A Pyramidal Approach to Content-Based Image Retrieval

Ze-Nian Li
School of Computing Science
Simon Fraser University
Burnaby, B.C., CANADA V5A 1S6
li@cs.sfu.ca

Abstract

Search based on image contents is an important issue in large image and video databases. In this paper, two methods for a better content-based image retrieval (CBIR) are presented, namely, the use of recognition kernel *and* locales. *Features of model objects are extracted at levels that are most appropriate to yield only the necessary yet sufficient details, together they form the kernel. Instead of relying on image segmentation, a method of feature localization based on locales is developed. It is shown that the deployment of the recognition kernel and locales in a pyramidal (multiresolution) framework delivers good retrieval results.*

1. Introduction

The inclusion of voice, image and video in multimedia databases has proven extremely effective in various applications such as education, entertainment, medicine, and e-commerce. Multimedia data is much richer than textual (alphanumeric) data. However, it also poses many new challenges. Existing computer vision technologies [5] can readily extract features at low and intermediate levels (color, texture, depth, edge, region, simple motion and shape, etc). Earlier papers [8, 9, 3] have reported various degrees of success in searching image and/or video databases by content. Recent work [10] attempted to incorporate user feedbacks to improve the relevance of the image retrieval results. However, the field is still young and the state-of-the-art has been quite primitive.

In an effort to enable efficient retrieval in a multimedia database, we have been developing a multi-level data-modeling and retrieval system to facilitate CBIR. The system integrates the multi-level descriptions of the image and video data and their associated confidence factors. Instead of replying on image segmentation, a method of feature localization based on *locales* is developed. Comparing to most existing approaches, our work has the following characteristics: (a) the exploration of CBIR from largely reduced image data, (b) the exploitation of intrinsic image features that are most effective for CBIR, and (c) the integration of CBIR methods into a pyramidal framework.

Section 2 introduces the multi-level recognition kernel for modeling and matching in CBIR. Section 3 describes locales for feature localization. Section 4 presents the experimental results. Section 5 is a brief conclusion.

2. Recognition Kernel for CBIR

New and better methods of *Modeling* and *matching* are essential for effective and efficient CBIR in image and video databases. It is shown in this section that multiresolution modeling offers substantial savings by matching at largely reduced scales when it is possible. Meanwhile it preserves necessary details when they are appropriate at various levels.

2.1. Definition of recognition kernel

A *recognition kernel* is defined as a multiresolution model for each object. Features of an object are extracted at levels that are most appropriate to yield only the necessary yet sufficient details. Together they form the kernel.

Fig. 1 illustrates a three-level recognition kernel in which different features are adopted at each level. Certain features (such as color) are known to be well-preserved under severe reduction of image resolution, they are hence used at low-resolution. Others (such as texture and shape) require relatively higher resolutions.

1. Color: Colors in a model image are sorted according to their frequency (number of pixels) in the color histogram. The first few *Most Frequent Colors* (MFCs) and their frequencies are generally quite important as characteristic measures of an object. In this design, since color is used at a very low resolution where only very few prominent colors are preserved, the MFCs become especially dominant.

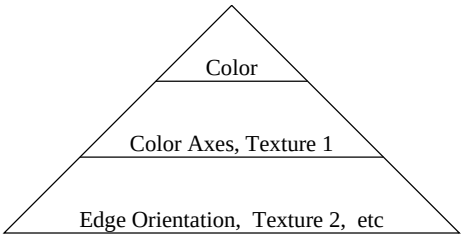


Figure 1. A Recognition Kernel.

2. Color Axes: For each MFC, the centroid of all pixels is located first. A *prime axis* for each MFC is defined about which the minimum moment is obtained. The color axis for the ith MFC is denoted by the orientation θ_i of its prime axis ($0^\circ \leq \theta_i < 180^\circ$, 30° increments). The angles between the color axes of MFCs characterize the shape and color distribution of the object.

3. Texture 1: Edge density ("edgeness") is used to give an estimation whether the area is highly textured. Edge detection is only performed on the luminance image Y, where $Y = 0.299R + 0.587G + 0.114B$.

4. Edge Orientations: Similar to sorting colors, the edge orientations can also be sorted according to their frequency (number of pixels) and the *Most Frequent Orientations* (MFOs) can be readily obtained.

5. Texture 2: At this highest resolution for modeling, second order statistics could be used to generate texture feature vectors. In the current implementation, edge density and edge axes of the MFOs (derived similarly as in color axes) are used.

