A Study of Approaches for Object Recognition

Wong Yuk-Man

Term Paper for the Degree of
Master of Philosophy
in
Computer Science and Engineering

Supervised by

Prof. Michael R. Lyu

November 2005

Abstract

Object recognition is an important part of computer vision because it is closely related to the success of many computer vision applications. A number of object recognition algorithms and systems have been proposed for a long time in order to address this problem. Yet, there still lacks a general and comprehensive solution. Recently, both model-based object recognition approach and view-based object recognition approach have demonstrated good performance on a variety of object recognition problems. In this paper, the most representative papers of each approach are first reviewed. After that, some possible extensions to the recently proposed view-based approach is proposed and discussed in the paper.

Contents

Chapter 1

Introduction

Object recognition is a task of finding 3-dimensional (3D) objects from two-dimensional (2D) images and classifying them into one of the many known object types. It is an important part of computer vision because it is closely related to the success of many computer vision applications such as robotics, surveillance, registration and manipulation etc. A number of object recognition algorithms and systems have been proposed for a long time toward this problem. Yet, a general and comprehensive solution to this problem has not be made.

1.1 Model-Based Object Recognition

In model-based object recognition, a 3D model of the object being recognized is available. The 3D model contains detailed information about the object, including the shape of its structure, the spatial relationship between its parts and its appearance. This 3D model provides prior knowledge to the problem being solved. This knowledge, in principle, can be used to resolve the potential confusion caused by structural complexity and provide tolerance to noisy or missing data. There are two common ways to approach this problem. The first approach involves obtaining 3D information of an object from images and then comparing it with the object models. To obtain 3D information, specialized hardware, such as stereo vision camera, is required to provide the 3D information in some forms. The second approach requires less hardware support but is more difficult. It first obtains the 2D representation of the structure of the object and then compares it with the 2D projections of the generative model.

Using 3D model has both the advantages and the disadvantages. On one side, explicit 3D models provide a framework that allows powerful geometric constraints to be used to achieve good effect. Other model features can

be predicted from just a few detected features based on the geometric constraints. On the other side, using models sacrifice its generality. The model schemas severely limit the sort of objects that they can represent and it is quite difficult and time-consuming to obtain the models.

1.2 View-Based Object Recognition

In view-based object recognition, 3D model of the object is not available. The only known information is a number of representations of the same object viewed at different angles and distances. The representations of the object can be obtained by taking a series of images of the same object in a panorama fashion. Most of these operate by comparing a 2D, image representation of object appearance against many representations stored in a memory and finding the closest match. Matching of this type of recognition is simpler but the space requirements for representing all the views of the object is large. Again, there are many ways to approach this problem. One of the common way is to extract salient information, such as corner points, edges and region etc, from the image and match to the information obtained from the image database. [17] Another common approach extracts translation, rotation and scale invariant features, such as SIFT, GLOH and RIFT, from each image and compares them to the features in the feature database [13, 14].

View-based object recognition systems have the advantage of greater generality and more easily trainable from visual data. View-based approach is generally a useful technique. However, since matching is done by comparing the entire objects, some methods are more sensitive to background clutter and occlusion. Some methods solve this problem by applying image segmentation on the entire objects so as to divide the image representations into smaller pieces for matching separately. Some other methods avoid using segmentation and solve the problem by employing voting techniques, like Hough transform methods. This technique allows evidence from disconnected parts to be effectively combined.

□ End of chapter.

Chapter 2

Model-Based Object Recognition

Generative models are the commonly used models in modeling object for matching. They are models described by mathematical functions and operators and is sufficiently complete to generate images of target objects. One of the common usage of model-based object recognition is face modeling and recognition. Generative face models can generate realistic images of a human face, with changeable facial expression and pose. The general steps to recognize an object using model is:

1. Locate the object,
2. locate and label its structure,
3. adjust the model's parameters until the model generates an image similar enough to the real object.

Generative models that are of particular interest are deformable models because objects in a class are often not identical. To be able to identify all objects within the class, we have to deal with variability. Deformable models are the models that capture the principle components of the class of objects and they can deform to fit any of the object in a class.

