van Gool, L., Bay, H., Ess, A.: Speeded-up robust features (surf). Comput. Vis. Image Underst. **110**(3), 346–359 (2008)
10. Abdel-Hakim, A.E., Farag, A. et al.: Csift: a sift descriptor with color invariant characteristics. In: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 1978–1983. IEEE (2006)
11. Suga, A., Fukuda, K., Takiguchi, T., Ariki, Y.: Object recognition and segmentation using sift and graph cuts. In: 19th International Conference on Pattern Recognition, ICPR 2008, pp. 1–4. IEEE (2008)
12. Ruf, B., Kokiopoulou, E., Detyniecki, M.: Mobile museum guide based on fast SIFT recognition. In: Detyniecki, M., Leiner, U., Nürnberger, A. (eds.) AMR 2008. LNCS, vol. 5811, pp. 170–183. Springer, Heidelberg (2010)

13. Mehrotra, H., Majhi, B., Gupta, P.: Annular Iris recognition using SURF. In: Chaudhury, S., Mitra, S., Murthy, C.A., Sastry, P.S., Pal, S.K. (eds.) PReMI 2009. LNCS, vol. 5909, pp. 464–469. Springer, Heidelberg (2009)
14. Khoshelham, K.: Extending generalized hough transform to detect 3d objects in laserrange data. In: ISPRS Workshop on Laser Scanning and SilviLaser 2007, 12–14 September 2007, Espoo, Finland. International Society for Photogrammetry and Remote Sensing (2007)
15. Flitton, G.T., Breckon, T.P., Bouallagu, N.M.: Object recognition using 3d sift in complex ct volumes. In: BMVC, pp. 1–12 (2010)
16. Knopp, J., Prasad, M., Willems, G., Timofte, R., Van Gool, L.: Hough transform and 3D SURF for robust three dimensional classification. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part VI. LNCS, vol. 6316, pp. 589–602. Springer, Heidelberg (2010)
17. Zhong, Y.: Intrinsic shape signatures: a shape descriptor for 3d object recognition. In: 2009 IEEE 12th International Conference on Computer Vision Workshops (ICCV Workshops), pp. 689–696. IEEE (2009)
18. Drost, B., Ulrich, M., Navab, N., Ilic, S.: Model globally, match locally: efficient and robust 3d object recognition. In: 2010 IEEEConference on Computer Vision and Pattern Recognition (CVPR), pp. 998–1005. IEEE (2010)
19. Papazov, C., Burschka, D.: An efficient RANSAC for 3D object recognition in noisy and occluded scenes. In: Kimmel, R., Klette, R., Sugimoto, A. (eds.) ACCV 2010, Part I. LNCS, vol. 6492, pp. 135–148. Springer, Heidelberg (2011)
20. Tombari, F., Salti, S., Di Stefano, L.: Unique signatures of histograms for local surface description. In: Maragos, P., Paragios, N., Daniilidis, K. (eds.) ECCV 2010, Part III. LNCS, vol. 6313, pp. 356–369. Springer, Heidelberg (2010)
21. Tombari, F., Di Stefano, L.: Hough voting for 3d object recognition under occlusion and clutter. IPSJ Trans. Comput. Vis. Appl. **4**, 20–29 (2012)
22. Rodolà, E., Albarelli, A., Bergamasco, F., Torsello, A.: A scale independent selection process for 3d object recognition in cluttered scenes. Int. J. Comput. Vis. **102**(1–3), 129–145 (2013)
23. Shafer, G., et al.: A Mathematical Theory of Evidence, vol. 1. Princeton University Press, Princeton (1976)
24. Rusu, R.B., Cousins, S.: 3d is here: point cloud library (pcl). In: 2011 IEEE International Conference on Robotics and Automation (ICRA), pp. 1–4. IEEE (2011)
25. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. **60**(2), 91–110 (2004)
26. Smets, P.: Constructing the pignistic probability function in a context of uncertainty. In: UAI, vol. 89, pp. 29–40 (1989)