2.2. Taking care of various object sizes

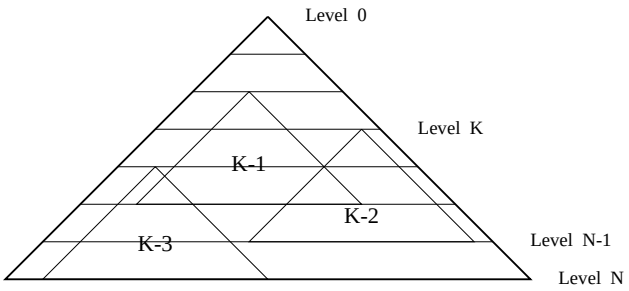


Figure 2. Match Objects with Various Sizes.

For this discussion it is assumed that the multiresolution images are square, and their sizes are $2^k \times 2^k$ at Level k in the image pyramid.

Objects are modeled at a fixed size. While matching the model with objects in the image database, variable object sizes must be dealt with. As shown in Fig. 2 the same recognition kernel will be applied at different levels in the image pyramid.

Table 1. Effective Sizes of the Search Window for Features at Multi-levels

	Kernel Resulution	*Effective Search Window*		
		K-1	K-2	K-3
Color	$32 \cdot 32$	$128 \cdot 128$	$64 \cdot 64$	$32 \cdot 32$
Color Axes, Txt 1	$64 \cdot 64$	$256 \cdot 256$	$128 \cdot 128$	$64 \cdot 64$
Txt 2, Edge Orient	$128 \cdot 128$	$512 \cdot 512$	$256 \cdot 256$	$128 \cdot 128$

Table 1 assumes that the recognition kernel consists of three levels at resolutions 32×32, 64×64, and 128×128, respectively. When the bottom level of the kernel is placed at Level 7 with the resolution of 128×128, it is denoted as K-1 in Fig. 2. K-1 is capable of matching objects with the largest size. As illustrated in Table 1, the effective sizes of the search windows for features at the three levels are 128×128, 256×256, and 512×512. When the targeted size of the object in the image is small (less than 128 in each dimension), K-3 will be used in which the bottom of the recognition kernel and the bottom of the image pyramid are at the same level. The effective sizes of the search windows for the multi-level features of K-3 are 32×32, 64×64, and 128×128. Similarly, K-2 is used for searching objects with a medium size.

Since the above scheme would only allow matching at full-size, double-size, and quad-size, an additional *scaling factor S* is introduced for the further adjustment of the model size. With $S \in (0.63, 0.8, 1.0)$ and the recognition kernel applied at 3 levels (K-1, K-2, and K-3), the effective scaling factor is $S_e \in (0.63, 0.8, 1.0, 1.25, 1.6, 2.0, 2.5, 3.2, 4.0)$.

2.3. Color matching with multi-level kernels

When the recognition kernel with a certain S is placed at a certain level in the image pyramid, features at the three levels will all be matched. Since the entire matching process will likely take substantial amount of time, a coarse-to-fine strategy is devised. Namely, the search will start at the coarsest level of the kernel using colors only.

In case of K-1, since the coarsest level of the kernel is placed at Level 5 of the image pyramid, they have identical resolutions, i.e., the entire Level 5 image is the search window. Each of the RGB axes of the color histogram is quantized into 8 intervals. The frequency (number of pixels) of each of the $8 \times 8 \times 8$ bins is calculated. Clusters with high frequency cells are identified and their neighboring cells in the RGB color histogram space are treated as *similar colors* and merged into the dominant color. The color counts are then sorted to generate the five MFCs.

For K-2, the coarsest level of the kernel (32×32) is

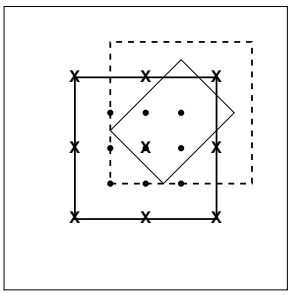


Figure 3. Searching Color Matches with K-2.

placed at Level 6 (64×64) of the image pyramid. In other words, the search window is half the size of the image and the targeted object must fit inside the search window. A search pattern as illustrated in Fig. 3 is created. At its first round, the 32×32 search window will be placed at nine locations where the "X" marks the center of the window. The nine search windows provide a very generous degree of overlap which is necessary for an accurate positioning. As in the case of K-1, the MFCs of the image within the window and the MFCs of the model will be compared. If a potential match is identified (in Fig. 3 it is at the center position), then nine new center locations (marked as "•") of the search window will be determined for a second-round search. Note, the size of the search window is not altered, whereas the distance of the search windows is reduced to half as compared to the previous round. The purpose of this second round is to obtain an improved match. The best matching window is drawn in dashed lines in Fig. 3 which indicates the northeast center position is the winner after the second round.

