Visual navigation method for indoor mobile robot based on extended BoW model

Xianghui Li¹, Xinde Li², Mohammad Omar Khyam³, Chaomin Luo³, Yingzi Tan¹

¹Key Laboratory of Measurement and Control of CSE (Ministry of Education), School of Automation, Southeast University, Nanjing 210096, People’s Republic of China
²Department of Electrical and Computer Engineering, Faculty of Engineering, National University of Singapore, 117580, Singapore
³Department of Electrical and Computer Engineering, University of Detroit Mercy, Michigan, USA
⁴School of Cyber Science and Engineering, Southeast University, Nanjing, 211189, People’s Republic of China
✉ E-mail: xindel@seu.edu.cn

Abstract: This article proposes a new navigation method for mobile robots based on an extended bag of words (BoW) model for general object recognition in indoor environments. The scale-invariant feature transform (SIFT)-detection algorithm with the graphic processing unit (GPU) acceleration technology is used to describe feature vectors in this model. First, in order to add some redundant image information, statistical information of the spatial relationships of all the feature points in an image, i.e. relative distances and angles, is used to extend the feature vectors in the original BoW model. Then, the support vector machine (SVM) classifier is used to classify objects. Also, in order to navigate conveniently in unknown and dynamic indoor environments, a type of human–robot interaction method based on a hand-drawn semantic map is considered. The experimental results show that this new navigation technology for indoor mobile robots is very robust and highly effective.

1 Introduction

In the research field of mobile robots, navigation is very important and necessary, with its aim to make a robot move to the desired destination and while completing specified tasks as expected.

The first step in navigation is to build a model of the environment which can be a grid, geometric, topological, or 3D one. However, Thrun proposed a hybrid method that combines grid and topological models [1, 2], using the grid to represent local features and a Voronoi graph [3] topological structures. Wu Xuejian et al. also proposed a visual navigation method for mobile robots based on a hand-drawn map [4], which provides the start and end information and approximate distance between these points, the approximate positions of landmarks the robot might meet during navigation etc. Relative to traditional map-building models, such as grid, geometric, and topological ones, we use the convenient hand-drawn map [5]. Although some researchers used artificial [6–10] and quasi-artificial landmarks (with some taken photos of natural ones in advance [4, 5]), recognising natural ones is essential for navigating successfully in real environments. However, unlike an artificial landmark, a natural one is not labelled.

Also, we should not restrict natural landmarks to specific objects due to the changeable environment; in other words, even if they change to some extent or adopt another form, a robot should still be capable of recognising them using a visual sensor. In order to achieve our goal, we solve the problem of general object recognition in a real environment.

The bag of words (BoW) model is effective arithmetical means of general object recognition because of its simple strategy and robustness for determining an object’s position and deformation in an image. However, as each feature is independent of the others in it, there is no spatial relationship to consider whereas such relationships among features could be useful for describing the internal structures of objects or highlighting the importance of their contextual visual information. Although research on this theme is becoming more and more popular [11–19], there is still room for improvement.

Here, we propose a novel extended BoW method which works in the following statistical way. A multi-dimensional vector is used to describe an image with all its elements divided into two parts: one describes its local features; and the other is the spatial relationships among them. Then, we use the support vector machine (SVM) classifier to train our model to obtain discriminant functions and, taking real time into account, graphic processing unit (GPU) acceleration arithmetic to speed up image processing. Finally, this method is successfully applied to indoor mobile robot navigation.

2 Building model of environment

Building a model of the environment aimed at planning a path for a mobile robot involves feature extraction and information representation.

It is not necessary for many biological systems, such as those of human beings, butterflies, and bees, to obtain precise distance information when perceiving their environments through their visual systems as they navigate by remembering some key landmarks based on qualitative analysis.

Based on the method proposed in [5], we design a hand-drawn map for guiding a robot’s navigation with the advantage that we do not need to input detailed environmental information into the robot and enables our navigation model to handle dynamic situations, such as when some landmarks change often or a person walks without stopping around the robot. We require only the starting point and orientation of the robot, the route and its approximate physical distance, and rough estimates of the locations of landmarks the robot might encounter during navigation.

Our novel extended BoW model for general object recognition proposed to overcome the challenge of recognising natural landmarks is discussed in the next section.

