18th International Conference on Information Fusion
Washington, DC - July 6-9, 2015

Generic Object Recognition Based on the Fusion of 2D and 3D SIFT Descriptors

Miaomiao Liu, Xinde Li
School of Automation, Southeast University
Nanjing, Jiangsu, China 210096.
Email:xindeli@seu.edu.cn

Jean Dezert
The French Aerospace Lab
F-91761 Palaiseau, France.
Email: jean.dezert@onera.fr

Chaomin Luo
ECE Dept., Univ. of Detroit Mercy
Detroit, MI, USA.
Email: luoch@udmercy.edu

***Abstract*—This paper proposes a new generic object recognition (GOR) method based on the multiple feature fusion of 2D and 3D SIFT (scale invariant feature transform) descriptors drawn from 2D images and 3D point clouds. We also use trained Support Vector Machine (SVM) classifiers to recognize the objects from the result of the multiple feature fusion. We analyze and evaluate different strategies for making this multiple feature fusion applied to real open-datasets. Our results show that this new GOR method has higher recognition rates than classical methods, even if one has large intra-class variations, or high inter-class similarities of the objects to recognize, which demonstrates the potential interest of this new approach.**

Keywords: Generic object recognition; Point cloud; 2D SIFT; 3D SIFT; Feature fusion; BoW; SVM, belief functions, PCR.

I. Introduction

Generic object recognition (GOR) in real environment plays a significant role in computer vision and artificial intelligence. It has important applications in intelligent monitoring, robotics, medical image processing, etc [1]–[3]. Contrariwise to specific object recognition[1], GOR is much more difficult to accomplish. Mainly because the generic features of objects which express the common properties in the same class and help to make the difference between classes need to be found out, instead of defining characteristics of particular category as used in specific object recognition (SOR) methods. The current main techniques for GOR are based on local feature extraction algorithms on 2D images, typically the 2D SIFT (scale invariant feature transform) descriptors [4], [5]. However, 2D images lose the 3D information of the objects, and are susceptible to change due to various external illumination conditions. To solve this drawback, 3D SIFT descriptors based on volumes [3], [6]–[10], and 3D descriptors based on point cloud model [11]–[13] have been proposed recently by several researchers because point cloud model of object is obtained from the depth images which only depends on the geometry of the objects. Such point cloud model has nothing to do with the brightness and reflection features of the objects. That is the main reason why we are also interested by these technique in this paper. 3D SIFT descriptors have been applied successfully in motion recognition of consecutive video frames by Scovanner et al. [6]. They show good performance in medical image processing [3], [7]–[9] as well. Object recognition has also be done with 3D SIFT in complex Computed Tomography (CT) for airport baggage inspection and security by Flitton et al. [10].

The object recognition algorithms based on single feature only often generate erroneous object recognitions, specially if there are big intra-class variations and some inter-class high similarities, or if there exist important changes in pose and appearance of objects. In these conditions, the use of a single feature is insufficient to make a reliable recognition and classification. To overcome this serious drawback, new recognition algorithms based on multiple features and fusion algorithms have been proposed recently in the literature [14]–[17]. Compared with the recognition algorithm using single feature only, the feature fusion algorithms combine multiple features information which can improve substantially the recognition rate.

In this paper, we propose a new method for GOR based on feature fusion of 2D and 3D SIFT descriptors, which consists of two main phases: 1) a training phase, and 2) a testing phase. In the both phases, we consider two types of inputs:

1) The first type of input is a database with 3D point cloud model representation of different objects from different categories (classes). In this work, our database has been just obtained from the web[2]. It is characterized by 3D SIFT descriptors adapted (in this paper) for point cloud – see the next section for details.
2) As second input, we use the same database with 2D images including some objects that are characterized by their 2D SIFT descriptors.

From these two inputs, the 2D and 3D SIFT feature descriptors are transformed into the corresponding Bag of Words (BoW) feature vector [18]. In the training phases, these two BoW feature vectors (drawn from the 2D and 3D SIFT) describing the object are used to train Support Vector Machines (SVMs) [19] to get the prediction functions. After this training phase, the system is used to recognize unknown objects in the testing phase. These two BoW feature vectors describing the object are used to make the object recognition in the testing phase. In this paper, we test:

1) the feature-level fusion strategy, where we combine (fuse) directly the two BoW-based feature vectors and

[1] such as face recognition [1] (SOR) where only certain objects or certain categories need to be recognized, which can be accomplished by training mass samples.

[2] http://rgbd-dataset.cs.washington.edu/dataset.html

978-0-9964527-1-7©2015 ISIF

we feed the trained SVM with the fused vector to get the final recognition result.

2) the decision-level fusion strategy, where each of the two BoW-based feature vectors feeds its corresponding trained SVM to get the corresponding recognition result separately. Then we test different fusion rules to combine these two recognition results to get the final recognition result.

The paper is organized as follows. The recognition algorithm is described in details in section 2. Section 3 evaluates the performances of this new method on real datasets. Conclusions with perspectives are given in section 4.

II. New generic object recognition method

This new method of object recognition consists in three main steps (features extraction and representation, features fusion, and classifier design) that we present in details in this section. To achieve the good recognition of objects, we propose to combine 2D scale-invariant feature transform (2D SIFT) characterizing the object features, with 3D SIFT (based on point clouds model). We need at first to recall the principle of 2D SIFT [4], [5], and we explain improved 3D SIFT descriptors applied in point cloud.

Step 1: Features extraction and representation

Feature extraction and representation are necessary for any object recognition algorithm. In many situations the object recognition task is very difficult because it is possible that some (partial) similarities exist in different classes of objects, as well as (partial) dissimilarities in the same class of objects. So the feature extraction process must be done as efficient as possible in order to help the recognition of objects by making the difference between object classes biggest, and by making the difference in the same class smallest. The objects need also to be represented at a certain level of semantic, using limited training objects to represent the class [2].