In general, the search algorithm for matching colors at the coarsest resolution (32×32) of the recognition kernel using K-1, K-2, and K-3 is as below:

Algorithm: (Matching color using K-1, K-2 and K-3)

begin
for $p = 1$ to 3 // K-1, K-2 and K-3
  Place the coarsest level of recognition kernel at
    Level $p + 4$ of the image pyramid
  $img\_size = 2^{p+4}$;
  $r = img\_size/32$;
  for $i = 1$ to $(2r - 1)$
    for $j = 1$ to $(2r - 1)$
      Center the search window at $(16i, 16j)$
        to match MFCs of the search window and
        the model at $S \in (0.63, 0.8, 1.0)$;
      if similar, generate a tighter bounding box;
end

2.4. Search at multiresolutions

The matching process at the subsequent two levels of the recognition kernel with higher resolutions uses a rather straight forward method. Since the location and size of the bounding box for a potential matching object is hypothesized at the previous color matching step, additional features of the recognition kernel as defined in Section 2.1 at corresponding location and levels will be examined. Since several more certainty factors C_i are introduced at each step to measure the degree of success in potential matching, a combined certainty factor $C = \prod_i C_i$ is defined. When C exceeds a selected threshold τ, the detection of an object is declared.

3. Locales for Feature Localization

Since CBIR considers objects within an image, it is common to apply some sort of *segmentation* to identify regions of objects — say patches that have about the same color [3]. However, we have shown [7] that it is more useful to use a set of *locales* to express not a complete image segmentation but instead a *feature localization*.

3.1 Feature localization vs. image segmentation

For image segmentation: If R is a segmented region,

1. R is usually connected; all pixels in R are *connected* (8-connected or 4-connected).
2. $R_i \cap R_j = \phi,\ i \neq j$; regions are *disjoint*.
3. $\cup_{i=1}^{n} R_i = I$, where I is the entire image; the segmentation is *complete*.

Object retrieval algorithms based on image segmentation permit imprecise regions by allowing a tolerance on the region matching measure [3]. This accounts for small imprecision in the segmentation, but over- and under-segmentations are common because a satisfactory image segmentation based on low level features is unattainable. A more effective and attainable process than image segmentation is a coarse localization of image features based on proximity and compactness.

Definition: A *locale* $\mathcal{L}_f$ is a local enclosure of feature f.

A locale $\mathcal{L}_f$ uses blocks of pixels called *tiles* as its positioning units, and has the following descriptors:

1. Envelope L_f — a set of tiles representing the locality of $\mathcal{L}_f$.
2. Geometric parameters — mass $M(\mathcal{L})$, centroid $\mathbf{C}(\mathcal{L})$ and eccentricity $E(\mathcal{L})$.
3. Color, texture, and shape parameters of the locale. For example, locale chromaticity, elongation, and locale texture histogram.

Initially, an image is subdivided into square tiles (e.g., 8×8 or 16×16). While pixel is the building unit for image segmentation, tile is the building unit for feature localization. Tiles group pixels with similar features within their extent, and are said to have feature f if enough pixels in them have feature f (e.g., 10%). Tiles are necessary for good estimation of initial object-level statistics and representation of multiple features *at the same location*. However, locale geometric parameters are measured in pixels, not tiles. This preserves feature granularity. Hence, feature localization is not merely a reduced-resolution variation on image segmentation.

After a feature localization process the following can be true:

1. $\exists f : \mathcal{L}_f$ is *not always connected*.
2. $\exists f \exists g : \mathcal{L}_f \cap \mathcal{L}_g \neq \phi,\ f \neq g$; locales are *non-disjoint*.
3. $\cup_f \mathcal{L}_f \neq I$, *non-completeness*; not all image pixels are represented.

Locales are generated using a dynamic 4×4 overlapped pyramid linking procedure. On each level parent nodes compete for inclusion of child nodes in a fair competition. Image tiles are the bottom-level child nodes of the pyramid, and locales are generated for the entire image when the competition propagates to the top level [6]. Fig. 4 shows the color locales generated from a sample image.