A number of models have been proposed to address the problem, they include Active Contour Models, Morphable Models, Active Blobs, Active Shape Models and the recently proposed Active Appearance Models [4, 5] etc. Active Appearance Models have been proved to be highly useful models for face recognition and several major extensions to these models have recently emerged. They are Direct Appearance Models [10] and Inverse Compositional Active Appearance Models [15] etc. We will give a brief description on Active Appearance Models and one of its extensions in the following sections.

2.1 Active Appearance Models (AAMs)

AAMs are non-linear parametric models that are commonly used to model faces. They model shape and appearance of objects separately. Shape of an AAM is defined by the vertex locations of a mesh. In advance, the shape $\mathbf{s}$ can be expressed as the sum of a base shape $\mathbf{s_0}$ and a linear combination of n orthonormal shape vectors s_i. It can be mathematically defined as follow:

$$\mathbf{s} = \mathbf{s_0} + \sum_{i=1}^{n} p_i \mathbf{s_i}$$

The appearance of an AAM is defined based on the base mesh $\mathbf{s_0}$. If we define the set of pixels that lies inside the base mesh as $\mathbf{x} = (x, y)^T$, then the appearance of AAM $A(\mathbf{x})$ as the sum of a base appearance $A_0(\mathbf{x})$ and a linear combination of m appearance images $A_0(\mathbf{x})$

$$A(\mathbf{x}) = A_0(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i A_i(\mathbf{x})$$

We can generate the shape of AAM with solely the AAM shape parameter $\mathbf{p} = (p_1, p_2, ..., p_n)^T$ and generate the appearance of AAM with solely the AAM appearance parameter $\lambda = (\lambda_1, \lambda_2, ..., \lambda_m)^T$. To generate a complete AAM model instance, we need both the shape parameter $\mathbf{p}$ and appearance parameter λ. It can be created by warping the AAM appearance A from the base mesh s_0 to the AAM model shape s. This warping is better represented by Matthews et al. [15] as the piecewise affine warp function $\mathbf{W}(\mathbf{x}, \mathbf{p}$, which can be used to warp any pixel x in s_0 to the corresponding pixel location at s. The AAM model instance M can then be mathematically defined as follow:

$$M(\mathbf{W}(\mathbf{x}, \mathbf{p})) = A(\mathbf{x})$$

As fitting an AAM to an image of object, non-linear optimization solution is applied which iteratively solve for incremental additive updates to the shape and appearance coefficients. Given an input image I, the goal of AAM fitting is to minimize the following error image E with respect tot the shape parameters $\mathbf{p}$ and the appearance parameters λ:

$$E(\mathbf{x}) = A_0(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i A_i(\mathbf{x}) - I(\mathbf{W}(\mathbf{x}, \mathbf{p}))$$

2.2 Inverse Compositional AAMs

Inverse Compositional AAMs are proposed by Matthews et al. [15]. The major difference of these models with AAMs is the fitting algorithm. The fitting algorithm adopted by AAMs is an additive incremental update approach. The algorithm is run to solve for $\Delta\mathbf{p}$ and update the parameter $\mathbf{p}$ to $\mathbf{p}+\Delta\mathbf{p}$. Matthews et al. argued that there cannot be any efficient algorithm that solves for the incremental parameter $\Delta\mathbf{p}$. They believed that another parameter update scheme can be made more efficient. That is inverse compositional algorithm. The algorithm updates the entire warp by composing the current warp with the computed incremental warp with parameters $\Delta\mathbf{p}$ following the following update rule:

$$\mathbf{W}(\mathbf{x},\mathbf{p}) = \mathbf{W}(\mathbf{x},\mathbf{p}) \circ \mathbf{W}(\mathbf{x},\mathbf{\Delta p})^{-1}$$

