3D Object Recognition Based on Image Features: A Survey

Khaled Alhamzi
Dept. of Information Systems, Faculty of Computers and Information, Mansoura University
Mansoura, Egypt
Kalhamzi {at} yahoo.com

Mohammed Elmogy
Dept. of Information Technology, Faculty of Computers and Information, Mansoura University
Mansoura, Egypt

Sherif Barakat
Dept. of Information Systems, Faculty of Computers and Information, Mansoura University
Mansoura, Egypt

Abstract—**Object recognition is a basic application domain in computer vision. For many decades, it is considered as an area of extensive research especially in 3D. 3D object recognition can be defined as the task of finding and identifying objects in the real world from an image or a video sequence. It is still a hot research topic in computer vision because it has many challenges such as viewpoint variations, scaling, illumination changes, partial occlusion, and background clutter. Many approaches and algorithms are proposed and implemented to overcome these challenges.**

In this paper, we will discuss the current computer vision literature on 3D object recognition. We will introduce an overview of the current approaches of some important problems in visual recognition, to analyze their strengths and weaknesses. Finally, we will present particular challenges in 3D object recognition approaches that have been used recently. As well as, possible directions for future research will be presented in this field.

I. Introduction

Object recognition is considered to be one of the high level computer vision problems. Object recognition research community has been split it into two parts: Those who deal with 2D images and those who deal with 3D pointclouds or meshes. 2D images are created by projecting the scene onto a plane by capturing the light intensity detected at each pixel. Alternatively, 3D pointclouds captures the 3D coordinates of points in the scene. The main difference between these two types of data is that 3D data includes depth information whereas 2D does not. Cheaper sensors have been developed to acquire the 3D data from real environment, such as RGB-D cameras [1, 2]. RGB-D cameras, such as the Microsoft Kinect, capture a regular color image (RGB) along with the depth (D) associated with each pixel in the image.

On the other hand, object recognition is generally considered by vision researchers as having two types [1]: the instance level and the category level recognitions. In the instance level, we try to identify distinct object instances. For example, a coffee mug instance is one coffee mug with a particular appearance and shape. In contrast, the category level recognition determines the category name of an object, for examples, building, computers, or cars.

To evaluate an object recognition technique, each application imposes different requirements and constraints, such as [3]:

1) *Evaluation time:* Especially in industrial applications, the data has to be processed in real time. Of course, evaluation time depends strongly upon the number of pixels covered by the object as well as the size of the image area to be examined.
2) *Accuracy:* In some applications, the object position has to be determined very accurately. The error bounds must not exceed a fraction of a pixel.
3) *Recognition reliability:* All recognition techniques try to reduce the rates of "false alarms" (e.g., correct objects erroneously classified as "defect") and "false positives" (e.g., objects with defects erroneously classified as "correct") as much as possible.
4) *Invariance:* Virtually, every algorithm has to be insensitive to some kind of variance of the object to be detected. Depending on the application, it is worthwhile to achieve invariance with respect to [4]:
   a) *Illumination*: Gray scale intensity appearance of an object depends on illumination strength, angle, and color. In general, the object should be recognized regardless of the illumination changes.
   b) *Scale:* The area of pixels, which is covered by an object, depends on the distance of the object to the image acquisition system. Algorithms should compensate for variations of scale.
   c) *Rotation:* The rotation of the object is not known a priori and should be determined by the system.

d) *Background clutter:* Especially natural images don't show only the object, but also contain background information. This background can vary significantly for the same object. The recognition technique shouldn't be influenced by background variation.
e) *Occlusion:* Sometimes, the system cannot rely on the fact that the whole object is shown in a scene image. Some parts might be occluded by other objects.
f) *Viewpoint changes:* The image formation process projects a 3D-object located in 3D space onto a 2D-plane (the image plane). Therefore, the 2D-appearance depends strongly on the relative position of the camera to the object (the viewpoint), which is unknown for some applications. The design of the object recognition algorithm should aim at ensuring at least partial invariance for a certain viewpoint range.

The paper is organized into 7 sections. Section 2 discusses briefly the different object recognition approaches. Section 3 introduces the 3D object recognition and how to generate 3D data. In section 4, we explain in detail how today's state-of-the-art techniques utilize local features to identify an object in a new image. This entails efficient algorithms for local feature extraction. Section 5 reviews related works about using image features in 3D object recognition. Section 6 lists the 3D object recognition challenges, and looks at some future research directions in this area. Finally, conclusion and future work will be discussed in section 7.

II. Object Recognition Approaches

Many object recognition techniques have been implemented over multiple decades. Object recognition approaches can be classified according to number of characteristics. In this paper, we focus on model acquisition (learning) and invariance to image formation conditions. Therefore, the object recognition techniques are categorized into four groups: geometry-based methods, appearance-based methods, three-dimensional object recognition schemes, and descriptor-based methods [3, 5, 6, 7]. In geometry- or model-based object recognition, the knowledge of an object's appearance is provided by the user as an explicit CAD-like model. Typically, this model only describes the 3D shape and omits other properties such as color and texture [3, 5, 6]. Appearance-based methods do not required explicit user-provided model in object recognition. The object representations are usually acquired through an automatic learning phase, and the model typically relies on surface reflectance properties [6]. Some methods intend to locate the 3D position of an object in a single 2D image, essentially by searching for features which are invariant to viewpoint position. Descriptor-based approaches represent the object as a collection of descriptors derived from local neighborhoods around characteristic points of the image [3].

