REVIEW OF IMAGE AND VIDEO INDEXING TECHNIQUES IN PIXEL AND COMPRESSED DOMAIN

Meenakshi Choudhary[1] | Prof. Sushma Lehri[2]

[1](Institute of Engineering and Technology, Dr. B. R. Ambedkar University, Agra UP India, choudhary69@gmail.com)

[2](Institute of Engg and Technology, Dr. B. R. Ambedkar University, Agra UP India, sushma_lehri2003@yahoo.co.in)

Abstract— *Visual database systems require efficient indexing to facilitate fast access to the images and video sequences in the database. Recently, several content-based indexing methods for image and video based on spatial relationships, color, texture, shape, sketch, object motion, and camera parameters have been re-ported in the literature. The goal of this paper is to provide a critical survey on image and video indexing technique in both pixel and compressed domain.*

1. INTRODUCTION

The access to visual database has two main components: storage and retrieval. In the storage process, images and video are processed to extract features which describe their semantics. The extracted features are then represented, organized, and stored in the database. In the retrieval process, the system analyzes a query and extracts the appropriate feature vector and then a search process is per-formed. The search process is carried out by computing the ‘‘similarity’’ (using a similarity metric) between the feature vector of the query and those of the candidate images and video stored in the database. The retrieved images and video are presented to the user in the descending order of the similarity to the query. Several image and video database systems have been proposed [1]. Architecture for a generic image/video database system is shown in Fig. 1. It consists of the user interface, content-based retrieval, organization, and database management modules. A functional description of each module is presented below.

1.1 User interface:

In visual information systems, user interaction plays an important role in almost all of its functions (e.g., semiautomatic and manual feature extraction, navigation, selection, and refinement). The user inter-face consists of a query processor and a browser to provide the interactive graphical tools and mechanisms for que-ring and browsing the database, respectively. The query processor provides the means to query images and video using a variety of query methods and interfaces. A query can range from a simple keyword-based to a complex query where the user specifies a sketch or an object track. In contrast to textual database systems, image and video data-bases are required to evaluate properties of the data specified in a query. For example, to retrieve all images like a query image based on color, the color attributes (e.g., color histogram) of the query image must be calculated. After obtaining the responses to a query, the browser is used to display the results. The browser allows users to further refine and navigate through the database visually.

1.2 Content-based retrieval module:

The content-based retrieval module consists of the following

Scene Change Detection: The first step in video indexing is to decompose a video sequence into shots. Once a video sequence is segmented into shots, a set of representative frames is then selected to represent the shot [2]. Each shot is represented using spatial and temporal features. The spatial features refer to the spatial content of the representative frame of a shot, while the temporal features represent the temporal content of a shot. The representative frames of a shot are fed to the image preprocessing stage in order to generate the spatial features of the shot, while the shots are input to the feature extraction and representation stage in order to extract the temporal features.

Image preprocessing: The image is first processed to extract the features which describe its contents. The processing might involve decompression, enhancement, filtering, normalization, segmentation, and object identification. If the input image is in the compressed form, decompression is required to facilitate execution of the pixel domain algorithms. The output of the image preprocessing stage is typically a collection of objects and regions of interest.

Feature extraction and representation: In this stage, these mantics of image/video content are extracted and represented. Features (of the objects, regions, and/or the whole image) such as texture, color, etc., are used to describe the content of a still image. For video, the spatial features are generated using still image techniques [3] while the temporal features are extracted based on motion and/or camera operations within the shot. Image and video features can be classified into primitive and logical features. Primitive features such as color, shape centroids, etc., are quantitative in nature and can be extracted automatically or semi automatically. Logical features are qualitative in nature and provide abstract representations of visual data at various levels of detail. Typically, logical features are extracted manually. One or more features can be used in a specific application. For example, in a satellite information system, the texture features are important, while shape and color features are more important in trademark registration systems. Once the features have been extracted, the textual, numerical, alphanumerical, etc., index keys are derived.

1.3 Organization

Efficient query processing necessitates the organization of image/video indices such that efficient search strategies can be used. We note that image/video indices are approximately represented, may have interrelated multiple attributes and may not have an embedded order [3]. Therefore, conventional indexing structures like B-tree and hashing, etc., cannot be used for the organization of image/video indices. Flexible data structures should be used in order to facilitate storage/retrieval in visual information systems. Structures such as R-tree family [4], R *-tree [5], quad-tree [6], and grid file [7] are commonly used. Each structure has its advantages and disadvantages; some have limited domains and some can be used concurrently with others. Niu et al. [8] have discussed some issues concerning novel indexing structures for image retrieval. Ahanger and Little [9] have also presented a review of indexing structures for video.