Here the update function of the affine warp $\mathbf{W}$ is an inverted version of the incremental warp, $\mathbf{W}(\mathbf{x},\mathbf{\Delta p})^{-1}$. The reason is that in inverse compositional algorithm, the roles of the input image and the template $A_0(\mathbf{x})$ are reversed. The incremental warp is computed with respect to the template $A_0(\mathbf{x})$. The error image that the AAM fitting algorithm minimizes becomes:

$$E(\mathbf{x}) = \sum_{\mathbf{x}} [I(\mathbf{W}(\mathbf{x},\mathbf{p})) - A_0(\mathbf{W}(\mathbf{x},\mathbf{\Delta p}))]^2$$

Through these changes, most of the computation in computing $\Delta\mathbf{p}$ can be moved to a pre-computation step and thus this results in a efficient image alignment algorithm.

□ End of chapter.

Chapter 3

View-Based Object Recognition

This type of object recognition is also known as appearance-based recognition. There are many object recognition approaches of this type. Correlation-based template matching [12, 6] is one of the approaches that is very commonly used in commercial object recognition system. It is simple to implement and effective for certain engineered environments. However, this type of method is not effective when the illumination of the environment, the object posture and the scale of the object are allowed to change. The result of this method is even more poor when occlusion may occur and image databases is large. An alternative approach is to use color histogram [20] to locate and match image with the model. While this approach is robust to changing of viewpoint and occlusion, it requires good isolation and segmentation of objects from image.

Another approach is to extract features from the image that are salient and match only to those features when searching all locations for matches. Many possible feature types have been proposed, they includes region [1], shape context [2], line segments [8] and groupings of edges [17] etc. Some of these features are view variant and resolution dependent. They have worked well for certain classes of object, but they are often not detected frequently enough for reliable recognition. Therefore, approach that matches these kinds of features generally has difficulty in dealing with partial visibility and extraneous features. A highly restricted set, such as corners, don't have these problems. They are view invariant, local and highly informative. These feature types can be detected by SUSAN detector [19] and Harris corner detector [9]. Based on these features, image descriptor can be created to increase the matching performance. Schmid & Mohr [18] used the Harris corner detector to automatically detect interest points and create a image descriptor for each point that is invariant to affine transformation and scale. They have also proposed a voting algorithm and semi-local constraints that

make retrieval of features from database efficient, and showed that Harris corner detector is highly repeatable. Lowe [13, 14] pointed out that this approach has a major failing at the corner detection part which examines an image at only a single scale. Because of this failing, attempt has to made to match the image descriptors at a large number of scales. Lowe further extended Schmid & Mohr's approach and created a more distinctive image descriptor which is also more stable to changes in affine projection and illumination. Lowe proposed an efficient method to identify stable invariant key points in scale space such that image descriptor for each key point can be calculated only at the same scale as that of the point.

3.1 Recognition Based on 2D Boundary Fragments

Nelson [16, 17] proposed a method of 3D object recognition based on the use of a general purpose associative memory and a principal views representation. He used automatically, robustly extracted 2D boundary fragments as keys. These keys have sufficient information contents to specify the location, scale and orientation of an associated object and sufficient additional parameters to provide efficient indexing and meaningful verification. The keys extracted from an image are fed into an associative memory to generate a set of all objects that could have produced those keys. He called the results generated from the associative memory as hypotheses. These hypothesis are then fed into a second stage associative memory which maintains the probability of each hypothesis based on the statistics of the occurrence of the keys in the associative memory. Since this recognition approach bases on grouping of local feature rather than global features, it is robust to occlusion and clutter, and does not require prior segmentation.