3 Recognition of landmarks

Recognising different objects which belong in the same category and filtering complex backgrounds are the most difficult problems that might be faced in the course of general object recognition in a real environment. Every determinate object has its own characteristics,
such as local parts and spatial relationships with others. While a human being can understand the advanced semantic features of a picture, a computer can comprehend only the raw information in an image. However, it is still very helpful to refer to the human vision system because the course of general object recognition is analogous to a human’s judgment, that is, first, descriptors of general objects are established, then their categories determined through machine learning and, finally, the learned model used to classify and recognise new objects [20].

3.1 Overview of recognition algorithm

The method proposed here, the framework of which is presented in Fig. 1, follows the principle of general object recognition, that is, first, it describes an image, then learns the object model and, finally, classifies objects.

3.2 Building vision code library

In 1999, Professor G. David [21] proposed the scale-invariant feature transform (SIFT) algorithm based on the scale space, which is invariant in terms of translation, rotation and zoom, and then improved it in 2004 [22]. It is widely used for object recognition, and has a very strong image-matching capability with the following general characteristics.

(a) A SIFT feature is a local feature of an image which remains invariant of translation, rotation, zoom, illumination, occlusion, and noise and even maintains some degree of stability with a change in vision and affine transformation.
(b) It has an abundance of information which is very helpful for fast and accurate image matching.
(c) It is fast which may satisfy the real-time requirements of image matching after optimisation.
(d) It has strong extendibility which may be associated with other feature vectors.

Therefore, here, we choose the SIFT algorithm to detect key points; for example, if we want to construct a code library of cars, first we choose pictures of different cars from different views and then detect key points using this algorithm.

After completing feature detection, we need to establish a vision code library using the large number of ‘words’ we obtain from detecting many images. However, as some SIFT descriptors are similar to those of other algorithms, it is necessary to cluster these vision words. K-means is a common clustering method [23] which aims to group elements in K centres according to the distance between an element and a centre, and obtain the clustering centres based on continuous iterations; for example, if the elements for clustering are \((x_1, x_2, x_3, ..., x_{n-1}, x_n)\) and every element is a \(d\)-dimensional vector, they are expected to cluster into K sets according to the sum of their minimum average variances \((s = \{s_1, s_2, s_3, ..., s_{k-1}, s_k\})\) as:

\[
\arg\min_k \sum_{i=1}^{k} \sum_{s_i \in s_k} ||x_i - \mu_k||^2
\]  

(1)

where \(\mu_k\) is the average of \(s_k\) [24].

Based on experience, we adopt \(k=600\) and complete the construction of a vision code library in which every code is a 128-dimension SIFT description vector after K-means clustering.

3.3 Pre-processing images

As a computer can understand only raw information, acquiring an advanced meaning that reflects an object’s appearance is the major difficulty of general object recognition.

As previously mentioned, we obtain a vision code library of certain objects and, before describing an image, compute the similarity between a local feature and every word in the library. If a similarity meets the threshold, this local feature is considered a key point that belongs to this image; for example, if there are \(N\) vision codes in a library and \(M\) local features in an image, the pseudo-code is as follows.

**Algorithm:**

While \(i < M\)

While \(j < N\)

If similarity\((P_i, Q_j)\) < Threshold

Return true;

Else return false.

where \(P_i\) denotes the SIFT descriptor of the \(i\)th local feature in an image and \(Q_j\) that of the \(j\)th vision code in the library. We define

\[
\text{similarity}(P_i, Q_j) = \frac{|P_i - Q_j|}{|P_i| \times |Q_j|}
\]  

(2)

After performing local feature extraction, we take the normalised operation for every SIFT descriptor and believe that, although the remaining local features belong to the image, some are produced from the background. Therefore, an extra operation is required to delete them if: (i) the number of local features obtained from an image is much greater than that from the background after computing similarities; or (ii) we want to further reduce the background disturbance based on the density distribution of the local features.

Fig. 2 shows the results if we obtain \(T\) local features from \(M\) ones after computing similarities.
Obviously, if we want to obtain the object in the rectangle despite some disturbance, we may use RANSAC to reduce the negative influence on later image descriptions. For convenience, we use a circle to cover the area where the density of features is very high using the following pseudo-code.

While iteration < Times

The points inside the model equal to the key points that were randomly selected from T datum, with the possible centre of a circle:

\[
\text{maybe_center} = (x_0, y_0) = \frac{1}{P} \left( \sum_{i=1}^{P} x_i, \sum_{i=1}^{P} y_i \right)
\]

(3)

The radius (R) can be defined as the maximum of the distances between the key points in the model and the possible centre of a circle.

For every key point that doesn’t belong to the model, if the distance is <1.2R, consider that it does belong and add 1 to the number of key points in the model.