1.4 Database management module

The database management module provides internal level physical storage structure and access path to the database. The database management module has the following characteristics: (i) Provides insulation between programs and data, (ii) Pro-vides users with a conceptual representation of the data, (iii) Supports multiple views of the data, and (iv) Ensures data consistency.

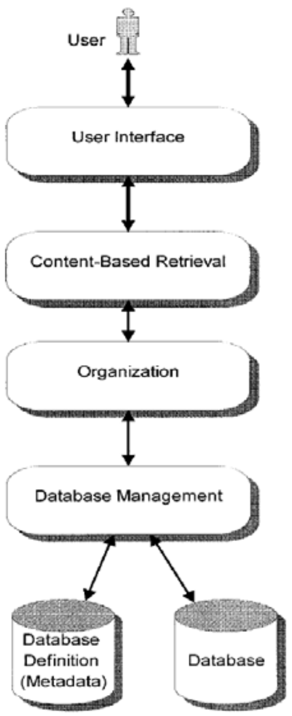

FIG. 1. Storage and retrieval of images and video.

2. IMAGE INDEXING IN PIXEL DOMAIN

The pixel domain indexing of visual data is based on features such as texture, shape, sketch, histogram, color, moments, etc. For example, the query by image content (QBIC) system developed by IBM [7] retrieves images based on color, texture, shape, and sketches. The content based retrieval engine (CORE) for Multimedia Information Systems proposed by Wu et al. [9] employs color and word similarity measures to retrieve images based on content and text annotation, respectively. We now briefly describe the state of the art approaches in image indexing.

2.1 Color

The color feature is one of the most widely used visual features in image retrieval. It is relatively robust to background complication and independent of image size and orientation. In image retrieval, the color histogram is the most commonly used color feature representation. Statistically, it denotes the joint probability of the intensities of the three-color channels. Swain and Ballard proposed histogram intersection, an L1 metric, as the similarity measure for the color histogram. To take into account the similarities between similar but not identical colors. Furthermore, considering that most color histograms are very sparse and thus sensitive to noise, Stricker and Orengo proposed using the cumulated color histogram. Their research results demonstrated the advantages of the proposed approach over the conventional color histogram approach [8]. Besides the color histogram, several other color feature representations have been applied in image retrieval, including color moments and color sets. To overcome the quantization effects, as in the color histogram, Stricker and Orengo proposed using the color moments approach [9]. The mathematical foundation of this approach is that any color distribution can be characterized by its moments. Furthermore, since most of the information is concentrated on the low-order moments, only the first moment (mean), and the second and third central moments (variance and skewness) were extracted as the color feature representation. Weighted Euclidean distance was used to calculate the color similarity.

2.2 Texture

Texture refers to the visual patterns that have properties of homogeneity that do not result from the presence of only a single color or intensity [4]. It is an innate property of virtually all surfaces, including clouds, trees, bricks, hair, and fabric. It contains important information about the structural arrangement of surfaces and their relationship to the surrounding environment [5]. Because of its importance and usefulness in pattern recognition and computer vision, there are rich research results from the past three decades. Now, it further finds its way into image retrieval. More and more research achievements are being added to it. In the early 1970s, Haralick et al. proposed the co-occurrence matrix representation of texture features [59]. This approach explored the gray level spatial dependence of texture. It first constructed a co-occurrence matrix based on the orientation and distance between image pixels and then extracted meaningful statistics from the matrix as the texture representation. Many other researchers followed the same line and further proposed enhanced versions. For example, Gotlieb and Kreyszig studied the statistics originally and experimentally found out that contrast, inverse deference moment, and entropy had the biggest discriminatory power [5]. Motivated by the psychological studies in human visual perception of texture, Tamura et al. explored the texture representation from a different angle. They developed computational approximations to the visual texture properties found to be important in psychology studies. The six visual texture properties were coarseness, contrast, directionality, line likeness, regularity, and roughness. One major distinction between the Tamura texture representation and the co-occurrence matrix representation is that all the texture properties in Tamura representation are visually meaningful, whereas some of the texture properties used in co-occurrence matrix representation may not be (for example, entropy). This characteristic makes the Tamura texture representation very attractive in image retrieval, as it can provide a more user-friendly interface. The QBIC system and the MARS system further improved this texture representation. In the early 1990s, after the wavelet transform was introduced and its theoretical frame work was established, many researchers began to study the use of the wavelet transform in texture representation. Smith and Chang used the statistics (mean and variance) extracted from the wavelet subbands as the texture representation. This approach achieved over 90% accuracy on the 112 Brodatz texture images. To explore the middle-band characteristics, a tree-structured wavelet transform was used by Chang and Kuo further improve the classification accuracy. The wavelet transform was also combined with other techniques to achieve better performance. Gross et al. used the wavelet transform, together with KL expansion and Kohonen maps, to perform texture analysis in [5]. Thyagarajan et al. [9] and Kundu et al. [7] combined the wavelet transform with a co-occurrence matrix to take advantage of both statistics-based and transform-based texture analyses.