Nelson carried out experiments using keys based on groups of 2-D boundary fragments. He ran tests with databases built for 6, 12, 18 and 24 objects, shown in Figure 3.1, and obtained overall success rates of 99.6%, 98.7%, 97.4% and 97.0% respectively. The total number of training images for the 24 object database was 1802. Training data consisted of 53 clear images per object, spread fairly uniformly, with approximately 20 degrees between neighboring views. Some examples of extracted key features from some testing images are shown in Figure 3.2. The accuracy of the system reported is good. However, the test cases of this experiment is quite ideal because:

1. The number of training images for each object stored in the database is large.

Figure 3.1: The objects used to test the Nelson's recognition system

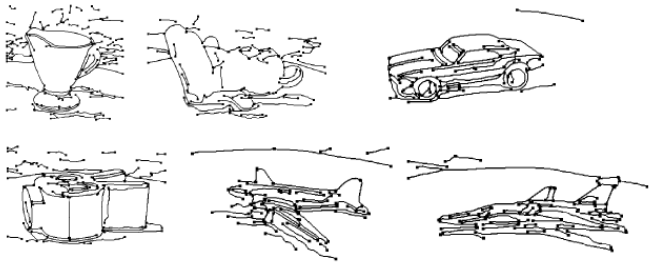

Figure 3.2: The extracted key features of Nelson's testing images

2. The images under test is clean.
3. The objects used were chosen to be different in that they were easy for people to distinguish on the basis of shape.
4. The object viewed from the images has more or less the same scale.
5. There are no occlusion and clutter in every image.

3.2 Recognition Based on SIFT

SIFT stands for Scale Invariant Feature Transform. It is a novel method proposed recently by Lowe [13, 14] for extracting distinctive invariant features from images that can be used to perform reliable matching between different views of an object. The extracted features are invariant to image scale and rotation, which means that the same set of features can be detected after the image is scaled and rotated. Image descriptor for each extracted features is carefully designed to provide robust matching across a range of affine distortion, change in 3D viewpoint, addition of noise and change in illumination to the view of an object. The keypoint descriptors are highly distinctive, which allows a single feature to find its correct match with high probability in a large database of features.

The SIFT features are extracted in a cascade filtering approach in which the more expensive operations are applied only at locations that pass all prior tests. The major steps of generating SIFT features from an image are discussed in the following sections.

3.2.1 Scale-space extrema detection

The first stage of keypoint detection is to identify the locations and scales of the keypoint that can be repeatedly detected under different views of the same object. As the keypoint can be repeatedly detected, we call it stable features. Detecting stable features that are invariant to locations is achieved by searching for most of the locations over the image. To extend its invariance to scales, all possible scales of the image are searched instead of one scale only.

The scale space of an image which is defined as a function, $L(x, y, \sigma)$, can be prepared by repeatedly convolving the initial image with a variable-scale Gaussian function $G(x, y, \sigma)$:

