패치와 시멘틱 구조 특성을 이용한 비주얼 객체 인식 및 검색

Visual Object Class Recognition and Retrieval using Local Patch and Semantic Image Structure Features

2009 년 08 월 25 일

조선대학교 대학원

정보통신공학과

Ahmad Nishat

Visual Object Class Recognition and Retrieval using Local Patch and Semantic Image Structure Features

지도교수 박종안

이 논문을정보통신공학 박사학위신청 논문으로

제출함

2009 년 4 월

조선대학교 대학원

정보통신공학과

Ahmad Nishat

Ahmad Nishat 의 박사학위논문을 인준함

위원장 최태선 (인)

위 원 한승조 (인)

위 원 박승진 (인)

위 원 이진이 (인)

위 원 박종안 (인)

2009 년 06 월

조선대학교 대학원

Table of Contents

List of Figures

List of Tables

ABSTRACT

Visual Object Class Recognition and Retrieval using Local Patch and Semantic Image Structure Features

Nishat Ahmad
Adviser: Prof Park Jongan
College of Electronics and information engineering
Graduate School Chosun University

Content based image retrieval has emerged as an important field encompassing fields like image processing, computer vision and artificial intelligence. Near the turn of the 21st century researchers finally got convinced that next evolution of systems would need to understand the semantics of an image, not simply the low level underlying computational features i.e., “bridging the semantic gap”. The image retrieval systems need to be more intelligent, to be able to recognize generic objects and visual object classes at the least and also abstract meanings as feelings, in the far run. This can be stated as the dawn of second generation research in Image retrieval.

Recognition of a multitude of objects as dogs, cars etc. is an un-noticeable every day activity, hardly considered an achievement of any subtle order. In contrast, it is the ultimate scientific challenge of computer vision. After 40 years of research, robustly identifying the familiar

objects (chair, person, pet), scene categories (beach, forest, office), and activity patterns (conversation, dance, picnic) depicted in family pictures, news segments, or feature films is still far beyond the capabilities of today's vision systems [Preface: Towards Object level Categorization Eds. Ponce J., et. al., 2006].

Visual object class recognition has gradually evolved from structure based approaches to appearance based techniques and presently processes of the human vision are under immense focus. The thesis proposes a new approach to visual object class recognition with an aim to better understanding and exploration of the underlying principles of human vision. The thesis investigates the basic level of semantic structure formation in the human vision inferential processes which is hierarchically combined with other semantic structures to form meanings at an abstract level. This is a micro level approach compared to other approaches considering the whole image structure as a unit or geometric modeling approaches. Using this approach two sets of semantic features have been derived for visual object class recognition.

The algorithm uses the hypothesis in line with Gestalt laws of proximity that; in an image, basic semantic structures are formed by line segments (arcs also approximated and broken into smaller line segments based on pixel deviation threshold) which are in close proximity of each other. Based on the notion of proximity a transitive relation is defined, which combines basic micro level semantic structures hierarchically till such a point where a semantic meaning of the structure can be extracted. The algorithm extracts line segments in an image and then forms semantic groups of these line segments based on a minimum distance threshold from each other. The line segment groups so formed can be differentiated from each other, by the number of group members and their geometrical properties. The geometrical properties of these semantic groups are used to generate rotation, translation and scale invariant histograms used as feature vectors for object class recognition tasks in a K-nearest neighbor framework.

In the second approach a semantic group based on the proximity distance is clustered and modeled as a graph vertex. The line segments which are common to more than one semantic group are defined as semantic relations between the semantic groups and are modeled as edges of the graph. This way an image object is transformed into a graph using micro level structure formations. Each vertex and edge is labeled using translation, rotation and scale invariant properties of the member segments of each vertex and edge. From a set of training images, a graph model is constructed for visual object class recognition. The graph model is constructed by iteratively combining the training graphs and frequency labeling the vertices and edges. After the combining phase, all the vertices and edges whose repetition frequency is below a threshold are removed. The final graph model consists of the semantic nodes which are highly common in the training images. The recognition is based on graph matching the query image graph and the model graph. The model graph generates a vote for the query and ties are resolved by considering the node frequencies in the query and model graph.

The algorithm has been applied to classify 101 object classes at one time. The results have been compared with existing state of the art approaches and are found promising. Results from above approaches show that low level image structure and other features can be used to construct different type of semantic features, which can help a model or a classifier make more intelligent decisions and work more effectively for the task compared to low level features alone. Our experimental results are comparable, or outperform other state-of-the-art approaches. We have also summarized the state-of-the-art at the time this work was finished. We conclude with a discussion about the possible future extensions.

I. Introduction

A. Overview

Content-based image retrieval, a technique which employs visual contents to search images from large scale image databases according to users' interests, has been an active and fast advancing research area since the 1990s. During the past decade, remarkable progress has been made in both theoretical research and system development. However, there remain many challenging research problems that continue to attract researchers from multiple disciplines. We briefly review in this chapter the world of content based image retrieval (CBIR) till to date by following developments in the field with respect to time line and breakthroughs. In the process we highlight the fundamental theories for CBIR and look at the development of CBIR techniques. Then, as the most important part of this chapter, we introduce some latest trends coupled with ongoing and future directions in the present day visual content descriptions.

B. First generation of Content Based Image retrieval

Early work on image retrieval can be traced back to the late 1970s. In 1979, a conference on Database Techniques for Pictorial Applications [1] was held in Florence. Since then, the application potential of image database management techniques has attracted the attention of researchers [2 – 5]. Early techniques were not generally based on visual features but on the

textual annotation of images. In other words, images were first annotated with text and then searched using a text-based approach from traditional database management systems. Comprehensive surveys of early text-based image retrieval methods can be found in [6, 7]. Text-based image retrieval uses traditional database techniques to manage images. Through text descriptions, images can be organized by topical or semantic hierarchies to facilitate easy navigation and browsing based on standard Boolean queries. However, since automatically generating descriptive texts for a wide spectrum of images is not feasible, most text-based image retrieval systems require manual annotation of images. Obviously, annotating images manually, is a cumbersome and expensive task for large image databases, and is often subjective, context-sensitive and incomplete. As a result, it is difficult for the traditional text-based methods to support a variety of task-dependent queries.

In the early 1990s, as a result of advances in the internet and new digital image sensor technologies, the volume of digital images produced by scientific, educational, medical, industrial, and other applications available to users increased dramatically. The difficulties faced by text-based retrieval became more and more severe. The efficient management of the rapidly expanding visual information became an urgent problem. This need formed the driving force behind the emergence of content-based image retrieval techniques. In 1992, the National Science Foundation of the United States organized a workshop on visual information management systems [8] to identify new directions in image database management systems. It was widely recognized that a more efficient and intuitive way to represent and index visual information would be based on properties that are inherent in the images themselves.

Much early research, exemplified by projects such as TRADEMARK [9], QBIC [10] and Photobook [11], established the feasibility of retrieving images from large collections using automatically-derived features. Researchers from the communities of computer vision,

database management, human-computer interface, and information retrieval were attracted to this field. Since then, research on content-based image retrieval has developed rapidly [12 – 17]. Since 1997, the number of research publications on the techniques of visual information extraction, organization, indexing, user query and interaction, and database management has increased enormously. Similarly, a large number of academic and commercial retrieval systems have been developed by universities, government organizations, companies, and hospitals. Comprehensive surveys of these techniques and systems can be found in [18 –20].

Content-based image retrieval uses the visual contents of an image such as color, shape, texture, and spatial layout to represent and index the image. In typical content-based image retrieval systems (Figure 1-1), the visual contents of the images in the database are extracted and described by multi-dimensional feature vectors. The feature vectors of the images in the database form a feature database. To retrieve images, users provide the retrieval system with example images or sketched figures. The system then changes these examples into its internal representation of feature vectors. The similarity distances between the feature vectors of the query example or sketch and those of the images in the database are then calculated and retrieval is performed with the aid of an indexing scheme. The indexing scheme provides an efficient way to search for the image database. Recent retrieval systems have incorporated user's relevance feedback to modify the retrieval process in order to generate perceptually and semantically more meaningful retrieval results. In the next section, we discuss fundamental techniques for content-based image retrieval.

C. Image Content Descriptors

Generally speaking, image content may include both visual and semantic content. Visual content can be very general or domain specific. General visual content include color, texture,

shape, spatial relationship, etc. Domain specific visual content, like human faces, is application dependent and may involve domain knowledge.

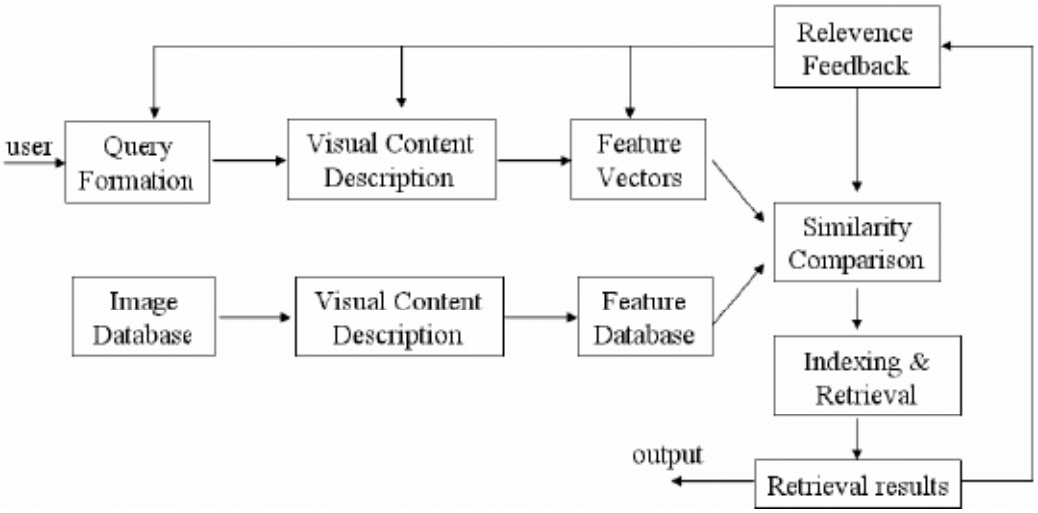


Figure 1.1 A typical content-based image retrieval system

This section concentrates on general visual contents descriptions. Later sections discuss domain specific and semantic contents. A good visual content descriptor should be invariant to the accidental variance introduced by the imaging process (e.g., the variation of the illuminant of the scene). However, there is a tradeoff between the invariance and the discriminative power of visual features, since a very wide class of invariance loses the ability to discriminate between essential differences. Invariant description has been largely investigated in computer vision (like object recognition), but is relatively new in image retrieval [21]. A visual content descriptor can be either global or local. A global descriptor uses the visual features of the whole image, whereas a local descriptor uses the visual features of regions or objects to describe the image content. To obtain the local visual descriptors, an image is often divided into parts first. The simplest way of dividing an image is to use a partition, which cuts the image into tiles of equal size and shape. A simple partition does not generate perceptually

meaningful regions but is a way of representing the global features of the image at a finer resolution. A better method is to divide the image into homogenous regions according to some criterion using region segmentation algorithms that have been extensively investigated in computer vision. A more complex way of dividing an image, is to undertake a complete object segmentation to obtain semantically meaningful objects (like ball, car, horse). Currently, automatic object segmentation for broad domains of general images is unlikely to succeed. In the following, we will discuss some widely used techniques for extracting color, texture, shape and spatial relationship from images.

1. Color

Color is the most extensively used visual content for image retrieval [22 – 31]. Its three-dimensional values make its discrimination potentiality superior to the single dimensional gray values of images. Before selecting an appropriate color description, color space must be determined first. Each pixel of the image can be represented as a point in a 3D color space. Commonly used color spaces for image retrieval include RGB, Munsell, CIE L*a*b*, CIE L*u*v*, HSV (or HSL, HSB), and opponent color space. There is no agreement on which is the best. However, one of the desirable characteristics of an appropriate color space for image retrieval is its uniformity [27]. Uniformity means that two color pairs that are equal in similarity distance in a color space are perceived as equal by viewers. In other words, the measured proximity among the colors must be directly related to the psychological similarity among them.

RGB space is a widely used color space for image display. It is composed of three color components red, green, and blue. These components are called "additive primaries" since a color in RGB space is produced by adding them together. In contrast, CMY space is a color

space primarily used for printing. The three color components are cyan, magenta, and yellow. These three components are called "subtractive primaries" since a color in CMY space is produced through light absorption. Both RGB and CMY space are device-dependent and perceptually non-uniform.

The CIE L*a*b* and CIE L*u*v* spaces are device independent and considered to be perceptually uniform. They consist of a luminance or lightness component (L) and two chromatic components a and b or u and v. CIE L*a*b* is designed to deal with subtractive colorant mixtures, while CIE L*u*v* is designed to deal with additive colorant mixtures. The transformation of RGB space to CIE L*u*v* or CIE L*a*b* space can be found in [26].

HSV (or HSL, or HSB) space is widely used in computer graphics and is a more intuitive way of describing color. The three color components are hue, saturation (lightness) and value (brightness). The hue is invariant to the changes in illumination and camera direction and hence more suited to object retrieval. RGB coordinates can be easily translated to the HSV (or HLS, or HSB) coordinates by a simple formula [22].

The opponent color space uses the opponent color axes (R-G, 2B-R-G, R+G+B). This representation has the advantage of isolating the brightness information on the third axis. With this solution, the first two chromaticity axes, which are invariant to the changes in illumination intensity and shadows, can be down-sampled since humans are more sensitive to brightness than they are to chromatic information.

In the following paragraphs, we will introduce some commonly used color descriptors: the color histogram, color coherence vector, color correlogram, and color moments. Color moments have been successfully used in many retrieval systems (like QBIC [32, 33]), especially when the image contains just the object. The first order (mean), the second (variance) and the third order (skewness) color moments have been proved to be efficient and effective in

representing color distributions of images [29]. Usually the color moment performs better if it is defined by both the L*u*v* and L*a*b* color spaces as opposed to solely by the HSV space. Using the additional third-order moment improves the overall retrieval performance compared to using only the first and second order moments. However, this third-order moment sometimes makes the feature representation more sensitive to scene changes and thus may decrease the performance. Since only 9 (three moments for each of the three color components) numbers are used to represent the color content of each image, color moments are a very compact representation compared to other color features. Due to this compactness, it may also lower the discrimination power. Usually, color moments can be used as the first pass to narrow down the search space before other sophisticated color features are used for retrieval.

The color histogram serves as an effective representation of the color content of an image if the color pattern is unique compared with the rest of the data set. The color histogram is easy to compute and effective in characterizing both the global and local distribution of colors in an image. In addition, it is robust to translation and rotation about the view axis and changes only slowly with the scale, occlusion and viewing angle. Since any pixel in the image can be described by three components in a certain color space (for instance, red, green, and blue components in RGB space, or hue, saturation, and value in HSV space), a histogram, i.e., the distribution of the number of pixels for each quantized bin, can be defined for each component. Clearly, the more bins a color histogram contains, the more discrimination power it has. However, a histogram with a large number of bins will not only increase the computational cost, but will also be inappropriate for building efficient indexes for image databases. Furthermore, a very fine bin quantization does not necessarily improve the retrieval performance in many applications. One way to reduce the number of bins is to use the opponent color space which enables the brightness of the histogram to be down sampled. Another way is to use clustering

methods to determine the K best colors in a given space for a given set of images. Each of these best colors will be taken as a histogram bin. Since that clustering process takes the color distribution of images over the entire database into consideration, the likelihood of histogram bins in which no or very few pixels fall will be minimized. Another option is to use the bins that have the largest pixel numbers since a small number of histogram bins capture the majority of pixels of an image [15] Such a reduction does not degrade the performance of histogram matching, but may even enhance it since small histogram bins are likely to be noisy.

When an image database contains a large number of images, histogram comparison will saturate the discrimination. To solve this problem, the joint histogram technique is introduced [28]. In addition, color histogram does not take the spatial information of pixels into consideration, thus very different images can have similar color distributions. This problem becomes especially acute for large scale databases. To increase discrimination power, several improvements have been proposed to incorporate spatial information. A simple approach is to divide an image into sub-areas and calculate a histogram for each of those sub-areas. As introduced above, the division can be as simple as a rectangular partition, or as complex as a region or even object segmentation. Increasing the number of sub-areas increases the information about location, but also increases the memory and computational time.

In [34] a different way of incorporating spatial information into the color histogram, color coherence vectors (CCV), was proposed. Each histogram bin is partitioned into two types, i.e., coherent, if it belongs to a large uniformly-colored region, or incoherent, if it does not. Let α_i denote the number of coherent pixels in the ith color bin and β_i denote the number of incoherent pixels in an image. Then, the CCV of the image is defined as the vector $<(\alpha_1, \beta_1), (\alpha_2, \beta_2), \ldots, (\alpha_N, \beta_N)>$. Note that $<\alpha_1+\beta_1, \alpha_2+\beta_2 \ldots \alpha_N + \beta_N>$ is the color histogram of the image. Due to its additional spatial information, it has been shown that CCV provides better retrieval

results than the color histogram, especially for those images which have either mostly uniform color or mostly texture regions. In addition, for both the color histogram and color coherence vector representation, the HSV color space provides better results than CIE L*u*v* and CIE L*a*b* space.

The color correlogram [24] was proposed to characterize not only the color distributions of pixels, but also the spatial correlation of pairs of colors. If we consider all the possible combinations of color pairs the size of the color correlogram will be very large ($O(N^2d)$), therefore a simplified version of the feature called the color autocorrelogram is often used instead. The color autocorrelogram only captures the spatial correlation between identical colors and thus reduces the dimension to O(Nd). Compared to the color histogram and CCV, the color autocorrelogram provides the best retrieval results, but is also the most computational expensive due to its high dimensionality.

Color not only reflects the material of surface, but also varies considerably with the change of illumination, the orientation of the surface, and the viewing geometry of the camera. This variability must be taken into account. However, invariance to these environmental factors is not considered in most of the color features discussed above. Invariant color representation has been introduced to content-based image retrieval recently. In [35], a set of color invariants for object retrieval was derived based on the Schafer model of object reflection. In [36], specular reflection, shape and illumination invariant representation based on blue ratio vector (r/b, g/b, 1) is given. In [37], a surface geometry invariant color feature is provided. These invariant color features, when applied to image retrieval, may yield illumination, scene geometry and viewing geometry independent representation of color contents of images, but may also lead to some loss in discrimination power among images.

2. Texture

Texture is another important property of images. Various texture representations have been investigated in pattern recognition and computer vision. Basically, texture representation methods can be classified into two categories: structural and statistical. Structural methods, including morphological operator and adjacency graph, describe texture by identifying structural primitives and their placement rules. They tend to be most effective when applied to textures that are very regular. Statistical methods, including Fourier power spectra, co-occurrence matrices, shift-invariant principal component analysis (SPCA), Tamura feature, Wold decomposition, Markov random field, fractal model, and multi-resolution filtering techniques such as Gabor and wavelet transform, characterize texture by the statistical distribution of the image intensity. In this section, we discuss a number of texture representations [38 – 56], which have been used frequently and have proved to be effective in content-based image retrieval systems.

The Tamura features [55], including coarseness, contrast, directionality, line likeness, regularity, and roughness, are designed in accordance with psychological studies on the human perception of texture. The first three components of Tamura features have been used in some early well-known image retrieval systems, such as QBIC [32] and Photobook [57].

Wold decomposition [41, 48] provides another approach to describing textures in terms of perceptual properties. The three Wold components, harmonic, evanescent, and in-deterministic, correspond to periodicity, directionality, and randomness of texture respectively. Periodic textures have a strong harmonic component, highly directional textures have a strong evanescent component, and less structured textures tend to have a stronger in-deterministic component.

The SAR model is an instance of Markov random field (MRF) models, which have been very successful in texture modeling in the past decades. Compared with other MRF models, SAR uses fewer parameters. In the SAR model, pixel intensities are taken as random variables. The SAR model is not rotation invariant. To derive a rotation-invariant SAR model (RISAR), pixels lying on circles of different radii centered at each pixel (x, y) serve as its neighbor set. To describe textures of different granularities, the multi-resolution simultaneous auto-regressive model (MRSAR) [52] has been proposed to enable multi-scale texture analysis. An image is represented by a multi-resolution Gaussian pyramid with low-pass filtering and sub-sampling applied at several successive levels. Either the SAR or RISAR model may then be applied to each level of the pyramid. MRSAR has been proved [63, 75]to have better performance on the Brodatz texture database [38] than many other texture features, such as principal component analysis, Wold decomposition, and wavelet transform.

The Gabor filter has been widely used to extract image features, especially texture features [44, 58]. It is optimal in terms of minimizing the joint uncertainty in space and frequency, and is often used as an orientation and scale tunable edge and line (bar) detector. There have been many approaches proposed to characterize textures of images based on Gabor filters.

Similar to the Gabor filtering, the wavelet transform [40, 50] provides a multi-resolution approach to texture analysis and classification [39, 47]. Two major types of wavelet transforms used for texture analysis are the pyramid-structured wavelet transform (PWT) and the tree-structured wavelet transform (TWT). According to the comparison of different wavelet transform features [49], the particular choice of wavelet filter is not critical for texture analysis.

3. Shape

Shape features of objects or regions have been used in many content-based image retrieval systems [59 – 62]. Compared with color and texture features, shape features are usually described after images have been segmented into regions or objects. The state-of-art methods for shape description can be categorized into either boundary-based (rectilinear shapes [61], polygonal approximation [63], finite element models [64], and Fourier-based shape descriptors [65, 66, 67]) or region-based methods (statistical moments [68, 69]). A good shape representation feature for an object should be invariant to translation, rotation and scaling. In this section, we briefly describe some of these shape features that have been commonly used in image retrieval applications. For a concise comprehensive introductory overview of the shape matching techniques, see [70].

Classical shape representation uses a set of moment invariants. The central moment can be normalized to be scale invariant [44]. Based on the central moments, a set of moment invariants to translation, rotation, and scale can be derived [68, 69].

The contour of a 2D object can be represented as a closed sequence of successive boundary pixels for which a turning function or turning angle can be defined, which measures the angle of the counterclockwise tangents as a function of the arc-length according to a reference point on the object's contour. One major problem with this representation is that it is variant to the rotation of object and the choice of the reference point. Therefore, to compare the shape similarity between objects A and B with their turning functions, the minimum distance needs to be calculated over all possible shifts and rotations.

