Segmentation based Image Retrieval

Andreas Siebert
Department of Computer Science
The University of British Columbia

ABSTRACT

We have been developing an image retrieval system, called MIPS (Multiscalar Image Processing and Retrieval System), for use in uncontrolled environments.

On insertion into the image database, the images are automatically segmented into homogeneous regions. Generic features are computed and stored for each segment. Specifically, we maintain not only geometric and photometric attributes but also simple spatial information for each extracted region.

This approach asks the user to construct queries in terms of the given primitives, i.e. regions and their spatial relations. Preliminary results show that the success of the system depends on how well the images can be modelled by homogeneous regions, on how useful the generic features are for the given application, and on the knowledge that the user puts into the formulation of the queries.

A fully automatic segmentation algorithm is of paramount importance. We have designed an algorithm called Perceptual Region Growing that combines region growing, edge detection, and perceptual organization principles, without resorting to any kind of high level knowledge or interactive user intervention. Decision thresholds and quality measures are directly derived from the image data, based on image statistics. Search through critical parameter spaces is the key idea to cope with noise in uncontrolled environments. The dynamics of the region growing process is constantly monitored and exploited.

Keywords: image retrieval, segmentation, region growing

1. INTRODUCTION

The areas of application we have in mind for our image retrieval system are collections of images from various domains, e.g. digitized stock photos, vacation photos, or images in digital libraries. The images we currently experiment with are taken from the internet. This implies that we don't have any control of the scene geometry, the lighting conditions, the sensor characteristics, the number and kind of objects in the scene, and so on. Therefore, good performance in uncontrolled environments and genericness, i.e. domain independence as far as possible, are important design goals.

Unlike early retrieval systems, we don't just look at global image properties but assume that an image contains important local structure. Generally, the spatial position of and the relations between local image structures contain significant information. We compute simple spatial information (position of center of gravity of a segment, bounding box, orientation) that allow to check for spatial relationships at query time (e.g. adjacency relations). Therefore, we expect our system to give better results than purely histogram based systems that don't maintain any spatial information.

The "atomic" structure, or primitive, of our approach is the region. Images are segmented into homogeneous regions that do not necessarily cover the whole image. All features and spatial relations are based on the extracted regions. As a consequence, a user must construct each retrieval query in terms of regions and their features. The computation of the regions is based exclusively on the intensity values. A region is the result of a region growing process and it is limited to a large degree by a strong contour. No external constraints or models, e.g. smooth contours, are employed. The underlying critical assumption is that many real world objects give rise to homogeneous regions in images, which in turn capture a lot of information about the objects.

In order to achieve our goals, we use the system architecture shown in figure 1. The system is centered around a conventional alphanumerical database. At image insertion time, the new image is fully automatically segmented into regions. For each region, several generic features (color, size, contour strength, moments, etc.) are computed and stored in the database. We do not know all properties that a user might be interested in. Therefore, the general idea is to use properties that are as generic as possible. The user can enjoy all the flexibility provided by a querying

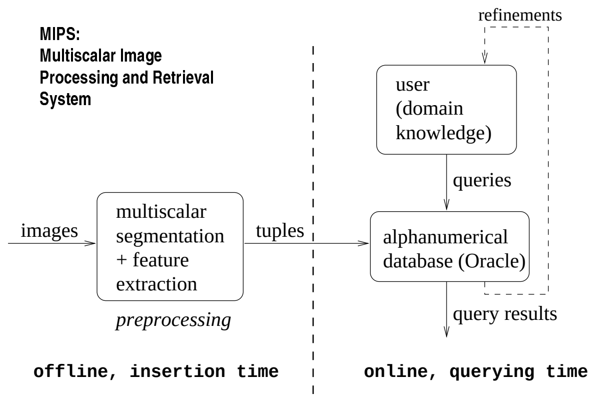

Figure 1. System architecture

language like SQL. Queries can be refined based on the outcome of previous queries. In our domain independent approach, it is the user who at this point introduces knowledge into the system by formulating the queries according to the problem at hand. It is straightforward to provide query support tools, like e.g. a graphical interface.

2. PREVIOUS WORK

Previous work can be split into three categories: segmentation algorithms, perceptual organization, and image retrieval systems.