2.3 *Shape*

In image retrieval, depending on the applications, some require the shape representation to be invariant to translation, rotation, and scaling, while others do not. Obviously, if are presentation satisfies the former requirement, it will satisfy the latter as well. Therefore, in the following we will focus on shape representations that are transformation invariant.

In general, the shape representations can be divided into two categories, boundary-based
and region-based. The former uses only the outer boundary of the shape while the latter uses the entire shape region. The most successful representatives for these two categories are Fourier descriptor and moment invariants. The main idea of a Fourier descriptor is to use the Fourier transformed boundary as the shape feature. To take into account the digitization noise in the image domain, Rui et al. proposed a modified Fourier descriptor which is both robust to noise and invariant to geometric transformations. The main idea of moment invariants is to use region-based moments which are invariant to transformations, as the shape feature. In [6], Hu identified seven such moments. Basedon his work, many improved versions emerged. In [169], based on the discrete version of Green's theorem, Yang and Albregtsen proposed a fast method of computing moments in binary images. Motivated by the fact that most useful invariants were found by extensive experience and trial-and-error, Kapur et al. developed algorithms to systematically generate and search for a given geometry's invariants. Realizing that most researchers did not consider what happened to the invariants after image digitization, Gross and Latecki developed an approach which preserved the qualitative differential geometry of the object boundary, even after an image was digitized. In [10] a framework of algebraic curves and invariants is proposed to represent complex objects in a cluttered scene by parts or patches. Polynomial fitting is done to represent local geometric information, from which geometric invariants are used in object matching and recognition.

2.4 *Color Layout*

Although the global color feature is simple to calculate and can provide reasonable discriminating power in image retrieval, it tends to give too many false positives when the image collection is large. Many research results suggested that using color layout (both color feature and spatial relations) is a better solution to image retrieval. To extend the global color feature to a local one, a natural approach is to divide the whole image into subblocks and extract color features from each of the subblocks. A variation of this approach is the quadtree-based color layout approach [10], where the entire image was split into a quadtree structure and each tree branch had its own histogram to describe its color content. Although conceptually simple, this regular subblock-based approach cannot provide accurate local color information and is computation- and storage expensive. Amoresophisticated approach is to segment the image into regions with salient color features by color set back-projection and then to store the position and color set feature of each region to support later queries [9]. The advantage of this approach is its accuracy while the disadvantage is the general difficult problem of reliable image segmentation.

To achieve a good trade-off between the above two approaches, several other color layout representations were proposed. In [11], Rickman and Stonham proposed a color tuple histogram approach. They first constructed a code book which described every possible combination of coarsely quantized color hues that might be encountered within local regions in an image. Then a histogram based on quantized hues was constructed as the local color feature. In [12], Stricker and Dimai extracted the first three color moments from five predefined partially overlapping fuzzy regions. The usage of the overlapping region made their approach relatively insensitive to small region transformations. In [10], Passet al. classified each pixel of a particular color as either coherent or incoherent, based on whether or not it is part of a large similarly colored region. By using this approach, widely scattered pixels were distinguished from clustered pixels, thus improving the representation of local color features.

2.5 *Segmentation*

Segmentation is very important to image retrieval. Both the shape feature and the layout feature depend on good segmentation. In this sub section, we will describe some existing segmentation techniques used in both computer vision and image retrieval.

In [13], Lybanon et al. researched a morphological operation (opening and closing) approach in image segmentation. They tested their approach in various types of images, including optical astronomical images, infrared ocean images, and magnetograms. While this approach was effective in dealing with the above scientific image types, its performance needs to be further evaluated for more complex natural scene images. In [14], Hansen and Higgins exploited the individual strengths of watershed analysis and relaxation labeling. Since fast algorithm exists for the watershed method, they first used the watershed to subdivide an image into catchmen basins. They then used relaxation labeling to refine and update the classification of catchment basins initially obtained from the watershed to take advantage of the relaxation labeling's robustness to noise.