Fourier descriptors describe the shape of an object with the Fourier transform of its boundary. Again we consider the contour of a 2D object as a closed sequence of successive boundary pixels. Three types of contour representations, i.e., curvature, centroid distance, and complex

coordinate function, can be defined. The Fourier transforms of these three types of contour representations generate three sets of complex coefficients, representing the shape of an object in the frequency domain. Lower frequency coefficients describe the general shape property, while higher frequency coefficients reflect shape details. To achieve rotation invariance (i.e., contour encoding is irrelevant to the choice of the reference point), only the amplitudes of the complex coefficients are used and the phase components are discarded. To achieve scale invariance, the amplitudes of the coefficients are divided by the amplitude of DC component or the first non-zero coefficient. The translation invariance is obtained directly from the contour representation.

We can use the shape Circularity, Eccentricity, and Major Axis Orientation. Circularity is computed as:

$$\propto = \frac{4\pi S}{P^2} \qquad 1\text{-}1$$

where S is the size and P is the perimeter of an object. This value ranges between 0 (corresponding to a perfect line segment) and 1 (corresponding to a perfect circle). The major axis orientation can be defined as the direction of the largest eigenvector of the second order covariance matrix of a region or an object. The eccentricity can be defined as the ratio of the smallest eigen value to the largest eigen value.

Regions or objects with similar color and texture properties can be easily distinguished by imposing spatial constraints. For instance, regions of blue sky and ocean may have similar color histograms, but their spatial locations in images are different. Therefore, the spatial location of regions (or objects) or the spatial relationship between multiple regions (or objects) in an image is very useful for searching images. The most widely used representation of spatial relationship is the 2D strings proposed by Chang et al [71]. It is constructed by projecting

images along the x and y directions. Two sets of symbols, V and A, are defined on the projection. Each symbol in V represents an object in the image. Each symbol in A represents a type of spatial relationship between objects. As its variant, the 2D G-string [72] cuts all the objects along their minimum bounding box and extends the spatial relationships into two sets of spatial operators. One defines local spatial relationships. The other defines the global spatial relationships, indicating that the projection of two objects are disjoin, adjoin or located at the same position. In addition, 2D C-string [73] is proposed to minimize the number of cutting objects. 2D-B string [74] represents an object by two symbols, standing for the beginning and ending boundary of the object.

D. Second generation of Content Based Image retrieval

This section reviews recent changes in trends in the field and outlines and future directions and argues that further advances in the field are likely to involve the increasing use of techniques from the field of artificial intelligence.

Most current CBIR techniques are geared towards retrieval by some aspect of image appearance, depending on the automatic extraction and comparison of image features judged most likely to convey that appearance. The features most often used include color, texture, shape, spatial layout, and multi-resolution pixel intensity transformations such as wavelets or multi-scale Gaussian filtering. At least three CBIR packages making use of such techniques were made commercially available: QBIC from IBM, the VIR Image Engine from Virage, Inc, and VisualRetrievalWare from Excalibur, Inc. While the technology behind current CBIR systems is undoubtedly impressive, user take-up of such systems has so far been minimal. This is not because the need for such systems is lacking-there is ample evidence of user demand for better image data management in fields as diverse as crime prevention, photo-journalism,

fashion design, trademark registration, and medical diagnosis. It is because there is a mismatch between the capabilities of the technology and the needs of users. The vast majority of users do not want to retrieve images simply on the basis of similarity of appearance. They need to be able to locate pictures of a particular type (or individual instance) of object, phenomenon, or event [77].

Gudivada and Raghavan [75] have drawn a useful distinction between retrieval by primitive image feature (such as color, texture or shape) and semantic feature (such as the type of object or event depicted by the image). Eakins [76] has taken this distinction further, identifying three distinct levels of image query, each of which can be further subdivided:

(1) Level 1, retrieval by primitive features such as color, texture, shape or the spatial location of image elements (e.g. find all pictures containing yellow or blue stars arranged in a ring).

(2) Level 2, retrieval by derived attribute or logical feature, involving some degree of inference about the identity of the objects depicted in the image (e.g. find pictures of a passenger train crossing a bridge).

(3) Level 3, retrieval by abstract attribute, involving complex reasoning about the significance of the objects or scenes depicted (e.g. find pictures illustrating pageantry).

Using this framework, the extent of the mismatch between user requirements and the capabilities of the technology becomes clear. Although the volume of research into user needs is not large, the results of those studies which have been conducted to date (e.g. [77]) suggest strongly that very few users need level-1 retrieval. The majority of image queries received by picture libraries are at level 2, though a significant number (particularly in specialist art libraries) are at level 3. The overwhelming majority of CBIR systems, both commercial and

experimental; offer nothing but level 1 retrieval. A few experimental systems now operate at level 2, but none at all at level 3. What are the prospects of bridging what has been referred to as the semantic gap [75], and delivering the image retrieval capabilities that users really want? This section aims to answer this question by reviewing current research into semantic image retrieval, with particular emphasis on the contribution which techniques from related fields such as artificial intelligence (AI) are making to developments in this area. CBIR may have its roots in the field of classical image analysis; it relies on many standard image analysis techniques, such as convolution, edge detection, pixel intensity histogramming, and power spectrum analysis. But a successful solution to the problems of semantic image retrieval (if one exists at all) may well require a significant paradigm shift, involving techniques originally developed in other fields. CBIR has already benefited greatly from insights derived from related fields. A prime example of this process is the technique of relevance feedback, originally developed for text retrieval, where users indicate the relevance of each item of output received, and the system amends its search strategy accordingly. Relevance feedback is showing considerable promise in the image retrieval area, largely because users can rapidly judge the relevance of a retrieved image within seconds.. Other examples where CBIR has benefited from insights from related fields include relatively efficient direct access via multidimensional indexing, from the database management field, and retrieval by subjective appearance, drawing on Gestalt psychology.

1. Intelligent image retrieval

One crucial difference between primitive and semantic- level retrieval seems to lie in the extent of intelligent behavior needed to decide whether a given image meets the specified search criteria. At the primitive level, images can normally be matched by algorithmic means purely

on the basis of information contained within the images themselves. For example, color similarity matching requires nothing more than the computation and comparison of two histograms representing the distribution of pixel colors across the two images. There is no requirement for what might be considered intelligent behavior in reference to an external knowledge base, reasoning with conflicting or incomplete data, or learning from past experience.

Semantic retrieval requires the identification of images depicting desired types of object, scene, event, or abstract idea. According to the definition above, this is a process requiring intelligence, as it requires reasoning about the nature and significance of primitive visual cues from the image, and their relationships to each other and to the viewer's past experience. This latter aspect appears to be of crucial importance. Even at the simplest level (such as recognizing a curved yellow region in an image as a banana), extraction of an image's semantic content seems to require reference to some external store of knowledge. To identify a banana in an image requires experience of the range of color, shape and texture combinations which have characterized previously-encountered examples, and the ability to use this knowledge to predict which yellow curved regions are in fact bananas, and which (say)parts of yellow rubber rings.

Identifying even a relatively simple artifact such as a chair is a rather more complex process. Since chairs come in a wide variety of colors, textures and shapes, primitive image features are unlikely to suffice on their own. The problem of recognizing a chair is not perceptually more difficult than that of recognizing a banana. The difference lies in the degree of interpretation necessary. Recognition of an object as a chair requires reference to some higher-level model, defining spatial, structural and perhaps other constraints. Such a model needs to be susceptible to modification, to include the possibility that new designs of chair may appear in the future (not a problem one would expect to encounter with bananas!). Humans build up and refine such

a model automatically from past experience: for machines, the process is less straightforward. The need to gain such experience directly is one reason why Brooks [78] has advocated designing robots in humanoid form.

Identifying complex human artifacts is still more problematic. Experienced engineers can readily recognize a pressure-limiting valve in an engineering drawing, even though its actual shape may vary considerably, presumably because their training enables them to draw reasonable inferences from the appearance and layout of key components, as well as the nature of any larger structures in which they appear. But even a highly intelligent human would find such a task impossible without the requisite engineering training. The need to update one's mental model of a specialist device of this kind is likely to be even greater than for an everyday object such as a chair, since new designs are likely to appear at frequent intervals.

Yet another layer of complexity is encountered when trying to interpret scenes depicting specific types of event. To recognize a photograph as that of a child's birthday party demands not only the identification of objects which might be present in such scenes (young human figures, balloons, lighted candles), but a further level of reasoning about the relationship of these objects to each other and the extent to which these conform to prior expectations of what occurs at such events. Again, the ability to update such mental models in the light of changing circumstances is crucial.

The issues surrounding human recognition and classification of images have been extensively studied by Rosch et al. [79]. The most significant findings from these studies in the present context are as follows:

Humans naturally categorize objects they encounter into basic categories such as chair or banana. Although visual appearance is of major importance in identifying these classes, other

factors such as commonality of the motor movements needed to interact with such items (such as grasping with the fingers) also play a part in such characterization.

The basic category appears to be a favored level of abstraction for many purposes. Participants in experiments in free-naming of pictures, for example, overwhelmingly preferred to use basic category names rather than more specialized or generalized levels (hammer rather than tool or claw-hammer, for example). Developmental studies with young children show that basic category names are learnt earlier in life than those of other levels. Basic categories generally have a higher proportion of attributes common to all member of that class than subordinate or super ordinate categories. In many (but not all) cases it is possible to construct an averaged shape from typical members of the class which humans can readily recognize.

These findings give some indication of the likely success of semantic image retrieval techniques which rely on automatic derivation of object or scene labels from visual features of the image. Such techniques are most likely to succeed for objects within an image which correspond to basic classes (such as banana or horse) whose members share a strong visual similarity. For such objects it should be possible to construct or learn suitable object models permitting recognition of typical examples of each class. For other types of object (such as bird or tree), a similar approach based on visual similarity of subclasses (probably, though not necessarily, based on existing taxonomic divisions such as sparrow, parrot or eagle) may prove more effective. For object classes where many defining attributes are non-visual (such as chair or pump), however, this approach appears doomed to failure, though the fact that humans can recognize such objects from visual cues alone suggests that the problem is in principle soluble.

To develop a complete understanding of image contents at the semantic level is a formidable task, well beyond the capabilities of any current machine. Fortunately, such a complete level of understanding is not an essential prerequisite for successful semantic image retrieval, as several

researchers in the field have pointed out [80, 81]. Empirically, a retrieval system can be regarded as successful if it has the ability to classify a sufficiently high proportion of objects sought by users accurately enough for its retrieval output to satisfy a searcher's needs. In many contexts (including photo-journalism) this means that quite low classification accuracy may be acceptable, provided the searcher can in fact find a usable picture. An analogous situation holds in text retrieval, where effective retrieval systems have been around for years, despite continuing difficulties with automatic text understanding. Unfortunately it is not yet clear what level of image understanding is in fact required for successful classification and retrieval. The only way to resolve this question appears to lie in the development and evaluation of semantic image retrieval techniques.

2. Semantic image retrieval - current trends

Research into semantic image retrieval per se has a relatively short history; the vast majority of papers mentioned above date from 1996 or later. Many of the techniques now being applied to the problem have been adapted from related areas such as ‘classical' object recognition or machine learning, and it is not always easy to distinguish between research into image understanding for its own sake and research motivated by a desire to develop better storage and retrieval systems. As yet, it is difficult to discern any body of techniques or hypotheses which belong solely to the field of semantic image retrieval. This is possibly an indication of the relative immaturity of the field. However, semantic image retrieval is a topic of growing research interest, at least at level 2 as discussed above (retrieval by derived attributes such as the type of object or scene depicted). Several different areas of activity can be distinguished within the field, though many of the techniques used are common to more than one area, and the distinctions between different approaches are not always clear-cut.

By contrast, no significant research has yet been reported into CBIR at level 3 (retrieval by abstract attribute such as freedom). The issues involved are dauntingly complex. Little is known about the way in which humans interact with images at this level, making it almost impossible even to identify potentially fruitful lines of investigation.

II. Local patch based approach for Image retrieval

The chapter covers feature vector construction methodology developed using local image patches and their performance in image retrieval. As the geometric shapes and corners form a major paradigm in the evaluations and identification of graphical information by brain (human perception) [82]. The patches are extracted from around so called corner points in an image. The algorithm uses information sampled from detected corner points in the image. A corner detection approach based on line intersections has been employed using Hough transform for line detection and then finding intersecting, near intersecting or complex shaped corners. As the affine transformations preserve the co-linearity of points on a line and their intersection properties, the corner points obtained as such retain the much desired property of repeatability and hence ensure the similar pixel samples under various transformations and are robust to noise. K-means unsupervised learning approach is used to assign class labels to the corner patches by learning a Gaussian Byes classifier to classify whole training image dataset. Histogram of the class members in an image has been used as a feature vector. The retrieval performance and behavior of the algorithm has been tested using four histogram similarity measures to check the strengths and weaknesses of the approach.

A. Literature in perspective

There is an abundance of literature on corner detection. Moravec [83] observed that the difference between the adjacent pixels of an edge or a uniform part of the image is small, but at the corner, the difference is significantly high in all directions. Harris [84] implemented a technique referred to as the Plessey algorithm. The technique was an improvement of the Moravec algorithm. Beaudet [85] proposed a determinant (DET) operator which has significant values only near corners. Kitchen and Rosenfeld [86] presented a few corner detection methods. The work included methods based on gradient magnitude and gradient direction, change of direction along edge, angle between most similar neighbors, and turning of the fitted surface. Lai and Wu [87] considered edge-corner detection for defective images. Tsai [88] proposed a method for boundary-based corner detection using neural networks. Ji and Haralick [89] presented a technique for corner detection with covariance propagation. Lee and Bien [90] applied fuzzy logic to corner detection. Fang and Huang [91] proposed a method which was an improvement on the gradient magnitude of the gradient-angle method by Kitchen and Rosenfeld [86]. Chen and Rockett utilized Bayesian labeling of corners using a gray-level corner image model [92]. Wu and Rosenfeld [93] proposed a technique which examines the slope discontinuities of the x and y projections of an image to find the possible corner candidates. Paler et al. [94] proposed a technique based on features extracted from the local distribution of gray-level values. Rangarajan et al. [95] proposed a detector which tries to find an analytical expression for an optimal function whose convolution with the windows of an image has significant values at corner points. Arrebola et al. [96] introduced corner detection by local histograms of contour chain code. Shilat et al. [97] worked on ridge's corner detection and correspondence. Nassif et al. [98] considered corner location measurement. Sohn et al. [99] proposed a mean field annealing approach to corner detection. Zhang and Zhao [100]

considered a parallel algorithm for detecting dominant points on multiple digital curves. Kohlmann [101] applied the 2D Hilbert transform to corner detection. Mehrotra et al. [102] proposed two algorithms for edge and corner detection. The first is based on the first-directional derivative of the Gaussian, and the second is based on the second-directional derivative of the Gaussian. Zuniga and Haralick [103] utilized the facet model for corner detection. Smith and Brady [104] used a circular mask for corner detection. No derivatives were used. Orange and Groen [105] proposed a model-based corner detector. Other corner detectors have been proposed in [106 – 109]. Mokhtarian [110] used the curvature-scale-space (CSS) [111, 112] technique to search the corner points. The CSS technique is adopted by MPEG-7.

The Hough transform [113] later introduced in generalized form for lines and curve detection [114] has been focus of research interest after it was popularized by the journal article of D.H. Ballard [115]. Davies [116] applied the generalized Hough transform to corner detection. Diou, A. et al. [117] proposed an analytical approach for the calculation of the theoretical Hough transform on standard images for research of straight lines. Anastasios & Nikos [118] proposed the Inverse Hough Transform. Fei Shen & Han Wang [119] used modified Hough transform for corner detection. Yu-Hua Gu [120] presented corner based feature extraction for object retrieval using smoothed object boundary curve and 2D rotationally symmetric band pass filter for detecting sharp angles (corners) and used the corner information for object matching and retrieval. For object matching they used normalized arc-lengths between adjacent corners, corner to centroid distances and object boundary curves modeled by a constrained active B-spline curve model.

Corner or Interest point detection has a long tradition in classic computer vision for finding point correspondences to reconstruct 3D scenes from 2D views. There exist a lot of evaluation

papers that try to judge the quality of interest point detectors, e.g. [121 ~ 124]. The evaluation criteria are mainly repeatability (i.e. robustness against varying imaging conditions like viewpoint, scale, illumination changes) and information content. After detecting the location of interest, a group of pixels extracted from around the detected interest points is used to construct a descriptor which can be used for point correspondence, object class recognition, image retrieval etc. Various descriptors have been proposed in the past and are also called as local features or local descriptors. A recent performance evaluation of Local Descriptors has been carried out by Mikolaiczyk et al. [125].

B. Corner Definition

In the present day the term corner and interest point is being used interchangeably. A corner can be defined as the intersection of two edges or a point for which there are two dominant and different edge directions in a local neighborhood of the point. The characteristic feature of such a point is mentioned by Moravec [83] that the difference between the adjacent pixels of an edge or a uniform part of the image is small, but at the corner (edge intersection); the difference is significantly high in all directions. Where as, an interest point is a point in an image, which has a well defined position and can be robustly detected. This means that an interest point can be a corner but it can also be, for example, an isolated point of local intensity maximum or minimum, line endings, or a point on a curve where the curvature is locally maximal. In practice, most so-called corner detection methods detect interest points in general rather than corners in particular. Closely related to these are blob and ridge detectors. The exact meaning of even “interest point” differs from author to author. Agarwal et al. [126] defines them to be “points that have high information content in terms of the local change in signal.”, Cordelia

Schmid et al. [127] as “points where a signal changes two dimensionally” or Loupias et al. [128] just as “points where something happens in the signal at any resolution”.

For the purpose of our algorithm, we defined the corner as an intersection of two or more intersecting or near intersecting straight lines (edges qualified to be straight lines). The reason for using the line intersections for defining corners is that lines are invariant to various affine transformations shown in figure 2.1.

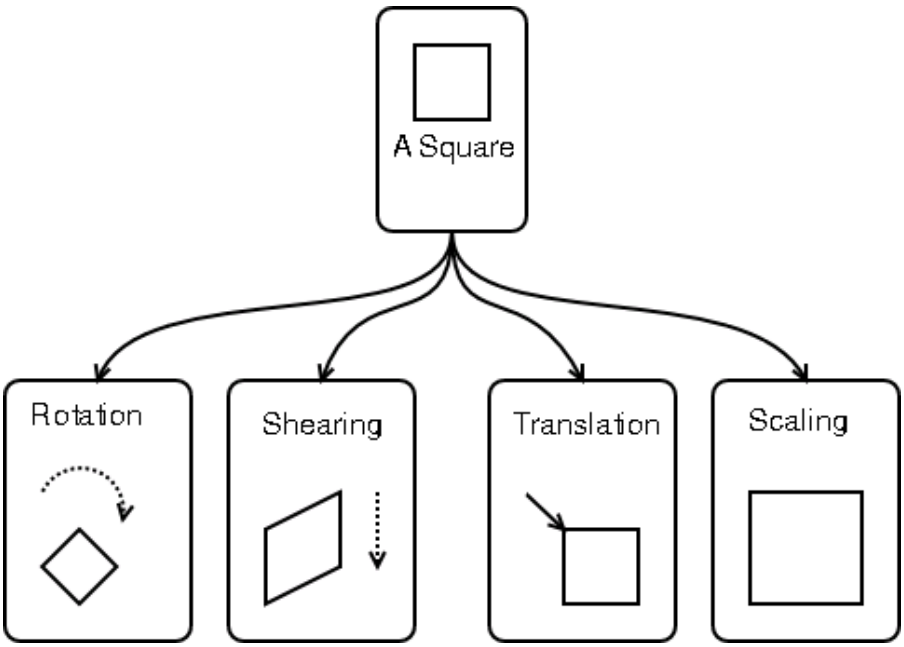


Figure 2.1 Common Affine transformations

An Affine transformation is a geometrical transformation which is known to preserve the parallelism of lines but not lengths and angles. We can also say that these preserve co-linearity of points and their intersection properties. In other words, three points that lie on a line will continue to lie on that line after an affine transformation as shown in figure 2.2 Affine mappings are of the form $Ax + b$ where A is an [n x n] square matrix and x and b are vectors in $\boldsymbol{R}$.

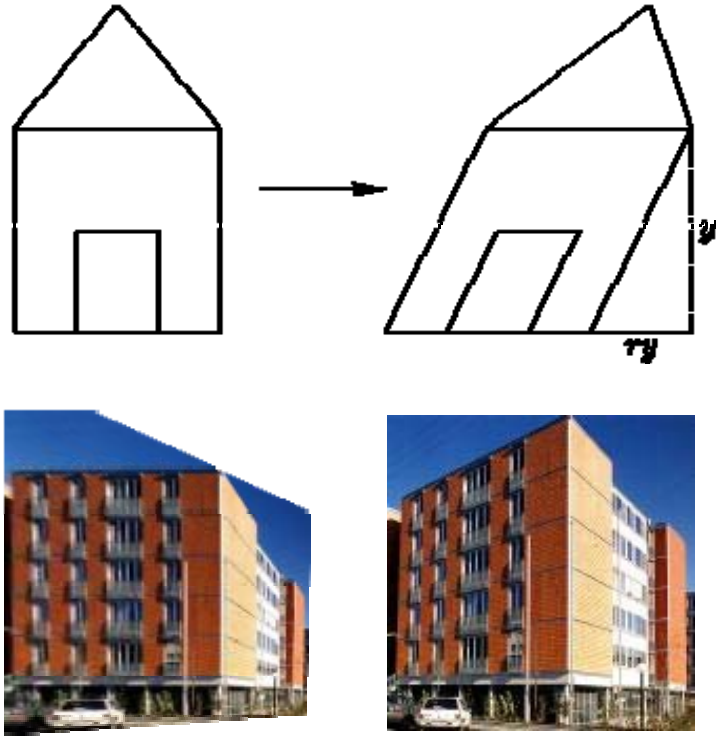


Figure 2.2 Change of line lengths & Angles after shear transformation

More general type of affine transformations shown in figure 2.1 can also be applied in combination as well as selective. For example scaling in only one axis can be termed as squeezing. Corner defined as such may not be the exact edge intersection point which Moravec [83] defined as a corner, but will be in the near vicinity of that, and it will be possible to extract the properties of the region surrounding the edge intersections. As the affine transformations preserve the co-linearity of points on a line and their intersection properties, we can obtain the lines and their intersection point under affine transformations. Furthermore, the corner features are invariant to noise.

C. Line Detection

Hough transform can be efficiently used to search the straight lines in the images [114] using the parameterized line equation (2-1).

$$\rho = x\cos\theta + y\sin\theta \tag{2-1}$$

Each line in the image can be associated with a couple (ρ, θ) which is unique if $\theta \in [0, \pi]$ and $\rho \in \mathbb{R}$, or if $\theta \in [0,2\pi]$ and $\rho \geq 0$. The (ρ, θ) plane is sometimes referred to as *Hough space*. From the Hough space the lines can be found using the inverse Hough transform [118].