Haralick and Shapiro[12] summarize the more traditional segmentation techniques. A more recent review[23] includes references to segmentation methods based on fuzzy logic, neural networks, and color. Ohlander[22] was an early attempt to dynamically compute the segmentation parameters from the image data. Kees et al.[16] spawned a whole family of segmentation algorithms based on active contour models ("snakes"). Global segmentation schemes usually employ a statistical framework.[10,11] Zhu and Yuille[36] try to combine active contours, region growing, and minimal description length methods into a unified segmentation scheme. Leclerc[17] has obtained good segmentation results by modeling images as the corruption of underlying piecewise smooth images. There is also a large number of schemes that aim at integrating edge detection with region growing.[3,6]

In perceptual organization, two themes are relevant to this paper: saliency and shape completion. Ullman[33] introduced *subjective contours* "that are perceived when the visual system fills in the gap between distinct edges". Kanizsa[15] provided some famous figures to demonstrate such subjective contours. Sha'ashua[28] defined structural saliency in terms of the length and smoothness of line segments. Recently, Williams and Jacobs[35,14] proposed stochastic completion fields to deal with the completion shape and salience problem. Marr[21] suggested that the first stage in human perception happens without recurrence to high-level knowledge. He proposed a hierarchical representational framework whose first stage he called *primal sketch*, which employs primitives like zerocrossings, blobs, edges, groups, boundaries.

QBIC,[8] Photobook,[24] and Virage[2] were among the first general image retrieval systems. Wavelet based systems were proposed by Jacobs[13] and Smith.[30] White[34] introduced a low level query language for visual pattern matching. Some more specialized retrieval systems include Bach's visual information management system for faces,[1] Forsyth's and Fleck's "Finding Naked People[9]", and Soffer's and Samet's retrieval in symbolic image databases.[31]

3. SEGMENTATION ALGORITHM

For a good performance of the overall system the quality of the segmentation algorithm is of paramount importance. The most critical design goals of the algorithm can be summarized as follows:

1. The algorithm is to extract homogeneous regions. By definition, regions have closed contours, even where edge detectors indicate gaps in the edges.

2. The algorithm is to perform in uncontrolled environments. No specific domain knowledge or models can be assumed.

3. The algorithm is to run fully automatic. No operator interference or hand tuning of parameters is acceptable.

4. The algorithm should provide some quality measures for the extracted regions.

We have designed an algorithm called Perceptual Region Growing (PRG) that combines region growing, edge detection, and perceptual organization principles, leading to a data driven approach similar to exploratory data analysis.[7] PRG is an improvement of an earlier algorithm called Dynamic Region Growing.[29] It continually monitors the growth process and derives critical, local parameters dynamically from the growth analysis. Note that the resulting image descriptions, are purely appearance based.

The key elements of PRG are as follows: PRG identifies seed points which are micro regions of constant intensity. These are expanded circularly by merging adjacent pixels if the difference in intensity between the region and the adjacent pixel is less than a certain threshold ϵ_i. This ϵ_i, a critical parameter in any region growing algorithm, is iteratively incremented, resulting in a dynamic growth of each region. A region is assumed to be limited by a strong contour. Therefore, PRG monitors the *contour cover* of each region, i.e. the percentage of contour points of a region that are classified as being *strong gradient points* (or *busy* points). Once a region's contour cover is larger than a suitable threshold, the growth process continues until an *overspill* occurs. The region as it is immediately before the overspill occurs is declared the desired region.

Figure 3 presents the pseudo code for PRG. The details of the algorithm are as follows.

First, some image statistics are computed. In particular, the contrast average and deviation is determined. Here, the contrast is simply defined as the average difference between intensities of adjacent pixels. The minimum region size is derived from the image size. In our experiments, we set it to 0.25% of the input image size for the first iteration. Without a minimum region size, a large number of tiny regions might be produced, especially in noisy or textured image areas.

Figure 2. (a) Fields (b) Edges extracted with Canny edge detector (c) Strong Gradient Points Image (edge cover 33%). The gray level of the pixels corresponds to the magnitude of the gradient.

The *strong gradient points image* is a binary image where all pixels where the gradient exceeds a certain threshold $\Theta_{strong\_gradient}$ are set to one, and zero elsewhere. We implemented both a First Derivative of a Gaussian with a small scaling parameter ($\sigma = 1.0$), and a Sobel edge detector. These two operators lead to similar sets of busy points and therefore to similar segmentation results. $\Theta_{strong\_gradient}$ is derived from the gradient average $\mu_{gradient}$ and deviation $\sigma_{gradient}$. In our experiments, we set $\Theta_{strong\_gradient} = \mu_{gradient} - 0.1 * \sigma_{gradient}$. This resulted in an *Edge Cover*, i.e. in a percentage of pixels classified as strong gradient points, ranging roughly between 25% and 45%. An examples is given in figure 2c. Note that the edge cover contains many more pixels than what a thresholded edge detector typically delivers. This is a desirable feature for the gap filling and overspill detection later on. Figure 2b, showing the results of the Canny edge detector (edges thresholded and linked), clearly demonstrates the occurrence of gaps and spurious edges that make the step from edges to regions a hard one in noisy real world images.