A. Geometry-Based Methods

Early attempts of object recognition were focused on using geometric models of objects to account for their appearance variation due to viewpoint and illumination changes. The main idea is that the geometric description of a 3D object allows the projected shape to be accurately predicate in a 2D image under projective projection, so facilitating recognition process using edge or boundary information (which is invariant to certain illumination changes). Most attention was made to extract geometric primitives (e.g., lines, circles, etc.) that are invariant to viewpoint change. It has been shown that such primitives can only be reliably extracted under limited conditions (controlled variation in lighting and viewpoint with certain occlusion) [5].

Geometry base techniques for object recognition have many advantages, such as [8, 9]:

- Invariance to viewpoint: Geometric object descriptions allow the projected shape of an object to be accurately predicted under perspective projection.
- Invariance to illumination: Recognizing geometric descriptions from images can be achieved using edge detection and geometric boundary segmentation. Such descriptions are reasonably invariant to illumination variations.
- Well developed theory: Geometry has been under active investigation by mathematicians for thousands of years. The geometric framework has achieved a high degree of maturity and effective algorithms exist for analyzing and manipulating geometric structures.
- Man-made objects: A large fraction of manufactured objects are designed using computer-aided design (CAD) models and therefore are naturally described by primitive geometric elements, such as planes and spheres. More complex shapes are also represented with simple geometric descriptions, such as a triangular mesh.

B. Appearance-Based Methods

In contrast, most recent efforts have been centered on appearance-based techniques as advanced feature descriptors and pattern recognition algorithms. The core idea of these techniques is to compute eigenvectors from a set of vectors where each one represents one face image as a raster scan vector of gray-scale pixel values. Each eigenvector, dubbed as an eigenface, captures certain variance among all the vectors, and a small set of eigenvectors captures almost all the appearance variation of face images in the training set. Given a test image represented as a vector of gray-scale pixel values, its identity is determined by finding the nearest neighbor of this vector after being projected onto a subspace spanned by a set of eigenvectors. In other words, each face image can be represented by a linear combination of eigenfaces with minimum error, and this linear combination constitutes a compact reorientation [5].

Appearance based methods typically include two phases [10, 11, 12]. In the first phase, a model is constructed from a set of reference images. The set includes the appearance of the object under different orientations, different illuminants and potentially multiple instances of a class of objects, for example faces. The images are highly correlated and can be efficiently compressed using e.g. Karhunen-Loeve transformation (also known as Principal Component Analysis - PCA) [13]. In the second phase, parts of the input image (subimages of the same size as the training images) are extracted, possibly by segmentation (by texture, color, motion) or by exhaustive enumeration of image windows over whole image. The recognition system then compares an extracted part of the input image with the reference images (e.g. by projecting the part to the Karhunen-Loeve space) [6].
A major limitation of the appearance-based approaches is that they require isolation of the complete object of interest from the background. They are thus sensitive to occlusion and require good segmentation [14, 15].

C. *Three-Dimensional Object Recognition Schemes*

Some applications require a position estimate in 3D space (and not just in the 2D image plane), e.g., bin picking applications, where individual objects have to be gripped by a robot from an unordered set of objects. Typically, such applications utilize sensor systems which allow for the generation of 3D data and perform matching in 3D space. Another way to determine the 3D pose of an object is to estimate the projection of the object location in 3D space onto a 2D camera image. Many of the methods utilize so-called range images or depth maps, where information about the z-direction (e.g., z-distance to the sensor) is stored dependent on the [*x*,*y*]-position in the image plane. Such a data representation is not "full" 3D yet and therefore is often called 2½D [3].

D. *Descriptor-based Methods*

When object recognition has to be performed in "real-world" scenes, characterization with geometric primitives like lines or circular arcs is not suitable. Another point is that the algorithm must compensate for heavy background clutter and occlusion, which is problematic for global appearance methods. In order to cope with partial occlusion, local evaluation of image information is required. Additionally, gradient-based shape information may not be enough when dealing with a large number of similar objects or objects with smooth brightness transitions. To this end, Schmid and Mohr [16] suggested a two-stage strategy for the description of the image content: the first step consists of the detection of so called interest/key points i.e., points exhibiting some kind of salient characteristic like a corner. Subsequently, for each interest point a feature vector called region descriptor is calculated. Each region descriptor characterizes the image information available in a local neighborhood around one interest point, as shown in Fig. 1.

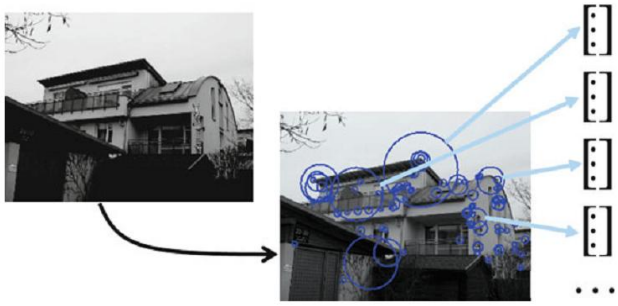

Figure 1. Illustrative example of the strategy suggested by Schmid and Mohr [15]: first, interest regions are detected (middle part, indicated by blue circles). Second, a descriptor is calculated for each interest region (right part) [3].

Object recognition can then be performed by comparing information of region descriptors detected in a scene image to a model database. Usually the model database is created in an automated manner during a training phase. During the last decade, there has been extensive research on this approach to object recognition and many different alternatives for interest point detection and region descriptors have been suggested [17, 18].