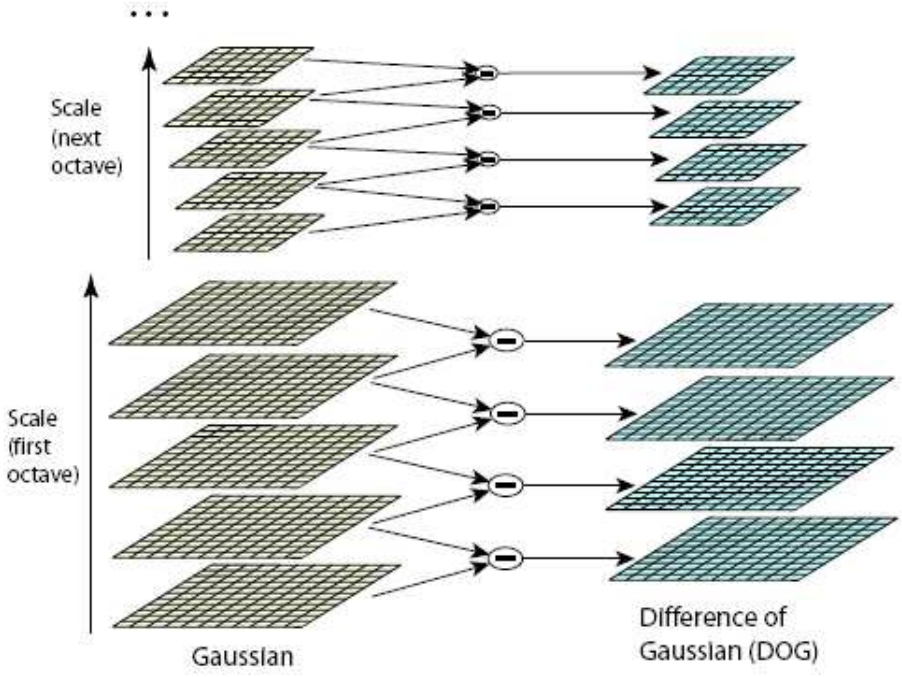

Figure 3.3: A diagram illustrating how differences of gaussian images is prepared from the initial image. The initial image is repeatedly smoothed by Gaussian function, which is shown on the left. Adjacent Gaussian images are subtracted to produce the difference-of-Gaussian images, which is shown on the right.

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2}$$
$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y)$$

To efficiently detect stable keypoint locations in scale space, Lowe proposed [13] using scale-space extrema in the difference-of-Gaussian function, $D(x, y, \sigma)$, which can be computed from the difference of two nearby scales of smoothed images, $L(x, y, \sigma)$, separated by a multiplicative factor k.

$$\begin{aligned} D(x, y, \sigma) &= (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \\ &= L(x, y, k\sigma) - L(x, y, \sigma) \end{aligned}$$

The scale space of the input image is prepared in the way illustrated by Figure 3.3. The difference-of-Gaussian function has been proved to be a close

approximation to the scale-normalized Laplacian of Gaussian. Therefore, finding extrema in difference-of-Gaussian space is approximately equivalent to finding extrema in Laplacian space. After the scale space has been prepared, each sample point is compared to its eight neighbors in the current image and nine neighbors in the scale above and below in order to detect the extrema of $D(x, y, \sigma)$.

The advantage of searching keypoint over a complete range of scales is that both small keypoints and large keypoints are detected. Small keypoints help solving occlusion problem while large keypoints contribute to the robustness of the system toward noise and image blur.

3.2.2 Keypoint localization

The second step is to reject the keypoints that have low contrast or are localized along an edge. Low contrast keypoints are rejected because they are sensitive to noise. Keypoints localized along an edge are also rejected because they in general do not make significant difference with nearby points.

To reject keypoints with low contrast, the scale-space function value at each extremum, $D(\hat{x})$, is examined:

$$D(\hat{x}) = D + \frac{1}{2}\frac{\delta D}{\delta x}^T \hat{x}$$

For the experiments done by Lowe in [14], all extrema with a value of $|D(\hat{x})|$ less than 0.03 were discarded.

To reject keypoints on edges, Hessian edge detector is applied. The difference-of-Gaussian function, D, will have a large principal curvature across the edge but a small one in the perpendicular direction. Hessian matrix, $\mathbf{H}$, can be computed at the location and scale of the keypoint by:

$$\mathbf{H} = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix}$$

The derivatives, D_{xx}, D_{xy} and D_{yy}, can be estimated by taking differences of neighboring points around the sampling keypoint.

The eigenvalues of $\mathbf{H}$ are proportional to the principal curvatures of D. Thus, the ratio of the two eigenvalues reflects that the keypoint is on the edge or not. The solution can be simplified by just checking the following condition:

$$\frac{Tr(\mathbf{H})^2}{Det(\mathbf{H})} < \frac{(r+1)^2}{r}$$

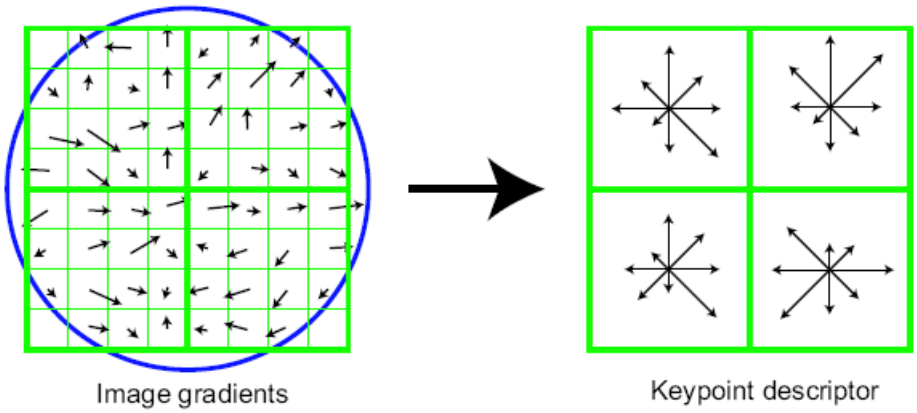

Figure 3.4: Computation of a keypoint descriptor based on the gradient and orientation of each image sample point in a region around the keypoint.