Seed regions are small, homogeneous regions with an intensity variance close to zero. The whole image is searched for the seed region with the smallest variance in a 3×3 window. Ties are broken by considering 5×5 and 7×7 windows. If there is still a tie, the seed region closest to the image center is selected.

```
compute image statistics
compute gradient image, gradient statistics
/* iteration 1 */
while seed regions can be found
      find next seed region R_i(0)
      ε = ε_low
      /* phase 1 */
      while ε ≤ ε_high and Contour Cover < Θ_cc
            grow region R_i(ε) concentricly
            ε = ε + 1
      end while
      /* phase 2 */
      while ε ≤ ε_high and Contour Cover ≥ Θ_cc
                        and no overspill
            grow region R_i(ε) concentricly
            ε = ε + 1
      end while
      compute contour features
      compute region features
      if   size(R_i) < Minimum Region Size or
            Contour Cover < Θ_cc
      then discard R_i
end while

compute region statistics
select best regions

/* iteration 2 */
increase Θ_cc
decrease Minimum Region Size
keep selected regions
repeat iteration 1
```

- image statistics:
  $\mu_{intensity}$, $\sigma_{intensity}$,
  $\mu_{contrast}$, $\sigma_{contrast}$,
  Minimum Region Size
- gradient statistics:
  $\mu_{gradient}$, $\sigma_{gradient}$,
  $\Theta_{strong\_gradient}$, *Edge Cover*
- contour features:
  $length_{contour}$, $strength_{contour}$
- region features:
  $\mu_{intensity}$, $\sigma_{intensity}$,
  size, *circularity*, *color*,
  bounding box, *center of gravity*,
  orientation, *second moment*,
  simplified contour
- region statistics:
  $\mu_{\sigma_{intensity}}$, $\sigma_{\sigma_{intensity}}$,
  μ_{size}, σ_{size},
  $\mu_{contour\_strength}$, $\sigma_{contour\_strength}$,
  $\mu_{circularity}$, $\sigma_{circularity}$

Figure 3. Perceptual Region Growing PRG

ϵ is an integer valued parameter that controls region growing: A pixel p with intensity I_p that is spatially adjacent to the region $R(\epsilon)$ having average intensity μ_ϵ is merged with $R(\epsilon)$ if $|\mu_\epsilon - I_p| \leq \epsilon$. The starting value for ϵ, i.e. ϵ_{low}, is set to the rounded value of the average contrast, $\mu_{contrast}$, which turns out to be a reasonable estimate for the image noise.

Regions grow *concentricly*, meaning that the adjacent pixels to be merged simultaneously are *all* those neighboring the current region. We were careful to avoid scan line bias in our implementation. Note that the seed point selection strategy, combined with concentric growth, guarantees the rotational invariance of PRG.

The growing of each region is divided into two phases. In phase one, the region growth continues until enough of the region's contour points are classified as strong gradient points, i.e. until *Contour Cover* $\geq \Theta_{cc}$. In phase two, maintaining a sufficient Contour Cover, the growth continues while no *overspill* occurs. An overspill is defined as a "discontinuity", i.e. an abrupt change, in either region size, average intensity, intensity variation, or circularity, where circularity is defined to be the ratio of the region size and the square of the contour length. The necessity of overspill detection stems from what Haralick and Shapiro[12] referred to as region chaining: in region growing, two distinct regions might be merged if they are not completely separated by an edge. Fortunately, overspill detection is a fairly robust operation since the changes of the region features are usually large.

Figure 4 shows a typical scenario: An initial region is grown in several steps, i.e. in several increments of ϵ. The corresponding growth behavior is shown in fig. 4d: The values of region properties like size, average intensity (mu), intensity variation (sigma) and circularity, as well as contour properties like contour strength (cs) and contour cover, are shown versus the growth steps. It can be seen that phase one is over at step six when the threshold Θ_{cc}=60% is surpassed. The overspill occurs at step eight and is clearly characterized by a large increase in region size, a sharp decrease in circularity, and a fairly large decrease in average intensity and contour cover. The region intensity deviation has also increased significantly, while the contour strength is almost constant. We define the contour strength simply to be the average intensity difference between the pixels on the region contour and the 4-neighborhood adjacent pixels that are not part of the region.