– 2D SIFT descriptor

In 1999, David Lowe [4] did present for the first time a new method to extract keypoints of objects in images, and to describe their local features that allows to make generic object recognition, for example in computer vision applications. His method has then been improved in [5], and extended to 3D by other authors (see next paragraph). The feature description of the object drawn from a training image is then used to identify the presence (if any) of the object in real (usually cluttered) observed scene. To get good object recognition performances, Lowe proposed a (2D) SIFT (scale-invariant feature transform) that warranties that the features extracted (i.e. the key-points) from the training image are detectable under changes in image orientation, scale, noise and illumination, and even if partial object occlusions occur in the observed scene. Lowe's SIFT feature descriptor is invariant to uniform scaling, orientation, and partially invariant to illumination changes and robust to local geometric (affine) distortion. The stable key-points locations of SIFT are given by the detection of scale-space extrema in the Difference-of-Gaussian (DoG) function $D(x,y,\sigma)$ convolved with the image $I(x,y)$. More precisely, one defines [5]

$$D(x,y,\sigma) \triangleq L(x,y,k\sigma) - L(x,y,\sigma) \tag{1}$$

where $L(x,y,k\sigma) \triangleq G(x,y,k\sigma) * I(x,y)$ and $L(x,y,\sigma) \triangleq G(x,y,\sigma) * I(x,y)$ are Gaussian-blurred images at nearby scale-space σ separated by a constant multiplicative factor[3] k, and where $*$ is the convolution operator and $G(x,y,\sigma)$ is the centered Gaussian kernel defined by

$$G(x,y,\sigma) \triangleq \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2} \tag{2}$$

The local extreme points of $D(x,y,\sigma)$ functions (DoG images) define the set of keypoint candidates (the SIFT descriptor). To detect the keypoints, each sample point (pixel) is compared to its eight neighbors in the current image and its nine neighbors in the scale below and above. The sample point under test is considered as a keypoint (local extrema) if its value is larger (or smaller) than all of its 26 neighbors. The localization of a candidate keypoint is done by the 2nd-order Taylor expansion of the DoG scale-space function $D(x,y,\sigma)$ with the candidate keypoint taken as the origin [5]. However in general there are too many candidate keypoints and we need to identify and remove the bad candidates that have too low contrast[4], or are poorly localized along an edge. For doing this, a contrast thresholding is applied on $D(x,y,\sigma)$ to eliminate all the candidate keypoints below a chosen[5] threshold value τ. To eliminate the candidate keypoints that are poorly localized along an edge, Lowe [5] uses a thresholding method based on the ratio of the eigenvalues of the Hessian matrix $\mathbf{H}$ of the DoG function, because for poorly defined extrema in the DoG function the principal curvature across the edge would be much larger than the principal curvature along it. More precisely, if the ratio $Tr(\mathbf{H})^2/Det(\mathbf{H}) > (r_{th}+1)^2/r_{th}$ then the candidate keypoint is rejected. Here, r_{th} is a chosen threshold value of the ratio between the largest magnitude eigenvalue of $\mathbf{H}$ and the smaller one[6].

Once all the keypoints are determined, one must assign a consistent orientation based on local image properties, from which the keypoint descriptor can be represented, hence achieving invariance to image rotation. For this, the scale of the keypoint is used to choose the Gaussian-blurred image L with the closest scale. The keypoint descriptor is created by computing at first the gradient magnitude $m(x,y)$ and its orientation $\theta(x,y)$ at each pixel (x,y) in the region around the keypoint in this Gaussian-blurred image L as follows [5]

$$\begin{cases} m(x,y) &= \sqrt{L_x^2 + L_y^2} \\ \theta(x,y) &= \tan^{-1}(\frac{L_y}{L_x}) \end{cases} \tag{3}$$

[3] The choice for $k = 2^{1/s}$ is justified by Lowe in [4], where s is an integer number of intervals

[4] because they are sensitive to noise.

[5] We have chosen $\tau = 0.02$ in our simulations.

[6] In [5], Lowe takes $r_{th} = 10$.

with $L_x \triangleq L(x+1,y) - L(x-1,y)$ and $L_y \triangleq L(x,y+1) - L(x,y-1)$. In [5], a set of orientation histograms is created on 4x4 pixel neighborhoods with 8 directions (bins) each. These histograms are computed from magnitude and orientation values of samples in a 16×16 region around the keypoint such that each histogram contains samples from a 4×4 subregion of the original neighborhood region. The magnitudes are weighted by a Gaussian function with σ equal to one half the width of the descriptor window. The descriptor then becomes a 128-dimensional feature vector because there are $4 \times 4 = 16$ histograms each with 8 directions. This vector is then normalized to unit length in order to enhance invariance to affine changes in illumination. Also a threshold of 0.2 is applied to reduce the effects of non-linear illumination, and the vector is again normalized. The figure 1 shows an example of 4×4 keypoint descriptor, where the space delimited by the purple ellipse is the neighborhood under consideration.

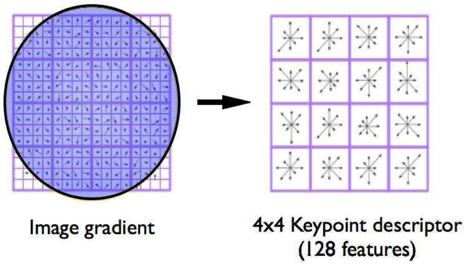

Fig. 1: A 4×4 Keypoint descriptor (Credit: J. Hurrelmann).