The figure 2.3 shows an original image and lines detected using the Hough transform in the original and affine transformed images of the original. The co-linearity of points has been preserved in the affine transformed image.

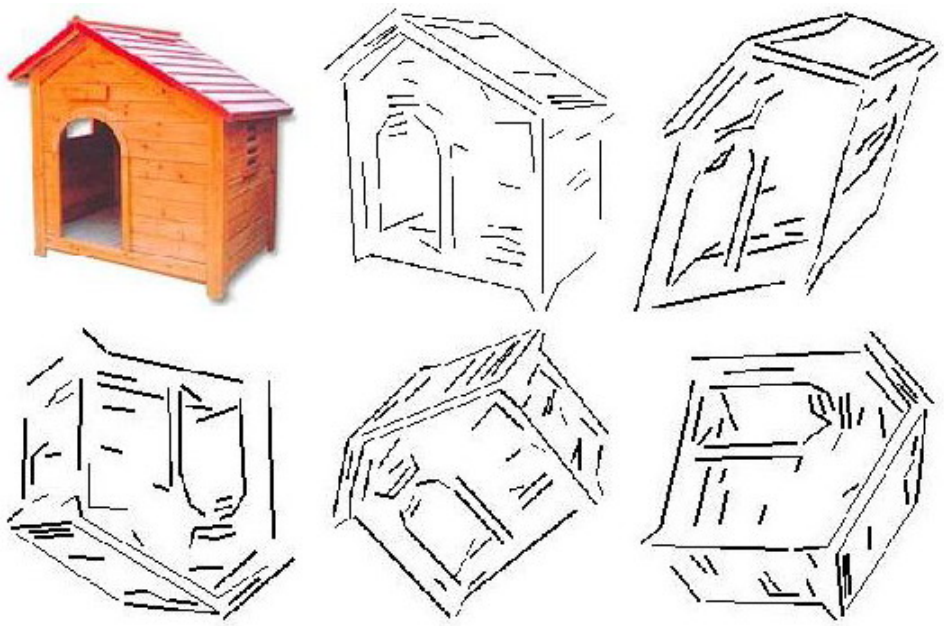


Figure 2.3 Detected lines in transformed image

D. Corner detection

Ideally, a corner is an intersection of two straight lines. However, in practice, corners in the real world are frequently deformed with ambiguous shapes. As corner represent certain local graphic features at abstract level, corners can intuitively be described by some semantic patterns (see Fig. 2.4). A corner can be characterized as one of the following four types:

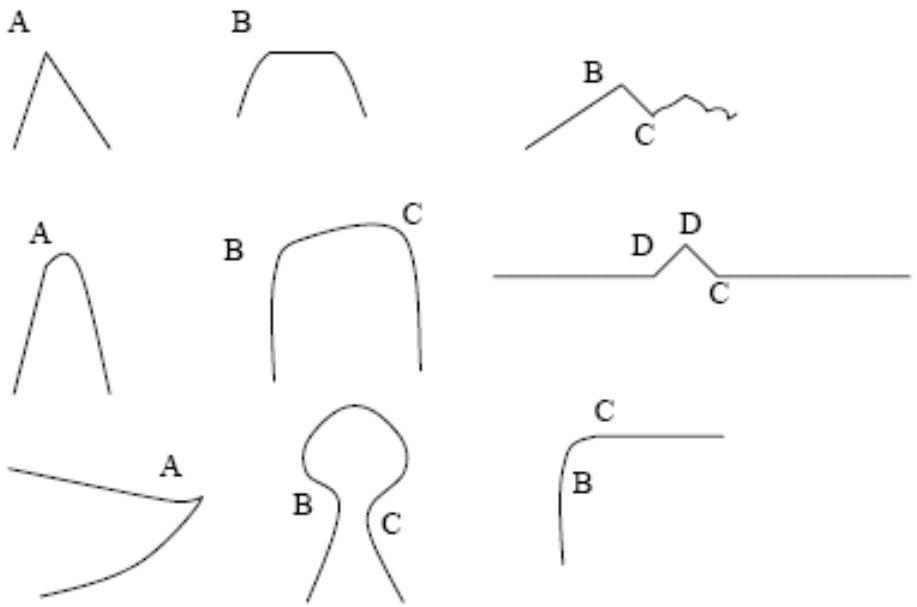


Figure 2.4 Four types of corners

- Type A: A perfect corner as modeled in [109], i.e., a sharp turn of curve with smooth parts on both sides.
- Type B: The first of two connected corners similar to the END or STAIR models in [109], i.e., a mark of change from a smooth part to a curved part.

- Type C: The second of two connected corners, i.e., a mark of change from a curved part to a smooth part.
- Type D: A deformed model of type A, such as a round corner or a corner with arms neither long nor smooth. The final interpretation of the point may depend on the high level global interpretation of the shape.

Figure 2.4 shows some examples of the four types of the corner. It is obvious from the figure that the corner points at very small level are the intersection points of the two straight lines.

For two given line segments with end point coordinates P1,P2, P3, and P4 as shown in figure 2.5 below

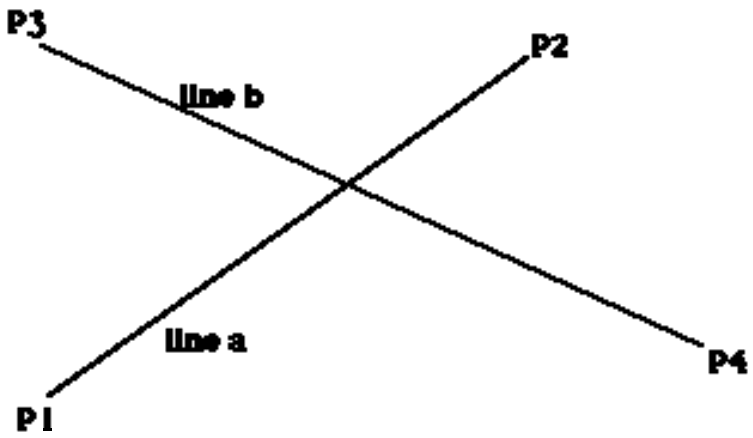


Figure 2.5 Two intersecting line segments

The equations of the lines are:

$$Pa = P1 + u_a (P2 - P1)$$
$$Pb = P3 + u_b (P4 - P3) \quad (2\text{-}2)$$

Solving for the point where Pa = Pb gives the following two equations in two unknowns (u_a and u_b)

$$x1 + u_a (x2 - x1) = x3 + u_b (x4 - x3)$$
$$y1 + u_a (y2 - y1) = y3 + u_b (y4 - y3) \quad (2\text{-}3)$$

Solving gives the following expressions for u_a and u_b

$$u_a = \frac{(x4 - x3)(y1 - y3) - (y4 - y3)(x1 - x3)}{(y4 - y3)(x2 - x1) - (x4 - x3)(y2 - y1)}$$
$$u_b = \frac{(x2 - x1)(y1 - y3) - (y2 - y1)(x1 - x3)}{(y4 - y3)(x2 - x1) - (x4 - x3)(y2 - y1)} \quad (2\text{-}4)$$

Substituting either of these into the corresponding equation for the line gives the intersection point. If the denominator for the equations for u_a and u_b is 0 then the two lines are parallel. If the denominator and numerator for the equations for u_a and u_b are 0 then the two lines are coincident. There are other cases also, such as if point of intersection lies on the projected lines. Because of many intersections of lines, false corners are also detected. To avoid false candidates, the detected corners whose vicinity does not contain any edge point are discarded. Figure 2.6 shows the lines detected using Hough transform and their intersection points in red. Corners were detected by finding the intersecting or near intersecting lines as the corner consisting of two or more intersecting lines is always not possible because of edge deformations. We discarded lines by setting a threshold on the line lengths, so that only prominent lines are considered for finding corners. Figure 2.7 shows close up of a single corner and the detected intersecting lines.

Figure 2.6 Detected lines and corners

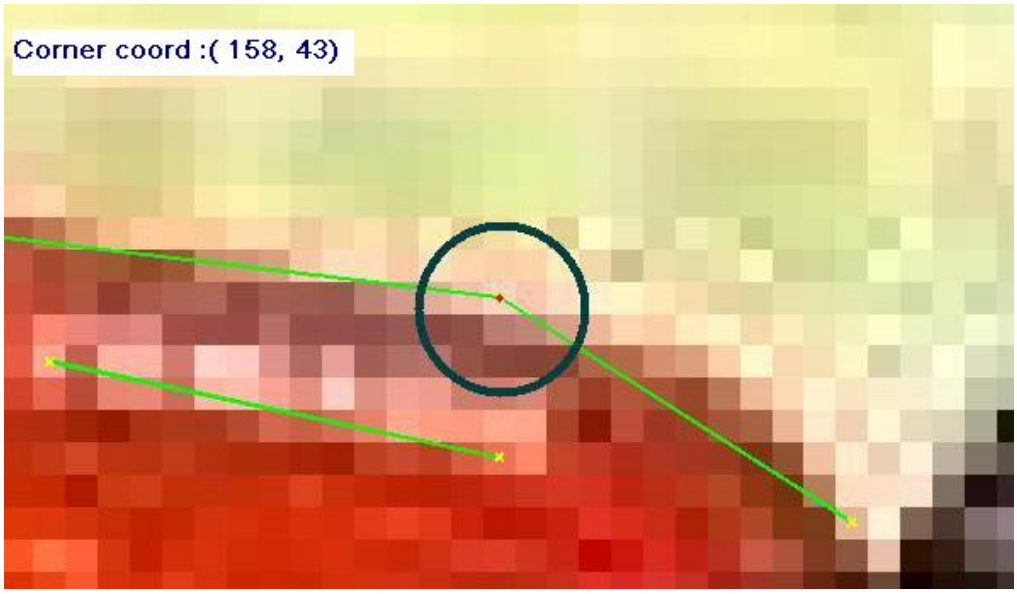


Figure 2.7 A selected corner with coordinates

E. Feature Vector Construction

Once we have the coordinates of the corner point, we need to extract a patch of pixels from around the corner. The shape and size of the patch depends on the approach being followed. In the literature, both square and circular patches have been used. From the extracted patches we compute features to be used of image retrieval.

1. Patch extraction

From the corner point information, feature vector is extracted using a neighborhood operation, for image retrieval. The neighborhood operation can be understood by the following pseudo code:

```
Visit each point p (corner) in the image data and do {
  N = a neighborhood or region of the image data around the point p
  result(p) = f(N)
}
```

Denoting detected corner by 'C' and neighborhood pixel by 'p' we can write:

$$C_i = p_1, p_2, p_3, \ldots\ldots\ldots\ldots\ldots\ldots\ldots p_n$$
$$\text{Where } pi = 0\sim255 \text{ and } i = 1, 2, 3, 4, \ldots, n \qquad (2\text{-}5)$$

From around the corner points a square block of gray level pixel values is extracted. The size of the block is very important and should be such that to capture maximum information around the corner to provide good discrimination characteristics. After experimentation, the size of the neighborhood matrix was chosen as (11x11). So each corner in the image is represented by an

11x11 matrix of the neighboring pixels. Figure 2.8 shows the block of pixels around the corner highlighted in the previous figure. The corner is shown by a red square box.

Corner Coord: (158, 43)					223	224	226	229	229	225	221	219
216	206	208	210	214	219	223	226	228	229	223	216	210
211	213	210	209	213	220	226	229	228	230	223	214	208
224	220	214	209	213	222	229	230	228	229	222	214	209
208	220	227	213	235	210	223	236	209	247	230	211	233
212	228	227	213	230	222	217	220	236	219	192	221	217
111	107	1■5	169	187	206	223	213	229	242	242	233	223
100	101	98	108	93	100	141	150	174	197	222	216	236
127	181	114	130	134	102	91	87	90	121	125	161	173
168	219	167	185	180	148	148	157	115	87	111	106	92
124	139	158	190	183	181	192	189	94	56	108	84	90
74	69	99	108	99	127	148	168	104	94	82	88	85
108	90	84	78	76	81	88	96	100	97	81	86	92

Figure 2.8 A selected corner with neighborhood pixels

2. Descriptive statistical features computation

We take the sample mean and sample variance of each neighborhood and store this for each image of the training data base as its characteristic representation. The sample mean of a set $\{x_{1\ldots}x_n\}$ of n observations from a given distribution is defined by

$$m \equiv \frac{1}{n}\sum_{k=1}^{n} x_k \tag{2-6}$$

The sample variance m_2 (commonly written as s_2 or sometimes s_N^2) is the second sample central moment and is defined by:

$$m_2 \equiv \frac{1}{N}\sum_{i=1}^{N}(x_i - m)^2 \tag{2-7}$$

Where, $m = \bar{x}$ is the sample mean and N is the sample size.

The features that are extracted from the image form a two-dimensional space of mean and variance values. The distribution of these training samples is given in Figure 2.9 which is a 2 dimensional scatter plot of the corresponding mean and variance samples.

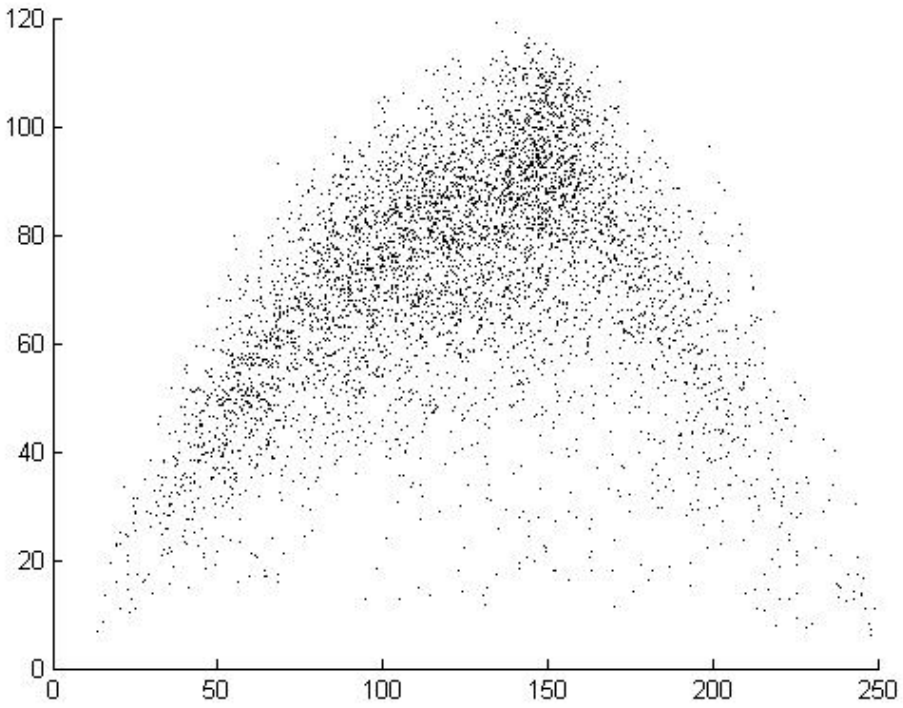


Figure 2.9 Scatter plot of the training data

3. Classification into Class labels

Using the standard K means unsupervised learning approach we partitioned this two-dimensional feature space into a fixed set of Q classes. During training, 7,127 samples of extracted values from 300 randomly selected images were used to train a 20-bin quantizer. The resulting clusters with centroids, after 1,000 iterations are shown in Figure 2.10.

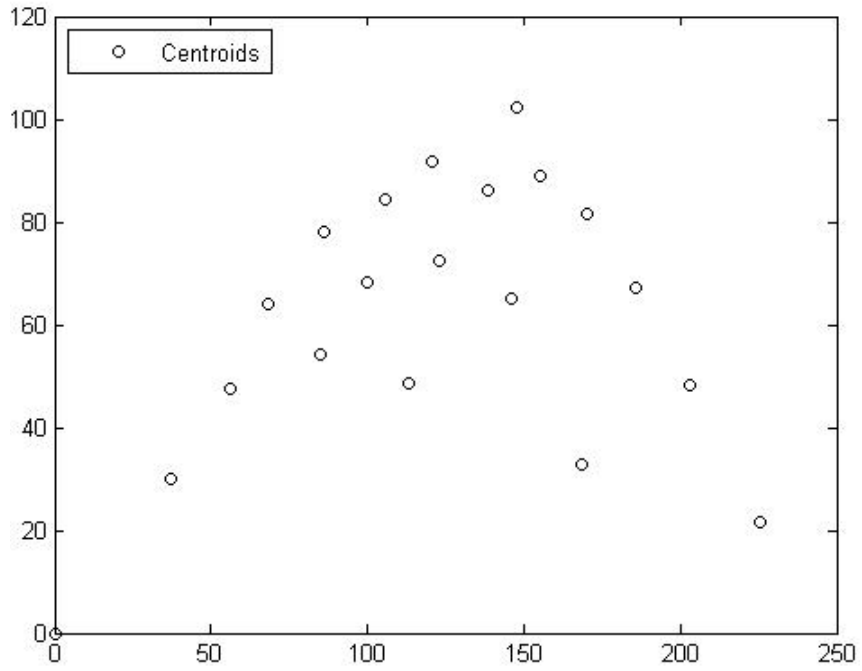


Figure 2.10 Centroids of each cell

We use this labeled data to obtain a Gaussian model of the data which is characterized by the mean and covariance for each class within the data along the number of dimensions. New data points can then be classified by using Bayes rule as in equation 2-8. For each new data point we calculate the posterior probability that point came from each class; the data point is then assigned to the class which gave the highest probability.

$$P(B|A) = \frac{P(B|A)P(A)}{P(B)} \tag{2-8}$$

- P(A) is the prior probability of A. It is "prior" in the sense that it does not take into account any information about B.
- P(A|B) is the posterior probability of A, given B.
- P(B|A) is the conditional probability of B given A.
- P(B) is the prior or marginal probability of B, and acts as a normalizing constant.

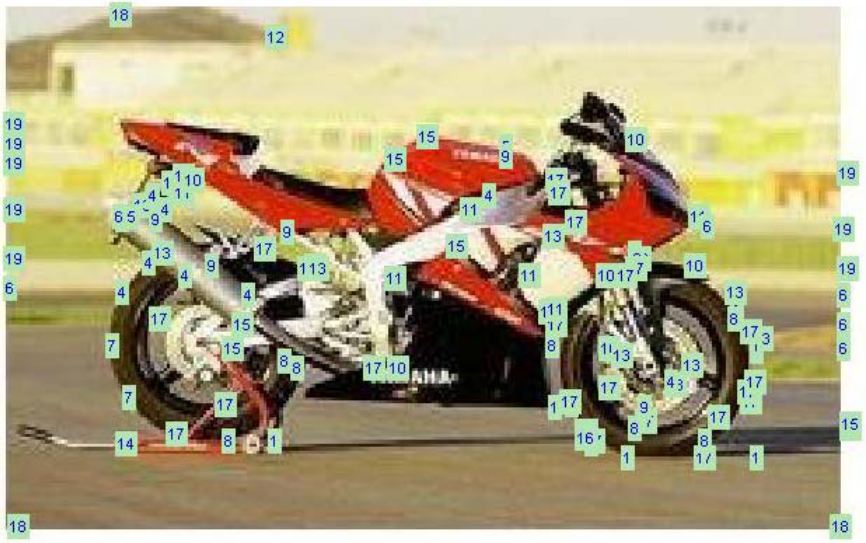


Figure 2.11 Class labels in an image

Using the Gaussian Bayes classifier we assign class labels to all the training image data. Figure 2.11 shows the class distribution in an image. In the figure, the class labels associated to each corner has been pasted on the location of the particular corner from where these were extracted for visual analysis. Notice the similar values for each object, road, grass and bike. Within an

object, motorbike tyres form one group and the metallic body parts form their own clusters. This way one group of classes can describe a particular object

4. Histogram Feature Vector Computation

After the class labels have been assigned, a feature vector is computed by counting the number of mean variance pairs that are assigned to each class. The feature vector for each image is then the Q-dimensional vector which has for its q'th component the number of mean variance pairs that fall into that q'th class In this case Q was taken as 20. This forms the feature histogram or feature vector as shown in figure 2.12.

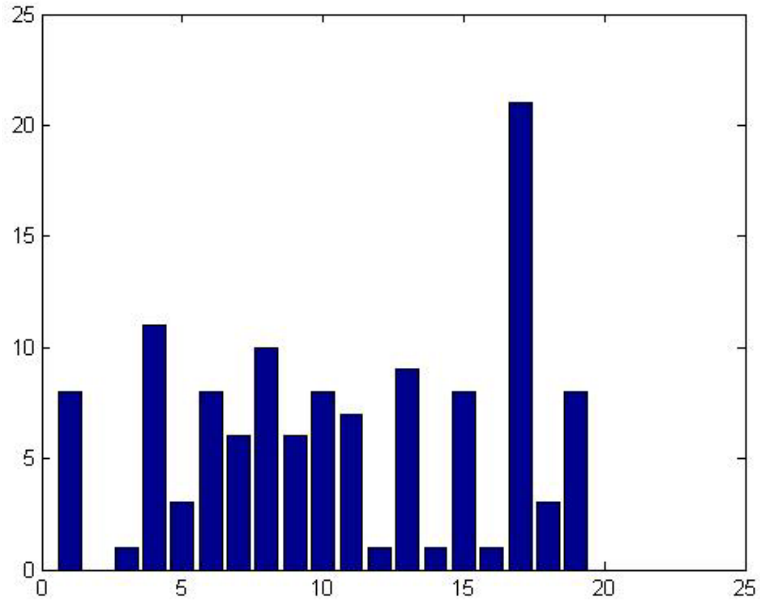


Figure 2.12 Feature Histogram

Figure 2.13 shows the flow process of the feature extraction module from selection of training images from the test dataset to the storage of feature vectors in the feature database.