If the number of key points in the model is larger than E(\(E = 80\% \times T\)), consider this model to be correct and save the possible centre of a circle and key points.

For \(j\) number of correct models:

\[
\text{distance}(j) = \sum_{i=1}^{P} |\text{keypoint}(i) - \text{radius}(j)|
\]

(4)

If the distance \((r, r \in [1, T])\) is the minimum of all distances, save the model’s possible radius \((r)\) and 80% of its key points closest to the possible centre and consider that they belong to the image.

3.4 Describing images

In our BoW model, a multi-dimensional vector was used to describe an image with all its elements grouped in two parts, one of which describes its local features and the other describes the spatial relationships among them.

(a) The local features are described based on the numbers of times words appear in the vision code; for example, if there are \((x_0, x_1, x_2, ..., x_{p-2}, x_{p-1})\) vision words in the library, the dimension of the vector is \(P\), with the number of each dimension denoting the number of times the corresponding vision code appears.

(b) To describe the spatial relationships among the local features, we use the distance between a key point and the centre of a circle, and the relative angle. The new centre of the key points is expressed in (5), where \(m\) indicates the number of key points after processing, and the geometric centre is the centre of a circle. As shown in Fig. 3, the marks around it denote key points, for example, for the five stars in the upper right corner, the corresponding distance and angle are \(L\) and \(\theta\), respectively.

\[
(x, y) = \frac{1}{m} \left( \sum_{i=1}^{m} x_i, \sum_{i=1}^{m} y_i \right)
\]

(5)

The Euclidean distances between every key point and the geometric centre \((x', y')\) were calculated as the distance, i.e. from \((l_1, l_2, l_3, ..., l_{m-1}, l_m)\). We take the middle distance as the unit length \((L)\) and divide all the remaining ones into \(0 - 0.5L, 0.5L - 1.5L, 1.5L - MAX\) according to the ratio \(l_1/L\).

A random key point were chosen and the angles were calculated between its half line and those of all the other key points to the centre of the circle, i.e. \(\theta\) in Fig. 3. We obtain these angles after simple mathematical transformations \((\theta_1, \theta_2, \theta_3, ..., \theta_{m-1}, \theta_m)\), with that of one key point measured counter-clockwise. In fact, each angle is not very large and we divide them into five ranges of \(0° - 30°, 30° - 60°, 60° - 90°, 90° - 120°\) and \(120° - MAX\).

Therefore, \(4 \times 5 = 20\) spatial relationships were generated and can be described every image using the vector as \((x_0, x_1, x_2, ..., x_{p-2}, x_{p-1}, y_1, y_2, ..., y_{q-2}, y_{q-1})\).

The dimension vectors \((P\) and \(Q)\) stand for the numbers of times each vision word and a certain spatial relationship appear in an image, respectively. Since the distances and angles are relative and the descriptions of spatial relationships are invariant of translation, rotation and zoom.

3.5 Obtaining discriminant function

During navigation, the camera continually acquires images and arrives at judgments according to the discriminant function which is trained offline and obtained as follows.

Generally speaking, there are two kinds of classifiers that depend on the degree to which a human participates during learning, i.e. supervised and unsupervised. As the SVM has attracted a great deal of attention and recently achieved some good results, we use it to train our models as supervised classifiers. Its aim is to classify two kinds of patterns as far as possible according to the principle of the minimum structural risk by constructing a discriminant function.

The training set is supposed to be \(\{x_i, y_i\}, i = 1, ..., l, x_i \in \mathbb{R}^n, y_i \in \{-1, 1\}\), which can be separated by a hyperplane \((\langle w \cdot x \rangle + b = 0)\), with a linear hyperplane at a distance \((\Delta)\) from the samples:

\[
\langle w \cdot x \rangle + b = (w \cdot x) + b = 0, \quad \|w\| = 1,
\]

\[
\text{if} \quad \langle w \cdot x \rangle + b \geq \Delta, \quad y = 1,
\]

\[
\text{if} \quad \langle w \cdot x \rangle + b \leq -\Delta, \quad y = -1.
\]

Then, the distance between two linear hyperplanes is \(M = 2\Delta/\|w\|\).

During the course of our training, images containing objects are regarded as positive inputs and others negative ones. From them, we can obtain the trained SVM discriminant function for general object recognition offline which is very helpful for a robot’s navigational recognition.