The simplest method to find the best candidate match for each keypoint would consist in identifying its nearest[7] neighbor in the database of key points from training images. Unfortunately, SIFT-based keypoint matching requires more sophisticate methods because many features from an image will not have any correct match in the training database because of background clutter in observed scene and because of possible missing features in training images, see [5] for details. SIFT method is patented by the University of Bristish Columbia (US Patent 6,711,293 – March 23, 2004) and a demo is available in [20]. Open SIFT codes can be found on the web, for example in [21].

– 3D SIFT descriptor

The previous 2D SIFT descriptor working with pixels has been extended to 3D using volumes in different manners by different authors [3], [6]–[10]. In this paper, we adapt the 3D SIFT for point cloud inspired by [6], [13]. But all the methods require same functional steps as for 2D SIFT, that is 1) Keypoints detection; 2) Key points orientation; and 3) Descriptor representation. We present these steps in detail in the next subsections.

1) Keypoint detection

The scale space of a 3D input point cloud is defined as a 4D function $L(x,y,z,\sigma) = G(x,y,z,k\sigma) * P(x,y,z)$ obtained by the convolution of a 3D variable-scale centered Gaussian kernel $G(x,y,z,\sigma)$, with the input point $P(x,y,z)$, where

$$G(x,y,z,\sigma) = \frac{1}{\left(\sqrt{2\pi}\sigma\right)^3} e^{-(x^2+y^2+z^2)/2\sigma^2} \qquad (4)$$

Extending Lowe's approach [5], scale-space σ is separated by a constant multiplicative factor k, and the candidate keypoints in 4D scale space are taken as the local extrema (maxima or minima) of the multi-scale DoG defined for $i \in [0, s+2]$ by

$$D(x,y,z,k^i\sigma) = L(x,y,z,k^{i+1}\sigma) - L(x,y,z,k^i\sigma) \qquad (5)$$

To find extrema of the multi-scale DoG function, each sample point is compared to its $27+26+27 = 80$ neighbors, where 26 neighbors belong to the current scale, and each 27 neighbors in the scale above and below. A keypoint is chosen only if it is larger than all of its neighbors or smaller than all of them. To eliminate the bad candidate keypoints having low contrast, one uses a thresholding method to remove the erroneous points. A contrast threshold is applied on $D(x,y,z,k^i\sigma)$ to eliminate all the candidate keypoints below a chosen[8] threshold value τ.

2) Keypoint orientations

Similarly to 2D SIFT, once all the keypoints are determined in 3D, one must assign a consistent orientation based on local points properties, from which the keypoint descriptor can be represented, hence achieving invariance to object rotation. For this, The two-dimensional histogram is calculated by gathering statistics of the angles between the neighboring points and their center. The keypoint descriptor is created by computing at first the vector magnitude $m(x,y,z)$ and its orientations $\theta(x,y,z)$ (azimuth angle) and $\phi(x,y,z)$ (elevation angle) between each point (x,y,z) in the region around the keypoint and their center (x_c,y_c,z_c) as follows[9]

$$\begin{cases} m(x,y,z) &= \sqrt{(x-x_c)^2+(y-y_c)^2+(z-z_c)^2} \\ \theta(x,y,z) &= \tan^{-1}\left((y-y_c)/(x-x_c)\right) \\ \phi(x,y,z) &= \sin^{-1}\left((z-z_c)/m(x,y,z)\right) \end{cases} \qquad (6)$$

In 3D point cloud, each point has two values which represent the direction of the region, whereas in 2D case each pixel had only one direction of the gradient.

Extending Lowe's approach in 3D case, in order to find the keypoint orientations we construct a weighted histogram for the 3D neighborhood around each candidate keypoint. There are different ways for doing this. In this work, a 2D-histogram

[7] based on Euclidean distance metric.

[8] We have chosen $\tau = 0.5$ in our simulations.

[9] In Eq.(6), θ and ϕ refer to the original coordinate system. In the paragraph "Descriptor representation" on p. 4, they refer to the rotated coordinate system. (x_c,y_c,z_c) is not same as (x_p,y_p,z_p). The former refers to the center of the keypoints r-points neighborhood. The latter refers to the keypoint.

is produced by grouping the angles in bins which divide θ and ϕ into 10 deg angular bins. A regional Gaussian weighting of $\mathrm{e}^{-(2d/R_{\max})^2}$ for the points whose magnitude is d is applied to the histogram, where $R_{\max}$ represents the max distance from the center. The sample points at a distance greater than $R_{\max}$ are ignored. The histogram is smoothed using a Gaussian filter to limit the effect of noise. The dominant azimuth α and elevation β of the keypoint are determined by the peaks of the 2D-histogram. In order to enhance robustness, peaks in the histogram within 80% of the largest peak are also retained as possible secondary orientations.

3) Descriptor representation

Each keypoint p is described by its location $\mathbf{p} \triangleq [x_p, y_p, z_p]^t$, scale σ_p, and orientation angles α_p and β_p. The descriptor representation associated with a keypoint p is based on the local spatial characteristics around it to describe its features. To ensure rotation invariance of the descriptor, the r-points p_i $(i = 1, \dots, r)$ of coordinates $\mathbf{p}_i \triangleq [x_i, y_i, z_i]^t$ around the keypoint of interest p are at first transformed (rotated) in the dominant orientation of p by the following transformation

$$\mathbf{p}'_i = \begin{bmatrix} \cos\alpha_p\cos\beta_p & -\sin\alpha_p & -\cos\alpha_p\sin\beta_p \\ \sin\alpha_p\cos\beta_p & \cos\alpha_p & -\sin\alpha_p\sin\beta_p \\ \sin\beta_p & 0 & \cos\beta_p \end{bmatrix} \cdot \mathbf{p}_i \quad (7)$$