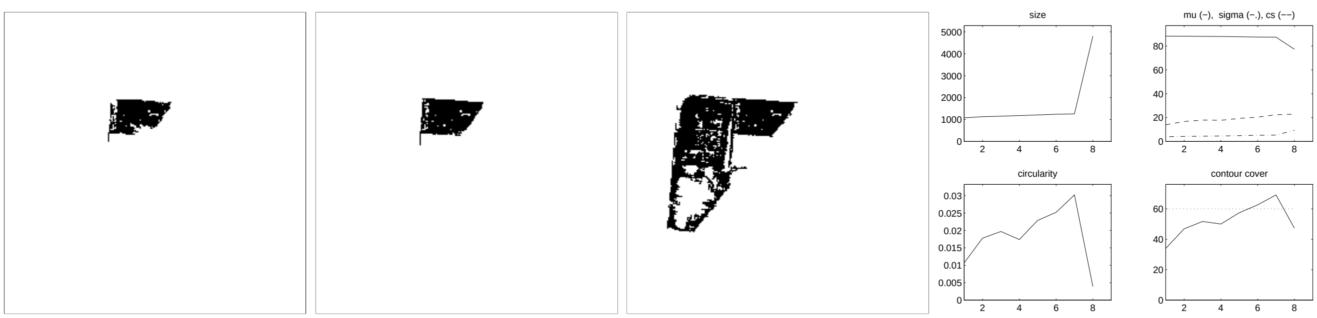

Figure 4. Region growing: (a) initial region (b) region after six growth steps (c) region after seven growth steps (overspill) (d) growth behavior: region and contour features versus growth steps

At the end of phase two, PRG computes additional region features. These are color, bounding box, center of gravity, second moments, and orientation of each region. Also, we use a simplified Douglas-Peucker algorithm[5] to approximate the region contours by a simple polygon. In the current implementation, we allow a maximum of 10 vertices, but fewer vertices are used if the approximation error is small.

Regions that don't meet the minimum size requirement are discarded, trading efficiency in time and memory versus accuracy. Regions which never reached the required edge cover are also removed.

We have experimented with criteria to keep only the "best" regions of an image. In the absence of any specific domain knowledge, PRG depends on the generic features to evaluate the quality of a region. More precisely, PRG computes the mean and deviation of the intensity deviation, size, contour strength, and circularity of the extracted regions. The deviations are added up, and those with a positive sum are declared the "best" regions. It is straightforward to give weights to or to add or drop certain features for a modified meaning of "best" region.

Keeping only a few of the extracted regions opens a recursive approach for PRG: the first iteration can be repeated for those areas of the image that are not part of a selected region yet. For the second iteration, some parameters could be modified. We implemented a scheme where in iteration two the minimum region size is decreased by 25% relative to iteration one, and the threshold for the required contour cover is increased from 60% to 70%. The latter modification has the effect of increasing the extracted region's size as well as the probability that the region is discarded for not reaching the required contour cover.

The processing steps of PRG are demonstrated in figure 5. Unlabeled pixels are shown in black. The extracted regions are rendered in arbitrary intensities.