III. 3D Object Recognition

With the advent of new generation depth sensors, the use of three-dimensional (3-D) data is becoming increasingly popular [19]. 3D object recognition can be defined as the problem of finding all instances of the database models in an arbitrary scene with determining the pose of the detected objects. The pose of the object is the rigid transformation that aligns the object in the database to the object's instance in the scene [1].
In 2D, the rigid transformation has three components. They are two translations in each of the *x* and y directions and a rotation in the *xy*-plane. In 3D, the pose has six components which are the translations in each of the of *x*, *y*, and *z* directions as well as a rotation about each of these three axes. Therefore, solving the problem of 3D object recognition is the problem of finding a known object in the scene along with its 6D pose [1].
3D data can be captured from a multitude of methods including 2D images and acquired sensor data. Acquisition from 2D images, 3D data acquisition and object reconstruction can be performed using stereo image pairs. Stereo photogrammetry or photogrammetry based on a block of overlapped images is the primary approach for 3D mapping and object reconstruction using 2D images. Acquisition from acquired sensor data, Can using a variety of sensors, including [1, 20]: stereo cameras, time of fight laser scanners such as LiDARs, as well as infrared sensors such as the Microsoft Kinect or Panasonic DI-Imager. All of these sensors can only capture a single view of the object with a single scan. This view is referred to as a 2½D scan of the object. Therefore, to capture the entire 3D shape of the object, the sensor captures multiple instances of the object from different viewpoints [1, 20].

3D object recognition includes the important process of determining the similarity between the scene object and the model stored in a database. This is often performed by computing the distance between feature vectors. In general, the recognition has to search among the possible candidate features for identification of the best match and then assign the label to the matched object in the scene [21]. Based on the characteristics of the shape descriptor, the 3D object recognition methods can be divided into two main categories: global feature methods and local feature methods [21]. Generally, global feature techniques can characterize the global shape features of a 3D model. The implementation of these methods is relatively direct with an easily defined distance for comparison in the matching process. Since the global features are not especially discriminative about object details or undesirably sensitive to optical occlusion, these methods are still yet to be further applied to practical applications. On the other hand, these methods may be employed as an active pre-filter, or they can be used in combination with other methods to improve performance [21]. The other methods are based on local feature characteristics. Local features can be extracted from interest points or local regions. Fig. 2 shows the architecture of a general three-dimensional object recognition system.

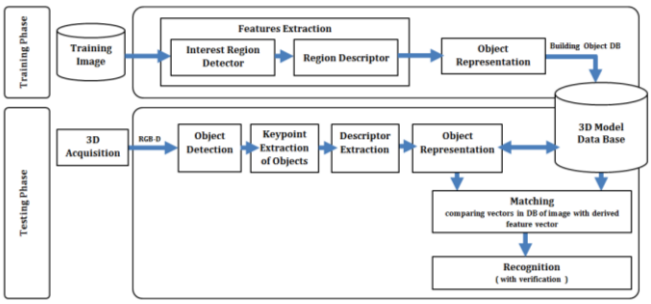

Figure 2. A general three-dimensional object recognition system.

IV. Feature-based Object Recognition Techniques

Extracting the points from an image that can give the best definition for an object are called keypoints/features and they are very important and valuable. These features have many applications in image processing like object detection, object and shape recognition, and object tracking. By extracting the features, we can use them for finding objects in other images. If the keypoints are correctly identified, they achieve the best information from the image [22].

A. *Harris Corner Detector*

Harris and Stephens [23] developed an approach to extract corners and infer the contents of an image. Corner detection is frequently used in many applications, such as motion detection, image registration, video tracking, panorama stitching, 3D modeling, and object recognition. Corner detection overlaps with the topic of interest point detection [24]. The Harris corner detector is popular because it is independent to rotation, scale, and illumination variations. However, the Shi-Tomasi corner detector [25], the one implemented in OpenCV library [26], is an improvement of this corner detector [19]. A corner is so special because, since it is the intersection of two edges, it represents a point in which the directions of these two edges change. Hence, the gradient of the image (in both directions) has a high variance, which can be used to detect it [27].

B. *The SIFT Algorithm*

Lowe [28] developed a feature detection and description technique call SIFT (Scale Invariant Feature Transformation). This means that an image is looked for important points. These points, called keypoints, are then extracted and described as a vector. The resulting vectors can be used to find reliable matches between different images for object recognition, camera calibration, 3D reconstruction, and many other applications [29].

SIFT consists of three basic stages. First, the keypoints are extracted from the image. Then, these keypoints are described as 128 vectors. Finally, the last step is the *matching* stage. Several stored vectors in the database are matched against the calculated vectors of the tested image using the Euclidian distance. Fig. 3 gives an overview of the three main stages of the SIFT technique. In the top figure, the extracted keypoints are drawn on the image using arrows. The length of the arrow represents the scale of the keypoint, while the angle of the arrow represents the orientation of the keypoint. The middle figure shows how a keypoint is described. The third figure shows another example in which the box in the right image has to be found in the left image [29].

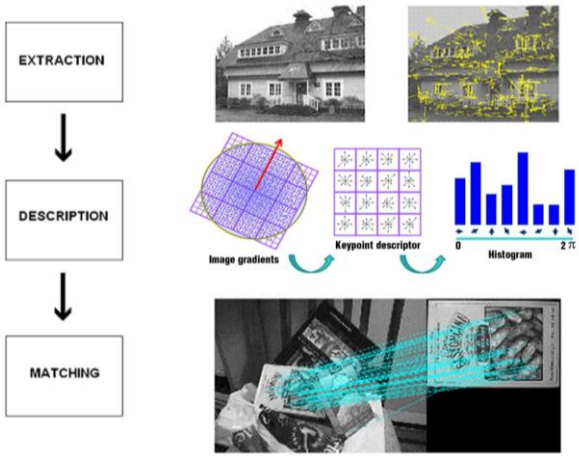

Figure 3. Graphical representation of the SIFT algorithm and a matching example [21].