Then the vector $\mathbf{n}$ at the key point which is normal to the surface of the r-points neighborhood is calculated according to the routine available in the open Point Cloud Library (PCL) [22]. For each (rotated) point $\mathbf{p}'_i$ $(i = 1, \dots, r)$ in the r-points neighborhood of the (rotated) keypoint p', we calculate the vector $\mathbf{p}'\mathbf{p}'_i$ and the magnitude m and angles θ and ϕ according to Eq. (6). The angle δ between $\mathbf{n}$ and $\mathbf{p}'\mathbf{p}'_i$ is given by

$$\delta = \cos^{-1}\Big(\frac{\mathbf{p}'\mathbf{p}'_i \cdot \mathbf{n}}{|\mathbf{p}'\mathbf{p}'_i| \cdot |\mathbf{n}|}\Big) \quad (8)$$

Therefore, a keypoint p' with its neighbor p'_i is represented by the 4-tuple $(m, \theta, \phi, \delta)$. To reduce the computational time, instead of dividing the neighborhood into $n \times n \times n$ subregions (with $n = 4$ as in Lowe's 2D SIFT descriptor), we take directly the entire neighborhood, which means that we have $n = 1$. The histogram used to generate the 3D descriptor at the keypoint p' is derived by splitting (θ, ϕ, δ) space into 45 deg bins, and adding up the number of points with the Gaussian weighting of $\mathrm{e}^{-(2m/R_{\max})^2}$. So the dimension of our 3D SIFT descriptor is $n \times n \times n \times 4 \times 4 \times 8 = 128$ (as for the 2D SIFT descriptor described previously), because $n = 1$; the azimuth angle $\theta \in [0, 360]$ deg which is split into 8 bins of 45 deg; the elevation angle $\phi \in [-90, 90]$ deg which is split into 4 bins of 45 deg; and $\delta \in [0, 180]$ deg which is also split into 4 bins of 45 deg. Each 3D SIFT descriptor is normalized to unity.

The 2D and 3D SIFT descriptors summarize efficiently the useful information contained in 2D and 3D images. Instead of working directly with whole images, it is usually more interesting (in terms of computational burden reduction) to work directly with 2D and 3D SIFT descriptors, specially if real-time object recognition is necessary. Generally, the objects characterized by 2D and 3D SIFT descriptors have different number of keypoints which makes the feature fusion (FF) problem for object recognition very challenging. For example, for a simple object like an apple, we can get 45 keypoints using 3D SIFT descriptor, and 38 keypoints using 2D SIFT descriptor. To overcome this problem, we adopt the Bag of Words (BoW) model [18] to gather the statistics of the 2D and 3D SIFT descriptors to describe the objects.

– BoW model for features vector

In the BoW feature model, the feature descriptors of all the interest points are quantized by clustering them into a pre-specified[10] number of clusters. Instead of using k-means algorithm as in [2], we use the k-means++ method [23] which selects more effectively the initial cluster centers to complete this step. The resultant cluster centers are now called *visual words*, while the collection of these cluster centers is referred to as the *visual word vocabulary*. Once our vocabulary is computed, the descriptors are matched to each *visual word* based on the Euclidean distance and the frequency of the visual words in image and in point cloud is accumulated into a histogram, which is the BoW feature vector of the image and of the point cloud. So each object in 2D image and in 3D point cloud is described by a 1×300 BoW-based feature vector denoted respectively $\mathbf{BoW}_{2D}$ and $\mathbf{BoW}_{3D}$. These two BoW-based feature vectors will be used for feeding the trained SVM classifiers to get the final object recognition.

Step 2: Classifier design

Once the object description is completed, SVMs are trained to learn objects categories and to perform the object classification. SVM is a supervised and discriminative machine learning method providing usually good performance. Through offline training of pre-limited samples, we seek a compromise between model complexity and learning ability, to get a good discriminant function [19]. Linear SVM classifier is applied for its efficiency and it is a typical classifier for two categories problems. In many real-life applications, we are face to multi-category classification problems and we use trained 1V1 SVMs between classes to set up a multi-category classifier. The training process is done as follows: for training samples belonging to the ith category, we make a pairwise SVM training with respect to all the other classes. So, we get $C_n^2 = n(n-1)/2$ 1V1 SVM classifiers for training samples of n categories.

Step 3: Features fusion strategies

When the two BoW-based features vectors of the object to recognize have been computed from 2D and 3D SIFT descrip-

[10] In our simulations, we took $K = 300$.

tors, we have to use them to achieve the object recognition thanks to the trained SVMs from the BoW-based features vectors of known objects of our data base. In this paper, we present briefly the following different strategies that we have tested:

1) *The direct feature-level fusion strategy*: this feature-level fusion is for feeding SVM classifiers in training phase and then making object recognition. With this strategy we combine (fuse) directly the two BoW-based feature vectors $\mathbf{BoW}_{2D}$ and $\mathbf{BoW}_{3D}$, and we feed the trained (global) SVM classifiers with the fused vector to get the final recognition. The principle of our method based on this strategy is summarized in Fig 2.

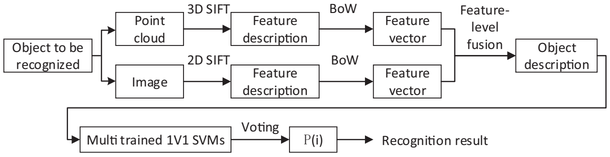

Fig. 2: Direct feature-level fusion strategy.

2) *The decision-level fusion strategy*: each BoW-based feature vector $\mathbf{BoW}_{2D}$ and $\mathbf{BoW}_{3D}$ feeds a specific trained SVM to get separately the corresponding recognition result. Then we test different fusion rules to combine these two recognition results to get the final fusioned recognition result. In this work we have evaluated the performances of the following rules:
   - Average weighted fusion rule,
   - PCR6 fusion rule of DSmT [24],
   - Murphy's rule of combination [26].