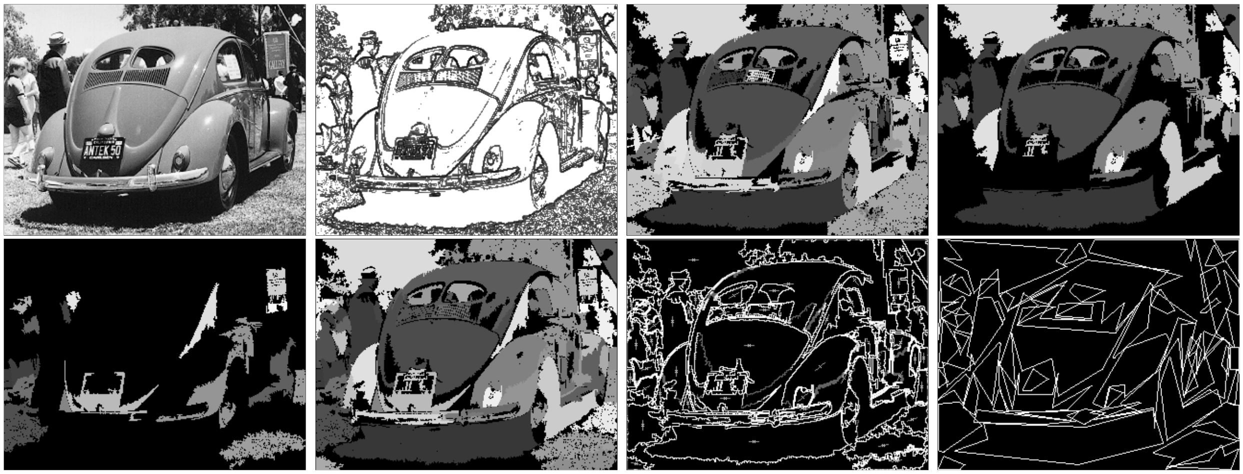

Figure 5. (a) Car (b) Strong Gradient Points Image (edge cover 37%) (c) All regions extracted in iteration one (d) Best regions extracted in iteration one (e) Best regions extracted in iteration two (f) All regions extracted in iterations one and two (g) Contours of extracted regions, plus markers for center of gravity and orientation (h) Polygon approximation of contours

4. PRG: EXPERIMENTAL RESULTS

In order to demonstrate the capabilities and limits of PRG, we present several segmentation examples in figures 6 and 7. The figures show the original images, the results of a one step segmentation, the best regions selected from iteration one, and the results of two iterations.

Clearly, the segmentation is not exhaustive, leaving textured regions as well as tiny regions that do not meet the minimum size requirement unlabeled. Textured regions do not fit the assumed model of an image a a set of homogeneous regions, and therefore should be handled by a separate texture segmentation algorithm.[19,24]

The contours of the extracted regions are often rugged since no smoothness constraints are incorporated. Contour smoothing could be done as a postprocessing step, but this seems unnecessary in the given image retrieval context.

If there are large regions of uniform intensity in an image, then they are extracted by PRG. More interesting, we can ask how well the extracted regions describe the given image, i.e. how well the image is modelled as a set of homogeneous regions and their attributes. The answer depends subjectively on both the images and the application. The extracted regions in the *Fields* image (fig. 6, third row) seem to capture well the image content, but this is certainly not the case in the *Barbara* image (fig. 6, fourth row).

In an uncontrolled environment, there cannot be any assurance that an extracted region corresponds to a semantically meaningful object (or object part). Often, entities we like to think of as a semantic unit and of uniform intensity, give rise to seemingly arbitrary regions of various intensity, see e.g. rock and sky in the *Ayers Rock* image (fig. 6, sixth row).

The main problems we encountered can be summarized as follows:

1. Shading, shadows, and highlights cause optically uniform real world objects to be mapped to a wide range of intensities.

2. Small but visually important features (e.g. eyes) may not be extracted.

Figure 6. (a) Boston; Building; Fields; Barbara; Brandenburger Tor; Ayers Rock (b) All regions extracted in iteration one (c) Best regions extracted in iteration one (d) All regions extracted in iterations one and two

Figure 7. (l) Woman[36,4] (r) Arc de Triomphe

3. Textured regions are not extracted. However, a complimentary texture segmentation algorithm can be added in the obvious way.

4. Unwanted merging of regions is unlikely but it still happens.

We did a few experiments to evaluate the effectiveness of simple filtering methods to support PRG. The hope was that by suppressing details and noise, low-pass filtering would improve the segmentation results. However, both Gaussian smoothing and median filtering introduce artifacts. Gaussian smoothing blurs edges, thereby creating a transition zone between two regions that might be labeled as a new region. Similarly, median filtering leads to shapes that are not present in the original data. Figure 8 demonstrates the aforementioned problems. We haven't yet experimented with more advanced smoothing schemes, like anisotropic diffusion.[25]

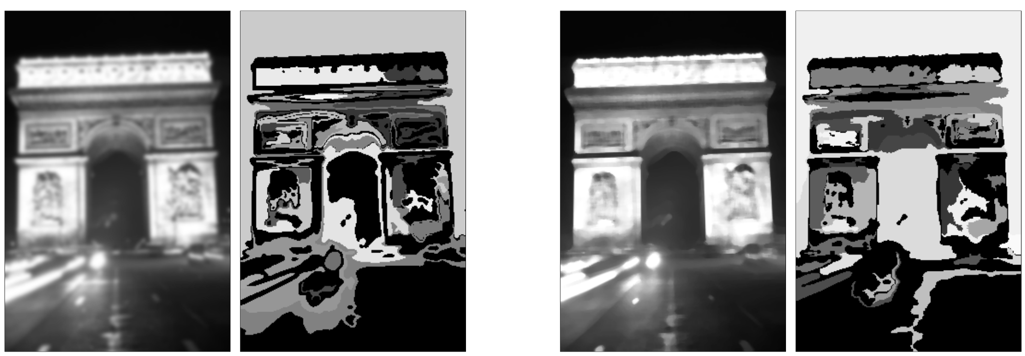

Figure 8. Arc de Triomphe: (l) Gaussian smoothing, sigma=2.0 (r) Median filtering, window size 7×7