3. Video Indexing in Pixel Domain

A video sequence is a set of image frames ordered in time. Generally, video indexing refers to indexing of individual video frames based on their contents and the associated camera operations involved in the imaging process. However, the neighboring frames in a video sequence in general are highly correlated Hence, for computational efficiency, the video sequence is segmented in a series of shots. A shot is defined as a sequence of frames generated during a continuous operation and representing a continuous action in time and space. A frame in each shot is declared as a representative frame. Indexing is performed by applying the image indexing technique on representative frames from each shot. Each shot in a video sequence consists of frames with different scenes. There are two ways by which two shots can be joined together: (i) abrupt transition; and (ii) gradual transition. In abrupt transition, two shots are simply concatenated, while in the gradual transition, additional frames may be introduced using editing operations such as fade in, fade out or dissolve. A good video segmentation technique should be able to detect shots with both types of transition. The apparent motion in a video sequence can be attributed to camera or object motion. Motion estimation/ compensation plays an important role in video compression. The objective is to reduce the bit rate by taking advantage of the temporal redundancies between adjacent frames in a video sequence. Typically, this is accomplished by estimating the displacement (motion vectors) of uniformly sized blocks between two consecutive frames. In general, motion vectors exhibit relatively continuous changes within a single camera shot, while this continuity will be disrupted between frames across different shots.

Detecting camera motion is becoming important with potential applications in low bit-rate video coding and video editing. We note that there are seven basic camera operations tracking, tilting, booming, zooming, and dollying. Since both object motion and camera motion are reflected in the observed motion vectors of a block coding scheme, it is generally difficult to estimate the camera motion. However, several models have recently been proposed to improve the estimation. A review of camera motion estimation is outside the scope of this paper.

4. Compressed-Domain Image/Video Indexing and Searching

4.1 Texture Discrimination & Search

In an effort to provide feature-based image query, we have derived automatic algorithms for extracting low-level signal features from the transform compressed images [6]. One specific example is to define the texture features based on the spatio frequency decomposition of the images. Textures has been used to describe content of many real-world images; for example, clouds, trees, bricks, hair, fabric all have textural characteristics. Psycho-physical studies have shown that humans perceive textures by decomposing signals into components with different frequency and orientation. We use the feature sets defined in transform decomposition to approximate the texture feature. Transform decomposition of images can be obtained by taking DCT, sub-band transform, or wavelet transform of the images. From the decomposed signal bands, texture feature sets are defined by measuring each sub and energy. For example, for a 5-level wavelet decomposition, feature vectors with 16 terms are produced. For a N *N DCT transform, N2 signal bands can be obtained by regrouping transform coefficient (i.e., the DCT/Mandala trans-form). Other statistical measures such as first-order moments can also be derived from each subband in forming the transform-domain texture features. Based on our experiment of texture classification, the energy measurement seems to be the most effective one.

In order to reduce the search complexity, the above texture feature vector is further reduced by using the Fisher Discriminant technique. The criterion is to maximize the class separability among all different known texture classes in the chosen test set (i.e., the Brodatz Texture Set) [4]. The test set contains 112 different texture classes typically used in computer vision research. Based on this 112 texture classes, we generated an image database containing more than 2000 random cuts from the Brodatz set. Given an input image key, the transform-domain feature elements are mapped to a set of eigenvectors with the maximum separability significance. The Mahalanobis distance in the transformed feature space was used to measure the similarity between the input image key and every image in the database. Our experiment shows satisfactory correct classification rate. Even with only 6-8 feature elements per image, the classification rate remains at about the 90% level. In the comparison of different transform algorithms, the wavelet subband and the uniform subband have the highest classification rates compared to the DCT/Mandala transform. The widely used DCT has a decent classification rate, about 85%. The slightly lower classification rate by using the DCT transform is basically due to its less effective energy con-centration capability.

The above task of texture classification operates on the entire image and does not require texture segmentation. However, in order to discriminate distinctive local features, identification of local image regions of prominent features is necessary. We recognize that robust texture segmentation is a research issue still in progress. We also argue that for texture-based image query applications, accurate boundary information is not really necessary. Therefore, we relax the requirement and just aim to extract homogeneous image regions with prominent texture features. Using a modified quad-tree and the threshold derived in [6], we were able to use the transform-domain texture feature to extract prominent regions from each image in the database. One image may have zero or multiple prominent homogeneous texture regions. Given the input image key, the texture feature vector is derived from the transform domain and compared against every region contained in every image in the database. Images containing the most similar image regions were returned as matches.