   The principle of our method based on this strategy is summarized in Fig 3.

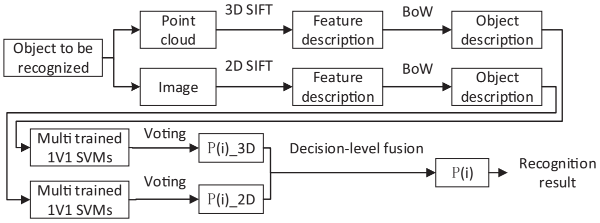

Fig. 3: Decision-level fusion strategy.

1) **The direct feature-level fusion strategy**

This strategy consists of the following steps:

1-a) For any object to classify, we extract its 2D and 3D SIFT descriptors associated with each keypoint. So we get N_{2D} 2D SIFT descriptors of size 1×128 if one has extracted N_{2D} keypoints from the 2D image under test, and we get N_{3D} 3D SIFT descriptors of size 1×128 if one has extracted N_{3D} keypoints from the 3D point cloud under test.

1-b) From the N_{2D} 2D SIFT descriptors of size 1×128, we compute 1×300 BoW feature vectors $\mathbf{BoW}_{2D}$, and from the N_{3D} 3D SIFT descriptors of size 1×128, we compute 1×300 BoW feature vectors $\mathbf{BoW}_{3D}$ thanks to the BoW model representation [18].

1-c) The direct feature-level fusion is done by stacking the BoW-based feature vectors $\mathbf{BoW}_{2D}$ and $\mathbf{BoW}_{3D}$ to get a 1×600 vector $\mathbf{BoW}_{2D,3D} \triangleq [\mathbf{BoW}_{2D}, \mathbf{BoW}_{3D}]$.

1-d) The feature-level fused vector $\mathbf{BoW}_{2D,3D}$ is fed in all 1v1 trained SVMs to get the corresponding discriminant results. The probability $P(i)$ of the object to belong to the category c_i $(i = 1, 2, \dots, n)$ is estimated by voting.

1-e) The object is associated to the category (or class) having the largest probability, that is:

$$\text{Class(Object)} = \arg\max_{1\leq i\leq n}\{P(i)\} \tag{9}$$

2) **The decision-level fusion strategy**

As stated before, with this strategy each BoW-based feature vector $\mathbf{BoW}_{2D}$ and $\mathbf{BoW}_{3D}$ feeds a specific trained SVM to get separately the corresponding recognition result. Then different fusion rules can be used to combine these two recognition results to get the final fusioned recognition result.

2-a) The average weighted fusion rule: This very simple rule consists of a voting procedure. The $\mathbf{BoW}_{2D}$ and $\mathbf{BoW}_{3D}$ vectors feed separately all corresponding 1v1 trained SVMs to get the discriminant results, and we compute the corresponding number of votes $vote[i]$ for each class c_i, $i = 1, 2, \dots, n$. We will denote $vote_{2D}[i]$ the distribution of votes drawn from 2D SIFT, and $vote_{3D}[i]$ the distribution of votes drawn from 3D SIFT. The probability $P_{2D}(i)$ of the object to belong to the class c_i based on 2D SIFT descriptors is estimated by $P_{2D}(i) = vote_{2D}[i]/\sum_{1=1}^{n} vote_{2D}[i]$, similarly we have $P_{3D}(i) = vote_{3D}[i]/\sum_{1=1}^{n} vote_{3D}[i]$. Then the voting results drawn from SVMs feeded with 2D and 3D SIFT are averaged to obtain the fusion result.

2-b) PCR6 combination rule: The BBA (Basic Belief Assignment) $m_1(.)$ and $m_2(.)$ are built from the empirical probability obtained by voting procedure described in 2-a). The elements of the frame of discernment Θ are the n different classes c_1, c_2, …, c_n. To get the final result, the BBA's $m_1(.)$ and $m_2(.)$ are fused using the PCR6 combination rule[11] [24], defined by $m_{PCR6}(\emptyset) = 0$ and for all $X \neq \emptyset$ in 2^Θ,

$$m_{PCR6}(X) \triangleq \sum_{\substack{X_1,X_2\in 2^\Theta\\ X_1\cap X_2=X}} m_1(X_1)m_2(X_2)+ \\ \sum_{\substack{Y\in 2^\Theta\setminus\{X\}\\ X\cap Y=\emptyset}} [\frac{m_1(X)^2 m_2(Y)}{m_1(X)+m_2(Y)} + \frac{m_2(X)^2 m_1(Y)}{m_2(X)+m_1(Y)}] \tag{10}$$

[11] PCR6 formula coincides with the formula of PCR5 fusion rule here because one considers only two BBA's to combine. If more than two BBA's have to be fused altogether, we advise to use PCR6 rather than PCR5 - see [25] for a theoretical justification.

where all denominators in Eq.(10) are different from zero. If a denominator is zero, that fraction is discarded. All propositions/sets are in a canonical form.

2-c) Murphy's rule: Taking the feature-level fusion of 2D and 3D SIFT as a separate feature, together with the 2D and 3D SIFT, there are three features. Then the BBA $m_1(.)$, $m_2(.)$ and $m_3(.)$ are built from the empirical probability obtained by the voting procedure. The vote results of the features are combined based on the Murphy rule[12] [26].

III. Simulation results

A. The experimental setup

We evaluate the recognition algorithm on a large-scale multi-view object dataset collected using an RGB-D camera [27]. This dataset contains color, depth images and point clouds of 300 physically distinct everyday objects taken from different viewpoints. The objects belong to one of 51 categories and contain three viewpoints. To test the recognition ability of our features, we test category recognition on objects that were not present in the training set. At each trial, we randomly choose one test object from each category and train classifiers on the remaining objects. We randomly choose 100 training samples and 60 test samples for each category. The object recognition rate (ORR) is calculated by

$$ORR = n_r/N \tag{11}$$

where n_r is the number of objects correctly recognized, and N is the total number of test samples.