For the experiments done by Lowe in [14], all extrema having a ratio between the principal curvatures greater than 10 are discarded.

3.2.3 Orientation assignment

This is the third major step of SIFT. The aim of this step is to pre-compute the gradient magnitude, $m(x, y)$, and orientation, $\theta(x, y)$, of the each image sample, $L(x, y)$, at each scale by using pixel differences:

$$m(x,y) = \sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2}$$
$$\theta(x,y) = \tan^{-1} \frac{L(x,y+1) - L(x,y-1)}{L(x+1,y) - L(x-1,y)}$$

The gradient and orientation information of each sample point can then be used to construct the keypoint descriptor for each keypoint in the next step.

3.2.4 Keypoint descriptor computation

This is the last step of SIFT. It is to compute a descriptor for the local image region that is highly distinctive.

The computation of the keypoint descriptor is illustrated in Figure 3.4. The approach is to create orientation histograms over 4×4 sample regions around the keypoint locations. Each histogram contain 8 orientation bins

that is the Gaussian-weighted aggregate of the gradient vectors over the corresponding region. For the case illustrated in Figure 3.4, a 32 element feature vector can be obtained for each keypoint.

3.2.5 Object recognition

Lowe described an approach to use SIFT features for object recognition. His approach first matches extracted features to a database of features from known objects using a fast nearest-neighbor algorithm. Among all the matches, some matches are mismatch to the wrong objects. Thus the second step is to filter out the wrong matches, this is done by identifying clusters of features belonging to a single object using an efficient hash table implementation of the generalized Hough transform. This is because the probability that several features will match to the same object is by chance much lower than the probability that any individual feature mismatches. Finally each cluster that agree on an object is subjected to a verification process in which the pose of the object is determined. A least-squared estimate is made for an affine approximation to the object pose. In this step, image features consistent with the approximated pose are identified and outliers are discarded. Probability that the cluster of features is belonging to certain object type is then computed.

□ End of chapter.

Chapter 4

Proposed Researches

Although model-based object recognition systems have recently demonstrated good performance in face recognition, they generally require a detailed 3D face model to be constructed and trained in supervised learning manner. When the same 3D model is used to model objects other than human face, the systems will fail. And before the system can represent certain object by a 3D model representation, it has to know which 3D model it should apply on that object. In order word, the system have to recognize at least the type of that object before using a generic 3D model to model it. Assume the system know which 3D model it should apply, in order to model every generic object, for each type of objects, the model-based recognition systems will have to obtain their models before they can recognize them. Since the type of objects in the world is tremendous, attempt to model each type of objects manually or train the systems in supervised learning manner is impossible. A possible way to get around with this limit is to teach a system to learn the 3D model structures of objects automatically, there have not been a versatile solution to this problem. Another way to get around with the problem is to adopt another type of approach, the view-based object recognition approach.

View-based object recognition system recently demonstrated good performance on a variety of problems too. Mikolajczyk and Schmid [11] recently evaluated a variety of approaches and identified the SIFT algorithm as being the most resistant to common image deformations. SIFT algorithm is being continuously extended by other researchers and the improved versions PCA-SIFT and GLOH have reported better performance.