4. Experimental Results

As a testbed, we developed the C-BIRD system (Content-Based Image Retrieval from Digital libraries). The database consists of over 1,500 test images and several dozens of video clips. Several searching methods are supported:

- Keyword — Keyword information is stored in the database in the conventional manner. Matching is trivial if correct keywords are provided in the query.
- Color Percentage and Layout — Similar to QBIC [8], a color palette and a square drawing area are provided for the user to draw a sketch of the desired color layout. In addition to color percentage, color axes are also compared. It handles images of multiple scales.
- Illumination Invariance — Illumination change can dramatically alter the color measured by camera RGB sensors, from *pink* under daylight, to *purple* under fluorescent lighting, for example. To deal with illumination change from the query image to different database images, each color channel band of each image is first normalized, and then compressed to a 36-vector [4].

Figure 4. The Locales generated for a sample image shown at the lower-right corner. Every subimage shows a different Locale which is composed of the color tiles.

- Texture Layout — Similar to color layout search, this allows the user to draw the desired texture distribution. Available textures are zero density texture, medium density and high density textures.
- Model — User can browse through the selection of models and make a choice, or use the drawing tool to select an example from any part(s) of the images in the database. The multi-level recognition kernel of the model is matched against images in the database. Locale-based match is also supported.

4.1. Search using recognition kernel

Fig. 5 shows some results from Search by Model for one of the white books in C-BIRD. Features at all three levels of the Recognition Kernel are used. Not only the book but also its color and edge distributions and orientations are located. The last two outputs are a similar book, although listed, their certainty factors are very low (in the 20% range).

4.2. Search using locales

The screening tests that are applied to locales in order to generate assignments and validate them are:

- Color locale-based screening tests:
  - Illumination Color Covariant Screening

Figure 5. Search Using Recognition Kernel.

  - Chromaticity Voting
  - Elastic Correlation

- Estimation of Image Object Pose
- Texture Support
- Shape Verification

The idea of color covariant matching is to realize that colors may change, from model to target, since the lighting may easily change. A diagonal model of lighting change states that the entire Red channel responds to lighting change via an overall multiplicative change, as do the Green and Blue channels each with its own multiplicative constant [4].

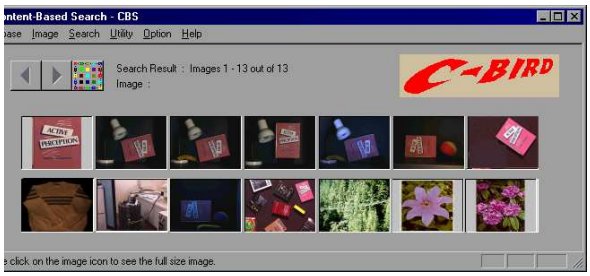
Locales *vote* on the correct lighting change, since each assignment of one model locale color to a target one implies a diagonal lighting shift. Many votes in the same cell of a voting space will imply a probable peak value for lighting change. Using the chromaticity voting scheme, all image locales are paired with all model locales to vote for lighting change values in a voting array.

The feasibility of having an assignment of image locales to model locales is evaluated using the estimated chromaticity shift parameters by a type of *elastic correlation* [6].

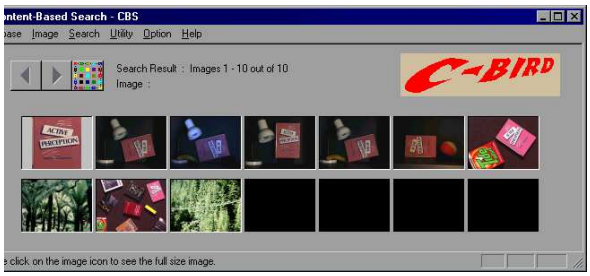
The pose estimation method uses geometrical relationships between locales for establishing pose parameters. For that reason it has to be performed on a feasible locale assignment. Locale spatial relationships are represented by relationships between their centroids. The number of assigned locales is allowed to be as few as two, which is enough geometry information to drive estimation of a rigid body 2D displacement model with four parameters to recover: x, y translation, rotation $\mathbf{R}$, and scale s [4].