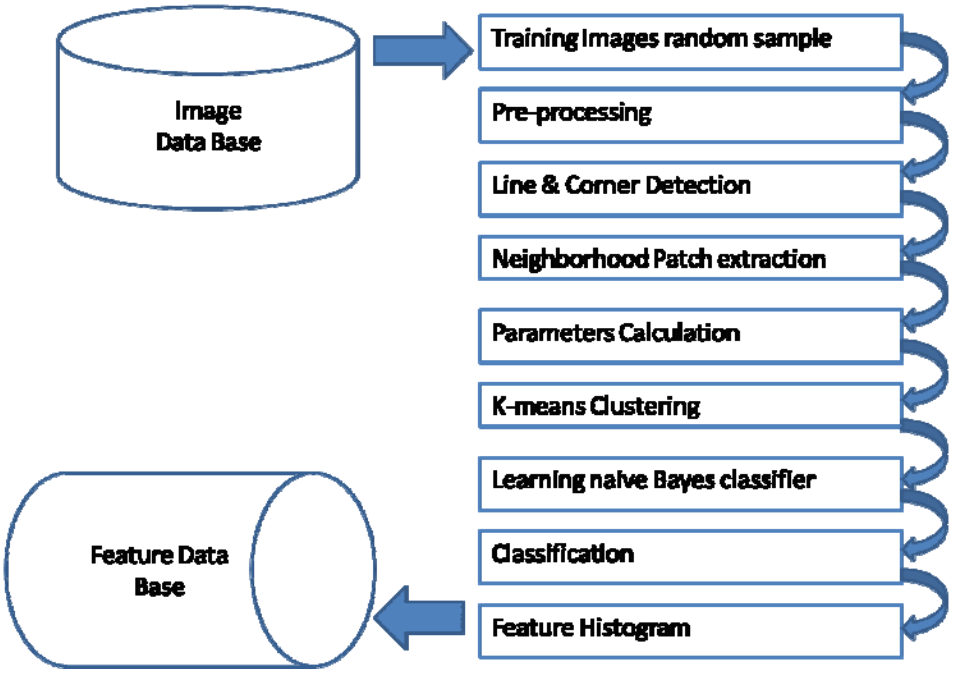


Figure 2.13 Pre-processing and feature extraction module

F. Histogram Similarity Measures for Image Retrieval

Content-based image retrieval calculates visual similarities between a query image and images in a database instead of exact matching. Many similarity measures have been developed for image retrieval based on empirical estimates of the distribution of features in recent years. Different similarity/distance measures will affect retrieval performances of an image retrieval system significantly. In order to evaluate the affect of different similarity measures on the algorithm we used four known similarity measures for histogram matching.

1. Euclidean Distance Measure

The most commonly used Euclidean distance is given as:

$$d_{eucl}(H,H') = \sqrt{\sum_{i=1}^{n}(H_i - H'_i)^2} \quad (2\text{-}9)$$

2. Relative Histogram Deviation Measure

Relative deviation gives the deviation between two histograms as:

$$d_{rd}(H,H') = \frac{\sqrt{\sum_{m=1}^{M}(H_m - H'_m)^2}}{\frac{1}{2}\left(\sqrt{\sum_{m=1}^{M}H_m^2} + \sqrt{\sum_{m=1}^{M}H'^2_m}\right)} \quad (2\text{-}10)$$

3. Relative Histogram Bin Deviation Measure

The Relative bin deviation is the bin-wise deviation between two histograms.

$$d_{rbd}(H,H') = \sum_{m=1}^{M}\frac{\sqrt{(H_m - H'_m)^2}}{\frac{1}{2}\left(\sqrt{H_m^2} + \sqrt{H'^2_m}\right)} \quad (2\text{-}11)$$

4. Quadratic Distance Measure

Quadratic Forms are capable of considering the similarities between different bins by incorporating a matrix $A = A_{m,n}$ with $A_{m,n}$ denoting the dissimilarity between the bins m and n [14]. Let H and H' be the histograms represented as vectors, the Quadratic form can be calculated as:

$$d_{qd}(H,H') = \sqrt{(H - H')^T \cdot A \cdot (H - H')} \quad (2\text{-}12)$$

A high dissimilarity between the underlying values of different bins H_m and H'_n is denoted by a high value $A_{m,n}$, thus differences between these bins are taken into account stronger than differences between bins $H_{m'}$ and $H'_{n'}$ where $A_{m',n'}$ is a low value. A common setting for the $A_{m,n}$ is

$$A_{m,n} = 1 - \frac{d_2(v^m, v^n)}{d_{\max}} \tag{2-13}$$

Where $d_2(v^m, v^n)$ is the Euclidean distance between the values represented by bins m and n respectively and

$$d_{max} = max_{m,n}\, d_2(v^m, v^n) \tag{2-14}$$

G. Test Data set

For testing the proposed idea, we used the dataset provided by The Californian Institute of Technology (Caltech) on the institute's website [129]. The dataset is in the form of various visual object classes facilitating the evaluation process. The size of each image is roughly 300 x 200 pixels. We used the classes, Motorbikes, airplanes, soccer ball, doors, leaves, grand piano, helicopter, pyramid, schooner, scissors, starfish, stop sign, stapler, chair and minaret. The number of images in each class varies between few hundreds to around 50. For performance evaluation of the algorithm on this database, we used the precision and recall measure.

H. Image Retrieval experiments

The framework for image retrieval is based on a query or example image or sketch as input to the system and the result is a list of images ranked by their similarities with the query image.

Figure 2.14 depicts the process flow of a query module. First the feature vector of the query image is computed in run time and then similarity matrix is computed. Based on which the list of relevant images is sent as output. Since content based image retrieval is all about visual information retrieval, in order to discuss various aspects of experiments carried out, four results from the four distance measures used are displayed from the visual class motorbike. The results displayed are the first 25 results obtained in a random query.

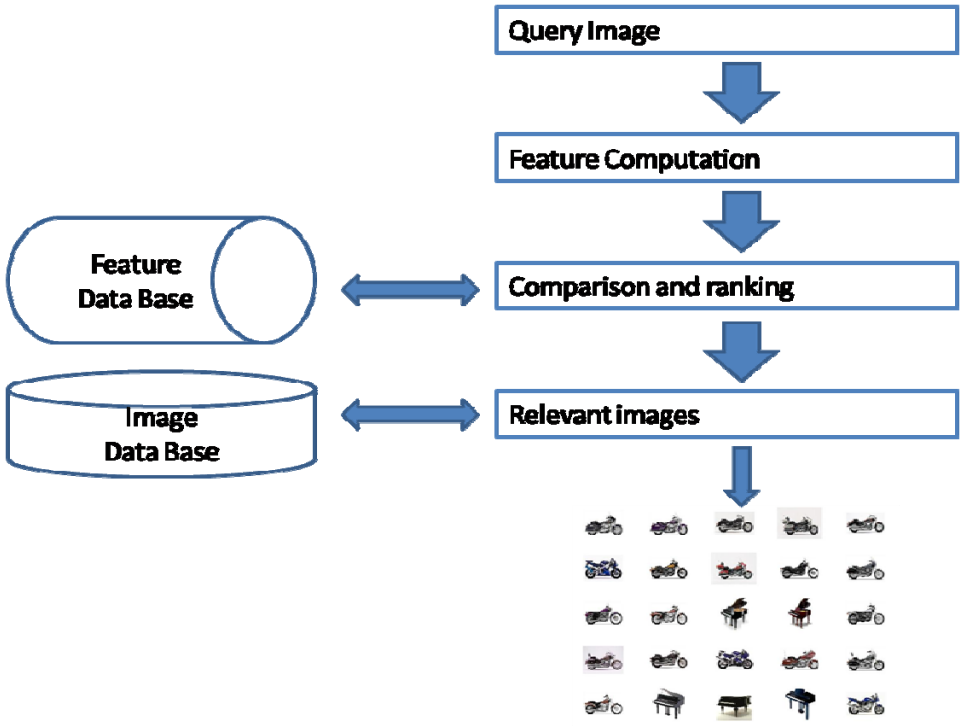


Figure 2.14 CBIR Query Module

Figure 2.15 below shows the query image for the displayed results from the class motorbike. From figure 2.16 till 2.19, we can see that the relative bin deviation measure has more discriminative power in this case. However with different databases the behavior of distance measure changes. One distance measure performing well for one data base may not be giving

that much accurate result in another. So the choice of similarity measure still remains an open ended question.

Figure 2.15 Query Image

The grand piano in case of Euclidean distance and relative histogram deviation has been detected as a false positive at different levels. In Euclidean distance measure, false positives are more than the relative histogram deviation measure. In figure 2.16, first false positive is at image 13 and then at 14 with a total of 5 false positives in a total of 25 results. Precision = 20/25 = 0.8

Where as in figure 2.17 of relative histogram deviation measure, we got total 3 false positives but their weightings are different. First false positive is image 10. Precision = 22/25 = 0.88

In figure 2.18 the relative bin deviation measure performed the best with a steady precision with recall. For the displayed result, Precision = 1

In case of figure 2.19 of quadratic distance measure, in the first 25 results there is only one false positive. However, in the averaged performance curve, its precision falls in the lowest category. For the displayed results, Precision = 24/25 = 0.96

Figure 2.16 Euclidean Distance Measure Results

Figure 2.17 Relative Histogram Deviation Measure Results

Figure 2.18 Relative Histogram Bin Deviation Measure Results

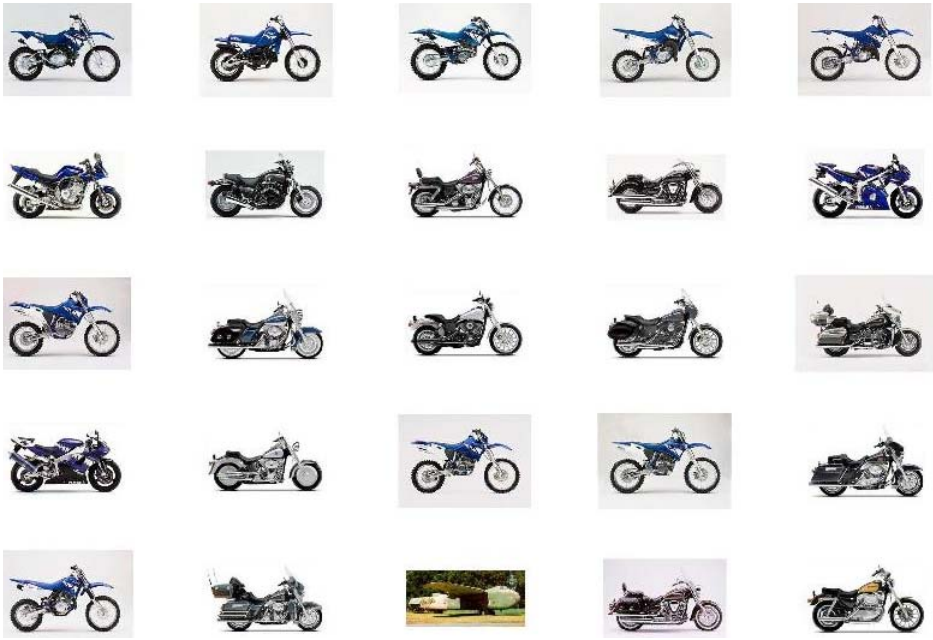


Figure 2.19 Quadratic Distance Measure Results

I. Performance Evaluation

Two traditional measures for retrieval performance in the information retrieval literature are precision and recall. Precision is defined as the percentage of retrieved images that are actually relevant

$$Precision = \frac{\#\ of\ Relevant\ Images\ Retrieved}{Total\ \#\ of\ retrieved\ Images} \tag{2-15}$$

Recall is defined as the percentage of relevant images that are retrieved

$$Recall = \frac{\#\ of\ Relevant\ Images\ Retrieved}{Total\ \#\ of\ relevant\ Images} \tag{2-16}$$

Given a query, high precision implies that very little irrelevant images have been retrieved and high recall implies that much of what is relevant in the database have been retrieved. Lack of precision can be compared to a type 2 error (false alarm) and deficiency in recall for a given search is comparable to type 1 error (misdetection). For performance evaluation, one can plot precision and recall as a function of the number of images retrieved as well as the precision versus recall curves for different numbers of images retrieved. To evaluate the overall retrieval performance (precision and recall), first, the database is queried with each of the images in test database consisting of images from different visual classes, then average precision and recall percentages are computed for the entire database. To rank-order the database images, distance measures discussed above are used. Figure 7 shows the averaged precision and recall for the entire database.

Figure 2.20 shows the averaged precision and recall for the entire database. The low precision part of the quadratic distance is because it considers the similarities between different bins. The concept behind this measure is that because of external conditions the positions of bins may shift so instead of matching the corresponding bins other bins must also be taken into account.

For increased recall the performance of quadratic distance gets better but inferior to other distances. However in case of color histograms it is considered to have superior performance.

One more question on the performance measure has been debated from the view point of users, that a user is mostly interested to check only first 20 or may be 40 results vis-à-vis time and interest constraints. This point of view considers the accuracy of only first pile of results which a user sets as threshold, “Like show me best 50 results”. Because the ultimate decision of relevance has to be from the user. With these arguments, the displayed results are good.

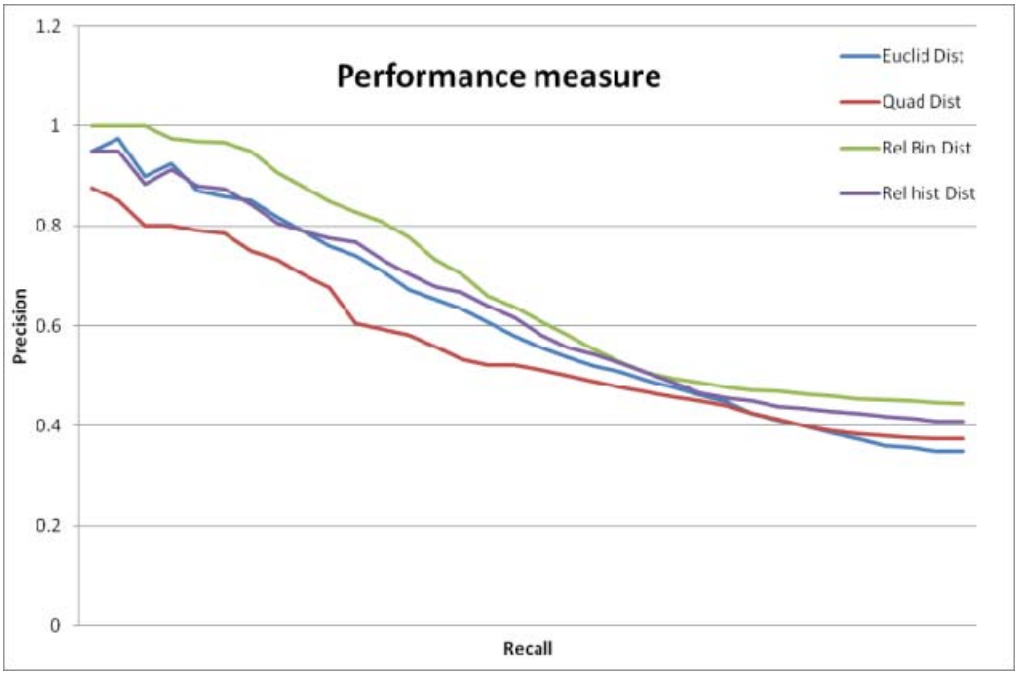


Figure 2.20 Averaged Precision and Recall Curves for Different Distance Measures

J. Comparison with known approaches

Figure 2.21 below shows the precision and recall curves for three known approaches, compared with our algorithm. For comparison, same conditions were applied to all the algorithms. The distance measure used was ‘relative histogram bin deviation measure’, as it performed better on

our algorithm. The curves marked as 3 and 4 are algorithms based on color only, which explains their low performance in a diverse data set of semantic objects having different colors and textures. The proposed feature set is significantly smaller in size compared to the algorithms using color features typically color correllograms and thus is computationally very efficient.

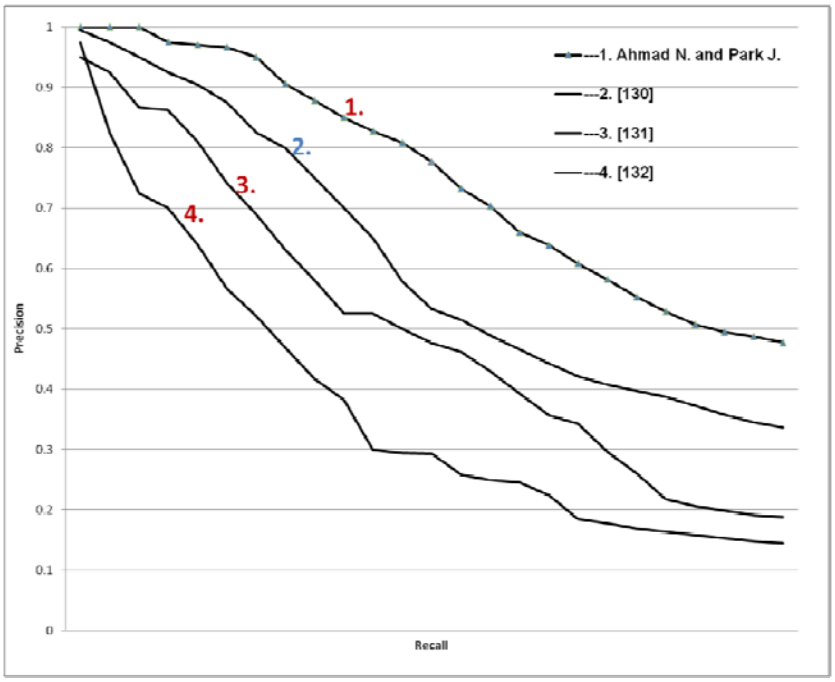


Figure 2.21 Comparative Precision and Recall Curves for Different Approaches

III. Visual Object Class Recognition using Semantic Image Structure

The chapter discusses two new approaches to explore and extract semantic meanings in visual object class structure for visual object class recognition. The approaches are based on exploiting semantic relations and micro-level semantic structural groupings, in a semantic object structure. The algorithms uses the hypothesis in line with Gestalt laws of proximity that; in an image, basic semantic structures are formed by line segments (arcs also approximated into line segments based on pixel deviation threshold) which are in close proximity of each other. These basic semantic structures are hierarchically combined till such a point where a semantic meaning of the structure can be extracted. One of the presented approaches exploits these hierarchical relations for constructing a semantic object to learn a classifier in a K-nearest neighbor framework. The other approach constructs a graph model for classification based on micro-level semantic groups and their inter-relations. Geometrical properties of the semantic relations are used to generate rotation, translation and scale invariant histograms which are used for making recognition decision in the K-nearest neighbor framework whereas in the graph model, invariant geometrical properties of the groups and relations are used as vertex and edge labels. The graph model presented, captures the inter class variability by analyzing the repetitiveness of structures and relations and uses it as a weighting factor for classification. The algorithms has been tested on standard benchmark database and results are compared with

existing approaches to understand the strengths and weaknesses of the semantic approaches vis-à-vis other approaches.

A. Exploring semantic level intelligence in data

Recognition of a multitude of objects as dogs, cars etc. is an un-noticeable every day activity, hardly considered an achievement of any subtle order. In contrast, it is a very active research area in computer world and the capability of computers in this regard makes an interesting reading. In the preface of the book [133], it is mentioned in these words:

Object recognition — or, in a broader sense, scene understanding— is the ultimate scientific challenge of computer vision. After 40 years of research, robustly identifying the familiar objects (chair, person, pet), scene categories (beach, forest, office), and activity patterns (conversation, dance, picnic) depicted in family pictures, news segments, or feature films is still far beyond the capabilities of today's vision systems.

It is interesting to note in this context that, for human vision; the general classification of an object such as a 'car' is usually easier than the identification of the specific make of the car [79]. In contrast, current computer vision systems can deal more successfully with the task of recognizing a specific car compared with classifying an object into a general category as 'car' [134]. So the problem in object recognition is to determine which, if any, of a given set of objects appear in a given image or image sequence. Essentially this is a problem of matching models from a database with representations of those models extracted from the low level image features such as color, texture, shape or the spatial location of image elements. In the image retrieval literature, we come across the notion of 'semantic gap' at various places as discussed in chapter I section D. The sprung up logic as a result of this thought process is very

simple; since we talk about visual solutions (like given by humans and they are really good at it), we should try to follow human's pattern of understanding an image.

Near the turn of the 21st century researchers finally got convinced that next evolution of systems would need to understand the semantics of an image, not simply the low level underlying computational features i.e., "bridging the semantic gap" [135]. From a pattern recognition perspective, this roughly meant translating the easily computable low level content-based media features to high level concepts or terms which would be intuitive to the user. The result of this thought process was the focus on the possibilities of bridging the semantic gap between the man and machine. The efforts made followed both the top down and bottom up approaches. The top down approaches studied how the human vision makes semantic decisions. Mojsilovic and Rogowitz [136] conducted psychophysical experiments to gain insight into the semantic categories that guide the human perception of image similarity. They used these data to discover low-level features that best describe each category. Lew [135] and others studied translating the easily computable low level content-based media features to high level concepts. In object recognition literature, we also find similar change in approaches, as Serre et al. [137] presented a set of features for object recognition, based on quantitative model of visual cortex. Such efforts trying to follow the human patterns of scene understanding implicitly imply that for visual solutions we can't ignore to explore the underlying principles of human vision.

B. Approaches to Generic Object Class Recognition

Significant progress has been made in the recent years towards object recognition [133]. Early attempts on object recognition were focused on using geometric models of objects to account for their appearance variation due to viewpoint and illumination change. An excellent review on geometry-based object recognition research by Mundy can be found in [138].

In contrast to early efforts on geometric model based object recognition works, later the focus shifted on appearance-based techniques. David Lowe [139, 140] pioneered the approach by using scale-invariant 'SIFT' features. Since then, there has been a lot of work using appearance based techniques. There is an excellent survey by Alexandra Teynor [141] covering the techniques used so far in the area of 'appearance', 'patch' or 'key-point' based approaches. There are other good evaluation papers covering strengths and weaknesses of various aspects of the feature based approaches [121 – 125].

Here we also witness that research inspired from human biological vision is getting the attention of researchers. A new set of biologically inspired features that exhibit a better trade-off between invariance and selectivity than template-based or histogram based approaches was proposed [137]. The latest work by Mutch and Lowe [142] is an extension of quantitative model of visual cortex by Serre et al. [137], proposing some modifications in the approach with improved performance.

The ideas of semantic or perceptual grouping for computer vision have their roots in the well known work of Gestalt psychologists in 1920's [143] who described, among others the ability of the human visual system to organize parts of the retinal stimulus into organized structures. The word Gestalt means "whole" or "configuration". Gestalt psychologists observed the tendency of the human visual system to perceive configurational wholes. With rules that governs the uniformity of psychological grouping for perception and recognition, as opposed to recognition by analysis of discrete primitive image features. The grouping principles proposed by Gestalt psychologists embodied such concepts as grouping by proximity, similarity, continuation, closure, and symmetry.

Perceptual organization is a primitive explanation of the processes that generated the image. Deeper explanations are constructed by labeling, elaborating, and refining the primitive ones

[144]. The goal of perceptual grouping in computer vision is to organize image primitives into higher level primitives thus explicitly representing structure contained in the image data [145]. The final structure obtained after grouping all lower level features to a higher-level structure will represent the shape of an object in an image. A precise model of the object may still be required for recognition. In case of humans we obtain that model through learning since birth and also through inherited knowledge.