4.1 Hybrid of bottom-up approach and top-down approach

The view-based object recognition systems designed by Nelson [17] and Lowe [14] are typically bottom-up approaches. Their recognitions are first carried out at the low level of machine vision, in which low-level features such as edges and local extrema are searched. The recognitions proceed to the mid level of machine vision, in which descriptors of features are created. And finally at the higher level of machine vision, descriptors extracted on the input image are matched with those in the database and matched descriptors are mapped to the possible object types stored in the database by mean of voting algorithm. Sometimes verification process is carried out to refine the vote.

We believe that the process that human recognizes objects is not a purely bottom-up approach. As we recognize an object, we may just observe for a few features on the object and then we will try to guess what the object is. A verification step may follow to pick up the best hypothesis. The major difference in this approach is that the recognition system searches for salient features and try to recognize the object based on its knowledge, memory or say, database, interlacedly. In this way, effort on trying to recognize all features is reduced. Moreover, having recognized an object, its pose and location, it should be easier to segment out that object from the image and make the recognition of other objects more easily and accurately.

Some attempts were made to approach the object recognition problem using a top-down perspective, where recognition is performed by determining if one of many known objects appear in the image. Those approaches would be a good reference to us.

4.2 Hierarchy of features

In the recognition system proposed by Lowe [14], an object is simply a group of features. When features vote for an object pose, all features have the same voting power. However, an object may have most of their features common to other objects and have just a few of them special. If all features have the same voting power, extracted special features may not have enough influence to bias the voting result to the correct object. Thus a weight should be assigned to the features. Instead of assigning a weight to each feature based on its probability of occurrence, we can group a group of features into an small object and assign weight to each small object. An object is composed of many small objects and each small object is composed of many features,

as shown in Figure 4.1. Grouping a set of features into a small object is reasonable because only one feature keypoint usually do not carry much information.

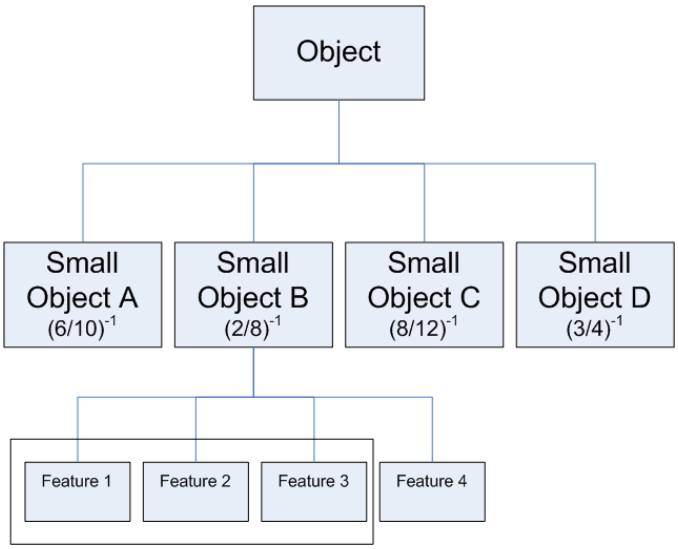

Figure 4.1: Each object is composed of many small objects and each small object is again composed of many features. Each small object is assigned a weight, say value 4 for object B, based on the probability of occurrence of the object B in the database. During features detection, not all features of a small object may be detected. As shown in the figure, there may just be 3 out of 4 features of object B found in the image. This statistic will affect the voting power of the small object.

Grouping of a set of features can be carried out by finding a cluster of features that appears in the image of an object that are geometrically close to each other. If a group of features is similar with any other groups in the database, its weight in voting will be lower. Refractoring process may be carried out when a new image of object is added to the database. This process is mainly to find the cluster of features, group them together and assign a weight to each of them. It also finds out the most distinctive group of features that an object contains. An image containing this distinctive group means that there exists that object in high chance.