Compared to the traditional technique for texture extraction, such as pixel-domain window-based Law filters, our transform-domain texture features also include local window-based filter-ing. For example, a 8*8 DCT transform has a local window size of 64 pixels. Determination of the transform size actually involves an interesting tradeoff between texture homogeneity and statistical confidence. Larger transform block sizes give higher statistical confidence, but potentially lower homogeneity of the texture feature.

4.2 Image Matching

Image matching has been used in many applications including image registration, pattern recognition, and stereoscopic image correspondence matching. Two critical factors in image matching are determination of the matching criterion and the search space. One example is the minimal distortion matching used in the popular motion estimation algorithm for video coding. In a paper, author have derived algorithms for doing motion estimation and inverse motion compensation in the DCT domain. For any orthonormal transforms like DCT, the Euclidean distance is preserved in the transform domain. However, because motion compensation is pixel-based while DCT is block-based, computation of the DCT of each reference block may involve significant overhead in realigning the DCT block structure. To compensate for this over-head, the search space may need to be reduced, using some heuristics such as the 3-point motion estimation technique.

If the images are encoded by wavelet or subband transforms, image matching can be implemented in an intelligent, hierarchical way as well. Suppose we adopt correlation as the matching criterion and use the exhaustive search space. Searching for the position with the highest correlation is equivalent to finding the peak value in the convolution. One can prove that the correlation criterion is closely related to the MSE or the correlation coefficient criterion. It has been shown that convolution of two 1-D sequences can be decomposed to the summation of convolutions of their subband components. Specifically, the summation of all intra-subband convolutions equals a subsampled version of the complete convolution. In [8], we took a similar approach to implement a hierarchical image matching method. If {h1, h2} and {g1, g2} are subsampled low-band and high-band signal decomposition of the original sequences h and g. If the analysis filters are ideal half-band low-pass and high-pass filters, the cross-subband terms will be zero. For practical filters, such as the Harr filter and QMF filters, these terms are non-zero although they are relatively small compared to the intra-subband convolutions. Author described an adaptive convolution scheme which adaptively approximates the complete convolution with the dominant subband convolutions. The criteria for choosing the dominant subband convolutions are based on two possible features — energy and feature. The energy-based approach chooses the subbands with the highest energy and approximate the complee convolution with the intra- and cross-subband convolutions associated with those dominant subbands. Note that the subband decomposition can be iterated more times in a uniform, logarithmic, or adaptive way to create signal decompositions at more levels. The above adaptive, hierarchical convolution can be easily repeated in each iteration. The hierarchical image searching method has been studied earlie, but only low-low band convolution was used to approximate the complete result. One alternative criterion for choosing the significant subbands is to use the signal features, such as edge and texture, in each subband. For example, if one subband has strong indication of edge or texture content, it is better to include that subband in the approximation.

Another promising technique for image matching in the wavelet subband domain is to incorporate the zero-crossing representation.

4.3 Video Indexing and Editing

Compared to still images, a video sequence can be further characterized by two additional "features" — (1) How the video is captured (i.e., the camera operations such as zooming and panning)? (2) how image features change over time (e.g., object motion and inter-scene temporal relationship)? There are existing techniques for extracting these dynamic visual features in the uncompressed domain. Work has been reported to detect scene changes in the transform domain and the MPEG domain. Independently, we have applied the compressed-domain feature extraction principal and developed a Compressed Video Editing and Parsing System (CVEPS), which allows automatic parsing of the MPEG-1 and MPEG-2 compressed video streams to detect scene changes, dissolve, and fade in/out. Abrupt scene changes can be detected by image intensity variance discontinuity and/or the distribution of the motion vectors in the B and P frames (e.g., the ratio among the numbers of forward predicted blocks, backward predicted blocks, and intraframe coded blocks). Dissolve scene changes can be characterized by modeling the image intensity variance with a quadratic form. Detection of scene change and dissolve requires establishment of some thresh-old values. We used an adaptive local threshold based on local video activities instead of a global threshold value. There is other useful information which can be derived from the compressed data.