In computer vision, the term “perceptual organization” has been used by various researchers in various contexts, at different levels of vision processing, and with respect to different feature types. This practice has blurred the meaning of the term “perceptual organization”. Perceptual groupings differ from one another with respect to the types of constituent features being organized and the dimensions over which the organizations are sought [146]. It means that different authors have considered different ideas under the banner of perceptual groupings and no two are alike.

The true heart of visual perception is the inference from the structure of an image about the structure of the real world outside [147]. Approaches extracting semantic meanings from the image structure including line segments, different shapes such as ‘L’, ‘U’, etc which the line segments make and incorporating other features as color and texture to make these more meaningful are found in literature. These approaches basically follow the human visual system, which has the ability to link together image features arising from the same physical source (e.g., the same object). Etemadi [148] proposed a frame work for low level grouping of straight lines following the work in perceptual grouping. He proposed to group parallel, collinear and intersecting lines in a hierarchical order. He then further subdivided parallel lines in overlapping and non overlapping line groups and grouped intersecting lines based on the location of their junction point, if it lies on or away from the lines. Further subdividing these on

the basis, if they form a ‘V’,’T’, ‘⊥’ or ‘L’ shape. He however did not take into consideration the distance or spatial placement of these line segments with respect to each other.

For detecting man-made objects in non-urban scenes, Lu and Aggarwal [149] proposed a framework based on perceptual organization. The framework grouped low level image features hierarchically into regions-of-interest (ROI), likely to enclose man-made objects or a substantial part of the man-made objects. For detecting large man made structures such as buildings, Iqbal and Aggarwal [150] proposed a framework based on perceptual line groupings. The approach was based on the ‘Principle of non-accidentalness’. Meaning that in case of man made features; line segments have an order where as in other cases the objects lack such an order. To exploit ‘non-accidentalness’ nature of man made structures they placed the extracted line segments from an image into various groups as, straight line segments, longer linear lines, co terminations, “L” junctions, “U” junctions, parallel lines, and Polygons. Based on these characteristics they trained a classifier on a database consisting of three classes: structure, non-structure and intermediate. The proposed framework takes an image and computes the above described line segment groupings of the whole image. The algorithm works globally and does not take into account spatial arrangements of the line segments in relation to each other and their contribution to form semantic objects.

C. Image structure analysis for semantic features

The algorithm builds on the idea that putting a minimum number of line segments in close proximity to each other forms a basic semantic structure. The other important properties are the relative segment lengths and angles. Hierarchically combining these basic semantic structures makes possible for human brain to interpret the whole structure as something meaningful.

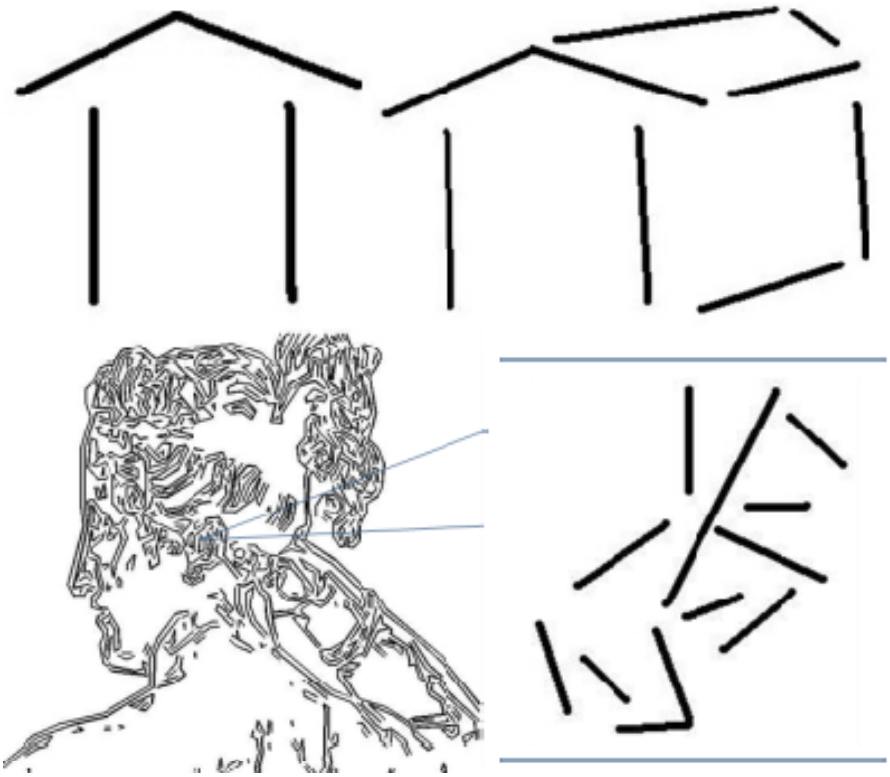


Figure 3.1 Semantic meanings to line segments

This can easily be explained using figure 3.1, showing the formation of basic semantic structures. First row of figure 3.1 shows two basic semantic structures made from different number of line segments. First consists of just four line segments and second consists of nine line segments. For both the structures we can use the word ‘hut’, ‘house’, ‘dog house’ or something else depending upon the person trying to describe the sketch. In the second row of figure 3.1, line sketch of a person’s side view is shown. Close up of line segment groups are shown in the callout. One thing can be appreciated here that, more line segments mean more clarity in making a categorization decision by the brain. In answering if it is a semantic structure or not and can the structure be semantically described using language. The line

segments in the figure get semantic meanings when they are placed at a close distance from each other at certain angles having certain lengths with respect to each other. The relationship of minimum distance remains the same under various geometric transformations though the segment lengths and angles may change. The basic semantic structure formed in this way can have some lower level or basic semantic meanings. Lower level means that the structure may not have any clear semantic level meanings on its own, without being combined hierarchically with other groups to give true semantic meanings.

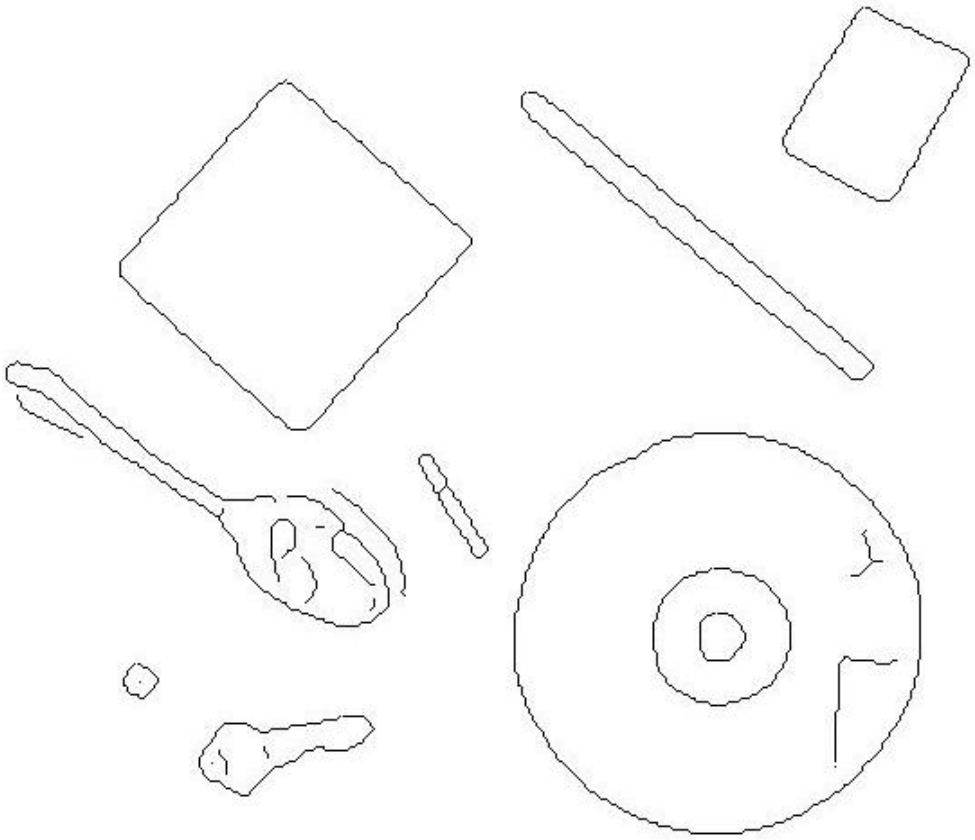


Figure 3.2 Edge pixels in an image

D. Transforming image structure into a line segment model

In order to follow the semantic grouping idea, we need to transform the image structure into a line segment model. We can think of an image edge map consisting of staggered lines and curves. Figure 3.2, shows binary edge map of an image showing different objects. The edges can be generalized as consisting of staggered lines, curves and circles. The semantic grouping approach discussed above, only talked about lines and not curves or circles. As the curved shapes and circles carry important information about the semantics of an object, these can not be ignored. So, the proposed semantic grouping approach has to account for curves and circles constituting a semantic object.

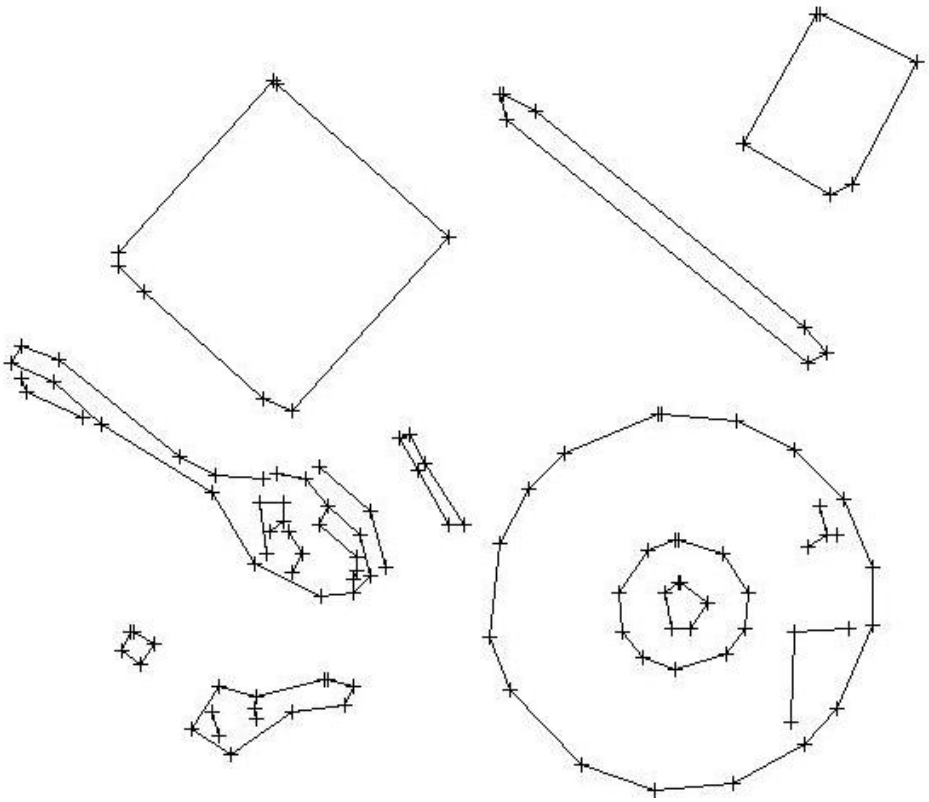


Figure 3.3 Extracted line segments

We have followed the approach of breaking down the curves and the circles into smaller line segments based on pixel deviation. This way the general semantic meaning of a shape or an object remains unchanged and we can implement the grouping approach. For this purpose we have adopted the algorithms of [151] and [152]. The algorithm takes the edge map of an image, performs edge linking, removing isolated pixels and edges below a threshold of pixel length. In the next step a parameter is introduced which controls the threshold of the maximum allowed line tolerance, i.e. pixels that are too far off the line segment. The pixels which are below the tolerance level are grouped into line segments. Similarly all the edge lists are converted into line segments. Then we combine the line segments which are within a specified distance and angle tolerance. Figure 3.3 shows the line segments obtained using the approach.

E. Parameter of proximity

In order to translate the notion of 'close proximity' between two line segments into the mathematical domain, we find a point on each line segment such that the distance between the two is minimum compared to other points on respective line segments. This will be our parameter of proximity for the grouping approach.

In case of an image domain the line segments are in a two dimensional plain and either are parallel or intersecting. The parallel line segments can be overlapping or non-overlapping and in case of intersecting line segments, the point of intersection may lie on or away from the line segments or even out of the image boundaries. For finding the minimum distance we use the below derivation.

Using the parametric line equation defined by two points we can write

$$L_1: P(s) = P_0 + s(P_1 - P_0) = P_0 + su \tag{3-1}$$

$$L_2: Q(t) = Q_0 + t(Q_1 - Q_0) = Q_0 + tv \tag{3-2}$$

Whereas, $P(s)$ is the line segment on line L_1, and $Q(t)$ is the line segment on line L_2. The parameters s and t are real numbers. Whereas;

$$u = P_1 - P_0 \ \ and \ v = Q_1 - Q_0 \tag{3-3}$$

are line direction vectors.

We have to find the two points, P and Q, whose distance is minimum compared to other points on the respective lines and the points P and Q must lie on the respective line segments.

Let $w(s, t) = P(s) - Q(t)$ be a vector between points on the two lines. We want to find the $w(s, t)$ that has a minimum length for all s and t.

Minimizing the length of w is the same as minimizing;

$$\begin{aligned} |w|^2 &= w \cdot w \\ &= (w_0 + su - tv) \cdot (w_0 + su - tv) \end{aligned} \tag{3-4}$$

which is a quadratic function of s and t. In fact, it defines a parabaloid over the (s, t)-plane with a minimum at intersection point $C = (s_c , t_c)$, and which is strictly increasing along rays in the (s, t)-plane that start from C and go in any direction.

We compute where the minimum occurs on each line segment by substituting s and t for 0 and 1 and solving the equation for vector w.

Consider the edge $s = 0$, by substituting in equation 3-4, we get,

$$|w|^2 = (w_0 - tv) \cdot (w_0 - tv) \tag{3-5}$$

Taking the derivative with t we get a minimum when:

$$0 = \frac{d}{dt}|w|^2 = -2v \cdot (w_0 = tv) \tag{3-6}$$

From equation 3-6 we can calculate t shown in equation 3-7. This gives a minimum on the edge at (s_0, t_0) where $s_0 = 0$ and $t = t_0$:

$$t_0 = v \cdot w_0 / v \cdot v \tag{3-7}$$

If $0 <= t_0 <= 1$, then this will be the minimum and $P(0)$ and $Q(t_0)$ are the two closest points of the two segments. However, if t_0 is outside the edge, then we will have to check other cases for the true minimum.

Similarly, for $s = 1$, $t_1 = (v \cdot w_0 + v \cdot u) / v \cdot v$ (3-8)

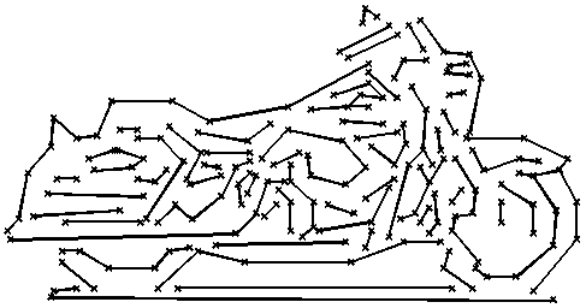
for $t = 0$, $s_0 = - u \cdot w_0 / u \cdot u$ (3-9)

for $t = 1$, $s_1 = u.v - u.w_0 / u \cdot u$ (3-10)

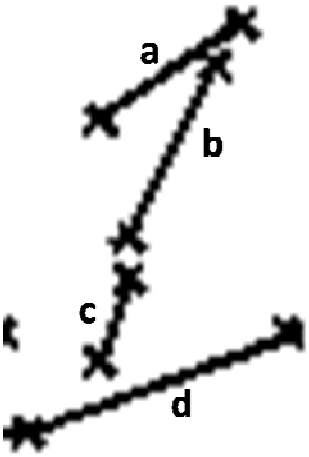
F. Defining a Transitive Relation for Semantic Modeling

The figure 3.4(a) shows a simple picture of a semantic object whose general category is 'Motorbikes'. Semantically this is not a complex category and it has a very peculiar structure like 'two wheels' and a 'handle', which helps in its identification even by children rather quickly. The figure 3.4(b) shows the line segment model or more generally line sketch of the object motorbike. For humans it is very easy to categorize this line segment model. There are hardly any chances that someone will categorize it as something else such as 'shovel'. The line segments in the figure get semantic meanings when they are placed at a close distance from each other at certain angles having certain lengths with respect to each other. The relationship of minimum distance remains the same under various geometric transformations though the segment lengths and angles may change.

(a)



(b)



(c)

Figure 3.4 Original Image, Line segment model and a close-up showing four closely placed line segments

The basic semantic structure made by one group of line segments close to each other at a certain threshold distance, can have some lower level or basic semantic meanings. Lower level means that the structure may not have any clear semantic level meanings on its own, without being combined hierarchically with other groups to give true semantic meanings. The criteria for bottom up hierarchical grouping can be explained easily using figure 3.4 (c), which shows four closely placed line segments ‘a’, ‘b’, ‘c’, and ‘d’. The approximate minimum distance between these four line segments can be determined by visual inspection. The line segment ‘a’ is close to ‘b’ compared to the other line segments. The line segment ‘b’ is close to ‘a’ and ‘c’ and ‘c’ is close to ‘b’ and ‘d’, whereas ‘d’ is close to only ‘c’.

We can define a binary relation 'is close' denoted by '$\Re$' based on a minimum distance threshold between line segments for all the line segments (a, b, c and d) in the image (X) of figure 3.4(c). A binary relation $\Re$ over a set X is transitive if it holds for all members a, b and c in X, that if a 'is close' to b and b 'is close' to c, then a 'is close' to c. Using predicate logic we can write this transitive relation as:

$$\forall a,b,c,d \in X, a\Re b \wedge b\Re c \wedge c\Re d \Rightarrow a\Re d \tag{3-11}$$

Or more simply as:

$$if \quad a = b,\ b = c \ and \ c = d, \ then \quad a = d \tag{3-12}$$

This way all the four line segments in figure 3.4(c) form part of a semantic hierarchical group.

1. Feature representation

After line extraction and minimum distance calculation between line segments, we form the line segment groups using the transitive relation of equation (3-11). This gives us semantic line groups in an image. For further processing, we have discarded lines by setting a threshold on the line lengths, so that only prominent lines are considered and the rest which mostly provide object details are discarded.

For feature construction using line segments, first we have to consider the effect of various affine transformations; as the affine transformations do not preserve line lengths and angles. A Euclidean distance matrix (EDM) is an $(n \times n)$ matrix representing the spacing of a set of n points in Euclidean space. If A is a Euclidean distance matrix and the points are defined on m-dimensional space, then the elements of A are given by.

$$A = (a_{ij});$$

$$a_{ij} = || x_i - x_j||_2 \tag{3-13}$$

Where $||.||_2$ *denotes the 2-norm on* R^m.

A common translation of all points will not affect an EDM since the change of the point co-ordinates is nullified. Similarly, an EDM is invariant against rotation and also against scaling if the matrix is normalized in the range of [0, 1], otherwise it is scale invariant up to a factor '*S*' [153]. In view of these invariance properties, we compute EDM's from the geometric properties of the line segments.

For each semantic group, let L = $\{l_i | i = 1, 2, ...,N\}$, be the set of line segments obtained. Then we can compute geometric properties of L: the angles formed by all segments between each other and the relative length of each segment with respect to all other line segments. The relative minimum distance between each has already been considered based on which we made the semantic groups. The angle between two line segments can be calculated as:

$$\cos\theta = \left| \frac{u \cdot v}{|u| \cdot |v|} \right| \tag{3-14}$$

where, u and v are line direction vectors of two line segments from equation 3-3. The length of segment $l(i)$ with end points (x_0, y_0) and (x_1, y_1) is given as:

$$len(l_i) = \sqrt{(x_0 - x_1)^2 + (y_0 = y_1)^2} \tag{3-15}$$

Relative lengths of the line segments for constructing EDM are calculated as:

$$a_{ij} = \left| l_i - l_j \right| \tag{3-16}$$

Where a_{ij} is the element of EDM from equation 3-13 with row '*i*' and column *'j'*.

We normalize the relative line length data in order to bring it in [0, 1] range as follows:

Given a lower bound l and an upper bound u for a feature component x,

$$\bar{x} = \frac{x - l}{u - l} \tag{3-17}$$

results in $\bar{x}$ being in the range of [0, 1]. Now we have angles in the range of (±π) and relative line lengths in the range of [0, 1].

Since every EDM is symmetric, we extract the upper triangle matrix and form a histogram from each EDM with different resolutions based on empirical testing.

$$\begin{aligned} H_{ang} &= \left\{ h_{b_a}^{ang} \right\}, b_c = \{1,2,3,\ldots\ldots,B_a\} \\ H_{len} &= \left\{ h_{b_l}^{ang} \right\}, b_l = \{1,2,3,\ldots\ldots,B_l\} \end{aligned} \tag{3-18}$$

where B_a and B_l denote the different number of bins of the three histograms. For H_{ang}, 72 or 36 bins that correspond to a 5 and 10 degree angle resolution, produced the best results. The resolution for H_{len} depends more on the application data then H_{ang}. However, we found out that, 25 bins result in a robust and compact histogram feature.

2. Experiments and results

In order to test the performance of the proposed algorithm and make comparisons of our results with state of the art algorithms we chose the classification results of several different authors.

Because many authors have reported the classification rates of their algorithms on a subset of the data and on class-wise classification methodologies, i.e. a classifier was trained in order to discriminate a single class among the subset from a background class consisting of arbitrary images. Multiclass object categorization has been dealt in a less frequency.