B. Experiment results and analysis

B.1 Accuracy of our 3D SIFT descriptor

In this simulation, we choose six categories with significant intra-class variations and high inter-class similarities. The objects to recognize are $apple$, $tomato$, $banana$, $pitcher$, $cereal\_box$, and $kleenex$. The Point Feature Histogram (PFH) [11] and PFHRGB methods in open PCL [22] outperform the existed 3D features based on point clouds [28]. In order to verify the advantages of the proposed 3D SIFT for GOR, we compare these tree feature descriptors under the same conditions. Keypoints are detected using SIFTKeypoint module in open PCL [22] for each feature descriptors. Then the vectors of different feature descriptors of the keypoints are calculated. The object recognition rates (ORR) that we get are shown in Table I.

Type of feature descriptor	ORR (in %)
PFH based on [11]	81.39
PFHRGB based on [22]	84.17
3D SIFT based on this paper	91.11

TABLE I: Object recognition rates (ORR) of three descriptors.

The PFHRGB descriptor is an improved PFH feature descriptor enriched with color information which allows to improves object recognition rate. As shown in Table 1, compared with PFH and PFHRGB, the object recognition rate we get with our 3D SIFT descriptor adapted for point cloud gains 6.94% w.r.t. PFHRGB and 9.72% w.r.t PFH.

B.2 Performances of feature fusion strategies

Here, we evaluate the performance (i.e. the ORR) of the different features fusion strategies presented in Section II (Step 3). We have chosen 10 categories ($apple$, $tomato$, $banana$, $pitcher$, $cereal\_box$, $kleenex$, $camera$, $coffee\_mug$, $calculator$, $cell\_phone$) having significant intra-class variations and high inter-class similarities. We compare our four fusion approaches: the direct feature-level fusion and the three decision-level fusions (by average weighted fusion, PCR6, and Murphy's rule). The results are shown in Fig. 4.

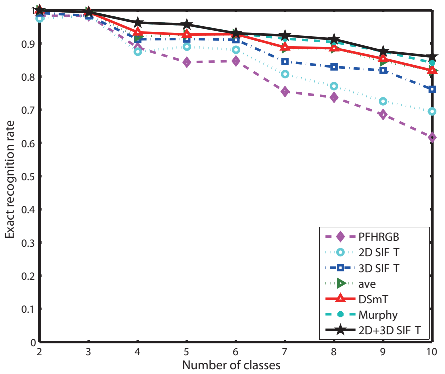

Fig. 4: Performances of the four feature fusion strategies.

where the legend of curves of Fig.4 must be read as follows: DSmT means PCR6 rule in fact, 2D+3D SIFT means the direct feature-level fusion of 2D and 3D SIFT, and *ave* means the average weighted feature fusion rule. The horizontal axis represents the total number of categories that we have tested. Due to the variability of the objects, the information provided by a single feature is too imprecise, uncertain and incomplete for getting good ORR. As shown in Fig.4, ORR obtained with the different feature fusion strategies are better than the ORR obtained with the best single descriptor. The results of average weighted fusion and PCR6 are close, but are lower than the other two fusion methods. Feature-level fusion of 2D and 3D SIFT is taken as the third feature for Murphy's rule. However, compared with the feature-level fusion, the performances of Murphy's rule do not improve. So, the direct feature-level fusion performs best among these fusion strategies, and the following experiments are completed based on the direct feature-level fusion. One clearly sees that 3D SIFT proposed in this work significantly outperforms 2D SIFT and PFHRGB descriptors for GOR. As shown in Fig.4, ORR decreases with the increasing of the number of categories because of the design of the multi-category classifier which consists of many 1V1 SVM classifiers. Each classification error will be accumulated to the final voting

[12] Because results of the fusion with Dempster's rule are very close to results with Murphy's rule in our applications, we do not report them in our analysis.

results, leading to an increasing of recognition errors.

B.3 Robustness to intra-class variation and inter-class similarities

In this study, we compare the ORR performances in different classes having high similarity (e.g., apple and tomato), and in the same class but having strong variation (e.g., pitcher object) as in Figs. 5 and 6 below. We evaluate the accuracy of PFHRGB, 2D SIFT, 3D SIFT and the feature-level fusion of 2D and 3D SIFT under the same conditions. Training and testing samples are the same as in the first experiment. Our simulation results are shown in Table II.

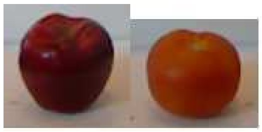

Fig. 5: Apple and Tomato.

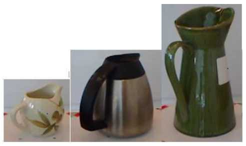

Fig. 6: Pitchers.

Feature descriptor	PFHRGB	2D SIFT	3D SIFT	2D+3D SIFT
ORR(apple)	61.67	53.33	71.67	65.00
ORR(tomato)	100	98.33	91.67	100
ORR(banana)	91.67	93.33	93.33	100
ORR(pitcher)	70.00	95.00	96.67	98.33
ORR(cereal_box)	91.67	98.33	95.00	95.00
ORR(kleenex)	90.00	90.00	100	100
Averaged ORR	84.17	88.06	91.11	93.06

TABLE II: ORR (in %) of different classes.

As we see from Table II, using 3D SIFT increases the ORR of 3.05% w.r.t. 2D SIFT. This shows that the introduction of the depth information improve the quality of object recognition. Three different objects of the pitcher class are shown in Figure 6. As we see, there are great differences within such class. 3D SIFT achieves ORR with 96.67% accuracy, much superior to the 70% obtained with PFHRGB. Apple and tomato displayed in Figure 5 look highly similar even if they belong to two distinct classes. 3D SIFT provides much better ORR than the other descriptors. As shown in Table II, our GOR method based on feature-level fusion of 2D and 3D SIFT offer better robustness to intra-class variations and inter-class similarities, and 3D SIFT gives higher accuracy than the other single descriptors.