Figure 4.2: Matching using color descriptors over color images done by Fergus *et al.*

4.3 Extension of SIFT Object Recognition

Lowe [14, 13] suggested several extensions to his work. Firstly, the features extracted by the SIFT object recognition system are from monochrome intensity images, so further distinctiveness could be derived from including illumination-invariant color descriptors and this has first been explored by Brown and Lowe [3], but detailed description on how to create a color descriptor is absent in that paper. Figure 4.2 shows a matching of invariant features between images using color descriptors. Secondly, local texture measures could be incorporated into feature descriptors. He believed that best results are likely to be obtained by matching many different types of features, which is capable to be implemented in invariant local feature approach. Thirdly, scale-invariant edge groupings can also be incorporated into feature descriptors such that local figure-ground discriminations would be done better at object boundaries. Finally, we could apply SIFT to generic object class recognition. Fergus *et al.* [7] has shown the potential of SIFT in recognizing generic classes of objects by unsupervised scale-invariant learning.

□ End of chapter.

Chapter 5

Conclusion and Future Work

In this paper, a survey of the recent researches on object recognition was given. Several recently proposed object recognition approaches are described and discussed. From the survey, we know that creating distinctive keypoint descriptors for object matching is a critical step to the whole object matching process because it is the invariance property of the intelligently made descriptors that make the object matching invariant to the scale, orientation, 3D affine distortion, addition of noise and change in illumination. SIFT features are undeniably a good kind of features for object matching. Yet there are still some rooms to further improve the object recognition approach using SIFT feature.

While view-based object recognition approach is generally a bottom-up approach, it is possible to incorporate top-down approach technique into it to make the object recognition faster and more accurate. Hierarchical weighting mechanism can also be applied during object matching to make the voting for object more reasonable. We will study the current extensions to SIFT: PCA-SIFT and GLOH, and try to improve them or extend SIFT in other aspects.

□ End of chapter.

Bibliography

[1] R. Basri and D. Jacobs. Recognition using region correspondences. *IJCV*, 25(2):145–166, November 1997.

[2] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *PAMI*, 24(4):509–522, April 2002.

[3] M. Brown and D. Lowe. Invariant features from interest point groups. In *BMVC02*, page Poster Session, 2002.

[4] T. Cootes, G. Edwards, and C. Taylor. Active appearance models. In *ECCV98*, page II: 484, 1998.

[5] T. Cootes, G. Edwards, and C. Taylor. Active appearance models. *PAMI*, 23(6):681–685, June 2001.

[6] L. di Stefano and S. Mattoccia. Fast template matching using bounded partial correlation. *MVA*, 13(4):213–221, 2003.

[7] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning, 2003.

[8] W. Grimson and T. Lozano-Perez. Localizing overlapping parts by searching the interpretation tree. *PAMI*, 9(4):469–482, July 1987.

[9] C. Harris and M. Stephens. A combined corner and edge detector. In *Alvey88*, pages 147–152, 1988.

[10] X. Hou, S. Li, H. Zhang, and Q. Cheng. Direct appearance models. In *CVPR01*, pages I:828–833, 2001.

[11] C. S. Krystian Mikolajczyk. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27:1615–1630, October 2005.

[12] W. Li and E. Salari. Successive elimination algorithm for motion estimation. ieee transactions on image processing, 4(1):105 – 107, january 1995.

[13] D. G. Lowe. Object recognition from local scale-invariant features. In *ICCV '99: Proceedings of the International Conference on Computer Vision-Volume 2*, page 1150, Washington, DC, USA, 1999. IEEE Computer Society.

[14] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, 2004.

[15] I. Matthews and S. Baker. Active appearance models revisited. *IJCV*, 60(2):135–164, November 2004.

[16] R. Nelson. 3-d recognition via 2-stage associative memory. In *Univ. of Rochester*, 1995.

[17] R. C. Nelson and A. Selinger. Large-scale tests of a keyed, appearance-based 3-d object recognition system. *Vision Research*, pages 2469–88, Aug. 1998.

[18] C. Schmid and R. Mohr. Local grayvalue invariants for image retrieval. *PAMI*, 19(5):530–535, May 1997.

[19] S. Smith and J. Brady. Susan: A new approach to low-level image-processing. *IJCV*, 23(1):45–78, May 1997.

[20] M. Swain and D. Ballard. Indexing via color histograms. In *DARPA90*, pages 623–630, 1990.