Our recognition framework is based on k-nearest neighbor classifier (k-NN). The k-NN classifier generalizes in a straightforward manner to multi-class classification. Given a training set E of m labeled patterns, a nearest neighbor procedure decides based on a distance function, that some new pattern, X, belongs to the same category as do its closest neighbors in E. More precisely a k-nearest neighbor method assigns a new pattern, X, to that category to which the plurality of its k closest neighbors belong. We used relative histogram deviation measure as a distance function for performing better that L_2 measure. The measure gives the deviation between two histograms as:

$$d_{rd}(H,H') = \frac{\sqrt{\sum_{m=1}^{M}(H_m - H'_m)^2}}{\frac{1}{2}\left(\sqrt{\sum_{m=1}^{M}H_m^2} + \sqrt{\sum_{m=1}^{M}H_m'^2}\right)} \tag{3-19}$$

Using relatively large values of k decreases the chance that the decision will be unduly influenced by a noisy training pattern close to X. But large values of k also reduce the acuity of the method. The k-nearest neighbor method can be thought of as estimating the values of the probabilities of the classes given X. Of course the denser are the points around X and larger the value of k the better the estimate. Cover and Hart theorem [154] related the performance of the 1-nearest neighbor method to the performance of a minimum probability-of-error classifier and also concluded that, for any number n of samples, the single-NN rule has strictly lower probability of error than any other k-NN rule.

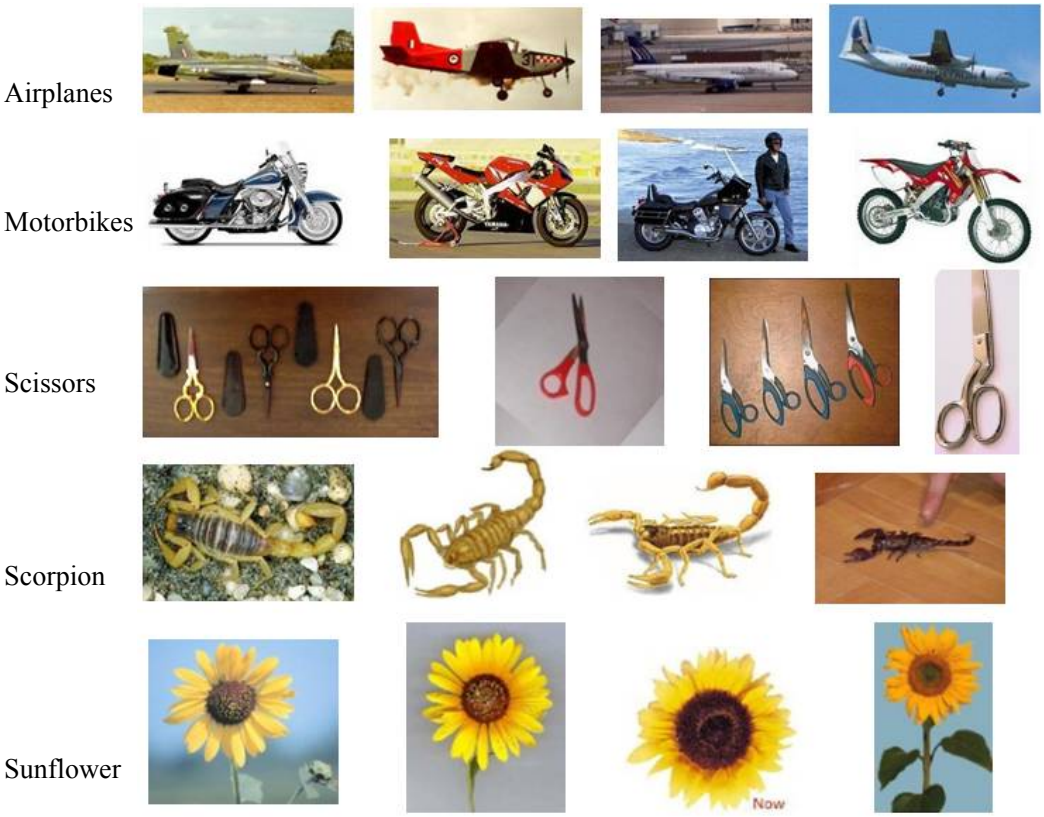


Figure 3.5 Few classes of the Caltech 101 database

a. Data set

For testing the algorithm we have used the Caltech 101 data set provided by 'The California Institute of Technology' (Caltech) for object class recognition. The Caltech 101 dataset contains 9,197 images comprising 101 different object categories. The dataset consists of pictures of objects belonging to 101 categories. There are about 40 to 800 images per category. Most categories have about 50 images. The size of each image is roughly 300 x 200 pixels. The data set is available on the institute's website [155].

Caltech 101 dataset is an extremely challenging dataset with large intra-class variation in color, pose and lighting. Secondly, a number of previously published papers have reported results on this data set. Figure 3.5 shows few classes from the dataset

b. Multiclass categorization task

We first discuss and compare our results with the published work using only the subset of Caltech 101 database. The table below shows results by Fergus et al. [156] and Fayin Li [157] along with our results in the similar way for comparisons.

Table 3.1 Classification results: Comparison with published results on subset of Caltech 101

Classes	Our Multi class	[156] Single class	[157] Multi class	[157] Single class
Airplanes	95.75	90.2	95.4	93.7
Faces	94.2	96.4	93.4	94.4
Motorbikes	95.3	92.5	93.1	96.1
Aver. Class	95.08	93.0	93.96	94.73

In [157] the authors additionally reported the class separation performance for visual object classes in the form of a confusion matrix. It is obvious that the latter approach is more challenging then a pure one-class problem. The results of [157] confirm that - for the class motorbikes - the classification rate dropped for about 3% between the one-class and multi-class problem. This result suggests that the inter-category separation is of a higher difficulty, but also gives a further insight into the discrimination ability of a feature.

Table 3.1 shows that our approach showed better results with the subset of the database compared to other methods. For the class faces our approach is slightly less better then the one in [156]. For the class motorbikes [157] reported a higher classification rate for the one-class approach. However, for the class separation task the performance drops below ours. The overall classification rate of our method is the highest with more than 95%. Better results are mainly because the semantic structures of these classes are very distinct from each other and can't be misjudged visually.

For comprehensive comparisons, we have shown results from published work on multiclass object categorization using whole of the Caltech 101 dataset. The algorithm was tested with the benchmark methodology of Grauman and Darrell [158], where a number (in this case 15 and 30) of images are taken from each class uniformly at random as the training image, and the rest of the data set is used as test set. The "mean recognition rate per class" is used so that more populous (and easier) classes are not favored. This process is repeated 10 times and the average correctness rate is reported.

The figure 3.6 above shows the number of training images per class on x-axis and mean recognition rate per class on y-axis. The best results on the Caltech 101 dataset for visual object class recognition have been published by Zhang et al. (2006) [159]. They have shown that a hybrid of SVM and KNN classifier has much better performance compared to all others. The work is a continuation of the previous work (2005) [161 and 162] as in this work, they have focused on improved classification using the same features. This brings forward the open question as to which classifier is the best with which features and distance functions. Figure 3.6 shows that our approach is comparable to other methods and has performed well above all except few.

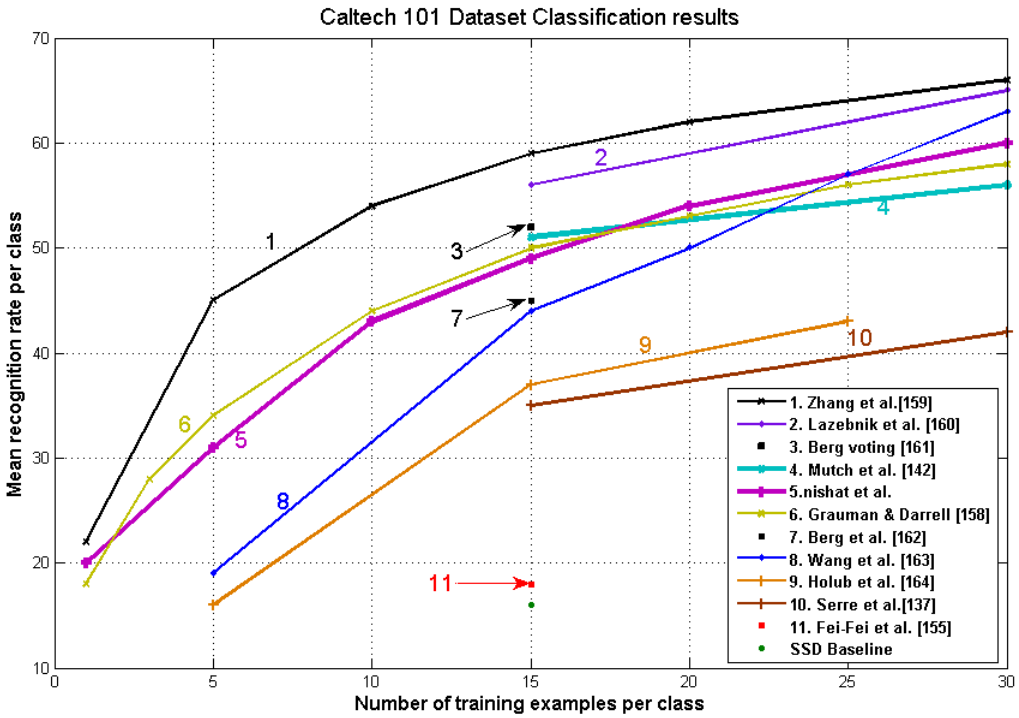


Figure 3.6 Classification rates Caltech 101 database

For the purpose of clarity, we have shown the published classification rates (correctness rates) using 15 and 30 training images per class in a tabular form in table 3-2. The blank cells indicate the unavailability of results in that category. Results for our algorithm are the average of 10 independent runs using all available test images. Scores shown are the average of the per-category classification rates.

When looking at the classification results of individual visual object categories, we find that our algorithm performed better for the classes which have distinctive semantic structure like airplane, motorbikes, grand piano, minaret, etc or represent coherent natural “scenes” (like

Joshua tree). Figure 3.7 shows some examples of categories for which the system performed well.

Table 3.2 Classification results: Comparison with published results using whole of Caltech 101

Model	15 training images/cat	30 training images/cat
Fei-Fei et al. [155]	18	
Serre et al.[137]	35	42
Holub et al. [164]	37	43
Berg et al. [162]	45	
Mutch and Low [142]	51	56
Grauman & Darrell [158]	50	58
Berg voting [161]	52	
Nishat and Park	**49**	**60**
Wang et al. [163]	44	63
Lazebnik et al. [160]	56	65
Zhang et al. [159]	59	66

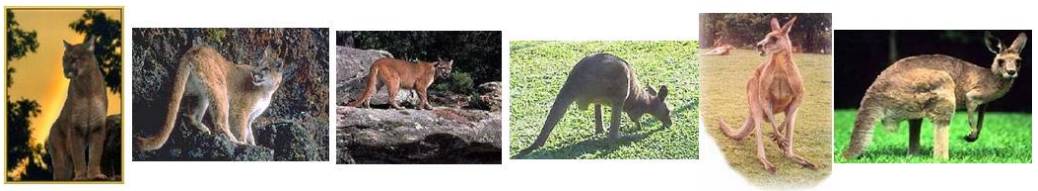
Compared to the easy categories, the categories which were difficult to categorize are those which are semantically more diverse, shown in figure 3.8 having greater shape variability due to greater intra-category variation and no-rigidity. The least successful classes are either textureless animals, or animals that camouflage well in their environment (like crocodile etc). Common misclassification errors has been shown in some works such as in [142 and 160], to understand the algorithms pattern of misclassification. Table 3.3 shows the most common classification errors found. A scrutiny of these errors shows that the misclassified objects have

structural similarities, which needs additional features to be considered. The most common confusions are schooner vs. ketch (both are sail boats with three or four sails, commonly indistinguishable by uninitiated) and lotus vs. water lily (both are almost similar flowers).

Grand Piano (93.7%) Inline skate (94.2%)

Ceiling fan (96.1%) Motorbikes (92.6%)

Figure 3.7 Caltech 101 dataset: Visual object classes on which the system performed better



Cougar body (24.9%) Kangaroo (24.2%)

Wild cat (23.7%) Crocodile (28.3%)

Figure 3.8 Caltech 101 dataset: Visual object classes on which the system performed poor

Table 3.3 Most common misclassification errors on the Caltech 101 dataset

Visual Object class1/class2	Class 1 misclassified as class 2	Class 2 misclassified as class 1
ketch/schooner	20.6	18.1
lotus/water lily	17.2	19.3
cougar body/wild cat	14.7	17.2
Ibis/flamingo	11.4	8.6
crayfish/lobster	9.3	8.9

G. Semantic group formation and Graph Modeling

Using equations (3-7 ~ 3-10), we can find the coordinates of the minimum distance points on the respective lines which are closest. From these points we can find the distance between them and the center point of the line joining them. All the line segments which are within the minimum distance threshold and whose center points of the line joining the minimum distance points are also within a defined distance threshold are grouped together as a semantic group. This can be understood from figure 3.9, SG 1 ~ 3 are three semantic groups and the mean center point of the group is shown as a black dot. The relation R1 and R2 are the line segments which are at a minimum distance with two groups and hence joining the two groups. The relations R1 and R2 are common members of both the semantic groups they are joining. This way we have transformed an image object into a relational structure of semantic groups and their inter linking. The relational structure obtained is shown in the tabular form as table 3.4 below, which can easily be represented as a linked list. In order to use this structure for identification purposes, we have to compute some of its properties and associate these with

individual semantic groups and relations. The only image information considered here is the extracted line segments. Since the line angles and lengths are subject to change under various transformations, we consider the effect of transformations and try to minimize or eliminate it.

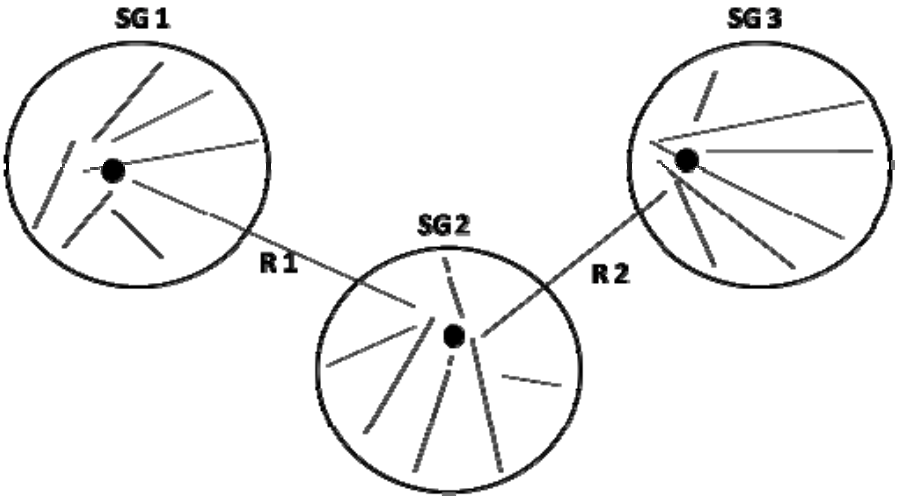


Figure 3.9 Semantic structures and their relations

Since Euclidean Distance Matrix (EDM) is invariant to rotation, translation and scaling (equation 3-13), we compute EDM's from the geometric properties of the line segments of each semantic group.

Table 3.4 Layout of semantic structures and their relations

Semantic group - ID	link - ID	Linked group
SG 1	R 1	SG 2
SG 2	R 1 R 2	SG 1 SG 3
SG 3	R 2	SG 2

For each semantic group, let $L = \{l_i \mid i = 1, 2, ..., N\}$, be the set of line segments obtained. Then we can compute geometric properties of L: the angles formed by all segments between each other and the relative length of each segment with respect to all other line segments. For the segment joining two semantic groups, we computed its geometric properties with respect to each group member of all the linked groups (equations 3-14 ~ 3-17).

Since every EDM is symmetric, we extract the upper triangle matrix and form a histogram from each EDM with different resolutions based on empirical testing (equation 3-18). For H_{ang}, 72 bins that correspond to a 5 degree angle resolution were used. The resolution for H_{len} was taken as 101 bins through experimentation. As shown in the below figure, for each semantic group we get two histograms and for each relation we get four histograms, two from each group. The histograms of the semantic groups are obtained from the EDM of the group members and the histograms of the relations are the relative difference of length and angles of the R1 segment with the rest of the respective group members.

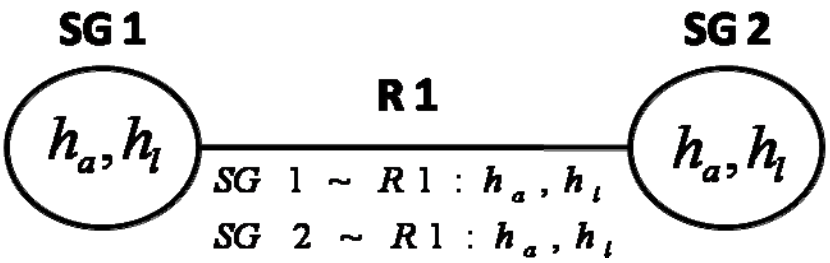


Figure 3.10 Properties of semantic groups and relations

Now we have a relational structure consisting of semantic groups with distinct properties and their inter-relations, also with distinct properties. This way we can simply define an image object in terms of its lower level semantic groups and their interrelations. Further we analyze

the relational structure and introduce another property of frequency for repetitive semantic groups using a distance function (histogram deviation measure, equation 3-19). We combine the semantic groups which are exactly same and combine their relations with other nodes, under one semantic node. So, in the relational structure of an image object, the semantic groups or nodes are unique.

The relational structure can easily be represented in the form of a 4-tuple labeled graph g as:

$$g = (V, E, \alpha, \beta) \tag{3-20}$$

where : - L denotes the finite set of labels for nodes

- f_v denotes the finite set of node frequencies
- M denotes the finite set of edge property labels for edges
- V is the finite set of vertices (semantic nodes)
- $E \subseteq V \times V$ is the set of edges (semantic relations)
- $\alpha : V \rightarrow L, f_v$ is a function assigning labels and frequency to the vertices
- $\beta : E \rightarrow M$ is a function assigning labels to the edges (semantic relation properties)

1. Graph model for classification

Since there is quite a considerable amount of variability at a structure level between the objects of the same semantic category, a graph model should take into account the commonality and variability in the semantic structures of the objects from same visual class. Going back to the basic argument of the idea that semantic objects are a combination of micro level semantic groups and their relations, we chart these factors for building a graph model. We build a graph model by iteratively merging the graphs of the test dataset and counting the frequency of the recurring semantic groups and relations. Groups and relations below a threshold are considered not essential in basic semantic labeling and are dropped. The resulting model graph is quite

small and is neither a subset, nor a super set of any image graph in the test data set and captures the variability in all the test samples. It contains the set of those semantic groups and relations which are common in at least few test sets. Retention of semantic groups and relations which are common over a spectrum of test samples from the same object class can be considered as a basic semantic skeletal structure which is essential to identify an object.

For building a graph model we use a general relational structure matching approach which is less restricted than graph isomorphism, because nodes or edges may be missing from one or the other graph. Also, it is more general than sub-graph isomorphism because one structure may not be exactly isomorphic to a substructure of the other. A more general match consists of a set of nodes from one structure and a set of nodes from the other and a 1:1 mapping between them which preserves the compatibilities of properties and relations. In other words, corresponding nodes (under the node mapping) have sufficiently similar properties, and corresponding sets under the mapping have compatible relations. We use association graph techniques of general relational structure matching [177] to build a graph model encompassing similarities and variability in visual object classes.

Given two structures $g_1 = (V_1, E_1, \alpha_1, \beta_1)$ and $g_2 = (V_2, E_2, \alpha_2, \beta_2)$. For each v_1 in V_1 and v_2 in V_2, construct a node of associated graph *'g'* labeled (v_1, v_2) if v_1 and v_2 have the same properties $[\alpha_1(v_1, L) \;\; iff \;\; \alpha_2(v_2, L) \;\; \forall v_1, v_2]$. Thus the nodes of G denote assignments, or pairs of nodes, one each from V_1 and V_2, which have similar properties. Now connect two nodes (v_1, v_2) and (v'_1, v'_2) of *'g'* if they represent compatible relations, that is, if the pairs satisfy the same relations $[\beta_1(e_1) \;\; iff \;\; \beta_2(e_2) \;\; \forall e_1, e_2]$. The properties can be compared using a suitable distance function (histogram deviation measure). A match between the two relational structures is just a set of assignments that are mutually compatible.

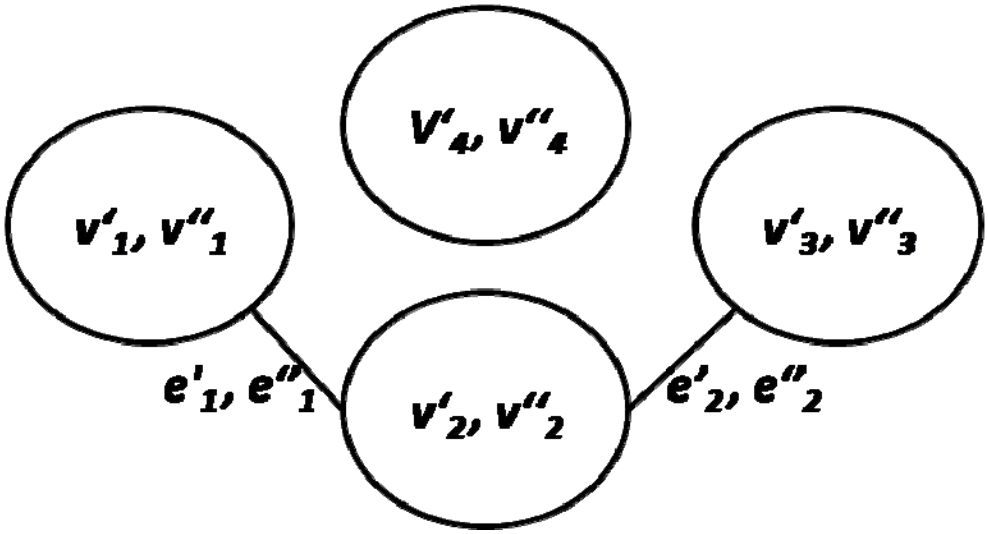


Figure 3.11 Association graph

In the above figure, three nodes of the association graph are linked as they have compatible relations and the 4th node (v'_4, v''_4) has no compatible edges with any other node pair. The association graph $g = (V, E, \alpha, \beta)$ obtained this way, gives us the similarity pattern between two object images of the same visual class.