B.4 Robustness to changes of the angle of view

In this experiment, we evaluate the performance of our GOR method when applied under different observation conditions, more precisely when the objects are observed under three very distinct angles of view (30 deg, 45 deg and 60 deg).Training samples are the same as the Experiment 1. Randomly select 60 objects from each view to be as the test samples. So for each view, there are 360 test samples from 6 categories. The experimental results are shown in Fig. 7.

From Fig. 7, one sees that ORR with 3D SIFT is relatively accurate and stable compared with PFHRGB descriptor. The direct feature-level fusion strategy (with $ORR > 90\%$) offers much better ORR than using the best single descriptor, which indicates that the combination of 2D and 3D SIFT is effective and robust for category recognition even under very distinct angles of view.

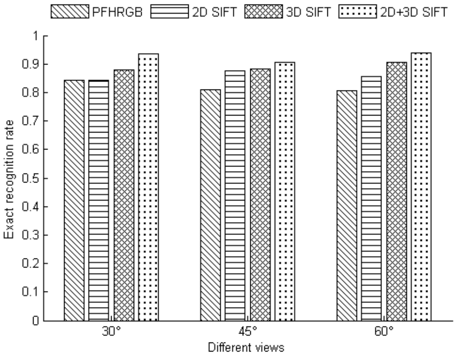

Fig. 7: ORR Performances under 3 angles of view.

B.5 Robustness to size scaling

The training samples are the same as in the first experiment. To evaluate the robustness of our method to size scaling (zooming), the test samples are zoomed out to $1/2$, $1/3$ and $1/4$. As shown in Table III.

Feature descriptor	PFHRGB	2D SIFT	3D SIFT	2D+3D SIFT
ORR (no Zoom)	84.17	88.06	91.11	93.06
ORR (Zoom=1/2)	74.44	77.50	76.67	82.78
ORR (Zoom=1/3)	63.33	64.17	65.28	68.89
ORR (Zoom=1/4)	61.39	46.94	61.67	63.05

TABLE III: Averaged ORR (in %) for different zoomings.

As one sees in Table III, our GOR method with fusion is superior to the algorithm based on single descriptor. However, the ORR of each feature descriptor has decreased. Especially when zoomed to 1/4, the accuracy of ORR with 2D SIFT is only 46.94%. The main reason is that part of the images, such as apple (whose original size is only 84×82) after scaling, reduces the number of useful keypoints. The feature-level fusion algorithm still provides an averaged ORR of 63.05%.

B.6 Computational time evaluation

The computational times (CT) of the different feature descriptors have been evaluated with an i7-3770@3.4GHz CPU, under x64 Win7 operating system and are shown in Table IV. The training and test samples are the same as in the first experiment. Because the Point cloud model contains a larger amount of data and richer information than image, therefore CT using point cloud is relatively long, which is normal. The largest proportion of CT in the whole recognition process is the feature extraction and description. 3D SIFT includes keypoints detection and description. If the points' number of the object is n, the time complexity of keypoints detection is $O(octaves \cdot scale \cdot k \cdot n)$. Because the pyramid layers $octaves$, scale of each layer $scale$ and neighborhood of key points k are constant, the time complexity is $O(n)$. For the detected m keypoints, the time complexity of calculating the descriptors of the key points is $O(mn)$. So the time complexity of 3D SIFT is $O(mn + n)$, ignoring lower-order item, the time complexity is $O(mn)$. As seen in Table IV, the CT of 3D SIFT has diminished of 34.75% w.r.t. PFHRGB, and the CT performance with fusion of 2D and 3D SIFT turns out to be faster (22.07%) than PFHRGB, and the ORR performance is substantially improved.

Feature descriptors	CT of	CT of
	360 test samples (in s)	each test sample (in s)
PFHRGB	3404.628	9.4573
3D SIFT	2221.608	6.1711
2D+3D SIFT	2653.272	7.3702

TABLE IV: Computational times for feature descriptors.

IV. Conclusions

Because there are many complex objects in the real scenes we observe in the nature and because of possible large intra-class variations and high inter-class similarities, the generic object recognition (GOR) task is very hard to achieve in general. In this paper we have proposed a new GOR method based on 2D and 3D SIFT descriptors that allows to calculate multiple feature vectors which are combined with different strategies, and feed SVM classifier for making object recognition. The evaluation of the performances based on real open-datasets has shown the superiority of our new 3D SIFT descriptor adapted for point cloud with respect to the existing 3D features such as PFHRGB. Our GOR method based on feature fusion of 2D and 3D SIFT works better than the one using best single feature. For now, if the environment substantially changes, we have to retrain the system. To overcome this problem we will also consider background segmentation within GOR in future works. Also, we would like to reduce the computational time needed for feature extraction and description in maintaining good recognition rate, and we want to explore more feature fusion strategies to improve (if possible) the recognition performances.

Acknowledgment

This work was supported by NNSF of China (No. 61175091), Qing Lan Project of Jiangsu Province, Aeronautical Science Foundation of China (20140169002), and Six Major Top-talent Plan of Jiangsu Province.

References

[1] Y. Lei, M. Bennamoun, M. Hayat, Y. Guo, An efficient 3D face recognition approach using local geometrical signatures, Pattern Recognition, Vol. 47(2), pp. 509–524, 2014.

[2] X.-D. Li, X. Zhang, B. Zhu, X.-Z. Dai, A Visual Navigation Method for Robot Based on a GOR and GPU Algorithm, Robot, Vol. 34(4), pp. 466–475, 2012 (in Chinese).