The SIFT is used as a benchmark for many propositions of interest point detectors and region descriptors. It is a strong hint of its good performance, especially in situations with heavy occlusion and/or clutter. A particular strength of this technique is that each step is carefully designed and, additionally, all steps work hand in hand and are well coordinated [3]. However, this technique only works well if a

significant number of keypoints can be detected in order to generate enough descriptor information and therefore relies heavily on the keypoint detector performance [3].

C. The SURF Algorithm

SURF is developed by Bay et al. [30] and it stands for Speeded Up Robust Features. SURF algorithm is actually based on the SIFT algorithm [28]. It uses integral images and approximations for achieving higher speed than SIFT. These integral images are used for convolution. Like SIFT, SURF works in three main stages: *extraction*, *description*, and *matching*. The difference between SIFT and SURF is that SURF extracts the features from an image using integral images and box filters. The extraction of the keypoints from an image is a process that requires image filtering. SURF implements these filters using box filters. A very interesting pre-processing step is the conversion of the original image into a so-called integral image [29].

Integral images are very easily computed by adding the right pixel values. In an integral image every pixel is the sum of all pixels located in a rectangular window formed by that pixel and the origin, with the origin being the most top-left pixel. Box filters are used as an approximation of the exact filter masks. By using integral images together with box filters a major speed up is realized. Another difference in the extraction of keypoints is that SIFT rescales the image, while SURF changes the filter mask. The term box-space is used to distinguish it from the usual scale-space. While the scale space is obtained by convolution of the initial images with Gaussians, the discrete box-space is obtained by convolving the original image with box filters at several different discrete sizes. In the detection step, the local maxima of a Hessian-like operator, the Box Hessian operator, applied to the box-space are computed to select interest point candidates. These candidates are then validated if the response is above a given threshold. Both box size and location of these candidates are then refined using an iterated procedure fitting locally a quadratic function. Typically, a few hundreds of interest points are detected in a digital image of one mega-pixel [31]. Therefore, SURF builds a descriptor that is invariant to view-point changes of the local neighborhood of the point of interest. Like in SIFT, the location of this point in the box-space provides invariance to scale and provides scale and translation invariance. To achieve rotation invariance, a dominant orientation is defined by considering the local gradient orientation distribution, estimated with Haar wavelets. Making use of a spatial localization grid, a 64-dimensional descriptor is then built, corresponding to a local histogram of the Haar wavelet responses [33].

D. The FAST Algorithm

FAST (Features from Accelerated Segment Test) is an algorithm proposed originally by Drummond [32] for identifying interest points in an image. An interest point can be defined as a pixel in an image which has a well-defined position and can be robustly detected. Interest points have high local information content and they should be ideally repeatable between different images. Interest point detection has applications in image matching, object recognition, tracking etc. [33].

The basic idea behind this approach is to reduce the number of calculations which are necessary at each pixel in order to decide whether a keypoint is detected at the pixel or not. This is done by placing a circle consisting of 16 pixels centered at the pixel under investigation. For the corner test, only gray value differences between each of the 16 circle pixels and the center pixel are evaluated, as shown Fig. 4 [3].

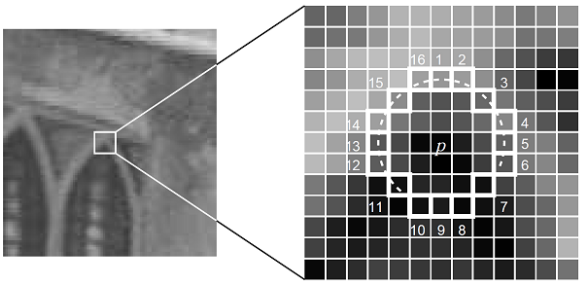

Figure 4. Image showing the interest point under test and the 16 pixels on the circle [25].

In the first step , a center pixel p is labeled as “corner” if there exist at least n consecutive “circle pixels”. If at least three of the four pixel values - I_1, I_5, I_9, I_{13} are not above or below I_P + T (I_P denotes intensity of this pixel, T threshold intensity value), then p is not an interest point (corner). In this case reject the pixel p as a possible interest point. Else if at least three of the pixels are above or below I_p + T, then check for all 16 pixels and check if 12 contiguous pixels fall in the criterion. Repeat the procedure for all the pixels in the image [33]. Therefore, a feature is detected by non-maximum suppression in a second step. To this end, compute a score function V for each of the detected points. The score function is defined as: “The sum of the absolute difference between the pixels in the contiguous arc and the center pixel”. Then, consider two adjacent interest points, compare their V values. Discard the one with the lower V value. The entire process can be summarized mathematically as follows:

$$V = \max \begin{cases} \sum(\text{pixel values} - p) \text{ if } (\text{value} - p) > t \\ \sum(p - \text{pixel values}) \text{ if } (p - \text{value}) > t \end{cases}$$

Compared to the other corner detectors, Rosten and Drummond [32] reported that the FAST algorithm is significantly faster (about 20 times faster than the Harris detector and about 50 times faster than the DoG detector of the SIFT scheme). Surprisingly, tests of Rosten and Drummond with empirical data revealed that the reliability of keypoint detection of the FAST detector is equal or even superior to other corner detectors in many situations.

On the other hand, FAST algorithm is more sensitive to noise (which stems from the fact that for speed reasons the number of pixels evaluated at each position is reduced) and does not

provide neither scale nor rotation information for the descriptor calculation [3].