Here we take the union of the two relational structures and add two more properties using the association graph, for intergroup frequency count of nodes (semantic groups) and edges (semantic relations). These labels show us the importance of a semantic group in the overall structure formation of that object class and also in deciding which semantic group would most likely be part of which object class in case there is a tie in the classification task. The union of the two structures is calculated as follows:

We have graphs *'g_1'*, *'g_2'* and the association graph *'g'*, such that $g \subseteq g_1 and g \subseteq g_2$. The difference of $g_1 - g$ and $g_2 - g$ is a graph g'= *(V', E', α', β')* and g''= *(V'', E'', α'', β'')*, where

$V' = V_1 - V$ $\quad\quad$ $V'' = V_2 - V$

$E' = E_1 \cap (V' \times V')$ $\quad\quad$ $E'' = E_2 \cap (V'' \times V'')$

$\alpha'(v') = \alpha_1 (v')$ *for any* $v' \in V'$ $\quad\quad$ $\alpha''(v'') = \alpha_2 (v'')$ *for any* $v'' \in V''$

$\beta'(e') = \beta_1 (v')$ *for any* $v' \in V'$ $\quad\quad$ $\beta''(e') = \beta_2 (v'')$ *for any* $v'' \in V''$

The difference graphs g' and g'' above are obtained by removing the sub-graph g from g_1 and from g_2, including the edges that connect g_1 and g_2 with the rest of the graph. These edges are removed by finding the embedding of g with g_1 and g_2. The embedding of g in g_1 and g_2, $emb(g, g_1)$ and $emb(g, g_2)$ is the set of edges that connects g with $g_1 - g$ and $g_2 - g$.

$$g_{emb} = emb(g, g_1) = E_1 \cap [(V \times (V_1 - V) \cup ((V_1 - V) \times V)] \qquad (3\text{-}21)$$

Where $\beta (e_{emb}) = \beta_1 (e_{emb})$ for any $e_{emb} \in emb (g, g1)$.

From the association graph $_c$we can count the inter-object frequency of the semantic group and relations and add two more labels f_g *and* f_l , such that:

- f_g denotes the finite set of group frequencies
- f_l denotes the finite set of edge (link) frequencies
- $\alpha : V \to L, f_g$ is a function assigning labels and frequency to the vertices
- $\beta : E \to M, f_l$ is a function assigning labels to the edges (semantic relation properties)

Now we have graphs $g' = (V', E', \alpha', \beta')$, $g'' = (V'', E'', \alpha'', \beta'')$ and $g = (V, E, \alpha, \beta)$ with $V' \cap V'' \cap V = \Phi$. We find the union graph $G = ((g' \cup g'') \cup g)$. Let $\overline{E} \subseteq (V' \times V'') \cup (V'' \times V')$ is a set of edges with labeling function $\overline{\beta} : \overline{E} \to M$. The union of g' and g'' is the graph $g''' (V''', E''', \alpha''', \beta''')$ where

$$V''' = V' \bigcup V''$$

$$E''' = E' \bigcup E'' \bigcup \overline{E}$$

$$\alpha'''(v''') = \begin{Bmatrix} \alpha'(v''') < if > v''' \in V' \\ \alpha''(v''') < if > v''' \in V'' \end{Bmatrix}$$

$$\beta'''(e''') = \begin{Bmatrix} \beta'(e''') < if > e''' \in E' \\ \beta''(e''') < if > e''' \in E'' \\ \overline{\beta}(e''') < if > e''' \in \overline{E} \end{Bmatrix}$$

We repeat the process iteratively for a set of training images. The semantic groups and relations which are essential to give semantic meanings to the visual object will have intra-object frequency greater than a threshold '*t*'. So, all the semantic groups with intra-object frequency $f_g \leq t$ are considered redundant and are removed along with any edges they have with other nodes. The resulting model graph contains the repetitive patterns in a training set of images containing a visual object class. The removal of redundant groups reduced the model size to a considerably small level. Using the same procedure, we build up relational structure models for all the visual object classes we want to test.

2. Classification steps

The classification task has been reduced to the graph matching between the model graph and the query graph. We constructed graph models of all the object classes in the test data set using 15 and 30 training images, selected randomly. The remaining images form the test data set. For the purpose of matching the query and model graph we used the association graph technique [166] and constructed a relational graph from the query and model graphs. We used relative histogram deviation measure (equation 3-19) as a distance function for building nodes and arcs of the relational graph. From the relational graph we find the maximum cliques in the graph

[166]. The decision is based on the voting by each model based on the maximum cliques as follows.

$$vote = \sum_{i=1}^{n} a \times x \tag{3-22}$$

Where, $a = \{2, 3, 4...\}$ are the *a-clicks*, $i = \{1, 2, 3 \ldots n\}$ is the number of total click counts and $x = \{1, 2, 3 \ldots\}$ is the frequency of an instance of *a-click*. In case of a tie, node and edge frequencies in the model graph are used as an additional vote for the nodes in the cliques. The vote for a node in case of a tie is calculated as: $node \quad vote = f_g \times f_n$ Where as, f_g is inter-object node frequency from the model and f_n is the node frequency from the query image. Final vote is formulated by counting the maximum number of node votes.

3. Experiments and results

For testing the algorithm we have used the Caltech 101 data set as a number of previously published papers have reported results on this data set, thereby making comparisons more meaningful. In literature multiclass object categorization has been dealt in a less frequency. Many authors have reported the classification rates of their algorithms on a subset of the data and on class-wise classification methodologies, i.e. a classifier was trained in order to discriminate a single class among the subset from a background class consisting of arbitrary images. For comprehensive comparisons, we have shown results from published work on multiclass object categorization using whole of the Caltech 101 dataset. The algorithm was tested with the benchmark methodology of [158], where a number (in this case 15 and 30) of images are taken from each class uniformly at random as the training image, and the rest of the data set is used as test set. The “mean recognition rate per class” is used so that more populous

(and easier) classes are not favored. This process is repeated 10 times and the average correctness rate is reported.

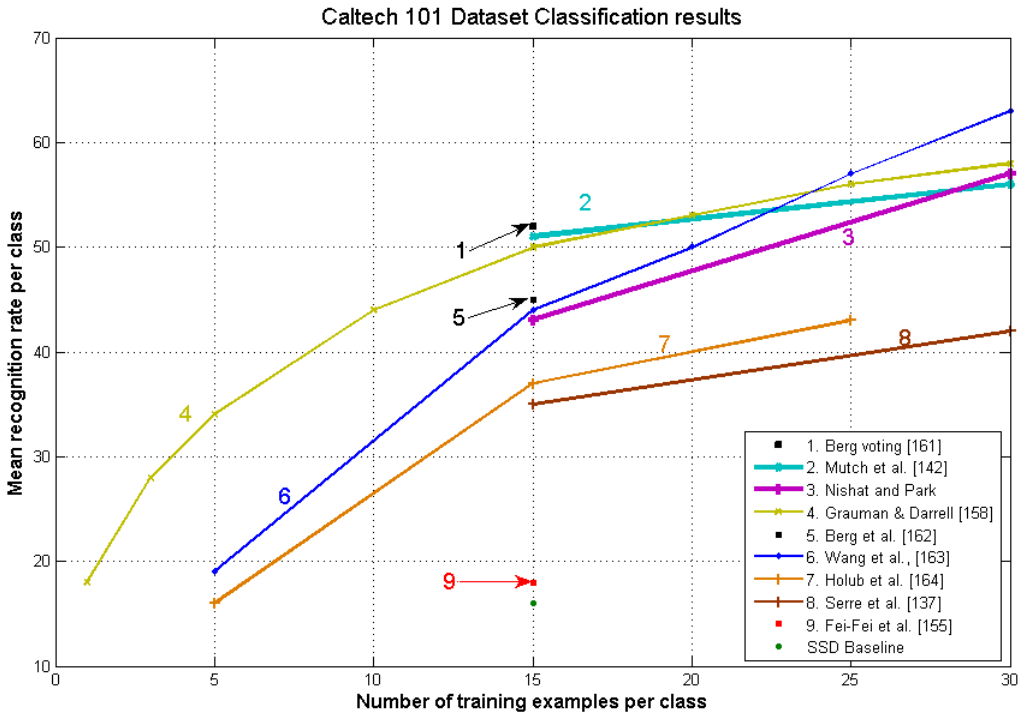


Figure 3.12 Classification rates Caltech 101 database

In figure 3.12, number of training images per class is shown on x-axis and mean recognition rate per class on y-axis. The figure shows that our approach is comparable to other methods and has performed well above all except few. For the purpose of clarity, we have shown the published classification rates (correctness rates) using 15 and 30 training images per class, in table 3.5. The blank cells indicate the unavailability of results in that category. Results for our algorithm are the average of 10 independent runs using all available test images. Scores shown are the average of the per-category classification rates.

Table 3.5 Classification results: Comparison with published results using whole of Caltech 101

Model	15 training images/cat	30 training images/cat
Fei-Fei et al. [155]	18	--
Serre et al.[137]	35	42
Holub et al. [164]	37	43
Berg et al. [162]	45	--
Mutch et al. [142]	51	56
Nishat and park	**43**	**57**
Grauman & Darrell [158]	50	58
Berg voting [161]	52	--
Wang et al. [163]	44	63

When looking at the classification results of individual visual object categories, we find that our algorithm performed better for the classes which have distinctive semantic structure like airplane, motorbikes, grand piano, minaret, etc. The categories which were difficult to categorize are semantically more diverse, having greater shape variability due to greater intra-category variation and no-rigidity. A scrutiny of misclassification errors show that the misclassified objects have structural similarities, which needs additional features to be considered. The most common confusions are schooner vs. ketch (both are sail boats with three or four sails, commonly indistinguishable by uninitiated) and lotus vs. water lily (both are almost similar flowers).

H. Conclusion and future work

The field of Content Based Image Retrieval (CBIR) has evolved very quickly due to the rapid advancement in technology, making possible unmanageable collections of image and multimedia data. The emphasis in future will be to make the CBIR systems more and more intelligent, mimicking human vision and intelligence. In this thesis work, effort has been made to understand the underlying principles of human vision perception and explore them to make the computer vision systems more intelligent in the task of image retrieval. David Marr wrote, "The true heart of visual perception is the inference from the structure of an image about the structure of the real world outside" [147]. This is the main objective of this thesis, to be able to infer a real world object from the structure of an image.

The thesis explores basic level of semantic structure formation in the human vision inferential processes in line with Gestalt laws and proposes micro level semantic structure formations and their relational combinations. Using this approach two sets of semantic features have been derived for visual object class recognition. The first algorithm uses the hypothesis in line with Gestalt laws of proximity that; in an image, basic semantic structures are formed by line segments (arcs also approximated and broken into smaller line segments based on pixel deviation threshold) which are in close proximity of each other. Based on the notion of proximity a transitive relation is defined, which combines basic micro level semantic structures hierarchically till such a point where semantic meanings of the structure can be extracted. The algorithm extracts line segments in an image and then forms semantic groups of these line segments based on a minimum distance threshold from each other. The line segment groups so formed can be differentiated from each other, by the number of group members and their geometrical properties. The geometrical properties of these semantic groups are used to

generate rotation, translation and scale invariant histograms used as feature vectors for object class recognition tasks in a K-nearest neighbor framework.

In the second approach a semantic group based on the proximity distance is clustered and modeled as a graph vertex. The line segments which are common to more than one semantic group are defined as semantic relations between the semantic groups and are modeled as edges of the graph. This way an image object is transformed into a graph using micro level structure formations. Each vertex and edge is labeled using translation, rotation and scale invariant properties of the member segments of each vertex and edge. From a set of training images, a graph model is constructed for visual object class recognition. The graph model is constructed by iteratively combining the training graphs and frequency labeling the vertices and edges. After the combining phase, all the vertices and edges whose repetition frequency is below a threshold are removed. The final graph model consists of the semantic nodes which are highly common in the training images. The recognition is based on graph matching the query image graph and the model graph. The model graph generates a vote for the query and ties are resolved by considering the node frequencies in the query and model graph.

The algorithms have been applied to classify 101 object classes at one time. The results have been compared with existing state of the art approaches and are found promising. Results from above approaches show that low level image structure and other features can be used to construct different type of semantic features, which can help a model or a classifier make more intelligent decisions and work more effectively for the task compared to low level features alone. Our experimental results are comparable, or outperform other state-of-the-art approaches. We have also summarized the state-of-the-art at the time this work was finished. We conclude with a discussion about the possible future extensions.

For the semantic hierarchical relational features, the most important highlight of the comparisons is the choice of a classifier for the object categorization task. Boiman et al. (2008) [165] and Zhang et al. (2006) [159] have proposed to use modified or hybrid versions of knn classifier for better performance. In the future work we would like to test and improve the algorithm performance with modified and improved classifiers and incorporate additional features to reduce the classification confusion further down. In case of Graph model, we would like to test and improve the algorithm performance with modified and improved models and incorporate additional features to reduce the classification confusion further down. Since color and texture forms very important components in recognition; their inclusion into the proposed features in a semantic perspective can further improve the performance in recognition.

References

1. Blaser, Database Techniques for Pictorial Applications, Lecture Notes in Computer Science, Volume 81, Springer Verlag, 1979.
2. N. S. Chang and K. S. Fu, A relational database system for images, Lecture Notes in Computer Science, Volume 80, Springer Verlag, 1979.
3. N. S. Chang, and K. S. Fu, Query by pictorial example, IEEE Trans. on Software Engineering, Volume 6, Issue 6, pages 519-524, 1980.
4. S. K. Chang, and T. L. Kunii, Pictorial database systems, IEEE Computer Magazine, Volume 14, Issue 11, pages 13-21, 1981.
5. S. K. Chang, C. W. Yan, D. C. Dimitroff, and T. Arndt, An intelligent image database system, IEEE Trans. on Software Engineering, Volume 14, Issue 5, pages 681-688, 1988.
6. S. K. Chang, and A. Hsu, Image information systems: where do we go from here? IEEE Trans. on Knowledge and Data Engineering, Volume 5, Issue 5, pages 431-442, 1992.
7. H. Tamura, and N.Yokoya, Image database systems: A survey, Pattern Recognition, Volume 17, Issue 1, pages 29-43, 1984.
8. R. Jain (Ed.), Proc. US NSF Workshop Visual Information Management Systems, 1992.

9. T. Kato, Database architecture for content-based image retrieval, in: A.A. Jambardino, W.R. Niblack (Eds.), Image Storage and Retrieval Systems, Proc. SPIE Volume 1662, pages 112-123, 1992.

10. W.R. Niblack et al., The QBIC project: querying images by color, texture and shape, Proc. SPIE, Volume 1908, pages 173-187, 1993.

11. Pentland et al., Photobook: tools for content-based manipulation of image databases, Storage and Retrieval for Image and Video Databases II, Proc. SPIE 2185, pages 34-47, 1994.

12. E. Cawkill, The British Library's Picture Research Projects: Image, Word, and Retrieval, Advanced Imaging, Volume 8, Issue 10, pages 38-40, 1993.

13. J. Dowe, Content-based retrieval in multimedia imaging, Proc. SPIE Storage and Retrieval for Image and Video Database, 1993.

14. Faloutsos et al, Efficient and effective querying by image content, Journal of intelligent information systems, Volume 3, Issue 3/4, pages 231-262, 1994.

15. Y. Gong, H. J. Zhang, and T. C. Chua, An image database system with content capturing and fast image indexing abilities, IEEE International Conference on Multimedia Computing and Systems, pages 121-130, 1994.

16. R. Jain, A. Pentland, and D. Petkovic (Eds.), Workshop Report: NSF-ARPA Workshop on Visual Information Management Systems, Cambridge, Mass, USA, June 1995.

17. H. J. Zhang, and D. Zhong, A Scheme for visual feature-based image indexing, Storage and Retrieval for Image and Video Databases III, Proc. SPIE Volume 2420, pages 36-46, 1995.

18. Furht, S. W. Smoliar, and H.J. Zhang, Video and Image Processing in Multimedia Systems, Kluwer Academic Publishers, 1995.

19. Y. Rui, T. S. Huang, and S. F. Chang, Image retrieval: current techniques, promising directions and open issues, Journal of Visual Communication and Image Representation, Volume 10, pages 39-62, 1999.

20. M. W. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, Content-based image retrieval at the end of the early years, IEEE Trans. on Pattern Analysis and Machine Intelligence, Volume 22, Issue 12, pages 1349-1380, 2000.

21. H. Burkhardt, and S. Siggelkow, Invariant features for discriminating between equivalence classes, Nonlinear Model-based Image Video Processing and Analysis, John Wiley and Sons, 2000.

22. J. D. Foley, A. van Dam, S. K. Feiner, and J. F. Hughes, Computer graphics: principles and practice, 2nd ed., Reading, Mass, Addison-Wesley, 1990.

23. J. Huang, S.R. Kumar, M. Metra, W. J., Zhu, and R. Zabith, Spatial color indexing and applications, International Journal of Computer Vision, Volume 35, Issue 3, pages 245-268, 1999.

24. J. Huang, et al., Image indexing using color correlogram, IEEE Int. Conf. on Computer Vision and Pattern Recognition, pages 762-768, 1997.

25. M. Ioka, A method of defining the similarity of images on the basis of color information, Technical Report RT-0030, IBM Tokyo Research Laboratory, Tokyo, Japan, Nov. 1989.

26. K. Jain, Fundamental of Digital Image Processing, Englewood Cliffs, Prentice Hall, 1989.

27. Mathias, Comparing the influence of color spaces and metrics in content-based image retrieval, Proc. International Symposium on Computer Graphics, Image Processing, and Vision (published by IEEE press), pages 371-378, 1998.

28. G. Pass, and R. Zabith, Comparing images using joint histograms, Multimedia Systems, Volume 7, pages 234-240, 1999.

29. M. Stricker, and M. Orengo, Similarity of color images, Storage and Retrieval for Image and Video Databases III, Proc. SPIE, Volume 2185, pages 381-392, 1995.

30. M. J. Swain, and D. H. Ballard, Color indexing, International Journal of Computer Vision, Volume 7, Issue 1, pages 11-32, 1991.

31. H. J. Zhang, et al, Image retrieval based on color features: An evaluation study, Conf. on Digital Storage and Archival, Proc. SPIE, Volume 2606, 1995.

32. M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker, Query by image and video content: The QBIC system, IEEE Computer, Volume 28, Issue 9, pages 23-32, 1995.

33. W. Niblack et al., Querying images by content, using color, texture, and shape, SPIE Conference on Storage and Retrieval for Image and Video Database, Volume 1908, pages 173-187, 1993.

34. G. Pass, and R. Zabith, Histogram refinement for content-based image retrieval, IEEE Workshop on Applications of Computer Vision, pages 96-102, 1996.

35. T. Gevers, and A.W.M.Smeulders, Pictoseek: Combining color and shape invariant features for image retrieval, IEEE Trans. on image processing, Volume 9, Issue 1, pages 102-119, 2000.

36. G. D. Finlayson, Color in perspective, IEEE Trans on Pattern Analysis and Machine Intelligence, Volume 8, Issue 10, pages 1034-1038, 1996.

37. T. Gevers, and A. W. M. Smeulders, Content-based image retrieval by viewpoint-invariant image indexing, Image and Vision Computing, Volume 17, Issue, pages 475-488, 1999.

38. P. Brodatz, Textures: A photographic album for artists & designers, Dover, NY, 1966.

39. T. Chang, and C.C.J. Kuo, Texture analysis and classification with tree-structured wavelet transform, IEEE Trans. on Image Processing, Volume 2, Issue 4, pages 429-441, 1993.

40. Daubechies, The wavelet transform, time-frequency localization and signal analysis, IEEE Trans. on Information Theory, Volume 36, pages 961-1005, 1990.

41. J. M. Francos. Orthogonal decompositions of 2D random fields and their applications in 2D spectral estimation, invited chapter, Signal Processing and Its Applications Volume, Handbook of Statistics , N. K. Bose and C. R. Rao (eds.), North-Holland Publishing Comp., pages 207-227, 1993.

42. J. M. Francos, A. A. Meiri, and B. Porat, A unified texture model based on a 2D Wold like decomposition, IEEE Trans. on Signal Processing, pages 2665-2678, 1993.

43. J. M. Francos, A. Narasimhan, and J. W. Woods, Maximum likelihood parameter estimation of textures using a Wold-decomposition based model, IEEE Trans. on Image Processing, pages 1655-1666, 1995.

44. A. K. Jain, and F. Farroknia, Unsupervised texture segmentation using Gabor filters, Pattern Recognition, Volume 24, Issue 12, pages 1167-1186, 1991.

45. Kankanhalli, H. J. Zhang, and C. Y. Low, Using texture for image retrieval, Proc. Third IEEE Conf. on Automation, Robotics and Computer Vision (ICARCV94), pages 935-939, 1994.

46. W. J. Krzanowski, Recent Advances in Descriptive Multivariate Analysis, Chapter 2, Oxford science publications, 1995.

47. Laine, and J. Fan, Texture classification by wavelet packet signatures, IEEE Trans. Pattern Analysis and Machine Intelligence, Volume 15, Issue 11, pages 1186-1191, 1993.

48. Liu, and R. W. Picard, Periodicity, directionality, and randomness: Wold features for image modeling and retrieval, IEEE Trans. on Pattern Analysis and Machine Learning, Volume 18, Issue 7, 1996.

49. W. Y. Ma, and B. S. Manjunath, A comparison of wavelet features for texture annotation, Proc. Of IEEE Int. Conf. on Image Processing, Volume II, pages 256-259, 1995.

50. S. G. Mallat, A theory for multiresolution signal decomposition: the wavelet representation, IEEE Trans. Pattern Analysis and Machine Intelligence, Volume 11, pages 674-693, 1989.

51. S. Manjunath, and W. Y. Ma, Texture features for browsing and retrieval of image data, IEEE Trans. on Pattern Analysis and Machine Intelligence, Volume 18, Issue 8, pages 837-842, 1996.

52. J. Mao, and A. K. Jain, Texture classification and segmentation using multi-resolution simultaneous autoregressive models, Pattern Recognition, Volume 25, Issue 2, pages 173-188, 1992.

53. T. Ojala, M. Pietikainen, and D. Harwood, A comparative study of texture measures with classification based feature distributions, Pattern Recognition, Volume 29, Issue1, pages 51-59, 1996.

54. R. W. Picard, T. Kabir, and F. Liu, Real-time recognition with the entire Brodatz texture database, Proc. IEEE Int. Conf. on Computer Vision and Pattern Recognition, pages 638-639, 1993.

55. H. Tamura, S. Mori, and T. Yamawaki, Texture features corresponding to visual perception, IEEE Trans. On Systems, Man, and Cybernetics, Volume 8, Issue 6, 1978.

56. H. Voorhees, and T. Poggio. Computing texture boundaries from images, Nature Volume 333, pages 364-367, 1988.

57. Pentland, R.W. Picard and S. Sclaroff, Photobook: Content-Based Manipulation of Image Databases, Proc. Storage and Retrieval for Image and Video Databases II, Volume 2185, 1994.

58. J. G. Daugman, Complete discrete 2D Gabor transforms by neural networks for image analysis and compression, IEEE Trans. on Acoustics, Speech and Signal Processing (ASSP), Volume 36, Issue 7, pages 1169-1179, 1998.