In image compression, there are image structure sensitive methods that first segment images into homogeneous regions which then are coded.[26] The compression rate achieved by such a method provides a way to quantify the adequateness of the underlying image model. However, unlike image retrieval, image compression does not require the extracted regions to correspond to meaningful parts or objects. Fitting regions to intensity values is not unlike the statistical modeling that tries to split data into data and noise (or *fit* and *residual*).[7]

Images that seem to lend themselves naturally to be modelled as sets of homogeneous regions are cartoons and clip art images. However, images with perfectly homogeneous regions (i.e. $\sigma_{intensity} = 0$) are generally *not* well segmented by PRG. The reason is that PRG depends on the growth process for proper overspill detection, while extraction of a perfectly homogeneous region consists of just one step. Interestingly, by adding random noise to the original image, PRG can be made to perform well. An example is given in fig. 9. It can be observed that the left part of the belly of the left penguin is merged with the ground in the segmentation of the original image. By contrast, if noise is added, that belly is correctly labeled as a separate region.

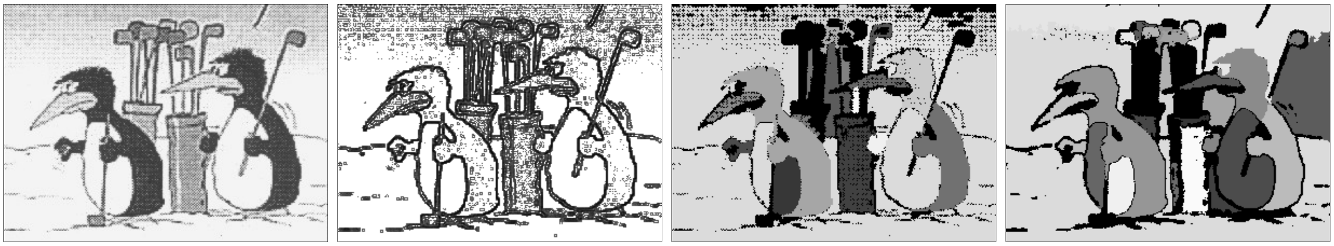

Figure 9. Cartoon example (Uli Stein penguins[32]): (a) original image (b) Strong Gradient Points Image (c) segmentation, original image (d) segmentation, noise added

5. QUERIES

In their most elementary form, queries take the form of SQL statements where the user has to construct a query in terms of the primitives stored in the database, i.e. in terms of the regions of uniform intensities and their attributes which have been computed for each image at insertion time. Such a SQL user interface certainly is too primitive and too uncomfortable for an operational database to be used by untrained users, but for now it is flexible and allows us to evaluate the system without compromising performance. The user's domain knowledge is expressed directly in the form of attributes and thresholds employed in a query. Adding a graphical interface to support the formulation of queries and to enable query by example should be straightforward. Another useful addition is to store previous queries for reuse and modification, as a kind of primitive knowledge modul.

Let's have a closer look at some examples.

1. The user wants to retrieve images of oceanfront cities. One way to formulate a query is to ask for all images that have a large blue region at the top (=sky), a large blue region at the bottom (=sea), and some smaller vertical structures in between (=skyscrapers). This query would return the *Boston* image in fig. 6 above, but it makes clear the problems of appearance based approaches: The query doesn't capture the semantics of "oceanfront cities". Therefore, it fails for images of oceanfront cities where the sky or the sea is not blue. The user could reformulate the query without or with relaxed color constraints, thereby increasing the number of hits (i.e. the *recall*) at the expense of more false positives (i.e. the *precision*). Note that the burden to provide adequate ranges for the values of "blue", "large", "top", etc. is also on the user.

2. The user wants to retrieve images of arches. The query might be to retrieve all images containing several grayish, vertical regions parallel to each other (=columns) and a grayish, horizontal region on top of that (=gable). This query would return the *Brandenburger Tor* image in fig. 6 above, but it fails for the *Arc de Triomphe* in fig. 7. Here, the reason is that the latter contains too much detail to be reduced to regions corresponding to whole columns.