3.6 Recognising landmarks online

The discriminant functions obtained offline, including those of many different kinds of objects, are used to build a database which a robot uses to recognise landmarks.

During the running of a robot, its camera continually acquires image information and then every image is processed using the SIFT algorithm to obtain feature points. After computing similarities, the feature points that meet the specified threshold are saved and calculated according to RANSAC arithmetic to reduce background disturbance. Every image is represented as a \(P + Q\) dimension vector recognised by the offline discriminant functions in the database. Then, a series of recognition results is obtained to localise the robot itself, the framework of which is shown in Fig. 4.

We can summarise our recognition algorithm in two parts, i.e. offline and online.

(i) Recognition of landmarks offline: in the phase of training and learning, \(I = \{l_1, l_2, l_3, ..., l_{m-1}, l_m\}\), where \(I\) is a set of images for generating visual words, \(A = \{A_1, A_2, A_3, ..., A_{p-2}, A_{p-1}, A_p\}\), where \(A\) is a set of \(n\) training images containing target objects with every image marked as \(+1\) when trained; and
\[ B = \{B_1, B_2, B_3, ..., B_{t-2}, B_{t-1}, B_t\}, \]

where \( B \) is a set of training images not containing target objects with every image marked as \(-1\) when trained. This phase involves the following three steps:

(a) A visual code library of image \( I \) was generated;
(b) every image in \( A \) and \( B \) was represented as a multi-dimensional vector with background disturbance reduced;
(c) the SVM was used to train images and finally obtain the discriminant functions.

(ii) Recognition of landmarks online:

(a) an image was obtained, and its background disturbance was reduced and represented as a multi-dimensional vector; and
(b) landmarks were recognised by using discriminant functions.

4 Robot navigation

4.1 Navigation arithmetic

The navigation flowchart shown in Fig. 5 is clearly explained in [5]. However, our proposed general object recognition method for recognising natural landmarks for robot navigation is different and requires photos of landmarks to be taken manually in advance and regarded as image-matching templates.

4.2 GPU acceleration during image processing

SIFT algorithm was used mainly to detect and acquire key points in the course of general object recognition by a 128-dimension vector. These points are some local extreme ones containing orientation information which are detected in different scale spaces of an image and can be described from the three views of scale, size and orientation. As this process takes a long time, it is necessary to use GPU acceleration to speed up the SIFT algorithm which occupies most of our algorithm’s time.

The GPU is a concept with a great advantage over a central processing unit (CPU) for image processing and the NVIDIA released an official development platform, i.e. CUDA, on which we can compute SIFT code in a parallel manner. We test images of different sizes with different numbers of key points using: operating system – 32 bit win7; flash memory – 2G; CPU – Intel(R) Core (TM) 2Duo E7500@2.93 GHz; GPU – nVIDIA GeForce310;
dedicated image memory – 512 MB; shared system memory – 766 MB; and compiler environment – Visual Studio 2010.

In the testing results in Table 1, the acceleration is obvious when an image is large and has many key points.

5 Experimental results

5.1 Experimental environment

A Pioneer 3-DX mobile robot equipped with a PTZ monocular colour camera, 16 sonar sensors, a speedometer, an electronic compass etc., is chosen for our experiments which are conducted in the SEU mobile robot laboratory shown in Fig. 6. The size is ~10 m × 8 m.

5.2 Specific experiments

5.2.1 Experiment 1: As our proposed algorithm aims to recognise general objects in this experiment, its main goal is to recognise different ones in the same category. This experiment is conducted three times and there are five kinds of key landmarks each time, i.e. chair, guitar, wastebasket, umbrella, and fan, which might be changed every time by retaining their corresponding same categories. The robot always starts from the lower left-hand corner and finishes at the upper right-hand one. The hand-drawn map and three paths it runs, respectively, during its navigation were shown in Fig. 7.

As can be seen, even if the landmarks are changed by their corresponding same categories, our algorithm can still work well. Therefore, the robot always reaches its destination successfully.

5.2.2 Experiment 2: The robustness of the navigation algorithm is tested when the landmarks were moved slightly. In the first navigation, all the positions of the landmarks and hand-drawn route remain unchanged while; in the second one, they are moved 1 m to the left and; in the third one, they are moved 1 m to the right.

As can be seen in Fig. 8, the robot can still reach its intended destination even if the landmarks are moved slightly from their original positions.

5.2.3 Experiment 3: The robot’s navigation performance is tested by reducing the number of landmarks from 5 the first time to 4 the second time and 3 the third time.