59. J. E. Gary, and R. Mehrotra, Shape similarity-based retrieval in image database systems, Proc. Of SPIE, Image Storage and Retrieval Systems, Volume 1662, pages 2-8, 1992.

60. W. I. Grosky, and R. Mehrotra, Index based object recognition in pictorial data management, Computer vision, graphics, and image processing (CVGIP), Volume 52, Issue 3, pages 416-436, 1990.

61. H. V. Jagadish, A retrieval technique for similar shapes, Proc. of Int. Conf. on Management of Data, SIGMOID'91, pages 208-217, 1991.

62. Tegolo, Shape analysis for image retrieval, Proc. of SPIE, Storage and Retrieval for Image and Video Databases -II, Volume 2185, Carlton W. Niblack; Ramesh C. Jain, Editors, pages 59-69, 1994.

63. M. Arkin, L.P. Chew, D.P. Huttenlocher, K. Kedem, and J.S.B. Mitchell, An efficiently computable metric for comparing polygonal shapes, IEEE Trans. Pattern Analysis and Machine Intelligence, Volume 13, Issue 3, pages 209-226, 1991.

64. S. Sclaroff, and A. Pentland, Modal matching for correspondence and recognition, IEEE Trans. on Pattern Analysis and Machine Intelligence, Volume 17, Issue 6, pages 545-561, 1995.

65. K. Arbter, W. E. Snyder, H. Burkhardt, and G. Hirzinger, Application of affine-invariant Fourier descriptors to recognition of 3D objects, IEEE Trans. Pattern Analysis and Machine Intelligence, Volume 12, pages 640-647, 1990.

66. H. Kauppinen, T. Seppnäen, and M. Pietikäinen, An experimental comparison of autoregressive and Fourier-based descriptors in 2D shape classification, IEEE Trans. Pattern Analysis and Machine Intelligence, Volume 17, Issue 2, pages 201-207, 1995.

67. Persoon, and K. Fu, Shape discrimination using Fourier descriptors, IEEE Trans. Syst., Man, and Cybern., Volume 7, pages 170-179, 1977.

68. M. K. Hu, Visual pattern recognition by moment invariants, in J. K. Aggarwal, R. O. Duda, and A. Rosenfeld (Eds.), Computer Methods in Image Analysis, IEEE computer Society, 1977.

69. L. Yang, and F. Algregtsen, Fast computation of invariant geometric moments: A new method giving correct results, Proc. IEEE Int. Conf. on Image Processing, 1994.

70. R. C. Veltkamp, and M. Hagedoorn, State-of-the-art in shape matching, Technical Report UU-CS-1999-27, Utrecht University, Department of Computer Science, Sept. 1999.

71. S. K. Chang, Q. Y. Shi, and C. Y. Yan, Iconic indexing by 2-D strings, IEEE Trans. on Pattern Analysis and Machine Intelligence, Volume 9, Issue 3, pages 413-428, 1987.

72. S. K. Chang, E. Jungert, and Y. Li, Representation and retrieval of symbolic pictures using generalized 2D string, Technical Report, University of Pittsburgh, 1988.

73. S. Y. Lee, and F. H. Hsu, 2D C-string: a new spatial knowledge representation for image database systems, Pattern Recognition, Volume 23, pages 1077-1087, 1990.

74. S. Y. Lee, M.C. Yang, and J. W. Chen, 2D B-string: a spatial knowledge representation for image database system, Proc. ICSC'92 Second Int. computer Sci. Conf., pages 609-615, 1992.

75. V.N. Gudivada, V.V. Raghavan, Content-based image retrieval systems, IEEE Computer Volume 28 No. 9, pages 18-22, 1995.

76. J.P. Eakins, M.E Graham, Content-based image retrieval, JISC Technology Applications Programme Report 39, October 1999.

77. L. Armitage, P.G.B. Enser, Analysis of user need in image archives, Journal of Information Science, Volume 23, No. 4, pages 287-299, 1997.

78. R.A. Brooks et al., Alternative essences of intelligence, Proc. 15th National Conference on Artificial Intelligence (AAAI-98), Publisher Association for the Advancement of Artificial Intelligence (AAAI), pages 961-968, 1998.

79. Rosch et al., Basic objects in natural categories, Cognitive Psychology Volume 8, Issue 3, pages 291 -440, 1976.

80. D.A. Forsyth et al., Finding pictures of objects in large collections of images, Proc. Int. workshop on Object Representation in Computer Vision II, Lecture Notes In Computer Science; Volume 1144, pages: 335 – 360, 1996.

81. J.M. Buijs, M.S. Lew, Visual learning of simple semantics in ImageScape, Proc. Third Int. Conference on Visual Information and Information Systems, Lecture Notes In Computer Science, Volume 1614, pages 131-138, 1999.

82. Liyuan Li and Weinan Chen, Corner detection and interpolation on planar curves using fuzzy reasoning, IEEE Trans. on Pattern Analysis and Machine Intelligence, Volume 21, Issue 11, November, 1999.

83. Hans Moravec, Towards Automatic Visual Obstacle Avoidance, Proc. 5th International Joint Conference on Artificial Intelligence, page 584 August, 1977.

84. Harris, C. & Stephens, M., A combined corner and edge detector, Proc. 4th Alvey Vision Conf, pages 147 – 151, 1988.

85. P. R. Beaudet, Rotationally invariant image operators, Proc. 4th Int'l Joint Conf on Pattern Recognition, pages 579-583, Nov 1978.

86. L. Kitchen and A. Rosenfeld, Gray level corner detection, Pattern Recognition Letters Volume 1, Issue 2, pages 95-102, 1982.

87. K. K. Lai and P. S. Y. Wu, Effective edge-corner detection method for defected images, Proc. 3rd Int'l Conf. on Signal Processing, Volume 2, pages 1151-1154, 1996.

88. D. M. Tsai, Boundary based corner detection using neural networks, Pattern Recognition, Volume 30, Issue 1, pages 85-97, 1997.

89. Q. Ji and R. M. Haralick, Corner detection with covariance propagation, Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR97), pages 362-367, 1997.

90. K. J. Lee and Z. Bien, Grey-level corner detector using fuzzy logic, Pattern Recognition Letters, Volume 17, Issue 9, pages 939-950, 1996.

91. J. Q. Fang and T. S. Huang, A corner finding algorithm for image analysis and registration, Proc. AAAI (Association for the Advancement of Artificial Intelligence) Conf., pages 46-49, 1982.

92. W. C. Chen and P. Rockett, Bayesian labeling of corners using a grey-level corner image model, Proc. IEEE Int'l Conf on Image Processing (ICIP 97), Volume. 1, pages 687-690, 1997.

93. Z. O. Wu and A. Rosenfeld, Filtered projections as an aid to corner detection, Pattern Recognition, Volume 16, Issue 31, 1983.

94. K. Paler, J. Foglein, J. Illingworth, and J. Kittler, Local ordered grey levels as an aid to corner detection, Pattern Recognition, Volume 17, Issue 5, pages 535-543, 1984.

95. K. Rangarajan, M. Shah, and D. Van Brackle, Optimal corner detector, Journal of Computer Vision, Graphics, and Image Processing, Volume 48, Issue 2, pages 230-245, 1989.

96. Arrebola, A. Bandera, P. Camacho, and F. Sandoval, Corner detection by local histograms of contour chain code, IEE, Electronics Letters, Volume 33, Issue 21, pages 1769-1771, 1997.

97. E. Shilat, M. Werman, and Y. Gdalyahu, Ridge's corner detection and correspondence, Proc. IEEE Conf on Computer Vision and Pattern Recognition (CVPR97), pages 976-981, 1997.

98. S. Nassif, D. Capson, and A. Vaz, Robust real-time corner location measurement, Proc. IEEE Conf. on Instrumentation and Measurement Technology, pages 106-111, 1997.

99. K. Sohn, J. H. Kim, and W. E. Alexander, Mean field annealing approach to robust corner detection, IEEE Trans. Systems, Man, and Cybernetics, Volume 28B, Issue 1, pages 82-90, 1998.

100. X. Zhang and D. Zhao, Parallel algorithm for detecting dominant points on multiple digital curves, Pattern Recognition, Volume 30, Issue 2, pages 239-244, 1997.

101. K. Kohlmann, Corner detection in natural images based on the 2-D Hilbert Transform, Signal Processing, Volume 48, Issue 3, pages 225-234, 1996.

102. R. Mehrotra, S. Nichani, and N. Ranganathan, Corner detection, Pattern Recognition, Volume 23, Issue 11, pages 1223-1233, 1990.

103. Zuniga and R. M. Haralick, Corner detection using the facet model, Proc. Conf. on Pattern Recognition and Image Processing (CVPR83), pages 30-37, 1983.

104. S. M. Smith and J. M. Brady, SUSAN—A new approach to low level image processing, Defense Research Agency, Technical Report no. TR95SMS1, Farnborough, England, 1994.

105. M. Orange and F. C. A. Groen, Model based corner detection, Proc. IEEE Conf. Computer Vision and Pattern Recognition, pages 690-691, 1993.

106. Bejamin Bell and L. F. Pau, Contour tracking and corner detection in a logic programming environment, IEEE Trans. On Pattern Analysis and Machine Intelligence, Volume 12, Issue 9, pages 913-917 September 1990.

107. Hong-Chih Liu and M. D. Srinath, Corner detection from chain-code, Pattern Recognition, Volume 23, Issue 1-2, pages 51-68, 1990.

108. Freeman and L. Davis, A corner-finding algorithm for chain-coded curves, IEEE Trans on Computers, Volume 26, Issue 3, pages 297-303, 1977.

109. Rattarrangsi and R. T. Chin, Scale-based detection of corners of planar curves, IEEE Trans. On Pattern Analysis and Machine Intelligence, Volume 14, Issue 4, pages 430-449, 1992.

110. Farzin Mokhtarian and Riku Suomela, Robust image corner detection through curvature scale space, IEEE Trans. On Pattern Analysis and Machine Intelligence, Volume 20, Issue 12, pages 1376-1381, December, 1998.

111. F. Mokhtarian and A.K. Mackworth, A theory of Multi-Scale, Curvature-based shape representation for planar curves, IEEE Trans. Pattern Analysis and Machine Intelligence, Volume 14, Issue 8, pages 789-805, Aug. 1992.

112. F. Mokhtarian and R. Suomela, Curvature Scale Space for robust image corner detection, Int'l Conf on Pattern Recognition (ICPR98), Volume 2, page 1819, 1998.

113. P.V.C. Hough, Method and Means for Recognising Complex Patterns, U.S. Pattern, No. 3069654, 1962.

114. Richard O. Duda, Peter E. Hart, Use of the Hough transformation to detect lines and curves in pictures, Communications of the ACM, Volume 15, Issue 1, Pages: 11 - 15,January 1972.

115. D.H. Ballard, Generalizing the Hough transform to detect arbitrary shapes, Pattern Recognition, Volume 13, Issue 2, pages 111-122, 1981.

116. Davies, E.R., Application of the generalised Hough transform to corner detection, Computers and Digital Techniques, IEE Proc. Volume: 135, Issue: 1 pages 49- 54 Jan 1988.

117. Diou, A. Voisin, Y. Santo, C., The Hough transform-a new approach, Proc. IEEE Conf on Industrial Electronics, Control, and Instrumentation, Volume 3, pages 1612 - 1617, 1996.

118. Anastasios L. Kesidis & Nikos Papamarkos, On the Inverse Hough Transform, IEEE trans. Pattern analysis & machine intelligence, Volume 21, Issue 12, Dec 1999.

119. Fei Shen & Han Wang, Corner detection based on modified Hough transform, Pattern Recognition Letters, Volume 23, Issue 8, Pages: 1039 - 1049, June 2002.

120. Yu-Hua Gu, Tjahjadi Tardi, Corner based feature extraction for object retrieval, Proc. Int'l conf on Image Processing, Volume 1, pages 119 – 123 1999.

121. Cordelia Schmid, Roger Mohr, Christian Bauckhage, Evaluation of interest point detectors, International Journal of Computer Vision, Volume 37, Issue 2, pages 151–172, 2000.

122. Nicu Sebe, Qi Tian, Etienne Loupias, Michael S. Lew, and Thomas S. Huang, Evaluation of salient point techniques, Proc. Int'l Conference on Image and Video Retrieval, pages 367–377, 2002.

123. Krystian Mikolajczyk and Cordelia Schmid, Scale and affine invariant interest point detectors, International Journal of Computer Vision, Volume 60, Issue 1, pages 63–8, 2004.

124. K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool, A comparison of affine region detectors, International Journal of Computer Vision, Volume 65, Issue 1-2 , pages 43 – 72, 2005.

125. K. Mikolajczyk and C. Schmid, A Performance Evaluation of Local Descriptors, IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume 27, Issue 10, pages 1615 – 1630, 2005.

126. Shivani Agarwal and Dan Roth, Learning a sparse representation for object detection, Proc. 7th European Conference on Computer Vision-Part IV (ECCV) pages 113 - 130, 2002.

127. C. Schmid and R. Mohr, Local gray value invariants for image retrieval, IEEE Trans. on Pattern Analysis and Machine Intelligence, Volume 19, Issue 5, pages 530–535, 1997.

128. E. Loupias, N. Sebe, S. Bres, and J-M. Jolion. Wavelet-based salient points for image retrieval, Proc. Intl Conference on Image Processing, Volume 2, pages 518-521, 2000.

129. L. Fei-Fei, R. Fergus and P. Perona. Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories. IEEE. CVPR 2004, Workshop on Generative-Model Based Vision. 2004. http://www.vision.caltech.edu/archive.html

130. Ahmad N. Jongan Park, Corner geometry representation using code vectors for image retrieval, The Second IEEE International Conference on Digital Information Management (ICDIM 2007) , Volume: 1, pages 87-91, 2007.

131. Aibing Rao, Srihari, R.K., Zhongfei Zhang, Spatial color histograms for content-based image retrieval, Proc. 11th IEEE International Conference on Tools with Artificial Intelligence, pages 183-186, 1999.

132. Greg Pass and Ramin Zabih. Histogram Refinement for content–based image retrieval. Proc. IEEE Workshop on Applications of Computer Vision, pages 96-102, 1996.

133. Ponce, J. Hebert, M. Schmid, C. Zisserman, A. (Eds.), Toward Category-Level Object Recognition, LNCS Volume 4170 Springer-Verlag, 2006.

134. Ullman, S. Sali, E. Object Classification Using a Fragment-Based Representation, Proc. Conf on Biologically Motivated Computer Vision, (BMVC2000), pages 73-87 Springer-Verlag, 2000.

135. Michael S. Lew, Nicu Sebe, Chabane Djeraba, Ramesh Jain, Content-Based Multimedia Information Retrieval: State of the Art and Challenges, ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP), volume 2, issue 1, pages, 1-19, 2006.

136. Mojsilovic, A. Rogowitz, B. Capturing image semantics with low-level descriptors, Proc. IEEE Int. Conf. on Image Processing (ICIP), Volume 1, Pages 18-21, 2001.

137. T. Serre, L. Wolf, and T. Poggio, Object recognition with features inspired by visual cortex, Proc. IEEE Int. Conf. on Computer Vision and Pattern Recognition (CVPR), Volume: 2, Pages, 994 - 1000, 2005.

138. Joseph L. Mundy, Object recognition in the geometric era: a retrospective, In J. Ponce, M. Hebert, C. Schmid, and A. Zisserman (editors), "Toward category-level object recognition", LNCS Springer-Verlag, Volume 4170. Pages 3-29, 2006.

139. David G. Lowe, Distinctive image features from scale-invariant key-points, International Journal of Computer Vision, Volume 60, Issue 2, Pages 91-110, 2004.

140. David G. Lowe, Object recognition from local scale-invariant features, International Conference on Computer Vision (ICCV), Pages 1150-1157, 1999.

141. Alexandra Teynor, Patch Based Approaches for the Recognition of Visual Object Classes - A Survey, Internal Report 2/06, IIF-LMB, University of Freiburg, Germany, 2006. http://lmb.informatik.uni-freiburg.de/people/teynor/index.en.html.

142. Jim Mutch and David G. Lowe, Object class recognition and localization using sparse features with limited receptive fields, International Journal of Computer Vision, Volume 80, Issue 1, pages 45-57, 2008.

143. Michael Wertheimer, D. King (Authors), Max Wertheimer and Gestalt Theory, Transaction Publishers, New Brunswick (USA) and London (U.K.), 2004.

144. Andrew P. Witkin, Jay M. Tenenbaum, What Is Perceptual Organization For? Alan Bundy (Ed.) Proc. 8th Int. Joint Conference on Artificial Intelligence (IJCAI) Volume 2, pages 1023-1026, 1983.

145. David G. Lowe, Perceptual Organization and Visual Recognition, Springer Int. Series in Engineering and Computer Science, Volume 5, 1985.

146. S. Sarkar and K. L. Boyer, Perceptual organization in computer vision: A review and a proposal for a classificatory structure, IEEE Trans. Systems, Man, Cybernetics Volume 23, Issue 2, pages 382–399, 1993.

147. David. Marr, Vision: A Computational Investigation into the Human Representation and Processing of Visual Information, W. H. Freeman and Co., ISBN 0-7167-1284-9, 1982.

148. Etemadi, A. Schmidt, J.P. Matas, G. Illingworth, J. Kittler, J.V, Low-Level Grouping of Straight Line Segments, Proc. British Machine Vision Conference (BMVC91), Pages 119-126, 1991.

149. Q. Lu and J. K. Aggarwal, Applying perceptual organization to the detection of man-made objects in non-urban scenes, Pattern Recognition, Volume 25, Issue 8, Pages 835-853, 1992.

150. Q. Iqbal and J. K. Aggarwal, Retrieval by Classification of Images Containing Large Manmade Objects Using Perceptual Grouping, Pattern Recognition Journal Volume 35, Issue 7. pages 1463-1479, 2002.

151. P. D. Kovesi, Edges are not just steps, Proc. Fifth Asian Conference on Computer Vision (ACCV), pages 822–827 Pub. Australian Pattern Recognition Society, 2002.

152. R. Pope and D. G. Lowe, Vista: A software environment for computer vision research, Proc. Int. Conference on Computer Vision and Pattern Recognition (CVPR) Pages 768–772, 1994.

153. Jon Dattorro, Convex Optimization & Euclidean Distance Geometry Meboo Publishing USA, 2004. http://meboo.convexoptimization.com/.

154. T. Cover and P. Hart, Nearest neighbor pattern classification, IEEE Trans on Information Theory, Volume 13, Issue 1, pages 21-27, 1967.

155. Li Fei-Fei, R. Fergus and P. Perona, Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories, IEEE CVPR2004, Workshop on Generative-Model Based Vision, 2004.

156. R. Fergus, P. Perona, and A. Zisserman, Object class recognition by unsupervised scale-invariant learning, Proc. Int. Conference on Computer Vision and Pattern Recognition (CVPR), volume 02, page 264, 2003.

157. Fayin Li, J. Kosecka, and H. Wechsler, Strangeness based feature selection for part based recognition, Proc. Int. Conf. on Computer Vision and Pattern Recognition (CVPR), 2006.

158. Grauman and T. Darrell, Pyramid match kernels: Discriminative classification with sets of image features, Technical Report MIT-CSAIL-TR-2006-020, March 2006.

159. Zhang, A. Berg, M. Maire, and J. Malik, Svm-knn: Discriminative nearest neighbor classification for visual category recognition, Proc. Int. Conference on Computer Vision and Pattern Recognition (CVPR), 2006.

160. S. Lazebnik, C. Schmid, and J. Ponce, Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories, Proc. Int. Conference on Computer Vision and Pattern Recognition (CVPR), 2006.

161. C. Berg, Shape Matching and Object Recognition, PhD thesis, Computer Science Division, University of California, Berkeley, 2005.

162. C. Berg, T. L. Berg, J. Malik, Shape matching and object recognition using low distortion correspondence, Proc. Int. Conference on Computer Vision and Pattern Recognition (CVPR), 2005.

163. Wang, Y. Zhang, L. Fei-Fei, Using dependent regions for object categorization in a generative framework, Proc. Int. Conference on Computer Vision and Pattern Recognition (CVPR), 2006.

164. Holub, M. Welling, P. Perona, Exploiting unlabelled data for hybrid object classification, Proc. NIPS Workshop on Inter-Class Transfer, 2005.

165. Boiman, O., Shechtman, E., Irani, M., In Defense of Nearest-Neighbor Based Image Classification, Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 1-8, June 2008.

166. Ballard, D. H., Brown, C. M. Computer Vision, Englewood Cliffs, New Jersey: Prentice-Hall, 1982.

저작물 이용 허락서

학 과	정보통신공학과	학 번	20067725	과 정	박사
성 명	한글:아흐매드니스핫 영문 : Ahmad Nishat				
주 소	74-F Vihari Pakistan 61100				
연락처	E-MAIL : nischat @ gmail.com				
논문제목	한글 : 로칼 패치와 시멘틱 구조 특성을 이용한 비주얼 객체 인식 및 검색 영어 :Visual Object Class Recognition and Retrieval using Local Patch and Semantic Image Structure Features				

본인이 저작한 위의 저작물에 대하여 다음과 같은 조건아래 조선대학교가 저작물을 이용할 수 있도록 허락하고 동의합니다.

- 다 음 -

1. 저작물의 DB 구축 및 인터넷을 포함한 정보통신망에의 공개를 위한 저작물의 복제, 기억장치에의 저장, 전송 등을 허락함
2. 위의 목적을 위하여 필요한 범위 내에서의 편집・형식상의 변경을 허락함. 다만, 저작물의 내용변경은 금지함.
3. 배포・전송된 저작물의 영리적 목적을 위한 복제, 저장, 전송 등은 금지함.
4. 저작물에 대한 이용기간은 5 년으로 하고, 기간종료 3 개월 이내에 별도의 의사 표시가 없을 경우에는 저작물의 이용기간을 계속 연장함.
5. 해당 저작물의 저작권을 타인에게 양도하거나 또는 출판을 허락을 하였을 경우에는 1 개월 이내에 대학에 이를 통보함.
6. 조선대학교는 저작물의 이용허락 이후 해당 저작물로 인하여 발생하는 타인에 의한 권리 침해에 대하여 일체의 법적 책임을 지지 않음
7. 소속대학의 협정기관에 저작물의 제공 및 인터넷 등 정보통신망을 이용한 저작물의 전송・출력을 허락함.

동의여부 : 동의(○) 반대( )

2009 년 8 월

저작자: Ahmad Nishat (서명 또는 인)

조선대학교 총장 귀하