3. The user wants to retrieve faces. A query might try to retrieve regions that could represent eyes, noses, etc. Such a query would fail miserably. PRG usually doesn't extract such special features. The example images *Barbara* and *Woman* show that faces are often segmented into a left half and a right half, induced by the lighting and the curved shape of faces. Therefore, it is more promising to search for skin colored regions.[9]

Note that a SQL query returns all records meeting the constraints as defined in the query, without ranking them. Smith[30] has proposed a simple, weighted metric for ranking matches based on both spatial relations and feature values. We haven't implemented a metric yet. Since different users need different metrics, the best approach seems to be to give the user the opportunity to craft a metric at query time.

To summarize, appearance based queries come with many surprises. Image projections of real world objects tend to be not as homogeneous as we often think they are. They may contain unexpected details, shadows, counterintuitive intensity patterns, and so on. On the other hand, in uncontrolled, unconstrained environments, the image data is all we have, and it seems reasonable to appeal to human users to build imaginative queries that hopefully capture the appearance of the objects to be retrieved.

6. CONCLUSIONS AND FUTURE DIRECTIONS

We have proposed an image retrieval system architecture that depends on the automatic segmentation of images at insertion time. The segmentation algorithm, Perceptual Region Growing, delivers a description of an image in terms of regions, providing local structure.

The generality of the approach implies a lack of specificity in the sense that the image primitives and the generic features extracted sometimes do not capture information that is in the original image and needed for some applications. While the traditional approach in image processing has been to define the problem domain and then to optimize the algorithms accordingly, our approach has been rather to come up with a generic algorithm and then to delineate the cases where it is not appropriate.

The appearance based primitives approach side-steps the gap between syntax and semantics that has haunted so many vision systems. For example, the system does not know what a "house" is, it has no model of it. Instead, the user has to construct queries for the possible appearance of a house, composing it of homogeneous regions. Such queries implicitly incorporate assumptions about the scene geometry, the illumination, and the details, which might harbor some unexpected features.

In an uncontrolled environment, no a priori knowledge about the different scales of the image objects is available. Grouping, i.e. aggregation of image primitives into larger units, is one, albeit expensive, way to overcome the detail problem mentioned above.[27] Alternatively, scale detection can be done in a multiscale representation,[18] for example wavelet-based,[20] which is what we are working on.

PRG needs to be complemented by a texture segmentation algorithm for those image areas that cannot be modelled adequately as homogeneous regions. We pointed out major problems with shading, shadows, and highlights. Tests on a larger image database will allow a more careful evaluation of our approach, especially when compared to histogram based methods.

ACKNOWLEDGEMENTS

I would like to thank B. Woodham for his valuable hints and suggestions.

REFERENCES

1. J. Bach, S. Paul, R. Jain, "A Visual Information Management System for the Interactive Retrieval of Faces", *IEEE Trans. on Knowledge and Data Engineering*, Vol.5, No.4, Aug.1993.
2. J. Bach et al., "The Virage Image Search Engine", *Proc. of SPIE Conf. on Storage and Retrieval for Still Image and Video Databases IV*, San Jose, 1996.
3. C. Chu, J. Aggarwal, "The Integration of Image Segmentation Maps Using Region and Edge Information", *IEEE Trans. PAMI*, Vol.15, No.12, pp.1241-1252, Dec.1993.
4. D. Comaniciu, P. Meer, "Robust Analysis of Feature Spaces: Color Image Segmentation", *CVPR '97*, Puerto Rico 1997.
5. D. Douglas, T. Peucker, "Algorithms for the reduction of the number of points required to represent a line or its caricature", *The Canadian Cartographer*, Vol.10, No.2, 1973.
6. R. Falah, P. Bolon, J. Cocquerez, "A Region-Region and Region-Edge Cooperative Approach of Image Segmentation", *ICIP-94*, 1994.
7. J. Elder, D. Pregibon, "A Statistical Perspective on Knowledge Discovery in Databases", in: U. Fayyad et al. (eds.), Advances in Knowledge Discovery and Data Mining, AAAI/MIT Press 1996.
8. M. Flickner et al., "Query by Image and Video Content: the QBIC System", *IEEE Computer*, Vol.28, No.9, pp.23-32, Sept.1995.
9. D. Forsyth, M. Fleck, "Finding Naked People", *Proc. 4th European Conf. on Computer Vision*, 1996.
10. S. Geman, D. Geman, "Stochastic Relaxation, Gibbs Distributions and the Bayesian Restoration of Images", *IEEE Trans. PAMI*, Vol.6, No.6, pp.721-741, Oct.1984.
11. D. Geman et al., "Boundary Detection by Constrained Optimization", *IEEE Trans. PAMI*, Vol.12, No.7, pp.609-628, July 1990.