As can be seen in Fig. 9, there is almost no effect on the path when the number of landmarks is reduced. However, if the environment is

Table 1 Comparisons with and without GPU

| Size   | Number of feature points | Time of common SIFT (ms) wasted | Time of GPU SIFT (ms) wasted | Ratio of time |
|--------|--------------------------|--------------------------------|------------------------------|---------------|
| 183 × 275 | 61                        | 78                              | 55                           | 1.418         |
| 205 × 246 | 57                        | 93                              | 62                           | 1.500         |
| 161 × 262 | 173                       | 125                             | 63                           | 1.984         |
| 225 × 320 | 71                        | 140                             | 72                           | 1.944         |
| 240 × 320 | 133                       | 157                             | 81                           | 1.938         |
| 240 × 240 | 340                       | 234                             | 78                           | 3.000         |
| 240 × 240 | 641                       | 628                             | 95                           | 8.373         |
| 377 × 508 | 1171                      | 687                             | 160                          | 8.373         |
| 480 × 640 | 2368                      | 1264                            | 244                          | 5.180         |
| 1024 × 1024 | 589                  | 1294                            | 170                          | 7.612         |
| 1000 × 1000 | 1376                 | 1513                            | 169                          | 8.953         |
| 1200 × 1600 | 4052                 | 6690                            | 392                          | 17.066         |

Fig. 7 Experiment 1: recognition of the same category of objects in different forms

Fig. 8 Experiment 2: navigation with the movement of landmark in different directions

Fig. 9 Experiment 3: navigation with the reduced number of landmarks
and spatial consistency of the robot is affected by the landmarks it 
can find on the hand-drawn map. However, the robot arrives at its 
destination based on other landmarks and the odometer as 
what is shown on the hand-drawn map. However, the robot arrives 
too large and the number of landmarks too few, the robot might not 
perform very well in terms of navigation according to the sketched 
map due to lacking of some necessary information. In this case, 
the odometer plays an important role.

5.2.4 Experiment 4: By setting obstacles in the path the robot 
might cover according to the hand-drawn map, we aim to test 
its capability to avoid them by recording its reactions when facing 
them. As shown in Fig. 10, the robot tries to avoid the obstacles it might 
encounter according to the navigation algorithm by using sonar 
sensors.

5.2.5 Experiment 5: To further test the robot’s navigational 
performance, first, the landmarks on the hand-drawn map 
correspond to those in a real environment; second, the fourth 
landmark is the rubbish bin while the real one is the chair. 
As can be seen in Fig. 11, during its second navigation, the robot 
cannot find the fourth landmark because it does not correspond to 
what is shown on the hand-drawn map. However, the robot arrives at 
its destination based on other landmarks and the odometer as 
well as the map.

6 Conclusions and future work

Here, we proposed a novel general object recognition arithmetic 
method using offline training and learning and online recognition 
which is helpful for recognising natural landmarks and even 
human–robot interactions. We successfully applied it to robot 
navigation based on a hand-sketchd map and natural landmarks, 
with the experimental results proving its advantages. Due to rapid 
changes in the world over time, categories of objects have become 
increasingly complex. As our current level of recognition 
arithmetic is very limited, it is important that our next work is on 
online learning and training for recognition.

7 Acknowledgment

This work was supported in part by the National Natural Science 
Foundation of China under Grant 61573097, 91748106, in part by 
Key Laboratory of Integrated Automation of Process Industry 
(PAL-N201704), in part by the Qing Lan Project and Six Major 
Top-talent Plan, and in part by the Priority Academic Program 
Development of Jiangsu Higher Education Institutions.