E. The ORB Algorithm

A number of binary visual descriptors have been recently proposed in the literature, including BRIEF and ORB. These techniques have several advantages over the more established vector-based descriptors such as SIFT and SURF [34]. ORB [35] is scale and rotation invariant, robust to noise and affine transformations. The algorithm is actually a combination of the FAST keypoint detection with oriented added to the algorithm. On the other hand, the BRIEF (Binary Robust Independent Elementary Features) keypoint descriptor algorithm modified to handle oriented keypoints [36]. FAST and BRIEF are provide high performance and have low cost. A fast and accurate orientation component is added to FAST, also efficient computation of oriented BRIEF features, analysis of variance and correlation of oriented BRIEF features, and a learning method for de-correlating BRIEF features under rotational invariance, leading to better performance in nearest-neighbor applications [37].

F. The Bag-Of- Features Algorithm

A Bag of Features method [38] is one that represents images as orderless collections of local features. The name comes from the Bag of Words representation used in textual information retrieval. There are two common perspectives for explaining the BoF image representation. The first is by analogy to the Bag of Words representation. The Bag of Features image representation is analogous. A visual vocabulary is constructed to represent the dictionary by clustering features extracted from a set of training images. The image features represent local areas of the image, just as words are local features of a document. Clustering is required so that a discrete vocabulary can be generated from millions of local features sampled from the training data. Each feature cluster is a visual word. Given a novel image, features are detected and assigned to their nearest matching terms (cluster centers) from the visual vocabulary. The second way to explain the BoF image representation is from a codebook perspective. Features are extracted from training images and vector quantized to develop a visual codebook. A novel image's features are assigned the nearest code in the codebook. The image is reduced to the set of codes it contains, represented as a histogram. The normalized histogram of codes is exactly the same as the normalized histogram of visual words, yet is motivated from a different point of view.
At a high level, the procedure for generating a Bag of Features image representation is shown in follow Figure and summarized as follows [38]:

1) *Build Vocabulary:* Extract features from all images in a training set. Vector quantize, or cluster, these features into a "visual vocabulary," where each cluster represents a "visual word" or "term." In some works, the vocabulary is called the "visual codebook." Terms in the vocabulary are the codes in the codebook.
2) *Assign Terms:* Extract features from a novel image. Use Nearest Neighbors or a related strategy to assign the features to the closest terms in the vocabulary.
3) *Generate Term Vector:* Record the counts of each term that appears in the image to create a normalized histogram representing a "term vector." This term vector is the Bag of Features representation of the image. Term vectors may also be represented in ways other than simple term frequency, as discussed later.

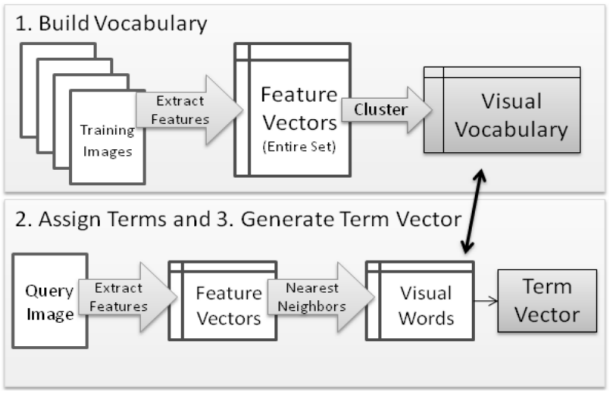

Figure 5. Process for Bag of Features Image Representation [37].

G. The MSER Algorithm

The maximally stable extremal region (MSER) detector described by Matas et al. [38] is a further example of a detector for blob-like structures. Its algorithmic principle is based on thresholding the image with a variable brightness threshold. Imagine a binarization of a scene image depending on a gray value threshold *t*. All pixels with gray value below *t* are set to zero/black in the thresholded image, all pixels with gray value equal or above *t* are set to one/bright. Starting from $t = 0$ the threshold is increased successively. At the beginning, the thresholded image is completely bright. As *t* increases, black areas will appear in the binarized image, which grow and finally merge together. Some black areas will be stable for a large range of *t*. These are the MSER regions, revealing a position (e.g., the center point) as well as a characteristic scale derived from region size as input data for region descriptor calculation. Altogether, all regions of the scene image are detected which are significantly darker than their surroundings. Inverting the image and repeating the same procedure with the inverted image reveals characteristic bright regions, respectively. The MSER detector reveals rather few regions, but their detection is very stable. Additionally, the MSER detector is invariant with respect to affine transformations, which makes it suitable for applications which have to deal with viewpoint changes [3].

V. 3D Object Recognition Based on Image Features

There have been numerous previous efforts in 3D object recognition. This section gives a brief overview of some

previous work targeting the recognition of 3D objects based on image features. We discuss this issue from three different aspects: 3D object recognition from range images, 3D local descriptors, and 3D object recognition based on stereo vision. In 3D object recognition from range images (or depth maps) used as sensor input data, range images sensors collect large amounts of three-dimensional coordinate data from visible surfaces in a scene and can be used efficiently in 3D object recognition. Many different 3D local descriptors have been proposed in the computer vision literature. They have been successful in a variety of applications, including 3D object recognition. These descriptors can be computed efficiently, are resistant to clutter and partial occlusion, and are somewhat insensitive to pose, i.e. they change relatively slowly as the view of the object changes. Lastly, Stereo vision systems determine depth (i.e. distance to real world objects) from two or more images which are taken at the same time from slightly different viewpoints. The most important and time consuming task for a stereo vision system is the registration of both images, i.e. the identification of corresponding pixels. Two pixels are corresponding when they represent the same point in the real world. Area-based stereo attempts to determine the correspondence for every pixel, which results in a dense depth map.