12. R. Haralick, L. Shapiro, "Image Segmentation Techniques" *Computer Vision, Graphics, and Image Processing*, Vol.29, pp.100-132, 1986.
13. C. Jacobs, A. Finkelstein, D. Salesin, "Fast Multiresolution Image Querying", *Computer Graphics Proc.*, pp.277-286, 1995.
14. D. Jacobs, "Robust and Efficient Detection of Salient Convex Groups", *IEEE Trans. PAMI*, Vol.18, No.1, pp.23-37, Jan.1996.
15. G. Kanizsa, "Organization in Vision", Praeger, New York 1979.
16. M. Kass, A. Witkin, D. Terzopoulos, "Snakes: Active Contour Models", *ICCV-87*, London 1987.
17. Y. Leclerc, "Constructing Simple Stable Descriptions for Image Partitioning", *Int.Journal of Computer Vision*, Vol.3, No.1, pp.73-102, 1989.
18. T. Lindeberg, J. Eklundh, "Scale Detection and Region Extraction from a Scale-Space Primal Sketch", *ICCV-90*, Osaka 1990.
19. B. Manjunath, W. Ma, "Browsing Large Satellite and Aerial Photographs", *IEEE Int.Conf.on Image Processing*, Lausanne 1996.
20. S. Mallat, S. Zhong, "Characterization of Signals from Multiscale Edges", *IEEE Trans. PAMI*, Vol.14, No.7, pp.710-732, July 1992.
21. D. Marr, "Vision", Freeman 1982.
22. R. Ohlander, K. Price, R. Reddy, "Picture Segmentation Using a Recursive Region Splitting Method", *Computer Graphics and Image Processing*, Vol.8, pp.313-333, 1978.
23. N. Pal, S. Pal, "A Review on Image Segmentation Techniques", *Pattern Recognition*, Vol.26, No.9, 1993.
24. A. Pentland, R. Picard, S. Sclaroff, "Photobook: Content-Based Manipulation of Image Databases", *Int.Journal of Computer Vision*, Vol.18, No.3, 1996.
25. P. Perona, J. Malik, "Scale Space and Edge Detection Using Anisotropic Diffusion", *IEEE Trans. PAMI*, Vol.12, No.7, pp.629-639, July 1990.
26. M. Reid, R. Millar, N. Black, "Second-Generation Image Coding: An Overview", *ACM Computing Surveys*, Vol.29, No.1, pp.3-29, March 1997.
27. E. Saund, "Adding Scale to the Primal Sketch", *IEEE Conf. on Computer Vision and Pattern Recognition*, San Diego 1989.
28. A. Sha'ashua, S. Ullman, "Structural Saliency: The Detection of Globally Salient Structures Using a Locally Connected Network", *ICCV-88*, Tampa 1988.
29. A. Siebert, "Dynamic Region Growing", *Vision Interface '97*, Kelowna 1997.
30. J. Smith, "Integrated Spatial and Feature Image Systems: Retrieval, Analysis and Compression", Ph.D. Thesis, Columbia University, 1997.
31. A. Soffer, H. Samet, "Retrieval by Content in Symbolic-image Databases", *Proc. of SPIE Conf. on Storage and Retrieval for Still Image and Video Databases IV*, San Jose, 1996.
32. U. Stein, Web page: www.ulistein.de, printed with permission by *Catprint Marketing GmbH*, Germany.
33. S. Ullman, "Filling-in the Gaps: The Shape of Subjective Contours and a Model for Their Generation" *Biological Cybernetics*, Vol.25, pp.1-6, 1976.
34. D. White, R. Jain, "ImageGREP: Fast Visual Pattern Matching in Image Databases", *Proc. of SPIE Conf. on Storage and Retrieval for Still Image and Video Databases V*, San Jose, 1997.
35. L. Williams, D. Jacobs, "Stochastic Completion Fields: A Neural Model of Illusory Contour Shape and Salience", *ICCV-95*, Cambridge 1995.
36. S. Zhu, A. Yuille, "Region Competition: Unifying Snakes, Region Growing, and Bayes/ MDL for Multiband Image Segmentation", *IEEE Trans. PAMI*, Vol.18, No.9, pp.884-900, Sept.1996.