Results of pose estimation are both the best pose parameters for an assignment and the minimization objective value, which is an indication of how well the locales assignment

(a)



(b)



(c)

Figure 6. Search result for the pink book model with illumination invariance support. (a): search results using pose estimation only; (b): search results using pose estimation and texture support; (c): search results using GHT shape verification.

fits using the rigid-body displacement model. If the error is within a small threshold, then the pose estimate is accepted.

The texture support screening test is utilizing a variation of histogram intersection technique, where the texture histograms of locales in the assignment are intersected. If the intersection measure is higher than a threshold then the texture match is accepted.

The final match verification process is shape verification by the method of Generalized Hough Transform (GHT) [2]. The GHT is robust with respect to noise and occlusion. Performing a full GHT search for all possible rotation, scale and translation parameters is computationally very expensive and inaccurate. Such a search is not feasible for large

databases. However, after performing pose estimation we already know the pose parameters, and we can apply them to the model reference point to find the estimated reference point in the database image. Hence, the GHT search reduces to a mere confirmation that the number of votes in a small neighborhood around the reference point is indicative of a match. This GHT matching approach takes only a few seconds for a typical search. The reference point used is the model center since it minimizes voting error caused by errors in edge gradient measurements.

Once we have shape verification, the image is reported as a match, and its match measure Q returned, if Q is large enough. After obtaining match measures Q_i for all images in the database, the Q_i measures are sorted according to decreasing value. The number of matches can further be restricted to the top k if necessary. An estimate of the correct illumination change follows from correct matches reported.

Fig. 6 shows some search results for the pink book in C-BIRD.

4.3. Video locales

We have also extended the notion of image locales to *video locales* [1].

Definition: A *video locale* is a sequence of image feature locales that share similar features in the spatio-temporal domain of videos.

Like locales in images, video locales have their color, texture and geometric properties. Moreover, they capture motion parameters such as the motion trajectory and speed, as well as temporal information such as the life-span of the video locale and its temporal relationships with respect to other video locales. Since video proceeds in small time steps, we can also expect to develop new locales from ones already known from previous video frames more easily than simply starting from scratch in each frame.

Fig. 7 shows that in fact while speeding up the generation of locales substantially, very little difference occurs in generation of locales from each image ("Intra-frame") and from predicting and then refining the locales ("Inter-frame").

It was shown [1] that the Inter-frame algorithm is always much faster than the Intra-frame one. Moreover, video locales provide an effective means towards real-time video object segmentation and tracking.

5. Conclusion

Content-based image retrieval (CBIR) is an important issue in the research and development of digital libraries which usually relies on large multimedia databases. This paper presented two methods for CBIR using a multi-level recognition kernel and locales for feature localization. The deployment of recognition kernel and locales in a pyramidal (multiresolution) framework facilitates multi-level abstraction of the model and is shown to improve the matching efficiency and quality.

(a) (b) (c)

Figure 7. Intra-frame and Inter-frame video locales algorithm results: (a) original images; (b) intra-frame results; (c) inter-frame results.

References

[1] J. Au, Z.N. Li, and M. Drew. Object segmentation and tracking using video locales. In *Proc. Int. Conf. on Pattern Recognition (ICPR 2002)*, pages 54–547, 2002.

[2] D. Ballard. Generalizing the Hough transform to detect arbitrary shapes. *Pattern Recognition*, 13(2):111–122, 1981.

[3] C. Carson, S. Belongie, H. Greenspan, and J. Malik. Blobworld: Image segmentation using expectation-maximization and its application to image querying. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(8):1026–1038, 2002.

[4] M. Drew, Z.N. Li, and Z. Tauber. Illumination color variant local-based visual object retrieval. *Pattern Recognition*, 35(8):1687–1704, 2002.

[5] D. Forsyth and J. Ponce. *Computer Vision: A Modern Approach*. Prentice Hall, 2003.

[6] Z.N. Li and M. Drew. *Fundamentals of Multimedia*. Prentice Hall, 2004.

[7] Z.N. Li, O. Zaïane, and Z. Tauber. Illumination invariance and object model in content-based image and video retrieval. *Journal of Visual Communication and Image Representation*, 10(3):219–244, 1999.

[8] M. Flickner, et al. Query by image and video content: the QBIC system. *IEEE Computer*, 28(9):23–32, 1995.

[9] A. 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*, 22(12):1349–1380, 2000.

[10] D. Tao, X. Tang, X. Li, and X. Wu. Asymmetric bagging and random subspace for support vector machines-based relevance feedback in image retrieval. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 28(7):1088–1099, 2006.