[3] S. Allaire, J.J. Kim, S.L. Breen, D.A. Jaffray, V. Pekar, Full orientation invariance and improved feature selectivity of 3D SIFT with application to medical image analysis, Proc. IEEE CVPR Workshops, Anchorage, AK, USA, 23–28 June 2008.

[4] D.G. Lowe, Object recognition from local scale-invariant features, Proc. of IEEE CCV Conf., Vol. 2, pp. 1150–1157, Corfu, Greece, Sept. 1999.

[5] D.G. Lowe, Distinctive Image Features from Scale-Invariant Key points, Int. J. of Computer Vision, Vol. 60(2), pp. 91–110, 2004.

[6] P. Scovanner, S. Ali, M. Shah, A 3-dimensional SIFT descriptor and its application to action recognition, Proc. of 15th ACM MM Conf., pp. 357–360, Augsburg, Germany, Sept. 23-29, 2007.

[7] W. Cheung, G. Hamarneh G. N-SIFT: N-dimensional scale invariant feature transform for matching medical images, Proc. of 4th IEEE Int. Symp. on Biomedical Imaging, pp. 720–723, Arlington, VA, USA, 2007.

[8] R.N. Dalvi, I. Hacihaliloglu, R. Abugharbieh, 3D ultrasound volume stitching using phase symmetry and Harris corner detection for orthopaedic applications, (Medical Imaging 2010) Proc. of SPIE, Vol. 7623, San Diego, CA, USA, 2010.

[9] M. Niemeijer, et al., Registration of 3D spectral OCT volumes using 3D SIFT feature point matching, Proc. SPIE Vol. 7259 (Medical Imaging 2009), Lake Buena Vista, FL, USA, 27 March 2009.

[10] G.T. Flitton, T.P. Breckon, N. Megherbi, Object Recognition using 3D SIFT in Complex CT Volumes, Proc. of BMV Conf., pp. 1-12, Aberystwyth, UK, Aug 31–Sept. 3rd, 2010.

[11] R.B. Rusu, N. Blodow, Z.C. Marton, M. Beetz, Aligning point cloud views using persistent feature histograms, Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Syst., pp. 3384–3391, Nice, France, 2008.

[12] R.B. Rusu, N. Blodow, M. Beetz, Fast point feature histograms (FPFH) for 3D registration, Proc. of IEEE Int. Conf. on Robotics and Autom., pp. 3212–3217, Kobe, Japan, 2009.

[13] S. Lazebnik, C. Schmid, J. Ponce, A sparse texture representation using local affine regions, IEEE Trans. on PAMI, Vol. 27(8), pp. 1265–1278, 2005.

[14] X.-D. Li, J.-D. Pan, J. Dezert, Automatic Aircraft Recognition using DSmT and HMM, Proc. of Fusion 2014, Salamanca, Spain, July 2014.

[15] L. Bo, K. Lai, X. Ren, D. Fox, Object recognition with hierarchical kernel descriptors, Proc. of CVPR IEEE Conf., pp. 1729–1736, Colorado Springs, CO, USA, June 2011.

[16] L. Bo, X. Ren, D. Fox, Depth kernel descriptors for object recognition, Proc. of IEEE/RSJ IROS Conf., pp. 821–826, San Francisco, CA, USA, Sept. 2011.

[17] M. Mirdanies, A.S. Prihatmanto, E. Rijanto, Object Recognition System in Remote Controlled Weapon Station using SIFT and SURF Methods, J. of Mechatronics, Elect. Power, and Vehicular Techn., Vol. 4(2), pp. 99–108, 2013.

[18] J. Sivic, A. Zisserman, Video google: A text retrieval approach to objects matching in videos, Proc. of 9th CCV Conf, pp. 1470–1477, 2003.

[19] B.E. Boser, I.M. Guyon, V.N. Vapnik, A training algorithm for optimal margin classifiers, Proc. of the 5th ACM Workshop on Comput. learning theory, pp. 144–152, Pittsburgh, PA, USA, 1992.

[20] SIFT demo program (Version 4, July 2005).
http://www.cs.ubc.ca/~lowe/keypoints/

[21] R. Hess, An Open Source SIFT Library, ACM MM, 2010.
http://robwhess.github.io/opensift/

[22] R.B. Rusu, S. Cousins, 3D is here: Point cloud library (PCL), Proc. of IEEE Int. Conf. on Robotics and Autom., pp. 1–4, Shanghai, China, 2011.

[23] D. Arthur, S. Vassilvitskii, k-means++: The advantages of careful seeding, Proc. of SODA '07, pp. 1027–1035, 2007.

[24] F. Smarandache, J. Dezert (Editors), Advances and applications of DSmT for information fusion, ARP, Rehoboth, NM, U.S.A., Vol. 1–4, 2004–2015. http://fs.gallup.unm.edu//DSmT.htm

[25] F. Smarandache, J. Dezert, On the consistency of PCR6 with the averaging rule and its application to probability estimation, Proc. of Fusion 2013, Istanbul, Turkey, July 2013.

[26] C.K. Murphy, Combining Belief Functions when Evidence Conflicts, Decision Support System, Vol. 29(1), pp. 1–9, 2000.

[27] K. Lai, L.-F. Bo, X.-F Ren, D. Fox, A Large-Scale Hierarchical Multi-View RGB-D Object Dataset, Proc. of IEEE Int. Conf. on Robotics and Autom., pp: 1817–1824, Shanghai, China, 2011.

[28] L.A. Alexandre, 3D Descriptors for Object and Category Recognition: a Comparative Evaluation, Workshop on Color-Depth Camera Fusion in Robotics at the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS 2012, October 7-12, Vilamoura, Portugal, 2012.