4.4 Compressed-Domain Image Manipulation

Author extend the above compressed-domain approach to image manipulation in this section. Image manipulation involves many useful operations for general multimedia applications. In general, it includes linear and non-linear operations. We have been focusing on the compressed-domain solutions for linear operations, such as filtering, geometrical transformation, multi-object composition, pixel multiplication, and convolution.

In other words, given the transform coefficients of the input images, we can directly calculate the transform coefficients of the output filtered images directly in the transform domain by using images have been truncated (as done in quantizers of practical coding methods), a great number of small coefficients may be truncated to zeros. Therefore, the computational complexity associated with the compressed-domain operations could be greatly reduced. In a test scenario which takes three input image sources, scaled each of them to different sizes, and translate them to different locations in the final composited scene, we have been able to reduce overall computational complexity by about 65% by using the proposal transform-domain image manipulation approach, compared to the traditional uncompressed domain approach.

To extend the image manipulation techniques to the motion-compensation domain is not directly feasible, due to the complication of the motion compensation algorithm.

Some operations, such as shearing and rotation, cannot be directly modeled by a linear operation. In general, they require different operations on different rows and columns. This problem can be solved by using the divide-and-conquer approach.

5. CONCLUSIONS

With the progress of multimedia technology, large amounts of visual data will be widely accessible and thus will become one of the primary sources of information, much as text is today. Whether the application is distance learning, digital libraries, interactive television, multimedia news, or banking, large volumes of visual data will be required to be accessed precisely and efficiently.

Image and video indexing is crucial in many applications for efficient retrieval of visual information from multimedia databases. A straightforward extension of existing text indexing techniques for image and video indexing is inefficient and complex. Moreover, text-based approaches are not generic and hence are not useful in a wide variety of applications. As a result, there has been a new focus on developing content-based indexing techniques which are domain independent and can be automated. This raises several issues which need to be addressed.

REFERENCES

[1] G. Salton and M. J. McGill, An Introduction to Modern Information Retrieval, McGraw–Hill, New York, 1983

[2] J. J. Fan and K. Y. Su, An efficient algorithm for matching multiple patterns, IEEE Trans. Knowl. Data Eng.5(2), April 1993, 339–351

[3] A. Gupta, T. Weymouth, and R. Jain, Semantic queries with pictures: The VIMSYS model, Proc. VLDB'91, 1991, 69–79.

[4] T. Sellis, N. Roussopoulos, and C. Faloutsos, The R1 tree: A dynamic index for multidimensional objects, in Proceedings of the 13th Inter-national Conference on Very Large Databases, 1987, pp. 507–518.

[5] N. Beckmann, H. P. Kriegel, R. Schneider, and B. Seeger, The R*-tree: An efficient and robust access method for points and rectangles, in Proceedings ACM SIGMOD the International Conference on the Management of Data, May 1990, pp. 322–331.

[6] I. Gargantini, An effective way to represent quadtrees, Commun. ACM 25(12), 1982, 905–910.

[7] J. Nievergelt, H. Hinterberger, and K. C. Sevcik, The grid file: An adaptable symmetric multikey file structure, ACM Trans. Database Systems 9(1), March 1984, 38–71.

[8] Y. Niu, M. T. Ozsu, and X. Li, A Study of Image Indexing Techniques for Multimedia Database Systems, Department of Computing Science, University of Alberta, Technical Report TR 95-19, July 1995.

[9] G. Ahanger and T. D. C. Little, A survey of technologies for parsing and indexing digital video, J. Visual Commun. Image Represent.7(1), March 1996, 28–43

[10] D. Copper and Z. Lei, on representation and invariant recognition of complex objects based on patches and parts, in Lecture Notes in Computer Science Series, 3D Object Representation for Computer Vision (M. Hebert, J. Ponce, T. Boult, and A. Gross, Eds.), pp. 139–153, Springer-Verlag, New York/Berlin, 1995.

[11] J. R. Smith and S.-F. Chang, Local color and texture extraction and spatial query, in Proc. IEEE Int. Conf. on Image Proc., 1996.

[12] M. Strickerand A. Dimai, Color indexing with weak spatial constraints, in Proc. SPIE Storage and Retrieval for Image and Video Databases, 1996.

[13] M. Lybanon, S. Lea, and S. Himes, Segmentation of diverse image types using opening and closing, in Proc. IEEE Int. Conf. on Image Proc., 1994.

[14] M. Hansen and W. Higgins, Watershed-driven relaxation labeling for image segmentation, in Proc. IEEE Int. Conf. on Image Proc., 1994.