A. 3D Object Recognition from Range Images

Tang et al. [40] presented an object recognition system which leverages the additional sensing and calibration information available in a robotics setting together with large amounts of training data to build high fidelity object models for a dataset of textured household objects. They demonstrated how these models can be used for highly accurate detection and pose estimation in an end-to-end robotic perception system incorporating simultaneous segmentation, object classification, and pose fitting. This system can handle occlusions, illumination changes, multiple objects, and multiple instances of the same object. They would like to investigate extensions to non-rigid or textureless objects. They would also like to investigate high fidelity 3D rendering approaches to verification.

Lai et al. [41] addressed joint object category and instance recognition in the context of RGB-D (depth) cameras. Motivated by local distance learning, where a novel view of an object is compared to individual views of previously seen objects, defined a view-to-object distance where a novel view is compared simultaneously to all views of a previous object. This novel distance is based on a weighted combination of feature differences between views. The proposed **I**nstance **D**istance **L**earning provides a distance measure for evaluating the similarity of a view to a known set of objects. They showed that using both shape and visual features achieved higher performance than either set of cues alone for both category and instance recognition.

Kim and Medioni [42] presented an approach to object recognition that boosted dissimilarity between queried objects and similar-shaped background objects in the scene by maximizing use of the visibility context. They designed a point pair feature containing discriminative description inferred from the visibility context. Also, they proposed a pose estimation method that accurately localized objects using these point pair matches. Two measures of validity were suggested to discard false detections. With 10 query objects, their approach was evaluated on depth images of cluttered office scenes captured from a real-time range sensor. The experimental results demonstrated that their method remarkably outperforms two state-of-the-art methods in terms of recognition (recall & precision) and runtime performance.

Kounalakis and Triantafyllidis [43] presented a system that is fusing efficient and state-of-the-art techniques of stereo vision and machine learning, aiming at object detection and recognition. The system initially creates depth maps by employing the Graph-Cut technique. Then, the depth information is used for object detection by separating the objects from the whole scene. Next, the SIFT is used, providing the system with unique object's feature key-points, which are employed in training an Artificial Neural Network (ANN). The system is then able to classify and recognize the nature of these objects, creating knowledge from the real world. Future work will be routed to advanced database management.

B. 3D Local Descriptors

Bo et al. [44] presented and designed a family of kernel descriptors which provide an unified and principled framework to turn pixel attributes (gradient, color, local binary pattern, etc.) into compact patch-level features. Introduced three types of match kernels to measure similarities between image patches, and construct compact low-dimensional kernel descriptors from these match kernels using kernel principal component analysis (KPCA). Kernel descriptors are easy to design and can turn any type of pixel attribute into patch-level features.

Bo et al. [45] proposed hierarchical kernel descriptors for extracting image features layer by layer. Their approach was based on the observation that kernel descriptors can be recursively used to produce features at different levels. They have compared hierarchical kernel descriptors to current state-of-the-art algorithms and shown that their hierarchical kernel descriptors have the best accuracy on CIFAR10 [46, 47], a large scale visual object recognition dataset to date. In addition, they also evaluated their hierarchical kernel descriptors on a large RGB-D dataset and demonstrated their ability to generate rich feature set from multiple sensor modalities, which was critical for boosting accuracy.

Bo et al. [48] proposed and studied a range of local features over a depth image and show that for object recognition they were superior to pose-invariant features like Spin Images. They presented five depth kernel descriptors (gradient, kernel PCA, size, spin and local binary pattern kernel descriptors) that capture different recognition cues including size, shape

and edges (depth discontinuities). Extensive experiments suggest that these novel features complement each other and their combination significantly boosts the accuracy of object recognition on an RGB-D object dataset as compared to state-of-the-art techniques.

C. 3D Object Recognition Based on Stereo Vision

Yoon et al. [49] presented a framework for general 3D object recognition, which is based on the invariant local features and their 3D information with stereo cameras. They extended the conventional object recognition framework for stereo cameras. Since, the proposed method was based on the stereo vision, it is possible to utilize 3D information of local features visible from two cameras. Can fully use stereo cameras without any additional cost. One simple way of using stereo cameras is to apply the conventional object recognition methods to both images independently. However, it does not fully utilize the 3D Fig. 6. Overall structure information obtained from stereo cameras.

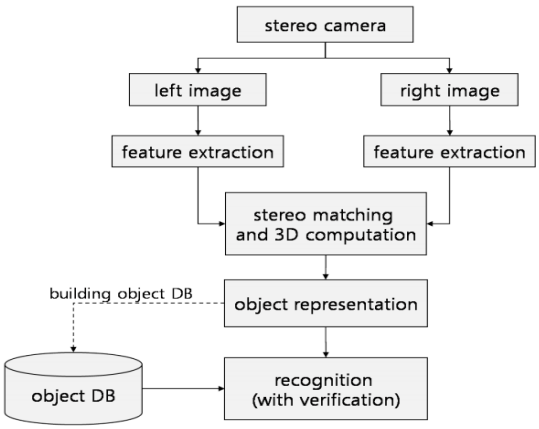

Figure 6. Overall structure [49].

Object recognition method in this proposed framework consists of four stages:

1) *Local feature extraction and matching:* The first step for local feature-based object recognition is to extract proper local features. Here, local features should be representative, invariant under photometric and geometric changes, and robust to image noise. The extracted features from a left image and a right image are then matched across images. After finding correspondences, Compute the 3D coordinates of matched local features with respect to the camera.
2) *3D Object representation:* An object is commonly represented by the set of un-ordered individual features.
3) *Generation of recognition hypothesis:* In the learning stage, construct for all object to be recognized.
4) *Hypothesis testing:* The selected candidates are then verified in the last stage. Have 2D image coordinates of extracted local features and 3D coordinates of stereo matched local features among all local features. Therefore, it is possible to use both the 2D homography and the 3D rigid transformation for verification.