References

[1] Thrun, S.: ‘Robotic mapping: a survey.’ CMU-CS-02-111, School of Computer 
Science, Carnegie Mellon University, Pittsburgh, 2002
[2] So, S., Lowe, D.: ‘Mobile robot localization and mapping with uncertainty using 
scale-invariant visual landmarks’, Int. J. Rob. Res., 2002, 21, (8), pp. 735–738
[3] Shamos, M.I., Hoey, D.: ‘Closest-point problems’. 16th Annual Symp. on 
Foundations of Computer Science, Berkeley, USA, 1975, pp. 151–162
[4] Wu, X.: ‘Visual navigation method research for mobile based on hand-drawn map’. (Nanjing Southeast University, Nanjing, 2011)
[5] Li, X., Zhang, X., Dai, X.: ‘An interactive visual navigation method using a 
hand-drawn-route-map in an unknown dynamic environment’, Int J Fuzzy Syst., 2011, 13, (4), pp. 311–322
[6] Kawamura, K., Peters, R.A.I., Wilkes, D.M., et al.: ‘Toward perception-based 
navigation using EgoSphere’. Proc. for SPIE 4573, Mobile Robots XVI, 
Boston, MA, United States, October 2001
[7] Kawamura, K., Koku, A.B., Wilkes, D.M., et al.: ‘Toward egocentric navigation’, 
Int J Rob. Autom., 2002, 17, (4), pp. 135–145
[8] Chronis, G., Skubic, M.: ‘Sketch-based navigation for mobile robots’. Proc. of 
the 12th IEEE Int. Conf on Fuzzy Systems (FUZZ-IEEE), St. Louis, USA, 
2003, vol. 1, pp. 284–289
[9] Skubic, M., Bailey, C., Chronis, G.: ‘A sketch interface for mobile robots’. Proc. 
of the IEEE Int. Conf. on Systems, Man and Cybernetics (SMC), WashingtonDC, 
USA, 2003, vol. 1, pp. 919–924
[10] Skubic, M., Billsard, S., Bailey, C., et al.: ‘Qualitative analysis of sketched route maps: translating a sketch into linguistic descriptions’, IEEE Trans. Syst. Man 
Cybern., 2004, 34, (2), pp. 1275–1282
[11] Li, T., Mei, T., Kwon, I., et al.: ‘Contextual bag-of-words for visual 
categorization’, IEEE Trans. Circuits Syst. Video Technol., 2011, 21, (4), pp. 381–392
[12] Liu, T., Liu, J., Liu, Q.S., et al.: ‘Expanded bag of words representation for object 
classification’. Processing of 2009 16th IEEE Int. Conf. on Image, Cairo, Egypt, 
2009, pp. 297–300
[13] Agarwal, S., Awan, A., Roth, D.: ‘Learning to detect objects in images via a 
sparsely, part-based representation’, IEEE Trans. Pattern Anal. Mach. Intell., 
2003, 25, (11), pp. 1475–1490
[14] Liu, X.L., Lou, Y.H., Yu, A.W., et al.: ‘Search by mobile image based on visual and spatial consistency’. 2011 IEEE Int. Conf. on Multimedia and Expo (ICME), 
Barcelona, Spain, 2011, pp. 1–6
[15] Cheng, X., Wang, J., Chia, L., et al.: ‘Learning to combine multi-resolution 
spatially-weighted co-occurrence matrices for image representation’. 2010 IEEE 
Int. Conf. on Multimedia and Expo (ICME), Suntec City, Singapore, 2010, pp. 631–636
[16] Sun, M., Hamme, V.: ‘Image pattern discovery by using the spatial closeness of visual code words’. 2011 18th IEEE Int. Conf. on Image Processing, Brussels, 
Belgium, 2011, pp. 205–208
[17] Elsayad, I., Martinet, J., Urruty, T., et al.: ‘A new spatial weighting scheme for 
bag-of-visual-words’. 2010 Int. Workshop on Content-Based Multimedia 
Indexing, Grenoble, France, 2010, pp. 1–6
[18] Lazebnik, S., Schmid, C., Ponce, J.: ‘Beyond bags of features: spatial pyramidal 
matching for recognizing natural scene cat- egories’. Proc. of the IEEE 
Computer Vision and Pattern Recognition, New York, USA, 2006, pp. 2169–2178
[19] Zhang, L., Wang, C., Xiao, B., et al.: ‘Image representation using bag-of-features’. Acta Autom. Sin., 2012, 38, (11), pp. 46–54
[20] Chen, H.: ‘Research on classifier based on distinguished learning’ 
(University of science and technology of China, He Fei, China, 2009)
[21] David, G.: ‘Object recognition from local scale-invariant feature’. Proc. of 
the Seventh IEEE Int. Conf. on Computer Vision, Kerkyra, Greece, 1999, 
pp. 1150–1157
[22] David, G.: ‘Distinctive image features from scale-invariant key points’, 
Int. J. Comput. Vis., 2004, 60, (2), pp. 91–110
[23] Hartigan, J.A.: ‘Clustering algorithms’ (John Wiley and Sons, Inc., New York, 
NY, USA, 1975)
[24] Sivic, J., Zisserman, A.: ‘Video google: a text retrieval approach to object 
matching in video’. Proc. of IEEE Int. Conf. on Computer Vision, Nice, 
France, 2003, vol. 2, pp. 1470–1477