VI. Challenges and Future Research Directions of 3D Object Recognition

In recent, the most popular approaches to object recognition represent an object by feature vectors, and the recognition problem is framed as one of supervised learning - training a classifier given a set of positive and negative examples [50].
On the other hand, matching and learning visual objects is a challenging problem. Instances of the same object category can generate very different images, depending on confounding variables such as illumination conditions, object pose, camera viewpoint, partial occlusions, and unrelated background "clutter". Different instances of objects from the same category can also exhibit significant variations in appearance. Furthermore, in many cases appearance alone is ambiguous when considered in isolation, making it necessary to model not just the object class itself, but also its relationship to the scene context and priors on usual occurrences [51].
Today's 3D recognition algorithms also face notable challenges in computational complexity and scalability. Highly efficient algorithms are necessary to accommodate rich high-dimensional image representations, to search large image databases, or to extend recognition to thousands of category types. In addition, scalability concerns also arise when designing a recognition system's training data: while unambiguously labeled image examples tend to be most informative, they are also more expensive to obtain. Thus, methods today must consider the tradeoffs between the extent of costly manual supervision an algorithm requires versus the advantages given to the learning process [51, 52].
There are other issues that add to the challenge of solving the problem of 3D object recognition and they include [1]:

- Occlusion: A part of the 3D object is always hidden due to self-occlusion or occlusion by other objects in the scene.
- Clutter: A scene may include many closely spaced objects, making it difficult to determine the source object of a data point.
- Noise: Sensors are not perfect and therefore a 3D representation of the same view of an object is never exactly the same and can possibly include missing parts depending on the quality of the sensor.
- Sampling Resolution: The 3D data of objects in the database might be captured using a different sensor with a different sampling rate than the one used to capture the 3D scene data.

We can expect to see better support for 3D depth sensors and combinations of 2D cameras with 3D measurement devices. In support of better object recognition, we can expect

a full-function tool kit that will have a framework for interchangeable interest-point detection and interchangeable keys for interest-point identification. This will include popular features such as SURF, HoG, Shape Context, MSER, Geometric Blur, PHOG, PHOW, and others. Support for 2D and 3D features is planned.

VII. Conclusion and Future Works

We have presented an overview of the literature of the 3D object recognition, pointed out some of the major challenges facing the community and stressed some of the characteristic approaches attempted for solving the recognition problem.
Through this survey we noticed that a great deal of the research focused on passive recognition, to some extent, on the feature selection stage of the recognition problem without taking into consideration the effects of various cost constraints discussed in the survey.

We are seeking to build system of the 3D object recognition based on depth map. So this system is used to recognize real 3D objects. The conventional local feature-based object recognition methods, such as SIFT, SURF, and ORB, that are used to retrieve the strong features of the object.

References

[1] Mohamad, Mustafa. "3D Object Recognition using Local Shape Descriptors." Tech. rep. School of Computing Queen's University Kingston, Ontario, Canada, 2013.

[2] Litomisky, Krystof. "Consumer RGB-D Cameras and their Applications." Tech. rep. University of California, 2012.

[3] Treiber, Marco. An Introduction to Object Recognition Selected Algorithms for a Wide Variety of Applications. Springer-Verlag London Limited, 2010.

[4] Urtasun, Raquel. "Visual Recognition: Introduction". TTI Chicago, 1 2012.

[5] Yang, Ming-Hsuan. "Object Recognition". University of California at Merced. n.d.

[6] Matas, Jir and Stepan Obdrzalek. "Object recognition methods based on transformation covariant features." Proc of European signal processing conf on EUSIPCO (2004).

[7] Outline of object recognition. http://en.wikipedia.org/wiki/Outline_of_object_recognition. 3 2014.

[8] Mundy, Joseph L. "Object Recognition in the Geometric Era: a Retrospective." Toward category-level object recognition. Springer Berlin Heidelberg (2006).

[9] Ponce, Jean, ed. Toward category-level object recognition. Springer, 2006.

[10] T.E. Boult, R.S. Blum, S.K. Nayar, P.K. Allen, and J.R. Kender. Advanced visual sensor systems (1998). In DARPA98, pages 939–952, 1998.

[11] Matthew Turk and Alex Pentland. Eigenfaces for recognition. Journal of Cognitive Neuroscience, 3(1):71–86, 1991.

[12] A. Leonardis and H. Bischoff. Dealing with occlusions in the eigenspace approach. In IEEE Conference on Computer Vision and Pattern Recognition, pages 453–458, June 1996.

[13] Principal component analysis. http://en.wikipedia.org/wiki/Principal_component_analysis.3 2014.

[14] H. Murase and S.K. Nayar. Image spotting of 3d objects using parametric eigenspace representation. In SCIA95, pages 325–332, 1995.

[15] B. Moghaddam and A. Pentland. Probabilistic visual learn-ing for object detection. In International Conference on Computer Vision (ICCV'95), pages 786–793, Cambridge, USA, June 1995.

[16] Schmid, C. and R. Mohr. "Local Grey Value Invariants for Image Retrieval." IEEE Transactions on Pattern Analysis and Machine Intelligence, 19:530–535 (1997).

[17] Pinz, A., "Object Categorization", Foundations and Trends in Computer Graphics and Vision, 1(4):255–353, 2005

[18] Roth, P.M. and Winter, M., "Survey of Appearance-based Methods for Object Recognition", Technical Report ICG-TR-01/08 TU Graz, 2008

[19] Aitor Aldoma, Zoltan-Csaba Marton, Federico Tombari Walter Wohlkinger Christian Potthast Bernhard Zeisl Radu Bogdan Rusu Suat Gedikli and Markus Vincze. "Three-Dimensional Object Recognition and 6 DoF Pose Estimation." IEEE ROBOTICS \& AUTOMATION MAGAZINE (2012).

[20] 3D data acquisition and object reconstruction. http://en.wikipedia.org/wiki/3D_data_acquisition_and_object_reconstruction. 2 2014.

[21] Liang-Chia Chen, Hoang Hong Hai, Xuan-Loc Nguyen and Hsiao-Wen Wu. "Novel 3-D Object Recognition Methodology Employing a Curvature-Based Histogram." International Journal of Advanced Robotic Systems 10 (2013).

[22] Oji, Reza. "An Automatic Algorithm For Object Recognition And Detection Based On Asift Keypoints." arXiv preprint arXiv:1211.5829 (2012).

[23] Harris, C. and M. Stephens. "A Combined Corner and Edge Detector." Alvey Vision Conference, 147–151 (1988).

[24] Corner detection. http://en.wikipedia.org/wiki/Corner_detection. 3 2014.

[25] Harris Corner Detector. http://www.aishack.in/2010/04/harris-corner-detector/. 4 2010.

[26] Welcome to opencv documentation. http://docs.opencv.org/trunk/.

[27] Harris corner detector. http://docs.opencv.org/doc/tutorials/features2d/trackingmotion/harris_detector/harris_detector.html.

[28] Lowe, D.G. "Distinctive Image Features from Scale-Invariant Viewpoints." International Journal of Computer Vision, 60:91–111 (2004).

[29] Kristof, Van Beeck and Heylen Filip. "Implementation of the SURF algorithm in an Embedded Platform." Ph.D. dissertation. 2008.

[30] Herbert Bay, Tinne Tuytelaars and Luc Van Gool. "SURF: Speeded Up Robust Features." Computer Vision–ECCV 2006. Springer Berlin Heidelberg (2006).

[31] Oyallon, Edouard and Julien Rabin. "An analysis and implementation of the SURF method, and its comparison to SIFT." Image Processing On Line (2013).

[32] Rosten, E. and T. Drummond. "Machine learning for high speed corner detection." in 9th Euproean Conference on Computer Vision, vol. 1 (2006).

[33] Viswanathan, Deepak Geetha. "Features from Accelerated Segment Test (FAST)." (n.d.).

[34] Muja, Marius, and David G. Lowe. "Fast matching of binary features." Computer and Robot Vision (CRV), 2012 Ninth Conference on. IEEE, 2012.

[35] Rublee, Ethan, et al. "ORB: an efficient alternative to SIFT or SURF." Computer Vision (ICCV), 2011 IEEE International Conference on. IEEE, 2011.

[36] brahmbhatt, samarth. Practical OpenCV. Technology in Action, 2013.

[37] Kulkarni, A. V., J. S. Jagtap, and V. K. Harpale. "Object recognition with ORB and its Implementation on FPGA."

[38] O'hara, Stephen and draper, bruce a. "Introduction to the Bag of Features Paradigm for Image Classification and Retrieval." arXiv preprint arXiv:1101.3354 (2011).

[39] Matas, J., Chum O. Martin U. and T. Pajdla. "RobustWide Baseline Stereo formMaximally Stable Extremal Regions." Proceedings of British Machine Vision Conference, 1:384–393 (2002).

[40] Tang, Jie, et al. "A textured object recognition pipeline for color and depth image data." Robotics and Automation (ICRA), 2012 IEEE International Conference on. IEEE, (2012): 8.

[41] Lai, Kevin, et al. "Sparse Distance Learning for Object Recognition Combining RGB and Depth Information." Robotics and Automation (ICRA), 2011 IEEE International Conference on. IEEE (2011): 7.

[42] Kim, Eunyoung and Gerard Medioni. "3D Object Recognition in Range Images Using Visibility Context." Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on. IEEE. (2011): 8.

[43] Kounalakis, T. and G. A. Triantafyllidis. "3D scene's object detection and recognition using depth layers and SIFT-based machine learning." 3D Research 2.3 (2011): 11.

[44] Bo, Liefeng, Xiaofeng Ren and Dieter Fox. "Kernel Descriptors for Visual Recognition." NIPS. Vol. 1. No. 2. (2010).

[45] Liefeng Bo, Kevin Lai, Xiaofeng Ren Dieter Fox. "Object Recognition with Hierarchical Kernel Descriptors." Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on IEEE (2011): 8.

[46] A. Torralba, R. Fergus, and W. Freeman. 80 million tiny images: A large data set for nonparametric object and scene recognition. IEEE PAMI, 30(11):1958–1970, 2008.

[47] A. Krizhevsky. "Learning multiple layers of features from tiny images". Technical report, 2009.

[48] Bo, Liefeng, Xiaofeng Ren and Dieter Fox. "Depth kernel descriptors for object recognition. "Depth Kernel Descriptors for Object Recognition." Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on. IEEE (2011): 6.

[49] Kuk-Jin Yoon, Min-Gil Shin and Ji-Hyo Lee. "Recognizing 3D Objects with 3D Information from Stereo Vision." Pattern Recognition (ICPR), 2010 20th International Conference on. IEEE (2010).

[50] Jean Ponce, Svetlana Lazebnik, Fredrick Rothganger Cordelia Schmid. "Toward True 3D Object Recognition." Reconnaissance de Formes et Intelligence Artificielle. (2004).

[51] Bradski, Gary and Adrian Kaehler. Learning OpenCV. Ed. Mike Loukides. O'Reilly Media, 2008.

[52] Grauman, Kristen and Bastian Leibe. "Visual Object Recognition" University of Texas at Austin and RWTH Aachen University, 2010.